

Design Journal

A document detailing my Text-based Adventure game



2023/27/06

Ariana jacpasad

2396624

|  |  |  |
| --- | --- | --- |
| Chapter | Name | Page |
| 1 | The idea | 1-4 |
| 2 | Plan | 5-6 |
| 3 | How to play | 7 |
| 4 | Reflection | 8 |

The idea

I wanted to make a text-based adventure and when researching I was met with multiple options. I was met with different definitions of text-based games such as Zork-like games, Doki Doki Literature Club, Stories Untold, and Ace Attorney. I was familiar with all these games besides Zork and therefore wanted to go a very traditional route where I used very little graphics and wanted to emulate the game genre from the 1970s, and 80s and majorly took inspiration from Stories Untold.

A text-based adventure game is a genre of video game that relies heavily on textual descriptions and commands to progress through a narrative and interact with the game world. Unlike modern graphical games, which use visual elements to convey information and provide immersive experiences, text-based adventure games primarily use written text to describe the game's environment, characters, and events.

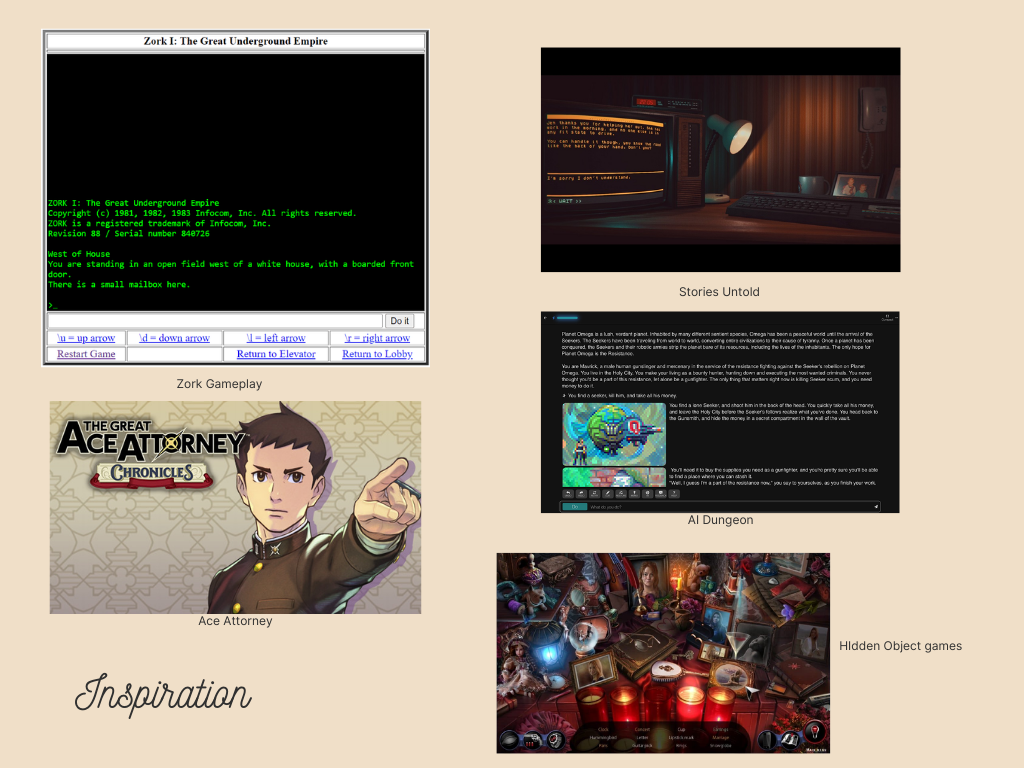
In a text-based adventure game, players typically read descriptive passages that set the scene and encounter various challenges, puzzles, and choices. They interact with the game by entering text commands through a command prompt or a text interface, such as "go north," "take the key," or "attack the monster." The game interprets these commands and responds with further descriptions, outcomes, or prompts for more actions.

The game world in a text-based adventure is often presented as a series of interconnected locations, which the player can explore by moving between them. Players can examine their surroundings, search for items, interact with non-player characters, solve puzzles, and make decisions that affect the outcome of the game. The narrative unfolds through the text descriptions, dialogues, and the player's choices, leading to multiple branching paths and possible endings.

Text-based adventure games encourage imaginative thinking, problem-solving, and exploration. Players must carefully read and interpret the provided text to understand their surroundings, gather clues, and make informed decisions. These games can vary widely in terms of themes, settings, and complexity, ranging from simple puzzles and linear narratives to complex, open-ended worlds with multiple storylines and intricate puzzles.

