This is to certify that the
thesis entitled

METHODS & CONSIDERATIONS IN DESIGNING
WEB-BASED REAL-TIME STRATEGY GAMES

presented by

JASON LENORD TYE

has been accepted towards fulfillment
of the requirements for the

Master of Arts	degree in	Telecommunication, Information Studies and Media

_____
Major Professor's Signature

_____8 - /8 - 04_____

Date

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE
JUL 3 1 2007		
APR 3 0 2008		
MAY 1 5 2008		

METHODS & CONSIDERATIONS IN DESIGNING WEB-BASED REAL-TIME STRATEGY GAMES

By

Jason Lenord Tye

A THESIS

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

MASTER OF ARTS

Department of Telecommunication, Information Studies and Media

2004

ABSTRACT

Methods & Considerations in Designing Web-based Real-Time Strategy Games

By

Jason Lenord Tye

For this masters thesis, I have developed a document which details the issues involved in developing a web-based real-time strategy game. Utilizing secondary research, I outline the technical issues and design constraints involved in the delivery of a web-based game. Through analysis and evaluation of existing games of the RTS genre, I outline the features of this game type and unique aspects of each title. Looking at the technical constraints and the game features, I then develop an outline for developing this type of game for delivery on the Internet. Using this outline, I developed a demo for this type of game. I then evaluated the design by calling for a group of volunteers to play and critique the different aspects of the demo and evaluate its design. The information is a guide for the development of a new generation of robust web-based games.

Table of contents

List of Tables

List of Figures

Chapter 1: The Genre and the Situation

Brief History of Video Games

Video games have come a long way since their inception. In 1958, a physicist named Willy Higinbotham created *Tennis For Two*, the progenitor of *Pong*, which is considered to be the first "video game". The games that were first created were very different from those we see today, consisting of little, to no, graphics and ran on any computing device the programmer could muster. Early video games were developed in a time before Nintendo's and Sony's, and desktop computers in nearly every home. A time when creating a game involved more soldering wires and vacuum tubes than programming.

The first games the general public saw were in the form of arcade machines, wood frame boxes containing a television screen and a control device, and all the wires and circuitry to run the game. The first electronic gaming system, or console, was available for the home in 1972, when Magnavox released the Odyssey. In subsequent years, other gaming consoles appeared. In 1974, Atari released a set-top version of *Pong*, and in 1977, the Atari VCS (commonly known as the 2600). Over the years, many other console systems were released, such as Coleco Vision and Nintendo Famicom, taking advantage of improvements in technology and offering improved quality of games.

Over the same period of time consoles emerged, other platforms for playing

video games emerged. In 1977, Apple Computers released the Apple II, the first

commercially successful mass-produced personal computer offering color graphics.

Also in 1977, Atari released the Atari 400 and 800 home computers. In 1981, IBM

released the IBM PC running MS-DOS. (Burnham, V. 2001)

The games found on personal computers offered the same visual quality as

those found on the consoles. But, the PC allowed for larger games with deeper stories

due to less storage restrictions then the consoles, which at the time were limited to

relatively small capacity cartridges, and PC's allowed more complex interactions

through keyboard and mouse input. But personal computers did something more than

just give the public a new venue for playing games, they also gave the ability to

create games, and spawned a new generation of amateur programmers. The PC

allowed these game creators to architect games in code, free from the soldering and

wiring of the earlier generation of developers. Many of the most well-know figures in

the video game industry emerged from this era.

Consoles and PC's have continuously 'leapfrogged' each other in

performance and abilities, but overall each are comparable in quality and ability.

Modern game consoles and Personal Computers often have the same processing

power and video processors, and both allow for play over the Internet. While you can

obtain similar input devices on both, often time what comes standard is keyboard and

mouse for the PC and control pads for consoles. The standard input device is one of

the factors that determine the types of games found on each platform. An example of a type of game affected by the input differences is that of Real-Time Strategy.

What are Real-Time Strategy Games

Typically, a traditional board games is broken up into player turns, during which time individual players take actions, such as roll die or move a piece. There are several reasons to do this. Separating the action of the game into turns gives structure to the game and makes it easy for human players to manage and keep track of game events and information. If players in a traditional board game were to all take actions at the same time, things would quickly turn chaotic. This is especially true for military simulation board games, in which the number of actions that a player makes in a turn can be many and the number of individual pieces in play can be numerous.

Real-Time Strategy Games (RTS) are the descendents of traditional board games. They evolved from traditional turn-based strategy games and take advantage of a computer's ability to manage a large number of complex rules relatively quickly. Strategic board games that involved many pieces, rules, large playing fields and rolls of the dice can be very time consuming and complicated to manage for a human, but can be handled easily by a computer. RTS games have similar rule constructs to turn-based strategy games, but with the computer doing all the number management and "dice rolling", it is possible for players to take action simultaneously, something that would lead to chaos if handled by humans.

An example of this is a 1981 release by Avalon Hill called *Axis & Allies,* in which there can be over 200 pieces in play at any given time. *Axis & Allies* comes with a set of 8 die, but an experienced player of the game knows that more are needed. Keeping track of so many pieces, rules and rolls of the dice can be a very time consuming and complicated task to manage for a human.

Strategy games are adapted very well to the computer. Many early strategy games found on the computer stuck firm to the turn based concept. Computers were very good at managing the large quantities of numbers and other types of game information. Computers were so good at managing game's rules and data that it became possible for game events to happen seemingly simultaneously. The multitude of calculations, rolls of die and applications of the game's rules could be calculated very quickly 'behind the scenes' by the computer, almost instantaneously. Instead of a game divided into distinct increments, or turns, in which the players have limitless time to make their decisions, we have a game in which players continuously make decisions for events that are taking place in real time. Time now becomes a factor. Players no longer have unlimited time to plan their moves in advance, there is no 'turn'. The player who managed their time better has an edge. This is real-time strategy. Simulating reality, players are now making decisions and taking actions simultaneously.

A Brief History of RTS Games

While there have been many games which have contained elements which are now common among RTS games, Westwood's *Dune II* is widely referred to as the first Real-Time Strategy game. Released in 1992, and based on the novel by Frank Herbert, *Dune II* was radically different from the previous *Dune* game. *Dune II's* designer believed that strategy games would be "out of control fun"(Geryk, B. 2002) if the real-time aspects of other games could be combined with the resource management of strategy games (Geryk, B. 2002) and indeed that is what was accomplished. In *Dune II* players would pick to play as one of 3 races, each of which are based on the different families from the Herbert novel, and each having different advantages and disadvantages. In the game, players would construct a base, which would allow them to deploy a harvester, a unit that would gather spice, which was the games currency. Once the player had enough spice, they could construct other buildings, which would in turn allow the construction of other units. The other types of units would be of a military type, used to attack and defeat the opponent by destroying all of their building, or to defend one's own buildings from destruction to avoid defeat. *Dune II* employed several feature concepts that have become the roots of many modern RTS game.
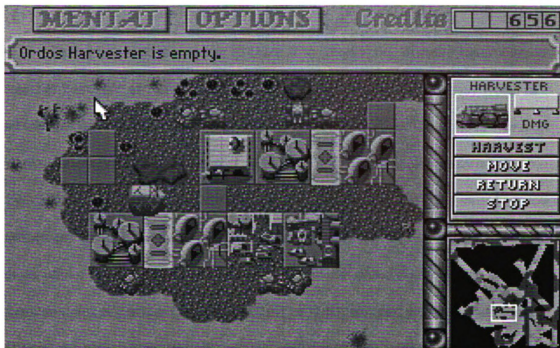
Figure 1: Troops defending the base in Westwood studio's *Dune II*

Figure 2: Base is under attack. *Dune II*.

Two years after the release of *Dune II* by Westwood, Blizzard Entertainment released *Warcraft: Orcs vs. Humans*. *Warcraft* put the player in a fantasy world in which the armies of humanity battle against the orcish hordes. The units in *Warcraft* were different from those in *Dune II*. While the units in *Dune II* were futuristic, *Warcraft's* units were true to the fantasy setting of the game, utilizing swords, arrows and magic. Though *Warcraft* incorporated only two races with essentially the same units, unlike *Dune II, Warcraft* utilized melee combat units. *Warcraft* also hosted a more complex economic system, consisting of two resources: wood and gold, and allowed the player to improve the combat abilities of their units through the research of technology.
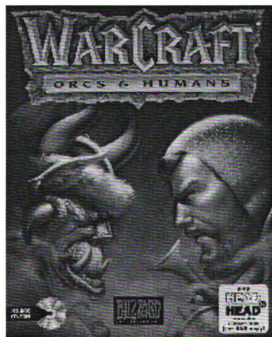
Figure 3: Blizzard entertainment's: Warcarft: Orcs & Humans

Figure 4: Footmen surround the town in Bizzard's *Warcraft*

Figure 5: *Warcraft*. A human town under siege.

