

The False Immortality Paradox

by

Chris Ferran

A thesis presented to the Honors College of Middle Tennessee State University in partial fulfillment of the requirements for graduation from the University Honors College.

Spring 2020

The False Immortality Paradox

by Chris Ferran

APPROVED:

Michael C. Murphy

Information Systems and Analytics

Dr. Timothy Greer

Information Systems and Analytics

Dr. John R. Vile, Dean

University Honors College

Abstract

For my Honors Thesis I have created a visual novel with multiple endings. This thesis was created to help me get a better understanding of how working on a long-term project can look like. The following paper discusses in great detail all of the different things that I had to do for this project, including what programs I used, all the problems I faced, how I overcame these problems, and more. This thesis takes place in the chronological order of events that I went through and goes through the process I went to make the finished project step by step.

Table of Contents

Abstract.....	i
Table of Contents.....	ii
List of Figures.....	iv
Introduction.....	1
Initial Vision.....	3
Ren'Py.....	7
Backgrounds.....	11
Character Creation.....	15
Testing Out Ren'Py.....	23
Gimp.....	27
Programming.....	30
Choices.....	31
Music.....	35
Sound Effects.....	36
Credits.....	37
The Characters.....	38
The Finished Product.....	40
Scheduling.....	41

The Different Mediums of Choose Your Own Adventures.....	43
Conclusion.....	48
Bibliography.....	50

List of Figures

Figure 1.....	6
Figure 2.....	19
Figure 3.....	21
Figure 4.....	22
Figure 5.....	23
Figure 6.....	26
Figure 7.....	44
Figure 8.....	45

Introduction

Choose your own adventure stories have always interested me. When I was a young child, my father would read them to me before I went to bed. He would let me choose which way the story would go, and it was always really exciting to see how my actions would impact the story. Later on in life, I would rediscover my love for the medium when I found out about a video game called *The Walking Dead*. I was a big fan of the television series so I figured I would purchase the game. I didn't know what the game was at the time. I figured it would be an action game where you just get to go around and shoot at zombies. What I found however, was a personal and moving tale of survival and fatherhood. The game was not focused on the zombies at all, it was all about the characters. What made this interesting was that you, the player, got to make the tough choices. Throughout the game you had to make choices of various magnitudes. Sometimes you just got to decide what you wanted to say to a character in the story, and this might affect how that character thought of you. Other times you might be put in a life or death situation and you had to decide who lived and who died. All of these choices impacted the story and made different play throughs feel entirely unique.

Telltale has also released plenty of other choose your own adventure games that are based on famous franchises from books, games, and comics. I have played through quite a lot of them and the fact that I could make choices that impacted the story always interested me greatly. A few years later Netflix released a movie called *Black Mirror: Bandersnatch*. What made this movie so interesting and unique was that it was also a choose your own adventure game. The only difference was the whole thing was done in live action. You got to watch scenes that were done by professional actors and choose

what paths you wanted this character to go on. Just like the Telltale games, the choices you make greatly impact how the story unfolds.

All of these books, games, and movies inspired me to try my hand at making my own choose your own adventure game. I had the idea of making my own a couple of years ago, but I had no real reason to pursue this desire at the time. I was way too busy with classes and schoolwork to do something as time consuming as this. Eventually, I learned that I was going to have to do a thesis for the Honors College and at the time I had no idea what I would do it over. I am an Information Systems major and I figured I would do something related to that. The problem was that I didn't have any idea of what to do. Should I make my own website? Maybe I could write some complex program in Visual Studios? I had some ideas, but I lacked any real inspiration or drive to nail down what I wanted to do. I ended up delaying my thesis by a semester so that I could give myself some more time to come up with a solid idea. A few more months passed, and it finally hit me: why don't I just make my own choose your own adventure game?

I felt so stupid for not coming up with the idea earlier. I knew that I wanted to make my own choose your own adventure story, and I knew that I had to do a thesis for Honors College, so how did I not put the two together sooner? I don't really know. What I do know was that I finally had my idea, so I went and asked one of my programming teachers, Michael C. Murphy, to see if he would be my thesis advisor throughout the whole process. He agreed and I was well on my way to creating a game. Most of the rest of this paper will be about my entire process for creating the game. From the initial visions I had for the story, to the details about all the different programs I used to create the final product.

Initial Vision

I now knew what I would be doing for my thesis, but I still needed to figure out all of the details. What would the story be about? What program would I use to create the game? How will I fit this project into my schedule? Will I ever get a full eight hours of sleep again? These were some of the many questions that I had in my mind during the first few days of the project. I decided that it would probably be best to come up with a basic premise for the story first and worry about all the other details later. I wasn't too sure about what I wanted the story to be about, so I spent hours looking through all of the things I have seen and played for some sort of inspiration.

One concept that I realized I liked was time travel. I took an entire class that was dedicated towards learning about time travel in fiction, so I had quite a lot of knowledge about that subject. I decided pretty early on that my story would incorporate time travel to some degree. One thing I remembered about many of the different time travel stories I have read and seen was how pretty much all of them involved science fiction. Most of the people time travelled by using some sort of device to get to different periods in time. I always thought that particular trope was a little overdone, so I wanted to do something unique. Most time travel stories deal with the consequences that come from time travelling, whether it be through the butterfly effect or some other means. I liked that, but I wanted to take that concept a step further. What if the time travelling itself came with some heavy consequences? What if the main character had to do something painful or immoral to go back in time? I really liked that idea so I decided to take that idea as far as I possibly could. I sat down and thought about what the worst thing someone would have to do to time travel would be. I figured out quickly that the worst thing would be to

sacrifice their own life. I decided to make it so that anytime the main character wanted to go back in time, they had to kill themselves. Obviously, this would only be temporary. As soon as they went back in time they would come back to life. But the process and ordeal that the main character would have to go through to go back in time would give the story some gravitas.

I was happy with myself for coming up with this idea. I thought it was unique and interesting. I was sure it had never been done before. I found out quickly that this idea has actually been done before. There is an anime called *Re: Zero – Starting Life in Another World*, where every time the main character dies, he is sent back in time to a predetermined point. Initially I was pretty upset to find out that my brilliant idea had already been done, but I decided to watch through the entire show to see exactly how the idea was done. The story follows the main character, Subaru, as he is suddenly transported to another world. He has no idea how he got there but eventually he meets some people and becomes friends with them. A little while later he ends up getting killed by someone. He is then transported back in time to around the time when he first arrived in this mysterious world. The rest of the show follows Subaru as he dies over and over again. Each time he figures out how he died and does everything in his power to stop that death from happening again. This detail is key to my whole project.

I realized that while my idea and *Re: Zero* did have a similar premise, the execution of the idea would be entirely different. My story was going to be all about the main character receiving this mysterious power of time travel and using it to his advantage. Subaru usually only uses his power by accident. He is murdered and then goes back in time against his will. My character would use his on purpose. That detail would be what

separates the two stories. On top of that the setting and characters that are present would be entirely different. So after a little scare that I might have to come up with a brand new idea to not seem like I was copying off of someone else, I was fine to move on. The next thing I needed to figure out was how the main character would receive this power of time travel. Like I said earlier, I decided that I wanted to forgo the science fiction setting that occupies most time travel stories and do something unique. I came up with the idea that the main character would be given the gift by some higher being. I thought this was unique enough to set it apart from other time travel stories. I also had another idea in mind that would coexist with this idea.

One other concept that I really enjoy in most stories is a plot twist. A plot twist is anytime that the story goes in a direction that one wouldn't expect. I thought that it might be interesting if I incorporated a plot twist into my own story. I decided to make it so that the higher being who gifts the power of time travel to the main character isn't who they say they are. I have them introduce themselves as Dolus, an Angel who has been sent by God himself to gift the main character the ability of time travel. Dolus is wearing an all-white outfit and is constantly clasping her hands together as if she is praying to God. Her eyes are also closed at all times to make it seem like she is praying.

The plot twist is that Dolus is not who she says she is. She is actually a demon who is trying to use the main character for her own benefit. When this revelation is realized, Dolus opens her eyes for the first time and the main character sees that her eyes are actually red, which signifies that she is probably evil. It further turns out that every time the main character goes back in time, he is actually inadvertently killing one random person on Earth. The way the time travel works in my story is that every time the main

character wants to go back, he has to kill himself and he is only sent back in time by one minute. This means that if he wants to go back in time by an hour, he has to kill himself 60 times, which means that 60 random people would be killed.



Figure 1: Normal Dolus vs. evil Dolus

One thing that I hate about a lot of plot twists is that they make no sense, or that there was absolutely no foreshadowing to lead up to the revelation. The best plot twist would be one that you don't see coming on your first watch, but you can see how the story reached that point on subsequent viewings. I wanted to make sure that the plot twist made sense, so I foreshadowed the ending in a few ways. The first way that I did this was in the name Dolus. In Greek mythology, Dolus is the spirit of trickery. It may be a little on the nose to name her after a mythological character, but I thought that most people would have no idea who Dolus is and wouldn't make any connection at first. The other way I foreshadowed this twist was that sometimes throughout the story, the main character would be watching the news and a special news report would come on. The news report would mention mysterious deaths that took place all across the world, and how these corpses would be found with their facial features missing. These scenes only took place after the main character time travels. I'm not sure if the plot twist is obvious or not, as I can't really be the one to judge that, but I am glad that it doesn't feel like a copout.

