HUMAN CREATIVITY THROUGH
COMPUTER GAMING
Computer Science Honors Thesis

Christine McGavran

Technical Report PCS-TR94-225

7/94
Human Creativity Through Computer Gaming

Christine McGavran, June 1994
Chapter 1
How I Became a Computer Games Programmer

When I was in third grade, my family bought a Commodore 64. The first program we got for it was LOGO, so that my father could claim that the computer was an educational tool for his children; the second thing we bought was a computer game, so that my father could play along. Life with the Commodore 64 oscillated between playing and programming. We bought every popular game, and even subscribed to a magazine that sent us a disk of free games every month. But my father and I also played with programming—we were in constant competition as to who could create the coolest program. He wrote a graphing program, I wrote a program to sing “Happy Birthday;” he studied assembly language, I typed in assembly code for games from the back of magazines. My brother was only interested in the games. He was a whiz at Donkey Kong, while I concentrated on The Pinball Construction Set, and we both worked together on text adventure games. Computer games and computer programming were both strong parts of my life.

But when we graduated from the Commodore 64 to a Macintosh and a Nintendo machine, I gradually lost interest in the games side of things. The Mac had almost no games, while the Nintendo only had games that bored me. My brother was heavily addicted, spending every afternoon with his friends playing their latest acquisitions. Super Mario Bros interested me for exactly ten minutes, Zelda perhaps amused me for twelve. By the time I reached Dartmouth, I had forgotten any past addiction to computer games, and concluded that most of them (besides Tetris) were pointless. So I was somewhat surprised to find myself working on computer games with Silicon Graphics my junior summer. But, surrounded by creative talent, I again became interested in computer games, and wondered exactly what it was that turned me off to them.
My brother and I are both computer science majors now, predictably. But sometimes I wonder if I would have chosen this major if my first game system had been a Nintendo instead of a Commodore 64. Games definitely fueled my interest in computers, and the popular game systems of today contain almost nothing to excite me. I undertook this thesis to investigate why games like Nintendo’s turn so many people off, especially females, and to find alternative ways of approaching games.

The first book I read on the subject was Sherry Turkle’s *The Second Self: Computers and the Human Spirit*. In this book, I found a fascinating interview with a young boy who wished that he could reach inside of a computer game and change its contents, to reprogram games to fit his fantasies. It struck me that this was how I had once viewed computer games, and games that came with source code or those such as *The Pinball Construction Set* (that encourages the player to create her own pinball machine) had allowed me this creative freedom. On the other hand, games such as *Super Mario Bros* for the Nintendo bored me, not only because I couldn’t identify with the fantasies and adventures involved, but also because they were so static and unchangeable.

The central idea that I will present in this thesis is that computer games should allow for more flexibility and creative input. Adding this dimension to games would have several important consequences. First, more creative games would encourage more creative thinking, making computer gaming an educational and mind-opening activity. Second, these types of games would help break down the barrier between computer users and computer programmers, allowing users to participate in the design of the program and encouraging them to become programmers themselves. Finally, game systems such as Nintendo currently present a limiting set of fantasies. Allowing the players to customize their games would free them from having to identify with many these fantasies, and encourage them to create new fantasies that are more personally appealing.
The next chapter of this thesis investigates the social impact of computer games, discusses their positive and negative consequences, and introduces ideas about how to make computer games more flexible. The third and fourth chapter discuss two games I have written for Silicon Graphics workstations and how they encourage creativity. Chapter 3 is about *Tagh*, a multiple-player game that allows freedom of rule and goal definition. Chapter 4 discusses *Compose*, a painting and music composition game whose attraction lies precisely in its creative potential. This thesis demonstrates that computer games do not have to provide rigid fantasy worlds and rule systems to be attractive, and that introducing creative freedom into computer games can increase their audience and effect positive social consequences.
Chapter 2
The Potential of Computer Games

2.1 Introduction

Computer and video games have been an influential part of American culture for over two decades. Since Pong, the first computer game, was introduced by Atari in 1972, computer games have fascinated and entertained both children and adults. By 1981, home video games had become a billion dollar industry [P91, p. 9]. Following a brief drop in popularity in the mid-1980's, the game market jumped again with the introduction of the Nintendo game machine, and by 1990 the game market was stronger than ever [P91, p.10]. Today, in 1994, video game revenues are $5.3 billion a year [E93]. Video games are again capturing public interest, as new technology paves the way for a series of new high-powered game machines. All the major players—Nintendo, Sega, and Atari—are introducing game machines that are custom designed for 3-D graphics and high speeds; a few new companies, such as 3DO, have also entered the race. These new machines, which will allow for games that are more impressive graphically and very different visually from those being sold now, admit the possibility of defining new standards in the design of computer games.

Computer games have the capacity for influencing whole generations. They have grown to be one of the chief forms of entertainment for children. In February 1989, 16 of the 20 top selling toys in the USA were video games or video game-related [P91, p. 12]. Computer games, therefore, have the potential for influencing millions of children. We should not take their presence lightly.

This chapter discusses the characteristics of computer games, their positive and negative effects, their role in American culture, and their potential as positive aspects in children's lives. We suggest new ideas for designing computer games and highlight games that seem exemplary. A central thesis throughout shall be that
computer games have an incredible creative potential that often goes untapped, and that if designed well, computer games can encourage creative thought and problem solving while still being enjoyable.

2.2 A Definition of Computer Games
Throughout this chapter, we shall refer to computer games as including video and arcade games. Traditionally, there has been a distinction between

- computer games, which are played on a computer not specifically designed for games,
- video games, which are played on video game systems designed to play a number of cartridge games,
- arcade games, which are played on a machine designed for only playing that particular game.

Since the ideas presented here apply equally to computer, video, and arcade games, we shall use the term "computer game" unless we are referring to a specific game system.

It is important to draw a distinction between computer games and computer tools. This distinction is not always obvious. For example, a drawing program can be used for practical purposes or it can be used in the context of play. In this case, we suggest that the drawing program is either a game or a tool, depending on the context in which it is used. Similarly, programming languages, especially those such as LOGO (a graphical language designed as an educational tool for children), can be used entirely for recreational purposes. Therefore, we will use a loose definition of a computer game, as suggested by Thomas Malone's distinction between tools and toys:
Toys can be defined as systems used for their own sake, with no external goals (computer games, puzzles, etc.). Tools can be defined as systems used to achieve external goals (text editors, programming languages, etc.). With respect to challenge, the requirements for good toys and good tools are mostly opposite. Since a good tool is designed to achieve goals that are already present in the external task, it does not need to provide a goal... In a sense, a good game is supposed to be difficult to play: that increases its challenge; but a tool should be as easy as possible to use. This distinction helps explain why some users of complex computer systems may take a perverse pleasure in mastering tools that are extremely difficult to use. To the extent that these users are treating the systems as toys rather than tools, the difficulty increases the challenge and therefore the pleasure of using them. [M81]

Similarly, in his book *Homo Ludens* (Man the Player), Richard Huizinga defines play as being defined by its purely internal goals:

Summing up the formal characteristics of play, we might call it a free activity standing quite consciously outside “ordinary” life as being “not serious,” but at the same time absorbing the player intensely and utterly. It is an activity connected with no material interest, and no profit can be gained by it. It proceeds within its own proper boundaries of time and space according to fixed rules and in an orderly manner. It promotes the formation of social groupings which tend to surround themselves with secrecy and to stress their difference from the common world by disguise or other means. [H49, p. 13]

Therefore, we shall see a game as defined by its internal challenges. Although most games are seen as such while being created and their challenges are designed into the structure of the game, it is possible for many applications designed as tools to be viewed as games if their players can find pleasing challenges in using them.

