

TOWARDS THE DESIGN,
DEVELOPMENT, AND APPLICATION OF SERIOUS GAMES

By

Declan Andrew McClintock

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science - Doctor of Philosophy
Information and Media - Dual Major

2024

ABSTRACT

Serious games research shows that games can increase engagement and improve learning outcomes over traditional instruction, but the impact of specific elements of serious games has yet to be fully explored across many contexts. Additionally, many existing intervention studies omit the details of the game design and development theory that informed the creation of the games used in the study. This abandons an important level of context surrounding why the games were successful and does a disservice to the field by not propagating useful design theory.

Two issues with existing game design theories are that they do not build fully on top of each other and that they leave out practical guidelines for their use in the design and development processes. This leads to further limiting the spread of useful design theory and limiting its impacts in industry and academia.

The work in this thesis carefully outlines the influence of existing game design theory on the design and development of a game project built to study the impact of the narrative element of serious games. Additionally, this thesis builds a new framework aimed at being more comprehensive, easily built on top of, and with clear practical guidelines for its use.

The main study in this thesis studies the engagement of students playing a single serious game with a cohesive narrative compared against multiple games without a narrative tying those games together. These two cases covered the same set of learning content and differ only in their narratives. The results suggest that either approach is likely to have the same results on engagement but that there is merit to explore learning outcomes further.

This study's research is supported by design research explaining the design theory behind the games developed for and used in the experiment as well as more specific details of the games' production. This allows the results to be understood within a larger serious game

design and development context that will help inform future work.

Additionally, this thesis expands on the lessons learned from the design research and criticisms of existing frameworks to produce the Iterative Game Design and Development framework (IGDD). IGDD provides a broader framework for game design and development with guidelines for its application in practice. The IGDD framework also provides an explanation for how it should be modified and built off of to both allow it to be used across many contexts and to allow future theory building to build collaboratively on top of previous works rather than adjacent to and in assumed competition with other design theory.

This work is dedicated to my Grandfathers Don McClintock and Ray Huber; my Mom and Dad; my family; and my students.

ACKNOWLEDGEMENTS

I must first thank my Grandfather Don McClintock for being a huge inspiration on my life and introducing me to the impact teachers can have on people's lives.

I must next thank my Father and Mother for continually supporting me in my academic endeavours and for always seeking to love and understand me. I must thank my Mother specifically for fueling my PhD journey with a consistent supply of cookies and brownies.

Thank you to Dr. Kenny Moorman and Dr. Mike LeVan for hosting, organizing, and running the summer computer camps at Transylvania University. Without you, I would have never found such a passionate community of like-minded friends and the career path I love.

Thank you to all my committee mentors: Dr. Charles Owen, Dr. Elizabeth LaPensée, Brian Winn, Dr. Esther Thorson, Dr. Wolfgang Banzhaf, and Dr. Yiyong Tong. Your advice has helped me tremendously.

And, of course, thank you to my students. You make all the work worth it.

TABLE OF CONTENTS

| | |
|--|-----------|
| LIST OF TABLES | x |
| LIST OF FIGURES | xi |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Contributions of this Thesis | 2 |
| 1.3 Thesis Structure | 3 |
| 2 Background and Related Work | 5 |
| 2.1 Important Terminology | 5 |
| 2.1.1 Serious Games | 5 |
| 2.1.2 Digital Game-Based Learning | 6 |
| 2.1.3 Games Studies | 6 |
| 2.1.4 More Specific Serious Games Terms | 7 |
| 2.2 Design Research and Frameworks | 7 |
| 2.3 Applied Serious Games Research | 9 |
| 2.4 Narrative in Educational Games | 10 |
| 2.5 Influence on this Work | 13 |
| 3 CircuBot Source Code and Games | 14 |
| 3.1 Game Overview | 14 |
| 3.1.1 Game Hardware | 14 |
| 3.1.2 Game Software | 15 |
| 3.1.2.1 Unity 3D Engine | 15 |
| 3.1.2.2 Game World | 15 |
| 3.1.2.3 Circuit Puzzles | 17 |
| 3.1.2.4 Game Progression and Story | 20 |
| 3.1.2.5 In-Game Assistance | 42 |
| 3.1.3 Learning Objectives | 44 |
| 3.1.3.1 Learning Content in Game | 44 |
| 3.1.4 Student Reception | 45 |
| 4 Design Research from CircuBot | 51 |
| 4.1 Introduction | 51 |
| 4.2 Frameworks for Serious Game Design and Development | 51 |
| 4.2.1 In-Game Elements | 52 |
| 4.2.1.1 The Mechanics Element | 52 |
| 4.2.1.2 The Narrative Element | 53 |
| 4.2.1.3 The Audio Element | 54 |
| 4.2.1.4 The Art Element | 55 |
| 4.2.1.5 The Game Cohesion Element | 56 |
| 4.2.2 Design Elements | 57 |

| | | |
|----------|--|-----------|
| 4.2.2.1 | The Polish Element | 57 |
| 4.2.2.2 | The Technology Element | 59 |
| 4.2.2.3 | The User Interface Element | 60 |
| 4.2.2.4 | The Target Audience Element | 60 |
| 4.2.2.5 | The Balance Element | 61 |
| 4.2.2.6 | The Application Context Element | 62 |
| 4.2.2.7 | The Gameplay/Play Dynamics Element | 64 |
| 4.2.2.8 | The Resulting Experience Element | 65 |
| 4.2.2.9 | The Designer Perspective | 66 |
| 4.2.2.10 | The Player Perspective Element | 67 |
| 4.2.2.11 | The Academic Perspective Element | 69 |
| 4.2.2.12 | The Content Expert Perspective Element | 70 |
| 4.2.2.13 | The Design Cohesion Element | 71 |
| 4.2.3 | Development Elements | 72 |
| 4.2.3.1 | The Playtesting Element | 73 |
| 4.2.3.2 | The Iterating Element | 74 |
| 4.2.3.3 | The Prototyping Element | 74 |
| 4.2.3.4 | The Resources Element | 76 |
| 4.2.3.5 | The Asset Acquisition Element | 77 |
| 4.2.3.6 | The Communication Element | 78 |
| 4.2.3.7 | The Extensibility Element | 79 |
| 4.2.3.8 | The Documentation Element | 81 |
| 4.2.4 | Serious Elements | 83 |
| 4.2.4.1 | The Purpose Element | 83 |
| 4.3 | Summary | 84 |
| 5 | Applying CircuBot: Single Narrative versus Multiple | 86 |
| 5.1 | Introduction | 86 |
| 5.2 | Research Question | 87 |
| 5.3 | Study Design | 88 |
| 5.3.1 | Study Population | 88 |
| 5.3.2 | The Games and Groups | 88 |
| 5.3.3 | Data Collection | 90 |
| 5.3.3.1 | Gameplay Data Collection | 90 |
| 5.3.3.2 | Survey Data Collection | 91 |
| 5.3.4 | Data Analysis Methods | 91 |
| 5.3.4.1 | Gameplay Data | 91 |
| 5.3.5 | Survey Data | 93 |
| 5.3.6 | Post-Survey Data | 94 |
| 5.3.7 | Pre- and Post-Survey Data | 94 |
| 5.4 | Results | 94 |
| 5.4.1 | Gameplay Data | 95 |
| 5.4.2 | Survey Data | 96 |
| 5.4.2.1 | Pre-Survey Data | 96 |
| 5.4.2.2 | Post-Survey Data | 98 |

| | | |
|----------|---|------------|
| 5.4.2.3 | Pre and Post Survey Data | 98 |
| 5.5 | Discussion | 99 |
| 5.5.1 | Interpretation of Results | 99 |
| 5.5.2 | Practical Suggestions | 100 |
| 5.5.3 | Limitations and Future Work | 100 |
| 5.5.3.1 | Engagement | 100 |
| 5.5.3.2 | Knowledge Questions | 101 |
| 5.5.3.3 | Additional Avenues for Future Work | 102 |
| 5.6 | Summary | 102 |
| 6 | An Iterative Game Design and Development Framework | 104 |
| 6.1 | Introduction | 104 |
| 6.1.1 | The Interdisciplinary Nature of Game Design and Development | 104 |
| 6.1.2 | Design, Development, and Practicality | 107 |
| 6.2 | The Iterative Game Design and Development (IGDD) Framework | 108 |
| 6.2.1 | Serious Elements | 110 |
| 6.2.1.1 | Purpose | 110 |
| 6.2.2 | Development Elements | 111 |
| 6.2.2.1 | Resources | 111 |
| 6.2.2.2 | Asset Acquisition | 112 |
| 6.2.2.3 | Prototyping, Playtesting, and Iterating | 113 |
| 6.2.2.4 | Communication | 115 |
| 6.2.2.5 | Extensibility | 116 |
| 6.2.2.6 | Documentation | 118 |
| 6.2.3 | Design Elements | 118 |
| 6.2.3.1 | Target Audience | 119 |
| 6.2.3.2 | Application Context | 120 |
| 6.2.3.3 | Resulting Experience | 122 |
| 6.2.3.4 | User Interface | 123 |
| 6.2.3.5 | Balance | 125 |
| 6.2.3.6 | Gameplay/Play Dynamics | 126 |
| 6.2.3.7 | Player Perspective | 127 |
| 6.2.3.8 | Designer Perspective, Academic Perspective, and Content Expert Perspective | 128 |
| 6.2.3.9 | Polish | 129 |
| 6.2.3.10 | Design Cohesion | 130 |
| 6.2.4 | In-Game Elements | 131 |
| 6.2.4.1 | Mechanics | 131 |
| 6.2.4.2 | Narrative | 132 |
| 6.2.4.3 | Audio | 132 |
| 6.2.4.4 | Art | 133 |
| 6.2.4.5 | Game Cohesion | 134 |
| 6.3 | How to use the IGDD Framework | 135 |
| 6.3.1 | In the Design and Development Process | 136 |

| | | |
|----------|---|------------|
| 6.3.1.1 | Initial Design Ideation and The Start of The Iterative Design and Development Loop | 136 |
| 6.3.1.2 | Inside the Prototype, Playtest, Iteration Loop | 144 |
| 6.3.2 | In Assessing Existing Games | 145 |
| 6.3.3 | In the Design Research Process | 146 |
| 6.4 | Discussion | 146 |
| 6.4.1 | Limitations and Future Work | 147 |
| 6.5 | Summary | 148 |
| 7 | Summary and Conclusion | 149 |
| 7.1 | Contributions | 149 |
| 7.2 | Limitations | 150 |
| 7.2.1 | Experiment | 150 |
| 7.2.2 | IGDD | 152 |
| 7.3 | Future Directions | 153 |
| | REFERENCES | 155 |
| | APPENDIX A: PRE-SURVEY | 166 |
| | APPENDIX B: POST-SURVEY | 174 |

LIST OF TABLES

| | | |
|----------|---|----|
| Table 1: | CRM Results, probabilities of transitions by group, and odds ratios . . . | 95 |
| Table 2: | Gender Self-Identification | 96 |
| Table 3: | Age | 96 |
| Table 4: | Ethnicity | 97 |
| Table 5: | Hours Spent Gaming Daily | 98 |
| Table 6: | Gaming Self-Identity | 98 |
| Table 7: | GEQ Mann-Whitney-U Test | 99 |

LIST OF FIGURES

| | | |
|------------|--|----|
| Figure 1: | Palomino et al.'s Breakdown of Narrative [14] | 11 |
| Figure 2: | Keyboard Controls for Game World | 16 |
| Figure 3: | OR Gate Pickup | 17 |
| Figure 4: | First Circuit Puzzle Solution | 18 |
| Figure 5: | Component Dialogue Box | 19 |
| Figure 6: | Result of Solving the First Puzzle | 20 |
| Figure 7: | Robo Meets CircRat | 21 |
| Figure 8: | Second Room | 23 |
| Figure 9: | Second Circuit Puzzle | 24 |
| Figure 10: | Second Game World Section Overview | 25 |
| Figure 11: | Side Room of the Second Game World Section | 26 |
| Figure 12: | Solution to The Left Circuit Puzzle | 27 |
| Figure 13: | Solution to The Right Circuit Puzzle | 28 |
| Figure 14: | Third Game World Section Overview | 29 |
| Figure 15: | Expression-Based Circuit Puzzle Solution | 30 |
| Figure 16: | Walking Into the Digit Recognizer Room | 31 |
| Figure 17: | The b Correctly Displayed | 32 |
| Figure 18: | Error in the Display | 33 |
| Figure 19: | Completed Code | 33 |
| Figure 20: | CircRat Confronts Robo in The Final Game World Section | 34 |
| Figure 21: | Addition Location | 35 |

| | | |
|------------|--|----|
| Figure 22: | 1 Bit Adder Complete | 36 |
| Figure 23: | Tabs Introduction | 37 |
| Figure 24: | Adder Used in Addition | 37 |
| Figure 25: | ALU Addition Screen Shows Computations | 38 |
| Figure 26: | ALU Subtraction Screen Shows Computations | 39 |
| Figure 27: | A Solution to the Compare Circuit | 40 |
| Figure 28: | Right Arithmetic Shift Screen Shows Computations | 41 |
| Figure 29: | End Game Screen | 41 |
| Figure 30: | Help Tab in The Circuit Puzzles | 42 |
| Figure 31: | Manual Opened in Circuit | 43 |
| Figure 32: | Course Content | 44 |
| Figure 33: | An Unimplemented Game World Section Introducing Karnaugh Maps | 46 |
| Figure 34: | An Empty Unity Scene Demo for Karnaugh Maps Playtesting | 46 |
| Figure 35: | Coverage of the Mechanics Element by Frameworks | 53 |
| Figure 36: | Coverage of the Narrative Element by Frameworks | 54 |
| Figure 37: | Coverage of the Audio Element by Frameworks | 55 |
| Figure 38: | Coverage of the Art Element by Frameworks | 55 |
| Figure 39: | Coverage of the Game Cohesion Element by Frameworks | 57 |
| Figure 40: | Coverage of In-Game Elements by Frameworks | 57 |
| Figure 41: | Coverage of the Polish Element by Frameworks | 58 |
| Figure 42: | Coverage of the Technology Element by Frameworks | 59 |
| Figure 43: | Coverage of the User Interface Element by Frameworks | 60 |

| | | |
|------------|---|----|
| Figure 44: | Coverage of the Target Audience Element by Frameworks | 61 |
| Figure 45: | Coverage of the Balance Element by Frameworks | 62 |
| Figure 46: | Coverage of the Application Element by Frameworks | 64 |
| Figure 47: | Coverage of the Gameplay / Play Dynamics Element by Frameworks | 65 |
| Figure 48: | Coverage of the Resulting Experience Element by Frameworks | 66 |
| Figure 49: | Coverage of the Designer Perspective Element by Frameworks | 67 |
| Figure 50: | Coverage of the Player Perspective Element by Frameworks | 68 |
| Figure 51: | Coverage of the Academic Perspective Element by Frameworks | 69 |
| Figure 52: | Coverage of the Content Expert Perspective Element by Frameworks | 70 |
| Figure 53: | Coverage of the Design Cohesion Element by Frameworks | 72 |
| Figure 54: | Coverage of Design Elements by Frameworks | 72 |
| Figure 55: | Coverage of the Playtesting Element by Frameworks | 74 |
| Figure 56: | Coverage of the Iterating Element by Frameworks | 74 |
| Figure 57: | Coverage of the Prototyping Element by Frameworks | 76 |
| Figure 58: | Coverage of the Resources Element by Frameworks | 77 |
| Figure 59: | Coverage of the Asset Acquisition Element by Frameworks | 78 |
| Figure 60: | Coverage of the Communication Element by Frameworks | 79 |
| Figure 61: | Coverage of the Extensibility Element by Frameworks | 81 |
| Figure 62: | Coverage of the Documentation Element by Frameworks | 82 |
| Figure 63: | Coverage of the Development Elements by Frameworks | 83 |
| Figure 64: | Coverage of the Serious Elements by Frameworks | 84 |
| Figure 65: | Comparison of SG and MG1 | 89 |

| | | |
|------------|---|-----|
| Figure 66: | Comparison of SG and MG2 | 90 |
| Figure 67: | The Continuation Ratio Model | 92 |
| Figure 68: | The Iterative Game Design and Development Framework | 109 |
| Figure 69: | The Iterative Design and Development Process | 135 |

1 Introduction

This thesis covers the design, development, and application of a serious game project meant to teach computer organization and architecture concepts to undergraduate computer science students. This includes a study examining the use of the project in the form of two separate game intervention cases where one group of students played a singular cohesive narrative game covering learning content against a second group which played a series of three games each with differing narratives from the other games. This experiment is covered in Chapter 5. The results of this experiment suggest that there is no significant difference between the two approaches in terms of engagement. Future studies may be able to detect a difference with a sample size in the thousands and they may be able to find differences in deeper measures of learning outcomes for samples in the hundreds.

In addition to this experiment, this thesis also covers: the details of the game project's production in Chapter 3; the design research understandings pulled from the creation of the games from utilizing several relevant game design and development frameworks in Chapter 4; and finally, a broader and more flexible framework for game design and development in Chapter 6.

1.1 Motivation

The impact of games in serious applications, such as education, has intrigued many researchers. Some have looked into the design aspects that make serious games successful [1, 2, 3, 4]. Others have looked specifically towards the application of serious games in educational contexts [5, 6], and some have combined both directions, using their design ideas to inform the serious game development process and the application of their game(s) to prove the merits of their design ideas [4].

The consensus in the literature is that serious games in education are beneficial to student engagement and learning outcomes [7, 8, 9, 10], but the literature suggests more needs to be

done to understand what parts of a game make it successful when others fail [7, 11].

The literature also argues for greater accessibility of educational games so that the games can be reused to confirm results or compare across contexts [12, 13, 10]. Related to this, the existing literature also argues that the context surrounding a game's application in research should be better documented [7]. This would make the possible reasons for success or failure more identifiable.

One interesting part of serious games that some studies have looked into is narrative [14, 15]. While existing work looks into the success of narrative centered games and finds them beneficial in their specific research application contexts [6, 16], more work needs to be done to understand how important narrative is across differing application contexts (different learning content, student backgrounds, etc.) Additionally, existing work has not looked into whether it is better to use only one cohesive narrative to tie together learning content or to use multiple narratives across the same learning content. Are students more engaged when a cohesive narrative leads them between all their content? Or are students more engaged when a variety of narratives are used? This thesis aims to answer these questions in an undergraduate computer science course context while also providing the source code of the games, the discussion of the design theory behind the games, and the development of a larger framework to address the issues recognized in prior work while providing a strong stepping stone for future work.

1.2 Contributions of this Thesis

The research conducted for this thesis addresses recent interests and concerns of the literature. The code for the serious games used in the research is available publicly [17] to allow others to confirm results or apply the games in new contexts.

The experimental part of this research in Chapter 5 looks primarily into the impact of narrative when used as a cohesive tie between learning content. The design, development and application context of the serious games used in the research is provided as well to

contextualize the results.

The contributions of this thesis are:

- A set of serious games to teach computer organization and architecture concepts, hereto referred to as the computer architecture games, which future researchers may use and adapt.
- A detailed discussion of the design theory used in creating the games used in the experiment. This provides additional context to understand the study which uses the game and provides a strong argument for developing a more comprehensive framework that combines concerns across existing works.
- A study of the effectiveness of a cohesive narrative as compared to disparate narratives in an educational serious game intervention within an undergraduate computer science context. The conclusions of this study suggest that the two approaches are likely the same for student engagement outcomes and that more should be more done to study deeper measurements of learning outcomes in these two cases.
- A broader iterative game design and development framework built from the lessons learned from the aforementioned contributions and criticisms of existing academic frameworks for game design and development.

1.3 Thesis Structure

The Background and Related Work chapter (Chapter 2) provides important high level terminology for understanding the work in this thesis and the broader context in which it fits. This chapter also clarifies the differences between several related terms and fields of research to clarify where exactly the work in this thesis fits. This chapter also connects the existing research to the work presented in this thesis.

The source code and the games in the CircuBot project [17] are covered in Chapter 3. This includes the details of how the games are played, the challenges the game presents,

the overall narrative structure connecting learning content, navigating the game world, the pacing, and other aspects of the games. This chapter also includes a brief overview of how the games were viewed from the perspective of students who played them.

The Design Research from CircuBot chapter (Chapter 4) explains the game design and development theory behind the creation of the computer architecture games as well as parts of the design of the games where the utilized theory did not have descriptions to support the development of the games. The result of this chapter is a richer contextual understanding of the design/development thinking and process behind the creation of the games that allows the utilized frameworks to be critiqued and better understood.

The Applying CircuBot: Single Narrative versus Multiple chapter (Chapter 5) goes through the full experiment comparing applying the computer architecture games to compare the single narrative versus multiple narrative approach when it comes to educational games' impact on student engagement in an undergraduate computer science classroom. The results suggest that both approaches are likely to yield similar results.

The An Iterative Game Design and Development Framework chapter (Chapter 6) presents a broader framework for game design and development constructed from the lessons learned in the work described in the prior chapters. The framework presented provides a more comprehensive framework that combines the guidance from existing theory and provides practical guidance for using the framework.

This thesis ends with a brief Summary and Conclusions chapter in Chapter 7 that pulls together the summaries of the main chapters, briefly restates the contributions of this thesis, explains the limitations of the work presented, and details an overview of future directions leading forward from the work presented in this thesis.

2 Background and Related Work

Existing work in the field of serious games research covers both design research on serious games as well as their impact in interventions. Design research has produced many frameworks and methodologies pointing to particular areas of serious game design that experts consider important. Research in interventions typically looks at effects on student engagement or learning outcomes. A set of these interventions focus on the narrative of serious games.

To understand this research around games, it is also important to understand the broader terminology used to describe various areas of research involving games.

2.1 Important Terminology

2.1.1 Serious Games

Serious games are games that take on an additional focus beyond being entertaining [2, 3, 18]. This includes educational goals, training, influencing political opinions, and so on. This does not mean that entertainment games cannot be used for serious purposes such as having students play *Civilization IV* [2005] as part of a history class [20]. The difference is that a serious game is specifically designed and developed around some additional goal. *Civilization IV* [2005], while borrowing from historical conversations and cultural contexts, was not aimed at thoroughly educating its player on those topics or even being used as an educational tool by instructors. *When Rivers Were Trails* [2019] by comparison was designed and developed to be used as an educational tool in addition to being entertaining [22, 23]. *When Rivers Were Trails* [2019] has the player directly engaging with the Indigenous perspectives on history and living, as the player character, through specific historical moments [23]. So, *When Rivers Were Trails* [2019] would be considered a serious game while *Civilization IV* [2005] would not be. As long as there is an additional goal alongside entertainment, a game

can be considered a serious game.

Some examples of serious games are the products made by Age of Learning, such as ABCmouse which includes a wide assortment of games meant to teach children aged 2 to 8 reading, math, art, and so on [24, 25].

2.1.2 Digital Game-Based Learning

Digital game-based learning (DGBL) describes learning that involves some digital game being played [26]. For DGBL, the game involved must be digital and the research focuses stem directly from the expectations around those described as "Digital Natives" [26], learners who have grown up naturally surrounded by technology and are assumed to benefit from digital learning.

In the more general serious games case, the game need not be digital. For example, the educational games *Race to the Moon* [2019] [28] and *Miner Madness* [2017] [30] are physical games without a digital component that have the goal of being both entertaining and educational, the educational purpose being learning about the space race and programming logic respectively.

All games meant for DGBL fit under the term Serious Game, and the computer architecture games created for the research work done for this thesis fit underneath both terms.

2.1.3 Games Studies

Games studies is an interdisciplinary field of research focused on the cultural impact of games of various kinds and the relationships between games and other cultural phenomena [31]. Games studies looks at games as a whole and their impacts, and the games involved need not be intended to affect societal/cultural change or be educational or otherwise have a purpose beyond entertainment as they would in serious games research. Nieborg and Hermes [2008] outline some example questions that come out of game studies:

...what is the game like? What pleasure and what significance does it have for

players? In what kind of cultural context does it function? How does it represent the world; what morality is offered?... What kind of labour and labour relations are involved in producing both the hardware and software? [31]

These questions lean towards broader cultural points of interest than the research around serious games. Serious games research typically is more interested on improving the design and development of serious games [2, 3, 32] and/or researching their applications [33, 10, 7, 11, 34].

The work in this thesis fits best underneath the focus of serious games research and the questions that area seeks to answer. The work in this thesis does not aim to answer the broader cultural questions typically focused on in games studies.

2.1.4 More Specific Serious Games Terms

Underneath serious games there are also *persuasive games* (games meant to persuade) [35], *games for change* (games meant to change behaviors and attitudes) [36, 37], and *educational games* (games meant to educate or be used in education) [7]. All of these terms fit under the umbrella of serious games, and in works where they appear the keyword *serious games* often appears as well [7, 36, 35, 32, 38]. These terms are more restrictive in what could be included in them. However, they all can benefit from serious game design and development theory informing their production and application. As such, the Iterative Game Design and Development framework presented in this thesis is applicable to works that fall under any of these terms.

The serious game application study using the computer architecture games presented in this thesis falls under serious games and educational games specifically.

2.2 Design Research and Frameworks

Design research refers to the study of the design, development, and evaluation of a product [39, 40]. The benefit of design research is a more detailed communication of the thought

process behind the decision making in the creation of a product. How did the designer do their work? What influenced their decisions? Did the designed product achieve its goal? These kinds of questions are what design research seeks to answer [39].

The field of serious games research applies design research ideas in the formation of the frameworks and methodologies researchers have developed, proposed, or tested. Frameworks and methodologies for serious game design and development detail important elements of the design and development process for serious games. For example, the Mechanics, Dynamics, and Aesthetics framework (MDA) details how a game can be viewed as something a designer creates and a player consumes [1]. The designer creates the mechanics (the code and algorithms that run the game), which are then turned into dynamics (the behavior of the game mechanics when the game is played and input received), which are then perceived by the player and imparts some aesthetics (the emotional responses evoked in the player) [1].

No framework or methodology currently covers all aspects of the serious game design and development process. However, there are some commonalities across various frameworks. For example, a concern with the mechanics, defined as what the player does in the game from moment to moment, is common across MDA [1]; the Design, Play, and Experience framework (DPE) [2] which builds on the MDA framework; the Serious Game Design Assessment framework (SGDA) [3]; and the Gameplay Loop Methodology (GLM) [4].

It is also important to recognize that not every framework covers the same elements. For example, DPE points out how the technology a game will be played on affects what is feasible for a game's design [2]. The other frameworks do not discuss technology as a concern.

If only one framework is employed in the design and development of a serious game, the creator risks missing important elements of the process not covered by that framework. Therefore, it is important for any creator of a serious game to review several frameworks and extract all elements of the design and development process to inform their own work.

The design research work from the computer architecture games covered in this thesis

addresses the use of multiple frameworks in creating the games. This includes discussing the advantages from using multiple frameworks as well as some insights into what would be advantageous to do when creating a new framework.

2.3 Applied Serious Games Research

Serious games have been researched within educational contexts with varying degrees of success. The consensus in the literature is that they are beneficial to student engagement and learning outcomes [41, 7].

Despite this consensus, there are still interventions and published papers where games have been shown to not be favorable, either just as good or worse, than other instruction methods [7, 8, 9, 10]. Because the majority of papers support the benefits of games used in education, one important question arises from the papers where game interventions are unsuccessful: Why?

The same literature reviews and meta analyses that show the majority support for the benefit of games in education provide a general direction to move towards in answering the question of why some games fail in their educational purpose while others succeed. Sara de Freitas's literature review of educational games ends with an argument that "more active design studies are needed to ensure that the best interests of the learner are met in different contexts" and that one of the next areas for research is looking into the best ways to implement games in the classroom [7]. This concern with the application context of games in education is shared by educational researchers. Marjaana Kangas et al. point out in their literature review of educational games in the classroom that the teacher's role in game-based learning is sometimes overlooked despite its importance [11].

The recommendations the literature reviews provide suggest that any serious game research should include detail of the context of the game's design and application, or otherwise provide an additional design research paper that does. Knowing the application context and design details of the serious games will allow future researchers to better innovate on what

was done and compare games across contexts more easily. The research work done with the computer architecture games used in this thesis covers both the design theory that informed their creation and their application in a study comparing a singular narrative game intervention against multiple games with varied narratives intervention.

Another problem in the application of serious games in education recognized by experts in the field is *reproducibility*. One literature review from 2018 of serious games for programming found that not all of the research games used in the literature were available to be played [10]. As the authors of that same review point out: “The lack of access is problematic for both researchers developing computer science educational games and instructors seeking to find effective learning tools.” [10]

This concern is shared by the Open Educational Resources (OER) movement and the idea of Open Educational Games (OEG). OER are: “teaching, learning, and research resources that reside in the public domain or have been released under an intellectual property license that permits their free use or re-purposing by others.” [12] Similarly, OEG are: “games designed according to OER principles emphasizing the challenges and perspectives related to their development and their effective potential of use in education.” [13] OER and OEG provide a clear guide to resolving part of the issue of reproducing game study results by making games used in educational research accessible.

The source project and the source code for the computer architecture games used in the work within this thesis are available through a public Google drive [17] to allow the games to be applied in similar situations to address reproducibility, in new contexts to expand the research, or to be used to help educate and have continued impact.

2.4 Narrative in Educational Games

Narrative on its own contains a lot of power in enhancing memory, motivation, and engagement through the building of mental models and facilitating inferences between the parts of a story [42, 43]. Children can form attachments to stories which then shape their cultural

BREAKDOWN OF THE NARRATIVE DEFINITIONS COMMON FEATURES

| Features | Ricoeur | Manovich | Salen and Zimmerman | Marczewski |
|-----------------------|---------|----------|---------------------|------------|
| Actor | | X | X | |
| Choice or Possibility | | X | X | |
| Interactivity | | X | X | |
| Narrator | | X | X | |
| Plot | | | X | X |
| Sequence of events | X | X | X | |
| Space | X | | X | |
| Story/Data | X | X | X | X |
| Time | X | | X | |
| User Experience | | X | X | |

Figure 1: Palomino et al.’s Breakdown of Narrative [14]

beliefs [44] and narrative building presents one of the unique tools humans use to understand and organize the world around them [45]. Narrative in educational games serves a similar role.

Narrative in the gaming context can take on many different definitions. Fortunately, Palomino et al. put together a breakdown of existing narrative definitions, shown in Figure 1, along with a concise definition [14].

The breakdown provided by Palomino et al. describes narrative in games as having the following features: the actor (the player or student), choice (the player’s ability to make decisions which determine their narrative experience), interactivity (the game’s response to input), sequence of events (a logical progression of story points), space (the environment in which the player character resides), date (when in time the game takes place), time of interaction (how long the game is played for or takes to finish), and user experience (is the game enjoyable or not) [14].

The definition Palomino et al. settle on is:

”The Gamification narrative element can be understood as the process in which the user builds his own experience through a given content, exercising their freedom of choice in a given space and period of time, bounded by the system’s logic.” [14]

While this definition of game narrative is concise, it risks leaving out an important distinction, the difference between embedded narrative and emergent narrative as discussed by Salen and Zimmerman [46]. *Embedded narrative* is effectively the narrative as created by the designer [46]. It acts as the player’s reason for taking part in the events in the game, and it is presented to the player before they interact with the narrative in the game.

Emergent narrative is the player’s individual narrative experience built while playing the game [46]. Every player may experience the same events in a game, but they will react to them differently and thus they come away from game play with different emergent narratives.

A solid narrative is expected to be important in serious game design because existing literature draws a link between strong narrative and engagement as well as learning outcomes [15]. Jemmali et al. looked at the difference between a “Rich Narrative” (including characters and communicating story reasons for in game actions) and a “Light Narrative” (without characters or story dialogue) [47]. They found that, in general, participants who played the Rich Narrative version of their game had better outcomes than their Light Narrative counterparts [47]. However, they also found that participants in both groups mentioned either the coding and learning (54%) as what they liked most or the puzzles and challenges (21%) as what they liked most [47]. This led the authors to consider that perhaps the narrative elements play more of a supporting role in their outcomes, and that perhaps their observed benefits are caused more by design decisions, which differed from previous work, than by their narratively different treatments [47].

This raises the concern of just how important narrative is in serious game design. Is its most important role to support a reasoning for player’s actions? The narrative is relatively light in one successful game intervention study that teaches mechanical engineering concepts [5]. The narrative in the game is simply to get a car to behave a certain way, which the authors present as an improvement over the traditional presentation of the same learning concepts as just math problems with little or no applicable use context [5].

Is narrative motivating for every student? In Dickey’s research with the narrative game

Murder on Grimm Isle, Dickey found that two of their participants disengaged from the story [6] and opted instead to take place in what could be called *transgressive play*, playing against the game’s own expectations of the player [48]. Dickey states that these players: “expressed more interest in . . . action games, as opposed to adventure-style games.” [6] This is unsurprising given that every player has a different background, and the idea that players play games for various reasons has existed for some time [49].

2.5 Influence on this Work

The issues in existing research with reproducibility, recognizing the context of the application of serious game interventions, and the continued improvement of design and development frameworks shape the goals of the research in this thesis: The computer architecture games developed for use in this research are publicly available [17]; The design details, utilized design theory, and context of their application is explained across Chapter 3 and Chapter 4; and a more comprehensive game design and development framework is presented in Chapter 6.

3 CircuBot Source Code and Games

The primary research focus of this thesis includes the design and development of computer architecture games with two different treatments. One treatment uses a single game with a cohesive narrative structure throughout. The other uses multiple games with the same mechanics as the first treatment, but with three different narratives breaking apart learning content. The game treatments have been the subject of a study designed to compare the effectiveness of the two treatments on student engagement.

The source project is referred to as CircuBot on account of the single game's theme. The source files of the project are available on a public Google drive link [17].

3.1 Game Overview

The games used in the study were designed and developed for use in Michigan State University's Computer Organization and Architecture course, CSE 320. The games cover a subset of the course's learning content. The games were developed using the Unity 3D game engine and editor [50]. Because CSE 320 is taught online, the games were designed and developed for web browsers using WebGL.

3.1.1 Game Hardware

Because the games used in the study were developed in Unity, various builds for different platforms were available during development. The games can be built and run on MacOS and Windows. However, the primary build type of the games used in the research of this thesis [51, 52, 53] is the WebGL build.

Within the applied research, the player must have access to a web browser and a link to a website hosting the game in order to run the game. Additionally, both a keyboard and mouse are required to play the game. The game does not have controller support.

3.1.2 Game Software

The games used in the study were built on the Unity 3D game engine. All code written for the games is in C#. The rest of this section will outline why the Unity 3D engine was chosen for development, what the controls are in the game world as well as in the circuit puzzles, the game progression and story for the cohesive narrative game, and the in-game assistance provided to players.

3.1.2.1 Unity 3D Engine

The Unity 3D engine was chosen for development for several reasons. The Unity engine was familiar to the primary developer of the game. Also, the Unity 3D engine is free to use under the circumstances of this project. The Unity engine is well documented and robust. The Unity manual and scripting Application Programming Interface (API) are rather extensive, which makes Unity easy to develop with. Additionally, Unity provides many development quality of life bonuses, such as the easy import of art assets and a functional physics engine.

3.1.2.2 Game World

The games used in the study have two different areas of gameplay with distinct controls. One of these areas is the game world. The game world areas of the game have the player navigating between circuit puzzles by moving around and jumping. The player uses the W, A, S, and D keys or the arrow keys to move the player character along the X, Z plane. The player uses the space bar to jump up and down on the Y axis.

All X, Z movement is done relevant to the location of the camera so that the player character moves to the left, right, forward, or backward relative to the player's view of the character on screen.

The player does not have control of the camera during the 3D platforming sections. Instead, the camera shifts in-between defined locations in the level depending on where the player is in the level. This was done to lower the barrier of entry for players unfamiliar

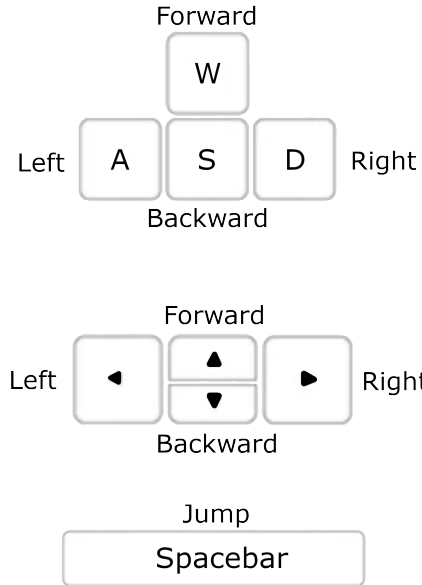


Figure 2: Keyboard Controls for Game World

with video games as it removes the need to learn how to control a camera, thus simplifying interaction.

The player can find and collect circuit pieces, such as the OR gate shown in Figure 3, while navigating the game world. The circuit pieces are added to the list of pieces available to the player when the player picks up a circuit piece. The circuit pieces include some polish on them, such as rotating the pickup to make it enticing for interaction. The circuit piece pickups destroy themselves after they have been collected. Each circuit pickup also adds a corresponding manual page to the in-game manual after collection.