Ren'Py

Now that I had a good enough foundation with the basic premise, I figured it would be time to start looking at the technical aspect. The first and biggest thing I needed to sort out was figuring out which program I would use to make the whole game. The only program I had ever used was Visual Studio, so I initially figured that I would just use that. Visual Studio is a computer application that can be used to write code and make programs. It is the application that is used in most of the programming classes that I have taken, so it is the one I am the most familiar with. That being said, I am by no means an

expert with it. I understand all of the basic uses that we went over in class, but Visual Studio is a massive application with many different uses. If I were to use this application to develop my game, I would need to become much more familiar with it.

Initially I planned on having my game be a text adventure. A text adventure would be a game with no visual or audio elements at all. The user would simply see text on screen that told them what was going on in the story and choose from a few options to move along in the story in whatever way the user saw fit. This medium of storytelling has a few key benefits. On one hand, the programmer does not have to fiddle with any of the visual elements of a story. They can sit down and focus on the actual storytelling and writing elements. This can save a lot of time, money, and recourses. Another benefit is that it leaves a lot of the details up to the players imagination. If there are no visual elements, then the story can be as epic as the player imagines it to be. These are the main advantages of a text-based game, and these were the main reasons that I was originally planning on having my game be a text adventure. Honestly, I just thought it would be easier if I didn't have to bother with any visual elements.

I started writing my thesis proposal with the mindset that I was going to be making a text adventure in Visual Studio. I finished the entire proposal a few weeks before it was due, and I was about to start coding when I stumbled across a program that I thought looked rather interesting. One night I was browsing the internet to look for other applications that could be used to make a choose your own adventure game. While I was planning on using Visual Studio, I thought it was best to keep an open mind and look for some alternatives. I remembered hearing about this medium of storytelling that was called a visual novel. A visual novel is exactly what the name implies, it is a type of

novel that has visual and often times audio elements added into it. It is also played on a computer, so it is a type of game. What interested me about this so much was that it seemed like it would be much more impressive to create a visual novel than it would be to just make a text adventure game. Not that there is anything wrong with a text adventure game, but to many people a game with visuals and audio would probably be much more impressive and entertaining.

That is why I decided to look for alternatives to Visual Studio. I could make a visual novel in Visual Studio, but it would be much more difficult and time consuming than making a normal text adventure. I was looking for a program that I could use to streamline the process of creating a choose your own adventure game. Something that I could use to have visual elements, but would still be relatively simple to code in. I wanted the coding to be simple because I don't consider myself to be all that great at programming in general. I do fine in most of my classes, but it is always a major struggle for me to maintain a good grade in those classes. That's why I thought it would be a good idea to see if there was a program out there that could be used to make the programming a little simpler, without sacrificing the negatives that come with simplicity.

The first program I came across was *Visual Novel Maker*. This was a game that I found on Steam (a platform to purchase video games). At first look I thought it would be perfect. It was a game that could be used to create visual novels. You could add in visuals, music, and branching paths, which was everything I needed if I was to create a visual novel. I looked at the reviews and to my surprise, they were mixed. It seems like most people agreed it was nice to have a mainstream way to create visual novels, but it had plenty of problems. Another issue was that the game cost \$60. Not that that's a

massive amount of money, but I don't that much spare income, so I needed to keep costs down as much as possible. If the game had amazing reviews and everyone said that it was the best way to make visual novels then I definitely would have bought it, but a combination of the mixed reception and the cost made me decided to look elsewhere.

One thing that I did notice about the reviews for *Visual Novel Maker* was that a lot of people were comparing it to something called Ren'Py. They would talk about how Ren'Py used a faster engine, or how it had more options, or just how it was a better experience all together. I had never heard of Ren'Py before, but I decided to investigate it since it seemed to receive a lot of praise from developers. After looking into it I was dumbfounded by just how perfect it sounded. For starters it had almost every benefit that *Visual Novel Maker* had. You could add visual and audio elements, as well as create branching paths for your story to take. The big plus, however, was that it was completely free. You got everything the program had to offer without paying a dime. On top of that there were no ads or any other way that I could see for the developers to monetize it. I also found out that a couple of pretty famous visual novels were actually created in Ren'Py. The biggest of which was *Doki Doki Literature Club*. This game was a visual novel that went viral a couple of years back, as it was a psychological horror game disguised as a cutesy dating sim. The fact that such a high caliber game used this program gave me peace of mind. It was exactly what I needed. In that moment I decided that I would drop Visual Studio and create the entire game in Ren'Py.

Backgrounds

Now that I knew which program I would use to develop the game, and since I decided to create a visual novel, the next step was finding the visuals to use. The first step was to find the backgrounds that the game would use. Most visual novels have two main visuals. The characters in the story and the backgrounds. Backgrounds are just used to help the player know what setting a scene takes place in. For example, if you wanted a scene to take place in a restaurant, then you would need to find a background to indicate to the user that this scene does indeed take place in a restaurant. Most developers make their own backgrounds for their visual novels, but there was one big problem with that: I have no artistic skills at all.

In that moment I realized it was probably a stupid idea to make a visual novel in the first place. If I don't have the talent to create the visuals myself, then how am I supposed to create a visual novel? I considered switching back to Visual Studios and just going with my original plan to create a text adventure, but something inside of me really did not want to go down that route. At that point in time I got it in my mind that I was going to create a visual novel. I would fantasize about how awesome it would be when the whole thing was finished, and how I could show it off to everybody. I could tell everybody that I created my own video game. The idea of reverting back to a text adventure just bummed me out, so I decided that I would do everything in my power to create my own visual novel.

This meant that I needed to find backgrounds on the internet. The first thing I did was google "Backgrounds for visual novels". This led me to a website called itch.io. I had never heard of itch.io before but it is a virtual marketplace for game developers to sell

their games and game assets. The key detail in that definition is game assets. Instead of purchasing an entire game to play, developers can purchase pre-made game assets made by other developers to use in their own game. Through this website, I was able to find a vast amount of visual novel backgrounds.

At this point I hadn't decided all the details of the game, but one thing I did have in mind was the setting. I intended the story to take place in an urban city, and it would follow the main character, a guy in his early 20's, as he navigated his way through a dead-end office job. It turns out that this premise had one major problem. I couldn't find any good backgrounds that took place in an office. I could find plenty of backgrounds that took place in a school, or in nature, or in some sort of fantasy setting, but none in an office setting. I realized at that point that beggars couldn't be choosers, and I knew that I would just have to compromise and change the setting of the story.

I decided that the new setting would take place in high school. The main reason I chose this setting was that it would still be close enough to my original idea that I wouldn't really have to change all that much. Instead of the main character working at a dead-end office job, he would just be a high school student who liked to slack off. Another reason I chose this particular setting was that there was a plethora of school backgrounds to choose from. It turns out that a lot of visual novels take place in high school, so I had plenty of backgrounds to choose from. One final reason I chose this setting is that I have plenty of experience with high school. I did spend four years of my life there after all. I figured this would give me some relatability with the story and make it easier to think of ideas for the game.

With all that in mind, I started to look for specific backgrounds to use. While I knew I needed school backgrounds, I also needed plenty of other backgrounds. Much of the story would indeed take place in a school, but many other scenes would take place in other settings such as at a house, a restaurant, or the beach. I was trying to find something that would include all of those backgrounds with a semi consistent art style. I figured that it would be best if I got as many backgrounds as possible from one place, as this would make the art style consistent and it wouldn't be as disorienting for the player as the scenes transitioned from setting to setting. Luckily, I was able to find a good background pack pretty quickly.

The background pack that I ended up using was simply called "Visual novel background pack" and was developed by Selavi games. After looking through all of the backgrounds that it included, I knew that it was exactly what I was looking for. It included plenty of backgrounds for a school setting, including backgrounds for a classroom, hallway, restroom, stairwell, cafeteria, front entrance, and rooftop. It also included other backgrounds that I knew I would need including ones such as a bedroom, living room, kitchen, restaurant, beach, bus stop, and a park. There were also a couple of other ones that were included but the ones I listed were the main ones I would use. The pack cost \$10 and I bought it without hesitation. One thing I was worried about was copyright, as I wanted to make sure I didn't get in any legal trouble since I was using other people's work. Luckily their page stated that "If you decide to purchase and use my backgrounds, you are free to edit them in any way you'd like, use them in commercial projects, but do not resell or redistribute them on their own. Crediting, if possible, would be appreciated." In other words, I was free to use it in any way that I see fit as long as I

didn't sell the backgrounds by themselves, which I had no plans of doing. I also made a mental note at the time to make sure to credit anybody's art that I used in the game's credits, but I will go into more detail about that near the end of the paper.

While I did have most of the backgrounds I needed, there were still a couple of settings that I was missing. I knew I needed other bedrooms and houses to use as the story would eventually have the main character go to one of his friend's house to work on a project. I also needed a couple of other miscellaneous backgrounds including a home bathroom and some background that I could use to represent Heaven, as the main character does end up dying at some point in every playthrough. After looking around I ended up finding this webpage called "UNCLE MUGEN'S FREE VN (OELVN) RESOURCES". This is a webpage that was dedicated to giving developers access to free visual novel resources, including backgrounds. Here I was able to find pretty much everything I needed. I found a good background to use for the house bathroom, as well as a background to use for the main character's friend's house. I couldn't find a perfect background for Heaven, but I was able to find one of a nice beach backgrounds, so I just decided to make Heaven a beach and call it a day. The terms of usage for these assets were the following:

- “▪ Feel free to use for whatever purpose it may serve best.
- OK for both Commercial and Free projects.
- Modifications are OK and highly encouraged.
- Not really required but I really do appreciate if you could tell me you have a project using these free resources.
- TLDR: Just take them... no strings attached...”