We have classified computer games as a form of play, which puts them in the same class as baseball, make-believe, and fingerpaints. In all of these systems, the activities exist as part of microworlds separate from the “real” world in that they have their own internal rules, fantasies, and conceptions. All games, including computer games, are partaken in because of the pleasures they help to produce. Therefore, we can see computer games as a new way of playing. Play is an activity
common to all children, and it is undoubtedly an influential part of human development. Since computer games are dominating the toy market, and therefore are a very popular way of playing, computer games contribute substantially to the development of children in today’s society.

It is important, however, to notice essential differences between computer games and more traditional methods of play. One key difference is in who defines the rules. Although the characteristics of a doll can affect the way a child perceives it, that child is still free to incorporate that doll into her games in whatever way she pleases. This freedom of rule definition exists around most children’s toys. With computer games, however, the player usually must conform to the rules that the programmer sets. For example, if you don’t kill the monster at the end of a level in games such as Teenage Mutant Ninja Turtles, then you won’t be able to progress any further, and your character will most likely die. Although some non-computer games, especially sports games, impose strict rules, players are still free to decide on new rules, strategies, or variations if they please. With many computer games, there is often only one way of doing things.

Another key difference is in the choice of fantasies. When a child plays a computer game, she is living in the fantasy world of the game designer. But if she is playing make-believe with her friends, they are able to design the world around them. While computer games are superior to static media such as television in their interactivity, most often they still do not let the player contribute to the definition of the microworld.

The above distinctions should not be seen as defining characteristics of computer games, but more as prevailing themes. It is possible to create a computer game that allows users to define their own rules and microworlds, and some games provide this freedom. However, the tendency in the game market is to distribute games that limit the users’ creative freedom.
A final feature of computer games is their freedom from the natural laws of the world. A character in a computer game does not have to conform to gravitational laws, use normal means of propulsion, or look like any recognizable being. Computer games can depict imaginary futures, social systems, and physical laws. Imagine a world, and a game programmer can probably produce something that looks and acts remarkably like it. In this way, computer games have an incredible potential for encouraging creative fantasies. This capacity of computer games, however, is most often reserved for the programmers.

Computer games are an element of play. We have loosely defined them to include any computer tool that the user perceives as a toy. Finally, although computer games are in many ways like other toys and are as influential, they possess certain unique characteristics. These characteristics are not necessarily rigid and can be seen positively or negatively.

2.3 Impact and Influence of Computer Games

We have argued that computer games play a large role in the development of children in the United States. In this section, we examine that role and discuss its positive and negative aspects.

When computer games came into the public eye in the early 1980's, they incited much debate about their social value. People were concerned about aggressive themes, addictive tendencies, playing expenses, gender differences, and arcade culture. For example, in 1983 the U.S. Surgeon General C. Everett Koop claimed that arcade and video games were among the top three causes of family violence [AF86]. Some communities considered banning video game arcades, arguing that the games were addictive and that children were spending too much money playing them [G84, p. 97]. However, the mood among researchers was highly optimistic. Patricia Greenfield, for instance, presented evidence that only a small
percentage of young people playing arcade games displayed addictive tendencies or spent lunch money on games [G84, p. 98–99]. A prevailing hope among researchers was that as game technology developed, the designers would become more conscious of the social implications of their games. For example, in 1983 Elizabeth and Geoffrey Loftus said the following about the disturbing lack of girls in arcades:

Video game companies are currently spending countless hours planning ways of gathering such neglected players as very young children, girls, and mothers into the fold. Of course, the companies have their own economic reasons for this strategy. But, assuming that the companies succeed, a potentially grave social split may be averted. [LL83, p. 107]

This split has not been averted. Today, males are still the major users of video games. In fact, Spectrum HoloByte, a game distributor, reports that for most of its games, 99% of the buyers are male [C94]. It may be that video games are becoming even less attractive to women. For example, in a discussion about computer games on a women in computer science mailing list (Systers), a prevailing theme was nostalgia. One woman wrote me,

We had an Atari game system when I was growing up, and most of the games appealed to both boys and girls—games like Pong, Breakout (bounce a ball off a wall and knock out bricks), and one whose name I can’t remember, but the goal was to shoot down these thread things as they fell from the sky, before they destroyed the cities on the ground. There was a distinct lack of “kill the bad guy in some obscenely violent way” games.

This comment strikes one as odd when it is realized that the last game she mentioned in her array of non-violent games was (or at least resembled) Missile Command, a game that was once repeatedly described as disturbingly violent. In 1982, for instance, Carl Rogers worried that the game might “constitute a trivialization of nuclear war.” As he put it, “We are making nuclear war thinkable by treating it as though it were just a game” [LL83, p. 100]. Now, in comparison to
such past relatively impersonal violent games, we see our present games as “obscenely violent” instead of just “violent.” For many of the women I talked to, impersonal violence has become acceptable in a day when the most disturbing video games show such animations as someone pulling the heart out of someone else's chest.

Two characteristics of computer games, violent and male-focused, are the most disturbing in terms of their possible social impact. Violence has received much more attention than gender issues. In fact, in interviews conducted by Eugene Provenzo, he found that few parents had thought that video games contained any gender bias, and if they were at all concerned about the games it was because of themes of violence and aggression [P91, p. 101]. While gender issues should be at least as troublesome, parents are right to be concerned about violence—many experts agree with these parents that the games’ violence is harmful.

One argument for the harmlessness of computer games has been that these games don’t compare to television in their realistic content. Computer games contain characters that cannot be identified with, and situations that are obviously not real. In 1984, when this might have been a true statement, Joseph Dominick argued that,

Video game violence is abstract and generally consists of blasting spaceships or stylized aliens into smithereens. Rarely does it involve one human being doing violence to another, as if often the case in conventional television. Video game violence might perhaps be more closely related to abstract violence in some TV cartoons. [P91, p. 66]

He would have a difficult time arguing this in 1994. Not only have video games increasingly focused on human characters in violent interactions, they have recently begun to include real video footage of this violence. For example, two recent and popular violent games, Night Trap and Mortal Kombat, both feature
live-action video [E93]. In addition, games have been striving to imitate reality through new virtual reality technology such as the Power Glove (a glove that translates hand movements into movement on the screen) and head-mounted displays (which promise to make the user feel as if she is inside the scene). As games become closer and closer to real life, how will we stress the differences between violence in the two worlds?

Many experts argue that computer game violence is more harmful than television violence, because it allows children to actively participate instead of merely observe. For example, University of Southern California professor Marsha Kinder, argued “It’s worse than TV or a movie. It communicates the message that the only way to be empowered is through violence” [E93]. Computer games can be more engaging than television, since the player has a vested interest in the outcome. They also encourage greater identification with aggressive characters.