The player can enter circuit puzzles from the game world. Transitioning to a circuit puzzle involves the game's dialogue system first displaying one of two dialogues to the player depending on whether or not they have met the requirements to enter the puzzle. Each circuit puzzle has requirements for what component types the player must already have as well as any *state switches* that must already be on for the requirements to be met.

A *state switch* is a recorded yes (on) or no (off) for has the player done X thing yet. *State switches* will be discussed in more detail in the game progression and story sub-section. If the player has not met the component and state switch requirements, then the dialogue shown

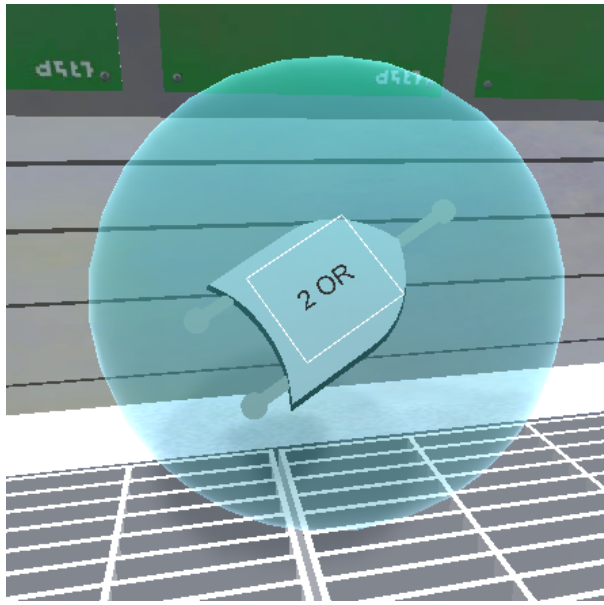


Figure 3: OR Gate Pickup

to the player corresponds to a hint on what the player should do to meet the requirements. If the player has met the requirements, then the dialogue displayed indicates that the player is ready for the impending puzzle and the player is transitioned into the circuit puzzle.

3.1.2.3 Circuit Puzzles

The circuit puzzles represent the second area of gameplay for the game. While they are accessed from the game world, the puzzles themselves use a different set of controls. The first circuit puzzle and its solution are shown in Figure 4.

The controls in the circuit puzzle section are:

- hold down middle mouse button (scroll wheel) and move the mouse to pan, or hold control and left click and move the mouse to pan the view
- scroll up and down on the mouse wheel to zoom in and out, or hold alt and left click and move the mouse forward or backward to zoom in and out
- left click and drag the images of circuit components from the side bar on the left of the screen to create them and release the drag to place them in the circuit

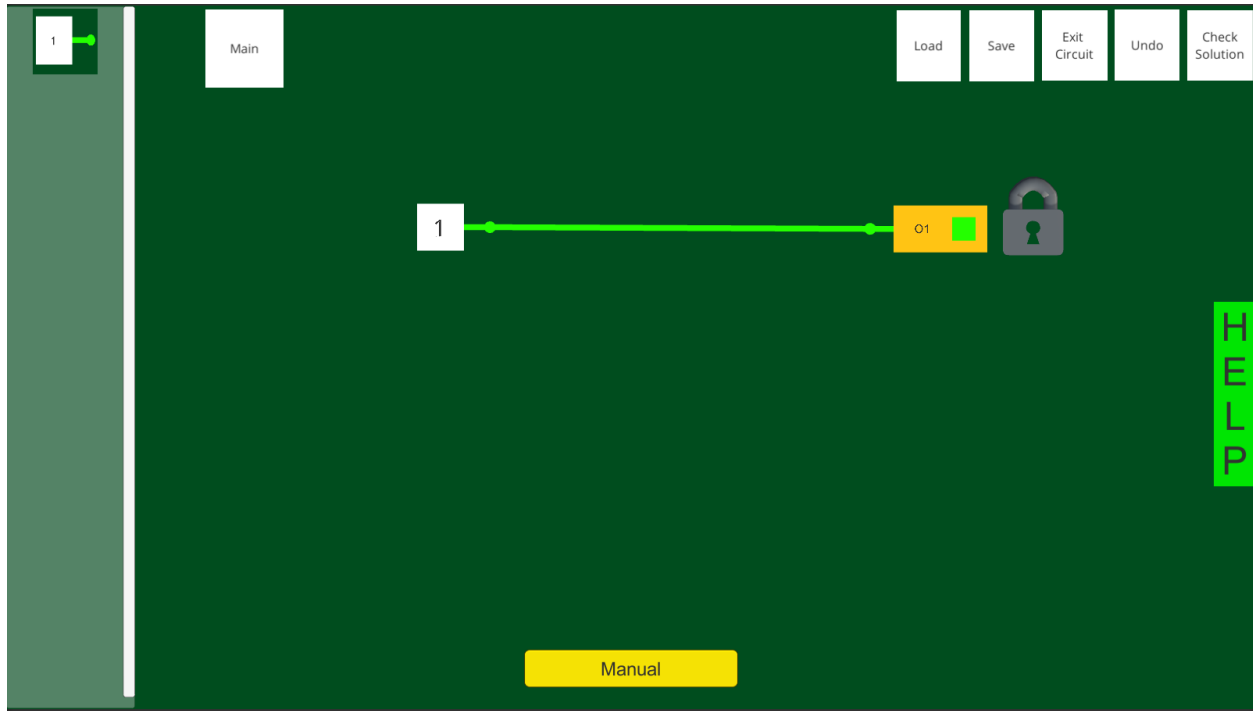


Figure 4: First Circuit Puzzle Solution

- left click and drag on the exposed circular part of an input or output to a component to begin creating a wire and then release while over an output (when dragging from an input) or input (when dragging from an output) to create a wire connecting them
- left click and drag a non-static component (in white is non-static, static is in orange) to move it around
- left click and drag anywhere to begin making a box selection, after a box selection is made it can be moved the same way a single component is
- double clicking a component opens a dialogue box where it can be renamed and more information about it is displayed, see Figure 5
- left click the load button to load the last saved version of the circuit puzzle
- left click the save button to manually save the current circuit puzzle solution
- left click the exit circuit button to return to the game world

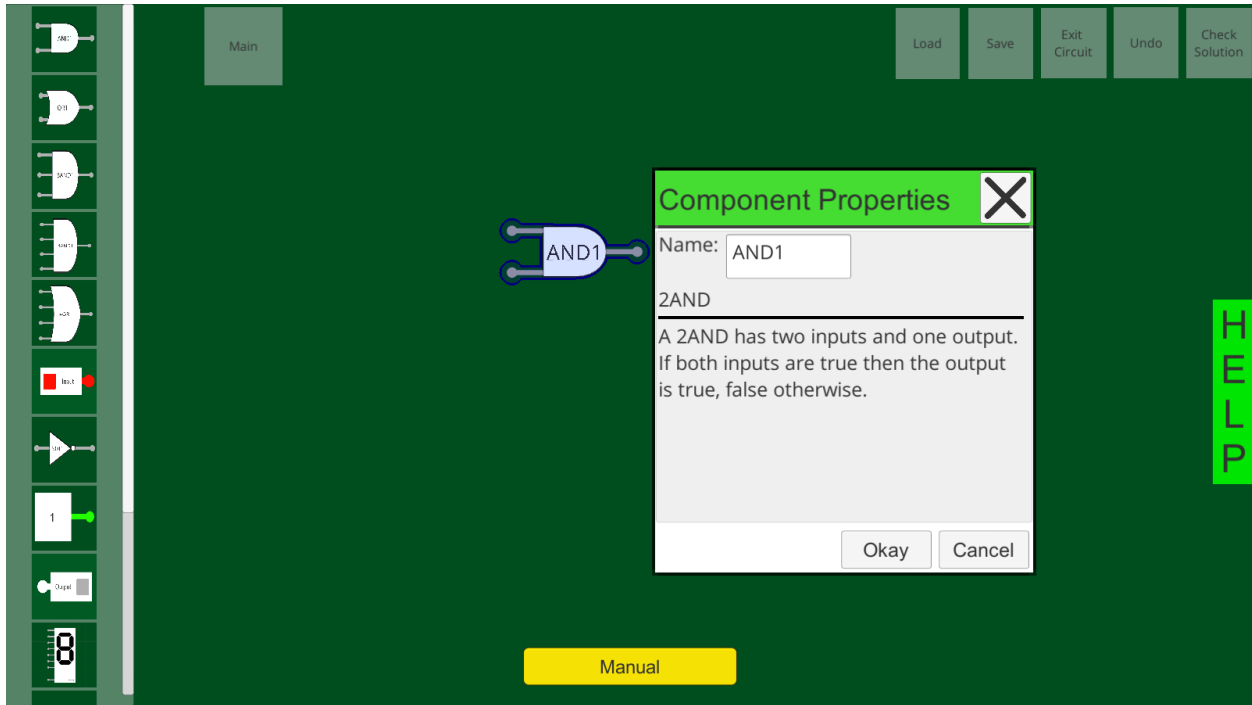


Figure 5: Component Dialogue Box

- left click the undo button to revert the last change made or the next change in the sequence of changes made
- left click the check solution button to make the game check if the circuit solution created solves the puzzle

The circuit puzzles are the part of the game that most closely aligns mechanics with the purpose of the serious game. The player solves the circuit puzzles by applying the knowledge they have learned in the game to build circuits that meet certain specifications.

Solving the puzzles also causes changes in the game world. For example, the first puzzle in the game introduces the *One* component which always outputs an on signal, see Figure 4. This puzzle shows the player how to drag and drop components into the circuit and how to connect components using wires.

When the player thinks they have completed the puzzle they can click the *check solution* button to have their puzzle solution checked for completeness. The player will watch as the puzzle goes through checking their solution, turning off and on certain inputs while checking



(a) The First Room



(b) Opening the First Door

Figure 6: Result of Solving the First Puzzle

the resulting outputs of the player's solution.

If the player's solution fails, then they will receive a window with feedback on what went wrong. If the solution succeeds the player receives an animation of a lock unlocking and they are brought back to the game world.

The result in the game world of completing the very first puzzle in the game is an animation of the door blocking their progress opening, rewarding the player with more space to explore which can be seen in Figure 6.

3.1.2.4 Game Progression and Story

The game progresses by cycling through game world sections and circuit puzzles connected by an overarching narrative. In the single cohesive narrative game, there is a single narrative that ties together all three sets of learning content targeted by the research. In the multiple game case, three separate narratives are used in each game (one per game of the three games) to tie together only the content within each individual game and not between them.

The basic overview of the story in the cohesive narrative game is that the player is a robot sent to investigate the power source of an alien ship so that it can be replicated for the robot's civilization.

There are three characters in the story that fill different roles. The player character is Robo, who is chipper and always ready to help. HQ is Head Quarters. HQ acts as a



Figure 7: Robo Meets CircRat

friendly mentor character to Robo, providing the necessary motivation dialogue for Robo and sometimes providing important tips. CircRat is the antagonist in the game. CircRat attempts to foil Robo's progress by eating circuits and vowing to eat the ship's power source before the player can copy its design.

CircRat provides a tie between narrative and mechanics. CircRat will sometimes steal component pieces from the player when they enter certain puzzles. This allows for a way to control what the player can use to solve problems. This is used in the game to make specific puzzles easier by preventing the player from being overwhelmed with options. Figure 7 shows a dialogue between CircRat and Robo.

The game begins with a circular fade out into a dialogue between HQ and Robo in which the basics of who Robo is and why they are on the alien ship are outlined. The player is introduced to two important aspects of the game, pickups and circuit puzzles, in the first room of the game.

The first room is small and presents the player immediately with a door and a pickup

near the center of the screen. If the player walks up to the door before collecting the pickup, they will receive a dialogue of Robo talking to themselves about how the door is broken and that they seem to be missing what they need to fix it. This dialogue also ends with Robo prompting the player to look around for any circuits first.

A similar dialogue plays out whenever the player approaches any obstacle they are not ready solve. When the player picks up the ONE circuit component pickup that is next to the door, Robo excitedly exclaims that this is a new circuit for them to use and prompts the player to check their manual to learn more about what they just found. This dialogue is similar for every pickup the player finds. When the player approaches the broken door in the room after getting the pickup, Robo states that they are ready to solve the problem and the game does a circular fade in to load the corresponding circuit puzzle for the door.

When the circuit puzzle is finished loading, the game does another circular fade out to reveal the puzzle screen. Robo relays important information about how to interact with circuit puzzles to the player in this first puzzle of the game. This puzzle introduces how to drag circuit components into the puzzle, how to connect them to each other by dragging wires out of their inputs and outputs, how to save a circuit, load a saved circuit, undo a change made, and how to check the solution. Once the player has solved the puzzle, the game does a circular fade in and out again to load back into the game world location the player was when entering the puzzle.

Upon returning from this first puzzle, Robo exclaims that the door has been fixed and now they can continue exploring. After that dialogue, the door animates open by sliding out of view to reveal the next area for Robo to explore.

The second room of the game presents the player with an electrical field, the ZERO circuit component pickup, the OR circuit component pickup, and a console as shown in Figure 8. The console and electricity act in the same limiting fashion as the door from the first room. Robo is shocked and forced back if the player tries to run through the electricity. Robo provides a dialogue to communicate this.

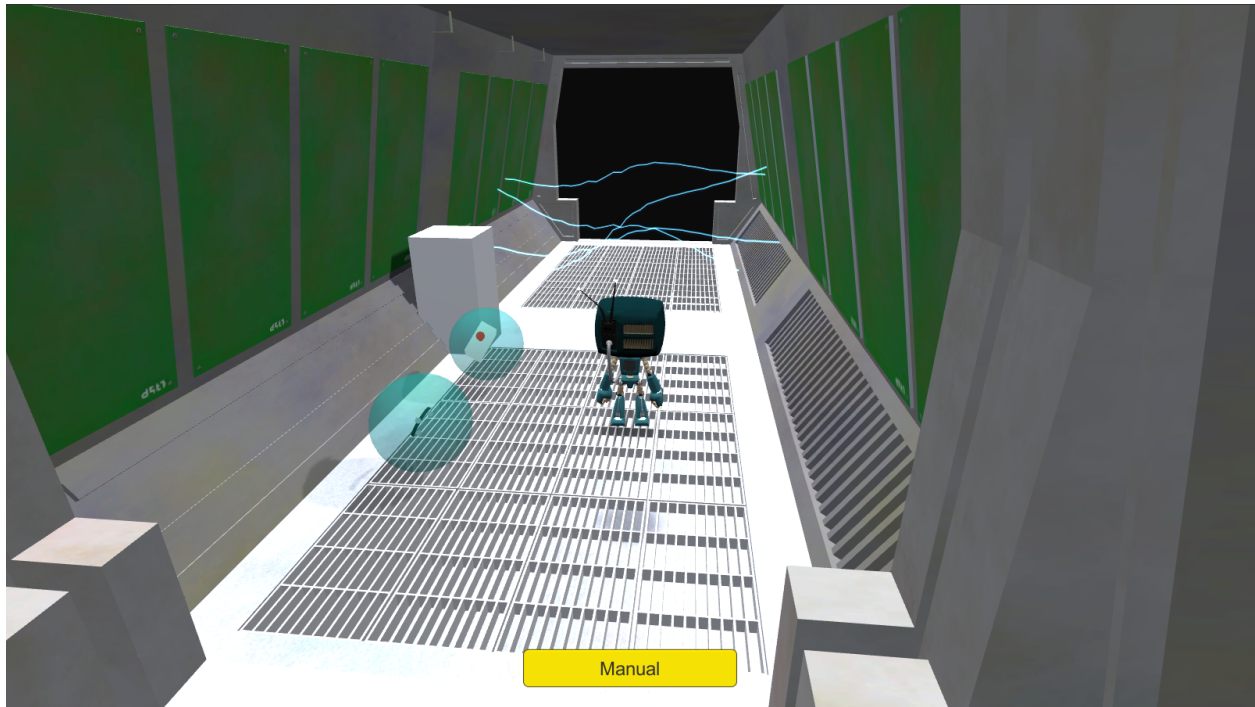


Figure 8: Second Room

If the player interacts with the console without first picking up the circuit components in the room they will be prompted to explore the room. If the player interacts with the console after getting both of the circuit components in the room, they will be loaded into the next circuit puzzle.

The second circuit puzzle, shown in Figure 9, of the game introduces additional basic interaction controls for the circuit puzzle sections. This puzzle requires the player to get an OFF signal into the output to turn off the electricity that is flowing in the game world. The puzzle begins with ONE components already sending ON signals through a *static* OR to a static OUTPUT component. Static pieces cannot be moved, removed, or have their existing wires deleted. The player must use the delete functionality to remove the one components. Then the player uses what they already know to drag in ZERO components and connect them to the OR component to get the desired output.

After solving the second circuit puzzle, the player returns to the game world and Robo exclaims that the electricity has turned off and they can proceed. This is the core gameplay

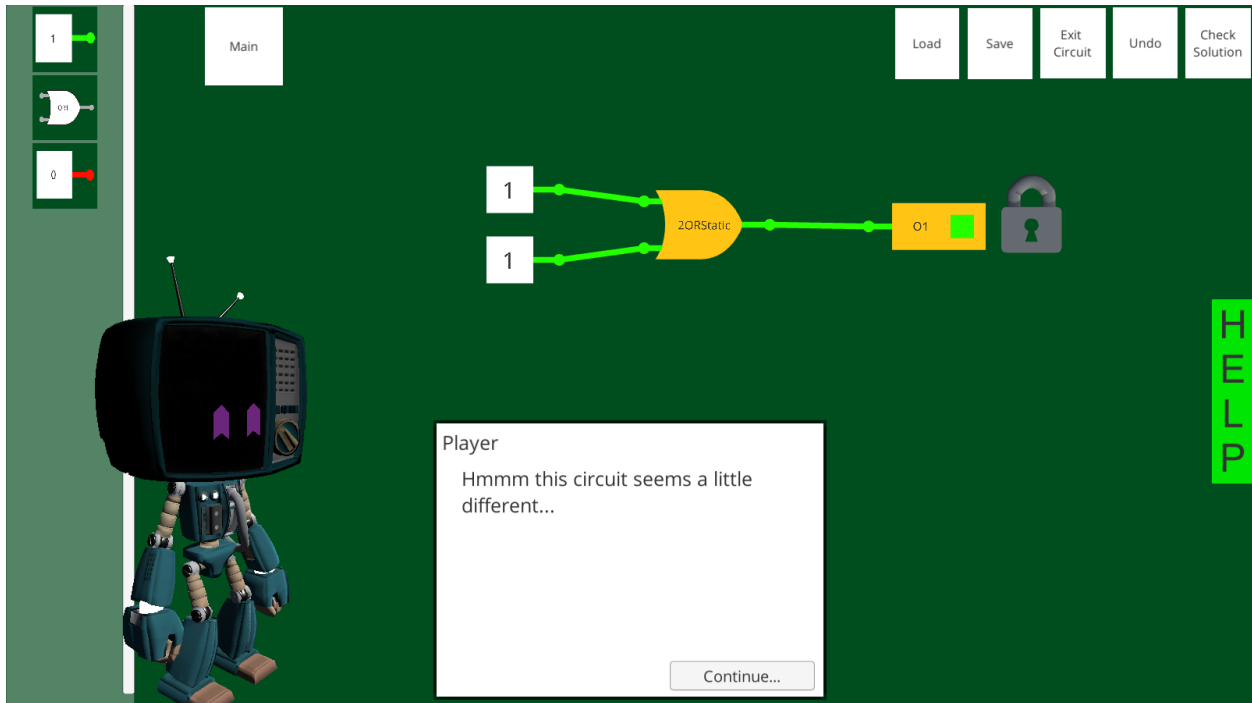


Figure 9: Second Circuit Puzzle

loop. The player is presented with a game world that limits their progression, they explore for components they need to solve the limitation, they find the components, they return to the source of the limitation in the game world, they load into a circuit puzzle, they solve the circuit puzzle, they return to the game world and are rewarded with more to explore. Story and narrative moments are spread throughout this loop to provide additional context and motivation for the player's actions.

The player is loaded into a new section of the game world when they proceed past the area that was previously blocked. This is done using the same circular fade in and out that masks all loading times in the game. This new section of the game world is slightly larger and slightly more complex than the first.

The second section of the game world consists of a long hallway with a door at the end and one side room as shown in Figure 10.

The dialogue presented to the player at the door prompts the player that there are two locks on the door so the player must find some way to open them. When the player

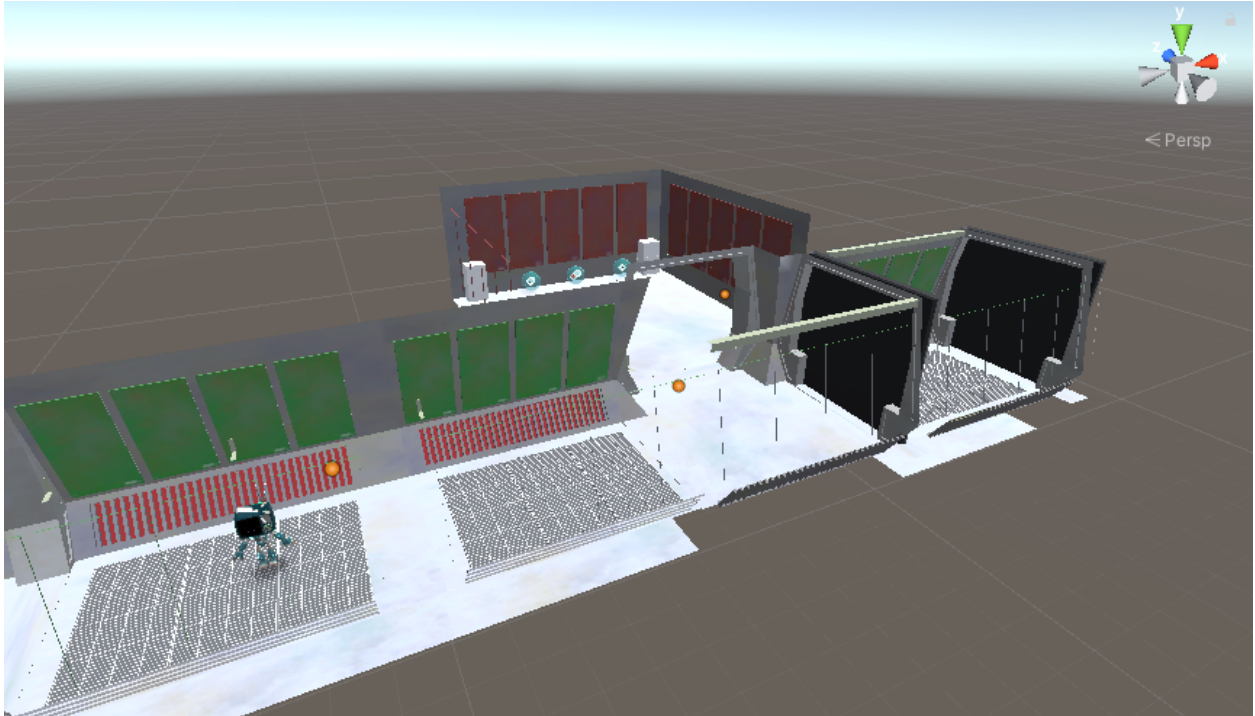


Figure 10: Second Game World Section Overview

approaches the open side room, they are introduced to CircRat through dialogue. This dialogue sets up the conflict between Robo and CircRat and reminds the player of their overall goal.

After the dialogue with CircRat, the player is able to explore the rest of the side room. Within this room there are two consoles, each of which opens a lock on the door when completed. The room also has three new component pick ups for the player to collect: the NOT component, the INPUT component, and the AND component which can be seen in Figure 11. The player needs to pick up at least the NOT component in order to enter the circuit puzzle associated with the console on the left side of the room. To access the console on the right side, the player needs to collect all of the component pickups in the room and complete the circuit puzzle associated with the left console. Dialogue prompts and hints are implemented for every possible interaction in this section to guide the player in the right direction. For example, if the player completes the first console and goes to the door, the player is informed by Robo that one of the locks has been unlocked but the other is still



Figure 11: Side Room of the Second Game World Section

locked so they should keep exploring.

When the player accesses the left console to complete the associated circuit puzzle, they are greeted by CircRat. The dialogue that ensues between Robo and CircRat conveys to the player that CircRat is stealing some of Robo's components, though Robo will get them back once the puzzle is completed. The goal of this puzzle is for the player to continue familiarizing themselves with the controls and to introduce them to the NOT component (inverter). The player is told by Robo that they need to make the output for this circuit be ON.

The solution to this is shown in Figure 12 and requires the use of the only two components available to the player in this puzzle, the ZERO and NOT components. The player can move on to the console on the right after solving this puzzle and collecting any remaining pickups in the room.

The player meets CircRat again after loading into the circuit puzzle attached to the right console. The conversation between Robo and CircRat introduces this puzzle as more

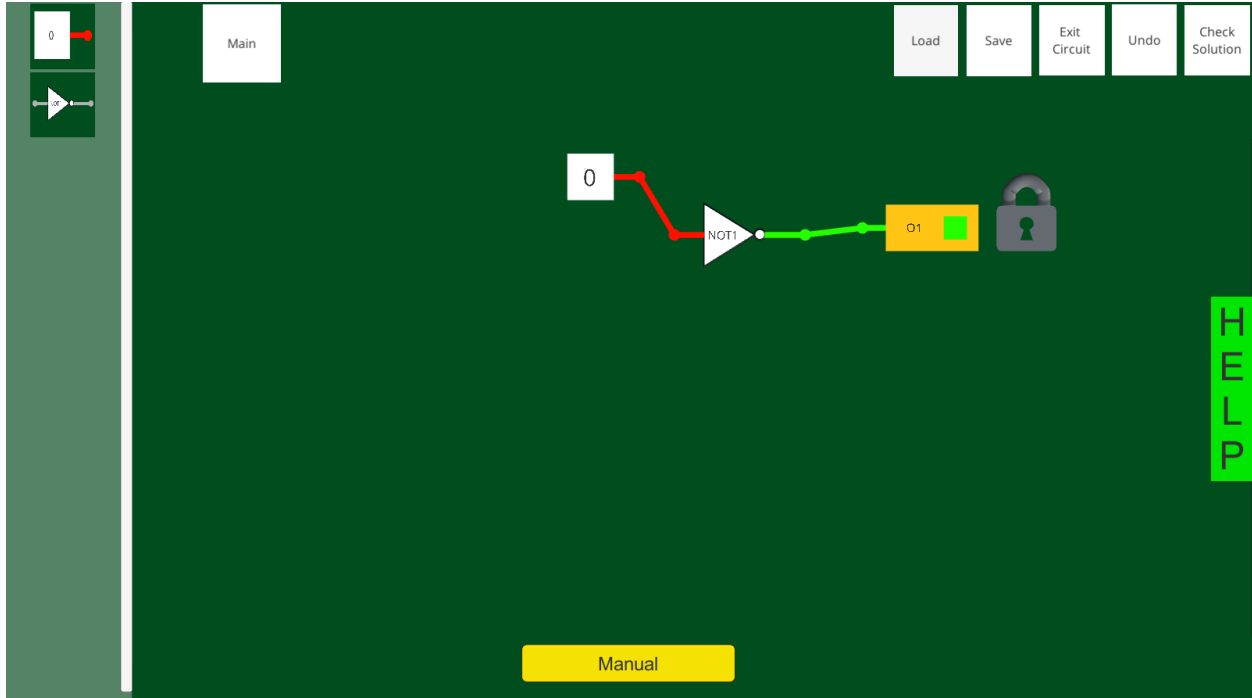


Figure 12: Solution to The Left Circuit Puzzle

challenging compared to the others Robo has already finished. This conversation introduces a few key points to the player. The first key point is that this circuit and future circuits will require the player to develop a circuit that takes specific inputs and gets specific outputs beyond just ON or OFF. The second key point is that the player has a help tab on the side of the screen they can open at any time for more details on the puzzle they are solving. This dialogue specifically does not explain the puzzle further in order to force the player into using the help tab at least once in order to receive the needed explanation and hints to solve the puzzle.

The solution to this puzzle requires the player to develop an understanding of the INPUT and AND components they just collected to make the circuit function as an AND component does. The simplest solution to this is to use the AND component directly with two INPUTs as shown in Figure 13. After completing this puzzle, the player is able to open the locked door and move on to the next section of game world.

This third section of the game world consists of a long hallway, one room with a console

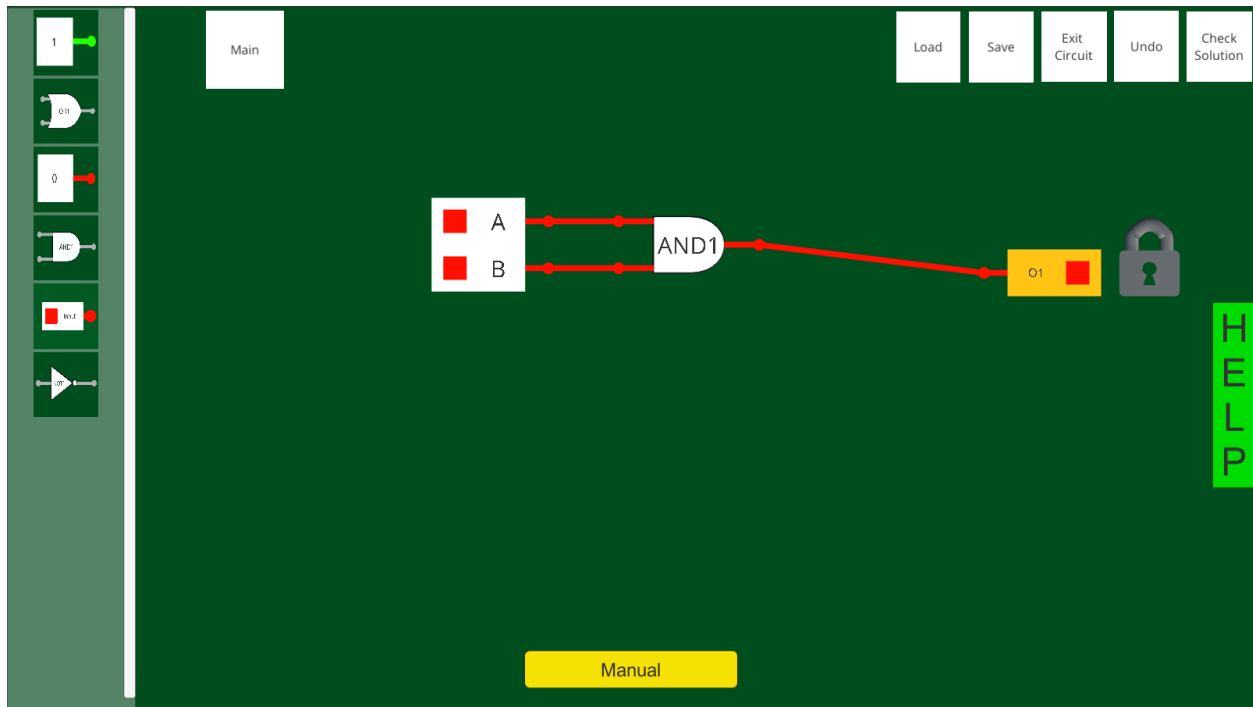


Figure 13: Solution to The Right Circuit Puzzle

and slightly hidden component pickup, and one raised room reachable by a platform as shown in Figure 14. This section acts as the first of three big challenges presented to the player now that they have learned the basics of gameplay from the preceding content.

The player must collect the hidden output component pickup in the room before they can access the console for the circuit puzzle in this section. The circuit puzzle itself is based off an assignment from the course the game is based off of. This puzzle requires the player to make a circuit that matches the following expressions for two outputs:

$$O1 = AB + (C(A + D))'$$

$$O2 = C + D + AB$$

In these expressions, O1 and O2 are output components and A, B, C, and D represent input components. Pieces of how to interpret these expressions are included with every component's entry in the manual the player always has access to. For example, selecting

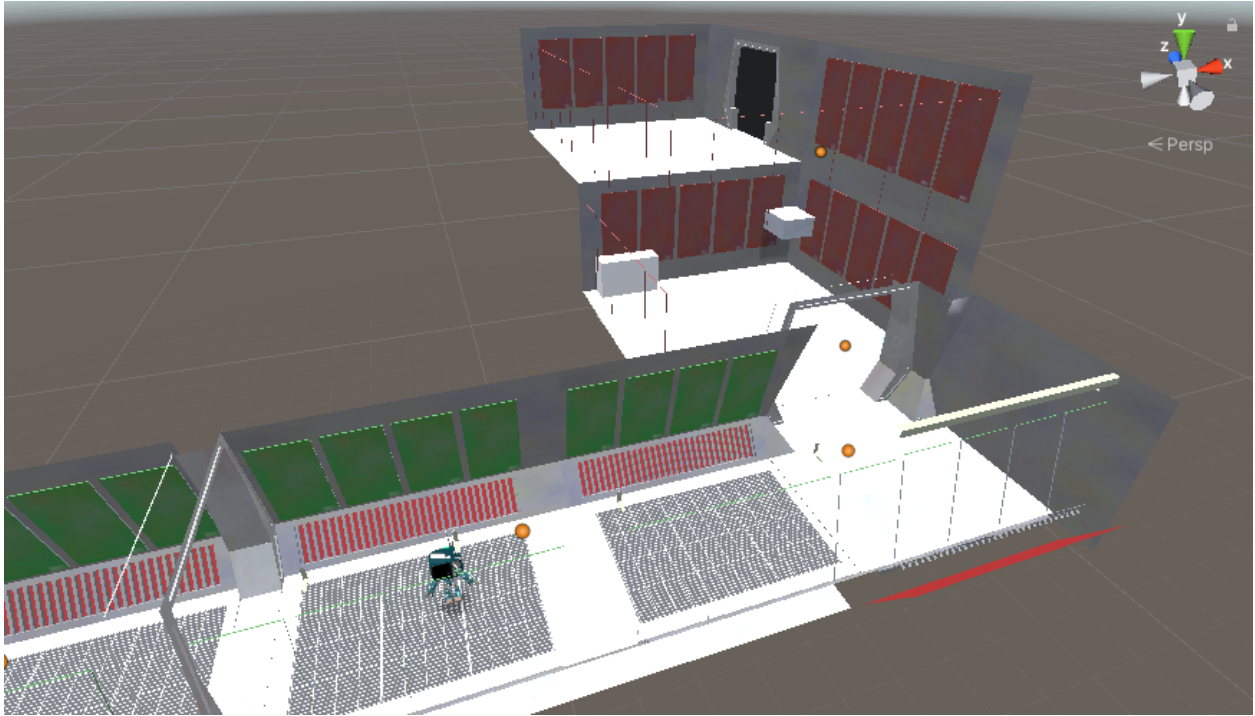


Figure 14: Third Game World Section Overview

the NOT component from the manual mentions that it is denoted with an apostrophe in an expression, such as A' .

In Figure 15 the player has created a solution for the puzzle. The player has also used the game's ability to rename circuit components, making it more clear how they relate to the expression. They have also used the wire's ability to create pivot points when dragged to make the overlap of wires slightly less jumbled.

The player is rewarded upon completing this puzzle with an upgrade for Robo. This upgrade is a jet pack that allows Robo to double jump. The upgrade allows the player to get on the platform and then onto the raised room where they can approach the door to load the next game world section.

This next section begins with a hallway that puts the primary challenge of the section, a digit recognizer, in full view to entice the player into approaching it as shown in Figure 16. When the player approaches the room containing the digit recognizer and seven segment display, they are introduced to the room through a conversation between Robo and HQ. HQ

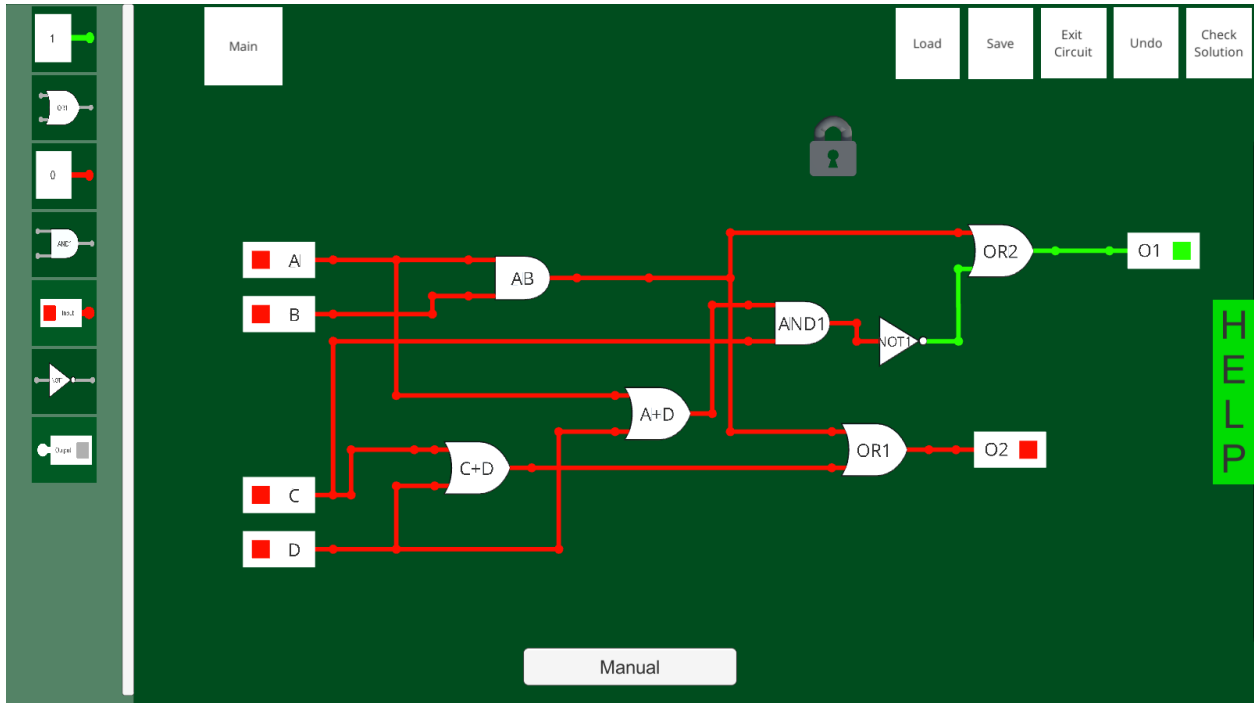


Figure 15: Expression-Based Circuit Puzzle Solution

informs Robo that the ship’s power source is behind the door leading out of the room. HQ also informs Robo that the door requires a specific code to be created and displayed by the circuitry in the room and that more information will be put in Robo’s manual.