A year after Blizzard released *Warcraft,* Westwood Studios released another RTS game called: *Command & Conquer.* *Command & Conquer* was *a* direct descendent of *Dune II,* and incorporated many features which the developers wanted to have in *Dune II.* Unlike *Warcarft, Command & Conquer* consisted of distinct sides, each with distinct units with varying abilities (Geryk, B. 2002). Like *Dune II, Command & Conquer* was set in the future and consisted of high-tech units engaged in futuristic battle. In the game the, forces of good, the *Global Defense Initiative,* battle the forces of the evil *Brotherhood of Nod.*

The unique characteristics of each side added to the appeal of *Command & Conquer.* Instead of a game consisting of two sides, each of relatively equal ability; the *Global Defense Initiative* had stronger, more conventional units while the *Brotherhood of Nod* had lighter and faster units.
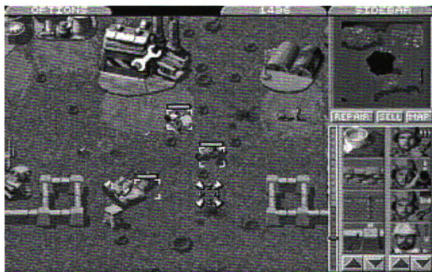
Figure 6: *Global Defense Initiative base. Command & Conquer.*

Figure 7: *Brotherhood of Nod base. Command & Conquer.*

The same year Westwood released *Command & Conquer,* Blizzard Entertainment released the sequel to *Warcraft* entitled: *Warcarft II: Tides of Darkness. Warcraft II* was a step forward in many respects; it was the first RTS to incorporate Super VGA graphics, much improving the visual quality of the game and hosted improved artificial intelligence in the units, adding to the game's challenge. It also introduced new aspects into the game, which are still utilized in RTS games today.

Figure 8: Blizzard Entertainment's *Warcraft II*

After 1995, a number of notable titles were released. In 1996, Westwood released *Command & Conquer: Red Alert* featuring new units, new strategies, and much-improved graphics over its predecessor. In 1997, Developer Cavedog released *Total Annihilation*, a game in which two factions of machines battle over control of earth. *Total Annihilation* was revolutionary in that it was the first RTS game to incorporate 3D graphics for the terrain and units which. Up until then unit and terrain been two dimensional graphics. Total Annihilation also allowed the player to issue multiple commands to a unit to be completed in sequence. Also released in 1997,

11

Dark Reign, by Auran, which also incorporated 3D graphics and multiple unit commands.
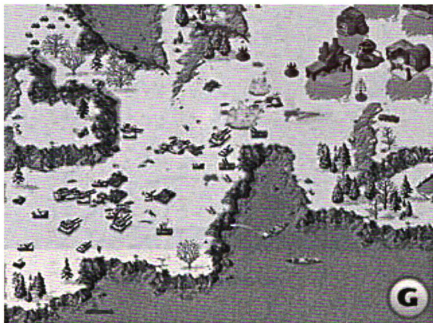
Figure 9: *Command & Conquer II: Red Alert*

Figure 10: *Total Annihilation*

In 1998, Blizzard Entertainment released *Starcraft*, a game that forever changed the landscape of Real-Time Strategy. *Starcraft's* interface was very reminiscent of that of Blizzard's previous two releases, but in many was *Starcarft* differed from its predecessors. In *Warcraft* and *Warcraft II,* the two opposing sides had relatively the same units, buildings, and paths of development, differing mostly in appearance. *Starcraft* incorporated three distinct but balanced races, similar to *Command & Conquer*, each with completely different units. But, unlike *Command & Conquer,* each race in *Starcraft* developed in a completely different way and each having its own strengths and weaknesses. *Starcraft* also hosted greatly improved graphics, improved network play, and more than any other RTS game, an improved story development. Until *Starcraft*, the story for most RTS games was told through a

series of cut scenes shown between levels of game play. The cut scene would often give some explanation of the next level, but once in the level, game play would proceed as usual and the cut scene would seem "somewhat divorced from the goings-on in the game"(Geryk, B. 2002). In *Starcraft*, the story was interwoven into every aspect of the game. In the game, units are portrayed as characters; the mission objectives may change in mid-scenario to unfold more story; and each level would reveal more of the overall plot (gamespot.com).

Blizzard Entertainment hosts a service called *Battlenet,* which tracks player statistics and allows players to connect and compete against others players around the world. *Starcraft* is one of the most successful RTS games of all time and has received numerous awards including number-one selling PC Game of 1998 (PC Data off Blizzard), 'Greatest Game of All Time' (Geryk, B. 2002 and Blizzard Entertainment), 'Best Online Game' (Reader's choice: Duelist Magazine and Blizzard) along with numerous 'Game of the Year' awards.

Starcraft also hosted many other greatly improved features such as greatly improved graphics quality and animations, improved unit artificial intelligence and improved multiplayer modes. *Starcraft* also achieved unprecedented domestic and international sales. In South Korea, it sold over 1 million copies in 1999 (South Korea's population is roughly 40 million)(Geryk, B. 2002) and South Korea also hosts whole television stations dedicated to the airing of *Starcraft* competitions. *Starcraft* is still actively played today.

Figure 11: *Blizzard Entertainments, Starcraft.*

Figure 12: Human forces fend off a Zerg attack in *Starcraft*.

Figure 13: Human forces battle the Protoss in *Starcraft*

The Structure of Real-Time Strategy Games

In their book *on Game Design,* Andrew Rollings and Ernest Adams talk about the three common elements of strategy games as the theme, the presentation, and the perspective (Rollings, A. and Adams, E. 2003).

The Theme

Andrew Rollings and Ernest Adams write about how the theme of a strategy game is often derived from the primary activity that takes place in the game

(Rollings, A. and Adams, E. 2003). While there are many subtle differences among RTS games, most of them fall into the distinct categories of games of conquest, exploration, or trade, although, in most cases they will include a small amount of all three to some degree (Rollings, A. and Adams, E. 2003).

The majority of RTS games can be categorized as games of conquest, indeed all of the early RTS game such as *Dune II, Warcraft I & II, Command & Conquer I & II, Total Annihilation, Dark Reign and Starcraft* were games based on military conquest. Combat centered RTS games are the most reminiscent of traditional turn based strategy games that dealt mostly with players trying to defeat one another's armies. The focus of a conquest RTS game is to generate an offensive force and defeat your opponent, often by destroying all of his or her means of production. The other aspects of exploration and trade are centered on this goal. Exploration is often conducted in order to gain information about the surroundings in order to better attack or defend against the opponent. Trade is often internalized and is often the conversion of resources into units, technology or other resources.

The primary goal of an exploration focused RTS is often to explore an unknown world, often while expanding a civilization. These games often include aspects of conquest and trade, but both are dependent on exploration, "You can't trade or conquer before you've explored who to conquer or trade with" Rollings, A. and Adams, E 2003). In these games, the player is usually the manager of a civilization, which the beginning is small and technologically undeveloped. The

player will then direct the civilization in the gathering of resourced and technological development. Often, the player will do this by exploring the civilizations physical surroundings for different resources and also by exploring different technologies within the game, often to make previous task easier or faster. Elements of conquest or trade often arise as the player's civilization comes into contact with another civilization.

In trade-oriented games, the player's focus is often on expanding the economy of an entity such as a civilization or business in order to increase its value Rollings, A. and Adams, E. 2003). Often conquest and exploration are skewed from their traditional forms, conquest coming in the form of the take over of an opponents assets and exploration taking the form of looking for opportunities for economic growth.

The Presentation

The presentation of a real-time strategy game encompasses both the visual aspects of the game, such as the setting: the look of the environment and the units. Also the interface: the display of buttons and information for the user to respond to and to interact with.