In a meta-analysis of available research on violence and computer games, Eugene Provenzo demonstrated that people who are given violent computer games to play, compared to those who are given non-violent games, tend to show increases in aggressive behavior, anxiety, and assertive fantasy behavior [P91, p. 69]. This indicates that violent games probably have a direct effect on behavior, at least immediately after they are played. Provenzo concludes,

What emerges from the studies on video games and aggressive behavior is that there does seem to be a significant relationship between aggressive behavior on the part of subjects and the playing of video games. What the long-term impact of the games on aggressive behavior is not known. [P91, p. 69]

Provenzo also stresses that most of this research was done with violent games of the 1980’s, which do not compare in intensity to those being played today. What the new effects of more realistic games are is not yet known.
A secondary effect of violent computer games is their effect on females. First, it might be the case that girls are less interested in aggressive themes in games, and if most games are aggressive, then few will be attractive. An article printed recently in the Wall Street Journal hints at this relationship. The article was about the game Tetris, a non-violent game that involves fitting geometrical shapes together. 40 percent of the buyers of Tetris for Nintendo were females, compared to 1 percent for most other Nintendo games [C94]. Nintendo has become intrigued by this statistic, and has hired psychologists to discover why Tetris is so attractive to women. Preliminary speculations are that, “Tetris may be particularly good at releasing female endorphins... because it satisfies a woman’s craving for order in a way that other video games don’t” [C94]. When I posted a summary of this article to two computer discussion groups (Systers and rec.games.programmers), the predominating reaction was disbelief. None of the many women who reacted to this article believed the statement made by the Nintendo psychologist. Instead, most pointed out that Tetris is one of the few non-violent games available from Nintendo. For example, one woman wrote, “My gut feeling is that women play Tetris because it’s not a game based on killing.” Nintendo games certainly are heavily biased towards aggressive themes. Provenzo noticed that “of the 47 games identified by the [Nintendo Power ] Player’s Poll, only 7 do not have violence as their major theme.” [P91, p. 127]. The violent themes may very well contribute to their unattractiveness to females. At least one study has concluded that women feel less comfortable than men with aggressive themes in video games [P91, p. 60]

Another gender-related issue around violence is role identification. Most violent characters in games are male, while most females play passive or victim-oriented roles. For example, while surveying the covers of 47 Nintendo games, Provenzo noted that of the 115 male and 8 female characters depicted, “twenty males were identified as having dominant poses, while no females were identified in this
classification. Three females were clearly in submissive poses (one-third of the total), while no males were in corresponding representations” [P91, p. 108]. He also noticed that 30 percent of the games contained scenarios in which women were kidnapped and had to be rescued as part of the game [P91, p. 109]. These games are communicating to children that while males are supposed to be violent and aggressive, females should be submissive and are often passive victims of violence.

One of the most striking examples of role identification can be seen in advertisements for two Sega games. These ads appeared in popular game magazines, one for a game called Belle's Quest, the other for Wild Thing. Both games were made by the same company, and were both based on the Disney movie, Beauty and the Beast. It was evident that the first game was written as a girl’s game, and the second was intended for boys. Belle's Quest was presented on a pastel background, with the following text description:

Break The Spell—You'll need all of Belle's charm and wit to save her father, imprisoned by the hideous Beast. But to reach him you must first outsmart persistent Gaston. Then journey astride your faithful steed Phillipe, to a thick, black forest maze of wild animals and hidden traps to reach the enchanted castle on the other side. Search carefully for the castle's hidden secrets. Find them all to break the spell and return the Beast to his human form.

Wild Thing, on the other hand, featured a picture of a beast coming out of the page, growling and swiping a claw at the viewer. It is described as follows:

Roar of the Beast—Unleash the untamed power of the Beast! All of his rage and fury are yours to command. Protect your castle from sinister monsters, vicious rats, bloodthirsty wolves and the wicked Gaston. Summon all your strength. Because to free yourself from this inhuman form you must win every battle or be doomed to remain a beast forever.

Although Belle is perceived as an active player in her game, it is interesting to note the differences in the male and female versions. All of the descriptive words in
Belle's Quest are non-violent and suitable to girls, such as "enchanted," "carefully," and "faithful." This wording in itself is not disturbing, until we contrast it with the boys' game. Here all of the descriptive words fit into an aggressive theme, such as "untamed power," "vicious," "bloodthirsty," "strength," and "win every battle." Belle is not mentioned at all in this game. Instead, the message presented is that the Beast must control his own destiny through violent means, and Belle is a non-entity with little effect on the Beast. While a girl might envision that she is helping the Beast, a boy would never acknowledge the girl's active role in the adventure. Finally, the boy is never meant to pursue non-violent means for becoming human. This last theme contradicts the entire message of the movie, which is that the only reason the Beast wasn't human was because he had been cruel and unfeeling in the past. In order to become human, he has to be gentle enough to win Belle's love. The video game twists the plot of the movie in order to enforce the designers' notions about gender roles. The message here is that girls are passive and boys are violent. Moreover, it tells us that non-violent action is useless to boys.

Computer games have two disturbing tendencies in relation to gender. First, as we have discussed, they can impose damaging notions about gender roles. Certainly, we do no want to teach our children that there are any preferred roles for a gender group, much less that these are roles in which males dominate and females are seen as non-present or victims. The other tendency is that, for whatever reason, boys are much more attracted to most computer games. This unequal attraction is distressing because computer games are often the first significant contact with computers and technology. If girls are alienated by these games they may in turn feel alienated by computers. Since there are so few women pursing computer-related careers, there should be more effort to interest girls in computers and computer games early as a way of balancing the field.
With all of these disturbing characteristics, it would be easy to condemn computer games as one of society's evils. The situation, however, is not hopeless. Throughout this account we have said that most computer games are violent, or many computer games are gender-biased. There are many games available that do not fit these negative descriptions, especially on home computer systems, which are more available to low-budget game development. Anyone who dislikes the computer games that are available is free to write and distribute a new and more personally appealing game using their personal computer. A number of such games are available for home computer systems; some of these games are described at the end of this chapter. As the number of families with personal computers increase, the influence of these games grows.

New game technology brings as many promises as it brings fears. With new technology comes a wider range of possible games, many of which will undoubtedly be acceptable. For example, networking is a technology that is rapidly becoming more prevalent. With computers and televisions that can talk to each other, games can be developed (and in some cases have already been developed) that involve cooperation and communication among groups of real people. Such game playing can help to teach valuable social skills, as well as encourage creative problem solving, since the player is not dealing with pre-programmed characters whose actions can be memorized.

Those who are concerned about the negative effects of computer games should not give up hope. Games in 1999 will probably look about as different from the video games of today as Mortal Kombat looked to Pacman. With the proper encouragement, this change could be a change for the better.
2.4 The Creative Potential of Computer Games

In 1984, Sherry Turkle published an insightful book called The Second Self: Computers and the Human Spirit, that included a chapter on computer games. Part of this chapter explores a young boy’s fascination with computer games, and implies that video games open a window to the world of creating things with computers. For Jarish, the boy she interviewed, computer games are more than just a recreational activity. They inspire him to investigate the world of computers and possibly design his own games. As he said,

I would like to change games to make them crazier, like if you were in a two-player game, shooting another guy. I’d make it so like you’d fire these little weird rockets and then your friend could, let’s say, press a button, and the rocket would turn into a bunch of, let’s say, ants, and they’d fall around everything and you’d have little crater holes and missile silos coming out of the ground. [T84, p. 73]

Jarish obviously has a vivid imagination, one that he’d love to use more. A perfect game for him would be not only one that could be played, but also one that could be altered to fit his fantasies:

My biggest interest would be having a terminal. Like one that you can connect to any computer. That would be incredible. My friend has an Apple. She can attach it to a giant computer. It costs about a million dollars. If you can get into different computers you could get the different codes about the computers and different languages and things about it, and take games from them—you know, like games that you couldn’t find anywhere else—and transfer them to your own computer, and change the games into anything you like. That would be really terrific. That’s the stuff I’d like to do. [T84, pp. 73–74]

Evidently, Jarish is destined to be a computer programmer. Already, by the time of this interview, Jarish had learned to type game code from magazines into the computer and alter it. But what he really wants is a way to be able to easily modify any game, including those with their code hidden from the players.
Sherry Turkle has demonstrated through this interview that computer games have the capacity for exciting children about the computer culture and computer programming. This capacity is in itself a worthy feature of games, in that it transforms them into educational and creative tools. However, not all children will be as persistent as Jarish. What is needed is any easier way to cross the line between playing games and writing a computer program. Turkle speculates,

Will the player of the games of the future be in a more complex world than is offered by today's games, but still in a world that is created by someone else? Or will the player be the designer of his or her own game? In other words, will players continue to be “users” of someone else's program or will they be programmers in their own right? Will they be able to create new characters and change the rules of the game? [T84, p. 78]

Computer games have the potential to be incredible creative tools, especially if the designers can figure out a way for the rules to be flexible and the characters and microworlds to be defined by the imagination of the player.

