

MEMORABLE GAME DESIGN

Thesis Supervisor:

Rodion Podorozhny, Ph.D.
Department of Computer Science

Approved:

Heather C. Galloway, Ph.D.
Dean, Honors College

MEMORABLE GAME DESIGN

HONORS THESIS

Presented to the Honors Committee of
Texas State University – San Marcos
in Partial Fulfillment
of the Requirements

for Graduation in the Honors Program

By

Mark Anthony S. Sison

San Marcos, Texas

May 2014

MEMORABLE GAME DESIGN

Mark Anthony S. Sison

May 2014



Abstract

This paper describes the process of a project I wrote in C#. The project is an Action Platformer computer game that runs on the Windows platform. The purpose in making this game is to make something fun and memorable to the player. Many aspects of different games across all generations were studied in order to find out what makes games memorable and exciting experiences. I wanted the game to be able to draw in a player through child-like wonder of a world filled with things that were new, unknown, and unsolved. This is attempted by telling the story indirectly through the game's setting, where the player starts out knowing little about the world, slowly piecing together the story through experiencing different events and finding clues built into the setting. In addition, the game's music is meant to be cohesive and similar to other tracks within the game, burning the melodies into the player's mind for years to come.

Introduction

I have been playing video games as far back as I can remember, and the video game industry started making games even before I was born. A video game is a form of entertainment that emerged with the rise of personal computers. While it's still a comparatively young field, the creation of games has evolved rapidly since I first started playing and since the days of the first video games like *Pong* (Atari Games) and *Space Wars* (Cinematronics), for better and for worse.

No matter how old, some games will always be remembered by those that played them. These games offered something new and memorable, an experience the player will never forget because they've never seen anything quite like it before. Few can forget the feeling of climbing a towering enemy, frantically scrambling for its weak spot as you try to hold on. Few forget being placed out into an open field with no idea of what to do, with a secluded cave on your left and being surrounded by paths leading to the North, West, and East. No one forgets smacking heads into blocks to make mushrooms slide out, making you double in size upon eating them. These kinds of memorable and fresh experiences make games very enjoyable for the player, whether the player themselves notice or not.

The first time I had the thought of creating a game was when I was wondering how you could control the character on screen. I couldn't comprehend how the press of a button moved the character, and I originally thought that the developers of the game created millions of pictures, one for each possible screen that could be displayed depending on the player's button inputs. That seemed unfeasible even to my young mind because it would take forever for anyone to draw that many pictures, so there had to be

another way for these buttons to control the video game. This was a huge factor in me becoming a programmer; I wanted to know how video games were even possible.

As I grew, I played more games, finding many favorites through the years. This was an even bigger factor for me wanting to learn how to program; I not only wanted to understand how the games I enjoyed worked, but I also wanted to create games that were as fun as the ones I played. I wanted to make games that were innovative and would always be remembered, just like all of the well-known games that have been created throughout history. As I went through college learning how to program, I never had the time to devote to solely work on a video game. When I learned that the Honors Thesis could be an artistic work, I pursued that chance to have time to finally learn how to make a game.

My project is a game with large levels, scenery that tells a story, characters that interact with the player, and ways for the player to interact with the world. The main aspect of the game right now is just jumping around the world and exploring, trying to find secrets and figure out what's going on. With different ways to traverse the areas and overcome obstacles, I feel like my project does a decent job at being something memorable to the person playing it.

Overview of Creating my Game

I originally wanted to create my game from scratch. Over my tinkering before starting, I noticed that this would be infeasible for the length of time I had, so I began to look for game engines and libraries that are used by other aspiring game developers. Game engines and libraries handle the low level implementation that every application needs, like drawing things to the screen, playing sound, and running the main game loop that handles time. Using an engine instead of writing the game from scratch allows people to program more of a game instead of spending time reinventing the wheel, so I decided to use the Unity library to let me spend more time on my game.

Getting started on the game was one of the bigger hurdles I had to overcome. First and foremost, I had to learn the quirks of the C# programming language and Unity's API, an application programming interface that lets a programmer connect their own code to the library's code. Consequently, I also had to work around and deal with Unity's limitations. Using another developer's library saved me time in terms of what I had to develop myself, but using it means that the game will be limited by their implementations and be hindered by any bugs they have in their own code, which turned out to be enough that I had to change the way some things work until they release a fixed version. One obstacle I faced was how they implemented their collision detection and physics, so I spent more time creating workarounds to get it working how I wanted it to.