The Setting

The setting for a game is a vital characteristic. Two games with the same underlying mechanics can feel completely different based on their different settings.

For example, although *Starcraft* is technologically more advance than *Warcraft II*, both games have very similar game mechanics, in that the user uses a gathering unit to collect a resource and then spend that resource to construct a building from which they will produce an offensive unit. Both games have the same basic premise, but when described in context of the setting, feel is very different. In *Warcraft II* the player has their peasants collect gold and harvest wood, and then he directs another peasant to construct barracks, in which he later trains a footman. In *Starcraft*, the player orders their space-construction vehicle (*SCV*) to harvest minerals and to collect refined vespian gas, which the player then spends when they order another *SCV* to construct a barracks, where they later order the training of a space marine. In both of these scenarios the same basic actions happen. The three most common settings for RTS games are a science fiction future, a mythical fantasy world or some historical past. A science fiction based setting allows the developer a lot of freedom to generate new and interesting locations and technology, without having to base things on realistic facts. Historically based games give the developer a wealth of historical information from which to draw from (Rollings, A. and Adams, E 2003). The setting may be influenced, or in turn influence, the story the designer tells.

Interaction

Most RTS games tend to deal with groups of units vs. individual. The scale of the game often determines the type of interactions the player will. The player will feel a more intimate connection with a small squad vs. a large army. The player will have an easier time directing the actions of individual units in a small squad vs. a large

army. Where as a player is more likely to direct large numbers of units to take action rather than direct individual units. Another interaction model is further abstracted, where the player does not interact with any units but rather, manipulates data and statistics directly and indirectly (Rollings, A. and Adams, E. 2003).

Perspective

Real-time strategy games are almost exclusively presented in a top-down or isometric view, as if the player were a coach directing their players from a helicopter hovering above and off to the side of the field. RTS game usually involve the player directing a number of units over a large area, and one of the most important aspects of RTS games "is for the player to be able to get the big picture" (Rollings, A. and Adams, E. 2003). Trying to direct a large group of units from a grounded, personal perspective may be closer to reality. But, it would be frustrating and difficult for the average player. The top down perspective is very reminiscent of the way "…real life generals stand in little tents pushing toy soldiers around on 2D maps" (Rollings, A. and Adams, E. 2003).

Interface

The interface for a RTS game is how the information from the game is presented to the player in a way they can understand and react to. Often times in RTS games, the user has to keep track of multiple types of data at the same time. The most common way of breaking up the game information is the use of windows or portals

(Rollings, A. and Adams, E. 2003). Each window keeps track of different information and allows the user to take different action. Most RTS games have three basic windows to display the game general information.

The first is the game window, where the units and buildings of the game are displayed on the terrain and allows the user to view the events of the game as well as select units and items and take action in the game. It is also common for the users resources to be displayed along the border of this window. The second window, the status window, allows the user to view addition information and see additional actions that are associated with the selected unit, building, or item. Often in RTS games when the user has an offensive unit selected, information regarding the unit's health will be displayed in this window, along with actions that unit can take, such as attack or stop. The third common window is the 'mini-map', a small representation of the entire playing field or map. Usually, the mini-map is a direct top-down view reminiscent of traditional two-dimensional maps. It often displays unit and building positions in an abstract form, commonly dots and squares, which allows the user to keep track of their units and them a idea of the relative positions of different locations.

Economics

The economics are the model of commodity acquisition and exchange in the game. The currency in *Dune II* was spice, which was gathered and used to purchase

new buildings and new units; this conversion of resources is commonly referred to as the game economy. As RTS games have evolved, the economies found within the games have as well, often employing multiple types of resources to be gathered. In modern games a player may be required to gather X amounts of resource A, B, and sometimes C, or convert A and B into C and D before they are able to produce a certain unit. In general, the more powerful a unit, the more resources it will take to create. (Rollings, A. and Adams, E. 2003).

Another common aspect of RTS economies is what is called the "tech-tree". The tech-tree is a tool developers use to control the pace of the game by "limiting and rationalizing the spread of technology" (Rollings, A. and Adams, E. 2003). The tech-tree keeps players from obtaining more powerful or valuable units or technologies without paying some due price, for example:

In the RTS game *Starcraft*, one of the most powerful units in the game is the *Battle cruiser*, which costs 400 units of minerals and 300 units of gas to produce. But, in order for a player to build a *Battle cruiser* they must first build a *Starport*. A *Starport* costs 200 units of minerals and 100 units of gas. But before they can build a *Starport* they must build an *Engineering bay*, 125 mineral units, and a *Factory*, 200 mineral units and 100 gas units and both of these structures require a *Barracks*, 150 mineral units and a *SCV* to build it (50 units).

In this scenario, the player will have to take the time to collect 1120 units of minerals and 500 units of gas, plus take the time to construct all the supporting buildings

before they can obtain the vaunted *Battle cruiser*. This technique allows the designer

to pace the game effectively, so that the progression from game beginning to game

end unfolds in a somewhat realistic fashion (Rollings, A. and Adams, E. 2003).
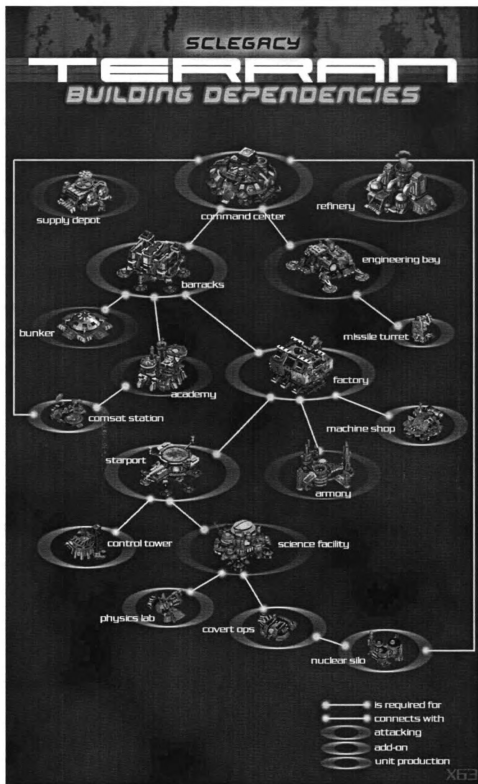
Figure 14: Terran Tech-tree in *Starcraft*.

Fog of War

In most traditional strategy games, the number and type of units an opponent had was open to the other player, because in most cases the game is played on a single board which both players can openly see. When we move to the computer, many aspects of the game can be kept hidden from the player. One of the most lasting features introduced in *Warcraft II* was fog of war. Fog of war added an aspect of exploration to the game by limiting what a player could see in the game (Rollings, A. and Adams, E. 2003). In the beginning of a game, a player can only see the area of the game map, which is occupied by their units; the rest of the map is hidden (black). As the player moves units into the hidden areas of the map, those areas become revealed. If the player then moves units out of the area is becomes 'foggy' or overcast. They can see the layout of the land they have explored but they can no longer actively see what goes on in the area. The fog of war in *Warcraft II* was also persistent, changing within the sight range of the players unit. Exploration was the only way to determine the actions of the opponent.

There have been a variety of RTS games since *Dune II, Warcraft, and Starcraft* games, each giving their own unique twist to the genre. RTS games are almost exclusively found on the desktop platform. This is mostly due to the interface required for this type of game. RTS games are designed around the robust input flexibility of a keyboard and mouse. RTS games often require the user to select units using a pointer; an action generally accomplished faster with a mouse then a game

pad. But there is an area of PC gaming in which RTS games are under represented, the area of web-based games.

Web-based games

Web-based games are a special class of personal computer games. With a traditional computer game, a user would obtain a physical copy of the game software, on a CD or DVD, and then install or copy the game application to the hard drive of their computer. After installing the software, the user would then be able to play the application from their computer's local hard drive. In some cases, the essential components to run the game would be installed on the computer and during game play the application would access other data from the external disk as needed.

Web-based games can be viewed as a separate class from standard PC games. A major distinction is that web-based games utilize the Internet for game delivery rather than a hard drive. Typically the user will navigate to a website containing a link to the game application. After clicking on the link the game application is loaded into the web browser's cache, a temporary place in the computer's memory or hard drive, from which the program is run. Like games installed in the traditional manner, often it is the essential parts of the program that are downloaded and run, while non-essential data is read in later. But, unlike traditional games this data is read from a server rather than an external disk. An application's placement in the browser's cache is not permanent. The cache is periodically cleared and in order to play a web-based game again, the user must reconnect to the server and access the game application again.

