TABLE OF CONTENTS

**CHAPTER**

[1 GAME OVERVIEW 4](#_Toc17277386)

[1.1 Game Summary 4](#_Toc17277387)

[1.2 Movement 4](#_Toc17277388)

[1.3 Potion Control 5](#_Toc17277389)

[1.4 Game Flow 5](#_Toc17277390)

[1.5 Target Platform 6](#_Toc17277391)

[2 GAME MECHANICS 7](#_Toc17277392)

[2.1 Technical Specification 7](#_Toc17277393)

[2.1.1 Game Engine and Tools 7](#_Toc17277394)

[2.1.2 Technical Obstacle 7](#_Toc17277395)

[2.2 Architecture 7](#_Toc17277396)

[2.3 Software Pattern 8](#_Toc17277397)

[2.4 Features Implementation 8](#_Toc17277398)

[2.4.1 Potion Projectile Motion 8](#_Toc17277399)

[2.4.2 Fire Flame 10](#_Toc17277400)

[2.4.3 Ice Wall 10](#_Toc17277401)

[2.4.4 Space Teleportation 11](#_Toc17277402)

[2.5 Enemy AI Implementation 12](#_Toc17277403)

[2.5.1 Normal Enemy 12](#_Toc17277404)

[2.5.2 Boss 13](#_Toc17277405)

[2.5.3 Space Portal 14](#_Toc17277406)

[2.6 Statues and Hints Implementation 14](#_Toc17277407)

[2.6.1 Lava Statue Obstacle 15](#_Toc17277408)

[2.6.2 Heal Fountain Statue 16](#_Toc17277409)

[2.6.3 Fire Hint 17](#_Toc17277410)

[2.6.4 Ice Hint 18](#_Toc17277411)

[2.6.5 Space Hint 19](#_Toc17277412)

[2.6.6 Map Hint 19](#_Toc17277413)

[3 OTHERS IMPLEMENTATION 20](#_Toc17277414)

[3.1 Map Design 20](#_Toc17277415)

[3.1.1 Ground 20](#_Toc17277416)

[3.1.2 Wall 20](#_Toc17277417)

[3.1.3 Vertical Obstacle 20](#_Toc17277418)

[3.1.4 Horizontal Obstacle 20](#_Toc17277419)

[3.2 User Interface 21](#_Toc17277420)

[3.2.1 Start Scene 21](#_Toc17277421)

[3.2.2 Story Scene 21](#_Toc17277422)

[3.2.3 Main Scene 22](#_Toc17277423)

[3.3 Camera Setup 23](#_Toc17277424)

[3.4 Audio Setup 24](#_Toc17277425)

[4 PROJECT MANAGEMENT 25](#_Toc17277426)

[4.1 Coding Convention 25](#_Toc17277427)

[4.2 Assets Format 26](#_Toc17277428)

[4.3 Gantt Chart //left to finish 27](#_Toc17277429)

# 

## GAME OVERVIEW

### Game Summary

Alchemystic is an adventure game. The player is an alchemist and he have to fight against the enemy. Player has 3 types of potions to play with, which are fire potion, ice potion, and space potion. Each potion has its functionality and player has to defeat the enemy to fight the boss.

Fire potion is an attack potion. It will damage enemy whenever enemy is pass through. Besides, ice potion is a defend potion. It functions as blocking the enemy. On top of that, space potion is a teleportation potion. It will teleport the player to the ground where the potion land. The player has to both defeat the boss and seal the portal in order to win the game. There will be obstacle and hints that the player will be needed to explore the map themselves.

### Movement



In Alchemystic, the player movement is mainly control by these 3 keys:

* “A” (or Left) to move left
* “Space” (or Up) to jump
* “D” (or Right) to move right

### Potion Control







These 3 potions that player hold can be select by these 3 keys:

* “Z” (or 1) to select Fire Potion
* “X” (or 2) to select Fire Potion
* “C” (or 3) to select Fire Potion

### Game Flow

The game will begin with a loading screen. After that, a prologue will show to indicate the background of the game. After the prologue, the game will start. In the main game scene, there will be short tutorial that teaching the players the basic movement of the character. As there will be enemy scattering around the maps, character may have to eliminate enemy in order to move forwards. Enemy will move towards to the character when there are nearby.