While Turkle views computer games as positive media that could be improved by more interactivity with the surroundings, Eugene Provenzo sees the lack of this interaction as a largely negative feature of games. He notes that game programmers create highly limited microworlds that often have disturbing violent or sexist themes. A child that plays the game is forced to live with these themes, and has no way of changing them to fit her tastes. As he puts it, “children are trapped in microworlds created by computer programmers, which, although highly appealing, are ultimately limited in terms of the needs and interests of the children who play the games” [P91, p. 48]. He does not believe that this has to be the case with these games, but in the case of Nintendo (which is currently the most popular video game system), designers have not realized their potential to create games that allow for creative world-definition:
A child picking up a pencil and applying it to paper can potentially create an infinitely rich microworld—one limited only by the constraints of her imagination. In the electronic environment of video games—at least as they are currently designed and function—the possibilities of the microworlds that can be created are much more limited. In the case of the computer (and by extension, the video game), tremendous flexibility and the potential for exploration and self-definition can be built into any system. This is not the case with Nintendo. [P91, p. 48]

In comparison to many other forms of play, computer games often limit the creative potential of children and trap them into role identifications that they did not choose for themselves. This limitation is especially striking in terms of gender, where girls most often have no choice but to play male characters interacting with worlds defined by male programmers. If girls were able to help define their microworlds, then perhaps they would be more interested in playing the games. There are many ways that programmers can stimulate creative thought through their games. We will suggest only a few.

As Jarish envisioned, computer games would be more attractive if they allowed the players to change their features. While it would be impractical to release the source code for most games, especially since only a few people would be able to decipher it, it should be possible for most games to create an editor for modifying many of a game’s features. For example, in a game such as Super Mario Bros, there could be an editor that would let you edit text messages, draw new characters, edit the background scenery, and change the order of worlds you have to go through to reach the end. It would be especially nice if you could change the ending scene so that you are rescuing someone other than “the princess.”

One game that provided such an editor was The Pinball Construction Set that appeared in the 1980’s. The game was a computerized pinball game with a twist—the player was given an editor that let her create new pinball machines from key components (bumpers, flippers, etc.). The editor also included a layer that let the
player modify the physics of the board—how fast the ball would move, how much resistance the bumpers would give, and so on. The game was engaging and popular, proving that computer games do not have to provide rigid systems to be fun.

Another way that a computer game can encourage creativity is to not have the game be oriented around action at all, but around creation. While The Pinball Construction Set is an action oriented game with flexible rules and settings, games such as Maxis's SimCity are engaging primarily because of their focus on creation. In SimCity, the player is given a landmass and a certain amount of money, and is free to build any type of city she pleases with the tools provided. After zoning areas, building roads, setting taxes, and erecting police stations and schools, the city develops according to internal rules. A well-designed city may prosper, while a carelessly designed city may go bankrupt. These rules present internal challenges, while still letting the player choose the focus and appearance of the game.

Other games might provide a number of possible goals, and let the user choose which one to pursue. One such type of game is Multiple-User Dungeons (MUDs). MUDs resemble text adventure games such as Zork or Dungeons and Dragons, in that the player is associated with a character who develops strengths over the course of the game (usually long-term, such as weeks or months), and that character explores a world that contains challenges and adventures. There are two unique features of MUDs, though. MUDs are played over a network with dozens of other players, and in most MUDs players can eventually become MUD-designers, who program new areas of exploration in the world. These features allow for a number of possible goals, leading to a number of possible ways of playing and viewing a MUD. While many players aim to kill monsters, collect treasures, and gain strength, others join a MUD for social reasons. MUDs tend to be home to virtual bars, guilds, and social clubs, where players can meet, act out fantasies, team up for adventures, or even be married in MUD-land. One player wrote me that she
enjoys playing MUDs because of the social element—her goal in playing is, "mostly to talk to the friends I've met on the computer, but I also try to advance in level and explore the muds just for the fun of it... it's better than playing computer games alone." Finally, there is always a set of players who aim to become wizards (wizards design the MUD), so that they can contribute to the creative definition of the microworld. In allowing for so many different goals, MUDs give players the freedom to choose how they wish to play the game.

As we have seen with MUDs, networked and other multi-player games can encourage creativity since they add unexpected contributions to the game. In one-player games, the player generally deals with opponents played by the computer. These opponents usually behave according to predefined rules, so winning the game often involves memorizing these patterns of behavior. With real people playing opponents, however, new strategies have to be thought of constantly in order to deal with new types of behavior. One of those strategies may include cooperation, a welcome phenomena hardly seen in one-player games.

Computer games should not limit the user to a highly defined and highly predictable microworld. When at all possible, they should allow the players to conceive of new ways of viewing that world, new rules to follow, and new goals to strive for. As we have seen, computer games often have negative consequences, especially in relation to aggressive themes and gender roles. One way to begin to overcome those consequences is to allow the players to choose how they approach the game, so that some children will not avoid computer games (and even computers in general) only because of the limited types of fantasies they allow. The remainder of this thesis discusses two games I have written and how they encourage creative thought.
Figure 3.1. A creature chasing another, goalposts, and zone markings on the ground.
Figure 3.2. Two creatures, hole with chips in it, cracks, and goalposts in the distance.
Figure 3.3. Two creatures, a goalpost, a chip, and boundaries.
Figure 3.4. A creature carrying a chip, goalposts, another creature, and cracks.
Figure 4.1. A typical scene in Compose.
Chapter 3
Tagh

3.1 Introduction
Tagh is a three dimensional multiplayer game written for Silicon Graphics workstations. I designed its environment and rules so that the user will be able to enact creative rules and strategies. Tagh encourages creativity through loosely defined rules, the involvement of multiple players, and a visual environment that evokes a feeling of uncertainty and unpredictability. The idea for the game came from conversations with Drew Olbrich, and some of the basic ideas emerged from brainstorming sessions between the two of us. He also contributed some code and advice, which I will credit when discussed.

Tagh is based on the children’s game of “tag,” which is notable for its simplicity and infinite variations of play. The basic rule is that there is someone who is “it,” and that person’s goal is to touch someone else so that they become “it” instead. Therefore if you are not “it,” you would like to keep from being tagged; if you are “it,” you hope to tag someone else as soon as possible. The simplicity of this playground game has spawned a large number of variations, such as “freeze tag,” where you become frozen when tagged until someone unfreezes you; a variation where whatever limb is tagged is rendered useless for the rest of the game; and a version that resembles an ameba, in which all of those tagged link up together until “it” becomes so large that they can span the entire field and comb for the few lone survivors. The common feeling to all the variations is that the players do not want to be tagged, although there are often a few oddballs that enjoy the experience. The game of tag was appealing as a model because of its simplicity, and therefore it is easy to be creative in varying the rules and objectives.
Obviously, it would be very difficult to write a computer game that would allow the user to vary the rules as freely in the playground game. For instance, had the game writer not envisioned a “freeze” option it would be very difficult to add it without modifying the code. So instead, Tagh has another dimension of play that can be freely interpreted, allowing the users to improvise new rules. This additional aspect is playing chips, with few rules surrounding them. There are four chips per player, and any player but “it” is allowed to pick up one chip at a time. A chip can be dropped anywhere on the playing field, but if it is dropped into the deep pit in the center of the field, it cannot be picked up again by any player. When all of one player’s chips are in the pit, it is possible to restart the game. Thus there are many possible ways of interacting with the chips. For example, you might want to keep your chips out of the pit, put all the chips in the pit, hide other players’ chips, or ignore the chips altogether. The chips allow for creative variations on the basic rules of tag.