There are a number advantages and disadvantages to this that I will discuss later in

Chapter 3: *Issues of Development and Delivery*.

Over the past decade, due to developments in software and technology, such

as high-speed Internet access in homes and software development environments such

as Java, DHTML, Flash and Shockwave, the Internet has turned into a new medium

for the development of games. Web-based games utilize the same keyboard and

mouse interface that PC desktop games use, but have additional technological and

design constraints, which set them apart from standard PC games.

Current position

RTS games are one of the most popular genres of PC games. Many of the top-

selling and top-rated PC games have come from the RTS genre, *Starcraft* for

example. But, the position of RTS games in the desktop gaming market and the web-

based gaming market are very different. While RTS games make up a substantial

portion of the available desktop game they are all but nonexistent on the web. To

better illustrate this, I conducted a survey. On April 6, 2004, I visited three local

venues for purchasing desktop games in Okemos, Michigan, *Best Buy*, *GameStop*,

and *EB Games*. I counted the number of unique game titles available for the desktop

that classified, on the packaging, that they were RTS games. On the same, day I

visited three websites for online gaming and counted the number of games that could

be classified as real-time strategy. The results were as follows:

Name of venue	Number of unique game titles	Number classified as RTS	Percentage
Best Buy	310	29	9.4%
GameStop	536	56	10.4%
EB Games	283	26	9.2%

Table 1: RTS games available in retail outlets

Name of site	Number of games	Number classified as RTS	Percentage
Yahoo games at: Games.yahoo.com	84	0	0%
Electronic Arts Pogo games at: Pogo.com	58	0	0%
Shockwave games at: Shockwave.com	201	1	.5%

Table 2: RTS games available online

Tables 1 and 2 show that while RTS games made 9 to 10 percent of the desktop games available in the retail stores, they are almost unrepresented in the web-based venues.

In the following chapters, I will detail the issues involved in developing a web based real-time strategy game. Utilizing secondary research, I will outline the

28

technical issues and design constraints involved in the delivery. Through analysis and evaluation of existing games of the RTS genre, I will outline the features of this game type and unique aspects of each title. Looking at the technical constraints and the game features, I will then develop an outline for developing a RTS game for delivery on the Internet. I will also create a demo for this type of game. I will include an evaluation the design by a group of volunteers who have played and critiqued the different aspects of the game.

Chapter 2: Looking at Web-based games.

What is out there

Current web-based games are very simple, in that most are designed so that they are challenging while at the same time easy to learn. Often web-based games have few rules for the user to learn before they are ready to play.

For example: I examined games available on *Yahoo Games* website. There are six categories of games available on the site, and I looked at the first two games of each category. Of the 12 games, only one had more then a single screen of instructions.

The relative simplicity of web-based games makes them very easy to pick up, enjoy and walk away from without having to commit a lot of time into learning and playing the game. None of the online games that I looked at had any form of persistent story. Most of the games consisted of a single playing area. In some cases, the games were made up of multiple levels, very short, and often consisting of the same playing area with greater difficulty. This lack of deep plot means the user has no sense of loss from walking away from the experience.

The simplicity of web-based games is not exclusively found in nature of the game itself, many web-based games consist of very low quality graphics and little to no animation. This is partly due to the technical limitations of the delivery medium.

Development and Delivery

Development of a web-based game is very similar to that of a traditionally installed game. In general, both can have development teams consisting of multiple people, although it is possible for a single individual to develop a web-based game, while traditional commercial desktop games tend to be developed by very large teams. Both types of games start off with an initial idea, which is then fleshed out. Development of both types of games can be broken in to two parts, development of the games mechanics and the games artwork. Where the two game types start to differ, in the technical limitations and delivery.

Due to their dependence on the Internet, web-based games share many of the same technical issues and dependencies of other online services.

Web-based games require a web server on which the game file is stored and accessed. A Network Service Provider (NSP) would connect this server to the Internet. In some cases, an NSP will provide package services. "In the case of gaming, this organization might supply caching, security, authorization, copy protection, etc. host a gaming service or resell a gaming service provided by some

31

other organization." (Bahlmann, B. 2002). The fees of different NSPs can vary, but are often affected by the number on users or connections accessing the site at given time. In general, the higher number of connections, the higher the fee. If you are going to charge users to play the game, it is often necessary to enlist another service provider to process fees and credit transactions.

Download speed is one of the greatest limitations of a web-based game. This is effected several different factors such as throughput of the server, the server's connection speed, speed of the Internet and the users connection speed (Nielson, J. 2004). The throughput of the server is the amount of upstream and down stream activity on the server at a given time. As a site becomes popular, server activity can dramatically increase with user demand. The connection speed of the server can affect how difficult it will be for the user to connect to the server and for the server to respond. Like server throughput, as activity on the site increases, the connection speed of the server can become a greater factor as the connection is saturated. The speed the Internet can fluctuate. During peak hours, data transmission across the Internet can significantly decrease. Of all the factors, the connection speed of the user can be the most significant, as it is usually the slowest part in the chain. All told, bandwidth is important because it affects the amount of data that can be delivered to the user in a reasonable amount of time. Since users have come to expect relatively short download times, web-based games are traditionally very simple. As of January 2004, broadband penetration (cable modem or DSL) was reported at 38% (Nielson, J. 2004). In general, with an average broadband connection of 600KB/sec, we could

expect to deliver about 4.5 megabytes of data in less then a minute (broadband

reports.com). With such limitations on file size, the quality and amount of features

such as sound, graphics and animations are greatly decreased in web-based games as

compared to desktop games.

Other key factors that can affect the design and development of web-based

games are characteristics of the target user, such as browser, operating system, and

computer power. These factors can greatly affect, or sometimes hinder proper

playback of a web-based application.

The Audience

Time Commitment

Another way in which web-based games differ from desktop games is their

audience. Most web-based games are designed to be short and not require a large time

commitment from the user. Looking at this, we can say that players of web-based

games do not expect to invest huge amounts of time playing the game, and may be

looking for an engaging activity to pass the time.

Gender

The breakdown of male and female players is very different for desktop and

web-based games. The desktop game market is predominantly male-oriented, and this

is reflected in many of the types of games, which are often conflict or destruction-

based. The area of web-based games is very different. According to Nielsen/NetRatings, 41% of people who frequent web-based game sites are women (Fattah, H. and Paul, P. 2002), with women over 40 as the most active players of online games averaging 9.1 hours a week (Morris, C. 2004). This gender shift is reflected in the types of web-based games, which are often puzzle, cards, or problem solving type games. Stereotypically, These are often seen as the types of games women enjoy.

A recent study at Michigan State University looked at issues of gender in game design. The study involved separate groups of boys and girls from the 5th and 8th grade designing a space game for children of the same age. The preliminary results of the study found that the games designed by the female groups did not involve the killing or beating an evil entity, but rather the solving of a problem. The study also showed that while the games designed by both the male and female groups involved elements of danger, the threatening circumstances in the games from the female groups tended to be more light-hearted and silly (Heeter, C. et al. 2004). If we apply these findings to a real-time strategy game, we may be able to develop a game that women find appealing by simply changing the nature of some of the games elements.

Chapter 3: A Look at RTS Games

In this chapter, I will look at the design and game features of three popular RTS games. I will look at aspects such as the theme, the presentation, and the perspective of each game and identify the interesting features of each game. I will also look at the way in which the user is educated in playing the game.

Starcraft
Blizzard Entertainment

1998

"In the distant future a small group of human exiles have been doomed to fight for survival on the edge of the galaxy. Through military strength, espionage and deceit, a unified Terran government has maintained an uneasy peace. As resources run short, however, these Confederate nations find themselves looking towards the rich worlds of their alien neighbors, the enigmatic Protoss. To further complicate matters, it seems that a previously unknown and deadly species known only as the Zerg has entered Protoss space and is destroying everything in its path. The time for war has come..."(Blizzard Entertainment).