The player can either to defeat the first enemy encountered or just pass through the enemy. After that, the player can go under the map and explore. As the exploring continues, the player will find a scroll that containing the hints for defeating boss. After that, player has to effectively use the space potion in order to go back to the top of the map. Besides, as the adventure progress, the player had defeated several enemy. The player will face an obstacle, called Breath of Lava. Player has to throw ice potion in order to stop the flowing lava in order to jump over the obstacle.

Before the bossing stage, player will pass by a healing fountain. The fountain will heal the player to maximum health for the preparation to fight the boss. At the boss stage, player has to seal the portal and also defeat the boss in order to win the game. Besides, the portal and boss will randomly spawn an enemy. Thus, the player has to overcome the problem at the same time achieve both objectives to win the game.

### Target Platform

Windows and Mac are the target platform of this game developed. As the game involved a lot of movement, clicking and projectile shooting, PC will be the suitable devices to play the game. Windows and Mac will be the best platform for Alchemystic as mouse can be used to shoot the potions.

## GAME MECHANICS

### Technical Specification

This session will discuss the game engine used, development tools involved in the development process and technical obstacle.

#### Game Engine and Tools

The game is developed using Unity, a game engine that provide the ability to create game by simplifying the process of making a game with the features provided. Besides, the scripts are mainly edited through Visual Studio 2017. For the animation sprites editing, Photoshop CC 2018 is used. For the hardware requirement, Alchemystic is a low-end game, which any PC will be able to play without any underperformance of computer.

#### Technical Obstacle

The main technical obstacle in this game development is the algorithm to calculate the projectile motion and trajectory path of the potion throwed.

### Architecture

Alchemystic consists of three scenes, which is “Start Scene”, “Story Scene” and “Main Scene”. The sequence of scene is:

1. Start Scene
2. Story Scene
3. Main Scene

The details implementation and explanation of each scene will be discussed in the user interface session of chapter 3.

### Software Pattern

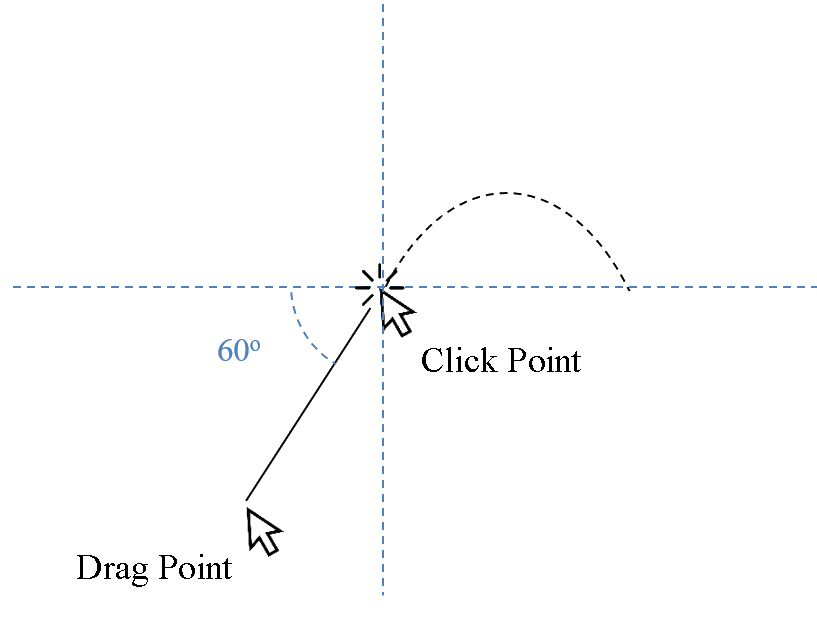
One of the software patterns used in this game is “Singleton pattern”. Singleton pattern is used for the purpose that restricts the instantiation of a class to one single instance. It applied in the “PlayerController” and “AudioManagerController” classes as these two scripts will always have exactly one object only throughout the game.