The next step was designing the game. I spent time on designing the main character, how the enemies would look, how the levels would be structured, along with anything else I could quickly sketch out onto loose leaf paper to get an idea of what I was making. I jumped back and forth between the early design process while making a

working prototype that I could test the game in. Research in how to program certain mechanics such as artificial intelligence had to be done, and even trivial things like figuring out when it was okay to let the player jump took a lot of time to solve and program. Slowly adding in more features like these, more music, and more images, the game started to become what it is today.

A video game consists of music, art, story, and gameplay, a program to handle the game logic and tie the music and art together to the player's input. The design of all of these aspects comes from the developer's tastes, and mine lie in video games with a sense of adventure and fast paced action. A big part of games with a sense of adventure is cohesion between all the different parts of the world. The music syncs together nicely, the character designs match up, and the stages that you play on eventually come together to make different parts of the game feel like they belong with each other. Games need to shoot unexpected elements at the player, but these elements need to mesh with each other in some way in order to not feel out of place. This is one of the biggest challenges in making a game, but it's also one of the most rewarding.

Gameplay

A sense of adventure adds to the immersion and how memorable a game is. Even in linear games, a sense of adventure can disguise the “Point A to Point B” gameplay, as it allows the player to do something else without pressuring them to complete the main objective. With this in mind, I tried to create worlds with many different secrets and small areas sprinkled in between the player’s current location and their immediate goal. Level design that allows players to complete the same task in many different ways helps immerse the player into the game, and level design that differs from the norm help it stick out in the player’s mind after they’re done playing. In designing my own levels, I started out with a rough sketch of the levels and what the level would contain. Later on, I worked from these sketches to slowly build prototypes of the levels to make sure they’re enjoyable to play (Figure 1).

In *Super Mario Sunshine* (Nintendo), you take the role of Mario, a plumber trying to take a vacation when his princess is taken from him. He then spends his time on the vacation island trying to rescue her by collecting stars, doing different things like saving a mayor who is surrounded by lava, or by trying to roll the largest watermelon across the beach without breaking it. *Super Mario Galaxy* (Nintendo) is the sequel to *Super Mario Sunshine*, and it’s similar in that you play as Mario trying to save the princess by collecting stars. As much as they sound the same, the games are vastly different in that they tackle the same problem of collecting stars in different ways. Both games feature you using your jumping skills to go from Point A to Point B to collect a star, but one is more open ended than the other.

Super Mario Sunshine's levels are large, open, and are filled with many different things to do. There's no set path to completing a course, and the levels are populated enough that you could just fool around if you felt like it. In contrast, *Super Mario Galaxy*'s levels do little to disguise the linearity of each course, and there isn't much to do in the game other than what the level designers want you to do. A level of freedom helps me remember a game not only for what the developers designed the game to be, but also for what I wanted to game to be. My game's beginning leaves the player out in an unfamiliar place with no real direction to follow aside from their own personal desire for exploration (Figure 2). The player can take any path that they want to get to the door at the end of the stage, and the thought process that they go through to decide what to do will help them remember that particular scene. Most of the levels are fairly open with many secrets hidden throughout, and the player isn't normally pressured to focus on an objective. There are many ways to get from the beginning of an area to the end of the area, and people playing the game will have different experiences with what they encounter to finish the same level.

Super Mario Sunshine also contains levels that are at their core are exactly the same as other levels. One level has you spraying water to clean things, another level has you spraying water to clean things, and another has you spraying water to clean things. While this sounds monotonous, they are actually portrayed differently: one level has you saving a village from flaming goop with your water, another has you taming a giant man-eating fish by cleaning its teeth while trying to stay alive, and another has you cleaning up graffiti on fruit stalls while trying to catch the criminals responsible for the act. What you're actually doing may be essentially the same, but the way that they're represented

keeps things fresh while still maintaining a cohesive theme: cleaning things while you're on vacation. Offering fresh looks at the same problems allow for familiarity while keeping the player on his or her toes.