So just like the other backgrounds I found, I won't get into any legal trouble for using them. I also made sure to remember to include the website and details in my games credits to give him the credit he deserves.

Character Creation

Now that I finally had all of the backgrounds I needed, the last element I had to get before starting my work on the actual game was the characters. My first idea was to find premade characters on the internet. I figured that since I was easily able to find plenty of backgrounds, finding good characters to use should be a breeze. I had a general idea of all the characters I needed but I was willing to be flexible with how they were designed. The only character that I had a somewhat specific design in mind already was Dolus, as I detailed above. After looking around through a bunch of different websites, I came up short. While I could definitely find characters to use, they all seemed rather ugly and unappealing to me. While I did say earlier that beggars couldn't be choosers, I was disappointed with the results I found. The main problem was that all of the characters were made in completely different art styles, which meant that if I did choose a bunch of characters at random, they would all clash with each other and look rather ugly. Also, I came to find that most of the characters only had one look and facial feature. A big part of visual novels is the fact that the characters will change their facial features to react to what the user has the main character say. This is used to give the game a bit more pizzazz, and make it look more professional and realistic.

Since I couldn't find any premade characters that struck my fancy, my next idea was to see if I could make my own characters. Like I said earlier, I am no artist, and I knew that I wouldn't be able to make the characters from scratch. My idea was that I might be able

to find an application that is used to make characters. It would be similar to how you can make and customize your own characters in video games like *The Sims* or *Grand Theft Auto*. I looked around the internet and my searching brought me right back to itch.io.

I was able to find an application called Mannequin, which is developed by a company called AR14. The webpage for Mannequin talks about how it is an application that can be used to create 2d character portraits for video games and visual novels. After looking around the internet for people talking about it, I found that it was rather new and still in Early Access, which meant the final product hadn't been officially released yet. This wasn't actually as big of a deal as it might sound, as I was still able to purchase and use the product. All Early Access means is that the application is still being worked on by the developers, and there might be some bugs and kinks that still need to be worked out. I wasn't able to find any other application that fit my needs as much as this one did, so I bit the bullet and purchased Mannequin for \$10.

I was able to download two different versions of Mannequin. One was a stable version of the game that was in a usable state and didn't feature any bugs. The other one was called a "Nightly" build of the game. This version was constantly being updated with new features and ideas. The only problem was this version was much more unstable than the other one. I still decided to download and try them both out to see what the differences were. I started out with the Nightly build as I wanted to see what all the extra features that the developers were working on were. The main differences I could find were that there were more clothing and posing options for the female characters. I found out quickly that the male characters wouldn't work at all, as the male body template wouldn't even load correctly in the Nightly build. As there were much more options for

the female characters in the Nightly build, I decided to create the female characters there and just make the male characters in the stable build.

I created the first female character without any major issues. The application did take quite a while to load the character as it was a Nightly build, but it was definitely usable. I decided to start with Dolus, as she was the only character who I had an exact idea of what I wanted in mind. I created her pretty quickly and without any issues and moved on to create all of the other characters. I decided to move over to the male characters, as I wanted to make sure that there weren't any major issues with the stable version of the application.

The first male character that I created was Darius, the main character's best friend. I quickly found out that I was going to have a major problem. The male characters barely had any options at all when it came to customization. You can customize your character in a variety of ways through Mannequin. One of those ways is through poses. Poses are just the way the character stands in the game. For example, a character might have a pose where they cross their arms, or one where they have their hand on their hip. You can use poses to give a character more personality than they otherwise would. For females, the characters had a wide range of poses. The males on the other hand, only had three. All three of these poses were pretty much the same as well. This meant that all of the male characters in my story would pretty much have the same pose throughout the entire game.

Another customization option is facial features. You can change the eyebrows, the shape and color of the eyes, the nose, the mouth, and the skin color. Luckily the male options for these were plentiful so I didn't have to worry too much about this. I decided to make Darius black to add some diversity to the game. One cool option in Mannequin is

that you can give the characters different facial presets. This means that you can have different versions of the same character where they will make different facial expressions. This is a good way to make the visual novel feel a lot more alive than it would if they had the same expression throughout the entire game. I decided to give Darius, and most of the other characters, three different facial presets. These three presets were default, angry, and happy. Default is exactly what it sounds like. The character would have a neutral expression on his face. He or she might have a slight smile on their face, but this would be the default face they would start most scenes with. The next preset is angry. This is just the face that the character would make when they were mad at something that was going on in the story. The last preset is happy. This is the face the character would make when something good happened in the story.

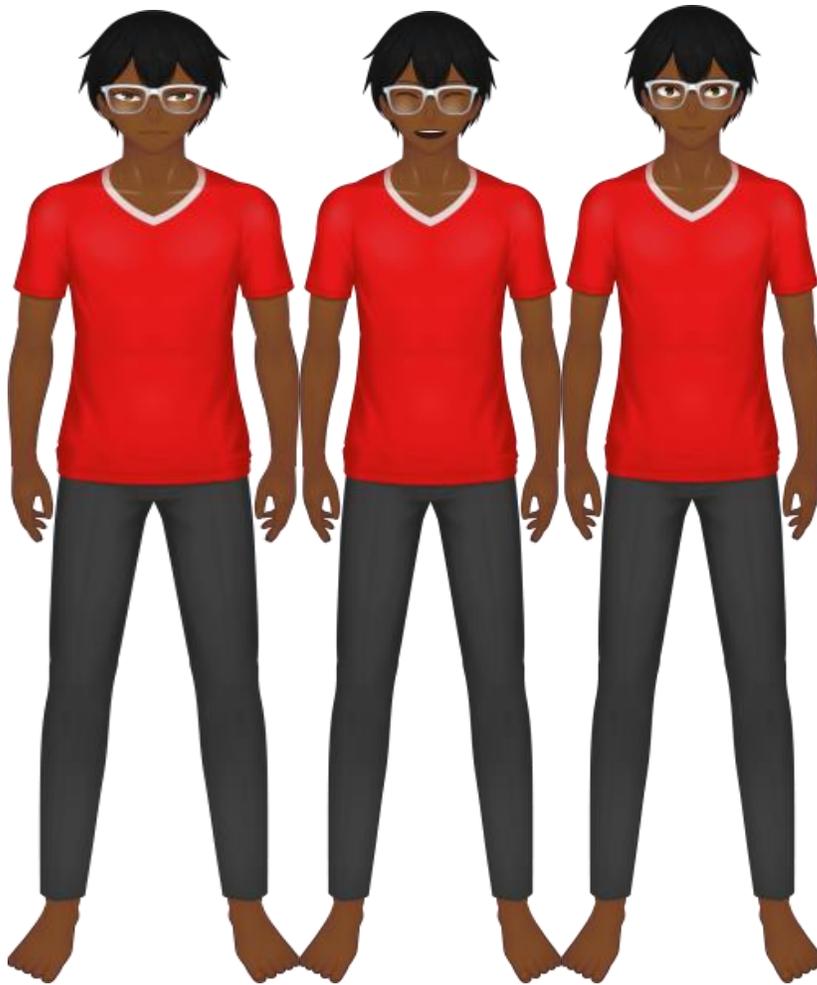


Figure 2: The different facial presets of Darius

I figured that these three faces would be enough to cover all of the different emotions that came up in the story. In hindsight, I do wish I gave each character a couple of more faces to represent more emotions, but by the time I realized this I had already imported all of the characters into the game and adding more facial features would be a major hassle and would take a lot of time that I didn't have. For example, there is a couple of moments in the story when the character is supposed to be sad. I didn't have a facial

expression to represent sadness, so I decided to use the angry preset, as it was the closest thing I had.

After facial features, I had to give the characters a hair style. This was another customization option where the male versions were severely lacking. The females had at least 20 different hair styles to choose from, while the males only had two. Those two were bald and generic male hairstyle. It was at this point where I realized I was going to have to switch things up a bit. I originally planned on having about an equal amount of male and female characters, but after seeing how severely lacking the male options were, I decided to switch up the gender of a character to make things easier on myself. The teacher was going to be a male, but I decided I would make her a female. This would mean that the game would feature six female characters and three male characters. I decided to give Darius the only male hairstyle with hair.

After the hairstyles, the final customization option was clothing. Just like the hairstyles and poses, the male customization options were limited. They only had a few options for shirts, one option for pants, and no options for shoes. I was dumbfounded when I realized that the developers hadn't given the male characters any options for shoes yet. This meant that all of my male characters would have to go around barefoot for the entire game. I knew that wouldn't work so I decided to get creative with it. I let Darius stay barefoot, but I gave him a personality trait where he had a severe phobia towards wearing shoes. When it came to my other male characters, I knew I would have to get creative.

The next male character I created was Jerry. Jerry is supposed to be a guy who bullies the main character throughout the story. I knew I didn't want him to have the same hairstyle as Darius, so I decided to give him a girl's hairstyle that could be passed off as a

male hairstyle. I gave Jerry long, flowing white hair. Since there was only one pants option for the men, I colored Jerry's pants white to differentiate him a little more from Darius. I knew I would have to figure out a work around to the shoes problem, as I knew that I couldn't make every male character scared of shoes. What I ended up doing was cropping Jerry's character in Gimp, a photo editing application that I will talk about more in more detail later. I cropped him so that you couldn't see Jerry's feet. The picture below is an example of how this looked in the game.