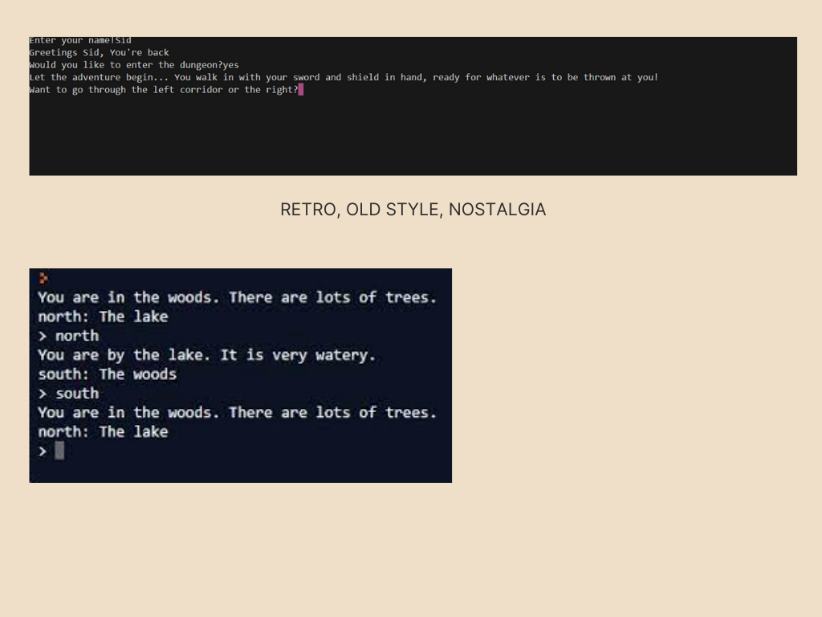
Popular examples of text-based adventure games include the early interactive fiction titles like "Zork," "Adventure," and "Colossal Cave Adventure," as well as modern successors like "Choice of Games" and "Twine" games.

A screenshot of a computer

Description automatically generated with medium confidence

As mentioned before, I wanted to follow the older versions of text-based games and therefore went from this aesthetic. Where I will be utilising the command prompt to make this aesthetic. I had initially looked into engines like Renpy and Quest and decided against them as they didn’t have exactly what I wanted for the game. I also thought of making the game using pygame but I could not figure out the software and therefore decided to use the command prompt because it gives the exact aesthetic I want.

As I wanted to be as close to the aesthetic I added in ASCII art as many old games have. Text-based games often come from independent developers and would be using the basics of a computer due to the technology at the time. This is why I thought to utilise the command prompt to bring back some of that from when these games were made on glass teletypes.

I added an external link such as the 8-bit music as to get the player both pulled out of and into the immersion. The player now has context however, the act of pressing the link pulls the player out of concentration and this was to lower subvert their expectations.

The retro game aesthetic refers to a style that evokes the visual and audio design of video games from the past, typically the 8-bit and 16-bit eras of the 1980s and 1990s. It is characterized by its pixelated graphics, limited colour palettes, chiptune music, and simple sound effects. The retro game aesthetic has gained popularity in recent years as a nostalgic tribute to the early days of gaming and has influenced modern game development and various forms of media.

The graphics of retro games are often pixel art, which uses square blocks of colour to represent characters, objects, and environments. Due to the technological limitations of older hardware, pixels were visible and defined the visual style of the games. The pixel art in retro games has a charming simplicity and relies on the creative use of limited pixels to convey details and emotions.

Colour palettes in retro games were constrained by the capabilities of early gaming systems. These palettes often consisted of a limited number of colors, usually bright and vivid to compensate for the lack of graphical fidelity. The result was a distinct and recognizable colour scheme that is synonymous with retro games.