Starcraft is set in the distant future. The player takes the role of the military leading of one of the three races. *Starcraft* is a game of conquest. The player goes

through various missions with each race; with story progression obtain only after

military victory. The player will direct their race as they gather resources and

construct armies of troops and ships as they battle on land, air and space.

Starcraft's story is told over the course of 30 missions, cut across three

chapters, with ten missions played with each race. Over the course of each mission

with a race, the player is slowly introduced to how that race works and how to utilize

its different units. By the end of the 9th mission, the player has been introduced to all

the units and technology of the race. The length of each mission can vary, but on

average each will take about 30 to 45 minutes. *Starcraft* also comes with a 20 page

manual, which outlines how to play, the different features of the game, and the use

and development system for each race.

Starcraft incorporates many common features found in RTS games. The game

contains three distinct races, each with a unique pattern of development. The interface

of the game is divided into four areas, a game screen, mini-map and

feedback/information window along with a command window, which provides users

with the list of commands available for the selected unit or building. *Starcraft* also

incorporates fog of war.

Starcraft's internal economics are based on two primary resources, vespian

gas and minerals, and a third resource food (referred to differently for each race)

derived from the processing of minerals into other items. Each race in *Starcraft* has a

tech-tree consisting of nine basic buildings, six advanced buildings and fifteen units. For each race, different improvements in technology are purchased with gas and minerals at various buildings they construct. The tech-tree also employs building dependencies, with advance buildings and units only becoming available after the constructions of specific basic buildings.

The game is presented in a top-down, fixed isometric view. The player can move to different areas of the game field by using the cursor keys on the keyboard or by moving the mouse to the edge of the screen in the direction that they would like to move the view. *Starcraft* also utilizes 'hotkeys'. Hotkeys are keyboard shortcuts, which are tied to specific commands for a unit and allow the user to issue orders to a unit more quickly.

The visual design of *Starcraft* is very futuristic. Interface elements are 2D generated graphics and changes to depict the style of each race, for example: the interface for the industrial Terrans is very metallic with lots of wire, while the interface for the organic Zerg has a fleshy, almost living feel to it. The majority of the in game elements are pre-rendered 3D graphics, with buildings designed in 3D then rendered into 2D images along with units animated in 3D and rendered as a series of 2D images.

Installed *Starcraft* requires 121 Megabytes of hard drive space, with full motion video and other game information read from the CD ROM.

Age of Empires

Microsoft

1997

"...an epic real-time strategy game spanning 10,000 years, in which players are the guiding spirit in the evolution of small stone age tribes. Starting with minimal resources, players are challenged to build their tribes into great civilizations." (Microsoft).

Age of Empires is set in the historic past. In the game, the player directs their tribe as they evolve through the ages. The player plays as different groups thoughout ancient history, with the story for each period portrayed through various in game battles. The game takes place in a historical context; Players research technologies such as the wheel, horseback riding and stone working and play with civilizations like the Greeks, Romans and Persians. Though primarily a game of conquest, *Age of Empires* does incorporate non-combative way to achieve victory, such as: victory through superior economic production and victory through the construction of an epic monument of a civilization.

The game comes with a 120 page manual long with a short, multi-part, in-game tutorial which explains the basics of how to play the game.

Similar to *Starcraft*, *Age of Empires* incorporates many common features found in RTS games. *Age of Empires* contains twelve different civilizations to play

38

as, each with unique strengths and weaknesses. The interface of the game is divided into the traditional four areas, a game screen, mini-map and feedback/information window along with a command window, which provides users with the list of commands available for the selected unit or building. *Age of Empires* does not incorporate true fog of war, but rather, after a player explores an area it remains visible to them for the remainder of the game.

Age of Empires internal economy is relatively large. The game's internal economy is based around four resources, wood, food, gold and stone with multiple sources for each. The game also included a tech-tree with one hundred nodes in addition to technology restrictions based on the age of development the player's civilizations has achieved. Each civilization available for play developed similarly and contained approximately 23 different units.

Similar to other RTS games, *Age of Empires* is presented in a top-down, fixed isometric view. The player can move to different areas of the game field by using the cursor keys on the keyboard or by moving the mouse to the edge of the screen in the direction that they would like to move the view. The game also utilizes hotkeys for giving faster commands to different units.

Overall design of *Age of Empires* is very ancient, appearing as if carved out of stone. Interface elements are 2D generated graphics and changes to depict carvings reminiscent of the civilization the player is leading. The majority of the in-game

elements are pre-rendered 3D graphics, with buildings designed in 3D then rendered into 2D images along with units animated in 3D and rendered as a series of 2D images.

Installed, *Age of Empire* requires 80 Megabytes of hard drive space, with motion video and other game information read from the CD ROM.

Command & Conquer: Generals
Electronic Arts

2003

Command & Conquer: Generals is a game of modern military warfare in which the three factions, the United State, China and the Global Liberation Army battle for supremacy. The story is told over three chapters, each with ten missions and centered on one of the three factions. Battle takes place in crowded cities filled with skyscrapers or rural countryside with rolling hills. Each race has unique strengths and weaknesses, as well as unique systems of development.

Command & Conquer: Generals incorporates many common features found in RTS games. Similar to Starcraft, *Command & Conquer: Generals'* three factions each have their own systems of development and each has unique strengths and weaknesses. The interface of the game is divided into the traditional four areas, a

game screen, mini-map, feedback/information window and command window and incorporates true fog of war.

The internal economy of the game is relatively simple. The main resource is supplies, which are collected and tracked as money, along with a secondary resource, power, with is generated by building power plants. Each army has a unique tech-tree, consisting of approximately twelve different units and eleven different buildings.

Unlike the previous games, *Command & Conquer: Generals* is a fully 3D game in which the terrain, buildings and units are 3D models, rendered and running in real-time. Presenting the game in full 3D allows the players to change the view from the traditional top-down view and rotate, zoom in and zoom out of the game screen. Just like the 2D games, the player can move to different areas of the game field by using the cursor keys on the keyboard or by moving the mouse to the edge of the screen in the direction that they would like to move the view. The game also utilizes hotkeys for giving faster commands to different units.

Command & Conquer: Generals comes with a 32 page manual and requires 1.6 Gigabytes of hard drive space for installation, with additional game data read from and external DVD ROM.

Chapter 4: Apply and Design

My goal for this master's thesis was to create a casual web-based real-time strategy game. To accomplish this goal, I carefully considered each of the technical issues and social issues associated with web-based games. I also took into consideration the common elements of RTS games, as well as the compelling features of the games that I analyzed. After designing the game, I worked with Professor Brian Winn to implement a playable prototype of the game over a six-month period. I generated all the visual elements, character animation and sound FX and music, and Professor Winn handled the games programming.

I tailored the design of my game to be appealing to current web-based game players. As a real-time strategy game, is falls into the category trade or exploration, and there are no combat or militaristic aspects in the game. Taking into account the demographic breakdown of the audience, I tailored the design of different elements in the game to be humorous and the challenges to be management or puzzle-like and involve problem solving rather than conflict. The characters in the game are designed to be humorous and sexually ambiguous, they could be male or female. The game's levels are designed to be challenging, but not overly difficult or overly time consuming. In contrast to the multiple page instruction manuals of desktop RTS games, the instructions on how to play the game I designed can be explained on a single page.

The game that I designed is called *Mudcraft*. It is the story of the little people made of mud called *Mudpeople,* brought to life by light from above and trying to

survive in a world filled with the hazards of Mother Nature. In *Mudcraft,* the user

directs the development *Mudperson* society, the collection of resources, construction

of buildings, and the avoidance of danger.

The premise

The overarching, goal of *Mudcraft* is to create mud people by gathering

different resources from around the map. At the start of the game, the player has two

mud people. The player must direct the *Mudpeople to* gather resources from around

the map. Thus, allowing them to create more mud people, who in turn gather more

resources.

Along with the overall game goal of promoting the *Mudperson* society, there

will be several mission specific goals such as:

- Starting with 2 or 3 *Mudpeople*, gather resources to build as many *Mudpeople*

as you can or reach a certain number, in a given amount of time. In order to

accomplish this goal, the player may have to build additional items outside of

just mud people to accomplish this. Items such as huts, that the mud people

can take cover in during the occasional rain, or walls, to help protect the mud

people from frogs and turtles.

- With a fixed number of units, survive a given amount of time, without being able to make more units. The player would have to build carefully and make optimal use of a set number of units provided to them in the beginning of the mission in order to survive a given amount of time.