This game world section contains three circuit component pickups for the player: the 4OR component pickup, the 4AND component pickup and the 3AND component pickup. These are not strictly necessary to complete the puzzles in this section or the remainder of the game, but they do simplify some puzzle solutions.

The challenge in this section is making the seven segment display only show, and show correctly, the characters that make up the six digit code on the wall (AC4d2b).

The section has eight total consoles that load into eight different circuit puzzles. Each of these consoles corresponds to an input for the seven segment display. These inputs are a , b , c , d , e , f , g , and en . The alphabetical inputs correspond to specific segments in the display and en corresponds to whether the seven segment display should be enabled or not (on or off).



Figure 16: Walking Into the Digit Recognizer Room

The visual display in the game world responds to the changes the player makes to each input of the underlying seven segment display circuit. The player can exit the circuit puzzles without checking their solution to see the effect it has on the visual display. If the player checks their solution for an input and it works, they are not allowed to re-enter that circuit puzzle. This was done to let the player know they have a correct solution to that part of the puzzle.

The digit recognizer cycles the seven segment display through all the possible outputs the seven segment display can show. Which of those it is trying to display at any time is shown by a green highlight on the upper display which shows all the possible outputs as shown at the top of Figure 17. If the part of the code that needs to be displayed is shown correctly when the display cycles to it, then that part of the code also changes to green to show the player their progress as shown in Figure 17.

If the display fails to output a part of the code correctly or displays one of the characters not in the code, then the bar under the code will display a red “Error” to let the player



Figure 17: The b Correctly Displayed

now something is wrong as shown in Figure 18. In Figure 18, the player has created correct solutions for the *a* segment of the display and the *en* part of the display so that the *a* segment is turned on as it is a part of the A which the code requires, but they have yet to get the other segments for A to be fully displayed.

After the player has completed all eight circuit puzzles successfully, the seven segment display will only show the characters in the code during the digit recognizer's cycle of displaying all of the outputs. When the cycle goes through without errors the display bar under the code will show a green "Good" and HQ will contact Robo to congratulate them as well as inform Robo that the door is now open. This is shown in Figure 19. The player can then move Robo to the door to load the final game world section of the game.

The final game world section begins with a hallway. At the end of the hallway the goal of Robo's adventure is clearly visible, the Arithmetic Logic Unit (ALU) Power Generator. Before Robo can enter the room and secure the power source they have been searching for from the start of the game, they are confronted by CircRat. CircRat taunts Robo that they



Figure 18: Error in the Display



Figure 19: Completed Code



Figure 20: CircRat Confronts Robo in The Final Game World Section

are too late and CircRat has already eaten all the circuitry in the room. This is shown in Figure 20. CircRat tells Robo that if Robo wants the power source then Robo will have to rebuild it on their own. CircRat then laughs and runs away.

HQ then calls Robo to inform Robo with details about how the ALU creates power by completing computations and the four computation types it can do: addition, subtraction, compare, and right arithmetic shift. HQ tells Robo that Robo will need to fix the computations in that order and that the first, addition, is located on the left side of the room on the second floor.

The next goal for the player becomes fixing the addition computation part of the ALU. To reach the addition relevant consoles, the player must use the light 3D platforming skills they have built up over the game to reach the second floor by completing two double jumps. All the other computation parts present the same platforming challenge to the player before the player can reach them.

The addition section of the room contains two consoles and one circuit component pick



Figure 21: Addition Location

up, as shown in Figure 21. The pickup contains the XOR component that must be picked up before any of the consoles can be accessed and their puzzles attempted. If any puzzle in this game world section is accessed before its prior requirements are met, the player is given Robo's dialogue with themselves about what they might need to do before they can access it. Once the player has the XOR component, they can access the leftmost console in the addition section.

This first console's circuit puzzle is introduced by HQ who explains that it is a one bit adder which is a subpart of the addition circuit Robo will need to make next. HQ also prompts the player to look at the help tab for more information on a one bit adder before proceeding to build one. When the player completes the one bit adder puzzle and returns to the game world, Robo informs the player that they are ready to get addition working and the red wire connection between the two consoles in the addition section has turned green, as shown in Figure 22.

The player can then access the second console in the addition section which contains



Figure 22: 1 Bit Adder Complete

the actual circuit puzzle that will do addition logic for the ALU. This puzzle introduces the player to the idea of tabs, which are additional circuits previously made by the player for the player to use in the current puzzle. Tabs introduce the concept of abstraction, taking a circuit that is complex and making it into its own component with a simpler view. The game automatically imports the tab for the one bit adder and puts it under the adder tab, but the player can hit the import tabs button at any time to reload their tabs if they make a change in one. Figure 23 shows HQ introducing tabs to the player.

A circuit from a tab can be used in any tab to the left of it. The player only has two tabs in the addition circuit puzzle that introduces tabs. These tabs are the *main* and *adder* tabs. To bring a circuit from a first tab into a second tab as a component in that second tab's circuit, the player must find the labeled circuit component in the side bar on the left side of the screen. In Figure 24 the Adder circuit can be seen in the side bar at the bottom left of the image and what it looks like when in the larger circuit can be seen in the center of the same figure.

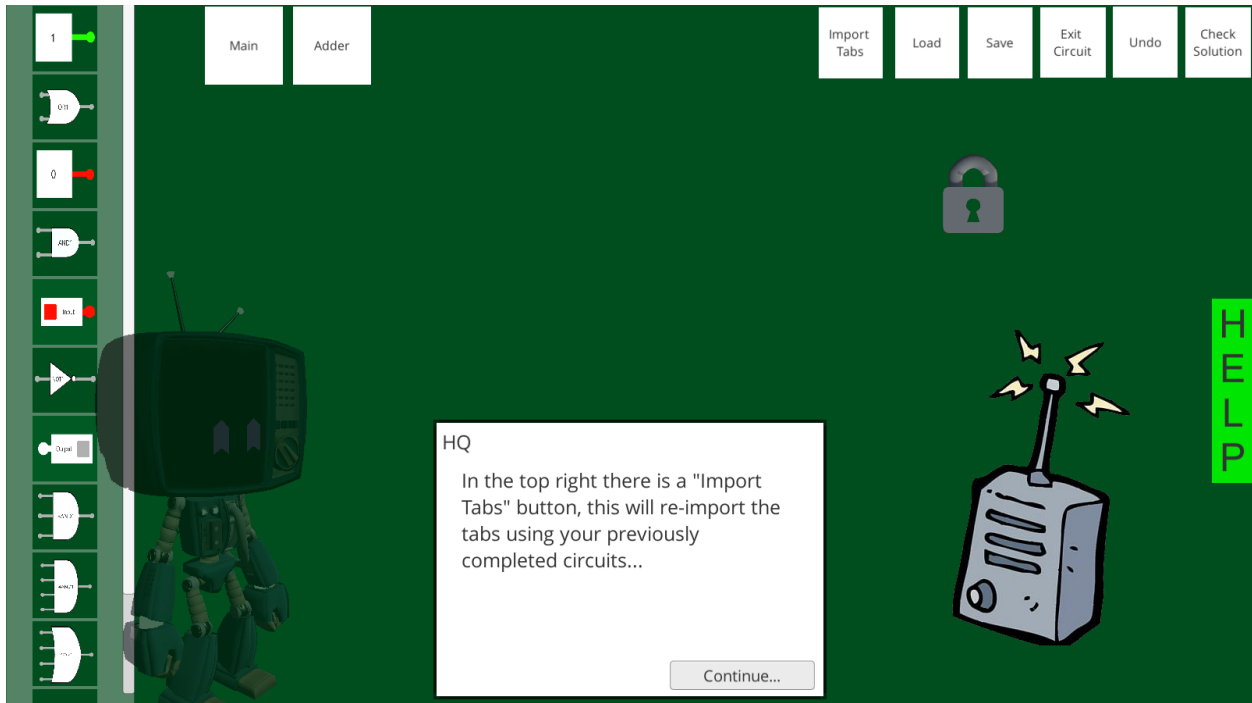


Figure 23: Tabs Introduction

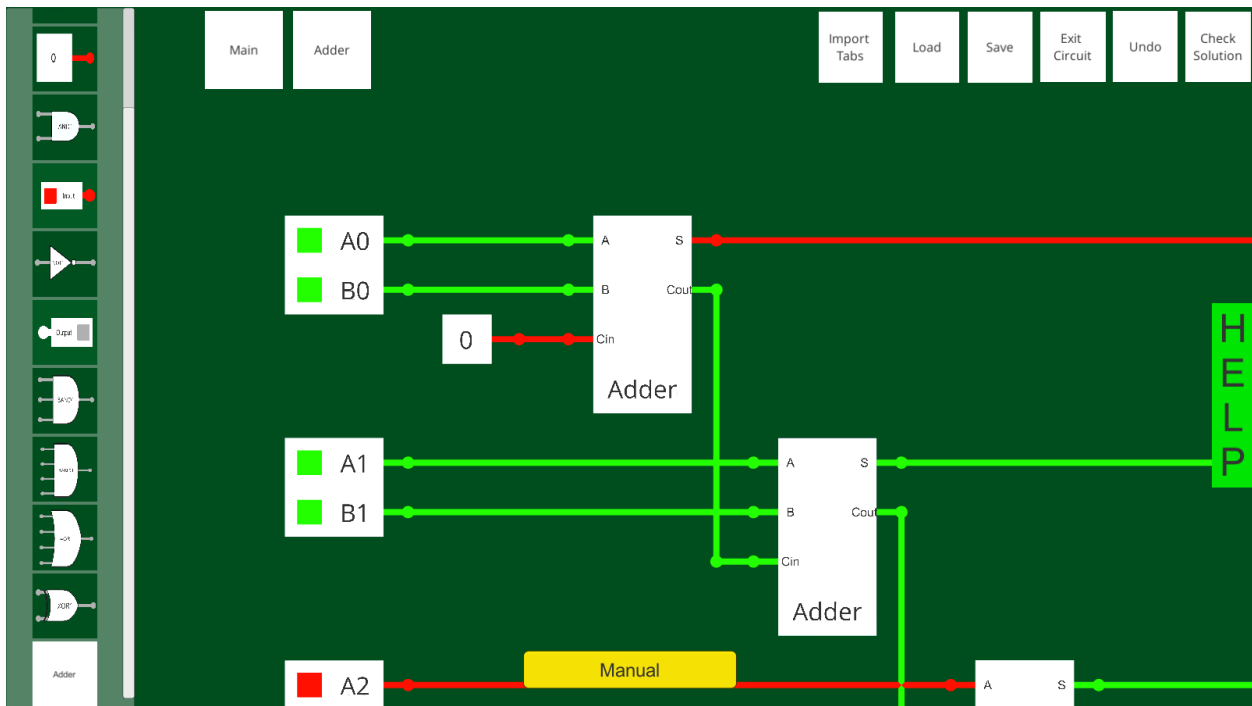


Figure 24: Adder Used in Addition

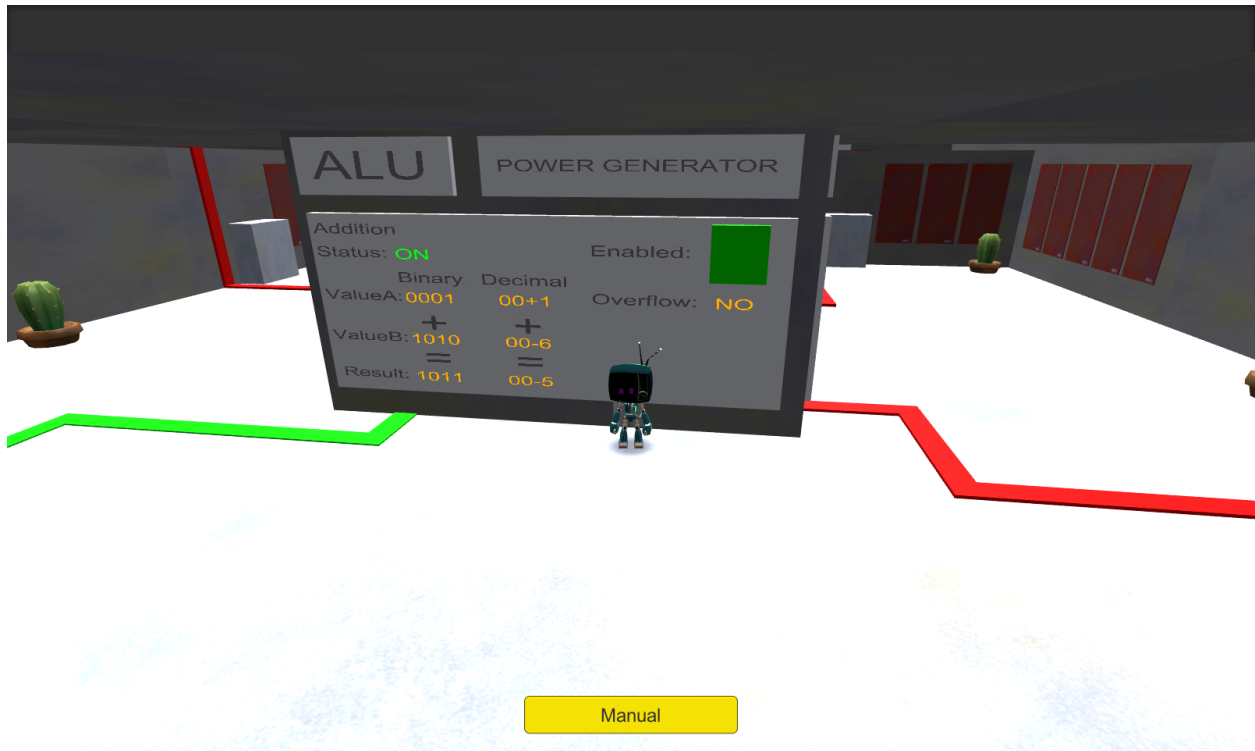


Figure 25: ALU Addition Screen Shows Computations

The addition circuit puzzle requires the player to make a four bit adder. When the player solves this puzzle, the addition computations from the ALU power source begin working again in the game world and the red wires between the Addition consoles and the power source in the center of this game world section become green.

If the player returns to the side of the power source which the green wires connect to, they will see that the addition screen on the ALU power generator has begun displaying random computations along with the results of those calculations as shown in Figure 25.

After fixing addition, the only console the player can access next is the one in the subtraction section. The console in this section loads the player into the circuit puzzle for subtraction.

The subtraction circuit puzzle is similar to the addition circuit puzzle. The player must make a four bit subtractor. The subtraction circuit puzzle allows the player to use the one bit adder they made in the same way as the addition puzzle. The subtraction panel of the ALU power generator in the center of the room also begins cycling through computations

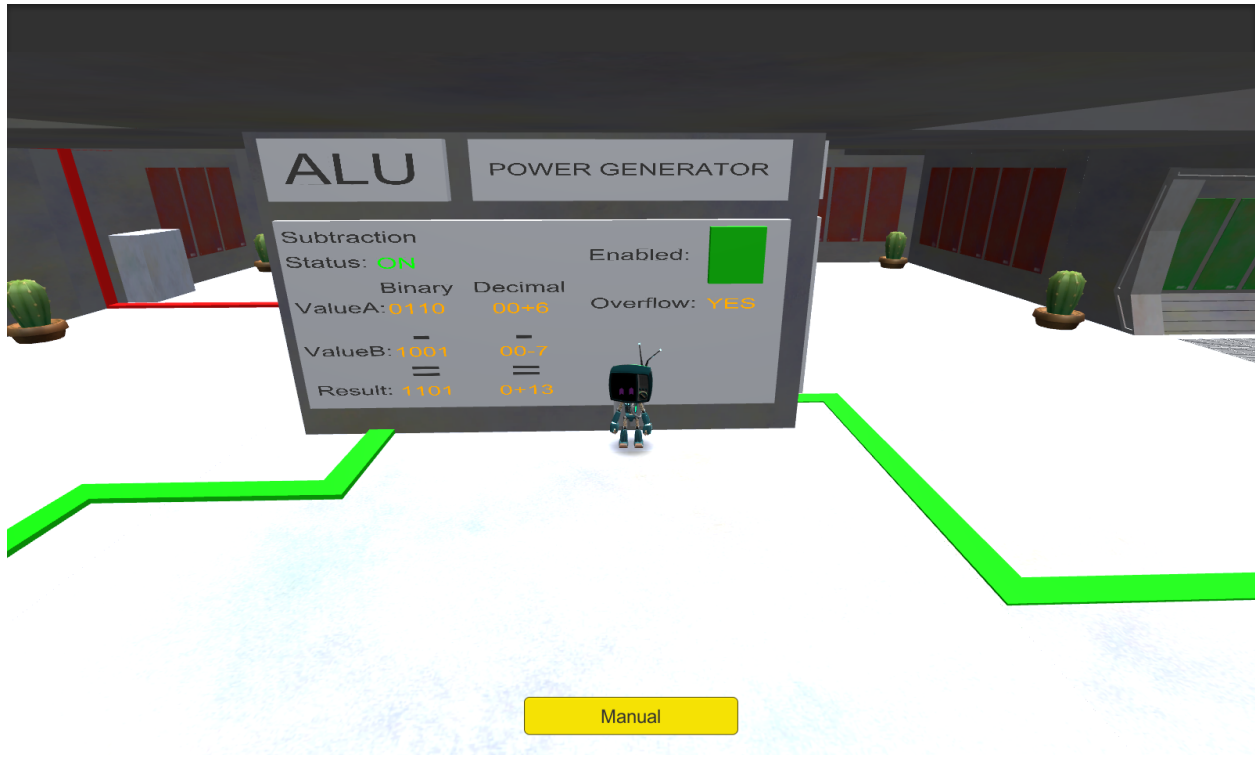


Figure 26: ALU Subtraction Screen Shows Computations

after the player solves the puzzle and returns to the game world as shown in Figure 26.

After completing subtraction, the player must fix the compare part of the ALU power generator. Again, this is in another corner of the level, is accessed by two double jumps, connects back to the ALU power generator by a red wire, and the display for it on the generator starts showing computations after the circuit puzzle is solved.

The compare circuit puzzle requires the player to make a four bit signed comparator that does the operation A less than B when it is enabled. The difference for this puzzle compared to the addition and subtraction puzzles is that the player is allowed to use the one bit adder and four bit subtractor they made to create their solution. This means they have three tabs in this puzzle: main, for the compare circuit; subtraction, for the four bit subtractor they made; and adder, for the one bit adder they made. A solution is shown in Figure 27.

After completing the compare circuit, the player can progress to the final circuit puzzle of the game, the right arithmetic shift puzzle. This circuit puzzle is much less complex than

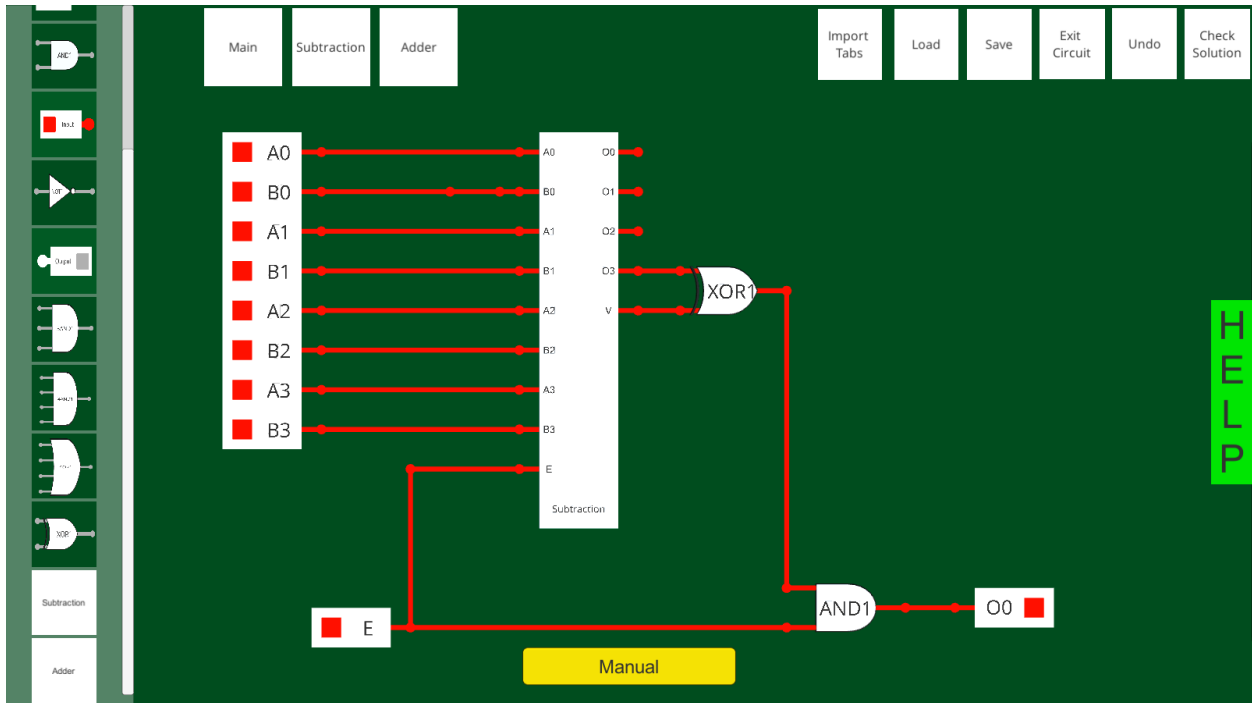


Figure 27: A Solution to the Compare Circuit

the three preceding it and does not have any additional tabs the player needs to keep track of. This final puzzle as well the other three in the final game world section have all of the hints and tips needed to solve each puzzle present in the help tab, whereas earlier on in the game some hints were communicated by dialogue between characters.

After the player completes the right arithmetic shift circuit puzzle and returns to the game world, Robo states that they have fixed the generator and need now only copy down the design from the generator's main console, which is located right under the right arithmetic shift display as shown in Figure 28.

When Robo accesses the generator's main console, Robo exclaims that they have done it and copied the power generator's design. The screen then does a final circular fade in and out to reveal a Thanks for playing screen as shown in Figure 29.



Figure 28: Right Arithmetic Shift Screen Shows Computations



Figure 29: End Game Screen

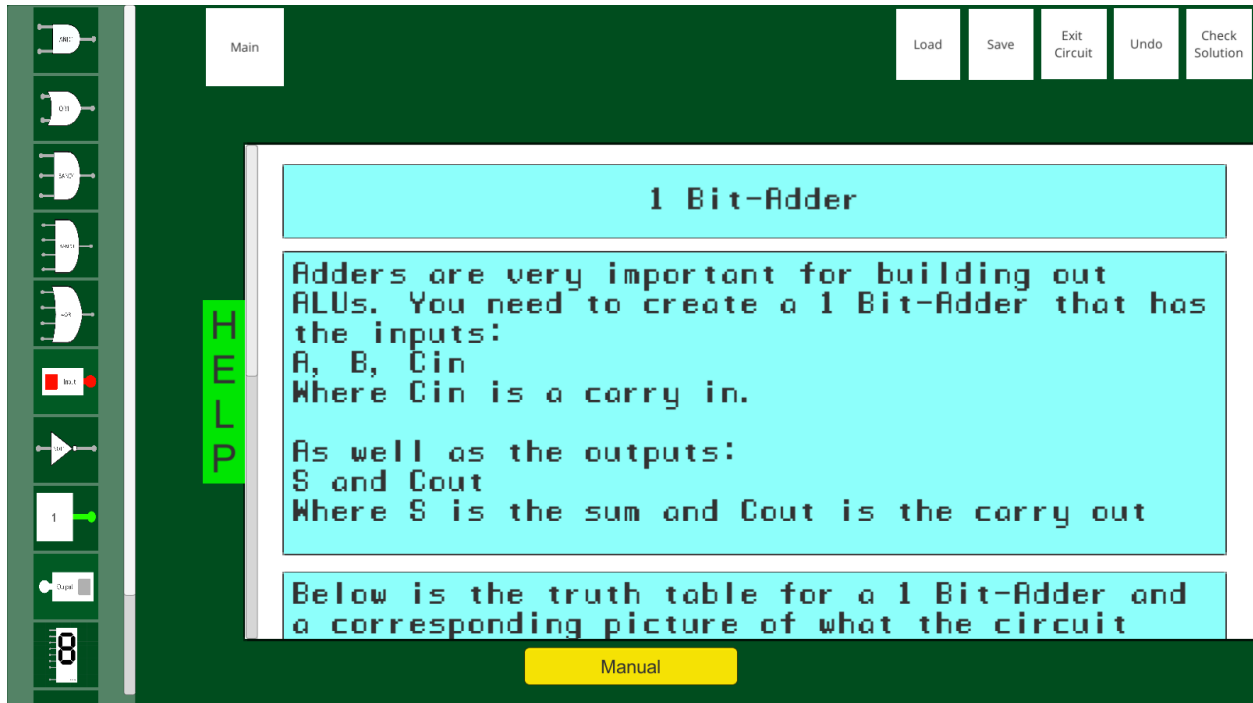


Figure 30: Help Tab in The Circuit Puzzles

3.1.2.5 In-Game Assistance

Features to make the games function with as little effort from an instructor were added to the games. The reasoning behind adding this in-game assistance was to make the games playable outside of the course context. Basically, the goal was to make the games able to stand alone if needed. The games offer two primary features for in-game assistance to help in this regard. These two features are the help tab and the manual.

The help tab is only available inside of the circuit puzzles and contains information only relevant to the circuit puzzle the player is currently in. The player accesses the help tab by clicking the help button on the right side of the circuit puzzle screen. This causes the help tab to animate outward until it is in full view as shown in Figure 30. The player can close the help tab by clicking the help button again to trigger the tab's closing animation.

The manual is available to the player in both the game world and the circuit puzzles. The player opens the manual by clicking the manual button at the bottom of the screen. The player can close the manual by clicking the same button. The manual contains pages for

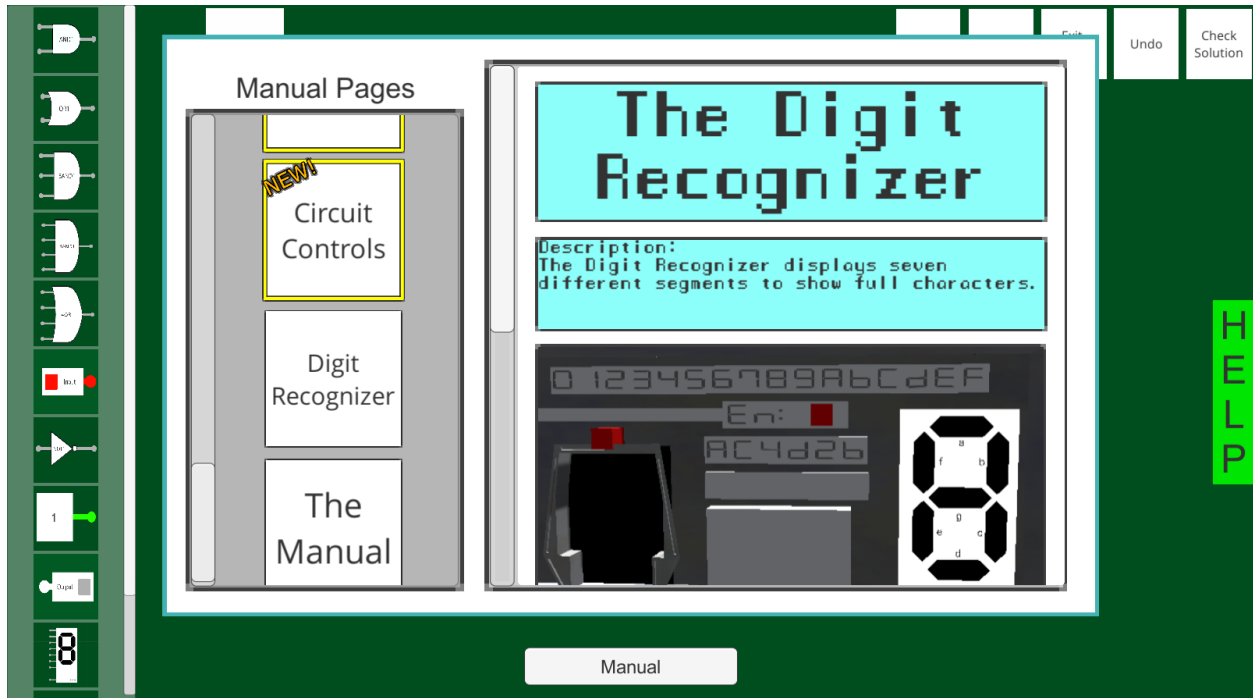


Figure 31: Manual Opened in Circuit

every component the player collects, controls explanations, as well as some specific entries related to challenges in the game world such as the digit recognizer page shown in Figure 31.

The manual does not have all of its pages at the beginning of the game. Instead, the manual is populated with pages as the player comes across the relevant parts of the game. A new manual page entry is typically accompanied by a dialogue that tells the player that they should check their manual for it.

The manual button becomes yellow every time a new page becomes available to draw the player’s eyes towards it and encourage interaction. Additionally, any page the player has yet to open has an added overlay in yellow and orange with “NEW!” across it so they can easily identify recent pages of interest for what they are doing in the game.

The information provided by this in-game assistance was initially planned to be more interactive in nature, tied into more of the games’ core mechanics and narrative. However, limitations on development time and other resources made this the best solution to making the games reach all of their learning content goals. Additionally, these systems and infor-

- **Combinatorial Circuits**
 - Circuits, Expressions and Truth Tables
 - Canonical Sum of Products
 - Minterms and Minimization
 - Karnaugh Maps
- **Implementing Combinatorial Circuits**
 - Karnaugh Maps to Circuits
 - Decoders
 - Multiplexers and Demultiplexers and Encoders
- **Arithmetic Logic Units**
 - Adders
 - Subtraction
 - Comparison and Shifting
- **Sequential Circuits**
 - Clocks and Clocked Circuits
 - Latches and Flip-flops
 - Registers and Register Files
- **Processors**
 - The Fetch/Execute Cycle
 - Registers and the ALU
 - Instruction Decoders
 - Immediate Instructions
 - ALU Instructions
 - Load/Store Instructions
 - Branching and Conditional Branching
- **Floating Point**
 - IEEE-754 Format
 - Float Point Addition
 - Multiplication

Figure 32: Course Content

mation would have been developed anyway to keep to the goal of having the game be able to stand on its own.

3.1.3 Learning Objectives

The learning objectives for these games are based on content covered in Michigan State University’s Computer Organization and Architecture course, CSE 320. Equivalent courses are required by most computer science degree programs. A subset of the targeted course content was implemented in the games due to resource limitations during development. Figure 32 shows all the content covered in the course.

3.1.3.1 Learning Content in Game

The learning content that made it into the game includes combinatorial circuits, implementing combinatorial circuits, and arithmetic logic units.

The content under combinatorial circuits is targeted by the game world and circuit puzzles up until the end of the section where the player obtains the jet pack. This section of game world and circuit puzzles is covered in the first game in the three game sequence of games for the multiple game group used in the study.

The game world section that contains the digit recognizer and seven segment display contains the circuit puzzles that target content under the implementation of combinatorial

circuits. This section of content is covered in the second game in the three game sequence of games for the multiple game group used in the study.

The final game world section that contains the ALU power generator targets content under the arithmetic logic units. This section of content is covered in the third game in the three game sequence for the multiple game group used in the study.

One criticism of the games' current implementation of this learning content is that it skews more heavily to checking the understanding of some of the underlying concepts of that learning content rather than building up the player's knowledge in addition to testing it. This is most clear in the later two sections of the game where the game acts more like a gamified version of sets of assignments done in the course rather than a fully stand alone game. For example, the second section of the game world's circuit puzzles can be simplified using Karnaugh Maps and an understanding of minterms and minimization.

The game's current released source version [17] has a demo for allowing the player to create and use Karnaugh Maps while also providing a gameplay means of making them and learning them, as well as minterms and minimization. However, this game content was incomplete at the time the game's were deployed. Figures 33 and 34 show portions of the Karnaugh Map focused game content.

3.1.4 Student Reception

In a research study the game was applied to [52, 53], a post survey included open ended questions that garnered some of the student response to playing the game.

When asked, "what did you enjoy most about the game or games you played?" students mentioned a wide variety of things. Several students made comments on how they found the game engaging:

"The game was a lot more enjoyable than the normal CSE 320 circuit assignment because it was more engaging."

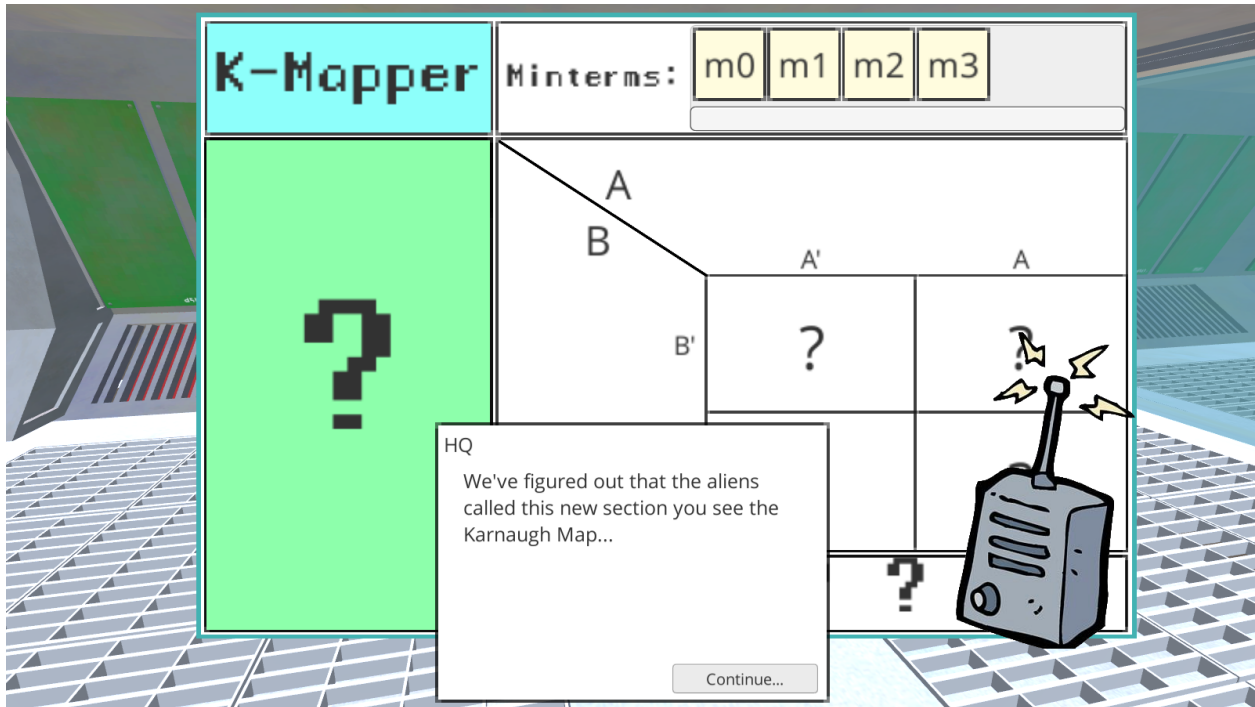


Figure 33: An Unimplemented Game World Section Introducing Karnaugh Maps

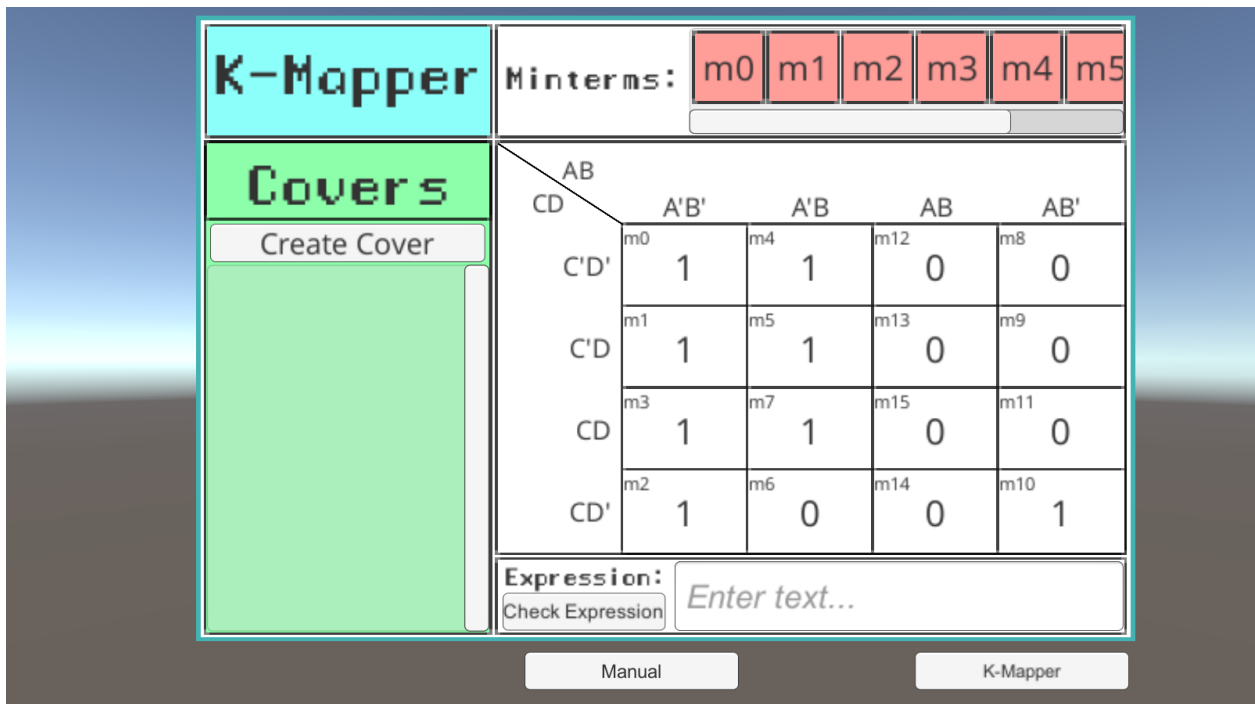


Figure 34: An Empty Unity Scene Demo for Karnaugh Maps Playtesting

“I enjoyed the characters and design of the game it made things for [sic] interesting.”