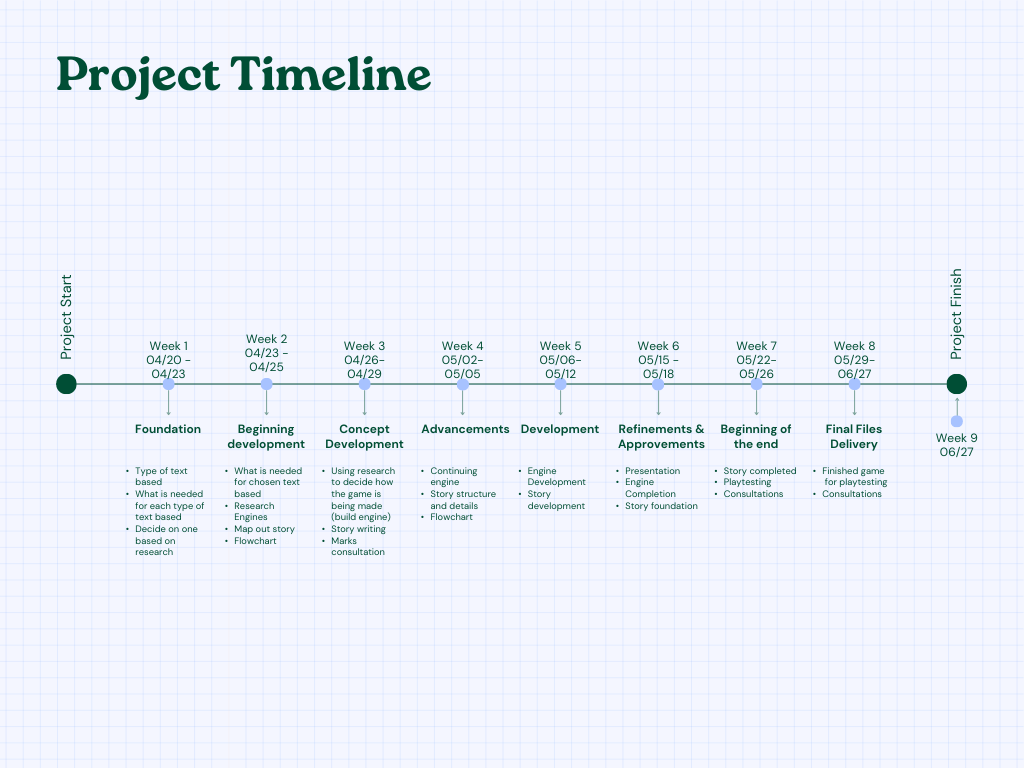
Chiptune music is another defining aspect of the retro game aesthetic. It refers to the electronic music compositions that were created using the sound chips found in older gaming consoles and home computers. Chiptunes have a distinctive melodic and rhythmic quality and often evoke a sense of nostalgia for gamers who grew up playing classic titles.

In addition to pixel art and chiptune music, the retro game aesthetic also encompasses the simple sound effects commonly heard in early games. These effects were often generated by the same sound chips and were limited in complexity. Beeps, boops, and blips were used to represent various actions, such as jumping, shooting, or collecting items, contributing to the overall nostalgic appeal.

The retro game aesthetic is not limited to visuals and audio alone but also extends to gameplay mechanics and design principles. Retro-inspired games often embrace straightforward gameplay, challenging difficulty levels, and a focus on skill-based mechanics. They may draw inspiration from classic genres such as platformers, shoot 'em ups, and adventure games while adding modern twists and innovations.

Overall, the retro game aesthetic captures the essence of early video gaming and pays homage to the nostalgia and charm associated with those formative years. It has become a beloved style that continues to influence contemporary games, art, and media, appealing to both long-time gamers and new enthusiasts seeking a taste of gaming history.