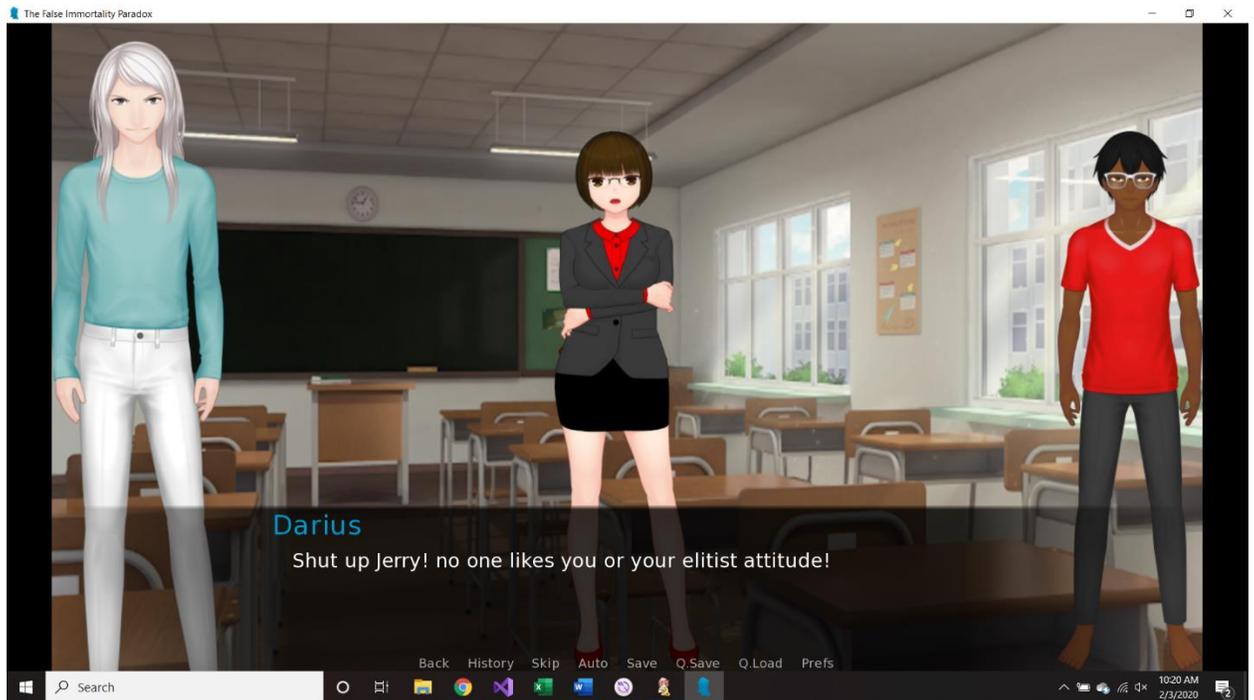


Figure 3: An example of how part of Jerry is cropped out

Notice how Jerry, the person on the left, doesn't have his feet in view of the screen. This is how I got around the shoe problem. In hindsight, I wish I would have done the same

thing for every character. I think it looks a little better when you can't see the entire character in frame. In this picture for example, Darius looks like he is floating in air instead of standing on the ground. This is another problem that I would have liked to fix if I had more time.

After Jerry, the last male character I needed to make was a thief that would kill the main character to get the story started. You would only see the thief at night, so I made him a silhouette by making every single-color option in the application pitch black. The only part of the character that you can see is his eyes. An example of this can be seen below.

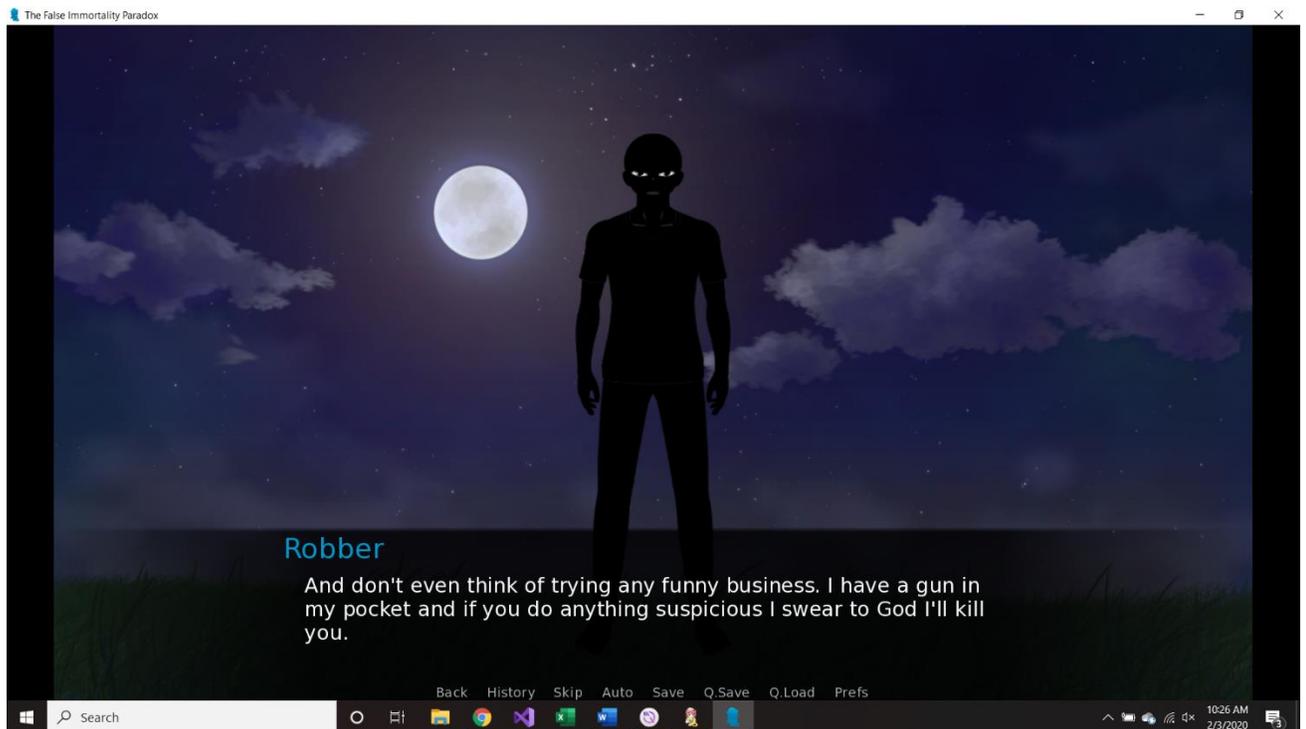


Figure 4: An example of how the thief looks in the game

After making all of the male characters, I switched over to the stable version of Mannequin to create the female characters. I did not run into any issues during this part of the process and everything went as smoothly as possible. With all of the extra customization options that were exclusive to the females, I was able to be a lot more creative with the design for these characters. They all have vastly different outfits, hair colors, hair styles, accessories, and poses.



Figure 5: All the female characters used in the game

Testing Out Ren'Py

After I finished making all of the characters, I thought I had everything I needed so I opened up Ren'Py to begin coding. Before I actually started coding, I noticed that Ren'Py

offered a few tutorials on how everything worked, so I decided to watch those, as I had no idea how to code in Ren'Py. It took about an hour to go through the tutorial, but I was happy that I did as coding in Ren'Py was going to be very different than any of the coding that I have done in my classes. In every class I have taken, we used a programming language called C#. A programming language is just the different words or phrases that you use to program. Every programming language uses different words to do the same task, so if you only know one programming language than you will have to look up what the differences between two languages are to code in another language. I had to do a similar task for this project, as Ren'Py does not use C#. Ren'Py instead opts to use Python. Luckily the programming that I was going to do in Ren'Py would be pretty simple, so I didn't have to learn too many new things.

The first thing I did in Ren'Py was set up my first scene. You do this by using the "scene" keyword. This keyword lets the program know what background you want the scene to take place in. For example, if I wanted a scene to take place in the classroom, I would type in "scene classroom". In this scenario, classroom was the name of the file that I gave to the classroom background that I used. If I wanted the scene to take place in the field background that I found, I would simply type in "scene moon over field", as moon over field was the name of the file that I used for that background. I knew that I wanted the game to start in the classroom, so I typed in "scene classroom".

The next step was to show one of the characters. The game begins with the teacher, Mrs. Striker, telling the class that they were going to have a pop quiz. The way in which you show a character in the game is by using the "show" keyword. This tells the program that you want to show some picture or image that is in your file. In this particular instance

I wanted to show a character, so I typed in “show striker”, as I wanted to show Mrs. Striker and striker was the name that I gave for the file that had Mrs. Striker’s default expression. If I wanted to have a character change their expression in the scene, then I would change their expression by typing in the “show” keyword and whatever the name of the file is that has the character’s other expression. For example, if I wanted to have Mrs. Striker change her expression from default to angry, I would type in “show striker angry”.

Now that I had the background and the character set up, the last step was to write a line of dialogue. The first line of dialogue in the game is an internal thought that the main character has. The way you show dialogue in this game is by using quotation marks. If you wanted to have the main character greet Darius, then you would simply type in “What’s up Darius?”. It gets a little more complicated if you want to have a character in the game say something, but it is still pretty simple overall.