Other students liked the learning aspect of the game:

“I enjoyed the aspect of being able to practice my knowledge in an interactive game.”

“The second to last task, dealing with the en and a-f gates, helped me become better at K-maps.”

Students also mentioned the circuit puzzles in response to this question:

“I liked how I felt that I was achieving something whenever I completed a puzzle.”

“I also liked the circuit platform design, it was like cirsim and easy to use.”

CIRSIM refers to the circuit simulation software that runs on the web typically used in the class [54]. It is basically a non-gamified version of the system used by the circuit puzzles in the game. Other students mentioned story or the characters as their most enjoyed part:

“I liked the characters.”

“I enjoyed the story and how it fit together with what we were learning in the course.”

Several responses even mention various in-game elements as what was enjoyed most:

“The special effects.”

“I enjoyed the character movement and the way the challenges were set up.”

“I didnt [sic] actually expect the game to have such various features, I was surprised about the jetback [sic] feature.”

“That they were interactive and the graphics were really cool.”

The responses to being asked what they enjoyed most about the games paint a positive image of student reception of the game in both the singular game and multiple game versions. Of course, this image shifts when looking at only the responses to the next question, “what did you most dislike about the game or games you played?”

There were two big sets of responses for the dislike question. One of these sets of responses points out technical issues or bugs. The other points out specific design issues and shortcomings of the game.

The technical issues and bugs responses contained comments such as:

“The areas where you actually create your circuit is extremely small if you’re using many components. Also, circuit components can get stuck behind the area where you drag them out. Last of all, circuits don’t save either correctly or at all when you leave the circuit creation page.”

This particular comment highlights an issue with communicating the controls as the area to work on the circuit can be made larger using the camera zoom control. However, considering this in combination with several other comments points to a general issue students had with the camera in the circuit puzzles:

“I felt that there wasn’t enough space and zooming out would make the labels very small.”

“That quality dropped a lot when zooming out.”

Fortunately, the majority of students did not seem to run into the issues of components getting stuck behind things. Most did not have saving issues either, though there is one more comment that mentions experiencing a few crashes that stopped them from saving.

Beyond technical issues with the way the camera works in the circuit puzzle, two other common technical issues arose in the responses. One was that several students really disliked the way the camera functions in the game world:

“The camera was frustrating it often was hard to look around at things that I thought I’d need to see.”

“It was difficult to move around and find the next step.”

The last common issue relates specifically to the final section of the game world’s puzzles. This issue is the time it takes for the game to check the student solutions:

“The game was really slow framewise and it took super long to test the circuits.”

“I disliked the time it took to run circuits, and the loss of several feature we have available to us on cirsim.”

These technical issues and bugs are resolvable with more development time. The long testing time experienced in the later puzzles, for example, can be fixed by testing a representative subset of all the possible combinations instead of all of them as the game does currently. The issues with the camera in the circuit puzzles can be alleviated somewhat by adjusting the zoom and pan speeds, or by adding an in-game setting to allow the player to adjust that speed themselves. The issues raised with the in-game camera could potentially be resolved by changing the camera type to something controlled directly by the player.

The set of responses related to design issues can similarly be alleviated in future iterations of the game. For example, one comment takes issue with needing to change the name of the inputs in certain puzzles:

“I dislike that we need to change the name of the inputs.”

In future iterations of the game, the inputs can be made to automatically name themselves to the same name as the inputs required by the code which checks correctness. Other design issues are raised in these student quotes:

“When building circuits, if I wanted to reread the instructions, they were difficult to locate.”

“The tutorials could be improved in some areas. Specifically, it took me forever to find out how to delete things efficiently when making the circuits.”

The things students disliked the most are issues that could be fixed with more development time, iterations, and play testing. There were even some positive pieces in responses to the most disliked part question:

“As a whole the control was very smooth, but the graphics were very hard to play”

“I found nothing in particular that I disliked about the game, I think it would’ve been cool if there were Easter eggs hidden in the game.”

Overall, the student reception of the game appears to generally be positive while providing some clear areas for improvements to be made.

4 Design Research from CircuBot

Portions of this chapter appeared in the following publications:

Declan McClintock. 2022. All together now, using multiple frameworks to inform serious game design and development. In *International Conference on Meaningful Play*.

https://meaningfulplay.msu.edu/proceedings2022/mp2022_paper_5725.pdf.

4.1 Introduction

The development of CircuBot included design research using the approach of employing three existing frameworks to primarily inform the process. The use of three frameworks ultimately led to more areas of design and development being addressed than if only one framework had been employed and the recognition of various elements commonly covered or not covered by the frameworks used. A paper covering the various elements observed during the design and development of CircuBot was presented at Meaningful Play 2022 [51], and the same information is presented here along with the inclusion of a methodology referenced during CircuBot’s development.

4.2 Frameworks for Serious Game Design and Development

There are many game design frameworks and methodologies that can be utilized in the design and creation of serious games. The ones used in creating the computer architecture games include the Mechanics Dynamics Aesthetics framework (MDA) [1], the Design Play Experience framework (DPE) [2], the Serious Game Design Assessment Framework (SGDA) [3], and the Gameplay Loop Methodology (GLM) [4]. Each framework emphasises different elements of game design and development. There is considerable overlap between them, though no single framework covers all elements of the design process.

The elements that influenced the design and development of the serious games used in the experiment in Chapter 5 break down into the following categories:

- In-Game Elements
- Design Elements
- Development Elements
- Serious Elements

4.2.1 In-Game Elements

In-game elements are elements that exist within the game and are primarily experienced directly by the player. While all in-game elements have to be thought out and designed by the designer(s) of the serious game, these elements are separated from the design elements by the fact that they are significantly viewed or interacted with from the player's perspective. Design elements are instead primarily from the view of the designer. In-game elements include mechanics, narrative, audio, art, and game cohesion.

4.2.1.1 The Mechanics Element

One in-game element that each framework discusses is mechanics. MDA, DPE, SGDA, and GLM all agree that a game's mechanics are either the most important or one of the most important elements of a serious game [1, 2, 3, 4]. Mechanics are: "The rules that define the operation of the game world, what the player can do, the challenges the player will face, and the player's goals." [2] Mechanics are what the player does when interacting with the game, moving the player character around, talking to non-player characters, moving puzzle pieces, and so on. GLM argues that existing literature neglects the importance of game mechanics and pushes hard for mechanics and learning goals to be tightly coupled, what the player is doing (mechanics) aligns with what the player is learning [4]. The other frameworks push along that same line of thinking. Ensuring that mechanics align with learning goals was the primary consideration in the development of the computer architecture games.

Figure 35 provides an overview of which frameworks cover the mechanics element. The letter **D** represents when an element has been directly accounted for by the framework, the letter **I** represents when an element has been indirectly accounted for by a framework, and a blank space means that the element is not mentioned directly or indirectly as a part of the framework. The other element tables use the same symbols.

| | Element |
|-------------------------|-----------|
| Framework / Methodology | Mechanics |
| MDA | D |
| DPE | D |
| SGDA | D |
| GLM | D |

Figure 35: Coverage of the Mechanics Element by Frameworks

4.2.1.2 The Narrative Element

Narrative refers to both the embedded narrative, which is the narrative created for the game by the designer [46], and the emergent narrative, the story the player ultimately experiences through their unique playthrough of the game [46]. In DPE, these are discussed as the designer’s story and player’s story respectively [2]. This designer’s story, or embedded narrative, is being included as a part of the narrative element within the in-game elements because it is something that the player interacts with and experiences.

SGDA describes the narrative element’s relation to the purpose of a serious game using the example of *Sweatshop*, a game that puts the player in the position of running a sweatshop [3]. The embedded narrative provided by the designer is that the player is a manager of the sweatshop that must balance the profit motive with taking care of their employees. The emergent narrative of *Sweatshop* arises through the player’s individual choices as a manager, attempting to balance profit with the happiness of their employees [3]. The game makes strong use of the narrative element to fulfill *Sweatshop*’s purpose of highlighting working conditions in sweatshops while also showing the pressures and conditions that create the

system [3].

The narrative element across the computer architecture games is split into two cases. In one of these cases the students progress through three sets of learning content underneath one cohesive narrative. The students in the second case play a set of three games, one for each set of learning content, and each with their own narrative that does not tie into the narratives of the other games. In each game, the narrative design provides a reasoning for the player to complete challenges and progress through game content.

Figure 36 provides an overview of which frameworks cover the narrative element.

| | Element |
|-------------------------|-----------|
| Framework / Methodology | Narrative |
| MDA | |
| DPE | D |
| SGDA | D |
| GLM | |

Figure 36: Coverage of the Narrative Element by Frameworks

4.2.1.3 The Audio Element

Audio refers to the sounds within the game that the player hears. This includes sound effects, background music, and any spoken dialogue. Only SGDA touches on the audio element as part of what it refers to as *aesthetics*, defined within SGDA as: “The audiovisual language conceptualized, chosen and developed by the designers for the visualization, and the display of the elements involved in the game.” [3] Despite mentioning audio in the definition, SGDA’s more detailed discussion of aesthetics focuses on the graphical part of their definition while leaving out audio. The lack of a detailed direct discussion of the audio element within these frameworks is concerning given the role sounds and music can have in communicating information to the player, increasing immersion, and setting the emotional tone of the game and any of its important moments. See Figure 37 for a breakdown of which frameworks discuss audio.

Unfortunately, the computer architecture games lack audio due to resource constraints that will be discussed later. However, both cases used in the research lack audio. This removes possible differences induced by audio in the results.

| | Element |
|-------------------------|---------|
| Framework / Methodology | Audio |
| MDA | |
| DPE | |
| SGDA | I |
| GLM | |

Figure 37: Coverage of the Audio Element by Frameworks

4.2.1.4 The Art Element

Art refers to the graphics and other visual elements of the game; everything the player sees. SGDA discusses the art element directly under their definition of aesthetics. SGDA argues that art, as part of their definition of Aesthetics: “Plays a fundamental role in the introduction of the game’s purpose and its impact on the player.” [3] See Figure 38 for a breakdown of which frameworks cover art.

In the computer architecture games, art quality was kept similar between the two cases. Changes were made to art assets to create the needed narrative differences between the games. Art in both cases was tied to the relevant narrative and educational purpose of the game in line with the suggestions from SGDA.

| | Element |
|-------------------------|---------|
| Framework / Methodology | Art |
| MDA | |
| DPE | |
| SGDA | D |
| GLM | |

Figure 38: Coverage of the Art Element by Frameworks

4.2.1.5 The Game Cohesion Element

Game cohesion refers to the intertwining of the other in-game elements with each other. Strong game cohesion occurs when the in-game elements work to support and enhance each other (inter-element cohesion) and themselves (intra-element cohesion).

An example of strong inter-element cohesion would be a low-gravity jump mechanic for the player character in combination with a narrative that has communicated that the player is on the moon and why, audio that provides the matching sound of an astronaut jumping and unsettling dust on the moon, art that presents a convincing visual representation of the moon around the player character alongside a suitable astronaut model for the player.

Poor inter-element cohesion would be in-game elements that do not match up with each other; e.g. art showing the player on the moon, but a jumping mechanic that is the same as when the character was on Earth.

An example of strong intra-element cohesion in the art element would be a consistent visual style and color palette used throughout the game, such as maintaining a cartoonish style.

An example of poor intra-element cohesion in the narrative element would be a story with contradictions, plot holes, and inconsistencies that are never resolved.

An example of poor intra-element cohesion in the audio element would be poor audio balancing where something that should be barely audible is instead loud, such as a sneaking character's footsteps being louder than a gun shot.

Game cohesion is important for serious games and games in general. Strong game cohesion helps maintain immersion by providing the player with a world that maintains consistent rules that the player agrees to and exists in during play. Providing poor game cohesion risks a game that repeatedly breaks the player out of their immersion.

Game cohesion was maintained as much as possible in the computer architecture games.

SGDA argues in favor of having more cohesiveness between the parts of a serious game and the game's purpose [3]. SGDA argues through its example games that more cohesion

strengthens a serious game by aligning the game’s parts with its purpose, which should result in a higher likelihood of the game’s purpose being realized [3]. Figure 39 shows how SGDA is the framework that most accounts for game cohesion. Figure 40 shows a summary of framework coverage of all the in-game elements.

| | Element |
|-------------------------|---------------|
| Framework / Methodology | Game Cohesion |
| MDA | |
| DPE | I |
| SGDA | D |
| GLM | |

Figure 39: Coverage of the Game Cohesion Element by Frameworks

| Framework / Methodology | In Game Elements | | | | |
|-------------------------|------------------|-----------|-------|-----|---------------|
| | Mechanics | Narrative | Audio | Art | Game Cohesion |
| MDA | D | | | | |
| DPE | D | D | | | I |
| SGDA | D | D | I | D | D |
| GLM | D | | | | |

Figure 40: Coverage of In-Game Elements by Frameworks

4.2.2 Design Elements

Design elements are elements that are viewed and considered primarily by the designer(s) of the serious game. The design elements include polish, technology, user interface, target audience, balance, gameplay/play dynamics, resulting experience, designer perspective, player perspective, academic perspective, content expert perspective, and design cohesion.

4.2.2.1 The Polish Element

The idea of polish comes from the game development industry and is not mentioned in the frameworks, see Figure 41. Polish is the finishing touches across the in-game elements. The game development industry refers to polish as the last ten percent of work [55, 56]. According to Naughty Dog’s Benson Russell, polish is the removing of imperfections from the game

that could take the user out of their immersion in the experience [56]. Any of the in-game elements can be at differing states of polish.

A mechanical example of lack of polish would be a character controller that gets stuck on the geometry or shape of a level. This would cause frustration for the player that can no longer play the game because of a fault of the game and not of their own skill. This then leads to the player being removed from their immersion. Inversely, a mechanical example of polish would be a character controller that does not get stuck on the geometry of a level.

Minor inconsistencies in the story, missing sound effects or unsuitable sounds, or a texture that is slightly out of place are all examples of lack of polish. These may seem like insignificant issues when compared to their game breaking counterparts, such as a completely incoherent story, but the game development industry still puts focus on polishing their games to minimize the likelihood of a player’s immersion being broken.

It is just as important in serious game design and development to look for and resolve these smaller issues so that the player remains immersed in the experience, making the purpose of the game more likely to be realized.

Polish was maintained in the computer architecture games as much as possible to maintain immersion. Part of this included the choice to not attempt to include audio as having fitting audio at the same level of polish was considered near impossible with the available resources and it would be expected that poorly fitting audio would break immersion more so than no audio at all. Also, the level of polish in the games between the two cases applied in the application research [53] is relatively the same.

| | Element |
|-------------------------|---------|
| Framework / Methodology | Polish |
| MDA | |
| DPE | |
| SGDA | |
| GLM | |

Figure 41: Coverage of the Polish Element by Frameworks

4.2.2.2 The Technology Element

Technology refers to the hardware and software that the game is built to use. A designer needs to be cognizant of the limitations and affordances their chosen technology supports. For example, a Virtual Reality (VR) game gives the designer a unique control scheme to work with in which they can have the player physically move their body to provide input to the game. This might lend itself well to a serious game with the purpose of encouraging exercise as the purpose can line up well with possible mechanics. However, the designer must also consider the limitations VR presents. VR set ups are expensive. They require a powerful computer and additional devices to support them. If the serious game needs to reach a community without access to VR devices, then the designer has good reason to not choose to develop a VR game.

DPE summarizes the technology element well: “Overall, the capabilities and limitations of the technology and the resources required to implement the technology may greatly influence the design and should be considered throughout the design process” [2]. See Figure 42 for a breakdown of which frameworks cover technology.

The technology element influenced the computer architecture games. The technology chosen for the games to run on was web browsers. Because of this the design of the games was adjusted to meet the lower computational power limitation that web browser development enforces.

| | Element |
|-------------------------|------------|
| Framework / Methodology | Technology |
| MDA | |
| DPE | D |
| SGDA | |
| GLM | |

Figure 42: Coverage of the Technology Element by Frameworks

4.2.2.3 The User Interface Element

User interface (UI) ”encompasses everything the user sees, hears, and interacts with and how that interaction happens (i.e., the control system.)” [2] This includes the physical interactive components of a game, such as a controller or keyboard, and the interactive screens the player sees in game, such as the manual used in the game shown in Figure 31. Figure 43 shows the breakdown of which frameworks consider UI.

A good user interface is easy for a user to miss, meaning they don’t notice it because it is so easy to interact with. In the computer architecture games user interface was considered in making sure the various buttons and drag-able pieces of the UI in the game were easy to interact with. This included making sure buttons in the same group, such as the scroll space of the manual pages on the left of the manual in Figure 31, were sized the same and spaced apart evenly.

| | Element |
|-------------------------|----------------|
| Framework / Methodology | User Interface |
| MDA | |
| DPE | D |
| SGDA | |
| GLM | |

Figure 43: Coverage of the User Interface Element by Frameworks

4.2.2.4 The Target Audience Element

Target audience refers to who is meant to play the game. All of the frameworks discuss the target audience to some degree [1, 2, 3, 4] as reflected in the breakdown in Figure 44. According to DPE: “Play is greatly influenced by not only the designer, but also the player, including his or her cognitive, social, cultural, and experiential background that he or she brings to the given play experience... The target audience for the game must be strongly taken into account throughout the design process.” [2] Both DPE and SGDA point out that every player is unique and comes into playing a game with their own experiences and

expectations [2, 3].

SGDA’s example on how this can cause issues is play literacy, where different players might already be game players and thus pick up the controls quickly while others are not and will need the game to bring them up to speed [3]. If the target audience is mostly new to gaming, the designer(s) must find a way to address that. A well-structured tutorial in the game or an instructor providing guidance in the classroom might resolve that problem.

The target audience for the games used in the study [53] is computer science undergraduate students, most of whom are in their third year. This means that the target audience will already have a basic computer science background, include both experienced game players and completely inexperienced players, and be socially and culturally diverse. This is the target audience that was considered during the development of the games. The consideration of this audience led to the decision to not allow control of the game world camera for the player and to keep platforming challenges simplistic to accommodate any student players who are completely new to video games.

| | Element |
|-------------------------|-----------------|
| Framework / Methodology | Target Audience |
| MDA | I |
| DPE | D |
| SGDA | D |
| GLM | D |

Figure 44: Coverage of the Target Audience Element by Frameworks

4.2.2.5 The Balance Element

Balance refers to how well the game’s challenges fit with the player’s knowledge and skill. DPE relates balance to Mihaly Csikszentmihalyi’s theory of flow [2]. The original theory of flow focuses on pairing a skilled person with a challenging task that is neither too hard nor too easy [57]. This perfect matching of task to skill would then put the person in a flow state in which they lose awareness of themselves, they concentrate only on the task, they

experience a sense of control, and their perception of time skews so that they do not know where the time went after they stop working on or complete the task [57]. DPE applies this to balance in games by pointing out that a game presents challenges at differing levels of skill requirements which can be matched to a player’s skill level [2].

Early on in a game the challenges should have relatively low skill requirements to match the new player’s likely weak skills. Then the game should maintain that balance of player skill over the course of the game. At any point in the game the challenge presented should provide the player with something they are ready for, but not something too easy such that it causes boredom for the player. Similarly, the challenges should never be so difficult that the player finds them frustrating. Balance in this sense is the designer crafting a sequence of increasing challenges that always match with the player’s skills, thus keeping the player in the flow state [2]. See Figure 45 for the breakdown of which frameworks cover balance.

The consideration of the balance element in the games used in the study encouraged a smooth ordering of challenges.

| | Element |
|-------------------------|---------|
| Framework / Methodology | Balance |
| MDA | I |
| DPE | D |
| SGDA | |
| GLM | |

Figure 45: Coverage of the Balance Element by Frameworks

4.2.2.6 The Application Context Element

The application context is where the game is being played. The place of play can be just as important to the designer as the experience and cultural background of the player. If the game is to be played in a live classroom, then the designer must consider the role of the teacher in the player’s experience with the game. If the game is to be played as part of a fixture in a museum, then what does that mean for the game’s design? Perhaps the museum

will only allow each player to play the game for a set amount of time. The application context introduces design challenges that should be addressed by the designer. See Figure 46 for a breakdown of which frameworks discuss application context.

In GLM’s example game application, *Antura and The Letters*, the game was made to teach five to ten year old Syrian child refugees basic literacy skills such as Arabic letters, spelling, and vocabulary [4]. The Application Context for the game in a 2017 study was a refugee camp [4, 34]. This particular Application Context raises some design problems and limitations for the game designer. As noted by the 2017 study: “Within conflict or refugee settings, technological infrastructure presents an even greater challenge, as basic or consistent availability of electricity is often unpredictable, making the simple act of charging a mobile device or tablet problematic. . . . Where [Digital Game-Based Learning] and similar solutions have been applied in these settings, adapting to the context and its limitations have been key.” [34] For this 2017 study, the application context ended up pushing the game towards being developed for use on mobile devices, as solar-power chargers for refugees alleviated the power problem [34]. As the study using *Antura and The Letters* shows, the application context can introduce important design limitations.

The application context for the games used in the study is a fully online classroom. This application context meant that the games would be played at home by students on various hardware and software.

Two possible options were produced by this application context. One was to create operating system specific builds of the games for the most likely used operating systems of Windows, MacOS, and Linux. The other was to produce WebGL builds that could be run on modern web browsers such as Chrome or Firefox. The option to make WebGL builds was chosen so that the game could be hosted on the same website as a course which taught the same content as the games. This meant that players would not have to install the game on their systems to play it.

| | Element |
|-------------------------|---------------------|
| Framework / Methodology | Application Context |
| MDA | |
| DPE | |
| SGDA | |
| GLM | D |

Figure 46: Coverage of the Application Element by Frameworks

4.2.2.7 The Gameplay/Play Dynamics Element

Gameplay/play dynamics refers to the interaction of the game’s rules and in-game elements with the player’s input.

MDA and DPE are the two frameworks that discuss gameplay/play dynamics as reflected in Figure 47. In MDA and DPE, dynamics sit in between the designer’s created mechanics and the player’s resulting emotional responses from the game play [1, 2]. By MDA’s definition: “Dynamics describes the run-time behavior of the mechanics acting on player inputs and each others’ outputs over time.” [1] In MDA, mechanics are focused on as what determines the dynamics [1]. In DPE’s multi-layered framework, it accounts for an influence between its layers of learning, storytelling, gameplay, and user experience [2]. In DPE a design decision on what content to teach as part of the learning layer exerts an influence on the mechanics designed in the gameplay layer which then has an effect on the dynamics.

DPE provides a deeper depth of understanding of how the dynamics could be influenced beyond just the designed mechanics. However, DPE does not fully include some of the in-game elements such as art and audio. In the definition of the gameplay/play dynamics element used in this thesis, the interaction of ALL the mentioned in-game elements play a role, not just the mechanics.

Gameplay/play dynamics played into the design of the games used in this thesis by keeping the designer mindful of how the various in-game elements would interact during play. For example, the art and mechanics elements interact when the player provides input to make their character jump. The character model animates in response to the input at the

same time the game physically moves the character model up and down along the jump arc. Whether or not that interaction provided the player with the desired resulting experience is the next consideration after that gameplay/play dynamic plays out.

| | Element |
|-------------------------|--------------------------|
| Framework / Methodology | Gameplay / Play Dynamics |
| MDA | D |
| DPE | D |
| SGDA | |
| GLM | |

Figure 47: Coverage of the Gameplay / Play Dynamics Element by Frameworks

4.2.2.8 The Resulting Experience Element

The resulting experience element refers to the goals to be accomplished from playing the game. This is the desired end result for the player that the designer must consider. DPE and MDA include this as the resulting emotional response from the player [1, 2]. DPE builds on top of MDA and provides a clear term for the player’s resulting emotional response [2]. MDA refers to the player’s emotional response as *aesthetic* which overlaps with the common use of the term to describe the visual and auditory parts of a game [1].

DPE changes this terminology to *affect* which is a term meaning emotion or desire [2]. DPE describes how the designer should consider the affective goals of their game, which are the emotional responses the game is trying to evoke [2]. DPE provides example affective goals as the forms of fun defined by Heeter et al. [58]. These forms of fun are beauty, immersion, intellectual problem solving, competition, social interaction, comedy, thrill of danger, physical activity, love, creation, power, discovery, advancement and completion, application of an ability, altruism, and learning [2, 58]. DPE also expands on the resulting experience the designer should consider beyond the affective goal.

DPE includes ”learning” as what the player learns from playing the game [2]. Learning is one purpose a serious game could have. SGDA goes more broadly, arguing that design should

revolve around the purpose of the serious game, which could be anything from learning to changing political opinions [3]. See Figure 48 for a breakdown of which frameworks cover resulting experience.

The resulting experience targeted by the games used in this thesis’s research included the learning goal of teaching computer architecture concepts. The desired resulting experience for the player was that they would learn the desired concepts and find the games more engaging than a typical class assignment. The affective goal for the resulting experience of the games included the learning and intellectual problem solving forms of fun.

| | Element |
|-------------------------|----------------------|
| Framework / Methodology | Resulting Experience |
| MDA | D |
| DPE | D |
| SGDA | D |
| GLM | I |

Figure 48: Coverage of the Resulting Experience Element by Frameworks

4.2.2.9 The Designer Perspective

The designer perspective element refers to the view of the game from the eyes of the designer(s). A designer has a unique perspective on the game because of their background in game design. The perspective of the designer is: “Focused on creating engaging and entertaining game play” according to DPE [2]. The designer is able to suggest ways in which the game can be made fun. They will know what in-game elements, such as specific mechanics, will likely fit best with the purpose of the serious game and most likely lead to the desired resulting experience for the player. See Figure 49 for an overview of which frameworks discuss designer perspective.

Researchers working with serious games should carefully consider the designer perspective and ensure that a game designer is present for the development of the game. Without the designer perspective the game risks losing the engaging interaction that is central to any

successful game.

The designer must consider their own design perspective and be knowledgeable of holes in their own perspective. It is very rare for any one designer to have a depth of knowledge and background for every in-game element. Someone who has spent their career as an audio designer will have great input for the audio element but will likely have weaker input for the mechanics element. Likewise, a gameplay designer is likely to have great input for the mechanics element but not the audio element. Filling the holes in a team's design perspective is important to ensuring that a serious game is the best game it can be.

The designer perspective was considered in the games used in the study by Declan McClintock whose background includes the design of mechanics. Holes in the designer perspective of the team were filled in through conversations with MSU game development faculty members knowledgeable in the relevant design spaces.

| | Element |
|-------------------------|----------------------|
| Framework / Methodology | Designer Perspective |
| MDA | D |
| DPE | D |
| SGDA | |
| GLM | |

Figure 49: Coverage of the Designer Perspective Element by Frameworks

4.2.2.10 The Player Perspective Element

The player perspective element refers to the player's unique view of the game. Target audience defines who the players are, and player perspective defines how those players view the game and feel about it.

The designer(s) begin with an initial idea of what the player perspective will be based on the defined target audience for the game. The player perspective the designer(s) design around changes over the iterative design process when the designer(s) receive player feedback after playtesting, where playtesting is the act of playing a game and providing feedback to

the game’s creators.

The iterative design process is the cycle of *design, prototype, playtesting*, then *design* that repeats itself over and over again until the prototype game becomes the final release game [2]. Players that fit the defined target audience playtest the prototype game and provide the designers with their feedback, the real player perspective, which then influences the next iteration of design changes.

The player perspective is important for designers to consider because the player is who the game is being made for. MDA and DPE argue the importance of considering the player’s perspective [1, 2]. MDA puts it like this: “When working with games, it is helpful to consider both the designer and player perspectives. It helps us observe how even small changes in one layer can cascade into others.” [1]. Under MDA’s definitions, that means even small changes to the mechanics lead to changes in the resulting experience for the player which can only be fully understood by considering the player perspective. The same is true for the more complete model of DPE which discusses the influence between its layers: “When making design decisions, you must always consider the impact of those decisions on the other aspects of the game.” [2] See Figure 50 for a breakdown of which frameworks cover player perspective.

During the development of the games used in the study, the player perspective was gathered from playtesting with the game’s prototypes using the closest available players to the defined target audience. These play testers consisted of players who were already familiar with the game’s learning content and those that were not.

| | Element |
|-------------------------|--------------------|
| Framework / Methodology | Player Perspective |
| MDA | D |
| DPE | D |
| SGDA | |
| GLM | |

Figure 50: Coverage of the Player Perspective Element by Frameworks

4.2.2.11 The Academic Perspective Element

The academic perspective element refers to the researcher’s view of the game. The researcher’s perspective on the game is unique in that they know the proper data collection and analysis methods that can be applied to answer the research questions the serious game is being used to answer. DPE states that the academic is: “Interested in various academic theories, be they from educational pedagogy, communication theory, etc.” [2] See Figure 51 for a breakdown of the frameworks covering academic perspective.

The designer must consider the academic perspective during development and in the game’s design to ensure that the resulting game is capable of being applied to answer the researcher’s questions. This could involve adding data collecting methods directly into the game and ensuring study participant anonymity in that data collection. The academic perspective element could also influence the other design elements, in-game elements, and even development elements. The nature of the researcher’s question could influence the technology that can be used for example.

If the research questions are focused on virtual reality capabilities only, then the game must be made for virtual reality devices. If the research question specifies a specific demographic such as K-12 students, then the target audience is restricted to K-12 students.

During the development of the games used in the study, the academic perspective was provided by Declan McClintock, as part of his graduate student research, and Dr. Charles Owen, as the professor leading the study and primary advisor to Declan. The academic perspective provided led to the data collection methods integrated into the game.

| | Element |
|-------------------------|----------------------|
| Framework / Methodology | Academic Perspective |
| MDA | |
| DPE | D |
| SGDA | |
| GLM | |

Figure 51: Coverage of the Academic Perspective Element by Frameworks

4.2.2.12 The Content Expert Perspective Element

The content expert perspective element refers to the view from a person knowledgeable in the content area of the serious game’s purpose. In a serious game for learning, this could be provided by the teacher(s) or instructor(s) familiar with the learning content the game targets. For example, a game targeted at teaching programming should gather perspectives from those already teaching the programming concepts and languages that the game will also teach. Figure 52 provides the breakdown of which frameworks cover content expert perspective.

Additionally, if the game is more specific in the community it targets, it should also employ a content expert on that community, preferable a member of the community, to provide their perspective to make the game’s design more impactful for that specific community.

In a 2019 paper on computer science pedagogy, it was found that: “Teacher moves that encouraged students to share their personal perspectives almost always coincided with observations of increased student engagement.” [59] If the serious game can incorporate the cultural and experiential backgrounds of its target audience, then it should do so.

For the development of the games used in the study, the content expert perspective was provided by Dr. Charles Owen, who has created course materials that teach the same content that is in the games. Dr. Owen has also taught a course covering the content multiple times. The content expert perspective Dr. Owen provided covers both the learning content the game targets and the community the game was designed for.

| | Element |
|-------------------------|----------------------------|
| Framework / Methodology | Content Expert Perspective |
| MDA | |
| DPE | D |
| SGDA | |
| GLM | |

Figure 52: Coverage of the Content Expert Perspective Element by Frameworks

4.2.2.13 The Design Cohesion Element

Design cohesion refers to how well the various design elements, including the design of the in-game elements, work together. One example of design cohesion is the alignment of the various perspectives mentioned.

DPE calls the aligning of the academic perspective, designer perspective, and content expert perspective the *heart of serious game design* [2]. DPE argues that properly aligning those three perspectives leads to a whole that is greater than the sum of its parts [2]. Aligning the player perspective with the other three as well ensures that the designed game is well received by its target audience. SGDA similarly pushes for cohesiveness in serious games as a whole, arguing that cohesion supports the purpose of the serious game [3]. See Figure 53 for an overview of which frameworks discuss design cohesion and Figure 54 for the full breakdown of what frameworks cover the design elements.

DPE also discusses how its various layers can influence each other: “Certain design decisions are complementary or conflicting across the layers.” [2] A designer should seek to maximize design cohesion by recognizing the complementary aspects of their design decisions and seeking them out.

At the same time, the designer should seek to minimize the conflicts across the design elements. For example, choosing an art style that requires a cutting edge graphics card while the application context limits the technology element to mobile devices is a conflict. Whereas choosing a low poly art style that can render cheaply on a mobile device when the application context also requires the use of mobile devices is complementary.

Similarly, incorporating the advice from a content expert into the art, narrative, audio, and mechanics to match the cultural and experiential background of the target audience is complementary, so long as it is done respectfully and with proper permissions.

The design cohesion element was considered across many design decisions in the development of the games used in the study. For example, the content expert perspective and mechanics element align in that the mechanics are modeled off of what would be taught in

class. Similarly, the balance of challenges and the progression of game content was modeled off of the content expert perspective.

| | Element |
|-------------------------|-----------------|
| Framework / Methodology | Design Cohesion |
| MDA | |
| DPE | D |
| SGDA | D |
| GLM | |

Figure 53: Coverage of the Design Cohesion Element by Frameworks

| Framework / Methodology | Design Elements | | | | | | |
|-------------------------|-----------------|--------|----------------|-----------------|---------|---------------------|--------------------------|
| | Technology | Polish | User Interface | Target Audience | Balance | Application Context | Gameplay / Play Dynamics |
| MDA | | | | I | I | | D |
| DPE | D | | D | D | D | | D |
| SGDA | | | | D | | | |
| GLM | | | | D | | D | |

| Framework / Methodology | Design Elements | | | | | |
|-------------------------|----------------------|----------------------|--------------------|----------------------|----------------------------|-----------------|
| | Resulting Experience | Designer Perspective | Player Perspective | Academic Perspective | Content Expert Perspective | Design Cohesion |
| MDA | D | D | D | | | |
| DPE | D | D | D | D | D | D |
| SGDA | D | | | | | D |
| GLM | I | | | | | |

Figure 54: Coverage of Design Elements by Frameworks

4.2.3 Development Elements

Development elements are elements of the game design and development process that fall primarily under the view of the developer(s), which can also be the designer(s) for the game.

The development elements are focused on the process of making the game. The design elements focus on the thoughts and expectations of the designer. The development elements then take those thoughts and turn them into the in-game elements. The in-game elements can then confirm or refute the designer’s expectations through the use of playtesting, after which the process repeats itself until the best game is created.

The development elements include: playtesting, iterating, prototyping, asset acquisition,

communication, resources, extensibility, and documentation.

4.2.3.1 The Playtesting Element

Playtesting refers to the act of playing the game to discover issues with the game that need to be addressed. These issues can be bugs, which are ways the game is not working as intended such as the game crashing. But these issues can also be problems with the resulting experience. This would be when a play tester finishes play testing the game but did not find it engaging or impactful to its purpose.