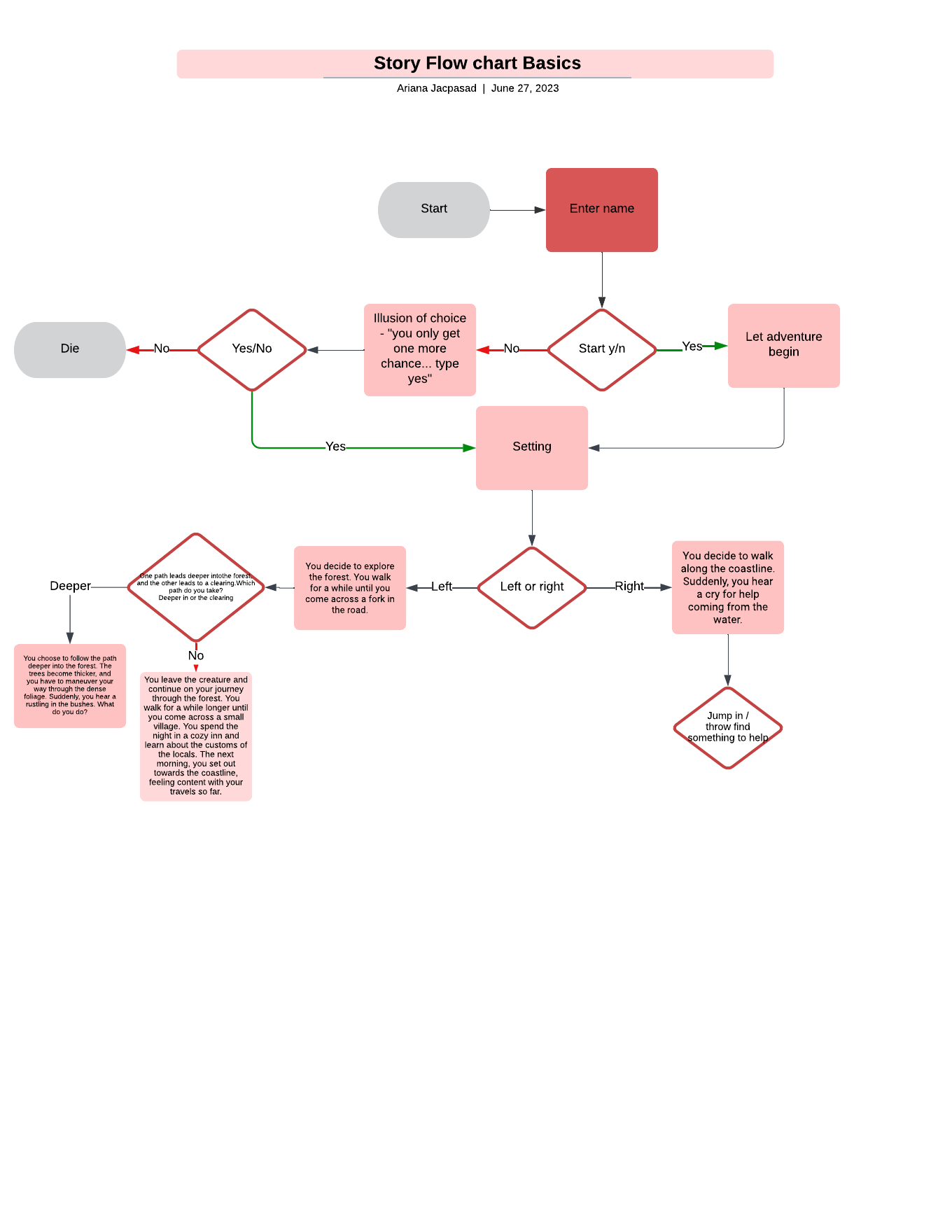
The plan

This is the timeline I created of what I wanted my progress to look like. I followed the dates and expectations I had of myself however, the game isn’t huge, but it does become complicated so where I wanted to expand on the story, I rather made it short and tried to fix what I could, for example, looping issues.

Week 1

I started off researching text-based games and wanted to make a typing game. I looked into Zork and Stories Untold. I wanted to challenge my writing skills and learn a new language at the same time. So, I chose inspiration from both.

