**Reflect**

By Jacob Hreshchyshyn

**Table of Contents**

* Game Features……………………………………………………………………………………………………..3
* Game Objects with Hierarchy, Sprites and Sounds……………………………………………...3-4
* Gameplay…………………………………………………………………………………………………………….4
* AI………………………………………………………………………………………………………………………...5
* UI………………………………………………………………………………………………………………………...5
* Hierarchy Chart……………………………………………………………………………………………………6
* Story……………………………………………………………………………………………………………………6
* Difficulty Graph……………………………………………………………………………………………………7
* Timeline………………………………………………………………………………………………………………7
* Concept Art/Pixel Art…………………………………………………………………………………………...8
* Appendix A…………………………………………………………………………………………………………..9

**Game Features**

Reflect will feature two characters that you control at the same time. They are separated from each other by a line representing a mirror placed in the center of the screen. The characters will be unable to cross the line. Additionally, obstacles such as walls and locked gates will be used to prevent the characters from moving. When either character (the actual face or the reflection) stops moving, the other character stops moving as well. In order to traverse past locked gates, key objects will also be used throughout the levels to unlock these gates.

There will also be buttons that the player characters can press to unlock certain gates. There will be two types of buttons. One button will unlock a gate without needing to be pressed constantly while the other button requires to be pressed constantly.

There will also be turrets that shoot bullets. These either target the player characters or shoot in one direction. Bullets that are fired from the turrets can interact with buttons, allowing for clever puzzles.

The player characters simply need to reach a goal point at the end of the level to progress to the next level and learn more of the story.

**Game Objects with Hierarchy, Sprites and Sounds**

There will be two separate player characters. They will not need to inherit another object’s properties since the vertical controls of the characters are opposite to each other. That is, if one character moves up, the reflection will move down. These characters will have the following sprites:





The red sprite is the reflection and the yellow sprite is the actual face of the character. As of now, no plans exist for adding sounds to these character objects, though an addition could be made if time allows.

Additionally, various physical barrier objects will exist in the game, such as walls and gates. These objects will inherit from the mirror object, which will stop the player characters from moving when travelling towards the mirror object. Sprites of the objects will be basic pixelated representations of walls and gates. No sounds are necessary for these objects.



There will be button objects, one of which will inherit the other. Additionally, keys to locks will inherit from the button object that does not require to be constantly pressed to keep previously locked gates unlocked Sprites of these objects will appear as different colored buttons and as black pixelated keys, respectively. Button press sounds and key collect sounds will be played when these items are interacted with.



Two separate turret objects will also be created that fire bullet objects. Based on the orientation of the turrets, bullet objects will move in the direction in which the turret is pointing. The first turret object will not follow the player while the second turret object will follow the player. Sprites of these objects will appear as pixelated guns resting on tripods. The bullets will be bright blue missiles. Sounds will be made for shooting turrets and bullet collisions.

A manager object will be created for the purpose of displaying UI. This UI will display the current level number and the number of lives in the game. The manager will also keep track of the lives in the game, which will be three. Additionally, the manager object will have a timer that, when depleted, causes the player to lose a life. This timer will also be displayed by the UI.

A second manager object will be created for the purpose of displaying UI for the tutorial as well as introducing the player to the various potential hazards by spawning and removing those objects.

A goal point object will also be created that advances the player to the next level and increases the level number for the object manager. The sprite will look like an exit sign. A sound of the player characters will play with them acknowledging that they beat the level.

**Gameplay**

The game requires use of the four direction keys to move your face and reflection. Vertical controls of the characters are the inverse of each other. As you move down, your reflection moves up towards you, and vice-versa. Both characters stop moving when either of them is blocked by a physical barrier such as a locked gate or a wall. Additionally, both characters die at once when either of them is hit by a harmful object, such as a turret’s bullet.

Because of the characters’ dependency on each other, the player must figure out how to successfully navigate both characters through each of the levels. When the player does reach the goal point of the level, the level ends.

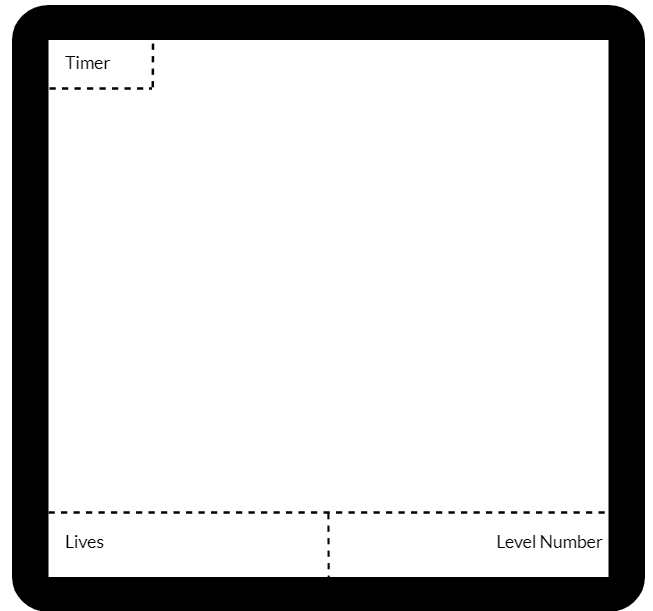
The appeal of the game comes from the puzzle elements and from the small story presented in the game.

**AI**

The only expected enemy AI that the player will have do deal with are turrets. These will either target the player or remain in a fixed direction. The turrets that target the player must first be activated when the character approaches the turret before it starts shooting bullets at the player.