Both MDA and DPE cover playtesting, as reflected in Figure 55. Playtesting is mentioned in MDA as part of their *tuning* process in which changes are made iteratively to better realize the design goals and remove flaws [1]. DPE covers it as part of its description of the *iterative design process* in which the game is designed, a prototype developed, the prototype play tested for feedback, and then the design is changed and the process repeats [2]. The key point to play testing is that it provides feedback for the team on whether the game is functional and whether design goals or the purpose of the game are being realized.

Playtesting can be done internally by the developers and designers to search for bugs that need to be fixed and to get an idea on whether the game is meeting its serious purpose. However, playtesting must be done with the target audience to get an accurate assessment of the resulting experience and purpose of the game at any state. Ideally this includes both first time play testers, who are members of the target audience that have never seen the game, and repeat play testers from the target audience.

First time play testers are important because they will give the team an idea of how the game would be received by the target audience if it was released as is. Repeat play testers from the target audience are important because they will develop a better understanding of the game over repeated sessions and thus provide deeper feedback over time.

Playtesting was done internally by the designer and externally with players closely resembling the target audience during the development of the games used in the study.

| | Element |
|-------------------------|-------------|
| Framework / Methodology | Playtesting |
| MDA | D |
| DPE | D |
| SGDA | |
| GLM | |

Figure 55: Coverage of the Playtesting Element by Frameworks

4.2.3.2 The Iterating Element

Iterating refers to the part of game development in which changes are made to the game based on the feedback received from playtesting. This includes the fixing of bugs and the changing of the in-game elements and design elements in an effort to better achieve the purpose of the serious game and improve the resulting experience. In MDA this is referred to as *tuning* [1] and DPE encapsulates iterating within its *iterative design process* [2]. See Figure 56 for which frameworks discuss the iterating element.

In the games used in the study, the iterating element and the iterative design process as described by DPE were used to keep the games on track to achieve their purpose and be engaging.

| | Element |
|-------------------------|-----------|
| Framework / Methodology | Iterating |
| MDA | D |
| DPE | D |
| SGDA | |
| GLM | |

Figure 56: Coverage of the Iterating Element by Frameworks

4.2.3.3 The Prototyping Element

Prototyping refers to the creation of a feature incomplete version of the game that is meant to be play tested for feedback that will then inform the iterations made. See Figure 57 for the breakdown of which frameworks discuss prototyping.

A prototype can be made to target specific areas of desired feedback and need not be digital. A paper prototype for example is a prototype created using physical media like paper and dice [2]. The advantage of making a paper prototype is that a designer can test their design, or an approximation of it, without resources needing to be spent developing the digital version.

Similarly, art prototypes can be made using a collage of images that represent the desired look for the game environment, characters, or other visual aspects. This allows the art direction of the game to be communicated and tested without creating any art assets for the game. The same can be done for mechanics by collecting video clips of the game's desired mechanics functioning in other released games. An early audio prototype could be a collection of music and sound effects not necessarily produced by the development team that fit the desired style for the game.

Prototyping, iterating, and playtesting should be done at all stages of a game's development. More complex prototypes can be made as development on the game is under way and assets are created. For example, a first playable prototype showing the mechanics of the game can be made by the programmer(s) utilizing programmer art, which is boxes and other primitive shapes. This prototype needs only the programmer's code that creates the desired mechanics in the game. It does not need any of the art created by the artists and the code need not be the final code used in the game. This prototype's purpose is to be played to receive feedback on the mechanics and how they could be improved.

Artists can do something similar by creating prototypes of the art to be used in the game. For example, the artists can create rough sketches of a character. These sketches can then be used to ascertain if the direction they are taking the character works with their own and the team's expectation of the character's personality. After some iterations, the artist can create the actual sprite or 3D model for the character that goes in the game.

Once the various assets the game needs are created and start being implemented into the game, the game itself becomes a prototype to be play tested. At this point in development

the game prototype should be kept in a play-testable state.

A play-testable state means the game has no game breaking bugs, such as those that cause a crash. Doing this allows play testing to provide the most relevant feedback on the game possible. Not doing so means that the version of the game ready for play testing does not resemble the current state of the game’s development. Which means that feedback received from play testing is less applicable to the current development state.

The games used in the study utilized prototyping to show basic functionality of the game’s mechanics, such as the dragging and dropping of puzzle pieces in circuit puzzle sections or the functionality of the camera inside of platforming levels. The games were also maintained in play testable states throughout their development.

| | Element |
|-------------------------|-------------|
| Framework / Methodology | Prototyping |
| MDA | |
| DPE | D |
| SGDA | |
| GLM | |

Figure 57: Coverage of the Prototyping Element by Frameworks

4.2.3.4 The Resources Element

The resources element refers to the people, skillsets, money, time, and other resources available to the team to be used in development. The available resources determine what designs are feasible for actual development. If the skillset of the team is not complimentary to a design, then that design can be scrapped. Similarly, if a design is too costly in money, time, or any of the available resources, it becomes unfeasible. A consideration of resources is not discussed by the frameworks, and this is reflected by the breakdown in Figure 58.

While the resources element and its imposed restrictions can reasonably be left out of the initial brainstorming of game ideas, it is valuable throughout the rest of development. In the remainder of the brainstorming process the restrictions imposed by the resources element

will quickly cut out unrealizable ideas. This saves the team from spending resources on an idea that will likely never come to fruition. Beyond brainstorming, the resources element acts as a check on whether the game’s development is still on track for completion. Either more resources need to be collected or cuts to the design need to be made if the resources are being depleted too quickly.

The resources element imposed large restrictions on the development of the games used in the study. The budget for developing the game was non-existent. The available skillsets to the team were programming, research, and teaching related. This meant that availability of art and audio assets were slim to none. Because of these limitations, resources were carefully allocated and the design tweaked to ensure the best possible game was still made.

| | Element |
|-------------------------|-----------|
| Framework / Methodology | Resources |
| MDA | |
| DPE | |
| SGDA | |
| GLM | |

Figure 58: Coverage of the Resources Element by Frameworks

4.2.3.5 The Asset Acquisition Element

Asset acquisition refers to the purchasing, creating, or otherwise obtaining of assets to be used in game. These assets include music, sound effects, 3D models, sprites, code, levels, etc. The asset acquisition element requires the developer(s) to consider how assets are going to be acquired for the game. Will they be created internally or purchased from elsewhere? Regardless of how assets are acquired, they are the essential building blocks for the game and therefore their obtainment is a serious concern. Figure 59 shows how the frameworks are missing asset acquisition in their discussion of serious game design and development.

Asset acquisition requires the developer(s) to consider the resources they have available. Specific assets are more easily created with complimentary skill sets. A trained 3D artist

will have an easier time making a quality 3D model than a sprite sheet. If the skillsets the team needs for a design are not available, then the developer(s) must consider how their other resources can fill that gap.

Money can be used to buy existing assets off of stores such as Unity’s Asset Store or Turbosquid [60, 61]. But finding existing assets that fit in well to the design of the game can be difficult. The developer(s) must manage their resources to keep development flowing smoothly, and they must coordinate with the designer(s) on limitations of resources because that availability will affect the design process.

Asset acquisition in the development of the games used in the study was limited by the available resources. While all of the code was made by the team, the lack of art related skillsets meant that very little quality art could be made by the team. This issue was resolved by reaching out to collaborators that were willing to allow the use of their work in the project.

| | Element |
|-------------------------|-------------------|
| Framework / Methodology | Asset Acquisition |
| MDA | |
| DPE | |
| SGDA | |
| GLM | |

Figure 59: Coverage of the Asset Acquisition Element by Frameworks

4.2.3.6 The Communication Element

The communication element refers to the building and maintaining of relationships as well as the transfer of information between team members. Properly building and maintaining the relationships between team members ensures a smoother development process, as it eliminates conflicts between team members. Transferring information efficiently similarly improves the development process. Figure 60 shows how the frameworks are missing a discussion of the communication element.

Communication is an important aspect to every team-based project, including serious game design and development. A serious game cannot be made easily without being able to effectively communicate the research goals, game design ideas, and purpose between the team members. Similarly, the asset creation process of a game in general is hampered by lack of communication.

Games have many pieces, typically developed by different people. In order to put the pieces together, ideally in a way that produces game cohesion, the artists, programmers, designers, and so on need to be in tight communication. A serious game developed with strong communication is more likely to have good game cohesion and design cohesion than a serious game developed without.

Tight communication was maintained during the development of the games used in the study. The development team met at least once a week and utilized the task management software Trello [62] to keep track of development. This made it easier for the team to make changes and cuts throughout the development.

| | Element |
|-------------------------|---------------|
| Framework / Methodology | Communication |
| MDA | |
| DPE | |
| SGDA | |
| GLM | |

Figure 60: Coverage of the Communication Element by Frameworks

4.2.3.7 The Extensibility Element

Extensibility refers to how easy it is to continue and build upon the project or any of its parts. Extensibility is important to consider throughout the development process and in design because any task that is finished using an extensible solution will make new related tasks easier to complete. The frameworks do not discuss extensibility and this is reflected in the breakdown in Figure 61.

An example of extensibility from the coding discipline would be creating scripts following a coding standard. A coding standard is a set of rules for programmers on a team that outlines how they should write code. This includes rules on how to write comments such as removing any large blocks of commented out code. Other rules in a coding standard can include how to name variables and classes. Coders on a team are able to easily work on each other's scripts if they all follow the same coding standard. This then improves extensibility as any new task related to a script can be picked up and done more easily by any programmer, not just the original creator.

This of course applies more generally. Every discipline has its own set of best practices that should be followed. Making sure the team follows best practices is a good way to improve extensibility on a project. However, there are threats to maintaining extensibility.

Perhaps the biggest threat to extensibility is simply the availability of resources and their proper application. When the team is put under a difficult deadline the likelihood of best practices being followed decreases, or it is decided that best practices be dropped purposefully in favor of faster yet more error prone development. The solution to this is to manage resources effectively. However, these situations sometimes become difficult to avoid given that the iterative nature of game design and development can produce unforeseen problems.

Beyond extensibility inside of development, there are also ways to provide extensibility beyond the end of development. One way is to package the completed game in a publicly available repository. That way the game can be used again in future research. Another even better way is to package the entire game project, including relevant documentation and assets so that future researchers can both use the game and make changes to it. Doing either of these things provides a way for the serious game to keep making an impact beyond its original application context.

Extensibility was maintained during the development of the games used in the study. A coding standard was established prior to the start of the games' development and the coding

standard was followed throughout the development. The computer architecture games source project is available online [17].

| | Element |
|-------------------------|---------------|
| Framework / Methodology | Extensibility |
| MDA | |
| DPE | |
| SGDA | |
| GLM | |

Figure 61: Coverage of the Extensibility Element by Frameworks

4.2.3.8 The Documentation Element

Documentation refers to the creation of documents related to the design, development, or use of a serious game. Documentation is unfortunately not covered by the frameworks and this is shown in Figure 62, and a full breakdown of the development elements is shown in Figure 63.

Various design documents make it easier to communicate the game’s design. Documenting the development of the game provides a useful resource to inform future work. Documenting the use of a serious game makes it easier for future researchers and others to understand the game’s application context. Additionally, documenting the use of the game makes it easier to reapply the game to similar contexts or determine ways to change it for different application contexts.

Examples of design documentation include: story boards, which outline the flow of the narrative; high concept documents, which outline what the game is and its goals quickly; and game design documents, which contain a detailed description of the game’s design. These examples all make it easier to communicate the game’s design through a combination of visuals and text.

Development documentation is any store of information that tracks the development process. Project management tools such as Trello [62] and Jira [63] are examples of development

documentation as they keep track of tasks, when those tasks were completed, how long it took to complete those tasks, and who completed them. Another example of development documentation is meeting notes, which serve as a reference for what was discussed in a meeting and what decisions were made during it. Overall, development documentation provides the team with tools to see their development trajectory and make appropriate development decisions.

Documenting the use of a serious game generally falls under the resulting research papers produced from applying the game. The application context, the purpose and goal, and the actual realized impact of the serious game is typically outlined in research papers.

Although it is worth noting that research paper publications are often under a page limit and so certain details might get left out. This means that it would be useful for researchers to create a more detailed document on the game’s application which can provide more detail. Then future researchers or educators can reference this document when using the serious game.

The documentation element should be considered throughout the development process. Maintaining good documentation makes it easier to communicate design ideas, track and guide the development process, and further any academic pursuits.

Due to resource constraints, documentation for the computer architecture games was limited. The documentation that does exist is included with the source project [17] and the academic design research documentation is discussed in McClintock [51].

| | Element |
|-------------------------|---------------|
| Framework / Methodology | Documentation |
| MDA | |
| DPE | |
| SGDA | |
| GLM | |

Figure 62: Coverage of the Documentation Element by Frameworks

| Framework / Methodology | Development Elements | | | | | |
|-------------------------|----------------------|---------------|-------------|-----------|-------------------|---------------|
| | Playtesting | Iterating | Prototyping | Resources | Asset Acquisition | Communication |
| MDA | D | D | | | | |
| DPE | D | D | D | | | |
| SGDA | | | | | | |
| GLM | | | | | | |
| Framework / Methodology | Development Elements | | | | | |
| | Extensibility | Documentation | | | | |
| MDA | | | | | | |
| DPE | | | | | | |
| SGDA | | | | | | |
| GLM | | | | | | |

Figure 63: Coverage of the Development Elements by Frameworks

4.2.4 Serious Elements

The serious elements are elements unique to serious game design and development. The academic perspective and content expert perspective elements could fit underneath this category, but they have been put under the design elements as those two perspectives are of most concern to the designers. This also allows more emphasis to be put on the element most unique to serious game design and development, the purpose element.

4.2.4.1 The Purpose Element

Purpose refers to the impact the serious game is trying to have on the player. This can include anything from learning to changing opinions. Because purpose is the primary point of making a serious game, it exerts influence over all the other elements. SGDA is the framework that puts the most emphasis on purpose, as reflected in Figure 64.

As SGDA puts it: “The driving force that functions as the pivotal influence over the elements of the game design should be the purpose of the game.” [3] This view is reflected in the other frameworks in how they all argue that mechanics should be tightly coupled with the purpose of the game [1, 2, 4]. Altogether, these frameworks make it clear that a serious game’s purpose should be the dominating concern in any design decisions. Similarly, purpose should govern development decisions.

The clearest example of how purpose influences development is the loop of the prototyping, playtesting, and iterating elements. The purpose of the game needs to be considered during playtesting to accurately judge whether the purpose is being achieved or not. Then that purpose relevant feedback is incorporated into the changes made in iterating which then culminates in the next prototype.

The purpose of the games used in the study [53] was to teach undergraduate computer science students computer architecture concepts. This purpose influenced every decision made in the design and development of the games.

| Framework / Methodology | Serious Elements |
|-------------------------|------------------|
| | Purpose |
| MDA | |
| DPE | |
| SGDA | D |
| GLM | |

Figure 64: Coverage of the Serious Elements by Frameworks

4.3 Summary

The design and development process of the computer architecture games in the CircuBot project [17] was informed by four existing serious game design frameworks/methodologies: MDA, DPE, SGDA, and GLM [1, 2, 3, 4]. During development, more elements of design and development beyond those mentioned by the frameworks were extrapolated and influenced the direction of the game’s design and development.

Using multiple frameworks to inform the design and development process of the computer architecture games led to more elements in those processes being adequately addressed than if only a single framework had been referenced. The frameworks were particularly useful in providing the iterative process from DPE which combined with the discussion of cohesion from SGDA led to a design for the games that emphasized their learning goals primarily through mechanics and supported by the other elements. However, there are still elements

found during those processes that the frameworks did not account for which had influence over the design and development of the games: polish, application context, resources, asset acquisition, communication, and extensibility.

Some of those missing elements exerted a large influence on the process for the computer architecture games. Application context, resources, and asset acquisition in particular put limitations on the design. This included pushing the technology the game was built on to be web browsers and forcing the use of whatever art could be obtained at no cost. It is not the fault of the frameworks for not discussing these elements, as the frameworks are made to focus on specific topics. However, it is worth noting that even with three frameworks referenced in the design and development process, there were still important elements encountered that perhaps should be incorporated into future frameworks. One goal of the work discussed in Chapter 6 is addressing this issue with a framework that not only puts these elements alongside each other but also details how they interact with each other.

5 Applying CircuBot:

Single Narrative versus Multiple

Portions of this chapter appeared in the following publications:

Declan McClintock and Charles Owen. 2021. Common narrative in educational video games: a design of games to teach circuits. In *The 16th International Conference on the Foundations of Digital Games (FDG) 2021*, 1–4.

Declan Andrew McClintock and Charles B. Owen. 2024. Analyzing Differences in Student Engagement Between a Single Narrative Game Intervention and Multiple Narrative Games Intervention in an Undergraduate Computer Organization and Architecture Course. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 812–818. <https://doi.org/10.1145/3626252.3630779>.

5.1 Introduction

The computer architecture games were applied in an Institutional Review Board (IRB) approved study conducted at Michigan State University with the Study ID STUDY00005010. This research was determined to be exempt under 45 CFR 46.104(d) 1. The consent form and all other study materials were approved by the IRB. This study looked into the impact of a single cohesive narrative game covering three sets of learning content on student engagement compared to using multiple games with self-contained narratives each covering a single set of learning content. The preliminary results of the study did not show a substantial difference between the single game group and the multiple game group [52]. More thorough analysis with the full data set provides reasonable evidence for this null result [53].

5.2 Research Question

Existing research suggests that games are powerful tools for engaging students and improving learning outcomes [7, 8, 9]. Most existing research focuses on the application of a single game in a course context. This is often compared to traditional course methods to see the effect of the game intervention on engagement and learning outcomes. With the comparisons to traditional instruction so well explored, one next area of interest is examining the best ways to apply games in the classroom [7]. An application choice an educator may have to make is whether to have a singular game or several separate games developed to cover their learning content. It would seem that a single game could create a larger narrative and common world space and, therefore, be more engaging. However, a set of multiple heterogeneous games may actually be more engaging due to the novelty of each of the component games.

Additionally, there are many different design considerations that go into making effective educational games [2, 3, 1]. It is important that any games applied in a classroom follow good design principles for any study of their application to accurately reveal what is good practice.

This study makes use of four games designed and developed following design principles outlined by existing research [2, 3, 1, 51] in order to examine differences in student engagement and learning effectiveness. One of these games formed the first treatment. That game covers three modules of the learning content of an undergraduate computer organization and architecture course as a single game with one ongoing narrative. The three other games formed the second treatment. Each of these games have a separate narrative and each covers one of the course modules. From this we have the following research question:

RQ1 Is it better for student engagement to utilize a singular narrative game to teach more learning content under one narrative context or to use multiple games with differing narratives to teach the same content in a computer science course?

In this study, student engagement is measured by continued interaction analyzed through

a continuation ratio model (CRM) and self-reflective statements from survey questions modeled after the Game Experience Questionnaire (GEQ)[64, 65].

5.3 Study Design

5.3.1 Study Population

Student participants were recruited by email from four different offerings of MSU's CSE 320 computer organization and architecture class across four semesters over two years. Participants were offered extra credit in the class to participate in the study. Students who consented to participating in the study were randomly assigned to one of two groups, the single game group (SGG) or the multiple game group (MGG). A total of 336 students opted to participate in the study, 185 in SGG and 151 in MGG. The disparity in numbers of participants between the two groups is a result of randomizing which group students were invited to be a part of. An equal number were invited to participate in either group.

The course is asynchronous online and students interact with all course content through a website. The games were hosted and playable on the same website. The games were only accessible to students who read and signed a consent form. Participants in the study could start and stop playing the games as many times as they liked through the entire semester, with access to the games ending with the end of the semester.

5.3.2 The Games and Groups

SGG played a singular game that covers all targeted learning content, henceforth referred to as the Single Game (SG). MGG played a set of three games created by splitting SG into three parts, one for each set of learning content covered by SG. The games used in the multiple game case were then given their own narratives that would exist only within each game individually and not tie into the others.

The first game in the three game series played by MGG covers combinatorial circuit logic

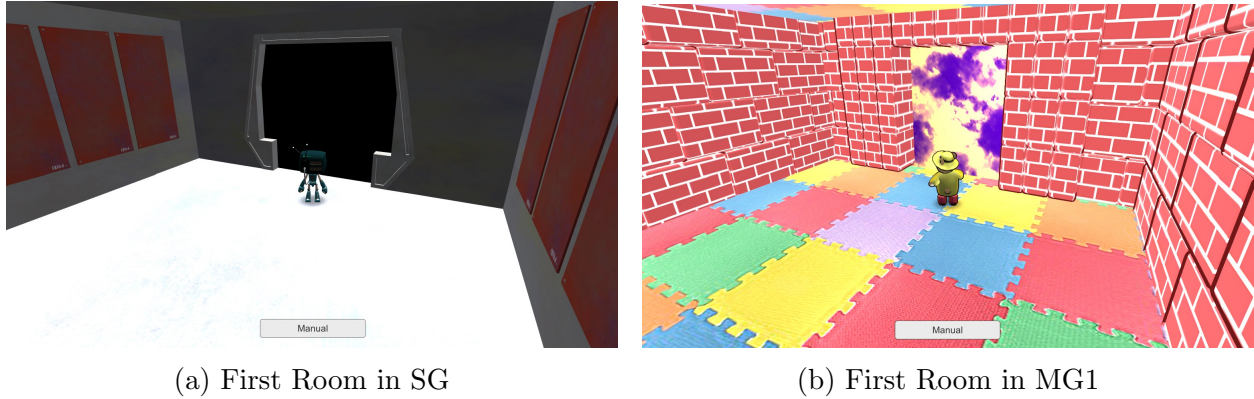


Figure 65: Comparison of SG and MG1

content and will be referred to as Multiple Game 1 (MG1). The second game in the series covers the implementation of combinatorial circuits and will be referred to as Multiple Game 2 (MG2). The third and final game in the series covers the Arithmetic Logic Unit (ALU) learning content and will be referred to as Multiple Game 3 (MG3).

The differences between SG and the series of games MG1, MG2, and MG3 are the changes made for the narrative, which included some art changes. The other in-game elements such as mechanics were kept the same. The dialogue system works the same way, the circuit puzzles are the same, the platforming and level design are physically the same. Only changes related to creating new narratives were done to games in the series of games. Figures 65a and 65b show how the first game in the series, MG1, compares to SG.

The physical space is laid out the exact same way, yet the visuals are changed to match the new narrative given to MG1. The visuals in the circuit puzzles stay the same. In MG1, the narrative is that a teddy bear has been trapped in a toy land by an evil wizard who uses circuits as magic. The dialogue spoken reflect this narrative and the images shown in dialogue reflect the new characters and their conflict. The second game in the series, MG2, is similar in its differences compared to SG. Figures 66a and 66b show how the second game in the series, MG2, compares to SG.

In MG2, the narrative is that the player is an astronaut sent to repair a space station. The game world visuals, dialogue, dialogue sprites, and characters are again changed to

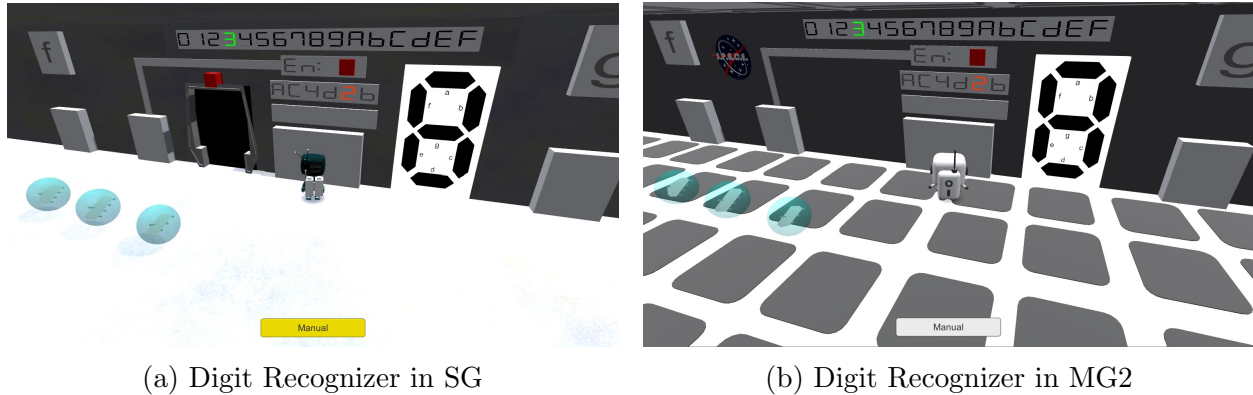


Figure 66: Comparison of SG and MG2

reflect this new narrative.

The third game in the series of games, MG3, actually has the same narrative as SG, though CircRat is removed. This was done to save development time and because it still provides a third game for the series that has a narrative not related to the other two in the series. CircRat was removed to simplify the story initial presented to the player in MG3. While CircRat acts as a central antagonist character tying together the player's struggles in SG, a mentor character called HQ provides narrative prompting to the player for the narrative context to drive MG3.

5.3.3 Data Collection

5.3.3.1 Gameplay Data Collection

Code in the games collected gameplay data automatically as students played the games and attached it to a participant ID. This data was stored on a secure web server. The data collected included the following with timestamps: whenever a game was accessed, whenever a game was completed (reached the end game screen), whenever a circuit puzzle was completed (correct solution checked), whenever a circuit puzzle was failed, whenever a gate was added to a circuit puzzle, and whenever a gate was removed from a circuit puzzle. This data was used to track how far each participant made it in the game or game(s) they played.

5.3.3.2 Survey Data Collection

Participants were asked to complete pre-surveys before starting to play the game or games. These pre-surveys included standard demographic questions as well as knowledge questions related to the content covered in the games the participants were about to play.

Participants were asked at the closing of their access to the game(s) to complete a post-survey which included the same knowledge questions as the pre-survey as well as a set of questions modeled off of the Game Experience Questionnaire (GEQ) [64, 65] visible in Table 7. All surveys were submitted through Google Forms.

5.3.4 Data Analysis Methods

All statistical tests use the standard $\alpha = 0.05$ for significance, $p \leq 0.05$ would be considered significant.

5.3.4.1 Gameplay Data

The game content in both cases share the same milestones of completing sections of learning content which must be progressed through sequentially. A participant must complete all of the game content covering learning content 1 before they may progress to 2, and all of 2 before progressing to 3. Because of this linear nature of progression, a continuation ratio model (CRM) [66] was used to examine the differences between SGG and MGG.

A CRM is a linear regression model particularly suited to sequential selection processes [66] such as in this study where participants may only progress after finishing prior content and are allowed to complete as much or as little as they like.

The CRM used the conventional null hypothesis that there is no difference between MGG and SGG. The model and the probabilities associated with it are shown in Figure 67. A Wald test was used to check the significance of the group and stage factors of the CRM.

A Wald test, or Wald chi-squared test, is one standard way to test significance of explanatory variables in a linear regression model such as a CRM [66]. The advantage of using

a Wald test is that it requires only one model and it produces equivalent results to the likelihood-ratio test and Lagrange multiplier tests for sufficiently large sample sizes [66].

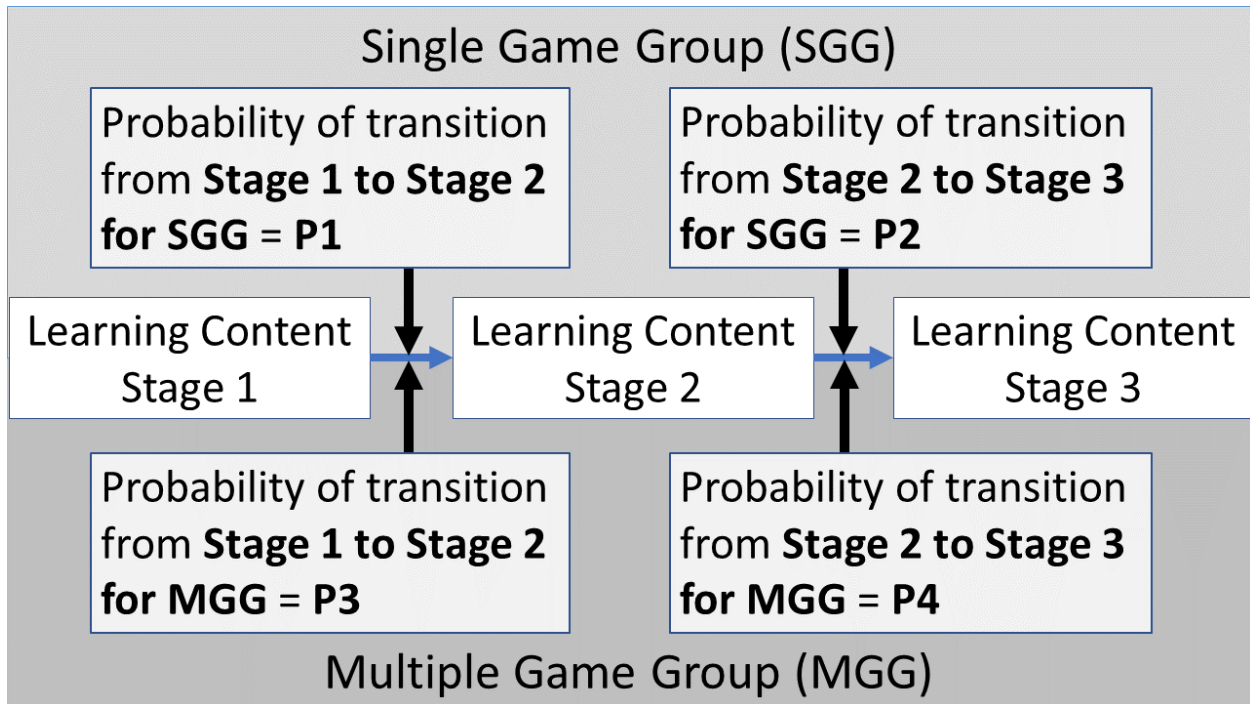


Figure 67: The Continuation Ratio Model

The probabilities in the CRM are: the probability of transitioning from learning content stage 1 to learning content stage 2 in SGG (P1), of transitioning from stage 2 to stage 3 in SGG (P2), of transitioning from stage 1 to 2 in MGG (P3), and of transitioning from stage 2 to 3 in MGG (P4).

From these probabilities there are four different odds ratios, odds ratios being an advantageous measure of effect size for a CRM like this. An odds ratio is simply the odds of one event divided by the odds of another event. This produces an unbounded measure that allows the magnitude of difference between two probabilities to be more obvious [67]. The important things to note about odds ratios are that an odds ratio of 1 implies equivalence and they can be inverted if less than 1 to clearly see the difference in the direction of the larger odds [67]. The calculation for odds is shown in Equation 1 [67] where p is the probability of the event occurring and q is probability of it not occurring.

The four odds ratios for the CRM are the following: the group effect on the transition from stage 1 to 2 shown in Equation 2, the group effect on the transition from stage 2 to 3 shown in Equation 3, the stage effect on the SGG shown in Equation 4, and the stage effect on the MGG shown in Equation 5.

$$\text{Odds} = p/(1 - p) = p/q \tag{1}$$

$$\text{Odds Ratio P1 over P3} = \text{Odds}(P1)/\text{Odds}(P3) \tag{2}$$

$$\text{Odds Ratio P2 over P4} = \text{Odds}(P2)/\text{Odds}(P4) \tag{3}$$

$$\text{Odds Ratio P1 over P2} = \text{Odds}(P1)/\text{Odds}(P2) \tag{4}$$

$$\text{Odds Ratio P3 over P4} = \text{Odds}(P3)/\text{Odds}(P4) \tag{5}$$

The CRM model was created in the R programming language using the `glm` function, the function for generalized linear models, of the R stats package [68]. R was also used to calculate the odds ratios and test significance of the results using the Wald test. The tests were performed using the Lavaan package's `wald.test` [69].

5.3.5 Survey Data

Of the 336 students who opted to participate in the study, 161 submitted at least their pre-survey, 117 submitted at least their post-survey, and 103 submitted both surveys. All survey data participants are subsets of the 336 participants who played the games, i.e. the 161 who submitted at least their pre-survey are from the 336 who participated in playing games in the study, the same is true for those completing the post survey. The 103 that completed both surveys are a subset of both those that completed the pre-survey and those that completed the post-survey.

5.3.6 Post-Survey Data

Of those that submitted at least the post-survey, 62 were in MGG and 55 were in SGG. A Mann-Whitney-U test compared the Likert responses between SGG and MGG to the questions modeled after the GEQ. This statistical test is acceptable for five-point Likert items [70] and has been used the same way in a similar study [47].

5.3.7 Pre- and Post-Survey Data

Of the 103 participants that completed the pre and post survey, 47 were in SGG and 56 in MGG. An unpaired two-sample Wilcoxon test was used to compare the change in pre-test to post-test scores between SGG and MGG. This test was used instead of an unpaired two-samples t-test because a Shapiro-Wilk test [71] showed that the data was not normally distributed.

The pre-post test quiz included eight multiple choice items structured similarly to the simplest quizzes in the course participants were recruited from. The questions would ask the participant to correctly identify a circuit component such as an OR gate from an image or to correctly identify the behavior of a circuit. For example, one question asked was: If the input to a four bit arithmetic right shift is 1010 what is the output? The primary purpose of this test was to ensure that students did actually improve in their basic understanding of the content covered in the games.

5.4 Results

The results of the gameplay data, pre-survey, post-survey, and relevant data across pre and post-survey were analyzed separately with only the participants that provided the relevant data in each. Since the gameplay data is not being examined across demographic data, it can be analyzed using the full set of 336 participants.

5.4.1 Gameplay Data

A total of 336 students opted into participating in the study, 185 in SGG and 151 in MGG. The disparity in numbers of participants between the two groups is a result of randomizing which group students were invited to be a part of. An equal number were invited to participate in either group.

In SGG, 50 participants made the transition from stage 1 to 2 and of those 50, 40 made the transition from stage 2 to 3. In MGG, 37 made the transition from stage 1 to 2 and of those 37, 28 made the transition from stage 2 to 3.

The results from the CRM are summarized in Table 1 as probabilities and odds ratios. The Wald test for the significance of group on the transition of stage 1 to 2 had these results: $x^2 = 0.57, df = 1, P(> x^2) = 0.45$. The Wald test for the significance of group on the transition of stage 2 to 3 had these results: $x^2 = 0.23, df = 1, P(> x^2) = 0.63$.

The Wald test for the significance of the stage transition on transition rates in SGG had these results: $x^2 = 23.4, df = 1, p(> x^2) = 1.3e^{-06}$. The Wald test for the significance of the stage transition on transition rates in MGG had these results: $x^2 = 18.7, df = 1, p(> x^2) = 1.6e^{-05}$.

| | Single Game Group | Multiple Game Group | Group Effect Odds Ratios |
|---------------------------------|------------------------------|------------------------------|---------------------------------|
| Stage 1 to 2 | P1 = 0.3704 | P3 = 0.3246 | Odds(P1) / Odds(P3) = 1.224 |
| Stage 2 to 3 | P2 = 0.8000 | P4 = 0.7568 | Odds(P2) / Odds(P4) = 1.286 |
| Stage Effect Odds Ratios | Odds(P1) / Odds(P2) = 0.1471 | Odds(P3) / Odds(P4) = 0.1545 | |

Table 1: CRM Results, probabilities of transitions by group, and odds ratios

5.4.2 Survey Data

5.4.2.1 Pre-Survey Data

161 participants filled out the pre-survey, 77 in SGG and 84 in MGG. Gender self-identification is summarized in Table 2. In SGG, 61 (79.2%) identified as male and 16 (20.8%) as female. In MGG, 67 (79.8%) identified as male, 14 (16.7%) as female, and 3 (3.5%) as non-binary.

| Gender | Single Game Group | Multiple Game Group |
|-------------------|--------------------------|----------------------------|
| Male | 61 (79.2%) | 67 (79.8%) |
| Female | 16 (20.8%) | 14 (16.7%) |
| Non-Binary | 0 (0%) | 3 (3.5%) |

Table 2: Gender Self-Identification

Age is summarized in Table 3. In SGG, 77 (100%) identified their age in the range of 18-25 years. In MGG, 83 (98.8%) were 18-25 and 1 (1.2%) was 26-30.

| Age | Single Game Group | Multiple Game Group |
|--------------|--------------------------|----------------------------|
| 18-25 | 77 (100%) | 83 (98.8%) |
| 26-30 | 0 (0%) | 1 (1.2%) |

Table 3: Age

Ethnicity is summarized in Table 4. In SGG, 38 (49.4%) identified their ethnicity as Asian, 29 (37.7%) as Caucasian, 1 (1.3%) as African-American, 3 (3.9%) as Latino or Hispanic, 1 (1.3%) as Turkish, 1 (1.3%) as Multiracial, 1 (1.3%) Mixed, and 2 (2.6%) preferred not to respond. In MGG, 39 (46.4%) identified as Asian, 39 (46.4%) as Caucasian, 3 (3.6%) as African American, 1 (1.2%) as Latino or Hispanic, 1 (1.2%) as Arab, and 1 (1.2%) preferred not to respond.

