Burning Choices

Project Designers: Ethan Seiber, Jamel Warren, and Hunter Haislip

Abstract

In the course of the next couple of months we will develop and present a 2D game named "Burning Choices". The game will primarily focus on the adventure genre. It will include multiple levels and hopefully branching story depending on the time constraints. The game took inspiration from Bandersnatch, a fictitious game in the Netflix original series Black Mirror that also included branching story line. The target audience will be those that enjoy classic action arcade style games. Things achieved thus far: we have made the functionality that will allow us to create obstacles, any interaction objects like doors and walls, in our window.xaml, along with this we have implemented character models to represent movement and an idle character.

1. Introduction

The project is a 2D adventure game that will operate around theme of angels, demons and humans. The end goal of the project is to tell a story that branches depending on the actions of the player. The audience for this project would be those who enjoy retro style games. The audience will play the game and leave feeling satisfied/ entertained if we do our jobs right. The reason why we have chosen this project is thanks to different interests ranging from a love of video games to things we enjoy in our own lives and would like to create ourselves. The story will follow the player as they find their way through our world overcoming any obstacles in their way. In doing so they are given the ability to control their own story.

1.1. Background

Throughout the paper these terms will be used in describing certain aspects or situations in the game. 2D graphics, branching storyline and linear storyline. 2D graphics refers to the dimensions of of the art style within the game i.e. 2D graphics is short for 2 dimensional graphics. Mario for the nintendo 64 is a great example of 2D graphics. Branching storyline refers to the story structure and how the story of the game is built. The story that is produced depends on the choice of the player. This means the player could end up with a completely different ending or story leading to the ending depending on what they do in the game. Linear storyline also refers to the story structure within the game. A linear storyline is much like reading a book. From beginning to end the story is static and unchanging with only one ending. This is a team project so we all have our own motivations for deciding to make a game. For both Ethan and Jamel it's the video game aspect of it because we are both fans of video games. For Hunter it's the idea. Hunter received inspiration from the Black Mirror episode Bandersnatch to make the game have multiple endings based on player choice. Hence the title Burning Choices.

1.2. Challenges

Some of the challenges we suspect to run into include programming the interaction between the player and environment, player movement and the creation many levels that are connected to each other so the player may move between levels. This is drastically different from any programming projects we have attempted before. We have learned about gathering user input and manipulating it, but we need to manipulate it in a way that interacts with objects or the player. Creating in game objects like obstacles, doors and keys leads in to the environment interaction challenge. Saved games are a necessity for games which entails gathering the players current place in the world as well as their choices in the story and then being able to use this to load their progress when they start up the game. The last expected challenge is to create our different story lines and make sure they lead to an ending. Lastly is the debugging. This is more of a patience challenge to try and break our game. To address these challenges we will need to do some research on useful libraries to help provide pre-defined functionality or create our own functionality.

2. Scope

The project will be finished when we have at least three in game levels, a linear storyline, can save, and the world builds and functions properly. The stretch goals we have set for this project are the branching storylines and more than three in game levels.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Starting the Story	Player	Easy	1
2	Game Saving	Player	Med	2
3	Exiting Game	Player	Easy	2
4	Load Game	Player	Med	2

TABLE 1. SAMPLE USE CASE TABLE

2.1. Requirements

The functional requirements for this project were gathered from experience in other games. The functional requirements for games are typically things that make life easier as well as providing a logical order to the project.

2.1.1. Functional.

- The program should load up the game "Burning Choices" and start on the title screen.
- The game will allow for saved games.
- The player can choose from the title menu to do things like load a game or start a new one or quit.

2.1.2. Non-Functional.

- Recoverability- If the game crashes then it can be rebooted and the user can restart using an auto save.
- Maintainability- Given this project is relatively small it makes it easy to find and get rid of any bugs within. This means in the long run being able to maintain this project will be relatively simple especially as we discover and fix bugs.

2.2. Use Cases

Use Case Number: 1
Use Case Name: Starting the story.
Description: The user is on the title screen and wants to start the story of the game.

1) The user will click on New Game to start a new story.
2) The game starts at the beginning of the story.