An easy way to achieve this is through unique level design. Many levels and worlds in games contain the same type of objects and give the player similar mechanics. Good level design disguises this by giving the same objects and actions different contexts that appear different to the player. The beginning stage of my game showed this; the stage is simply composed of blocks that the player can collide with, but the important part is how they're placed so that players find realizations on what they're supposed to do or where they're supposed to go. Another example in my game is the use of moving platforms, objects that move from point to point in the game's world that allow the player and other entities to ride on top of them. Moving platforms themselves are nothing new to video games, but riding a large bubble from the ocean's floor isn't very common (Figure 3). It's technically not very innovative, but applying an old concept to a new idea is another way to make the game feel new to the player.

One gameplay aspect I added a twist to is the enemy's behavior. In most video games, enemies are normally always hostile. I find enemies to be representations of the world's creatures, so I made some of my enemies start out being friendly unless attacked. It's nice to be able to walk around next to your foe and just watch them in their natural habitat instead of trying to kill each other immediately; it's a change of pace I placed in order to mix well with more peaceful areas that aren't supposed to present any feelings of immediate danger (Figure 4). I implemented the enemy behavior in a finite state machine, where enemies have states like an idle state for walking around, or a hostile state for

hunting the player. These states move to other states and change their behavior based on different inputs, like being able to see the player or if the player attacked them.

Most importantly as far as gameplay goes, the developer needs to introduce mechanics that the player might have never seen before. New gameplay aspects and mechanics are difficult to come up with, but doing so causes that game to be known as the game that did it first. The innovative gameplay mechanic is omitted from the copy of my thesis because I still plan to add onto my game before I release it to the world and I don't want anyone copying my idea before I do, but I'm sure that it's something that hasn't been done before.

Story

Storytelling in games has become more about presenting the plot to the player as the writers want the player to hear their story. Major plot points are given through the course of the game, and many games make storytelling an essential and required portion. In contrast, some games tell a story without telling the story in words. The player discovers the story piece by piece as they examine the setting, interact with the world, and talk to in-game characters at their own discretion outside of scripted scenes. A game that does this fairly well is *Cave Story* (Pixel Studios). There is a house in the game that is completely optional, and on entering you see rose petals on the floor. Before exiting the house, a mutated person attacks you. I knew from that point on that rose petals meant trouble, and that there was a reason for it explained somewhere in the game. Later on, it is explained that these flowers have been genetically modified to turn those who eat them into killing machines.

The developer could have simply said that to begin with, but coming into contact with it sooner on my own makes the story part of the gameplay. The story becomes a puzzle that you're slowly piecing together by yourself, immersing the player into the game more than they otherwise would have been. With my game, I wanted to tell a story more through the setting rather than through dialogue. It becomes more of what the player finds out on their own, rather than what the player was told. This sense of freedom of exploring the world more in order to find more clues for yourself and your own enjoyment adds to the sense of adventure and puts the story on a more personal level as something you discovered, instead of something that was revealed to you.

In particular, the player might figure out on their own that the game takes place underwater. There are large patches of seaweed growing here and there, streetlamps look like jellyfish, bubbles flowing up from the ground, and the end of an area has the player riding a bubble up to the surface. The player isn't told that it takes place underwater, but there are enough clues to stimulate their minds and have them come to that conclusion themselves.

Art

A game's art is normally the first impression that people get from a game, and in this respect it's valuable from a marketing standpoint. I noticed, however, that I eventually started to care about art less and less the more I played older games. The less defined in-game art assets are, you begin to see the artwork more for what it represents than what is actually shown on the screen. *Silent Hill* (KCET), a survival horror game on the Sony Playstation, is an early 3D game with primitive polygonal graphics. The models for the characters have sharp, noticeable corners and don't look all that great when taken at face value (Figure 5). When playing the game, however, the lack of detail shines through to you, subconsciously generating more detailed images into your mind. It leaves more to your imagination, and you begin to picture things that you personally think are scary instead of what the artists think is scary. The artwork is still important as what is portrayed on the screen directly influences the images you imagine, but knowing that players will fill in the gaps allows me to draw things as what they're representing instead of trying to sketch out every last detail.

However, it's difficult to keep a game cohesive by drawing things without a clear understanding of what you're trying to portray. Even though my game uses pixelated graphics, I drew out many designs to make sure that the final pixelated image evoked the same feeling that I was originally going for. Designing characters and objects in a detailed manner helped me draw better designs, moving from more basic designs to one with more personality and more cohesion with the rest of the game (Figure 6).