Hours spent gaming daily is summarized in Table 5. In SGG, 15 (19.5%) stated they did not spend any hours a day playing games, 17 (22.1%) spent less than an hour a day, 26 (33.8%) spent 1 - 2 hours, 14 (18.2%) spent 2 - 4 hours, and 5 (6.5%) spent 4 - 6 hours. In

| Ethnicity | Single Game Group | Multiple Game Group |
|---------------------------------|--------------------------|----------------------------|
| Asian | 38 (49.4%) | 39 (46.4%) |
| Caucasian | 29 (37.7%) | 39 (46.4%) |
| African-American | 1 (1.3%) | 3 (3.6%) |
| Latino or Hispanic | 3 (3.9%) | 1 (1.2%) |
| Turkish | 1 (1.3%) | 0 (0%) |
| Multiracial | 1 (1.3%) | 0 (0%) |
| Mixed | 1 (1.3%) | 0 (0%) |
| Arab | 0 (0%) | 1 (1.2%) |
| Preferred Not to Respond | 2 (2.6%) | 1 (1.2%) |

Table 4: Ethnicity

MGG, 13 (15.5%) spent no hours daily playing games, 19 (22.6%) spent less than an hour a day, 23 (27.4%) spent 1 - 2 hours a day, 19 (22.6%) spent 2 - 4 hours, 6 (7.14%) spent 4 - 6, 2 (2.4%) spent 6 - 8, and 2 (2.4%) spent 8 or more.

Gaming self-identity is summarized in Table 6. In SGG, 13 (16.9%) described themselves as hardcore gamers, 54 (70.1%) as casual gamers, 14 (18.2%) as not at all a gamer, and 3 (3.9%) as none of those descriptions. In MGG, 9 (10.7%) described themselves as a hardcore gamer, 51 (60.7%) as casual gamers, 14 (16.7%) as not at all a gamer, and 3 (3.6%) stated that none of the 3 other available responses described them.

While not all participants completed their pre-survey, those that did show that it is likely the two groups are indeed similar in demographic aspects due to the random assignment. Additionally, this provides an understanding of the demographic makeup of the class context the games were applied in.

| Hours Spent Gaming | Single Game Group | Multiple Game Group |
|---------------------------|--------------------------|----------------------------|
| 0 | 15 (19.5%) | 13 (15.5%) |
| < 1 | 17 (22.1%) | 19 (22.6%) |
| 1 - 2 | 26 (33.8%) | 23 (27.4%) |
| 2 - 4 | 14 (18.2%) | 19 (22.6%) |
| 4 - 6 | 5 (6.5%) | 6 (7.14%) |
| 6 - 8 | 0 (0%) | 2 (2.4%) |
| 8 or More | 0 (0%) | 2 (2.4%) |

Table 5: Hours Spent Gaming Daily

| Gaming Self-Identity | Single Game Group | Multiple Game Group |
|----------------------------------|--------------------------|----------------------------|
| Hardcore | 13 (16.9%) | 9 (10.7%) |
| Casual | 54 (70.1%) | 51 (60.7%) |
| Not at All a Gamer | 14 (18.2%) | 14 (16.7%) |
| None of These Describe Me | 3 (3.9%) | 3 (3.6%) |

Table 6: Gaming Self-Identity

5.4.2.2 Post-Survey Data

117 of the 336 participants completed their post survey, 55 from SGG and 62 from MGG. Mann-Whitney-U tests [70] performed on the post-survey questions modeled after the GEQ [65] showed no significant differences between SGG and MGG. This is summarized in Table 7.

5.4.2.3 Pre and Post Survey Data

Of the 103 participants that completed the pre and post survey, 47 were in SGG and 56 in MGG. Comparing the change in knowledge question scores from pre to post test between SGG and MGG with an unpaired two-sample Wilcoxon test in R produced $W = 1044.5$ and

| Question | W | p value |
|----------------------|------|---------|
| I lost track of time | 1647 | 0.7394 |
| I felt bored | 1714 | 0.9602 |
| I felt skillful | 1807 | 0.5413 |
| I felt frustrated | 1781 | 0.6651 |
| I felt challenged | 1781 | 0.6496 |
| I felt successful | 1937 | 0.1465 |
| I found it tiresome | 1780 | 0.6700 |

Table 7: GEQ Mann-Whitney-U Test

$p - value = 0.05807$. SGG had an average improvement of 14.6% and MGG had an average improvement of 6.38%.

5.5 Discussion

5.5.1 Interpretation of Results

The odds ratios show that the odds of continuing from stage 1 to stage 2 or from stage 2 to 3 for SGG are higher than the same odds for MGG, 1.224 times and 1.286 times respectively. However, the wald test reveals that this difference is not a significant predictive factor in the model ($p = 0.45$ for stage 1 to 2 and $p = 0.63$ for stage 2 to 3). The effect is in a direction favoring the singular narrative game approach but is not significant.

This suggests that the answer to the RQ1 in this study is that there is no strong evidence from the data in this study suggesting that a singular narrative game covering more learning content should be favored over multiple games of differing narratives covering the same content in a CS class context. The same is true of the reverse. The facts that the demographics across the SGG and MGG groups are similar, the data spans four semesters and two years of students, the sample size for the gameplay data is relatively large, and the GEQ questions asked had no significant differences in responses across SGG and MGG all support this conclusion.

Interestingly, the stage transition being made was a significant predictor in the CRM. The effect is similar in size across both SGG and MGG and favors the transition from stage

2 to 3 over the transition from 1 to 2. The odds of a participant in SGG continuing on to stage 3 content after completing content in stage 2 is 6.798 times ($1/0.1471$) higher than the odds of a participant in SGG continuing with content in stage 2 after completing stage 1. In MGG the similar difference is 6.472 ($1/0.1545$) times as likely to continue into stage 3 content after completing stage 2 over continuing into stage 2 content after completing stage 1. This means that participants were much more likely to continue engaging with content if they had already completed the prior content regardless of group.

5.5.2 Practical Suggestions

The results suggest that CS educators and game developers/designers could reasonably take either the larger single narrative game approach or the multiple game approach. It may be beneficial to favor the multiple game approach considering existing educational games covering sets of CS content don't always cover all aspects of a content area in the same detail or are necessarily even still accessible to educators [10].

5.5.3 Limitations and Future Work

5.5.3.1 Engagement

This study focused on comparing participants playing either a singular game with narrative tying together multiple sets of learning content against a set of games with differing narratives covering the same content in the CS education context. This means that generalizations should be properly contextualized. For example, while the specific difference in application of two narrative educational game approaches did not have significant differences between them in this study, it is possible that non-computer science contexts may find different results as their average student may react differently.

Additionally, this study was done on an intermediate-level CS class meaning that the participants come from a different pool than if a similar study was done on an introductory course. It may be worth exploring similar studies with games made to teach lower level

undergraduate CS courses as the student body may be substantially different, and differences in how games are applied may lead to retaining more students in CS programs that otherwise would leave after their first classes. Indeed, some recent research has suggested that video game play can have a positive impact on how student's feel about STEM [72] which suggests this direction is worth exploring further.

In regards to the effect of the group, it is still possible that a group effect does exist at the small size suggested but not significantly supported by this study's data. However, if future research using a similar CRM approach explores this, the numbers needed to observe and significantly support an effect that small will be at least an order of magnitude larger (around three thousand) in order to provide strong evidence that would impact practice. A power analysis was done using the WebPower package in R [73, 74] using the group effect probabilities from this study and standard α of 0.05 and β of 0.80 to get an idea of the sample size future researchers may need.

5.5.3.2 Knowledge Questions

While the difference in improvement of the knowledge question scores between the pre- and post-tests was *near* significance ($p = 0.05807$, with $p \leq 0.05$ as the standard for significance) potentially meriting further study, the result is subject to some important caveats.

The knowledge questions consisted of a total of eight multiple choice questions. This amounts to a simple quiz that only touches on the remember level of Bloom's taxonomy [75]. Future work may want to explore this difference in more detail by using tests that touch on higher levels of Bloom's taxonomy such as the concept maps approach used in Coller and Scott [5] where they found that their simpler measures of differences in learning did not show significant differences but their deeper measures of learning did.

Future work would ideally involve using the games directly in a course, which is the second important caveat. The games in this study were approved only for use as opt-in extra credit tasks alongside the course they were designed for. This means that student participants had

a significant confounding factor for any measure of learning content differences, although it should apply equally to both randomized groups. Any future study looking into learning content differences should aim to integrate the games directly in the classroom.

5.5.3.3 Additional Avenues for Future Work

This study looked only at adjusting the narrative element, along with any necessary changes to art, to observe the relevance of its impact on student engagement. Similar studies could be designed to look at adjustments made to other elements, and at different adjustment sizes. The changes made in this study to the narrative element don't really change the narrative quality all that much, rather it makes a small change that removes the narrative ties between learning content. These ties exist in SG for the single game group but not for the series of games for the multiple game group. A different study might make a more drastic change to a different element, such as including audio in one iteration of the game and comparing it to another where the audio is completely removed. Another example might be to compare games using different mechanics which might be a more interesting comparison given the field's existing interest in, and understanding of, how important mechanics are to serious games.

5.6 Summary

This study examined whether it is more beneficial for student engagement to utilize a singular game with a narrative tying together sets of learning content or to use multiple games of differing narratives to cover the same content in an undergraduate computer organization and architecture course. The study found insignificant evidence to suggest that there is a substantial difference between either educational game application approach in terms of engagement at a reasonable effect size. This suggests that educators in this area likely do not need to strongly consider this difference when choosing what games to apply to their classroom environments if all other aspects of the games themselves are equal.

One next step for research in this area is to study other application contexts in different classrooms. Another is applying the same approach but with games tightly integrated into the coursework in order to examine in proper detail any differences in learning between a singular narrative game approach and a multiple narrative games approach. Yet another is to examine variations across other design elements in applied serious games. The results of this study and future studies like it will then allow educators to be properly informed on how to best apply educational games as more become available.

6 An Iterative Game Design and Development Framework

6.1 Introduction

Existing game and serious game design frameworks are understandably focused on the design aspect of making games and serious games [1, 2, 3, 76, 77]. However, the development process of games also exerts an influence on a game’s design and therefore needs to be considered and understood when designing games.

The framework presented in this chapter combines the important design elements recognized in existing frameworks, game intervention papers, and design research papers with game development elements recognized in software development. The result of this combination will be a tool with clear practical guidelines that provides games researchers and developers with a more comprehensive view of the entire game design and development process that can then be used to create new games, assess old ones, and facilitate design research.

6.1.1 The Interdisciplinary Nature of Game Design and Development

Game development is multidisciplinary in nature in the sense that many disciplines must come together to create the game [78, 79]. If taken a step further, game development becomes interdisciplinary when the work from the disciplines blend together into something completely new and unique [79]. Borrowing Repko et al.’s [2020] two metaphors, multidisciplinary is a fruit salad where each discipline is represented by a fruit closely placed to other discipline fruits. The arrangement does not mix these disciplinary flavors together meaningfully. Whereas, interdisciplinarity is a fruit smoothie. The disciplinary fruits are blended together into something completely new where the individual discipline fruit flavors are not easily recognizable, resulting in a completely new experience [79, 80].

How does this apply to game design and development? Consider the proverbial "chocolate covered broccoli" [81, 82] discussed in educational games research where learning content is not aligned with game mechanics. This is the same as the "fruit salad" in that the discipline of education has not been blended with the disciplines of games resulting in the "chocolate covered broccoli" that is not tasty, failing to effectively improve learning outcomes or motivate learners as much as it could. By contrast, the "fruit smoothie" for games happens when the pieces of a game are cohesively aligned with each other, blended together. The relevant disciplines: art, programming, audio, narrative writing, and so on, all work together rather than separately. Let's consider one successful entertainment game as an example, *Stardew Valley* [2016] which has sold over 20 million copies [84]. *Stardew Valley's* [2016] *narrative* of escaping from monotonous work life to farm life is supported by its whimsical and relaxing sound track (*audio*) [85]. The *mechanics* support the escapism to blissful productivity and community through routine tasks and charming interactions with villagers supported by the portraits (*art*) which show the villagers reactions to you (the player) and their emotions in special story interactions. The point is, the pieces do not stand alone, they are tightly woven together and interact to support each other.

Can this be done for serious games? Yes. In fact, there are solid examples of games that blend their elements, including the educational or other purposeful pieces, all together. Consider the Indiecade award winning *When Rivers Were Trails* [2019] which is aimed at teaching about the land allotment in the 1890s [22, 23]. The learning content is integrated directly into the *mechanics* and *narrative* as the player assumes the role of an Indigenous person being forced to travel westward while balancing their "physical, emotional, mental, and spiritual wellbeing... while making choices about contributing to resistances as well as trading, fishing, hunting, gifting, and honoring the life they meet..." [23] all while interacting with characters that represent the cultures and views from the time from nations including "Dakota, Lakota, Blackfeet, Apsáalooke, Nimiipuu, and many more..." [23] with Indigenous writers contributing to inform the scenarios in the game. The *art* and *audio* is similarly

tied into the learning content [23]. In this sense, *When Rivers Were Trails* [2019] is like the "fruit smoothie" where the blending has produced something greater than the individual parts rather than the difficult to stomach "chocolate covered broccoli" or non-enticing "fruit salad."

Two more serious game examples of this blending, at various levels of success, are explored by Mitgutsch and Alvarado [2012] in their Serious Game Design and Assessment framework (SGDA) in which they build a case for the cohesion and alignment of the various elements of a serious game's design to the game's *purpose* (such as educational outcomes). Winn's [2009] Design, Play, and Experience framework (DPE) similarly discusses interactions and alignments between various layers that includes the Mechanics, Dynamics, and Aesthetics framework (MDA) [1] as the *gameplay* layer. With these frameworks already touching on the interdisciplinary blending, one can justifiably ask, "why do we need the new framework presented here?" The short answer is *nuance* and *context*.

Mansilla et al. [2009] point out that students doing interdisciplinary writing "must often overcome the inclination to give each discipline an equal share" [86]. While these existing frameworks are strong in their own right, they present their important elements in a way that implies a certain equality of importance across the concepts they highlight with the possible exceptions of: a serious game's purpose [3], the learning aspect in serious games for learning [2], and perhaps the *mechanics* that make a game *a game* instead of another type of media. The problem with only a few stand out elements and an implied equality amongst the rest is that it does not provide the room for *nuance* and an exploration of the *contexts* in which some elements may be made more or less important over others. Game design and development is not rigid. The process is iterative [46, 2] and, similar to the agile methods [87] in software development, it adjusts on the fly to the present reality of the development and design. If we want to have frameworks that can best align to practical usefulness in this *agile* process, we should have a framework that is flexible to match.

6.1.2 Design, Development, and Practicality

Many existing frameworks are understandably focused more on the *design* aspects of games: MDA provides a good explanation of how *mechanics* design leads to *player experience* [1]; DPE expands on this by incorporating several more design layers alongside *mechanics* including *storytelling*, *user experience*, *technology*, among others [2]; SGDA [3] provides a strong argument for the cohesion between many game elements and the game’s *purpose* (such as learning in educational games); Annetta’s *The "I's" Have It* framework [2010] provides a perspective on educational game design grounded more in education and psychology viewpoints. While it is perhaps beneficial that these frameworks are focused on design topics and relatively small sets of concepts, the interdisciplinary nature of game design and development makes it problematic to create the best games by only focusing on design or on a subset of the important concepts relevant to creating games.

Development realities influence design. The most obvious example is *resources* (time, skillsets available, money, etc.). If a design demands more resources than are available, then the design is either dropped in favor of a design that is possible or it is adjusted to be possible. While frameworks like MDA [1] and DPE [2] touch on the iterative development nature of game design and development, the existing frameworks do not do enough to communicate the connections between development and design.

This plays into a practicality of use issue for existing frameworks as well. In O’Shea and Freeman’s [2019] summary of game design frameworks, they provide an apt criticism of academic frameworks for game design. They state that academic conferences provide information in an abstract format that is at odds with developers who need fast and *practical* knowledge to apply to their projects [88]. They further expand on this thought to argue that developers with *strict publishing milestones* and *limited budgets* can’t realistically be expected to navigate the vast abstract information within their project’s timelines because existing frameworks do not do enough to explain the practical means to use them.

O’Shea and Freeman [2019] are pointing directly at realities of development (limited bud-

get, time, strict publishing milestones) as part of the practical reality preventing our existing frameworks from having a broader impact. With that in mind, the goal of the framework presented in this chapter is to make *practical* the concepts covered by existing frameworks, explain the relationships between them and the elements of development realities, and maintain a flexibility that allows for nuance across different development and design situations.

6.2 The Iterative Game Design and Development (IGDD) Framework

The IGDD framework builds on the important elements of serious game design recognized by several prior frameworks [2, 3, 1, 76, 77], the issues identified from various game intervention studies and design research papers including the design research from CircuBot [51], and those recognized by the game development industry. This framework puts all these elements together into a larger structure where the relationships between the elements can be more easily understood and their flexibility to unique situations recognized.

The IGDD framework is shown in Figure 68 and is broken down into four main groups: the *in-game elements*, the *design elements*, the *development elements*, and the *serious elements*. This follows a very similar breakdown as done in the design research presented in the CircuBot chapter (Chapter 4) [51]. This current chapter expands on the definitions provided previously [51] by expanding on the elements, their relationships and interactions, and by structuring them within a framework with practical guidance for its use.

Elements in IGDD are ordered within their groups based on their importance as understood through the review of the literature when this chapter was written. Elements higher up in a group are considered more important or influential in the design and development process or to the success of a game. Those on the same level are considered to be around the same importance, such as the *technology*, *target audience*, and *application context* elements under the *design elements*. The *palm icon* shown in Figure 68 holding the *design cohesion* element represents that IGDD allows its pieces to be moved or even removed. This makes

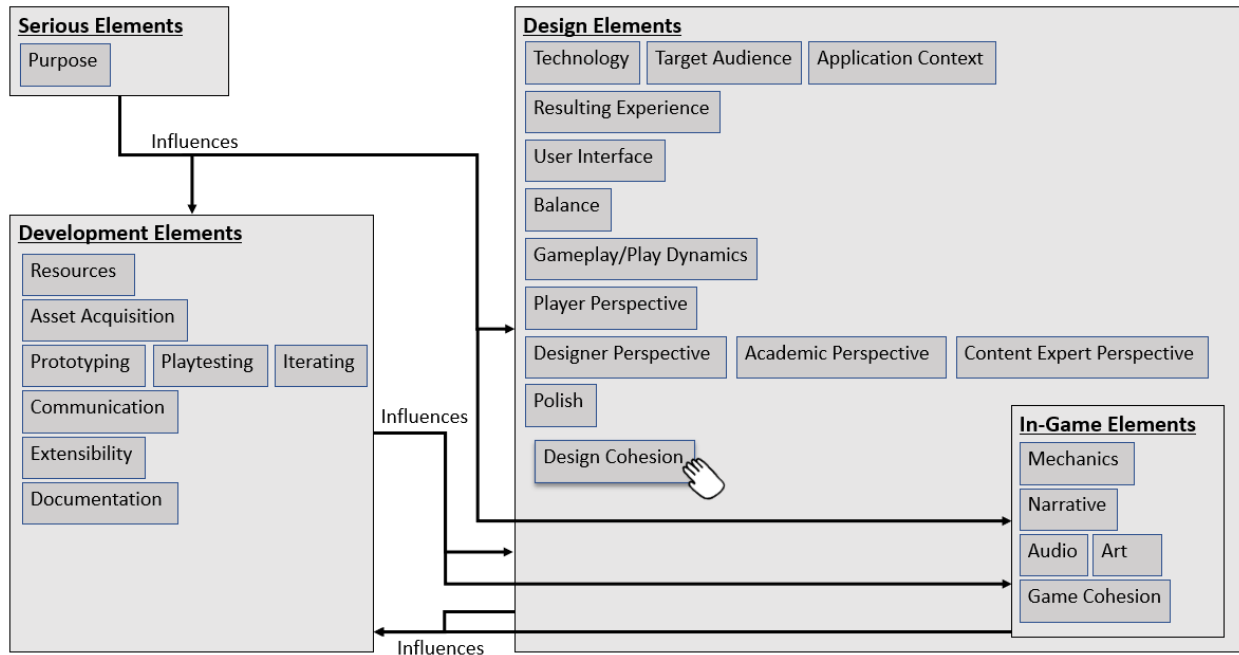


Figure 68: The Iterative Game Design and Development Framework

IGDD flexible to the unique situation that game developers may find themselves in. For example, the *mechanics* element of the in-game elements is placed at the top in recognition of how the existing frameworks put a large emphasis on a game’s mechanics. The *narrative* element follows next in the ordering presented in IGDD, however it could be moved to the top in situations where narrative becomes a greater concern. One example where this would be the case is in a narrative focused game such as *Murder on Grimm Isle* used by Dickey [2011] for “fostering argumentation writing.” In Dickey’s [2011] case, the narrative becomes more important as it provides the players with evidence they eventually use in their argumentative writing.

The arrows in Figure 68 show the direction in which groups of elements generally exert influence over other groups. For example, the *serious elements* exert influence over the *development elements*, *design elements*, and *in-game elements*. The reasons for the directions of these influences and more specific details for which elements specifically exert influence and how will be discussed as we go through the framework element groupings and individual elements starting with the elements with the most influence, the *serious elements*.

6.2.1 Serious Elements

Serious elements are those which are specific to serious game design and development and would typically not be applicable to games generally. The only element placed here is the *purpose* element. However, individual design and development situations may find other elements that fit this category and which similarly exert an influence over the other categories of elements.

6.2.1.1 Purpose

Purpose is the impact that a serious game aims to have on a player [51, 2, 3]. This purpose can be anything from educating the player [5, 6, 34] to effecting social change [38] to anything beyond simply providing an engaging experience [2]. The Serious Game Design and Assessment framework (SGDA) [3] puts purpose as the key influencing factor over all other parts of a serious game. However, the SGDA framework [3] discusses only a subset of elements that fall under the design elements and in-game elements discussed in this framework. Additionally the IGDD framework's additional development elements are also influenced by purpose as development decisions should be guided by the purpose of the serious game the same way as the design and in-game elements are. This should be the case to best ensure that the purposeful goal of the serious game is met.

Purpose influences the development elements by influencing available resources, how assets can be acquired, how communication is handled, and what documentation is needed. The example we will use for this is an educational game to be applied for research. If the specific purpose lines up with the interests of grant providing organizations such as the National Science Foundation (NSF) [2023], then the purpose has uniquely affected availability of resources. Any grant acquired may then have some requirements for how funds are used that would restrict the way assets used in the game are acquired, how communication is to be handled, and what documentation needs to be produced to continue receiving funding. Even without grant funding, a purpose that includes research may add stipulations to the

kinds of documentation needed to be produced. If the design process itself is part of the research, as is the case when conducting design research [90, 40], the developers may need to keep more notes on their own design and development processes.

Purpose holds an influence over design elements and in-game elements in a more obvious way. Every element in the design should be aimed at achieving the purpose. For an educational game, the mechanics should have the player *doing* the skill that is desired for the player to learn. The narrative in the game should, at the very least, provide ample reasoning to participate in the mechanics geared towards the learning goals and, even better, could communicate key concepts in the learning goals within the narrative itself.

Overall, the purpose of a serious game holds the key position of influence over every development and design decision made for serious games. For those making games without the serious purposes discussed here, this would be the first instance where the flexibility of this framework comes into play. Removing the serious element of purpose then leaves a framework applicable to the more general game design and development case where the development and design realities work together to influence decisions made throughout design and development processes.

6.2.2 Development Elements

Development elements are elements of the game design and development process that are primarily addressed by the developer(s) of the game [51]. The development elements exert influence over the design elements and in-game elements by determining what designs are feasible and generally governs the process in which a game's design is brought to life. Without the development elements, no game is ever published.

6.2.2.1 Resources

Resources are the people, skillsets, money, time and so on available to be utilized in the development process [51]. Resources are placed at the top of the development elements

group because they exert the largest amount of influence over the design elements and in-game elements that can be realized. Without the proper resources, some designs simply can not be achieved. For example, if a game design requires 3D art, yet the development team does not have the needed models on hand and no way to acquire them, then that design must be abandoned.

While a lack of resources may be distressing, it is important to consider available resources before starting development on a design which is later found out, the hard way, to be unachievable. Considering resources during the brainstorming part of the development process can help limit ideas down to those most likely to be achieved and thus set development up for success instead of doom it to failure. This concern around resources is reflected by the Independent Game Developer's Association (TIGA), with 33% of respondents in a survey citing finance as an obstacle to success and 30% citing missing skills [91]. The International Game Developers Association (IGDA) highlights the importance of resources in game design and development as 39% of respondents to their 2017 Developer Satisfaction Survey expressed concern over funding [92].

Available resources should also be considered throughout development. Resources available can be considered during the typical player-centered iterative design process of design, prototype, playtest, repeat to know if the design and development goals are still achievable. If the goals seem unlikely to be reachable given remaining resources, then the design can be adjusted accordingly and any necessary cuts made.

6.2.2.2 Asset Acquisition

Asset acquisition is the methods through which game assets, such as music, sound effects, 3D models, and so on, are obtained for the game [51]. This includes purchasing the assets, creating them within the development team, or any other method used to acquire them.

The methods available to the team for the acquiring of assets is important to consider as some methods have different limitations and affordances. Creating the assets using the

primary development team restricts development to only assets that can be made using the team's skillset, but hiring someone from outside the team who may have a more relevant skillset for the specific asset desired may be more expensive and require a search to find someone capable of creating the exact asset desired as well as onboarding time spent getting them familiar with the project and any proprietary tools. Buying assets from the various asset stores like the Unity Asset Store [60] or Turbosquid [61] limits developers to only what is available on those asset stores and there are no guarantees that those assets will work with the assets created by the main team, from other stores, or even other assets from the same store.

There is a lot to consider when deciding on the method of asset acquisition to use for any individual asset or set of assets. Resources, of course, influences what methods one can choose for asset acquisition. Hence, asset acquisition has been positioned after resources in IGDD's hierarchy. If developers do not have the funds to spend, then purchasing from asset stores or contracting someone from outside the team to create assets is not possible. Knowing which methods are available at all times and weighing their advantages and disadvantages will help developers pick what designs to pursue further or what changes/cuts will need to be made to finish a game already in development.

6.2.2.3 Prototyping, Playtesting, and Iterating

Prototyping in software development is described as producing early working versions of a system and experimenting with said prototypes [93]. *Prototyping* when it comes to games and serious games fits this definition. Prototyping in games includes this creation of early, or feature incomplete, versions of a game. These incomplete versions are *play tested* to gather insights and feedback on the present state of the game or some part of it. That feedback is then used to create the next *iteration* of the game. In this sense the game itself is a prototype at all stages of the iterative development process.

Given that *prototyping*, *playtesting*, and *iterating* are all core to the iterative design pro-

cess, they all occupy the same level of importance in the hierarchy of development elements in IGDD. They sit below *resources* and *asset acquisition* because those two elements apply limitations on the creation of prototypes. This does not mean that prototyping should not be done with limited resources however, in fact, early prototyping is important to do without applying a large amount of resources.

Early prototypes in game development include art and sound mood boards, which are arrangements of images (or sounds) that evoke the style desired in a specific design [94]. These are useful tools in understanding a game's art and sound direction. Other early prototypes include the paper prototype, a mock version of the game constructed out of paper and other typical office or household items used to test if the game's mechanics and rules can be fun before ever spending resources developing the digital systems for them [95]. Once the digital systems start to be created, the game can begin the core loop of the iterative design process and truly be play tested and iterated on.

Playtesting refers to playing the game with the goal of discovering issues that need to be addressed or to confirm that issues targeted in an *iteration* were successfully addressed [51, 1, 2]. Playtesting can be done by members of the design and development team to hunt down specific bugs that need to be fixed or to get an idea of the game's current state in relation to reaching its goals [51]. However, the most important playtesting to do is with the *target audience* for the game in order to get an accurate assessment of the *resulting experience* the game is conveying to players [51]. This kind of playtesting should be done with first time play testers, to get the most accurate assessment of how the game would be received if released as is, and with repeat play testers, to get more in-depth insights on the game from the target audience [51]. What exactly a *target audience* is is covered in the Target Audience section of this chapter.

With the insights gained from playtesting, the next step of the iterative design process is *iterating*. *Iterating* is the making of changes to an ongoing game prototype based on the feedback received from any playtesting. Once iterations are made, then a new prototype

exists to be play tested and have iteration made on it. The process repeats from here until release where the last prototype becomes the released game.

The usefulness of this iterative approach to game development is supported by the success and wide adoption of agile development in software development since the agile manifesto [87]; as well as the inclusion of the same ideas in the Mechanics, Dynamics, and Aesthetics framework (MDA) [1]; the Design, Play, and Experience framework (DPE) [2]; and a more recent paper explicitly pushing for the adoption of iterative methodologies in serious games [96]. How to apply the concepts and relationships outlined in this framework to this very typical game development process will be detailed in the How to Use the IGDD Framework section following the end of these sections explaining the concepts and their basic relationships to each other.

6.2.2.4 Communication

Communication in game design and development is the building and maintaining of relationships and the transfer of information between team members [51]. In software development, communication is recognized as an important factor leading to success when handled well, or alternatively leading to failure if handled poorly [97]. This is also true for game development, though game development teams are often more diverse in skillsets which makes effective communication more difficult and all the more important.

When developing games, team members should look for ways to improve the flow of communication between all team members and to resolve any confusions or conflicts as they arise. Doing so will help maintain a smooth and efficient development process. The agile methods relating to communication adopted by software developers [87] are a good starting point for developing a team's communication practices. Fowler and Highsmith argue that the "most efficient and effective method of conveying information with and within a development team is face-to-face conversation" [2001]. Yet actual face-to-face conversation can be rendered impractical, as can be observed by the work from home situations during

and post the initial Covid-19 pandemic. Importantly, what Fowler and Highsmith also get at is that what matters is ensuring *understanding* [87] which can be done by blending documentation and direct conversations.

The influence communication has over all other elements is wide ranging. Not double checking if an idea or technical requirement or feature request has been communicated properly can lead to an implementation of that idea, feature, artwork, etc. that does not match the intended original expectation. This situation is similar to the phone game where people stand in a circle and whisper one message from person to person all the way around until the last person tells the group the message which has become something totally different from the original intended message. The difference being that in the real world development context, a team has just lost a substantial amount of resources creating something that does not fit the requirements and may need to be redone.

6.2.2.5 Extensibility

Extensibility in game development is a measure of how easy it is to build upon a project or any of its pieces [51]. The idea of being able to extend an existing product is not new. In fact, the idea of making extendable programming languages has been around for some time [98], and extensibility as a term also exists within the field of human computer interaction (HCI) where it is defined as "the ability to build on the resulting outcomes of the interaction design research: either employing the process in a future design problem, or understanding and leveraging the knowledge created by the resulting artifacts." [40] However, extensibility as it is being used in this framework in relation to game development applies beyond *just* the ability to build off a project when it is finished to include the measure of *ease* of being able to build on the project or its parts at *any point* in development.

The way to ensure extensibility in the game development process is to follow the best practices outlined for each discipline and for development as a whole. For example, coding best practices are often referred to as *clean code* [99]. Clean code outlines rules for program-

mers to follow that results in code that is human readable and thus can be edited easily by any programmer on the team or easily continued on after being set aside for a long time. If clean code practices are not followed, then productivity stagnates as the mire of tangled up poorly written code prevents progress [99].

The same principles apply to the other disciplines involved. If a 3D art asset does not have its pivot set logically, its parts and materials named logically, and so on, then it becomes difficult to change later in the project.

One way to observe how extensible a project is is to have reviews of the assets in the project done by those knowledgeable about best practices for those assets. Code reviews are a common example in which one programmer reviews the code of another programmer to make sure it does what it should, that it follows best practices, can be easily understood, and can be extended. Conglomerating many reviews about the assets in a project can give the team an idea of how healthy each of its parts are and adjustments can be made accordingly. Alternatively or in addition to the above, a team could look towards the burndown charts sometimes used in industry. Burndown charts are graphs that show the remaining work on a project over time and whose slope gives an idea of the velocity (speed at which work is being completed) for the project. If the graph's slope has flattened out, progress has slowed, it may be an indication that technical debt or "mess" has been accumulated and extensibility has lowered.

Any sort of extensibility issue in one disciplinary part of the game project then hinders the progress of the project as a whole as well and lowers its extensibility. Just as when a code base becomes an unproductive mess [99], so too can a game project. Because of this, it is important to look for and follow best practices and consider a game project's current extensibility as a whole throughout the development process. Taking the time to clean house is an investment in long term success.

6.2.2.6 Documentation

Documentation refers to the creation of documents related to the design, development, or use of a game. Documentation includes game design documents (GDD), art bibles, story bibles, and so on [100, 101].

Creating documents, including meeting notes, can be very useful for facilitating effective communication. Using development progress documentation tools such as *Trello* [62] or *Jira* [63] can help track development progress as well. Other documentation tools such as burndown charts from agile development [102] can be particularly useful in knowing where a game project stands in terms of its completion. However, good documentation still comes at a cost.

While documentation can be great for understanding the state of a project and facilitating communication, resources have to be spent to have documentation done. Within a specific developmental reality, the team should consider what documentation, if any, will be worth the cost in resources to do. On small projects with small teams of one to four for example, it may be unnecessary to keep careful track of the completion of tasks through documentation as the small team size may be aware of the project's status if communication between members has been maintained well. If that is the case, then spending time tracking tasks through a document may be wasteful.

Borrowing from agile development again, the documentation chosen to be done should always match some benefit in providing understanding in the communication process for a team [87]. If there is no understanding benefit gained, then the documentation is not worth the resources to create it.

6.2.3 Design Elements

Design elements are elements that are handled by the designers on a game project [51]. These elements may inform the first cycle of the iterative design and development process but are ultimately more influenced by the development elements during the remainder of the

development process.

6.2.3.1 Target Audience

Target audience is the *who* a game is to be played by to *technology's what* a game is to be played on. The target audience for a game is considered an important design concern and guidance point both in industry [103] and in game design theory [2, 1, 3, 101, 100].

The reason to consider the target audience during the design process is because they are the ones serious game creators want the *purpose* of the game to reach, or to get the *resulting experience* to for entertainment games. For a purely entertainment-based game, this may just be to engage them as much as possible. For an educational game, creators may want to make sure that learning is happening. Defining a target audience helps make sure that a design achieves its intended purpose.

Let's use the target audience of *Antura and the Letters* [34] and the target audience of the computer architecture games discussed in McClintock [2022] and in the Design Research from Circubot target audience section of this thesis as examples to understand the influence the target audience should have on game design. For *Antura and the Letters* the target audience was "Syrian refugee children aged five to 10 with little or no schooling" [34] while the target audience for the computer architecture games was mostly third year undergraduate computer science students [51]. In *Antura and the Letters*, the game needs to be approachable to a much younger audience compared to the older audience for the computer architecture games. For *Antura and the Letters*, this means the game might benefit from a simpler and easier to understand control scheme for the younger audience as opposed to the computer architecture games that may be able to use a more complex control scheme which the older audience of technology adept students could understand. Knowing aspects of a target audience helps make good design decisions, which creators should of course check the impact of through play testing with the target audience.

Similar to technology, the target audience exerts influence over the rest of the design

decisions made. The relationship back to technology is that the designer may pick or design a technology based on what the target audience is already familiar with or other descriptive aspects about the target audience. For example, *Antura and the Letters*'s target audience generally had access to mobile devices [34]. Students in the computer architecture class were already familiar with using their web browser for class [51] so putting the games in the same place has some merit for matching with technology the students were already familiar with.

The target audience can influence what balancing gets done in the game. Is the game for "hardcore" gamers who are already knowledgeable about the typical controls and are skilled enough for the game to start with relatively high skill demands? Perhaps the audience is broader and players of many skill levels will play so balance needs to be done adaptively. A similar pattern of influence plays out for other design elements: what narrative would the target audience find compelling? what art style do they prefer? what kind of audio quality do they expect? what kind of gameplay/play dynamics are they searching for? what resulting experience do they want to walk away with? Ultimately a game's target audience is a very important design concern, especially when following player-centered or user-centered design [104, 105].

6.2.3.2 Application Context

If *technology* is the *what* and *target audience* the *who*, then *application context* is the *where* the game is meant to be played. This has a very strong influence on the design for educational games as they can be played in a few places including: in the classroom where a teacher is present and can take on active roles [11], an online class where the teacher is not actively facilitating game play [51], or played completely outside of a class context on the player's own volition.

There are many different situations in which a game could be applied and considering them will help guide a design into something that best fits that context. If a team chooses or must design a game around being played online they have the benefit of easy access for

students/players across devices with internet access but must design the game to run on web browsers which have considerably fewer resources than running directly on a personal computer. If a design is to be played in the classroom than the design of the educational game can take advantage of the presence of a teacher to facilitate play, but in doing so the game becomes tied to that context and perhaps not able to be played without a teacher present. Knowing which context a game is going to be applied in is important for designing the game to take advantage of that context's strengths while mitigating its weaknesses as much as possible.

If we consider the context of in class with a teacher present as an example, we can see some of the ways application context influences design choices. A teacher present in a game-based learning classroom can take on many roles including: planning how to cover concepts taught in gaming sessions, introducing students to the topic before game play, elaborating on concepts from the game after and during play, and guiding the process of play itself [11]. In this case, a teacher facilitating play can assist in balance across players, meaning that tutorial elements may not need to be as extremely detailed. While it might help if the game can effectively teach its own players how to play, the presence of the teacher as someone who can assist in this part of play means the development might shift to allocate resources to other important design aspects in an attempt to maximize quality. A teacher may also influence the resulting experience that the students leave the game with as the teacher can be elaborating on the game during and after play.