Another aspect of creativity through Tagh is in the possible strategies. The multiple player dimension increases the number of strategies. Unlike one-player games, where all of the opponents are pre-programmed to respond in a predictable way, the other players in Tagh are as unpredictable as any person can be. For example, another player’s strategy might include hiding in the distance, being confrontational and quick-paced, or teaming up with other players against another. Each of these strategies is likely to induce a different response from the remaining player. Therefore strategies must necessarily change from game to game, and mechanical playing, with a set sequence of moves for each game, is practically impossible.

Combining multiple players and flexible rules creates an interesting communication challenge: how do you communicate what rules you want to follow? If all of the players are in the same room, they might discuss how they want to play the game. In the absence of such direct communication, a player might try to persuade other players to play a certain set of rules through her actions. A player might decide to
ignore the given rules, or change her objective halfway through the game with the purpose of confusing the others. Tagh encourages the users to solve problems creatively through this communication challenge.

The character of the game is reflected in the scenery. When designing Tagh, I conceived of the rules and the scenery interdependently—a graphical change often involved a change in rules and objectives, and vice-versa. The noticable aspects of the scenery are a fog, which is both aesthetic and useful for hiding; color choices that are grayish and mix in with the fog; a pit, goal posts, and colored territory, which help orient the user; and pterodactyl-like creatures that represent the players. The scenery has random components, which adds to a nebulous and unpredictable feeling.

When I designed the game, I created four basic modules: graphics, user interface and movement, network, and sound. The rules are interspersed with the user interface code. Because they are interdependent, I wrote the graphics and user-interface code simultaneously. I implemented the network and sound last, because they are non-essential parts of the game. What follows is an outline of the implementation of Tagh, organized by the four basic modules. We start with some of the basic graphics concepts that we will need.

3.2 Key Terms and Concepts

This paper refers to some graphical techniques whose details are not central to the understanding of the game’s design. This section outlines these techniques.

When we compute lighting for a polygon, we compute a color that differs from the polygons’s original color according to the effects of light sources. In Tagh, there is one light source, and it is directional: we can represent it by a vector. Using this vector, we compute lighting by a dot product with the polygon’s normal vector [FvDFH90, 18.3.4].
Depth cueing simulates the atmospheric attenuation from an object to the viewer. The more distant an object is from the viewer, the more the object's color is blended with the depth-cue color [FvDFH90, 18.3.4]. Depth cueing creates the fog effect in Tagh.

Culling is a technique that determines whether a polygon is being blocked by other polygons. A back-facing polygon is one facing away from the viewer; it is characterized by a negative dot product between the polygon's surface normal and the vector formed from the viewer to any point on the polygon [FvDFH90, 15.2.4]. Culling is used for drawing the hole in Tagh’s playing field—all back-facing polygons are drawn in the color of the groundplane to give the appearance that the groundplane is blocking part of the front-facing polygons from view.

There are two ways of drawing a polygon: wireframe and shaded. Polygons drawn in wireframe have only the border lines drawn, as opposed to shaded polygons. The goalposts in Tagh are sometimes drawn in wireframe so that the viewer will be able to see through them.

Finally, the groundplane is the ground level of the playing field. In Tagh, the ground is flat, except for the hole in the center of the field, and all of the stationary objects rest on the ground or in the hole.

3.3 Graphics

Overall Design

The graphical environment of a game is along with the user interface the most crucial part of its design. The graphical environment sets the user’s mood, probably much more than the game’s specific rules. There may be many games that share a common set of rules and objectives, but each can be perceived as unique and enjoyable if it has an imaginative graphical environment to distinguish it from the others.

My goal was to create a surreal setting, utterly different from a playground, that would evoke a feeling of spookiness and uncertainty. I wanted to give the user a fluid
environment in which her imagination is free to run wild, as opposed to a clear and static environment that is either so ordinary or so detailed that the user does not envision anything that is not drawn on the screen. The vehicles I used to evoke this feeling were

- a fog that makes details hazy and allows distant objects to slip in and out of view,
- a choice of colors that mix in well with the fog,
- no distinct groundplane—instead I created the illusion of a groundplane through the orientation of objects to it,
- randomness in scenery generation, which means that every time the game is played the scenery is slightly different (since the point coordinates for many of the objects change between games).

The fog, one of the game’s most distinctive features, is implemented with depth cueing. The function for computing fog, originally created by Drew Olbrich, takes a vector from an object to the eye and the color value of that object, and returns a new color value grayed according to distance. This function is called every time an object is drawn (in each iteration of the event loop), and adjusts its color appropriately.

For examples of representative scenes from Tagh, see Figures 3.1-3.4 at the end of this chapter.

The Playing Field

The field contains a hole, cracks emanating from the hole, and a border around the edge of the world. The hole is based on two six-sided polygons, one at ground level and one three units below ground level. The two polygons are then connected by triangles representing the side of the hole, and lighting (which remains constant throughout the game since the hole doesn’t move) is calculated for each triangle. The vertices of the polygons vary randomly in their distance from the center, giving the hole the appearance of natural topography. When drawn, culling is used through a simple
vector calculation to give the appearance that the groundplane is blocking certain triangles from view.

There are six cracks that spread out from the hole. They also have a degree of randomness in their calculation, and decrease in thickness as they reach the edge of the world. The cracks are not only cosmetic, they also serve to divide the field into six sections and provide reference points for orientation. In addition, between every other set of cracks is a four sided-polygon drawn in the color of the player whose goals lie on that sector.

A bright yellow four-sided fence represents the edge of the playing field. This fence is just four rectangles drawn in wireframe. It turned out to be very important to give the field an edge, lest the player wanders off into the haze, never to be seen again. When a creature hits an edge of the playing field, its movement stops in that direction. It also seemed wise to let the users know when they had reached the edge, so I added a pinball effect that causes the whole wall to flash yellow when a creature collides with the edge.

The Goals
The goalposts were one of those miraculous creations that happen when you have only a vague idea of what you want and try something quickly, getting something totally different from what you imagined, but better. I envisioned a pair of monolithic structures that bend ferociously towards each other. To create these I quickly created a few lines, subdivided them at randomly generated points, and then connected them together with triangles. The first attempt was massively buggy—the triangles folded in on themselves and defied all laws of physics. But they looked even better than my original idea this way, so I based their color on the groundplane (to make them seem especially ghostly), lighted them, and left them at that. There are three sets of goals, spaced evenly around the edge of the world. A spark, the same color as the user owning
those posts, flashes between the tips of the goals. I added intersection calculations later
so that creatures bounce off the goals instead of flying right through them. The goals are
drawn in wireframe whenever they intervene between a player’s camera and her
creature.

The Chips

The chips are very simple—six triangles drawn around a base polygon (generated using
the same random code as in the holes) that meet at one point on top. There are four
chips per player, colored the same color as the player, and as the player enters the game
they are drawn in a row underneath the respective goal.

The Creatures

To draw animated and lifelike creatures without using a large number of polygons, I
eliminated the body (thinking it would be best left to the imagination). Each creature,
then, is made up of several unconnected components: wings, tail, beak, and eyes. For
efficiency, I didn’t bother calculating lighting, since the creatures’ movements would
require constant recalculation.

3.4 Movement and User Interface Design

Interaction with Tagh occurs in several ways—navigating the creature through the
playing field, performing key actions (such as picking up chips or tagging other
players), and catching cues from creatures about their current state of action. When
designing the game, I decided that I wanted interaction to be as simple and accessible as
possible. Therefore, all interaction occurs through mouse movement and one mouse
button. The low number of devices lead to certain limitations—for instance, the creature
is basically restricted to one plane of movement, since it would be extrememly
confusing to fully navigate three dimensions using only a mouse. The creature, then, only leaves its plane of flight when swooping.