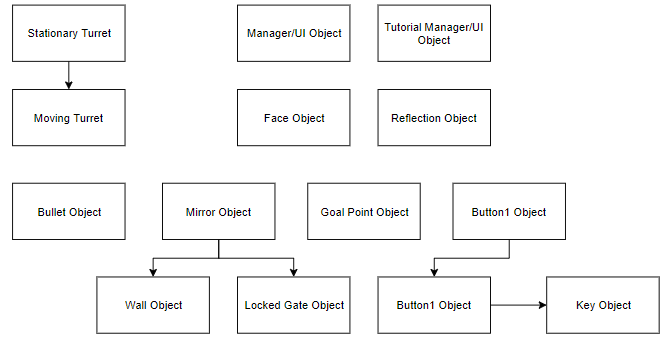
**UI**

Below is a chart for the layout of the UI.



**Hierarchy Chart**

Below is a chart depicting the hierarchy of the game objects. Arrows indicate what objects will pass on traits through inheritance. (For updated version, refer to Appendix A).



**Story**

The story of the game follows a face. Feeling discouraged in himself and feeling that he sets himself up to fail due to his recent loss of a job, he goes to sleep. He finds himself in a dream with a large mirror that seems to stretch out forever. He also sees a slightly distorted reflection of himself as well as other objects in the mirror that he does not see in the room he is standing in. The face feels compelled to leave this dream, but soon discovers that he is held back by his very own reflection, since it stops at obstacles in the mirror world. As a result, the face must escape the dream by overcoming his own imaginary obstacles, mere reflections in a dream. Once he escapes the dream, he overcomes his tendency to fall behind due to obstacles he sets up for himself and feels confident in his ability to accomplish what he desires.

**Difficulty Graph**

Below is a difficulty graph depicting the expected difficulty across the tutorial level and five main levels. Difficulty will be increased using more challenging puzzles and additional hazards.

**Timeline**

Below is a table depicting the expected timeline of the project and what accomplishments must be made by the deadlines.

|  |  |
| --- | --- |
| Create all necessary objects with simple sprites | April 6 |
| Design levels with the objects | April 13 |
| Refine and polish the project | April 20 |
| Complete and submit Final Project | April 26 |

**Concept Art/Pixel Art**





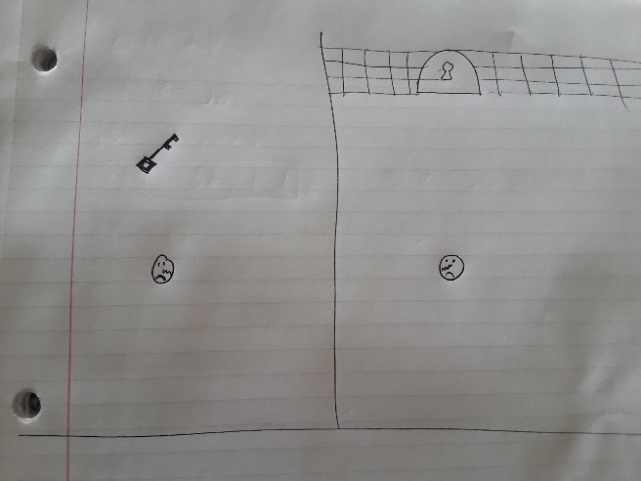
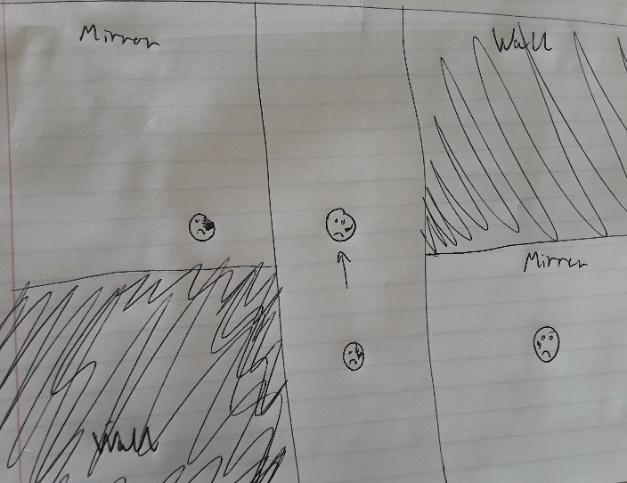
Expressions of the face and his reflection

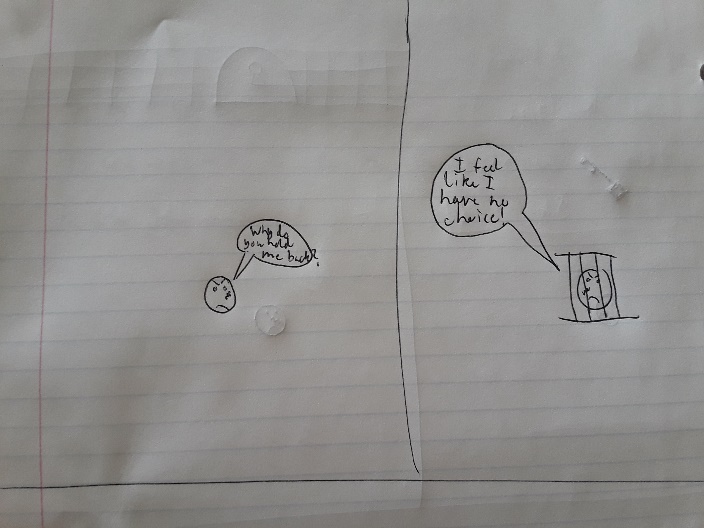


Wall block



Lock and key (key sprite is borrowed)





Early concept art

**Appendix A**

Below is the updated object hierarchy chart following the development of reflect. Please refer to this chart when comparing inheritance within Reflect to the inheritance in this design document.