The first step to make an in-game character say dialogue is you need to define the character. You only need to do this step once for every character. Defining a character is simply a way for you to tell Ren’Py that a character exists in your story. The syntax for this is a little complicated, so I will write the whole thing out and then explain it piece by piece. If you wanted to define Mrs. Striker, you would write “*define s = Character("Mrs Striker")*”. The first part of this line of code is the “define” keyword. This is your way of telling Ren’Py that you are about to define a character. After the define keyword, you type in the one or two letter abbreviation that you want to use for the character. The way dialogue for characters in Ren’Py works is similar to how it works for the main character’s internal monologue. The only difference is that before the quotation marks,

you need to type in the abbreviation that you have for a character. If you wanted to have Mrs. Striker tell the main character that they failed the pop quiz, you would write “s “I’m sorry, but you failed the quiz.””. In this example, the “s” before the quotation marks is the abbreviation that we gave for Mrs. Striker. If we wanted to have Darius say this line of dialogue instead of Mrs. Striker, you would simply change the “s” into a “da”, as “da” is the abbreviation that I gave for Darius. So you would write “da “I’m sorry, but you failed the quiz.””.

The next part of the syntax of “define s = Character(“Mrs Striker”)”, is “Character(“Mrs. Striker”).” This part of the line is used to tell Ren’Py what the full name of the character that you are defining is. This is used in the game to help the player know who is saying what line of dialogue. An example of this can be seen below

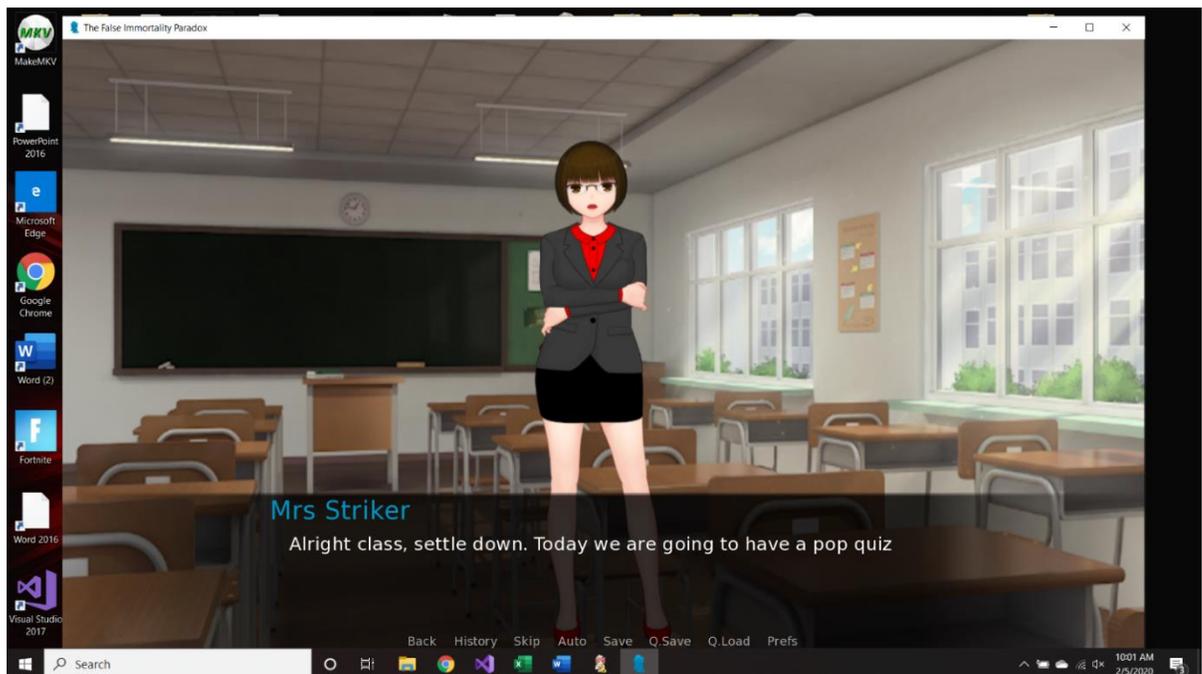


Figure 6: An example of what a character name looks like in the game

Right above the line of dialogue shows the name of the character who is speaking the dialogue. It might sound unnecessary when there is only one character on screen, but if there is two or three characters then having the character's name displayed is necessary to let the player know who is speaking what line of dialogue.

Gimp

At this point, I figured that I had everything I needed to test the program to make sure everything was working as I intended it to. I programmed in a scene, a character, and a line of dialogue. I saved my work and opened Ren'Py to launch my program. The second that I opened my game I knew that something was wrong. It did load without any problems, which was certainly a relief, but the background looked off. The entire thing was zoomed in, which made the characters look like they were less than a foot tall. I was perplexed and had no idea what was wrong at first, but I came up with a solution rather quickly. I figured that the size of my backgrounds might be different than the size of backgrounds that Ren'Py wanted me to use. The default size is 1280x720, and my backgrounds were a different size than that.

My immediate thought was to use Photoshop to resize my images. After looking around I found out that Photoshop costs quite a bit of money, so that was not an option. The next thing I did was look around the internet for any applications that could be used to resize images. The most prominent answer I found was an application for the computer called Gimp. Gimp is a lot like Photoshop. It is an application that is used to edit and customize pictures. It sounded like it was exactly what I needed so I downloaded it and opened it up. I was able to figure out how to change the size of the backgrounds rather quickly. All I had to do was select the background that I wanted to edit, click on a button called "scale

image”, and manually input the resolution that I wanted. In this case that resolution was 1280x720. I started with the classroom background as that was the one that was giving me problems. I changed the resolution and uploaded the new version to my file with the game in it. I tested it out and it worked perfectly. The background looked exactly how I wanted it to.

Next, I went back into Gimp and started to change the resolution of every image I had. This was a rather simple process, but it was tedious as I was just repeating the same steps over and over again for a lot of different images. All in all, it took around two hours to do. I went back into Ren’Py and put in all of the different images to make sure there wasn’t any problems, and to my relief every image worked out fine. I thought everything related to the images was finished, and I was ready to start working on the actual story, but another problem arose.

Just like the background images, the characters themselves weren’t the right size. They were so zoomed in that you could only see their waist on screen. At this point I knew that I needed to resize the images through Gimp, so I opened it up and went to work. At this point I found out one major problem. Ren’Py doesn’t have a set character size. This means that the only way for me to figure out the right size for the characters was through trial and error. This might not sound that bad, but between opening the file, changing the size of the image, saving and uploading the image to my game, and testing out the new size, it would take around five to ten minutes just to do one test. Since I didn’t know the right size, this meant that it could end up taking quite a long time. I decided to just bite the bullet and get to work. It probably took me five hours or so to find the right size, but

eventually I did. I found that setting the character size to 161x600 pixels seemed to do the trick. At this size, the entire character was in view of the screen.

I went through and changed the sizes of all of the characters to fit this new resolution. I wasn't done yet though, as I still had one more issue that I needed to solve. I needed to cut out Jerry's feet so that you couldn't tell that he wasn't wearing any shoes. This turned out to be a bigger problem than I was expecting. I was able to cut out his feet rather easily using a feature of Gimp called "Crop to Content". Using this feature, I was able to manually crop out Jerry's feet. This would have been the end of it, but just like all of the characters Jerry had multiple versions to crop. Since every character has multiple facial presets which are used to represent different emotions, they have different files to represent these emotions. That means that I had to scale not just every single character, but every single version of the characters as well. This wasn't an issue with all of the other characters as I could input the exact size they needed to be, which made them all exactly the same size. With Jerry, I didn't have this luxury, as I had to manually crop out his feet, which means that there was room for error when I did the actual cropping for all the different versions of Jerry.

This part turned out to be rather time consuming. I had to do my best to make the edits as perfect as possible on every version or else Jerry would be a different size whenever he changed facial presets in the game. Just like before with the other characters, this took a lot of trial and error. I would have to edit his feet out on one version, and then go to another version of Jerry and edit his feet out in exactly the same way as I did before. I would then compare the images, and if they weren't the same, I would have to start over from scratch. In the end, this ended up being the most annoying part of the process. I

never felt like I was making any progress, as I had to start over whenever I made a mistake. I was about ready to give up and just ignore the problem, when at some point I randomly managed to do the cropping perfectly on every version of Jerry. I'm not sure what I did differently on the one time that I did it right, but the important thing was that I was done, and I could finally move on.

If I could go back and restart this project from the beginning, I would have probably made the characters bigger. In most visual novels, you can only see the characters from the waist up. This makes it look like the character is right in front of the player. In my game you can see the entire character, from the head to the feet. In retrospect, this might not have been a good choice, as it makes the characters seem further away than I would have liked. That's not to say that the way I did it was completely wrong. I feel that having the characters further away can be beneficial if there are more than one character on the screen. It doesn't feel nearly as cramped as it would in a normal visual novel.

Programming

Now that I finally had all of the pieces of the puzzle put together, I could finally begin the actual programming of the game. I knew how I wanted the story to start, so I programmed the first 80 or so lines of code with ease. The story begins with the main character being annoyed at the fact that nothing ever seems to go his way. His teacher then tells the class that today they were going to have a pop quiz. The character gets the quiz and has to put his name on it. I decided to start the story this way as I wanted the player to get to choose the name of the main character. That way if someone who is playing the game is named Carl, then they could make the main characters name Carl, and everyone in the story would refer to him as Carl. This is a way to immediately hook

the player into the game and immerse themselves into the world by putting them into the shoes of the main character. I could have made it so that you type in your name before the game even begins, but I thought it would be fun to have the player write down their name in the actual game to have the story be more realistic.

