CIT419

Gaming and Simulation Capstone

Final Project Report

GHARLIERA

Hallie Harriman

Fall 2020

12/9/20

Pennsylvania College of Technology

TABLE OF CONTENTS

1 VISION STATEMENT

1.1 GAME/SIMULATION LOGLINE

Single handedly stop a catastrophe or meet your untimely demise, your fate is in your hands within this Visual Novel – Ready to Rebirth(?)

1.2 GAMEPLAY SYNOPSIS

- Goal – The overall experience is to provide entertainment through the ability to make choices throughout the game to progress to the end.
- Uniqueness – The overall style and setting of the game is vastly different from most other currently published visual novels. It provides a sense of fresh air and new feel to the genre in comparison to its competitors.
- Mechanics – The core mechanics of the game are cycling through dialogue with either SPACE or MOUSE1, and selecting buttons for choices.
- Setting – The setting is a dystopian cyberpunk city. Most of the settings present will be inside buildings, or amongst the streets within this city.
- Look and feel – The look and feel of the game, play wise, follows the same flow as other common visual novels. The game is supposed to provide the feeling of control and inquisitiveness within the game, based on the play and story.

1.3 AUDIENCE AND SYSTEM REQUIREMENTS

1.3.1 TARGET AUDIENCE

The target audience of the game is for anyone above the age of 13 because of the presence of violence and use of strong language. By ESRB Rating, this would be T for Teens. There is no specific gender or location that the game is tailored to. Although, the only language option presented in the game is English so, geographically, it may target English-speaking countries more than others.

1.3.2 PLATFORM

The platform this game runs on is PC. This is because of scope. If time permits it later, game consoles and mobile phones may be addition platforms this game will be available to play on.

1.3.3 SYSTEM REQUIREMENTS

For Play: For play, the minimum recommended specs are as follows: Memory: 2 GB GPU: NVIDIA GeForce 6100 CPU: Intel Pentium 4 2.00GHz File Size: 1GB OS: Windows 7+

These choices were made since the game is not intensive so minimal system requirements will be needed. *Refer to 1.3.2 PLATFORM* for information on the choice of Windows7+.

For Development: For development I have used Unity Version 2020.1.11f1 and Adobe Photoshop for programming and UI design. Unity is an open source software tool and I have free access to Adobe Photoshop through my school account.

The specifications of the machine I will be using for this development are as follows: Processor: Intel(R) Core™ i5-7300HQ CPU GPU: NVIDIA GTX1050 RAM: 8.00 GB Windows 10.