Use Case Number: 2
Use Case Name: Game saving
Description: The user reaches a point in the game where they wish to save their progress.

1) The user will hit the Esc key to pause the game.
2) A list of menu items will pop up.
3) The game environment will pause allowing the user time to read the items.
4) The user will select Save Game in the pause menu.
5) The game will store the player's story data in a file created by the executable.
6) The game will un- pause and continue where the user left off.

Use Case Number: 3
Use Case Name: Exiting Game
Description: After playing for a while the user will want to quit the game.

1) The user will hit the Esc key to pause the game.
2) A list of items will pop up.
3) If the user selects Quit the game will exit and stop running.

Alternative: User is on the title screen

1) The user will navigate to the bottom of the title screen.
2) The user will click quit and the game will exit.

Use Case Number: 4
Use Case Name: Load Game
Description: The user is on the title screen and wishes to continue where they left off in the story.

1) The user will see a menu item that reads "Load Game".
2) The user will click "Load Game".
3) The game will open the file holding the saved data of where the user last saved.
4) Using the saved data the game will load to the point where the user last saved in the story.

2.3. Interface Mockups

Figure 1. The gameplay so far

Currently we have the ability to add in walls as we want which allows us to start building on our game environment. In this picture the player is attempting to pass through a wall that cannot be passed through. This picture also shows the color and art style that can be expected within the game environment.

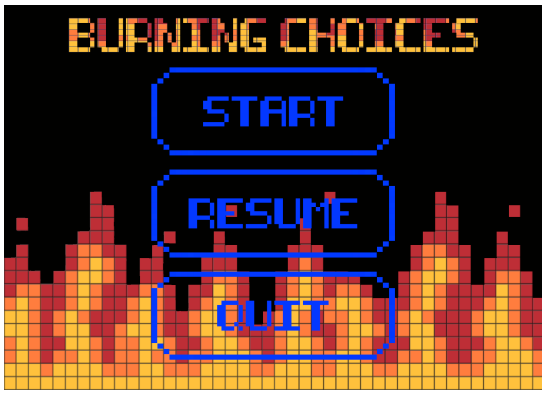

Figure 2. A layout of the Title Screen

This is what we want our title screen to look like once we are finished with the project except we plan to replace start with new game because we recognize how ambiguous it is. This image is the first thing that will pop up when you run the game. It shows you what we envision use cases 1 and 4 to be like. The Resume option, case 4, will allow the player to load a game to continue the story where they left off and the Start option, case 1, will begin a new game from the beginning of the story. The Quit option pertains to use case 3's alternative in which the user is on the title screen and wants to quit the game.

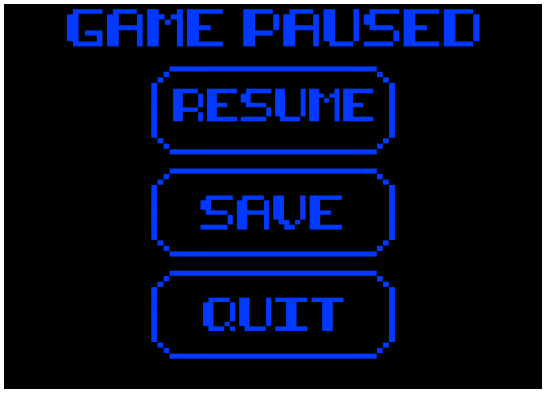

Figure 3. A layout of the pause screen

This is the current version of our pause screen. You know it's the pause screen because it tells you the game is paused at the top. The screen will give you the three options to choose from. The Resume button will un-pause the game and continue the user where they left off. The save option is use case 2 which will record the user's current progress within the story and save it to a file to be loaded later. Then at the bottom of the list is quit which is use case 3. This will close the game entirely.

3. Project Timeline

The time line for this project starts with the requirements phase. It will last from January 2019 to January 2019. This phase is simple and took little time to complete. We gathered what most games have that would be useful in the context of our game.

Next is the design phase. This phase is being developed still. Meaning we started out with our base line ideas we discussed in the beginning, but have modified them as time has gone on. This phase lasted from February through March because we will not be adding any new ideas.