The context where a teacher is not present is more akin to how entertainment games are often played, without the presence of someone facilitating/guiding play. In this context the designers of serious games must carefully consider if the game is able to scaffold learning/understanding of the game's purpose throughout the start of play until the end on its own. Some games such as Age of Learning's [25] *My Math Academy*'s games use adaptive systems that adjust the difficulty of presented game challenges (and thus learning content difficulty) dynamically as the player plays them [106]. While *My Math Academy* is designed

for use in a larger educational structure, the dynamic adjusting of in-game content for the player without the direct involvement of someone else is a good example of what would be beneficial for an educational game designed to be played without teacher facilitation.

When it comes to entertainment games, the application context is also important in a variety of ways. For arcade games, the game likely would want an "attract mode" that displays interesting parts of the game with loud sound effects or music to get people attracted to the machine. Games aimed to be played at home on the other hand wouldn't necessarily need the same mode created for them as the surrounding context is more relaxed. Games meant to be played on the go through hand held devices or mobile phones must account for the technological limitations/affordances inherent to that gaming situation such as specific input controls like D-pads, small touch screens, and a limited number of buttons compared to an at home PC gaming set up. Additionally, on the go gaming may be interrupted more so than arcade gaming or at home gaming such that games designed for this setting should aim to accommodate through consistent saving features and/or game play designed to be done in short bursts.

6.2.3.3 Resulting Experience

The *resulting experience* is the experience, feelings, emotions, and other end results designer's want the player to have upon stopping play of a game. While this does include the *purpose* of a serious game, it also includes the "form of fun" [2, 58] the player should experience and the emotions designers want the player to experience. These can be aligned to the purpose but are not the same as it.

For example, fear is an engaging emotion commonly used in horror games such as *Dead Space* [107] and is often considered a motivating factor in politics [108]. One could reasonably target fear as an emotion for a game's resulting experience when the purpose is to influence political change. In this hypothetical example, fear is determined as an emotional response desired in players of the game and thus it exerts influence over design decisions as the

designers pick what they expect to cause the desirable amount of fear.

Evaluating the resulting experience of players is part of the iterative design loop of prototyping, playtesting, and iterating. Designers may ask specific questions of players to judge if aspects of the game’s design are helping achieve the targeted experience goals. Additionally, academics, teachers, and/or context experts may be taking measures during play tests to see if the current game prototype iteration is having the desired purposeful outcomes for the target audience. The amount to which these experience goals and purposeful outcomes are judged to be reached then feeds back into changes made to the next iteration of the game prototype. Thus, resulting experience exerts influence over other aspects of a game’s design, particularly the in-game elements such as mechanics that are iterated on to better meet player experience goals and the purposeful goals of the serious game.

6.2.3.4 User Interface

User interface (UI) is well defined by the DPE framework as ”everything the user sees, hears, and interacts with and how that interaction happens (i.e. the control system.)” [2] The physical interactive parts of a game, including things like controllers and keyboards, as well as any interactive screens the player interacts with in game are all a part of UI [51].

How a user will interact with a game is an important consideration for design. Mobile devices may have touch screens, but no keyboards or controllers. This means a design cannot use the same control schemes on those devices as it would for a game made to be played on a console or personal computer. A virtual reality (VR) headset would allow a different set of controls that may be advantageous to an exergame, video games meant to be played for exercise [109], as they allow for more ranges of movement as interaction with a game. In this way, *technology* exerts an influence over what would be practical for the UI of a game.

Additionally, alternative controls, alternative controllers, and carefully chosen color schemes may be necessary to make a game more accessible for more people, and designing around these may be required if the target audience is significantly affected by a condition such as

color blindness. Ideally, educational games want to reach as wide an audience as possible. Therefore, taking the time to make UI that is accessible to all by following either accessible design or universal design principles [110, 111] is beneficial.

The design of interaction approaches argued for in accessible and universal design provides good examples as to how UI as a whole affects the other elements of a game's design and in-game elements. Accessible design typically aims to take an existing game and make it accessible to a specific disability or to build a game from the ground up aimed at a target audience with that disability [111], such as audio games which are games developed using only audio and targeted at visually impaired players [112, 113, 114]. In contrast, universal design aims to develop a game that is equally playable and enjoyable by everyone regardless of disability [111, 110].

For a game targeted at being accessible to visually impaired players but ignoring other disabilities or abilities such as the case for audio games, the UI will consist purely of audio to communicate information to the player. In this accessible design approach, audio becomes much more important over the art visuals as they are potentially no longer a part of the game at all. Thus the UI and target audience together influence what would be practical design for the in-game elements.

If we contrast this with a universal design approach, the universal design thinking aims to have the game equally playable by all parties [110]. This would suggest that visuals and art re-enter as equally important to audio in the same way. Audio design that follows proper ways of communicating information to blind players still needs to be present, but visuals that do the same for hearing impaired players will also need to be created at the same quality. This then requires a larger group of expert perspectives to cover issues and design methodologies that will help the game's design meet the needs of all the players in the target audience. This then of course means a larger pool of resources required as well to meet these needs.

Ultimately, UI is the "how" a player interacts with and is interacted with by the game,

and choices in how these interactions are designed has consequences on who is even able to play the game as well as the resulting experience players take away from it.

6.2.3.5 Balance

Balance is how well a game's challenges fit with the player's knowledge and skill level [51] and is often related back to Csikentmihalyi's theory of flow [2, 76, 57] where one is said to be in a *flow* state when the task at hand matches the skills of the one performing it.

DPE puts flow and its relation to balance into a useful perspective for understanding why balance is important in game design. According to DPE, a player will experience boredom if the game is too easy or frustration if the game is too hard [2]. As such, it is important for game designers to ensure through play testing that the target audience on average receives a balanced experience throughout the game from start to finish.

In relation to educational and serious games, this balance takes on another important task beyond balancing the difficulty of gameplay: balancing the understanding of the purposeful content. Educators are familiar with how some content and concepts require understanding of prior content and concepts in order for a student to understand the new content or obtain a new skill. Often, educators must provide scaffolding, interventions or assistance [115, 116, 117], in order to bring a student up to the next level of learning. Every game must do the same across its gameplay to bring its player smoothly into their next level of achievement in the game. In a serious game, the game must both balance the gameplay challenges and learning tasks to keep the player ever achieving new heights from start to finish without making something too challenging or boring that causes the player to leave.

This then relates back to the target audience. If a serious game for political change is targeted at an audience that already knows some background information on the political topic of the game, then the game may start its purposeful content at a higher level of understanding. But if the target audience includes those who are not at all familiar with the problem, the game will need to start off with introducing the issue and building understand-

ing of it from the very beginning of the game. To use an educational game example, an educational game may be targeted at teaching beginner programming concepts such as how code executes instructions line by line [118]. This would be targeted at beginners. But a game teaching higher level concepts such as computer architecture [52, 51] will not need to start off the game explaining beginner coding concepts as the target audience for the game is expected to already know prerequisite concepts. This would be similar to requiring students to have completed prerequisite classes before taking a higher level course.

Judging and iterating on the balance of either mechanical challenges of the game and/or the purposeful goals of the game is a part of the iterative development loop. Playtesting with the target audience will reveal whether the current prototype begins where it needs to for the entry skill level and the ending of the play test should provide insight as to whether the balance from start to finish was fitting to produce the desired resulting experience goals.

6.2.3.6 Gameplay/Play Dynamics

Gameplay/play dynamics are what results from the game's rules and in-game elements interactions with each other and the player's input [51]. This is observable during play and is wedged in-between the developed design of the game on one side and the resulting experience the player receives on the other in existing frameworks [1, 2].

Designing gameplay/play dynamics is a tricky endeavour as it requires the designer to consider the interaction of all the moving parts that make up their game and the target audience. However, it is worth doing during the iterative process of design and development as it can provide insights into what is contributing or detracting from a player obtaining the desired *resulting experience*. The example provided for this from the computer architecture games in McClintock [2022] shows how one instance of gameplay/play dynamics is observed:

...the art and mechanics elements interact when the player provides input to make their character jump. The character model animates in response to the input at the same time the game physically moves the character model up and

down along the jump arc. [51]

When observing the gameplay/play dynamics in a serious game's design, designers should question if specific interactions noticed are benefiting the targeted *resulting experience* and *purpose*. If they are not, perhaps they need to be redesigned or removed. An important thing to look for is *transgressive play* [48], play that is counter to the designed way to play and that prevents the player from obtaining the *resulting experience* and *purpose* aimed for. An example from this can be seen in Dickey [2011] in which two of their players took part in play completely counter to the game's core content which also meant they disengaged from the learning content and goals tied to the game. If transgressive play is happening, designers should seek to understand why so they can adjust the game accordingly to best fit the player. Doing this requires obtaining the *player's perspective*.

6.2.3.7 Player Perspective

The *player perspective* is the view of a game from the player's (target audience's) eyes. The player's perspective is a core part in the iterative design and development cycle and is obtained during playtesting done with a target audience. Because the player is the person the game is being made for, or the person the serious game is trying to affect, the *player perspective* is positioned above the other perspectives outlined in the framework.

The MDA and DPE frameworks emphasize the importance of taking into account the player's perspective in design as those frameworks argue that changes to design elements change the resulting experience for the player [1, 2]. This is true in the IGDD framework presented here. Also, whether or not the *purpose* and/or *resulting experience* has been achieved can only be known from gathering the player's perspective.

When designing a game, designers should consider an initial player's perspective based on their own design expectations from the defined target audience and intuition. Then throughout the project design and development cycle they should repeatedly obtain the true player's perspective from play testing with the target audience which should then exert

influence into changes made to the game's design.

6.2.3.8 Designer Perspective, Academic Perspective, and Content Expert Perspective

Beyond the player's perspective there are several other important perspectives that the designers must consider when creating their designs. These perspectives include: the *designer perspective*, the perspective from the designers on the team; the *academic perspective*, the perspective from any academics involved with the project such as researchers; and the *content expert perspective*, the perspective of any content experts on the team such as teachers familiar with the educational content targeted in an educational game.

The *designer perspective* is important to consider when designing games and serious games. The designer will have a unique viewpoint that will help reveal what in-game elements and designs are likely to best fit the *purpose* of the game and lead to the desired *resulting experience* for the player [51].

The *academic perspective* comes from the viewpoint of any researchers involved with the creation of a game, which is common in serious games and educational games. The academic perspective needs to be considered in the design process as it will provide influential insights into how data collection can be integrated directly into a game [76]. Additionally, considering this perspective during design and development helps ensure that the game created can answer any research questions targeted. For example, the funding to make the game might be obtained through a research grant focused on VR technology. As such, the academic perspective will be able to help guide the design to answer the relevant questions while also restricting the technology of the game to be VR.

In many entertainment games' design and development processes there may not be academic perspectives to consider, particularly for indie games with small teams. However, at large enough companies a researcher may be hired to handle deeper data analytics about the game or some part of it to inform future projects or updates to the same game project.

It is important for all team members to be aware of if academic perspectives are needed for a project as it may influence the work that needs to be done or the designs that can be reasonably created.

The *content expert perspective* is the perspective of those particularly knowledgeable on some set of relevant content for the game being created. In an educational game, this would typically be a teacher, instructor, or other expert on the learning content targeted by the game. Their perspective is useful in the design and development process as they can assess if the learning content is being obtained during play testing and may even have insights into the kinds of mechanics and game play that would fit the learning content.

Another content expert perspective of particular usefulness is any expert on the target audience for a game. This kind of expert can ensure the game being created fits well with the target audience's cultural and experiential backgrounds so that it will be well received, see *When Rivers Were Trails* as an example of this being done [23, 22, 21].

6.2.3.9 Polish

Polish is a term that comes from the game development industry [55, 56]. Polish is the last ten percent of work and the removing of imperfections from a game that could take the user out of their immersion in the experience [51, 55, 56]. Polish happens mostly near the end of a game's design and development cycle. It is often one of the last things done before release and how much polish is actually done is often determined by the remaining *resources* available to do any polishing after the core of the game has been created and is well tested.

One example of polish is the removal of difficult to find collision inconsistencies in platforming games that might get the player stuck. This would remove the player from their immersion and thus meticulously searching for and ironing out these small collision issues would be considered polish [51]. This is in contrast to game breaking collision issues which are those that prevent play often such as getting stuck on the majority of places the player can move or falling through the ground consistently. Those kinds of bugs should be resolved

long before any polishing is started.

Because polishing is focused on small improvements that make a game "shine" but are not core to realizing the *resulting experience* or *purpose* of a serious game. Polish is positioned near the bottom of the design elements hierarchy as it is beneficial to achieving the goals of a serious game but not typically as needed as the elements above it.

Some scholars may disagree with this positioning because educational games and serious games have often been compared to the triple A games released by the game development industry in a way that suggests they are in direct competition. However, educational games are often not actually competing for the *exact* same space as the industry is fighting over. The game industry, with the possible exception of Age of Learning [25], mostly fights over the free time of players which is outside of the educational application contexts that educational and serious games may find themselves in. That said, if a specific game design and development scenario *does* place a game in competition with triple A games, it would be wise to strongly consider polish as something far more important in that case.

6.2.3.10 Design Cohesion

Design cohesion is how well the design elements, including the in-game element designs, work together [51]. One example of this is how well the various perspectives are put together in the design process. According to DPE, this specific alignment is the *heart of serious game design* [2] and a strong alignment of these goals leads to a serious game that is much stronger than its individual parts. This idea is supported further by the SGDA framework which argues that tight cohesion across elements in serious game design leads to a better realization of the targeted *purpose* [3]. Another example of design cohesion would be the choice of a technology that fits the target audience, application context, and content expert perspective like the choice to use web browsers made in McClintock [2022] where the class the games were designed to be used in was already fully online, using a web browser, and the students were already familiar with online learning.

Design cohesion has been placed at the bottom of the design elements group because of its difficulty in being achieved and recognized given its greater level of abstraction. However, game designers should be looking for the ways their design elements overlap and affect each other to find ways to push the benefits the cohesion has for realizing their game’s purpose and/or resulting experience.

6.2.4 In-Game Elements

In-game elements are those that are significantly observed and experienced by the player [51]. These elements are also underneath the design elements in the IGDD framework because they still need to be designed before they can be played. However, players can more easily recognize and describe *in-game elements* than they can *design elements*. For example, a player experiences *art* much more directly than they do *balance* and can provide more immediately useful feedback on *art* than they can for *balance*, which requires a certain level of design knowledge to assess and discuss.

6.2.4.1 Mechanics

”Mechanics are what the player does when interacting with the game.” [51] They are ”the rules that define the operation of the game world, what the player can do, the challenges the player will face, and the player’s goals.” [2] Mechanics, under these definitions, is one of the most important elements and it is highly emphasized across several relevant frameworks [2, 1, 3, 77], hence why it is at the top of the in-game elements in the framework presented here.

The mechanics chosen for a game depend strongly on the target audience and the kinds of mechanics the designers expect that target audience to enjoy as well as the resulting experience desired for the player. In serious games, the *purpose* of the serious game takes on an additional influential role in determining the chosen mechanics as well. Because mechanics are the interactive part of games, and because interaction with educational content is the

central idea behind active learning [119], it makes sense that existing relevant frameworks put emphasis on making sure mechanics line up with the learning goals in educational games or the resulting experience goals of entertainment games [1, 2, 3].

6.2.4.2 Narrative

Narrative in games takes on two different types. There is the embedded narrative crafted by the designer and the emergent narrative, the unique narrative experience the player forms in their own interaction with the game [46]. The level of depth in the embedded narrative will depend on the game [2], though for serious games the general practical suggestion from existing frameworks is that the narrative should support the *purpose* of the game as much as possible [3, 2].

Narrative has been placed in the IGDD framework below mechanics in the in-game elements hierarchy, though it is worth noting that in some situations it may be just as or more important than mechanics, such as in Dickey’s work [2011] where the purpose of fostering argumentative writing skills demanded a rich narrative.

Narrative interacts with mechanics in that the mechanics programmed into the game allow the player ways to interact with the narrative in differing ways and at differing depths depending on the embedded narrative designed and the mechanical interactions allowed. Games like *Fallout: New Vegas* [120] and *Fallout 3* [121] allow the player to shape their emerging narrative through a dialogue system (mechanic) that provides options depending on the player’s stats and prior actions such as killing certain characters. *When Rivers Were Trails* provides a freely accessible serious game example of similar emergent narrative through dialogue mechanics [21, 22].

6.2.4.3 Audio

Audio is all the sounds that a player hears when playing a game, including sound effects, music, and any spoken dialogue [51]. Audio has the potential to be very important when it

comes to games in regards to helping the emotion part of the *resulting experience* be realized considering how music is able to trigger emotions [122]. However, a game can still be played without audio. This is perhaps why it is often overshadowed by mechanics and narrative in theoretical discussions around serious game design, and why it has been placed below those in the hierarchy of importance of in-game elements in this framework. Despite that, it is not unreasonable to imagine a situation where audio becomes just as or more important than mechanics or narrative in games, such as when designing games for the blind [123].

Audio has an obvious interaction with narrative through spoken dialogue, but it may also shape the emergent narrative the player has as they play by evoking certain emotions by "musical texturing" [124], the adding of depth to the mechanical interaction through music and sound [124].

Audio also interacts with mechanics and polish as mechanical interactions often receive more depth and psychological reinforcement [125] on player action by using musical or sound effect cues. This is evident in many entertainment games where audio cues reinforce basic interactions beneficial to the player's progress and to negatively reinforce interactions that are negative to the player's progression [126].

6.2.4.4 Art

Art is the graphics and other visual parts of a game [51]. Art and audio are often grouped together in design discussions, such as how they are combined into *aesthetics* in the SGDA framework [3], and similarly placed at the same level in the IGDD framework's hierarchy. However, IGDD allows the two to exist and be considered independently before considering the ways they work together. Art, like audio, can be used to trigger emotional responses and can be used for visual storytelling like the environmental storytelling in *Dead Space* [107] that helps build the suspense and fear in the game.

Art and the visual aesthetic of a game are typically kept to a singular style throughout a game in order to maintain immersion. A realistic model in a game of almost entirely

cartoonish models would break immersion in a similar way to how a mechanic working differently than it always has would break immersion, as the game world no longer adheres to its own rules.

Art interacts with mechanics and polish to sell interactions the player has, such as the facial animations of characters reacting to player decisions or the flashing white of characters taking damage [126].

Art and the aesthetic style also interacts with narrative as visual representations add to a player's immersion in their emergent narrative and aid in the player's connection to the character's and world they play in.

6.2.4.5 Game Cohesion

Game Cohesion is the in-game element specific version of the larger *design cohesion*. Game cohesion can exist within the individual elements themselves (intra-element cohesion) and between them (inter-element cohesion) [51]. For example, the art in a game can have strong intra-element cohesion if all the art matches the exact same style. Strong inter-element cohesion can be achieved when the various in-game elements align with each other, such as the art visual of a character's foot landing on wood followed immediately by the playback of a sound effect (audio) that matches what the player's eyes have just seen.

Game cohesion is at the bottom of the IGDD framework's in-game element hierarchy because the other elements need to exist in some form before they can be interwoven effectively. Game cohesion tends to come together later in development after many iterations and is typically most easily observable once polishing of the game has started.

The entertainment game *Escape from Tarkov* provides an interesting example for Game Cohesion [127]. For intra-element cohesion, *Escape from Tarkov* maintains an ultra-realistic approach to its art style to help sell the gritty resulting experience goal of the military simulation game. For inter-element cohesion, the player's can see and understand the various materials that can be stepped on in the game's maps. This means that if they see metal in

one place, and then hear the loud CLANG sound of footsteps on metal, then the player can identify that another character has likely just maneuvered through that spot. This then has a mechanical interaction in that it prompts the player to react by aiming in that direction, hiding to ambush, or fleeing the area to avoid confrontation.

6.3 How to use the IGDD Framework

Existing game design frameworks are not particularly prescriptive of how to make use of them in the design and development process. At the very least, they are not approachable enough or accessible enough for the academically researched frameworks to make an impact outside of the academic field [88]. While MDA [1] has over 3,000 citations on google scholar at the time of writing and is, to the author’s knowledge, used to teach at Ball State University’s game design and development concentration [128], it has had several years worth of time to reach this level of relevance. To compare, DPE [2] builds directly off of MDA [1] to create a richer framework and has had nearly the same amount of time in publication, yet it has only around 400 citations on google scholar at the time of writing and, to the author’s knowledge, is only used in instruction at Michigan State University’s game design and development program [2] where the author of DPE is a core instructor. These observations support O’Shea and Freeman’s [2019] point that the impact of our academic research is limited by writing without practical concerns in mind. This section addresses this by explicitly outlining how IGDD can be used in practice within the iterative design and development process shown in Figure 69.

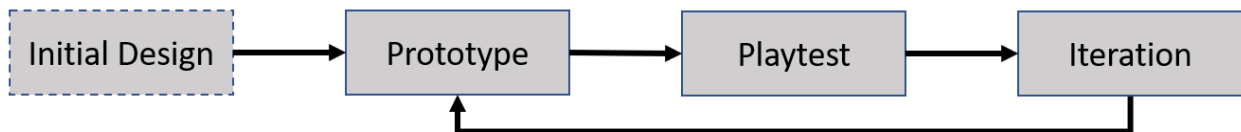


Figure 69: The Iterative Design and Development Process

6.3.1 In the Design and Development Process

The typical design and development process for games is similar in many ways to software development's now ubiquitous *agile* methods [87] in that it is flexible and reacts to the present reality of the design and development of the project.

Game design and development is an *iterative* process [2, 46, 1]. IGDD can be used as a guiding tool in this process to consider the relationships between serious elements, development elements, design elements, and in-game elements to help in the tough decisions that arise during development.

When applying the IGDD framework in the design and development processes the elements as outlined should be systematically considered throughout. Though before that, anyone using IGDD should consider what their unique design and development reality demands. Perhaps they do not have the resources to create extensive documentation or perfectly fitting audio, and that is perfectly okay. By recognizing that reality at the start, creators can remove those parts from their consideration given their own ample reasons to do so. This will let them adjust their design and development to focus on what they actually can do, or to focus on a design that fits both within the limitations and the purpose of the serious game. If creators are not creating a serious game, and want to use IGDD for a purely entertainment focused game, then the serious elements could be removed and what remains would be a useful framework for that endeavor.

6.3.1.1 Initial Design Ideation and The Start of

The Iterative Design and Development Loop

To begin work on any game project there must first be an idea to build the first prototype(s) from. The process of forming initial ideas is sometimes referred to as *ideation*, the creating and communicating of ideas [129]. The inspiration that sparks new game ideas can come from anywhere: lived experiences, movies and other media, nature, physical activity, and so on [130]. Additionally, there are many tools and processes to aid designers and developers

in this ideation process [130, 131, 132].

The IGDD framework fits into this process to help in the selection of the most fitting idea from ideation to funnel into the iterative design and development loop. Regardless of how the designers/developers went about getting their game ideas, the elements presented in IGDD can be considered in relation to these ideas to cut down or refine them until the best idea to begin prototyping is found. For serious games the obvious element to start from in evaluating ideas is *purpose*. If the idea seems unlikely to achieve the purposeful goals, then what can be changed about the idea to make it fit better? Or if all the ideas can be discussed around how well they fit the purpose, then the decision of which to start development on is easier, whichever fits purpose the best. However, there are more elements to consider when refining and selecting the final idea from ideation.

Ideally, each game idea is discussed in relation to each element in IGDD to find any glaring flaws that should be changed about the idea or that would remove it from consideration. The obvious element that will cut ideas out quickly for both serious game and entertainment game ideation is *resources*. Without the properly aligned and available resources at hand to make an idea a reality, it makes little sense to attempt creating it. Similar thinking can be applied to *asset acquisition*. If the team does not have a reliable means to get the assets needed to create an idea, then it should probably not be considered further.

Some elements may already be fixed in such a way as to aid in the cutting/refinement of ideas. For example, if the target audience is already known, some ideas may be easy to spot as particularly fitting or ill-fitted to that audience. If the target audience is already fixed to be young children, then perhaps ideas involving things typically considered inappropriate for children such as extreme violence and gore can be removed from consideration. Similarly, technology and application context can be used to cut ideas or highlight fitting ones quickly if either element is already known to be fixed.

Resulting experience, mechanics, narrative, audio, and art should be discussed by the team at this phase in relation to what ideas are the most exciting for team members to work

on. Having team members working on what they are most excited about can lead to better work by aligning the team while maintaining individual autonomy [101]. As such, discussing these elements with those who would be designing/developing them for each idea to get a sense of which ideas most excite the team overall is a useful step in making the best decision of which idea to continue into actual development.

Some elements may at first seem to be inconsequential to consider at this step for many developmental contexts. For example, many game project teams will likely use the same communication strategies for each game idea. However, some contexts may force teams to consider how they are going to manage communication for any particular idea. For example, a serious game idea whose target audience is outside the country of the producing team will require someone on the team to either travel to or otherwise set up some way for members of the target audience to communicate feedback to the team, whereas an idea that has a target audience local to the team may require less effort on communication set up. This is also an example of how interactions between elements would be useful to consider in selecting a game idea.

A team's discussion of each idea related to each element and any interactions is how IGDD aids in this first step of game design and development.

The practical approach for any team in any developmental context is to first discuss which elements are the most important in the team's specific context and rearrange the elements based on this discussion. Team's may also add to the IGDD framework any elements they agree to be important in their context and position those elements relative to the existing ones as they see fit.

With this idea of what is most important in the team's context, the next step is to begin discussing each idea related to the elements in the order of most to least important. Team's may not need to have a thorough discussion of all elements. The most efficient approach is likely to discuss each idea for the most important element and then cut down or refine ideas that the team agrees are poor fits. After discussing the ideas related to their

most important elements, the team should begin to develop an understanding of their ideas relative to their development possibilities and goals. From here, the team can continue to discuss each idea relative to each element, cutting ideas out of consideration or refining them during this process. The team may eventually move on to considering if any interactions between elements will strongly matter in selecting an idea, but a team may also stop and choose an idea at any point in this process. Additionally, team's do not need to follow this description exactly as stated.

Team's are encouraged to adapt the process suggested here to their own workflows. For example, some team's may prefer separating the team out individually to formulate ideas when brainstorming before coming back together as a team to see all ideas rather than brainstorming as a group. The same set up could be applied to elements discussion where each team member reviews the ideas relative to their own individual perspective before returning to a group discussion.

Some example questions to ask and use for discussion at this idea selection stage per element follow:

Purpose:

How well does this idea align to our purposeful goal?

What changes could be made to make it align better?

Resources:

What resources do we need in order to make this idea a reality? Do we have them?

Does the team have the skillsets the idea requires?

Asset Acquisition:

How would we go about getting the assets needed for this idea?

Prototyping:

What prototypes need to be made to best explore this idea?

Playtesting:

What will playtesting look like?

Are there any special cases for playtesting between this idea and others?

Iterating:

What parts of this idea do we expect to have to iterate on the most?

Communication:

How is communication going to be handled across the team?

Does this idea require the team to set up any special means of communication?

Extensibility:

How confident is the team that this idea will be easy to make extendable?

Documentation:

What kinds of documentation needs to be created for this idea?

Technology:

What technology can this game idea be run on?

Is the idea possible with the technological limitations enforced on the team?

Target Audience:

Who desires to play this game?

Does this game idea fit well to the audience?

Application Context:

Where will the game be played?

Does this game idea fit well to the expected application context?

Resulting Experience:

What resulting experience will this game idea produce?

Does this resulting experience align with other developmental goals, such as purpose?

User Interface:

How will players interact with this game? Does that interaction fit with the other goals of the game, such as purpose?

Does the team have a skillset that meets the idea's UI requirements?

Balance:

What is the player expected to be able to do when the game begins?

Is the target audience already familiar with this idea's gameplay?

What skills should the player leave with when completing the game?

Gameplay/Play Dynamics:

What do the in-game interactions look like for this game idea?

Will the gameplay dynamics of this idea facilitate the resulting experience and purposeful goals of the project?

Is the team interested in developing/designing the gameplay for this idea?

Player Perspective:

Who does this idea need player perspectives from? If the target audience is diverse, the player perspectives gathered may need to be from just as diverse a group.

Does the team think this idea will be welcomed and fitting to the player's in the target audience?

Designer Perspective:

Do the designer's expect that the idea will fit purposeful and resulting experience goals?

Academic Perspective:

If the game is being used in research, will this idea be able to answer the research questions?

Does this idea allow for the collection of the required data to answer any research questions?

Content Expert Perspective:

Do content experts expect the idea to work well with required content/contexts? For example, does the idea sound like it will help educate undergraduates on computer science concepts? Will it be a kind of game they will likely enjoy?

Do the content experts expect that the idea is appropriately respectful to any cultural or other backgrounds involved in the idea or targeted audience?

Polish:

What ways, if any, can the team foresee how they would polish this game?

Does the team expect that, given the scope of the idea, that there will be time to polish the game?

Design Cohesion:

Does the team think the idea clearly has ways in which its parts will work well together, particularly in relation to achieving resulting experience and purposeful goals?

Mechanics:

Do the mechanics required for the game sound like something relevant team members would be excited to work on?

Does the team have the skillsets to create these mechanics?

Are the mechanics aligned well with resulting experience and purposeful goals?

Narrative:

Does the narrative creation required for the game sound exciting to work on?

Does the team have the skillsets to create this narrative?

Is the expected narrative design aligned well with resulting experience and purposeful goals?

Will the narrative design and storytelling be appropriate and/or well received by the target audience?

Audio:

Does the audio required for the game sound exciting to work on?

Does the team have the skillsets to create this music and sound effects needed?

Art:

Does the art required for the game sound exciting to make?

Does the team have the skillsets to create the art needed?

Game Cohesion:

Do the various in-game elements in the idea seem like they will work well together in achieving the resulting experience and purposeful goals?

Does the team have the resources to maintain intra-element cohesion such as a consistent aesthetic style?

6.3.1.2 Inside the Prototype, Playtest, Iteration Loop

After an idea is selected to move to making prototypes for, IGDD can continually be applied to enhance the understanding of a game's design and developmental reality throughout the prototype, playtest, iteration loop. This is done by asking similar questions to those asked when seeking to understand initial ideas for selecting which to work on. However, it is now more important to dig deeper into any interactions so that the overall design and development process can be iterated on to steer the game in the best direction.

For example: for a serious game the team would ask "How well is our game achieving its serious purpose?" followed up with questions that get at interactions such as "Why is it achieving the purpose well? what elements contribute to achieving the purpose? which detract?" or more specific questions such as "are our mechanics not as good of a fit as we expected? what can be changed about them to help fit the purpose better? what has our target audience said about our mechanics that might point in the correct direction?" Repeatedly asking questions that dig deeper into the *why* certain decisions were made, the *why* they did or didn't work, the *what* the decisions led to, and other questions like this will lead to a better understanding of the game design, development, and the path that will take it to the best form possible. Additionally, properly documenting the answers to these questions and the results from the decisions made about them can bear fruit in future projects when similar questions arise.

These questions should be asked continually throughout the creation of prototypes, their playtesting and as iterations are made, but the most important time to have answers to them

is after playtesting has been done and before iteration is started. The reason this particular time in the cycle is important is because the answers to these questions mainly serve to inform the next iterations being made. Keeping the questions in mind throughout the loop and working to answer them at all stages means it is more likely that well reasoned answers will be available in time to inform the next iteration of the game.

6.3.2 In Assessing Existing Games

Beyond making games, the IGDD framework is usable to assess games similar to what the SGDA framework [3] is aimed at doing. To use the IGDD framework to assess a game, the definitions and questions for each element can be applied to the game by playing it to pull out the details of the elements used and examine the relationships between them, similar to how questions would be asked through the playtesting part of the development loop if you were making the game. One difference is that in assessing an existing game, the assessor likely does not have the same insight into development elements that the developers would. Thus IGDD really only works in assessing elements most notable through play such as the in-game elements and several of the design elements. However, if development documentation or assets such as the game's source code are available, then many of the remaining elements can be examined and deeper contextual understanding developed.

Regardless of if development documentation is available or not, assessing the existing game using IGDD involves playing it and observing it being played and then asking both questions about the observable elements and their interactions. For example: "What are the core mechanics of the game?", "Does the game appear to have a clear purpose/resulting experience?", "Do the mechanics, narrative, audio, art, etc. support the purpose/resulting experience? Do those elements work well together?", "Is the game balanced relative to its target audience?", and so on.

By asking questions relative to the elements as defined in IGDD, a game can be deconstructed and its design and development better understood through those elements and their

interactions as observed through play of the game by the assessor and by others the assessor observes playing, especially if any design/development documentation is available.

6.3.3 In the Design Research Process

Related to assessment, IGDD is useful when conducting design research on one's own game design and development processes as it provides easy slots to place reflections in and clear ways through which to attribute paths of influence from development into design and the finished game.

A strong cohesive design research can be achieved by asking similar questions as outlined in the Design and Development Process section as well as documenting the questions asked, their answers, and the observed changes from iterations. This then provides clear notes relative to specific elements in the design and development process that may prove easier for more readers to understand if the modular nature of referring to elements and their definitions is maintained.

6.4 Discussion

Game development's multidisciplinary nature requires many disciplines to work together to create successful games [78, 79]. If taken a step further, game development becomes interdisciplinary when the work from the disciplines blend together into something completely new and unique [79].

The IGDD framework presented in this chapter pulls together a wide array of disciplinary concerns and wraps them into a more cohesive and easily adaptable framework than those that existed before it. In doing so, IGDD better encapsulates the development side of game design and development and shows how it exerts influence over the design process while maintaining the various elements of concern raised by extant frameworks for both game/serious game design and development.

Additionally, IGDD includes practical ways it can be applied to aid the design and

development process, in assessing existing games, and in the design research process. This is done through the analysis of elements identified using questions that allow for reasoning behind decision making to be clear and documented to inform later decisions.

6.4.1 Limitations and Future Work

While the IGDD framework aims to combine the many elements identified by existing research, it likely still does not cover everything. However, an important part of this framework is that it is structured so that parts can be removed to better fit specific development contexts and, more importantly, elements can be *added* to better fit contexts. Developers and other researchers are encouraged to identify where concepts not named in the present work can be included and how they interact with other parts. In this manner we can build *upwards* a more comprehensive framework that encapsulates many related concerns rather than building *outwards* many frameworks that are assumed to be in competition with each other and cannot reach as high individually as they could when combined.

An important limitation of this and many existing frameworks is the lack of documentation of its practical use [88]. What will be important for future research on game design, development, and the application of games in research will be the proper documentation, recognition, and discussion of the frameworks used in informing those processes. Currently, this is not a standard practice and not doing so leads to difficulty in assessing and improving frameworks [88]. It is of utmost importance that any research that uses IGDD additionally include in an openly accessible publication the details of the framework's utilization, including constructive criticisms, comparisons to other frameworks, and any thing else that may help framework creators build a better framework and better practical guidance.

Additionally, it is important that any educators document the frameworks they use in instruction. While the present author is aware that MDA is used in Ball State's program [128, 1] and that DPE is used in Michigan State's [2], the author is only aware of this because of discussions with the instructors of those programs. To better document frameworks and

their use in such a way as to understand their impact, educators should put pressure on their institutions to at least list the frameworks used in their classes publicly on the institution's websites. The next best step would be to add as much detail of how the framework is being used in instruction as possible, including the instructor's own assessments of the frameworks.

6.5 Summary

The IGDD framework presented here provides an adjustable framework that combines many elements of concern in game design and development. IGDD provides definitions of these elements along with a description of how to practically apply the framework in the design and development process to assess elements when choosing a game idea and during the iterative processes. The flexible nature of the framework allows it to be adjusted through the removal and addition of elements which makes it easier to apply to varying development contexts.

One important next step is for future framework builders to build on top of IGDD with the recognition of other elements not yet addressed. This will allow framework builders to craft an ever more comprehensive framework rather than many frameworks in apparent competition.

The other important step is for those applying frameworks in research or in education to properly document their use of the frameworks and their criticisms of those frameworks, including IGDD, so that framework builders can create both better frameworks and better practical guidance for how to use them.

7 Summary and Conclusion

The field of serious games research has shown that games can provide greater engagement and learning outcomes when done well, and the field is concerned with improving the design, development, and application of serious games [41, 7, 10]. Doing so requires a better understanding of the elements that make up a serious game, the design of the game, the development of the game, and its application context. The field also has concerns with ensuring that new serious games used in research are made available so that they can be used again, either to confirm results or to adapt to new contexts [12, 13].

The work completed for this thesis addresses these concerns. A better understanding of the impact of the narrative element on the success of serious games is the primary focus of the experimental part of this work as presented in Chapter 5. To support this, the design and development theory behind the games used in the research, as well as their application context, is detailed in Chapter 3 and Chapter 4. This then led to the development of the IGDD framework in Chapter 6 aimed at providing a more powerful and practical framework to assist future work. Additionally, the games and their project files used in this thesis work are available publicly [17].