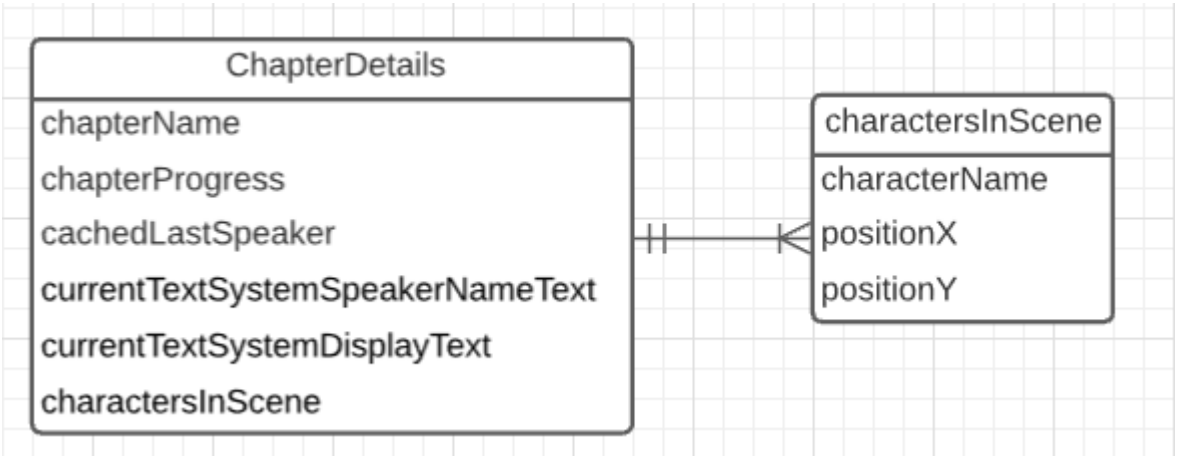
2 CAPSTONE REQUIREMENTS

2.1 PROGRAMMING COMPONENT

The programming necessary for this game was implemented through the use of a state machine, buttons, sliders, and text-scrolling. The state machine was used to transition between multiple different menus within the game. The buttons, sliders, and text-scrolling are for HUD, UI, and scene-changing capabilities. Choices will be made by the player clicking on a button that affects which dialogue is represented by changing the current chapter.

2.2 DATABASE COMPONENT

The database component is implemented through the use of saving the current state of the game so the player never loses their place in the story while playing. The components that saved are character details, text details, backgrounds, and audios. The database was created and implemented into Unity through the use of SQLite. The database fields are also saved locally to easily screenshot and transfer saves. *See INFORMATION ASSURANCE AND SECURITY COMONENT for information on how security was handled.*



2.3 WEB COMPONENT

The web component is purely supplemental with an option to download the game. The website will present additional information for the characters and story, showcasing previews of visuals present within the game, as well as providing a short summary and Frequently Asked Questions (FAQ's).

2.4 INFORMATION ASSURANCE AND SECURITY COMPONENT

Security is present within the use of "security-by-design". An example of this is by only making variables public and serializable when absolutely necessary. The locally saved database fields are also made secure as to not allow the player to tamper with any information they should not have access to. This is done through the use of JSON symmetrical encryption that simply garbles the text so the player cannot change values to cheat.

3 RESEARCH

3.1 RESEARCH COMPONENT

The research component of this project is to showcase knowledge of how to correctly use a state machine to properly. This was originally going to be for the state of the player, which was no longer needed, so it was later changed to add animations in when transitioning between menus. This provides a smoother feel when opening and closing the setting and save and load menu. When there are multiple UI elements and when the game is purely 2D, it can be difficult to differentiate between aspects on the screen without it being overwhelming. This is why the choice of adding animation transitions was incorporated.

3.2 RESEARCH SOURCES

Unity Technologies. (2020, November 24). *Animation State Machines*. Retrieved from Unity Documentation: https://docs.unity3d.com/2019.3/Documentation/Manual/AnimationStateMachines.html

Unity Technologies. (2020, November 24). *State Machine Basics*. Retrieved from Unity Documentation: https://docs.unity3d.com/Manual/StateMachineBasics.html

4 REFLECTIONS

I believe the work I completed increased my interest in my identified passion. Working on this project has been helpful in showcasing which aspects have worked well for designing and programming visuals novels. The work I have put in towards this project can now either server as a template for any future visual novels I design, or can be continued later on. There is still much more I can learn to apply that will help add more functionality to the game genre. For what I have learned, I can use the knowledge I gained, not only visual novels, but almost any type of game. This is because of increased experience of state machines and implementing databases from working on this project.

There was a lot I enjoyed about this project, as well as a lot I was not the most favorable of. The parts I enjoyed the most were writing the story, designing the flow of the game, and researching topics I was not familiar with. The part I liked the least was having to upgrade my version of Unity halfway through development, causing me to have to go back through rewriting lines of code to fix errors. Now during reflection, I realize I should have just started with the current version of Unity because I had truly no reason to be using an older version. I was thankful enough to have managed my time correctly so this didn't become an issue later on, but was more so just tedious at the moment.

Advice that I would give, and have given, to students in line to take this course is to start thinking of your game now. It is better to have an idea of what you would like to work on before the semester even starts. I would recommend to begin designing portions of the game ahead of time. Even if it is something as simple as a game diagram that displays the flow of the game, or starting one of the required components. This is all extremely helpful in time management later since you will only have one semester to work on the entire project. So, if you run into any issues, like myself, you will have more than enough time to address them. The last piece of advice I would give, is to always make multiple saves of your project. You never know what can go wrong, so even myself had around 5 different saves of my project by the time of completion.