- Reach a certain point. The player is provided a set number of units and must get one or more of those units to a certain point on the map, despite obstacles and other hazards.

Hazards

In the world of Mudcraft, there are a number of forces working against the player.

Rain

Being made of mud has its drawbacks. At various times during the game, the rains may come, and any exposed *Mudperson* will melt, and require another *Mudperson* to drop a load of dirt on them in order to become solid again. In order to avoid melting, the player will to have to develop huts to protect their mud people from the elements.

Sun

Similar to the rain, at various times during the game, a cone of intense sunlight will appear in the world. If a *Mudperson* is exposed to this light, they will become dried out and unable to move. To relieve this state, another *Mudperson* must spray water on the dried character.

Frogs

Periodically a frog may come into the scene. Frogs will try to lick the water from as many mud people as they can, leaving them dry and stiff. To relieve this state, another *Mudperson* must spray water on the dried character. To protect the mud people the player can build walls, which help keep the frogs out.

Turtles

Much like frogs, turtles are also a menace. But unlike frogs, walls will not stop them. Turtles mechanically plod along their path and will break down any structures in their way. And while they will not chase a mud person down, they will trample any that happen to stray into their path.

While the player does not have a direct presence in the game, they are directly tied to the mud people. Should something happen and all the *Mudpeople* under the player's control, such as they are all immobilized, the player loses. As long as one mud person is active, the player persists.

Economy

The basic resources in the game are dirt and water. These are used to create the *Mudpeople* and revive them from various immobilized states. Dirt will be obtained from holes in the ground, while water will be obtained from small ponds. The structures in the game are built using *Mudpeople,* for example: to build a hut the user selects a mud person and directs them to a location and tells them to start a hut, that *Mudperson* then transforms into the first level of a hut. The user then directs two additional *Mudpeople* to become the second level and finally the roof of the hut. By using the *Mudpeople* in this way, I put more attention and emphasis on them, thus make them the center of the game.

Perspective

The primary game play mode will be a top down view of a section of the world. The overall size of a given map may be larger than the player's game screen, and may require the player to scroll in order to view different sections on the map.

Interface

In order to make *Mudcraft* familiar to RTS players, we included some of the common windows found in RTS games, but eliminate some windows that, traditionally, change depending on the unit selected as this may confuse some users not familiar with the genre. Instead the interface for Mudcraft is static, allowing the user to learn and become comfortable with its elements. I came up with a popup menu

46

system, which the user brings up by holding the right mouse button down. The popup menu is context sensitive, displaying commands specific to where the cursor is located. It works like this:

Step 1: The player selects the unit they wish to give orders to by left clicking on the unit (figure 15).

Figure 15: unit selection

Step 2: Once the unit is selected, a ring will appear underneath the unit to denote selection (Figure 15).
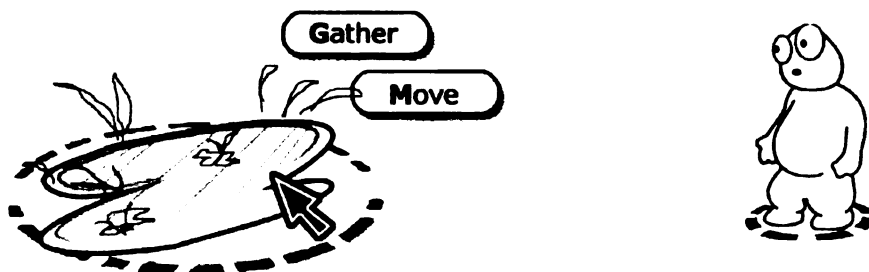
Figure 16: radial menu use

Step 3: They access the radial menu by holding the right mouse button down while the cursor is over the object they wish the selected unit to interact with. The menu will list the actions which the unit selected may perform at that location. The player can then move the curser over the action they wish to select and release the mouse button (Figure 16). If no action is selected the unit will continue with what it was doing.

The player may left click on a special location, such as pond, dirt pit to get additional options for that location. Information on special locations will appear overtop of the location. In Figure 16 the player can see that the unit selected can gather water or move to that location.

The user will control what part of the game map they are viewing through the game window by scrolling. Scrolling can be done with the arrow keys on the keyboard, pressing any of which will cause the viewable region of the screen to scroll

accordingly. Alternatively the player can click on the games mini-map, which will then center the game window on the location represented in the mini-map or mouse scroll by moving the cursor to the edge of the screen in the direction that they would like to move the screen.

Mudcraft also has hot key shortcuts similar to desktop RTS games. Users can speed up interactions with the game by utilizing these keys.

Mudcraft also uses sound to help the user and alert them to different game events. The characters in the game give audio responses when selected or given a command, and when something happens. This helps let the user know that a command was received, draws attention to an event or direct the users attention to something unseen.

Technical Design

There are several software technologies that are available for online game development, such as Flash, Java, DHTML and Shockwave. While each of these technologies has strengths and weaknesses, we chose to develop my project with Macromedia Director for Shockwave delivery. Shockwave has flexibility in the types of media it supports, a robust engine for running relatively large amounts of code and the Shockwave player is freely available for users to download. In addition, developing in Shockwave allows us the flexibility to deliver the game in multiple

ways, such as web-based and for desktop play, as well for multiple platforms, such as for Windows and Macintosh.

Graphics

For the development of the graphics I used two tools, Adobe Photoshop and Macromedia Flash. I used Flash to develop almost all of the artwork for the game. Flash offers tools for character animation. Although Flash is primarily a vector animation tool, I did not use vector animation in the game. Although vector artwork typically is very small in file size, it can be very processor intensive for the computer to render. This will eventually affect the performance of the game, as more and more elements are bought into play. In order to reduce the processing load of the games graphics, I utilized frame-by-frame animation for the character motion. Frame-by-frame animation reduces processor load because the computer needs only replace one graphic with another in sequence, as opposed to vector animation, in which the computer must calculate the mathematical equations needed to render the vector image. Although frame-by-frame animation can lead to an overall larger file size, since each frame of animation is stored as a separate image file, there are ways to manage its impact. Compression can greatly reduce the images' file size. Reducing the number of frames of animation in the game by only using as many as needed to create the illusion of character motion can further reduce the file size.

The interface elements, such as the radial menu buttons as well as the game menus, were designed in Flash, and then exported to Photoshop for cropping and retouching. I also used Photoshop to design the grass texture for the background.

Programming

In order to bring the characters of the game to life, *Mudcraft* incorporates elaborate scripting to allow the *Mudpeople* to appear alive and take action. Each action a character in the game takes must be scripted, and although the code itself may be daunting, the high-level premises are not.

To make the task of coding the characters more approachable, the actions a character might take were broken down into modes. Inside each mode there are additional states that take place. Behind the scenes, these modes and states are acted out through code. The core modes for a *Mudperson* are: *Melt, Stand, Move to Location, Enter Hut, Inside Hut, Gather* and *Drop at Location*, each with its own states:

Melt
- *Begin melt*
- *Melting*
- *Wait*
- *Being birth*
- *Birth*

Melt mode contains several actions. The first three take place when a character is hit by rain. The character then enters melt mode and the *begin melt* state, then the *melting step*, and finally the *wait* step, where they remain until an action is taken. The last two states *begin birth* and *birth* take place when a character is first created and when they are revived from being melted and transitions into *Stand* mode.

Move to Location

- *Move to location*
- *Wait, move to location*

When the command to move is given, a path is calculated between the characters current position and the target location. The *move to location* state moves the character along the path. If the path of the character becomes blocked, they move to the *wait, move to location* state where they will remain for a period of time, at the end of which the character will either continue on the original path or calculate a new path to the target location from the current position.

Stand

Stand mode is an idle state in which the character is waiting to receive orders.

Enter Hut

- *Move to location*

- *Wait, move to location*

- *Wait at hut*

- *Form 1*

- *Form 2*

- *Form 3*

- *Enter hut*

When a character is given commands to build or enter a hut they are put in *Enter hut* mode. The first two states function just like those in the *Move to Location* mode.