The findings from the experimental part of this thesis in Chapter 5 suggest that computer science educators using games could reasonably choose either a larger narrative game covering more course content or a set of multiple games of varied narrative, but otherwise of similar design, to cover the same course content together and expect similar results in student engagement with course content.

7.1 Contributions

The contributions of this work include: a better understanding of the impact of the narrative element in serious games, an expanded view of what elements are important in informing serious game design and development, a set of serious games and their project files for future

researchers to use and adapt, and a broader framework for game design and development with practical guidelines for its use.

All together, this work will inform future researchers' and practitioners' decision making when creating and applying serious games. It furthers the research into the importance of narrative and therefore serves to inform decisions on how resources should be spent on a serious game's narrative element. It provides a discussion of the important elements that should inform design and development decisions which researchers can reference when making their own games.

7.2 Limitations

7.2.1 Experiment

One limitation of this work lies in the implementation of the serious games. A concern from the creators that is also highlighted in student feedback is that the games have some features that are not communicated clearly or polished. This is caused in part by developmental issues and requirements on content that needed to be in at certain deadlines. This led to the pacing of the games being strong during the first set of learning content, but weaker in the later two sections of learning content. These issues exist in the same way in both the singular cohesive narrative game case as well as the multiple game case and so should not induce differences in the two cases in the study. However, it is still worth noting that it could be a cause for lowered engagement from students and is something that could be improved on.

Another limitation of this work is its limited application context. The research is done within a computer science undergraduate classroom context in the United States. While this provides insight into whether it is better to apply a singular cohesive narrative design or many narratives or perhaps either, it does so most reliable only for this application context. The result that there is no significant difference either way may not generalize to other learning content, to other classrooms, or other target audiences with their own unique backgrounds.

It may not even generalize to other computer science education contexts. The study involved mostly junior level students, meaning the demographics of the class may have been significantly different from introductory computer science courses which may affect what the players (students) in the classroom find engaging. Thus a similar study done with introductory computer science content in an introductory course could reasonably find a different result than the one found in this work.

An important limitation of the games themselves is their relatively ad hoc nature. Although the games are made in an accessible game engine and much of the source project is well organized and the code well commented, the nature of the development of the games kept it from being completely clean and more guiding documentation for its use from being made. As such, it may be difficult for future researchers to expand on despite the clear advantage over other educational games of these source files being made available in the first place and the active effort during development to keep the project clean.

One particularly important limitation on the experiment from a psychology perspective is the high complexity of the content being presented to both groups which results in issues in accurately sampling on the desired stimulus, in this case a larger narrative game approach against three smaller narrative games approach. The complexity of creating different narratives between the single and the multiple game cases required new characters, dialog, colors, and other changes that contribute to the larger difference between the two groups being studied. But any of these underlying smaller differences can be a cause of the observed outcomes. Any changes along smaller differences of a larger stimuli can cause different outcomes to be observed [133, 134]. A higher quality 3D model for the player character in the single game case compared to the games in the multiple case for example would still be a part of the larger narrative difference present but could significantly confound the differences measured.

Because of this, it is important to understand how this greatly limits the generalizability of the findings presented in the experiment down to, realistically, only the data and context of this specific experiment. Addressing this requires sampling on the stimulus. In the case

of this experiment that would mean developing more games where the narrative related differences that contribute to the larger overall narrative difference between the two groups are varied. So, a new cast of characters, aesthetic styles, dialog, etc. across multiple interventions. This then addresses the inherent complexity involved in using a game as a stimulus as recognized by the field of communications research [135, 136, 137].

7.2.2 IGDD

The IGDD framework has the obvious limitation of being new, and as of yet un-applied. The practical guidelines included with it will hopefully allow that to be changed easily and the encouragement argued for, and made an example of, in this work of citing the frameworks used in game design and development process will hopefully change the field's lack of citing its own theoretical frameworks for game design and development.

Another important limitation of the IGDD framework is that it, and much of the prior work that informed it, has a clear leaning towards aspects of games related research other than that of Games Studies. Meaning the IGDD framework leans towards the lens of those mostly interested in game design and development. While this perspective is clearly important in making games that have the most powerful impact on the things of interest to Games Studies scholars, it would still be worth having those with perspectives informed more by the cultural outcomes of games provide their perspectives. For example, a Games Studies scholar may have beneficial insights into expanding on the definitions and considerations of elements of game design and development that are related to the end results of game play, such as the *target audience*, *purpose*, and *resulting experience*. A Games Studies scholar may be able to better point to how the cultural connections between gaming being marketed towards male audiences for so long has affected aspects of the design. This insight would then be extremely useful to game designers in making games that are powerful for audience members of any gender as it would perhaps reveal gendered assumptions that the designers and developers hold without realizing them. This is of particular importance for serious

games where the target audience typically is highly diverse.

7.3 Future Directions

There are many different directions for the games and for future research.

One direction is the continued improvement of the serious games created for the research in this thesis. There are technical issues still in the games, issues with properly teaching game mechanics, a complete lack of audio feedback, and issues with flow in the later sections of the game that can be fixed to improve the experience and possibly the impact the games can have.

One direction for future research lies in more detailed examination of the elements of serious game design and development. While the mechanics element is already examined by existing literature, and the narrative element is examined in this thesis, the other elements considered important by the existing frameworks and those identified by this thesis still need to be examined in detail for their importance.

Additionally, many of the elements important to game design and development discussed in this thesis would likely benefit from more detailed concept explication done with them to better situate them and attribute clear means through which to measure and compare them across games.

The IGDD framework as described should continue to be expanded on and improved. The framework itself is argued to be flexible and should continue to be modified with new perspectives and should not be viewed as a rigid document. Future work is encouraged to be done with this framework to critique it, bend it, break it, and ideally create from it ever improving and more complete frameworks to inform game design and development. Of course, any new frameworks should continue to provide practical guidelines for their use to avoid the apt criticisms of existing frameworks lacking impact.

There are some important notes for any future directions taken in regard to serious games research.

Researchers should be prepared to carefully outline the context of their serious game's application as well as its design and development process along with any results from the game's use in research. Doing so allows games to be better understood and compared against other research and adapted for use by practitioners. Doing this will allow games research to improve at a much greater pace than it has currently.

If researchers have the resources to develop their experiments around sampling on the stimulus for the complex messaging games provide as suggested by communications research [135, 136, 137] then they should do so in addition to providing the full context of the design, development, and application of the games in research. If this is not possible due to resource constraints or for any other reason, then the bare minimum of providing those contextual details needs to be done to ensure that many similar studies together can effectively provide the effect of having done sampling on the stimulus.

Researchers should also be prepared to release their games and their project files publicly so that they can be used in future research in a variety of contexts and so that results can be verified. Failing to do this prevents games from having impact beyond their original research use.

Finally, any research involving serious games should be able to clearly point to the game design and development theory and frameworks that informed its creation so that the knowledge behind making the best games is properly distributed, credited, and continually expanded on rather than being stuck in obscurity.

REFERENCES

- [1] Robin Hunicke, Marc LeBlanc, and Robert Zubek. 2004. Mda: a formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI* number 1. Volume 4. San Jose, CA, 1722.
- [2] Brian M Winn. 2009. The design, play, and experience framework. In *Handbook of research on effective electronic gaming in education*. IGI Global, 1010–1024.
- [3] Konstantin Mitgutsch and Narda Alvarado. 2012. Purposeful by design? a serious game design assessment framework. In *Proceedings of the International Conference on the foundations of digital games*, 121–128.
- [4] André Czauderna and Emmanuel Guardiola. 2019. The gameplay loop methodology as a tool for educational game design. *Electronic Journal of e-Learning*, 17, 3, pp207–221.
- [5] Brianno D Coller and Michael J Scott. 2009. Effectiveness of using a video game to teach a course in mechanical engineering. *Computers & Education*, 53, 3, 900–912.
- [6] Michele D Dickey. 2011. Murder on grimm isle: the impact of game narrative design in an educational game-based learning environment. *British journal of educational technology*, 42, 3, 456–469.
- [7] Sara De Freitas. 2018. Are games effective learning tools? a review of educational games. *Journal of Educational Technology & Society*, 21, 2, 74–84.
- [8] Josephine M Randel, Barbara A Morris, C Douglas Wetzel, and Betty V Whitehill. 1992. The effectiveness of games for educational purposes: a review of recent research. *Simulation & gaming*, 23, 3, 261–276.
- [9] Fengfeng Ke. 2009. A qualitative meta-analysis of computer games as learning tools. *Handbook of research on effective electronic gaming in education*, 1–32.
- [10] Michael A Miljanovic and Jeremy S Bradbury. 2018. A review of serious games for programming. In *Joint international conference on serious games*. Springer, 204–216.
- [11] Marjaana Kangas, Antti Koskinen, and Leena Krokfors. 2017. A qualitative literature review of educational games in the classroom: the teacher’s pedagogical activities. *Teachers and Teaching*, 23, 4, 451–470.
- [12] Daniel Ewell Atkins, John Seely Brown, and Allen L Hammond. 2007. *A review of the open educational resources (OER) movement: Achievements, challenges, and new opportunities*. Volume 164. Creative common Mountain View.

- [13] Ismar Frango Silveira. 2016. Open educational games: challenges and perspectives. In *2016 XI Latin American Conference on Learning Objects and Technology (LACLO)*. IEEE, 1–9.
- [14] Paula Toledo Palomino, Armando M Toda, Wilk Oliveira, Alexandra I Cristea, and Seiji Isotani. 2019. Narrative for gamification in education: why should you care? In *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*. Volume 2161. IEEE, 97–99.
- [15] Emily Naul and Min Liu. 2020. Why story matters: a review of narrative in serious games. *Journal of Educational Computing Research*, 58, 3, 687–707.
- [16] Jonathan P Rowe, Lucy R Shores, Bradford W Mott, and James C Lester. 2011. Integrating learning, problem solving, and engagement in narrative-centered learning environments. *International Journal of Artificial Intelligence in Education*, 21, 1-2, 115–133.
- [17] Declan McClintock and Charles B Owen. 2023. Circubot source project. (2023). <https://drive.google.com/drive/folders/1exmTMwWE6jMgPyXPvIH006Q-ZlVmLU-x?usp=sharing>.
- [18] Clark C Abt. 1987. *Serious games*. University press of America.
- [19] Firaxis Games. 2005. *Civilization IV*. Windows, Mac OS X. 2K Games, Aspyr.
- [20] Aleksander Husøy. 2014. Using civilization iv for learning. (2014). <https://ngvcivilization.wordpress.com/>.
- [21] Elizabeth LaPensée and Indian Land Tenure Foundation, Michigan State University’s GEL Lab. 2019. *When Rivers Were Trails*. Windows, Mac. Elizabeth LaPensée and Indian Land Tenure Foundation, Michigan State University’s GEL Lab. <https://indianlandtenure.itch.io/when-rivers-were-trails>.
- [22] Elizabeth LaPensée. 2021. When rivers were trails: cultural expression in an indigenous video game. *International Journal of Heritage Studies*, 27, 3, 281–295.
- [23] Elizabeth LaPensée and Nichlas Emmons. 2019. Indigenizing education with the game when rivers were trails. *Amerikastudien*, 64, 1, 75–93.
- [24] Age of Learning. 2022. Abcmouse.com early learning academy. (2022). <https://www.abcmouse.com/abc/>.
- [25] Age of Learning. 2022. Age of learning products. (2022). <https://www.ageoflearning.com/products/>.

- [26] Marc Prensky. 2003. Digital game-based learning. *Computers in Entertainment (CIE)*, 1, 1, 21–21.
- [27] Paul Gestwicki, Makayla Hughes, and Tori Roberts. 2019. *Race to the Moon*. Board Game. Paul Gestwicki.
- [28] Paul Gestwicki, Makayla Hughes, and Tori Roberts. 2022. Race to the moon game exhibition. (2022). <https://meaningfulplay.msu.edu/program.php?presentation=14&type=game>.
- [29] EPIC, Kendall College of Art and Design, OST, Susan Bonner. 2017. *Miner Madness*. Board Game. Kendall College of Art and Design.
- [30] EPIC, Kendall College of Art and Design, OST. 2018. Miner madness, dig into code theory game exhibition. (2018). <https://meaningfulplay.msu.edu/2018/program.php?presentation=141&type=game>.
- [31] David B Nieborg and Joke Hermes. 2008. What is game studies anyway? *European Journal of Cultural Studies*, 11, 2, 131–147.
- [32] Teemu H Laine and Renny SN Lindberg. 2020. Designing engaging games for education: a systematic literature review on game motivators and design principles. *IEEE Transactions on Learning Technologies*, 13, 4, 804–821.
- [33] Yu Zhonggen. 2019. A meta-analysis of use of serious games in education over a decade. *International Journal of Computer Games Technology*, 2019.
- [34] Nedjma Koval-Saifi and Jan Plass. 2018. Antura and the letters: impact and technical evaluation.
- [35] Teresa de la Hera Conde-Pumpido. 2017. Persuasive gaming: identifying the different types of persuasion through games. *International Journal of Serious Games*, 4, 1, 31–39.
- [36] Rita Orji, Regan L Mandryk, and Julita Vassileva. 2017. Improving the efficacy of games for change using personalization models. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24, 5, 1–22.
- [37] Games for Change, Inc. 2022. Games for change. (2022). <https://www.gamesforchange.org/>.
- [38] Chris Swain. 2007. Designing games to effect social change. In *DiGRA Conference*.
- [39] Nigan Bayazit. 2004. Investigating design: a review of forty years of design research. *Design issues*, 20, 1, 16–29.

- [40] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. 2007. Research through design as a method for interaction design research in hci. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 493–502.
- [41] Mary Jo Dondlinger. 2007. Educational video game design: a review of the literature. *Journal of applied educational technology*, 4, 1, 21–31.
- [42] Gordon H Bower and Daniel G Morrow. 1990. Mental models in narrative comprehension. *Science*, 247, 4938, 44–48.
- [43] Raymond A Mar. 2004. The neuropsychology of narrative: story comprehension, story production and their interrelation. *Neuropsychologia*, 42, 10, 1414–1434.
- [44] Kristin J Alexander, Peggy J Miller, and Julie A Hengst. 2001. Young children’s emotional attachments to stories. *Social Development*, 10, 3, 374–398.
- [45] Michael Murray et al. 2015. Narrative psychology. *Qualitative psychology: A practical guide to research methods*, 85–107.
- [46] Katie Salen Tekinbas and Eric Zimmerman. 2003. *Rules of play: Game design fundamentals*. MIT press.
- [47] Chaima Jemmali, Sara Bunian, Andrea Mambretti, and Magy Seif El-Nasr. 2018. Educational game design: an empirical study of the effects of narrative. In *Proceedings of the 13th international conference on the foundations of digital games*, 1–10.
- [48] Espen Aarseth. 2014. I fought the law: transgressive play and the implied player. In *From literature to cultural literacy*. Springer, 180–188.
- [49] Richard Bartle. 1996. Hearts, clubs, diamonds, spades: players who suit muds. *Journal of MUD research*, 1, 1, 19.
- [50] Unity. 2024. Unity real-time development platform — 3d, 2d, vr & ar engine. (2024). <https://unity.com/>.
- [51] Declan McClintock. 2022. All together now, using multiple frameworks to inform serious game design and development. In *International Conference on Meaningful Play*. https://meaningfulplay.msu.edu/proceedings2022/mp2022_paper_5725.pdf.
- [52] Declan McClintock and Charles Owen. 2021. Common narrative in educational video games: a design of games to teach circuits. In *The 16th International Conference on the Foundations of Digital Games (FDG) 2021*, 1–4.
- [53] Declan Andrew McClintock and Charles B. Owen. 2024. Analyzing differences in student engagement between a single narrative game intervention and multiple narra-

- tive games intervention in an undergraduate computer organization and architecture course. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 812–818. ISBN: 9798400704239. DOI: 10.1145/3626252.3630779. <https://doi.org/10.1145/3626252.3630779>.
- [54] Charles B Owen. [n. d.] Cirsim. (). <https://github.com/charles-owen/cirsim>.
- [55] Mike Bithell. 2015. Indie polish: making the most of the last 10%. In Game Developers Conference. Mike Bithell Games. <https://gdcvault.com/play/1022080/Indie-Polish-Making-the-Most>.
- [56] Benson Russell. 2012. The last 10: going from good to awesome. In Game Developers Conference. Naughty Dog. <https://gdcvault.com/play/1015601/The-Last-10-Going-From>.
- [57] Mihaly Csikszentmihalyi and Mihaly Csikzentmihaly. 1990. *Flow: The psychology of optimal experience*. Volume 1990. Harper & Row New York.
- [58] C Heeter, K Chu, A Maniar, P Mishra, R Egidio, and B Winn. 2003. Comparing 14 forms of fun (and learning and gender issues) in commercial versus educational space exploration digital games. In *International Conference on Digital Games Research*.
- [59] Jean J Ryoo. 2019. Pedagogy that supports computer science for all. *ACM Transactions on Computing Education (TOCE)*, 19, 4, 1–23.
- [60] Unity. 2022. Unity asset store. (2022). <https://assetstore.unity.com/>.
- [61] Shutterstock. 2022. Turbosquid. (2022). <https://www.turbosquid.com/>.
- [62] Atlassian. 2022. Trello. (2022). <https://trello.com/>.
- [63] Atlassian. 2022. Jira. (2022). <https://www.atlassian.com/software/jira>.
- [64] Daniel Johnson, M John Gardner, and Ryan Perry. 2018. Validation of two game experience scales: the player experience of need satisfaction (pens) and game experience questionnaire (geq). *International Journal of Human-Computer Studies*, 118, 38–46.
- [65] Wijnand A IJsselsteijn, Yvonne AW De Kort, and Karolien Poels. 2013. The game experience questionnaire.
- [66] Alan Agresti. 2003. *Categorical data analysis*. John Wiley & Sons.
- [67] Akiva M Liberman. 2005. How much more likely? the implications of odds ratios for probabilities. *American Journal of Evaluation*, 26, 2, 253–266.

- [68] R-core. 1969. Stats package - rdocumentation. (1969). <https://www.rdocumentation.org/packages/stats/versions/3.6.2>.
- [69] Yves Rosseel. 2012. Lavaan: an r package for structural equation modeling. *Journal of statistical software*, 48, 1–36.
- [70] Joost CF De Winter and Dimitra Dodou. 2010. Five-point likert items: t test versus mann-whitney-wilcoxon. *Practical assessment, research & evaluation*, 15, 11, 1–12.
- [71] Samuel Sanford Shapiro and Martin B Wilk. 1965. An analysis of variance test for normality (complete samples). *Biometrika*, 52, 3/4, 591–611.
- [72] Christopher Ball, Kuo-Ting Huang, Shelia R Cotten, and RV Rikard. 2020. Gaming the system: the relationship between video games and the digital and stem divides. *Games and Culture*, 15, 5, 501–528.
- [73] Zhiyong Zhang, Yujiao Mai, and Miao Yang. 2021. Webpower: basic and advanced statistical power analysis. (2021). <https://cran.r-project.org/web/packages/WebPower/index.html>.
- [74] Zhiyong Zhang and Ke-Hai Yuan. 2018. *Practical statistical power analysis using Webpower and R*. Isdsa Press.
- [75] David R Krathwohl. 2002. A revision of bloom’s taxonomy: an overview. *Theory into practice*, 41, 4, 212–218.
- [76] Leonard A Annetta. 2010. The “i’s” have it: a framework for serious educational game design. *Review of general psychology*, 14, 2, 105–113.
- [77] Amri Yusoff, Richard Crowder, Lester Gilbert, and Gary Wills. 2009. A conceptual framework for serious games. In *2009 Ninth IEEE International Conference on Advanced Learning Technologies*. IEEE, 21–23.
- [78] Julie Thompson Klein. 2010. A taxonomy of interdisciplinarity. *The Oxford handbook of interdisciplinarity*, 15, 6, 15.
- [79] Allen F Repko and Rick Szostak. 2020. *Interdisciplinary research: Process and theory*. Sage Publications.
- [80] Moti Nissani. 1995. Fruits, salads, and smoothies: a working definition of interdisciplinarity. *The Journal of Educational Thought (JET)/Revue de la Pensée Éducative*, 121–128.
- [81] Alex Moseley and Nicola Whitton. 2014. *New Traditional Games for Learning*. Routledge, New York.

- [82] Izabela Hopkins and David Roberts. 2015. ‘chocolate-covered broccoli’? games and the teaching of literature. *Changing English*, 22, 2, 222–236.
- [83] Eric Barone. 2016. *Stardew Valley*. Windows, iOS, Android, Xbox One, PS4, Nintendo Switch. ConcernedApe LLC. <https://www.stardewvalley.net/>.
- [84] J Clement. 2022. *Stardew Valley lifetime unit sales 2022 — Statista*. (2022). <https://www.statista.com/statistics/1326529/stardew-valley-lifetime-unit-sales/>.
- [85] ConcernedApe (Barone, E.) 2016. *Stardew Valley OST — ConcernedApe*. (2016). <https://concernedape.bandcamp.com/album/stardew-valley-ost>.
- [86] Veronica Boix Mansilla, Elizabeth Dawes Duraisingh, Christopher R Wolfe, and Carolyn Haynes. 2009. Targeted assessment rubric: an empirically grounded rubric for interdisciplinary writing. *The Journal of Higher Education*, 80, 3, 334–353.
- [87] Martin Fowler, Jim Highsmith, et al. 2001. The agile manifesto. *Software development*, 9, 8, 28–35.
- [88] Zoë O’Shea and Jonathan Freeman. 2019. Game design frameworks: where do we start? In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 1–10.
- [89] National Science Foundation. 2023. Nsf- national science foundation. (2023). <https://www.nsf.gov/>.
- [90] John M Carroll and Mary Beth Rosson. 2003. Design rationale as theory. *HCI models, theories and frameworks: Toward a multidisciplinary science*, 431–461.
- [91] Suzi Stephenson. 2019. Uk video games industry set for year of growth. (2019). <https://tiga.org/news/uk-video-games-industry-set-for-year-of-growth>.
- [92] Johanna Weststar, Victoria O’Meara, and Marie-Josée Legault. 2018. Igda developer satisfaction survey 2017 summary report. (2018). https://igda-website.s3.us-east-2.amazonaws.com/wp-content/uploads/2019/04/11143720/IGDA_DSS.2017_SummaryReport.pdf.
- [93] Reinhard Budde, Karlheinz Kautz, Karin Kuhlenkamp, and Heinz Züllighoven. 1992. What is prototyping? *Information Technology & People*.
- [94] Andrés Lucero. 2012. Framing, aligning, paradoxing, abstracting, and directing: how design mood boards work. In *Proceedings of the designing interactive systems conference*, 438–447.

- [95] Amanda Chaffin and Tiffany Barnes. 2010. Lessons from a course on serious games research and prototyping. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, 32–39.
- [96] Sergio J Viudes-Carbonell, Francisco J Gallego-Durán, Faraón Llorens-Largo, and Rafael Molina-Carmona. 2021. Towards an iterative design for serious games. *Sustainability*, 13, 6, 3290.
- [97] Minna Pikkarainen, Jukka Haikara, Outi Salo, Pekka Abrahamsson, and Jari Still. 2008. The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13, 3, 303–337.
- [98] Thomas A Standish. 1975. Extensibility in programming language design. In *Proceedings of the May 19-22, 1975, national computer conference and exposition*, 287–290.
- [99] Robert C Martin. 2009. *Clean code: a handbook of agile software craftsmanship*. Pearson Education.
- [100] Jesse Schell. 2008. *The Art of Game Design: A book of lenses*. CRC press.
- [101] Colleen Macklin and John Sharp. 2016. *Games, Design and Play: A detailed approach to iterative game design*. Addison-Wesley Professional.
- [102] H Frank Cervone. 2011. Understanding agile project management methods using scrum. *OCLC Systems & Services: International digital library perspectives*.
- [103] Graham McAlister. 2015. Who is your game for? understanding, defining, and designing for your target players. In *Game Developers Conference*. Speaker presentation. <https://www.gdcvault.com/play/1022918/Who-is-Your-Game-for>.
- [104] Tracy Fullerton. 2018. *Game design workshop: a playcentric approach to creating innovative games Fourth edition*. CRC press.
- [105] Chadia Abras, Diane Maloney-Krichmar, Jenny Preece, et al. 2004. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, 37, 4, 445–456.
- [106] Khanh-Phuong Thai, Anastasia Lynn Betts, and Sunil Gunderia. 2022. Personalized mastery-based learning ecosystem: a new paradigm for improving outcomes and defying expectations in early childhood. In *Handbook of Research on Innovative Approaches to Early Childhood Development and School Readiness*. IGI Global, 1–30.
- [107] EA Redwood Shores. 2008. *Dead Space*. Playstation 3, Xbox 360, Microsoft Windows. Electronic Arts.

- [108] John T Jost, Chadly Stern, Nicholas O Rule, and Joanna Sterling. 2017. The politics of fear: is there an ideological asymmetry in existential motivation? *Social cognition*, 35, 4, 324.
- [109] Jeff Sinclair, Philip Hingston, and Martin Masek. 2007. Considerations for the design of exergames. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, 289–295.
- [110] Bill Fischer. 2022. A demonstration of design methodologies for creating accessible and universal games. In *International Conference on Meaningful Play*. <https://www.youtube.com/watch?v=zYoRH3qtC2k>.
- [111] Dimitris Grammenos, Anthony Savidis, and Constantine Stephanidis. 2009. Designing universally accessible games. *Computers in Entertainment (CIE)*, 7, 1, 1–29.
- [112] Johnny Friberg and Dan Gärdenfors. 2004. Audio games: new perspectives on game audio. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 148–154.
- [113] Sue Targett and Mikael Fernstrom. 2003. Audio games: fun for all? all for fun! In Georgia Institute of Technology.
- [114] Alexis Kirke. 2018. When the soundtrack is the game: from audio-games to gaming the music. *Emotion in Video Game Soundtracking*, 65–83.
- [115] Talip Gonulal and Shawn Loewen. 2018. Scaffolding technique. *The TESOL encyclopedia of English language teaching*, 1–5.
- [116] Irina Verenikina. 2003. Understanding scaffolding and the zpd in educational research.
- [117] LS Vygotsky. 1962. 1962. thought and language. cambridge, ma: mit press.
- [118] Osmo. 2020. Osmo coding starter kit. (2020). <https://www.youtube.com/watch?v=albgYhYepLQ>.
- [119] Charles C Bonwell and James A Eison. 1991. *Active learning: Creating excitement in the classroom. 1991 ASHE-ERIC higher education reports*. ERIC.
- [120] Obsidian Entertainment, Josh Sawyer, Mikey Dowling, Jason Fader, Matt Singh, Tess Treadwell, Frank Kowalkowski, Joe Sanabria, John Gonzalez, and Inon Zur. 2008. *Fallout: New Vegas*. Windows, Playstation, Xbox 360. Bethesda Softworks.
- [121] Bethesda Game Studios, Todd Howard, Ashley Cheng, Gavin Carter, Emil Pagilarulo, Guy Carver, Steve Meister, Istvan Pely, and Inon Zur. 2008. *Fallout 3*. Windows, Playstation, Xbox 360. Bethesda Softworks.

- [122] Rebecca NH de Leeuw, Sophie H Janicke-Bowles, and Qihao Ji. 2021. How music awakens the heart: an experimental study on music, emotions, and connectedness. *Mass Communication and Society*, 1–23.
- [123] Joyram Chakraborty, Suranjan Chakraborty, Josh Dehlinger, and Joseph Hritz. 2017. Designing video games for the blind: results of an empirical study. *Universal Access in the Information Society*, 16, 3, 809–818.
- [124] Tim Summers. 2016. *Understanding video game music*. Cambridge University Press.
- [125] P Ivan Pavlov. 2010. Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex. *Annals of neurosciences*, 17, 3, 136.
- [126] Mark Brown & Game Maker’s Toolkit. 2015. *Secrets of Game Feel and Juice*. (2015). https://www.youtube.com/watch?v=216_5nu4aVQ.
- [127] Battlestate Games and Nikita Buyanov. 2017. *Escape From Tarkov*. Windows. Battlestate Games.
- [128] Ball State University. 2023. *Major in Game Design & Development*. (2023). <https://www.bsu.edu/academics/collegesanddepartments/computer-science/academic-programs/majors/game-design>.
- [129] Ben Jonson. 2005. Design ideation: the conceptual sketch in the digital age. *Design studies*, 26, 6, 613–624.
- [130] Annakaisa Kultima. 2010. The organic nature of game ideation: game ideas arise from solitude and mature by bouncing. In *Proceedings of the international academic conference on the future of game design and technology*, 33–39.
- [131] Dorian Peters, Lian Loke, and Naseem Ahmadpour. 2021. Toolkits, cards and games—a review of analogue tools for collaborative ideation. *CoDesign*, 17, 4, 410–434.
- [132] Stephann Makri, Tsui-Ling Hsueh, and Sara Jones. 2019. Ideation as an intellectual information acquisition and use context: investigating game designers’ information-based ideation behavior. *Journal of the Association for Information Science and Technology*, 70, 8, 775–787.
- [133] Sarnoff A Mednick and Jonathan L Freedman. 1960. Stimulus generalization. *Psychological Bulletin*, 57, 3, 169.
- [134] Christina J Song, Jason C Vladescu, Kenneth F Reeve, Caio F Miguel, and Samantha L Breeman. 2021. The influence of correlations between noncritical features and reinforcement on stimulus generalization. *Journal of applied behavior analysis*, 54, 1, 346–366.

- [135] Sally Jackson and Scott Jacobs. 1983. Generalizing about messages: suggestions for design and analysis of experiments. *Human Communication Research*, 9, 2, 169–191.
- [136] Dale E Brashers and Sally Jackson. 1999. Changing conceptions of “message effects” a 24-year overview. *Human Communication Research*, 25, 4, 457–477.
- [137] D Michael Slater, Jochen Peter, and Patti M Valkenburg. 2015. Message variability and heterogeneity: a core challenge for communication research. *Annals of the International Communication Association*, 39, 1, 3–31.

Pre-Survey

This pre-survey will ask you some demographic questions, some game play experience questions, as well as some knowledge questions. You should only fill out this survey if you have already filled out the consent form and consent to participation.

Thank you for your participation!

* Indicates required question

1. Please Provide Your Study ID as provided on the page that directed you to this survey (This should fill automatically, if it is already filled do not change it):

Demographic Questions

2. What gender do you identify as? *

Mark only one oval.

- Male
- Female
- Prefer not to answer
- Other: _____

3. What is your age? *

Mark only one oval.

- 0-17
- 18-25
- 26-30
- 31-45
- 45+
- Prefer not to answer

4. What ethnicity would you describe yourself as? *

Mark only one oval.

- Caucasian (White)
- African-American
- Latino or Hispanic
- Asian
- Native American
- Native Hawaiian or Pacific Islander
- Prefer not to answer
- Other: _____

Game Play Experience Questions

5. How often do you play video games? *

Mark only one oval.

- Not at all
- less than an hour a day
- 1 - 2 hours a day
- 2 - 4 hours a day
- 4 - 6 hours a day
- 6 - 8 hours a day
- 8+ hours a day

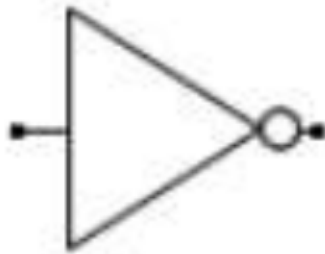
6. Would you describe yourself as one of the following? (Mark only one) *

Mark only one oval.

- Hardcore Gamer
- Casual Gamer
- Not at all a Gamer
- None of these describe me

Knowledge Questions

7. What logic gate is shown in the picture below? *



Mark only one oval.

- Xor
- And
- Or
- Not (also called Inverter)
- Input
- Output
- None of the options are the name of the gate shown
- I do not know

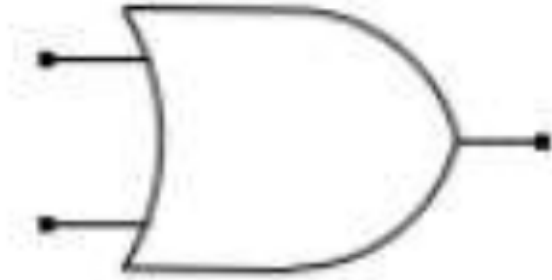
8. What logic gate is shown in the picture below? *



Mark only one oval.

- Xor
- And
- Or
- Not (also called Inverter)
- Input
- Output
- None of the options are the name of the gate shown
- I do not know

9. What logic gate is shown in the picture below? *



Mark only one oval.

- Xor
- And
- Or
- Not (also called Inverter)
- Input
- Output
- None of the options are the name of the gate shown
- I do not know

10. What logic gate is shown in the picture below? *



Mark only one oval.

- Xor
- And
- Or
- Not (also called Inverter)
- Input
- Output
- None of the options are the name of the gate shown
- I do not know

11. Do you know what a Karnaugh Map is? *

Mark only one oval.

- Yes
- No

12. How many inputs does a full adder have? *

Mark only one oval.

- 1
- 2
- 3
- 4
- More than 4
- I don't know

13. If the input to a four bit arithmetic right shift is 1010 what is the output? *

Mark only one oval.

- 0101
 - 1101
 - 1010
 - 0100
 - I don't know
-

APPENDIX B: POST-SURVEY

Post-Survey

This post-survey will ask you some knowledge questions, some engagement questions, as well as some open response feedback questions. You should only fill out this survey if you have already filled out the consent form and consent to participation.

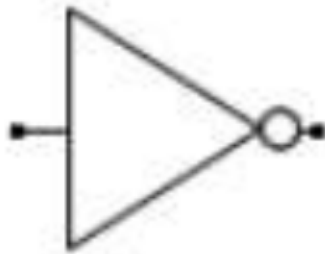
Thank you for your participation!

* Indicates required question

1. Please Provide Your Study ID as provided on the page or email that * directed you to this survey (This should fill automatically, if it is already filled do not change it):

Knowledge Questions

2. What logic gate is shown in the picture below? *



Mark only one oval.

- Xor
- And
- Or
- Not (also called Inverter)
- Input
- Output
- None of the options are the name of the gate shown
- I do not know

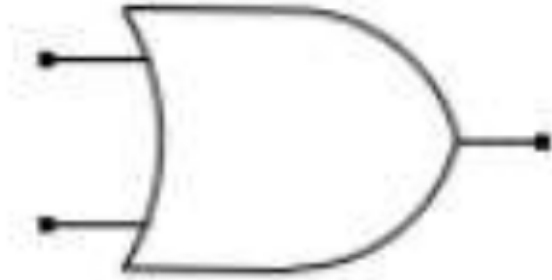
3. What logic gate is shown in the picture below? *



Mark only one oval.

- Xor
- And
- Or
- Not (also called Inverter)
- Input
- Output
- None of the options are the name of the gate shown
- I do not know

4. What logic gate is shown in the picture below? *



Mark only one oval.

- Xor
- And
- Or
- Not (also called Inverter)
- Input
- Output
- None of the options are the name of the gate shown
- I do not know

5. What logic gate is shown in the picture below? *



Mark only one oval.

- Xor
- And
- Or
- Not (also called Inverter)
- Input
- Output
- None of the options are the name of the gate shown
- I do not know

6. Do you know what a Karnaugh Map is? *

Mark only one oval.

- Yes
- No

7. How many inputs does a full adder have? *

Mark only one oval.

- 1
- 2
- 3
- 4
- More than 4
- I don't know

8. If the input to a four bit arithmetic right shift is 1010 what is the output? *

Mark only one oval.

- 0101
- 1101
- 1010
- 0100
- I don't know

Engagement Questions

For the following questions please indicate how much you agree with each of the following statements.

9. While playing the game or games I lost track of time *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree

10. While playing the game or games things seemed to happen automatically *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree

11. While playing the game or games I felt bored *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree

12. While playing the game or games I felt skillful *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree

13. While playing the game or games I felt frustrated *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree

14. While playing the game or games I felt challenged *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree

15. While playing the game or games I felt successful *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree

16. While playing the game or games I found them tiresome *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree

17. Having played the game or games I feel like I know more about circuitry *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree

18. (For player's who played the single game, Game A) I was engaged with the game's story *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree
- I did not play the single game (Game A)

19. (For player's who played the first game in the set of games, Game M) I was engaged with the game's story *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree
- I did not play the first game in the set of games (Game M)

20. (For player's who played the second game in the set of games, Game P) I was engaged with the game's story *

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree
- I did not play the second game in the set of games (Game P)

21. (For player's who played the third game in the set of games, Game *
Z) I was engaged with the game's story

Mark only one oval.

- Strongly Agree
- Agree
- Neither agree or disagree
- Disagree
- Strongly Disagree
- I did not play the third game in the set of games (Game Z)

Open Response

The questions in this section are open ended for you to provide some more detailed feedback. As much detail and feedback as you can provide is appreciated. If you prefer not to respond to any of the questions here you may leave them blank.

22. What did you enjoy most about the game or games you played?

23. What did you most dislike about the game or games you played?

24. What is your opinion of the story/narrative of each game you played?
(Please go through each game's story individually if you played more than one)

25. What would you most want to see changed about the game or games you played?

26. Any and all other feedback you have. Thank you for participating!