All mouse input from the user is handled in the main display loop, which processes events. A change in mouse position alters certain global variables that are referred to whenever the game is updated. Button presses result in the swoop() function being called, which either causes the creature to swoop and try to pick up a chip or attempt to tag another player if it is "it." The only other user event is pressing the escape key, which terminates the game. Speed is regulated by a timer, so that there are no handicaps due to machine speed.

Each time creatures_move() is called, both wing flapping and movement are updated. The wings flap constantly when the creature is moving forward, and remain in a high position when it turns. The speed of the flapping is regulated by the speed of the creature. Graphically, several rotations along the y and z axes control flapping. The rotations are specified by the program and implemented with library routines.

The creature moves forward along its current orientation at a time given by its current speed. However, movement is slightly faster when the wings are moving downwards than when they are moving up or stationary, for a more realistic effect. creatures_move() also updates swooping action when necessary, stops the creature movement in any direction that would move it off the edge of the field, and calls various intersection routines to check if the creature is about to collide with any goalposts. The intersection routines use various algorithms\(^1\) to check if any part of the creature is intersecting the triangles that make up the goalposts. If it is about to fly through a goalpost, the intersection code guarantees that the creature will instead

---

\(^1\) Algorithms used are: determining whether consecutive line segments turn left or right [CLR90, p. 888], determining whether two line segments intersect [CLR90, pp. 889–890], determining whether a point is inside a triangle (derived from above turn test), determining whether a triangle and a line intersect (derived from above line intersection and point in triangle tests), and determining whether a point is inside a convex polygon [PS85, p. 43].
deflect off of the post. The intersection code also checks if any goalposts are blocking the player’s vision and indicates that these goalposts be drawn in wireframe if they are.

Movement is directed by the mouse. The interface is similar to that of a joystick, in that mouse movement is interpreted in terms of distance and direction from a central point on the screen. There is a black cursor indicating the current position of the mouse, and the cursor’s coordinates are translated into game coordinates. If the cursor is in the center of the screen, the user’s creature is in a neutral state, flying slow and straight. As the cursor moves further and further from the center, the creature’s speed increases, being directly proportional to the distance from the cursor to the center point.

The plane of mouse movement is translated directly down to the $x$-$y$ plane, the same plane of creature movement. In appearance, it may seem that we are moving the cursor from a vertical to a horizontal orientation. Also, it may be confusing to the user that the cursor, when pointing up, is really pointing forward, or in the direction of movement. Because of this confusion, it would have been more intuitive to have a joystick as an input device instead of a mouse. Joystick movements are directional and don’t need a point of reference, so it would be possible to eliminate the cursor (which is unnaturally a two dimensional object imposed on a three dimensional space). A mouse, on the other hand, is more suited as a pointing device, which leads to confusion when the objects we are pointing at are in a three dimensional space.

To manipulate the creature, then, the player points the cursor at the spot where she wishes the creature to move to (on the $x$-$y$ plane). Since the creature is always in the center of the screen, and is always oriented so as to face directly away from the user, the same cursor movement will always turn the creature in the same way. The creature responds to the cursor movement as follows. It first tries to look at the cursor, rotating its eyeballs to see it. If the cursor is too far to the left or right, it will then rotate its head. Similarly, if the creature has to swing its head too far to see the cursor, it will rotate its body in the right direction (tilting a little around the $y$ axis as it does). There are two
natural body rotations, so that the creature can make both sharp turns and slow, leisurely turns. When the creature turns, the camera turns along with it so that the player is always looking at her creature's back.

The choice of an interface was very difficult and took a number of trial attempts. I chose the interface described above only after experimenting with several others. For example, one attempt discarded the cursor and judged each mouse movement as if it was from the center of the screen, moving the creature according to the direction and intensity of that movement. In effect, the mouse was imitating a joystick that snaps back into place after each movement. I didn't like that method because it required that the player constantly move the mouse to keep the creature circling a point. Since circling seemed a natural activity for a flying creature, it seemed that this should be easier to accomplish. Moreover, this method tended to leave the player disoriented without a cursor as a reference.

Another attempt was in many ways like the method I finally used, except that the camera constantly faced in the same direction while the creature rotated. This worked well except for two problems: circling was more difficult than in the final method, and the player was only able to look at the scene from one direction, leaving many possible views of the scene unappreciated. Once I started rotating the camera, I noticed that the scene was much more pleasing and complete when looked at from a number of different angles. With the final interface, circling, the closest activity to staying in one place, is accomplished by simply leaving the cursor in the lower left or right hand corners of the screen.

I chose Tagh's interface because of the advantages in circling and multiple viewpoints. Another advantage to my method was that, in keeping the camera constantly a few units behind the creature, it was constantly more difficult to see the parts of the playing field behind the creature than in front of it, which seemed logical. The disadvantages to my method were the translation from one plane to another (which
could be solved with a joystick), the confusion of not having a constant direction of reference for the scene, and the necessity of orienting everything around the center of the screen, which seems counter-intuitive to some people.

Corresponding to these weaknesses, I found that the three problems people tended to have when first playing Tagh were not understanding what it meant to point the cursor at a point on the x-y plane, getting lost, and tending to keep the cursor low on the screen (which means that the creature was always turning). It was also not immediately clear to many users that distance from the center corresponded to speed. Speed has traditionally been controlled by the keyboard in 3 dimensional games, which was my other option, but this method does not seem to be any more obvious than mine. My method also has the advantage of using as few devices as possible. With only a mouse, which is a two-dimensional device not designed for gaming, nearly every application that involves manipulating in three dimensions has a noticable learning time. While Tagh’s interface may not be immediately intuitive, most players that I talked to remarked that it was very easy to use once they had learned the basics.

3.5 Network

The framework for the network code was designed by Drew Olbrich, and I added the specific data structures for Tagh. His design is based on a client-server model. One machine acts as a server, and is responsible for sending information about the current state of the game to the client machines. The clients provide graphics, sound, and user interface for each player, and send any game updates to the server. Messages are sent in packets consisting of an opcode and a structure containing game data. The opcode represents either a short message, such as a request to join the game or a response that the game is full, or information about which data structure is being sent.

There are three types of data structures that can be sent in a packet from a client to the server: player data, chip data, and an index to “it.” A client sends new player data
with each iteration of the event loop, but only sends chip data when the player moves a chip, and sends updates to the "it" index when a player is tagged. Each data structure contains only essential information. For example, the player data structure contains information about creature position and creature status, but does not contain information about wing position, since that can be calculated locally.

When the server receives client packets, it updates the "world" data structure, which contains all the essential information about players and chips. It then broadcasts a packet containing this world data structure to all the clients. A client receives one server packet for each iteration of the event loop, and it uses this packet to update its local variables. The only other type of server packet is one containing an opcode message.

If a group of people wish to play Tagh, they need to decide which machine will act as a server. Each player then indicates that machine on the command line, such as "tagh servername." The machine designated as a server then forks off a separate server process, and all of the machines run a client process.

### 3.6 Sound

All of the sounds in Tagh are predefined in AIFF (Audio Interchange File Format) sound files. The sounds are played with certain game events by using a modified version of the code for playaiiff (a standard libaudio program). Sound is played on one channel, so only one sound file can play at a time.

Events that are associated with sounds are collisions, attempts to tag another player, tagging another player, becoming "it," picking up and dropping chips, wing flapping, and turning. Sounds associated with relatively rare events, such as tagging another player, are loud and distinct, while sounds that are played frequently, such as wing flapping and collisions with goalposts, are softer and have less character. The sounds for Tagh were borrowed from a variety of sources: other games and
applications, sound effect cd’s, and recordings. It was my goal that the sounds in Tagh add to the character of the game without distracting or annoying the user.