When the character reaches the location they move to the *wait at hut* state. During this step, the status of the hut at the target location is tested, which is either none, *form 1, form 2, form 3*, or complete. If there is no hut at the location, the area is checked to determine if a hut can fit at the location. If a hut can fit in the location, then the character moves to the *form 1* state. If not they shift to *Stand* mode. If the hut at the location has *form 1* or *form 2* status, the character will enter the next subsequent *form* state. Lastly, if the hut has complete status, the character will shift to the *enter hut* state. In the *form* states the character forms the next level in the hut and changes the status of the hut. In the *enter hut* sub step the character enters the hut and changes to *Inside Hut* mode.

Inside Hut

- *Inside hut*

- *Exiting hut*

The *Inside Hut* mode consists of two states, *inside hut* and *exiting hut*. *Inside hut*, which is an idle state, and *exiting hut*, which exits the hut and transition into *Standing* mode.

Gather

- *Move to A*

- *Wait, move to A*

- *Wait at A*

- *Enter A*

In *Gather* mode, the character is directed to a location to pick up a resource. The first two states function just like those in *Move to Location* mode, with A as the location. In the *wait at A* state it is determined if location open for the character to collect or if the character must wait for an opening to become available. When an opening is available, the character enters the *Enter A* state, in which the character gathers the resource and an amount of the resource is subtracted from the location and added to the character. From there, the character enters *Drop at Location* mode.

Drop at Location

- *Move to B*

- *Wait, move to B*

- *Drop*

The first two states function just like those in *Move to Location* mode, with B as the destination for the resource being carried. The next state is similar to that of the second state in *Gather* mode, it is determined if location open for the character to drop the resource being carried, if so they shift to the *drop* state, if not, they wait until an opening is available. In the *drop* state, the resources are transferred from the character to the destination and the character then returns to *Gather* mode, *Move to A* state.

Breaking the code for a character into different modes makes programming the game more approachable and more efficient, in that only the code associated with the character's actions needs to be handled at a given time.

Chapter 5: Evaluation

Methods and Demographics

After designing and implementing the demo, I conducted a user test to evaluate the game's design and gather information about how the game may be improved. For the evaluation, I recruited eight individuals, four males and four females, to play the demo for ten minutes. All were people with whom I am acquainted through work, classes or socially who are familiar with real-time strategy or web-based games.

All four male participants had played RTS games, and none of the female participants had done so. All four male participants had played web-based games, and only one of the four females had done so. Overall the females were younger. Female age ranged from 22 to 32 with an average of 25, while male ages ranged from 28 to 36 with an average of 32.

Pseudonym	Gender	Age	RTS Experience	Web-Game Experience
Sarah	Female	32	Never played an RTS game	Has played
Jessica	Female	25	Never played RTS game	Does not play
Mandy	Female	22	Never played RTS game	Does not play
Julie	Female	22	Never played RTS game	Does not play
Ben	Male	28	Has played RTS games	Has played
Kevin	Male	34	Has played RTS games	Has played
Andrew	Male	29	Has played RTS games	Has played
Tim	Male	36	Has played RTS games	Has played

Table 3: Subject profiles

Participants were asked to play the game for ten minutes while being observed. During the testing, the players were asked to think aloud and voice any difficulties they were having.

After playing the game for ten minutes, the players were asked to stop and participate in a short interview. During the interview, I asked a series of questions about their experience with the game.

Because there were such extreme disparities in experience with RTS games between the male and female subjects, and because research suggests strong gender differences in how people react to games, the observation and survey results will reflect the differences between the four (experienced) male subjects and the four (inexperienced) female subjects.

Play Observation Results

36 year old Tim has played RTS games and web-based games. Tim immediately commented that the game did not include a mouse scroll (not implemented in the version of the demo tested). He commented repeatedly on the need for this feature. Tim asked how to make *Mudpeople*. He created a hut with the initial two characters, causing him to fail the mission and then said the game should not let him do that. Tim used the pop up menu a lot at first, and later the hotkeys. He seemed to have a hard time controlling the characters at first, but improved over time and had the hang of it at the end. Tim commented when building a hut, that he was used to other RTS games

where you pick what you want to build then place and outline of it in the location you want to build it. He really enjoyed the game and did not want to stop.

34 year old Kevin has played RTS games and web-based games. Kevin questioned the dirt under the eyes of the dirt carrying *Mudperson;* He thought they looked liked mustaches. He asked about how many people can fit in a hut (twice). Kevin seemed to have a hard time understanding some of the audio cues, although he laughed at most of them. He did not immediately identify the dirt hole as the source for dirt. He seemed to enjoy the character animations. Kevin asked how to get the characters to collect resources and how to create another *Mudperson*. He had a hard time going through the interaction to create a hut. He commented that the delay for popup menu may be too long. Kevin missed the audio cue marking the collapse of the dirt hole. He did not recognize what the items on the mini-map were and asked what amount of dirt and water are needed to create a Mudperson. He seemed to really like the shortcut keys and used them a lot in giving orders.

28 year old Ben has played RTS games and web-based games.
Ben thought that there should be rollovers on the menu items. He laughed at the opening animations. Once in game, he asked what he should be doing and how to do things. Ben had difficulty selecting *Mudpeople* that were moving. He seemed to enjoy the animations of the characters as they went about different tasks. Ben had a hard time identifying different items in the world, such as the dirt hole. He asked questions about the nature of the different elements in the world and what they are

58

utilized for. He did not notice the option button and had some initial difficulties building a hut.

29 year old Andrew has played RTS games and web-based games. Andrew used the skip button on the intro right away. He did not immediately identify the dirt hole. He seemed very comfortable playing the game. Andrew seemed to like the behavior of the rain, and how it worked. He asked about how many people were needed to fill a hut. He liked the quick keys and used them a lot. Andrew only used the radial menu to build a hut. He commented that the rain should replenish the water hole, "logically". He liked the animation of the character gathering water and said he really liked the music.

32 year old Sarah had never played an RTS game but a frequent player of web-based games. Sarah seemed to enjoy the game. While playing she laughed and said, "oh, gross" when the animation of a *Mudperson* played at the start of the game. She laughed at the level 1 description screen. Sarah seemed to have some difficulty using the popup menu system at first. When she was able to successfully use the menu, she said: "it would have been cool to have a picture of how to use that (the menu)". Although she failed on the first attempt a level 1, due to difficulty giving orders to the characters, she wanted to play it again: "I have to try that again". After the initial attempt she seemed to catch on to the games interface and seemed to enjoy playing.

25 year old Jessica has never played RTS game and does not play web-based games much. Even so, Jessica did not read instructions initially, but went back after she got stuck. She frequently went back to the instructions to review them. Jessica misidentified the dirt hole as a hut and had a hard time with the interaction to create a hut. She needed direction (from the observer) on how to accomplish the task of getting the hut started and completed. She liked the sound a lot and giggled a lot at the different voices. Jessica was confused at first about what she should be doing. She did not notice the rain cloud represented on the mini-map at first. She made very good use of the 'get into hut' shortcut button and seems comically sad when a *Mudperson* melted.

22 year old Mandy has never played RTS games or web-based games. Mandy spent a lot of time reading the instructions. She immediately used the popup menu with no trouble. She referred to the instructions in the in-game menu when stuck. Mandy had a hard time creating a hut, thinking that she needed to click on the *Mudperson* where they stood, instead of an open area where she wanted the hut built. She needed help on how to revive a melted character. Her play improved over time and she used hotkeys more and more as time moved on.

22 year old Julie has never played RTS games and doesn't play web-based games. Julie giggled and laughed at the games intro animation. She needed further instruction beyond the in-game instructions on how to use the popup menu. She was confused with the interaction system at first, left clicking to select a character and

then right clicking to give an action. She commented, "I'm not very good at games".

Julie needed explanations of concepts such as gathering resources, a 'main base', dirt

hole and water hole. After 5 minutes of play she commented "I think I get this". After

ten minutes she started to use the hot keys a lot more. Julie never noticed the

representation of the cloud on the mini-map.