1) We deliberate on the type of gameplay we will have.
2) Discuss the story.
3) Decide a theme for the whole game.
4) Deliberate on some design patterns to use.

Then comes the implementation phase this will take the most time. We plan for this to last from February through April. In this time we plan to have these things finished.

1) Levels designed (In Progress)
2) Obstacle class made (In Progress)
3) The story implemented

This sums up the time line we have set for ourselves because the verification and maintenance steps will be taken care of in the implementation step as we progress in the project.

4. Project Structure

Currently we have decided to use an Abstract Factory design pattern to define and create any objects we will use within the game such as walls and doors.

4.1. UML Outline

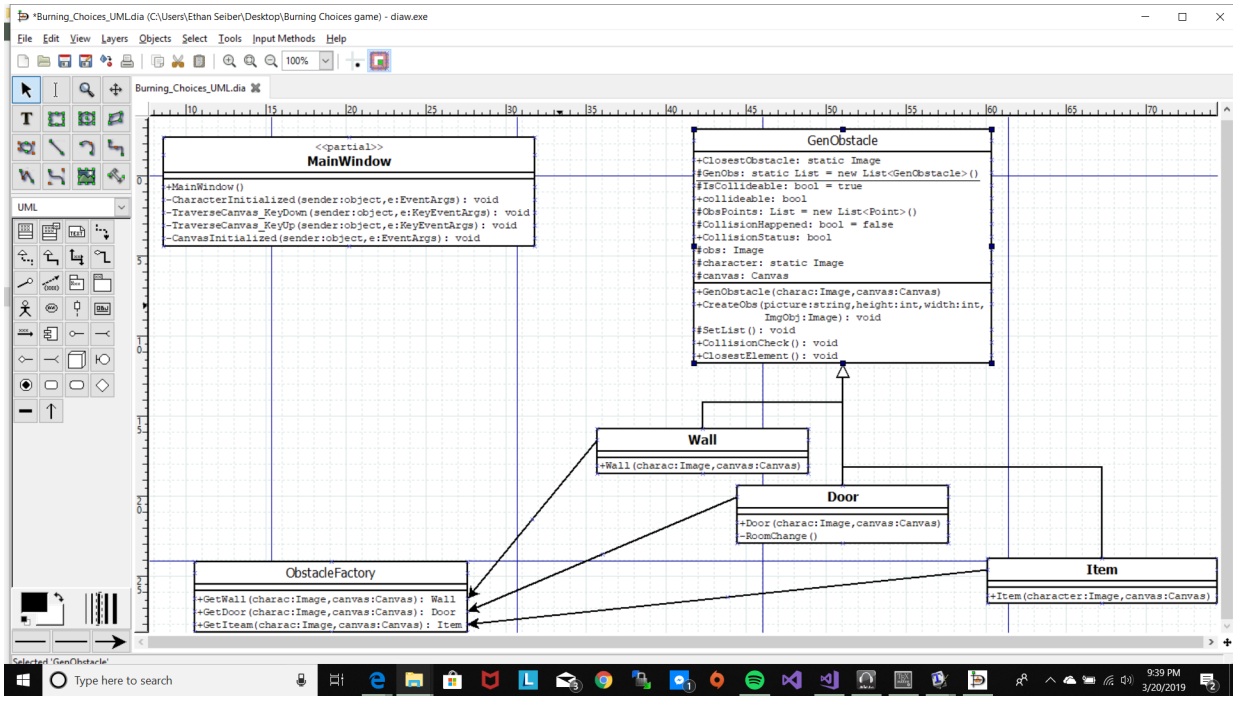

Figure 4. This is the current UML plan we have for our project. It is not finished, but thus far we have a plan for any in game object such as walls, doors and items as well as functionality for our MainWindow class

4.2. Design Patterns Used

Currently we are using the Abstract Factory design patter as one of our patterns.

5. Results

The project stands at being able to freely create walls to restrict the player's movement. This is still buggy when it comes to more than 3 walls is the current thought, but the belief could change as we begin debugging. It could be buggy for another reason.

5.1. Future Work

The next steps we will take are to smooth out the object tracking bugs and work out the implementation for doors i.e. objects that will open another level.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.