3.7 Evaluation

To evaluate Tagh, I arranged three formal test sessions, with three people playing in each one. Each session ran for about an hour, and consisted of a brief (5 minute) introduction to the game and interface, a playing session, a written evaluation, and discussion. In each session, two people played in one room and one person in another. In the first session, two females played in one room and a male played in the other; in the second, a male and a female played in one room and a female in the other; and in the final session, two males played in one room and a female in the other. I observed the game by watching the people as they played it, and also by watching a program I wrote which monitors the game and displays its progress on a map.

The questionnaire was intended to help answer several questions: whether the game was enjoyable, whether the game inspired creative thinking, whether the game was well-designed, what improvements could be made to the game, and how did perceptions of the game correspond to self-perceptions of the people who played it (to see a copy of the questionnaire, refer to the Appendix at the end of this thesis). The questionnaire was divided into three sections: personal information (including gender, areas of interest, and perceptions of computer games), ratings of a set of statements on a 1-5 scale, and free discussion about the positive and negative points of the game. The second section was the least straightforward because it covered a wide variety of topics. There were several statements related to user interface design, graphics and sound design, and creative potential. The remainder of the statements were formulated to determine whether Tagh was enjoyable and fun. These statements were derived from Thomas Malone’s article, “What Makes Computer Games Fun?” in which he lists components of an enjoyable game. Among these are a clear and meaningful goal (or an
environment which makes it easy to determine such a goal), an uncertain outcome, appealing fantasies, and well-designed audio and visual effects.

I received mostly positive responses to the questions about the user interface and graphics features of the games. Most players found the game to be visually appealing, fun to play, and relatively easy to learn. More variation was seen in the perception of concepts behind the game. Everyone agreed that the game had a goal, but some players disagreed that “Tagh had a clear goal,” while agreeing that “Tagh didn’t have a clear goal, but it was easy to determine one.” Similarly, while most players agreed or were neutral about the statement, “Tagh is a very competitive game,” they were very divided about the statement, “Tagh encourages teamwork and cooperation,” with answers ranging from “agree strongly” (with “yes! yes! yes!” written in the margin) to “disagree.” There was also contention as to whether Tagh is a violent game, with answers in all five points on the scale. All but one of the players agreed with at least one of the statements, “Tagh encourages creative strategies,” or “Tagh encourages creative thinking.”

In the questionnaires, players listed as the best features of Tagh: graphics, “instinctive controls,” “fun to scream to,” “encouraged cooperation between players,” the multiple player dimension, that it is “free form,” the fog, and being “interactive in a nice competitive way.” Negative features listed were that it was disorienting or confusing; that the controls were too sensitive, or, in another case, that there was too slow a response time; and that it was difficult to pick up a specific chip. The most common suggestions for improvement were to make a larger playing field with more objects to hide behind, or, for more disoriented players, to include a map. Other suggestions were to change the input devise to a trackball or joystick, to increase the personality of the birds, and to add more players.

In the discussions after the play sessions, I tried to ask everyone what they felt the goal of the game to be. I received some very interesting and varied answers. Several
players felt that the goal was to get their chips in the hole. On the other hand, many players weren’t as interested in that aspect of the game, and preferred instead to subvert the game through interactions with other players. These people enjoyed stealing other players’ chips and hiding them, staying “it” to scare the other players, or working with another player to confuse the third. The two women that were in the same room together perceived the game as encouraging teamwork, and spent the entire session teaming up together to confuse the male in the other room. Two players perceived the goal as increasing personal skill—one tried to “become one with the bird” and another practiced stealing chips off of other creatures’ backs. Finally, one female player, Dominique, (who played in a room by herself against two males), described an interesting way of perceiving the game:

I thought the point of the game was to get all the chips in the pit. I was disappointed when you stopped the game before we were done... I thought we were supposed to work together to build a home. Like, the chips were twigs or something in a nest...

What was especially interesting about this comment was that she was playing against two males with competitive perceptions of the game. One was frustrated that “it’s obvious that there’s always a winner and a loser—I feel like the playground game where some kid is always stuck being it.” The other player, Matt, was enjoying stealing Dominique’s chips and hiding them. She saw him pick up her chips, and assumed he was trying to help her. I found it interesting that in one game three players could have entirely different perceptions of a goal and different fantasies about the scenario behind the game.

I found the responses to be very positive. Overall, Tagh met most of the criteria Thomas Malone suggests for making enjoyable games, and players tended to agree that the game was fun. While I received negative comments and suggestions, most of these, if responded to, would have alienated other players. For example, adding a map to the
game or making the world less confusing would have alienated those players who felt that the world should be larger and include more surprises. Since there were no unanimous feelings about how the game should be improved, I felt I reached a happy medium between different gaming preferences.

The central question, whether Tagh encourages creativity, was answered to the affirmative through these sessions. In addition to agreeing with the statements concerning creative value, many of the players exhibited creative thinking and strategies while playing. Most significant was the perception of goals. We may conclude that Tagh provides a positive environment for experimenting with a variety of goals and rule systems, since the players perceived a wide varieties of goals in the game. Also, players discussed creative fantasies, such as Dominique's fantasy about building a home. As another example, in one conversation one player compared the bird to Groucho Marx, and another said that they looked like eyebrows. Only one player felt limited to the playground fantasy on which Tagh was based. While players indicated a wide variety of game preferences—some enjoying only competitive and fast paced games, a few expressing dislike for violent action games, and others preferring simulation games and non-goal oriented games—nearly all of the players were able to find a way of playing Tagh that was pleasing to them.

One other result of the playing sessions was gender-related. While there is not enough data to draw any firm conclusions, it is interesting to note that the only real cooperation between two players occurred when two females played together in the same room, and that Dominique (who played alone) expressed a strong desire to play the game cooperatively. On the other hand, the male who felt particularly frustrated with the game said that he disliked the competitive nature of Tagh, while not making any attempts to encourage cooperation with the other male in the room (in fact, there was hardly any communication between them even though I encouraged them to talk). Females seemed particularly attracted to the game—one wrote me several e-mail
messages afterwards to tell me how much she enjoyed it, another told me that she wished she could play it again, while another female, who only observed one of the sessions, repeatedly described the game positively to others. In comparison to the males who have played or seen Tagh, females seem to be much more enthusiastic. It would be interesting to investigate whether this attraction is because Tagh allows for a cooperative strategy, one that is inherently more attractive to women, or if there are other factors involved.

Overall, it appears that Tagh has succeeded as an enjoyable game which encourages creative thought. Its flexible rules have resulted in positive feedback, and the graphics and sound have proven to be impressive.
Figure 3.1. A creature chasing another, goalposts, and zone markings on the ground.
Figure 3.2. Two creatures, hole with chips in it, cracks, and goalposts in the distance.
Figure 3.3. Two creatures, a goalpost, a chip, and boundaries.
Figure 3.4. A creature carrying a chip, goalposts, another creature, and cracks.
Chapter 4
Compose

4.1 Introduction
Compose is a one-player game involving painting and music composition on Silicon Graphics workstations. It is modeled after a similar program displayed in San Francisco Exploratorium. When I played with that program, I found it so fascinating that I was determined to reproduce it for my own use. Compose is a painting program with a twist. The player is given a palette of eight colors and three grays, and is free to paint with these as she pleases on a canvas. Each color represents a note on the C-scale, and each gray represents a type of directional change. While the player is painting, four bugs, represented as boxes, move across the screen. Each bug represents an instrument, and when a bug moves over a color it plays the note that the color represents on its instrument. When an instrument hits a gray, it either changes direction by turning right or left or reverses its direction. In this way a painting is also a composition, since its colors represent rules for instruments to follow.