The problem with this is that I had no idea how to program something like that into Ren'Py, so I went to the internet to search for solutions, as the tutorial never covered it. Luckily Ren'Py has an entire website that is dedicated to answering these questions, and I was able to find the code to let the player choose their own name. The only other thing you need to do is set a default name in case the player decided not to input any name. I decided to choose the default name of Pat. I didn't have any real reason to choose this name, it was just the first thing that came to mind. Throughout the rest of the game, anytime I want a character to refer to the player as the name they chose, I would simply have to input the syntax [name] into a line of code. For example, if a player decided to make their name Carl, and I wanted to have Darius refer to the main character as the chosen name, then I would simply code a line that looked like the following: *"da "Hey [name], what's up?"*". To the player, this line would read "Hey Carl, what's up?".

Choices

The only other major programming tool that I knew I would use throughout the game was choices. This is a choose your own adventure game after all, so I needed to have the player be able to choose what they wanted to say or do in different scenarios. The syntax for this can be as simple or as complicated as you want. I decided to keep things simple, as I knew I didn't have much time or resources to make this game. Normal game development can take years and can be done by a large team of talented developers who

have a lot of experience and recourses at their disposal. I was one person with zero experience in game development who had a few months and no expendable income. On top of that I also had all my other classes to attend and do homework for. Now obviously my game is not nearly as complicated or professionally made as most others, but I was very tight on every resource possible, so I needed to keep things as simple as possible.

That meant that I wouldn't be able to make the game nearly as free as I wanted. Originally, I wanted every choice to matter and impact where the story went. I wanted there to be loads of endings that you could only reach if you made certain choices. If I had a couple of years to do this, then I would have. Unfortunately, that wasn't the case, so I had to keep things a little more basic. I still wanted to let the player make choices that had an impact on the story, so I decided to compromise.

I thought it might be good to have the player be able to make many choices that don't impact the story at large, but instead change small details of a scene. For example, there is a scene where the player is eating at a restaurant and they get to choose what they want to order. The choice doesn't affect the entire story, but it does change how the scene play out. At another point in the story the characters need to work on a group project for a class and the player gets to make the choice of where they go to work on it. This doesn't matter all that much, but it does change the background of the scene and you get to learn some interesting facts about different characters if you choose different options. These are examples of how most of the choices in the story go. They effect individual scenes, not the entire story.

That's not to say that there aren't any choices that change the entire story. The game has four different endings that can be reached by making certain key choices in the story.

One ending can be reached rather quickly in the game. This ending involves the player deciding not to accept Dolus's gift, and instead opts to just ascend to Heaven. If the player chooses this ending, then they will miss out on a large portion of the game. If the player doesn't choose this ending, they still have quite a bit of game left to experience. The next choice that effects the entire plot of the game happens rather late in the story. In this choice, the player gets to decide whether they want to go back in time to save their sister who died tragically in an event that is slowly unfolded to the player over the course of the story.

This particular choice has a huge impact on the tone of the game. If you choose to save your sister, you get a tragic story full of death and sacrifice. If you don't, you get a more traditional happy ending. Choosing not to save your sister also leaves to another choice that changes the ending. There is a character in the game called Yuuki, who plays a pivotal role in the end game for the different endings. If you manage to reach the happier path, you get to choose whether to romance Yuuki or not. This only changes a few lines of text, so while it is technically a different ending, it doesn't really feel all that different.

The actual programming of these choices was much simpler than I was expecting. All it takes is a few lines of code. You start by calling the menu item. This just tells Ren'Py that the player is about to be able to select from a couple of different options. Next you simply write what those options are, and you tell Ren'Py where those options are going to be programmed in relation to the rest of the code. Next you program whatever the outcomes of the choices are. Finally, you end the choice code and jump back to the main story.

For example, if you wanted Darius to ask the player how his day is going, the code may look like the following.

```
da "Hey [name], how is your day going so far?"
```

```
menu:
```

```
    "It's going pretty good."
```

```
        jump day_good
```

```
    "Shut up Darius."
```

```
        jump day_bad
```

```
label day_good:
```

```
    da "That's great!"
```

```
    jump day_done
```

```
label day_bad
```

```
    da "Well you don't have to take out your anger on me, jerk!"
```

```
    jump day_done
```

```
label day_done:
```

In this example, the player gets two choices to Darius's question. They can either tell Darius that their day is going good, or they can tell him to shut up. If they tell him that their day is going good, then the program will jump to the line of text where Darius tells the main player "That's great!". If the player tells Darius to shut up, then the

program will jump to the line of code where Darius calls the player a jerk. Either way, after Darius says his line in response, the game continues after that the exact same way no matter which option you chose. This particular example was simple, but when you are programming multiple endings with different choices, it can get much more complicated. However, Ren'Py makes it very simple to program in, so I never had much of a problem with programming the different paths. If you mess up at any time, Ren'Py will tell you exactly which line you messed up on and it will even offer suggestions about how you might have messed up. This makes it easy to tell when there is something wrong with your code and I could always figure out a solution quickly and without much problem.

As far as programming goes, the “menu:” label was my bread and butter while making this game. I used it often, as the player gets to make many choices throughout the game. Besides that, though, I still need to talk about a couple of different aspects of the programming side of the story. I mentioned earlier how all visual novels have two main elements, backgrounds and characters. What I didn't mention however, was that there is actually a third element that most have. This element is music and sound effects.

Music

I didn't initially plan on having these elements in my game, as I didn't think I would have enough time to implement them and I thought they would be way too difficult to program. I did end up finishing the game quicker than I anticipated, so I figured I would at least investigate adding these elements. The first thing I did was look into how you actually program music into Ren'Py. Just like everything else, I found this to be very simple. All you must do is use the “play music” keywords, followed by whatever the

name of the track is in your game files. Since I now knew how easy it would be to do the actual programming, I decided to find some music to put in the game.

I needed to make sure I found copywrite free music. If I used songs that I wasn't allowed to, I could get into some serious legal trouble. All I did to find free music was look up "copywrite free music" into Google. I quickly found a website called <https://incompetech.com>. This website's entire purpose was to have royalty free music available for download. This works by clicking on a genre of music that you want to use, listen to a list of available songs, and download the ones you want to use. Each song has conditions that you need to make sure you follow to use the song in your work for free. Every single song I used happened to be made by the same guy, so they all had the same condition. All I had to do was make sure I gave the guy credit in a specific way in the game's credits.

I copy and pasted all of the song's conditions into a word document, so I knew exactly how to give him proper credit and downloaded all of the music I needed. I changed the names of the songs in my game files to the name of the scenes that I would use the song in. For example, I named the song that I would use in the classroom setting "classroom". This way it would be much easier to add the music into the right scenes. There were a couple of music tracks that I knew I wanted to use in certain scenes to match the tone I was trying to go for, so I gave them more generic names.

Sound Effects

After I finished adding all of the music, I went back and looked into adding sound effects. As I mentioned earlier, the main character's only way to time travel is to die.

Eventually, this leads to the main character using a gun to quickly kill himself so that he can go back in time as easily as possible. Whenever the main character shoots himself, I had a line of code that simple read “*BANG!!!*”. I wanted to add a sound effect to this word to add the proper oomph that I thought it needed. Just like the music, I decided to first look and see how you add sound effects into Ren’Py. As was tradition at that point, I found out it would be easy to program in sound effects. All you had to do was use the “play sound” keyword followed be the name of the sound effect you had in the game’s files.

I found and added the sound effects in a similar way that I added the music. I simply looked up “copywrite free sound effects” into Google. I was able to find a website called freesound.org, which just like the website I used for music, was dedicated to offering free sound effects for people to use however they see fit. Certain sound effects don’t require any credit, and you can use them in any kind of project. Others require you to give credit and restrict you from using them in a commercial product. I made sure that all of the sound effects I used were from the former, as I didn’t want to have to deal with any specific copywrite laws. I did decide to give the people credit anyways by providing a link to the webpage that I found the sound effect in the game’s credits. After I found all of the sound effects that I needed, I downloaded them all and added them into the game where I needed them.

Credits

The last part of the actual programming that I needed to do was the credits. This was where I was going to give proper credit to all of the people and places that I used for the backgrounds, characters, music, and sound effects. My initial plan was to give the game

the same type of traditional scrolling credits that you would find in a movie. I was able to get a somewhat working version of this up, but it ended up not working out how I wanted, and it caused some issues with the game, so I decided to come up with a workaround. I simply programmed in the credits in the same way that I would program any other character. I defined a credits character and then had a line of code dedicated to every person that I needed to credit. Now that I was finished with the credits, I was finally done programming for the actual game.

The Characters

While I was programming this game, I did realize one major issue I had with my own story. I hated a lot of the characters. I thought they were boring, uninteresting, and one dimensional. I realized that I wasn't enjoying writing for them as I didn't have much for them to say. This didn't affect every character, but it definitely affected some of the main ones. The big three characters I ended up hating were Darius, Winrey, and the main character's mom. I hated all of these characters because I realized that I didn't think their personalities through all that well in the planning stage.