Synthesis of Observations

	Tim	Kevin	Ben	Andrew	Mandy	Jessica	Sarah	Julie
Giggle/laugh	N	N	Y	N	Y	Y	Y	Y
Read Instructions	Y	Y	Y	Y	Y	N	Y	Y
Looked at in-game instructions	N	N	N	N	Y	Y	N	Y
Understood how to use popup menu based on written instructions	N	N	N	Y	N	N	N	N
Used hotkeys	Y	Y	Y	Y	Y	N	N	Y
Had trouble at first	Y	Y	Y	N	Y	Y	Y	Y
Understood what to do to achieve mission goal	N	Y	N	Y	Y	N	N	N
Got the hang of it after 10 minutes of play	Y	Y	Y	Y	Y	Y	Y	Y
Asked for help	Y	Y	Y	Y	Y	Y	Y	Y
Misidentified dirt hole	Y	Y	Y	N	N	N	N	Y
Misidentified water hole	N	N	Y	N	N	N	N	N
Recognize mini-map	Y	Y	N	Y	Y	N	N	N
Enjoyed the game	Y	Y	Y	Y	Y	Y	Y	Y

Table 4: User observation synthesis

61

Five of the eight subjects laughed or giggled while playing, including all of the 4 female subjects and 1 of the male subjects. Three of the four female subjects used the in-game instructions repeatedly, while none of the males did. Despite their experience with RTS games, all of the male subjects asked for or needed help trying to play the game, as did all of the less experienced female players. All but one of the players had trouble figuring out the game at first, but most eventually all got the hang of it. All of the males and two of four females eventually used the hotkeys.

Survey Results

After playing for 10 minutes, subjects were asked 17 questions about their impressions of the game. All 8 subjects enjoyed the game. All would like to play a future version, know others who would like to play the game and would like to play other games like this online. They all like the concept and all found it engaging. Sarah remarked it might be little too engaging because "I could see myself wasting a whole lot of time (playing the game)". Seven of 8 respondents thought it was just the right length, while Andrew wanted it to be longer. No one thought it was too long. The players were very positive about the look of the game and its sound effects and music. Males tended to comment that it looked "cool" or "neat" or "funny," while females called it "cute" as well as "cool" and "funny." Although Sarah disliked the music, saying it was "too twangy," Mandy liked it so much she wanted to obtain the sound track. Kevin noted that sometimes the sounds got repetitive, but again, overall, people strongly liked the music and sound effects.

The participants were so positive about the game but suggesting that the instructions were a bit lacking. Two of the females said it was OK eventually although a little hard at first. The other two females said it was not too complicated. Two of the males said it was too complicated at first, and the other two males thought it was fine. This same pattern of responses occurred when people were asked whether it was easy to learn. Three respondents said the first level was too hard for a first level. Several mentioned it also gets harder when you have more mud people to manage.

As far as suggestions for improvements, all of the players suggested more instructions or documentation, Ben: "what should I be doing". Two suggested a tutorial of some kind.

User Test Summary

All of subjects showed some form of initial confusion when they first began playing the game, but all of them were able to get the hang of the game after the first ten minutes. The confusion seemed to last longer for those who had never played a real-time strategy game, but no one was unable to figure the game out.

Initially I believed that the instructions for the game should be brief and only cover the basic mechanics, in order to resemble current web-based games. After testing, I believe that web-based RTS games will need much more detailed

63

instructions available for the user. I believe that the instructions should also contain visual images, identifying the different game elements and menu items. I also believe that the first level of the game may need to be a simple, tutorial level, which explicitly shows how to control the characters and tack various actions. Looking at the user testing feedback, I believe that much of the confusion could have been reduced if the demo the subjects played included these elements.

Initial confusion aside, all of the subject were able to understand and enjoyed the game after ten minutes of play. All the subjects, except for Sarah, asked to continue to play past the allotted time. And all the subjects, except for Sarah, asked if and where the game would be available for play. Since the testing, all of the test subjects have contacted me with addition positive feedback about the game. Each of subjects seemed very impressed with the game and enjoyed playing it.

Chapter 6: Conclusion

Possibilities

After the successful development and testing of Mudcraft I can conclude that it is entirely possible to create real-time strategy game for web-based delivery. Furthermore, by taking into consideration the disposition of the audience and looking at the design of popular web-based as well as the compelling features of current RTS games, it is possible to deliver a web-based RTS game that is appealing to a wider audience than the current, male dominated audience.

The most appealing features of Mudcraft is its humorous design and ease of play. It is challenging, but not too daunting to learn and shows that and RTS games can be designed that are fun and appealing without being focused or including any combat or violence. The game is 4.6 Megabytes in size yet contains many of the features found in RTS games available today.

By creating a game of this genre, with this complexity, we show that it is possible to create games with greater quality and complexity than those found on the web today and that the Internet is a viable delivery medium for robust games. This opens the door not only for more real-time strategy games, but games from other genres that are also underrepresented on the Internet.

As broadband availability and use becomes more prevalent, and as Internet access speeds increase, the most limiting technical factor, bandwidth, will become less of an issue. I believe that as this happens, and more and more devices have access to the Internet, users will look for more ways to spend there time, and more compelling games to play. As I have shown, it is possible for we, as game designers, to deliver those games.

The Future of *Mudcraft*

Brian and I really enjoyed working on Mudcraft and we plan to continue to develop the game. Many of the changes asked for in the user testing, such as mouse scroll and more detailed instructions, have already been implemented. We will be adding more interactions and antagonists, and creating new environments for the Mudpeople to explore. We have created a web site, www.mudcraft.com, where people can go and play the most current version of the game, and submit feedback. We plan to continue improving the game and hope to entice a large audience of web-based game player into becoming *Mudcraft* fans.

Bibliography

Aspyr Command & Conquer: Generals. Electronic Arts, 2003.

Auran Dark Reign. Activision, 1997.

Bahlmann, B. Online Gaming: Getting Down to Business. Birds-Eye.net, March 2002.
http://www.birdseye.net/article_archive/online_gaming_getting_down_to_business.htm

Blizzard Entertainment Warcraft: Orcs & Humans. Blizzard Entertainment, 1994.

Blizzard Entertainment Warcraft II. Blizzard Entertainment, 1995.

Blizzard Entertainment. Starcraft Blizzard Entertainment, 1998.
http://www.blizzard.com/starcraft/

Burnham, V. Supercade: a visual history of the video age. MIT Press, 2001.

Cavedog Total Annihilation. Activision, 1997.

Fattah, H. and Paul, P. Gaming Gets Serious - market, survey data on online gaming - Statistical Data Included. Looksmart.com May 2002.
http://www.findarticles.com/p/articles/mi_m4021/is_2002_May_1/ai_88679445

Geryk, B. A History of Real-Time Strategy Games. Gamespot.com 2002.
http://www.gamespot.com/gamespot/features/all/real_time/

Heeter, C. et al. Alien Games: What happens when girls design space learning games? Michigan State University, 2004.
http://spacepioneers.msu.edu/

Leiner, B. et al. A Brief History of the Internet. Isoc.org, Dec 2003.
http://www.isoc.org/internet/history/brief.shtml

Microsoft Age of Empires. Microsoft, 1997.
http://www.microsoft.com/games/empires/

Mitchell, B. DSL vs. Cable Modem Comparison – Speed: Predictable performance or raw speed... can you choose? About.com
http://compnetworking.about.com/library/weekly/aa111200b.htm

Moran, J. NPD Funworld Releases Video Game Industry Sales Figures. Gamemarketwatch.com, Nov. 2002

http://www.gamemarketwatch.com/news/item.asp?nid=2604

Morris, C. Madden football headed to Xbox Live. CNNMoney.com, May 2004.
http://money.cnn.com/2004/05/11/technology/e3_xboxlive/index.htm

Mudd, G. Marketers Take Note: The Elusive 18-34 Year-Old is Habitually Online.
Comscore.com, March 2004.
http://www.comscore.com/press/release.asp?id=445

Nielson, J. The Need for Speed. Useit.com, March 1997, updated Jan. 2004.
http://www.useit.com/alertbox/9703a.html

Rollings, A. and Adams, E. On Game Design. New Riders Publishing, 2003.

Speed. Broadbandreports.com – website
http://www.dslreports.com/speed

Starcraft legacy – website
//www.sclegacy.com/

US Broadband Penetration Jumps to 45.2% - US Internet Penetration Nearly 75% -
March 2004 Bandwidth Report. WebSiteOptimization.com, June 2004.
 http://www.websiteoptimization.com/bw/0403/

Westwood Command & Conquer. Virgin Interactive, 1995.

Westwood Command & Conquer: Red Alert. Virgin Interactive, 1996.

Westwood Dune II. Virgin Interactive, 1992.