Compose is structured like a painting tool, but it is designed as a game. While it has no predefined goal or fantasy, it also has no practical purpose. A player, when using Compose, should find the structure of the program to be appealing and define her own internal goals. In this way, Compose resembles simulation-style games such as SimCity, in that it presents a set of rules and lets the user decide how to approach them.

Compose encourages creativity in more obvious ways than Tagh. In Compose, it is next to impossible to live in somebody else’s microworld, since the game starts with only a blank canvass and a palette. The player has little choice but to create an original painting and follow her own ideas about what she wishes to
accomplish. Therefore, Compose encourages the player to create her own microworld that operates within the few rules that are predefined.

4.2 Graphics
Compose has very simple graphics. A window contains a canvas on which to paint and a palette full of colors. A two-dimensional array represents the painting, with each index holding a color or a gray. Each colored or grayed index in this array is drawn as a square, using SGI's graphics library. The drawing is done in an orthogonal coordinate system. If we wish to change the resolution of the painting, we change the coordinate system so that it displays a larger or smaller section of the painting in the window. The palette is drawn as a series of framed boxes on the bottom of the window, and instruments are drawn as wireframe colored boxes on the painting.

For an example of the graphics in Compose, refer to Figure 4.1 at the end of this chapter.

4.3 User Interface
To paint a picture in Compose, the player uses the mouse to select a color from the palette by clicking on its box on the bottom of the window, and then fills in boxes on the canvas by clicking where she wants to paint. To aid the player, the mouse cursor is displayed as a colored arrow, the color indicating the current color chosen from the palette. This arrow adds color one box at a time. There are no other painting tools, since other tools would not help much in directing the composition.

In the future, I would like to add a menu bar to Compose, so that the player can open and save paintings, perhaps import other graphics files, and customize the settings. In particular, I will let the player change the painting's resolution, the tempo, and the instruments associated with each bug. I have structured the program
so that these are easily modifiable. In particular, it should be easy for the user to add new instruments by adding a sound file for each note to the data directory and indicating this instrument on the control panel.

4.4 Sound

The instruments in Compose are dictated by a clock, which maintains a constant tempo (tempo is a global variable). With each beat, the instruments move forward in the direction they are facing, possibly change direction, and play a note if they move over a color. Notes are represented by AIFF (Audio Interchange File Format) files. Most of these were collected by sampling sounds from Dartmouth Music Department’s Synclavier. The sounds currently available include, among others, an english horn, a bass guitar, and an Indian tabla drum.

Compose uses SGI sound hardware to its full potential, since it writes to multiple sound channels. (SGI allows users to write to up to seven sound channels at once, so that up to seven sounds can be played simultaneously.) Data for each sound file are loaded when the instruments are initialized, using SGI’s audiofile library. To play a sound, the program writes a pre-loaded buffer to one of the sound channels using library routines. Each instrument writes to a separate channel.

4.5 Evaluation

I did not find it necessary to arrange formal test sessions for Compose, since the game is very straightforward and obvious in its creative potential. I did, however, let several people play the game, and received positive feedback from these sessions. The players found the game to be enjoyable and amusing, and appreciated its creative potential.

The players demonstrated several ways of approaching the game. A couple of people, for instance, found its musical potential attractive, and discussed ways of
playing chords and musical scores. On the other hand, another player was more
intrigued with the movement patterns of the bugs, and enjoyed experimenting with
that. He also remarked that the game is positive because it forces the player to
construct his or her own goals and strategies.

I am very intrigued with the possibilities for Compose. I have several ideas
for options to add to the game. One would be to import other types of graphics files
into the game. It would be amusing, for instance, to see how a Monet “sounds.” Or,
let the painting be defined by cellular automata—a dynamic graphics system
operating under a set of rules. In this way, both the painting and the music could be
dynamic. A few people have expressed interest in porting the game to other
platforms or adding code to fit their fantasies. Compose can encourage creativity
among both players and programmers.

With Compose, we have seen how a game can be enjoyable while providing a
loose framework and no set goals or fantasies. Such a game is dependent on creative
thought, unlike Tagh, which encourages creativity in more subtle ways. Compose is
fun precisely because it involves creation.
Figure 4.1. A typical scene in Compose.
Conclusion

In this thesis, we have shown how computer games can help to stimulate creative thought and problem solving. We discussed how computer games influence society, and noted negative tendencies in today's computer game market in relation to gender and aggression. However, we suggested that by providing players the freedom to customize the fantasies, rules, and goals in computer games, we can begin to change these games into something more positive. We have introduced two games which attempt to provide players with this freedom, and discussed how they have succeeded in stimulating creative thought.

One of my main concerns, after thinking about the issues presented in this thesis, was how much game designers could succeed in transforming games into a more positive social medium while still producing a marketable product. It may be that most computer games today are violent and male-centered precisely because these are the ingredients that make a computer game fun. Perhaps a company producing games that didn't include these ingredients would go bankrupt. However, I think that we can see that computer games do not have to operate within such a limited sphere by looking at the success of games such as SimCity and Tetris. While these games may appeal to an entirely different audience than those currently making the game industry wealthy, the point is that there are other people who could be convinced to love computer games, especially females, and this new market can only be tapped through concepts in computer gaming that are different from those currently dominating the industry.

Having shown this, I tend to wonder how I, as a single programmer without much experience or prestige, will be able to hold to the standards I have suggested for computer games. What would I do, for instance, if I was working for a company that asked me to create a game that compromised my principles? This is a dilemma
that quite possibly affects many talented game writers with creative new ideas. I would hope that these programmers can find places with companies that are wise enough to experiment with new ideas.

In the future, I hope to experiment more with user-definable games. As I have described, I will continue to modify and enhance Compose, and may also add new features to Tagh. In addition, I plan on playing with other ideas for game flexibility, especially simple editors and languages that a player can use to customize the features of any game. This thesis has provided me with a starting point for thinking about the issues in designing computer games.
Appendix: Questionnaire for Tagh Players

Name:

Gender:

What is your major or strongest area(s) of interest?

Please rank your interest in computer/video games on a 1–5 scale:
(1 = High interest, 5 = Almost no interest)

Please rank your experience with computer/video games on a 1-5 scale:
(1 = Extensive Experience, 5 = Almost no experience)

What types of computer/video games do you enjoy the most?

What types of computer/video games do you enjoy the least?

For the following statements, please indicate your level of agreement on a 1-5 scale:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Agree Strongly</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Disagree Strongly</th>
</tr>
</thead>
<tbody>
<tr>
<td>The creature was easy to manipulate.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>I was very disoriented.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>I didn't know what to do.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh had a clear goal.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh didn't have a clear goal, but it was easy to determine one.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh had an emotionally appealing fantasy world.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh included surprises.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh included constructive feedback.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Statement</td>
<td>Agree Strongly</td>
<td>Agree</td>
<td>Neutral</td>
<td>Disagree</td>
<td>Disagree Strongly</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>----------------</td>
<td>-------</td>
<td>---------</td>
<td>----------</td>
<td>-------------------</td>
</tr>
<tr>
<td>The graphical effects in Tagh were visually appealing.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>(If applicable) The audio effects in Tagh were appealing.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>It was easy to interact with other players.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh is a very competitive game.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh encourages teamwork and cooperation.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh is a violent game.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh is fast paced.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh is action-packed.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh was fun to play.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh is boring.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh is addictive.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh encourages creative strategies.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Tagh encourages creative thinking.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Please comment on any of the above statements.
What were the most positive features of Tagh?

What were the most negative features of Tagh?

Do you have any comments on how the game could be improved?

Add any additional comments here.
References