For Darius, the only interesting characteristic he has is that he is afraid of shoes. That aspect of his character only came around due to necessity. All he does throughout the story is ask the main character how he is doing and talks about random things. I hated writing for him, so I decided to sideline him when it came to the main plot, and only used him when necessary. I hated Winrey for similar reasons. She is on the soccer team and is the class president, but that's about it. As far as personality goes, she is as bland as a wet rag. Originally, she was going to be the main character's love interest, but I abandoned that idea when I introduced Yuuki into the story, who is much more interesting. The

mother does have a few interesting parts in the story that come up through nightmares and flash backs that the main character has, but in the actual story she has the exact same problem as Darius. All she does is ask how the main character is doing and she will occasionally cook the main character breakfast.

While I do hate these characters, there is an equal amount of characters that I do enjoy and feel like offered something unique to the table. The first one is Jerry. As I mentioned earlier, Jerry is the person who bullies the main character throughout the story. What makes Jerry so interesting to me is that he has no friends. In most stories I have seen, the bully is always really popular and successful, so I thought it might be interesting to make him lame. He also goes through some interesting character developments throughout the story, and the main character begins to understand Jerry more and more as the story goes on. His main development happens when his father dies randomly to a mysterious illness. The main character tries to comfort Jerry out of pity, and the two get in an argument.

The next character I really enjoyed writing was the robber who kills the main character at the beginning of the story. Originally, he was just supposed to be a plot device and I didn't intend to bring him back at all. Later on in the story however, the player gets to make a certain choice, and one of the possible outcomes leads to him meeting the robber again. The robber kidnaps the main character, and this leads to an interesting scene where you get to see some insight about what happened to the robber after the fact. I had fun writing him because I made him a really tragic character with an interesting backstory.

The final character that I really enjoyed writing for was Yuuki. I find this funny as I didn't even originally intend on having Yuuki be in the story at all. She only came to exist because I was messing around with the character creation in Mannequin. I ended up

bringing her into the story around the time when I was getting sick of writing Darius and Winrey's character. What I find so interesting about Yuuki is her personality. I based it off of a Japanese term called Chūnibyō, or "middle school syndrome". These characters usually don't have many friends and want to stand out, so they pretend that they have superpowers or some magical abilities to make themselves known to everyone around them. I thought this would be an interesting character, and it is something that I have never seen done in western media, so I decided to make Yuuki a Chūnibyō.

Yuuki is constantly telling anyone who will listen about her powers. The funny thing about this is she can never get her story straight. Every time she interacts with the main character, she tells him about a different power she has or a different adventure she has gone on. She even likes to refer to herself in the third person sometimes, but she can never remember her own last name. This leads the main character to realize that she is a compulsive liar. Throughout most of the story, the main character just finds her to be annoying. He just wishes that she would go away and bother someone else. Slowly throughout the story, you slowly get to find out why Yuuki acts the way she does, and it leads to some of the best scenes in the game in my opinion.

The Finished Product

While I do wish that I made characters like Darius and Winrey more interesting, I am overall happy with how the story came out. It is not perfect by any means, as it is my first foray into long term storytelling, but I do think I succeeded in making an interesting and unique story. One other small problem I had with the story was the pacing, especially in the midsections of the story. Before I started writing, I knew exactly how I wanted the story to begin. I also knew a couple of the endings that I wanted the story to reach. One

thing I didn't have much of a plan for was the middle. I needed to find a way to get the characters to the points that I wanted them to be for the final part and I wasn't sure how I wanted to get them there. This led to the story dragging on in the middle. I don't think there is any particular scene I could cut out to improve the pacing, but if I could go back to when I first started writing I would have probably thought things through a little bit more.

One other issue I have with the story is one of the endings. The ending that I had in mind before I started writing is dark and depressing. That particular ending reaches points that can be quite horrific. I love this ending and it is definitely the one I am the most proud of. That being said, this is supposed to be a choose your own adventure game, and what would that be if there weren't multiple endings? I figured that since my first main ending was sad and disturbing, I would make the other one a happy ending. The problem is that I never really set up any way for there to be a happy ending. This led to the ending having a pretty big deus ex machina, which isn't exactly great storytelling.

Scheduling

Overall, I am pretty happy with how this game came out. I worked really hard on it as I will now talk about. Ever since I started working on this project, I have tried my best to keep to a consistent schedule. I tried to work on the project for a minimum of one hour a day from Sunday through Thursday. This would give me at least five hours a week to work on the project. I decided to not work on the project, or any homework for my classes, at all on Friday and Saturday. I decided to do this because I think it is important to give yourself a break from working and dedicate some time every week to relax and not worry about anything. This also gives me something to look forward to every week

which makes me more motivated to work on the project during the days that I set aside to work on it.

I didn't always keep this schedule. Oftentimes I would find myself wrapped up with homework or tests in other classes and I wasn't able to dedicate an entire hour to the project. Other times I didn't feel inspired enough to work on it. Other times I lost track of time while working on the project and ended up spending three or four hours at a time on it. This usually balanced itself out, and on average I worked on the project for about five hours a week. That was how it went until Winter Break.

Once Winter Break started, I knew I would need to work on the project for longer than an hour a day. I knew that the deadline was slowly approaching, and if I didn't work on it a lot on Winter Break, I might not finish in time. I didn't set myself an exact time limit, but I usually just worked on it for as long as my brain would properly function. After a couple of hours, I would find myself unable to think clearly, and that is when I knew I needed to stop for the day. I didn't keep track of exactly how long I was working on the project, but it would probably average out to three hours a day. Just like before, I took Friday and Saturday off, so I was probably averaging 15 hours a week. I did this consistently every week except for Christmas. During this week, I had relatives over, and I didn't have any time to work on the thesis at all.

I was able to get a majority of the game done over Winter Break. Before the break started, I had around 2,000 lines of code. Once Winter Break was finished, I had over 9,000. I was able to finish the game over the break, and the game ended up being 9,328 lines of code. To put that number into perspective, I have had to do plenty of projects in my classes that involved coding. Out of all of those projects, the longest one was

somewhere around 500 lines of code long. This meant that this game was almost 20 times as large as any other project that I have ever worked on. It is also worth noting that that number only includes the actual game itself. It doesn't include all the time I put into making characters, resizing images, planning, meeting with Professor Murphy, or writing the actual thesis. What I am trying to say is that I worked really hard on this project and I have easily put more time and effort into this than I have on any other project in my entire life.

The Different Mediums of Choose Your Own Adventures

Now that I have talked about my entire process with the project, I think it would be beneficial to briefly discuss the differences between the different mediums that are used to make choose your own adventure stories. The three mediums that I will compare are visual novels, video games, and film. The first medium is visual novels. This is of course the medium that I chose to create my choose your own adventure story. What makes this different than all the other mediums of storytelling is just how accessible it is to create a story using this medium. I've detailed throughout this entire thesis about just how simple this whole process was. Sure, I ran into some difficulties here and there, but overall the process of creating my own visual novel was pretty simple. While I do have some experience with coding, I feel like Ren'Py is easy enough to use that anybody could create their own visual novel if they were willing to put in the time and effort. An example of a professionally made game that was created in Ren'Py would be *Doki Doki Literature Club*. While this game does look a lot better than mine in a variety of aspects, there are quite a lot of similarities between the two that I noticed while looking at both of the games. The biggest similarity is the UI, or the user interface. This is simply the menu

that the player sees and uses when they are playing the game. An example of this would be the “save” button that a user would click when they wanted to save their game.

Comparisons between my game and *Doki Doki Literature Club* can be seen in Figure 7 and Figure 8.

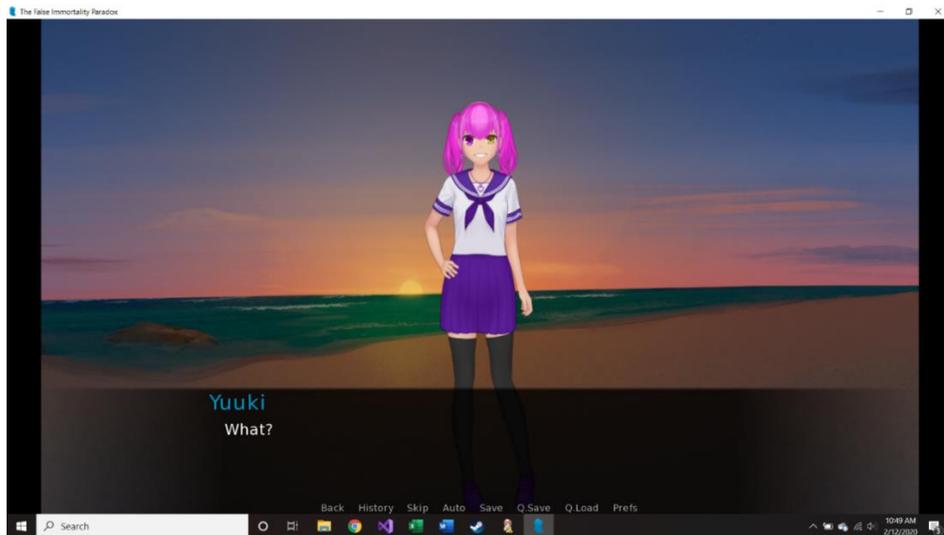


Figure 7: A comparison between *The False Immortality Paradox* and *Doki Doki Literature Club*

As you can see, both games feature a “History”, “Skip”, “Auto”, and “Save”, button. These were made by Ren’Py, and neither of us had to do any coding to get these buttons to work.