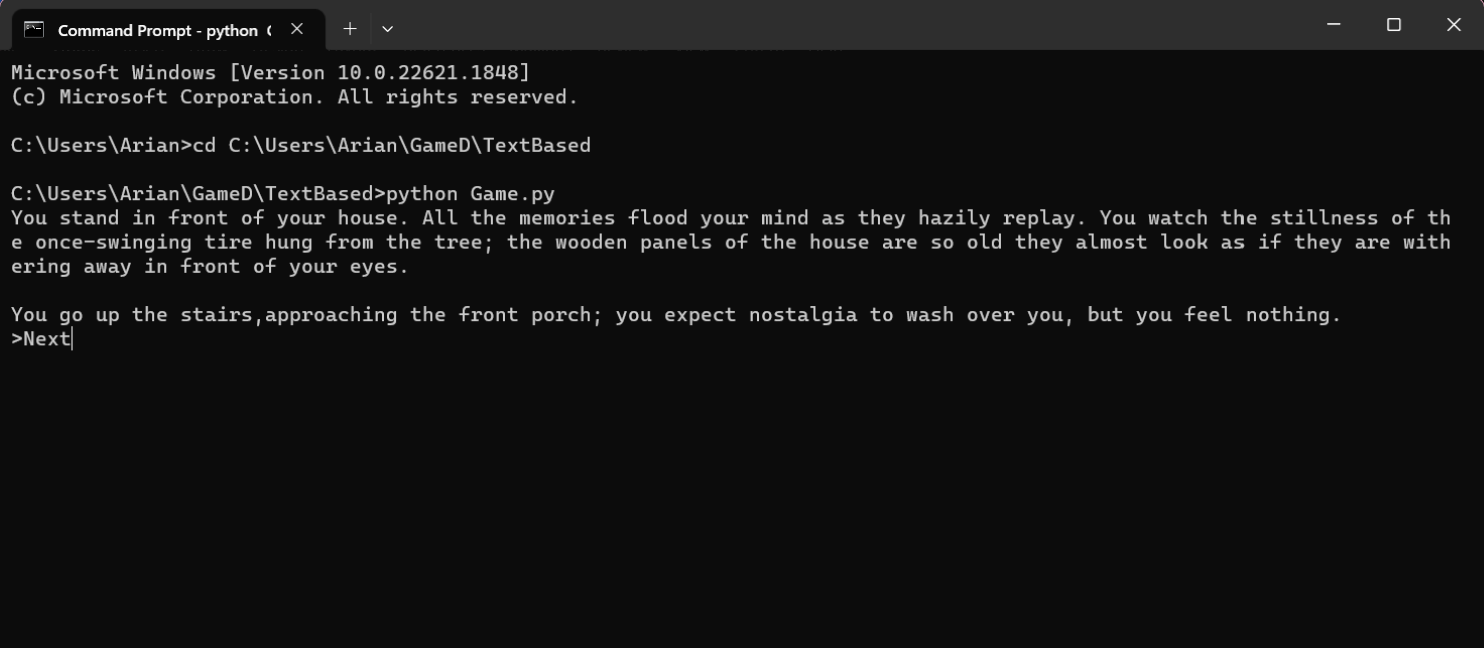
Week 2

I figured out what was needed of me to be able to make these types of games. I needed a typing engine and a story using a flowchart. The flow chart was difficult for me to use as it was overwhelming for me to understand so I used it to begin the story and mapping of my narrative. I also deviated from the flowchart to make the game make sense.

The rest of the weeks were just working on the story and figuring out where to make the game as my decision kept changing throughout the course of making it. I couldn’t decide what to use because I wanted to have the right aesthetic. These weeks we're just slowly building the narrative. While the game seems as though it doesn’t require much work, writing each path takes time as well as creating the loops.

How to play

You open your command prompt, copy the file path of the game from your computer after downloading it and type in “cd (path)”. Path refers to the location of the file which you simply just copy from the file and paste into the command prompt. After that to begin the game you type “python Game.py” into the command prompt and it should play the game. I provide the player with multiple options for their answers in terms of the format, for example, the player can type “yes”, “Yes, or “YES”. I also provided options so players can see what they are meant to type however, I also allowed different variations of the phrase. It is necessary to place the file in local storage rather than something like one drive. Here is an example.



Reflection

I faced many difficulties while making this project, from figuring out what software or engine to use to make my own, learning Python and doing the tedious task of looping the game to find errors and loop issues.

I struggled to choose where to make my game as it became difficult when I wanted to specifically make a typing game. Most platforms provide a storytelling template rather than a Zork template. Hence why I decided to make the game the way I did. I believe there are still some looping issues I have missed where the player is taken to the wrong option. I also think the format of my writing makes it difficult to read on the command prompt. The black-on-white makes it hard to read as well as there are formatting issues for the ASCII art which will distort based on how big the player’s window is. I figured this could be avoided if I used something like Avocado Navigator which I started to use but decided against it.

I also had difficulty writing the story. I feel as though I made the mistake of going between past and present tense which can happen when you speak using second person narrative. The actual narrative I feel falls short towards the end because I started to just write walls of text rather than have the player interact with the game. I think this could be fixed if I had worked more on the story rather than stressing about the platform to use and having to learn a whole new software. Looking back now I would use something like pygame or try to make the game in unity to make things easier with formatting and time I lost learning the software, although I now know the basics of Python I would like to continue using that language to learn it better.

I think this type of game would benefit from more extensive playtesting with a diverse group of people. I feel as though this is why my game isn’t as good as it could have been because I missed this crucial thing. I would like to use more people rather than the few I did because they were not willing to play a game with this much reading.

I also realised that, as someone with ADHD, these types of games become incredibly difficult to play yourself as you become impatient quickly. Although I can see this happening with just anyone. This also is one of my biggest obstacles when it comes to creating anything as I procrastinate and struggle to concentrate. I would like to develop a better working ethic to avoid this in the future.

I feel as though this is something I’m interested in exploring more, and refining. I would like to make bigger stories and learn Python better so that I can develop a more efficient way of coding. I enjoyed the experience of writing a narrative as I don’t get to often do this and would like to get better at this skill. I often do the art for my games and this time wanted to focus on more coding and narrative and that is why I picked such a simple looking aesthetic, I still used ASCII art because research showed that it has been used in even the oldest of text-based adventure games from the 1960s. As much as I prefer to work alone so that I can hone my skills in, I feel as though to explore this type of game to the extent that I want I will need more people to help me. I hope to better my skills in coding as that’s what I struggle with and shy away from the most. This is why I struggled with finding a platform to use because I was intimidated by the new flow of work.