Music

There are just some songs that you never forget. Players are drawn to a game because of its art, they keep playing because the gameplay is fun, and they will remember the game for its music. Nothing brings on the nostalgia as hard as hearing a song from a game you played years ago, and humming catchy tunes as you do your daily chores puts pictures of the game in your mind, replaying clips of the game that you can remember playing when that music was beating out of your TV. Many games nowadays use music that is closer to ambient sound, solely there to fit an atmosphere for the moment that you are playing. While this is effective for drawing emotion as people play the game, there isn't that high of a chance that it'll bring back memories of a game when you hear it ten years later.

First and foremost, I wanted to make music that was catchy. Older games like *Super Mario Bros.* (Nintendo) and *Mega Man* (Capcom) contain songs that are forever burned into people's minds. The songs aren't necessarily great enough to put on your MP3 player to listen to on repeat, but they're catchy and are easily recognizable. Just thinking about these tunes is enough to paint pictures in one's mind.

Games like *Mega Man Battle Network 3* (Capcom) take it a step further. Not only are their soundtracks catchy and great to listen to, but many songs play off of each other by borrowing tunes from each other or remixing the main theme song. In *Mega Man Battle Network 3* in particular, the main theme song is spread throughout the game. It's there when you turn on your system, it's remixed when you edit your character, it's tied into some portions of some of the battle music, and it's mixed into the music leading up to the final boss. It's the accompaniment to the song that plays in the credits, which is

composed of different parts of other tracks played in the game. The main theme song is actually spread into the game's entirety, fitting for a theme song, whereas most other games only have one rendition of the tune in that particular title.

Other games like *Phoenix Wright: Ace Attorney* (Capcom) and *The Mystical Ninja Starring Goemon* (Konami) achieve a similar effect. Both games have songs that are remixed and played throughout different portions of the game, sounding different and familiar at the same time. What differentiates these two from the earlier example is that they also contain songs with buildup. For instance, in *The Mystical Ninja Starring Goemon*, the music in each stage starts out relatively tame. As you progress further and near the boss chamber, the melody is added onto and remixed, progressively becoming more exciting as you come to the climax of the stage while still having the same tune. *Phoenix Wright: Ace Attorney* is similar; as you come closer to solving the case, saving your client, and convicting the true criminal, the music ramps up into a turnabout. The songs are largely the same, but the way that they're played makes them sound different enough that you don't get tired of the same songs, and on top of that they're still etched into your mind as you hear them over and over.

I will never forget these songs, and I will never forget the games that these songs came from. In trying to find catchy tunes for my own game, I hummed or whistled different beats in my spare time. When finding something that was remotely interesting, I would keep humming the tune until I made it to a device that I could record the sound in to make sure I wouldn't forget it. I started this practice because initially I only tried to remember a tune from memory until it was time to make a song, but this proved fatal for those particular tunes because it seems pretty difficult to remember a few notes out of

context. Recording the different tunes has also had a side effect on my workflow: my computer is filled with tiny sound clips that are readily available for me to pick, choose, and create songs from. I made sure to make variants of songs for different parts of the game from these sound clips by changing the tempo, instruments, and overall feel of the song to represent a different scene. By doing so, I can have more time for players to memorize the songs while not boring them with the exact same track playing constantly. Making variants of songs and remixing different songs together makes for a more cohesive game overall.

Conclusion

As I exported a demo version of my game for this project, I knew that this game is far from finished in terms of what I still wanted to do with it. There is always the chance to add more music, more levels, more characters, more story elements, and more gameplay mechanics than those already present in the game. My knowledge of game design comes from the years I spent playing video games, so as I continue working on my game and continue to play new games, my knowledge as a game designer will grow, and hopefully I'll eventually have a game that I can take to the presses and without hesitation say, "it is finished."

Works Cited

- Atari Games. *Pong*. Atari Games, 1972. Arcade.
- Capcom. *Mega Man*. Capcom, 1987. Nintendo Entertainment System.
- Capcom. *Mega Man Battle Network 3*. Capcom, 2003. Nintendo Gameboy Advance.
- Capcom. *Phoenix Wright: Ace Attorney*. Capcom, 2005. Nintendo DS.
- Cinematronics. *Space Wars*. Cinematronics, 1977. Arcade.
- KCET. *Silent Hill*. Konami, 1999. Sony Playstation.
- Konami. *Silent Hill Image*. 1999. *Silent Hill Paradise*. Web. 23 April 2014.
- Konami. *The Mystical Ninja: Starring Goemon*. Konami, 1998. Nintendo 64.
- Nintendo. *Super Mario Bros*. Nintendo, 1985. Nintendo Entertainment System.
- Nintendo. *Super Mario Galaxy*. Nintendo, 2007. Nintendo Wii.
- Nintendo. *Super Mario Sunshine*. Nintendo, 2002. Nintendo Gamecube.
- Pixel Studios. *Cave Story*. Pixel Studios, 2004. PC.

