Design Document – Final

The Plan:

Game Name:

Choose Your Farewell

Team:

Benjamin Perry & Daniel Blatner -> [bjperry@wpi.edu](mailto:bjperry@wpi.edu) / [dblatner@wpi.edu](mailto:dblatner@wpi.edu)

Genre:

Simulation / Visual Novel

Game Description:

There is a calendar counting down the days you have left (to live). On each day the player must make a choice and that choice will impact their legacy. The goal is to complete the game with the best legacy (or desired legacy).

Technical Features:

* A choice Object the player is able to select via keyboard input once the game begins (represented as Text). We have decided to implement a total of a hundred-sixty Choices. These Choices are stored in a single .txt file and parsed in the ChoiceManager. The ChoiceManager is also the system that selects Choices to spawn in a given day. Each day spawns a total of four unique Choices. Choices are displayed above and below the Day sprite, as that is what looked cleanest during thorough testing. They can be selected on a given day by pressing ‘WADS’, where left to right in WADS corresponds to the top Choice down. Selecting a choice by pressing W,S,A, or D will tell the PointsSystem to allocate a certain amount of points to some categories, which ultimately depends on the Choice selected. Points received are dictated and assigned by the ChoiceManager
* Seconding off the Choice Objects, the game also must have Points attributed to each choice to affect the end legacy the player receives (Points System). The PointsManager simply holds all the possible end legacies and sub-categories to which the end result will depend on; to elaborate basically examples of sub-categories would be Cruelty & Kindness, where excellent scores in both could lead to a certain legacy, while excellent scores in Cruelty and some other subcategory could lead to a different legacy. The PointsManager calculates the end total of all sub-categories then determines the legacy based on the sub-categories the player scored the most points in. The sub-categories that correspond to certain legacies, as you can imagine, were determined primarily by our common sense and what seems appropriate. After determining the legacy (in the TitleManager which works with the PointsManager to calculate the end results), it is sent over to the LegacyResults object to be displayed once the game finishes
* A Day Object that works as the system for counting down the days the player has left ; Houses a ViewObject to show the number of days the player has left (DayNumDisplay in code), a ViewObject to present the prompt asking the player “What they are going to do today” (Prompt in code), a sprite shown in the middle of the choices, and of course the four choices (along with the standard Object attributes. Setters are necessary to individualize Days and spawn other objects with the creation of the Day. The Day sprite is swapped depending on the day number as the Day system essentially tells the visual story of an evil man who nukes the world (hard to represent in ASCII, but works)

Artistic Assets:

* \* NOTE \* We didn’t want to introduce many bugs while working with art since the text files were an absolute pain to decode, which was weird considering the engine we used displayed all the Dragonfly sprites perfectly fine ; if you are reading this professor, feel free to comment on the text file sprites and look at how awkward our sprites had to be written in the text files to appear correctly in the game (As an aside we will also include a ‘spritesNOTdecoded’ folder in the final zip so you can see the art without having to boot up our game to examine it)
* Day Sprites -> Original Plan was to make a unique Sprite for each Day; I didn’t do that for the reason listed above in the NOTE ; Instead, we ended up creating a story using the sprites I did end up making, where an evil man is pictured to be destroying the world with nukes. Take that as you will. The sprites featured in the Day system include four sprites of the man (each supposedly supposed to be creepier than the prior shown version), a cloud (for smooth transition between man sprites), two nuclear clouds, and an alarm clock ; the Day system swaps between blue and red sprites each Day. Each Sprite used in the Day system is scaled to fit between the top and bottom set of choices and is a modified version of a free source asset from this link: [ASCII Art Archive](https://www.asciiart.eu/) ; Of course, we doubled checked to ensure that the assets were free source, but of course I modified them beyond what their original look anyways (and I don’t just mean the funky format I had to use to get them to display in game)
* Start Screen -> Since the title was larger than Saucer Shoot, naturally the start screen had to be larger. I didn’t opt for much larger though, as we were more so looking to simplicity with this piece of art. We also wanted to display the controls on the start screen, so we added more lines below the title as well, and we think the end result looks pretty nice
* End Screen -> A simple, but elegant screen that displays the player’s ending legacy. I also created an amazing tombstone sprite which I couldn’t quite find a reference for, but I needed seem to need one anyways I guess

Implementation Plan:

* Created Visual Design Plan
* Implemented Day System (core)
* Added Art & Audio Support
* Added Temporary Legacy Results screen & functionality
* Implemented Choice System
* Implemented Choice Legacies

List of Legacies For Reference:

* Absolute Legend
* Loser
* Unsung Hero
* Average Joe (Or Whatever Your Name Is)
* An Enigma
* A Gentleman
* An Unfulfilled Man
* The Main Antagonist
* Etc. or less if necessary

Distribution Of Work:

Programming -> Git Redo -> Day System (Ben); Points System (Dan) -> collab if needed;

Audio -> Start Screen (Ben); Choice Select (Dan); End Screen (Ben); Calendar Music (Ben) -> collab to agree of audio

Art -> Calendar Art, Start Screen, End Screen -> Ben; -> Dan was busy & to be honest the art was really, really aggravating, so I saved him the trouble since I had already spent hours fighting the sprite text files

Writing -> Choices (Dan); This Document (Ben)

Schedule: (Maybe Not Relevant)

Friday -> 11 – 12 Done

Saturday -> Ben started at 12 pm, Dan joined at 4 pm

Sunday -> Ben started at 10 am, Dan joined at 8 pm