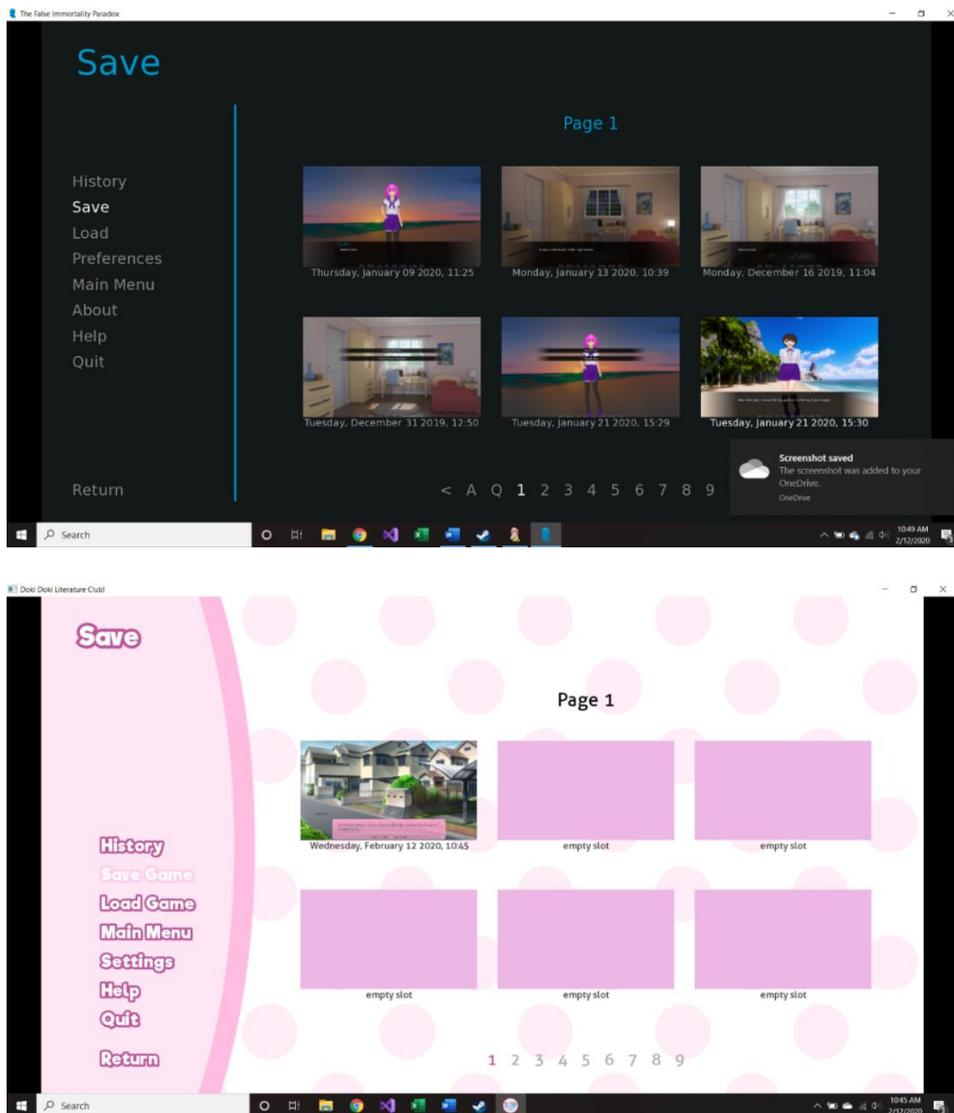


Figure 8: A comparison of the save screens of *The False Immortality Paradox* and *Doki Doki Literature Club*

These two images show the screen that comes up when the user clicks the “Save” button. Both of these pages have a lot of similarities, since they were both made in Ren’Py. These comparisons between the two games show that anybody could make a visual novel. I had no experience going into this project, but I was still able to make a game that looks somewhat good. You can only imagine what kind of game could be possible if someone with a lot of time, money, and experience were to make their own game. Another thing worth mentioning is that a visual novel doesn’t even need to be a choose your own adventure game. If the creator so chooses, they can tell a linear story using the visual novel medium.

Some positives of a visual novel that come from the players end are as follows. First, the player is allowed to take as much time as they want. The player can read through the dialogue as slowly as they want and, at least in my game, they can take as long as they want to make the choices that impact the story. This is a luxury that isn’t always available in other mediums such as video games, as often times the player only gets a few seconds to choose an option. Another positive aspect is the availability of these games. Visual novels take very little processing power so pretty much any laptop or computer can play them. Some video games that aren’t visual novels can take quite a lot of processing power so they can’t be played on all computers.

The next medium for choose your own adventures are video games. Technically, visual novels are video games as well, but when I use the term video games, I am specifically referring to the way that Telltale Games makes choose your own adventure stories. In these games, there are sections where the player is free to move their character around and talk to people in the story or interact with certain objects. When the player gets to

make choices, there is usually a strict time limit in which they get to choose a specific option. I don't think this is a negative, as it can make the player more immersed in the story as they have to make a choice in the same amount of time that the character would actually have to make the choice. For example, if two characters are being attacked by zombies and the main character only has time to save one person, then the player will only have a few seconds to choose who to save. This adds purposeful stress on the player which helps them understand how the main character is feeling in the moment.

The main difference between making a visual novel and a video game is the amount of time, money, and experience that it takes to make a video game. While making a visual novel can be somewhat simple, the same cannot be said for making a full-fledged video game. Visual novels can be made by one person, as I have shown, but making a video game that is as big as the ones that Telltale makes can take an entire team of experienced programmers. This makes traditional video games a lot less accessible to the average person, as most people couldn't just make a video game on their own that is of the same caliber of the ones Telltale makes.

The last medium that I will discuss is film. While choose your own adventure stories in film is a rather new idea, Netflix has already released one that quickly became quite popular. *Black Mirror: Bandersnatch* is a spin-off film based on the popular television series *Black Mirror*. What makes this film unique is that the viewer gets to choose where the story goes by selecting different routes for the main character to take. Since the viewer gets to choose different routes, this film is in fact a choose your own adventure story. Having a choose your own adventure story take place in film offers a couple of unique attributes.

The biggest attribute that sets this film apart from others, is that it was filmed in live action and uses real actors. This means that they had to get actors to act out all of the different paths that the player could take. In visual novels and traditional video games, the characters are created and the whole thing is animated, so a live action choose your own adventure is an anomaly. Another thing that sets this film apart is how it is played. Visual novels and video games are downloaded from the internet so the player can play them wherever they like. This film is streamed off of Netflix, so there is no download required. This makes this film much more accessible to the average person, as there is a lot less that they have to do to start the game. Also, as long as you have a Netflix subscription, this film can be played for free.

As you can see, all of the different mediums that a choose your own adventure game can take have different attributes that set them apart from each other. The most important thing to realize is that there isn't any one medium that is better than the others. They all have their challenges and benefits, but they are all equally valid ways to tell a story. It is simply up to the creator to decide how they want to tell their story. If you are completely new to programming and don't have much resources, then creating a visual novel will probably be your best bet as Ren'Py is a free and helpful tool that anybody can learn.

Conclusion

In conclusion, this entire process was a valuable learning experience for me. I learned all about how hard it can be to work on a long-term project due to all of the different steps you have to take to make sure you finish on time. I learned that not everything is going to go your way, and that it is of the utmost importance to know when to improvise.

Finally, I learned the importance of working on a consistent schedule as that will help you keep yourself on track, so you aren't tempted to procrastinate.

Bibliography

Corcoran, Leaf. "Download the Latest Indie Games." *Itch.io*, 3 Mar. 2013, itch.io/.

Jong, Bram de. "Freesound." *Freesound*, 5 Apr. 2005, freesound.org/.

Kimball, Spencer, and Peter Mattis. "Gimp." *GIMP - GNU Image Manipulation Program*, 2.10.14, 31 Oct. 2019, www.gimp.org/.

"Lemma Soft Forums."  *UNCLE MUGEN'S FREE VN (OELVN) RESOURCES*  - Lemma Soft Forums, Studio Mugenjohncel, 27 Sept. 2012, lemmasoft.renai.us/forums/viewtopic.php?f=52&t=17302.

MacLeod, Kevin. "Music and Also Graph Paper." *Incompetech*, incompetech.com/.

"Mannequin." *Mannequin by AR14*, 22, 14 Feb. 2020, ar14.itch.io/mannequin.

"Microsoft Visual Studios." *Visual Studio IDE, Code Editor, Azure DevOps, & App Center - Visual Studio*, 16.4.3, 14 Jan. 2020, visualstudio.microsoft.com/.

Rothamel, Tom. "Ren'Py." *The Ren'Py Visual Novel Engine*, 7.3.5, 17 Oct. 2019, www.renpy.org/.

Slade, David, director. *Black Mirror: Bandersnatch*. Netflix, 28 Dec. 2018, www.netflix.com/title/80988062.

"Visual Novel Backgrounds by Selavi Games." *Itch.io*, Selavi Games, 19 Apr. 2017, selavi.itch.io/visual-novel-backgrounds.

“Visual Novel Maker.” *Visual Novel Maker*, Degica, 16 Nov. 2017,
visualnovelmaker.com/.

Yokotani, Masahiro. *Re:Zero – Starting Life in Another World*, 4 Apr. 2016.