Appendix

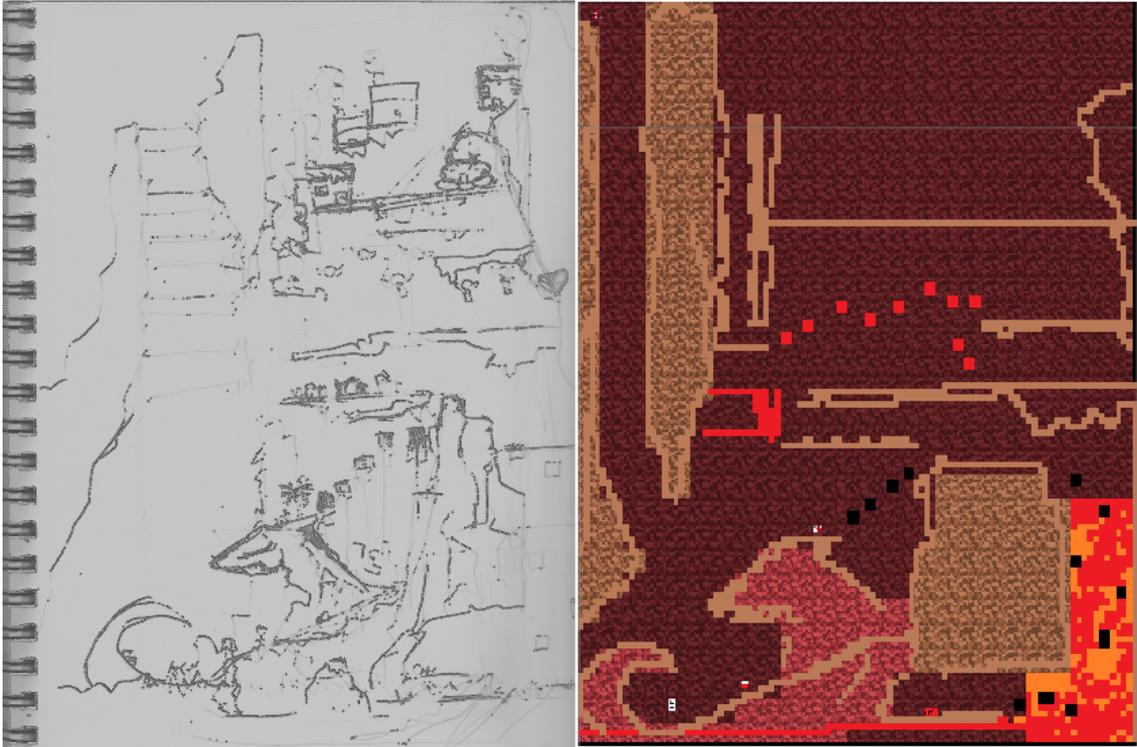


Figure 1. Level Creation – Moving from a sketch to a working prototype of a level. The working prototype is far from complete at this stage, but making it allows me to check that the level is fun and functional before devoting hours of work into it.



Figure 2. Beginning of the game – I want the player to feel like they need to make decisions on what they need to do next and that they're in control of the outcome of the game by setting them out in the open without telling them exactly what to do.



Figure 3. Bubble elevators – While moving platforms aren't new, the thought of riding a bubble sounds more interesting than your normal fare of elevators.



Figure 4. Peaceful enemies – These enemies are peaceful unless attacked, giving a new perspective on an old foe.



Figure 5. Silent Hill – While these graphics weren't half bad at the time of release, the models are starting to show their age. It's still just as scary as it was then because we fill in the gaps in our imaginations while playing.



Figure 6. Evolution of an Enemy Grunt – What started out as a basic enemy grunt slowly morphed into a sleek chameleon styled enemy. The first design was too familiar to be memorable to the player, so I kept modifying the design until it looked new. These are only a few of the many designs the basic grunt went through.