### Features Implementation

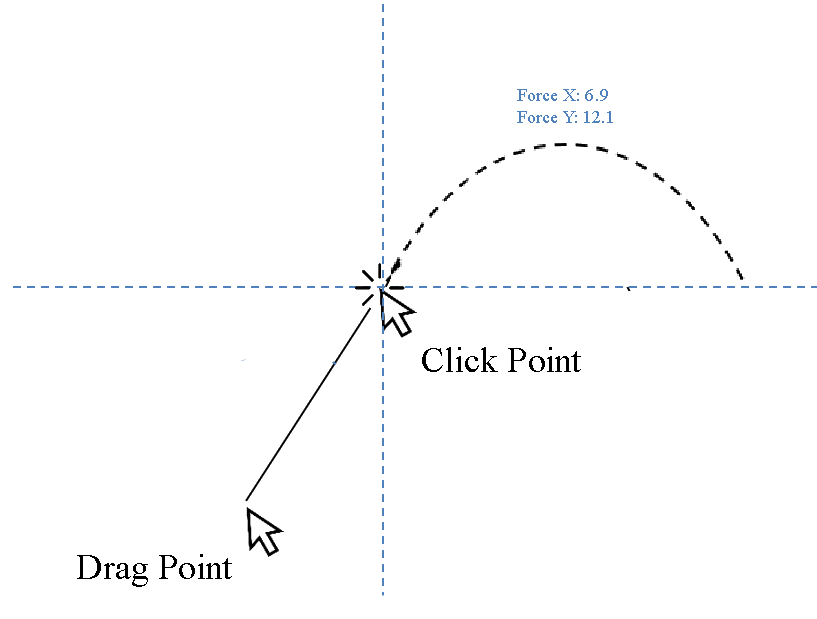
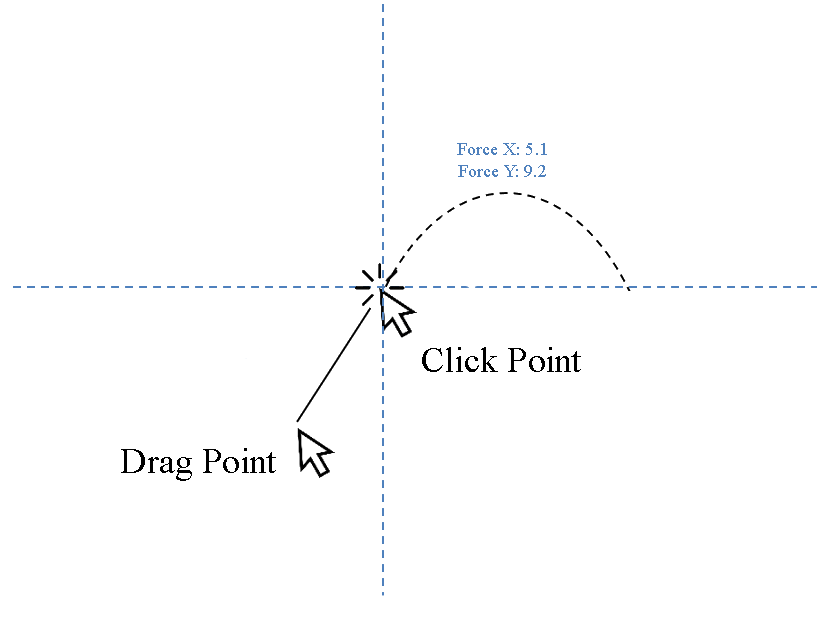
This session will explain the implementation approach of the main feature of the game developed.

#### Potion Projectile Motion

The trajectory path is calculated based on input angle, input force and velocity of it was set as constant. The angle is the angle between the point of the first mouse’s click and the drag point which the mouse’s click hold and move around. For example:



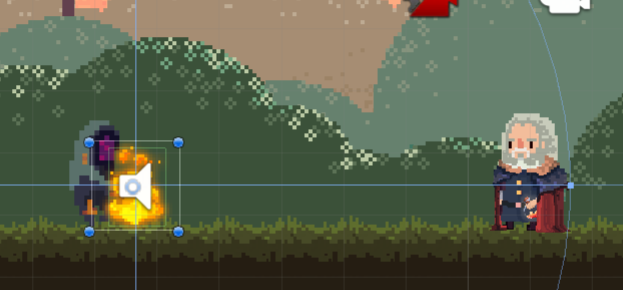
The input force will be determined based on the distance between the point of first mouse click and the drag point which the mouse click hold and move around. The longer the distance between the two-specific point, the higher the input force to the projectile motion for the potion. For example, the figure below shown that the left one is shorter distance and smaller force, while the right one is longer distance and larger force.



The trajectory path will be show when the player’s mouse clicked, and the potion will be thrown by adding horizontal force and vertical to the potion when the player released the mouse click. The potion’s collider will then collide with the ground’s collider and instantiate a new game object based on the potion type that player throwed.

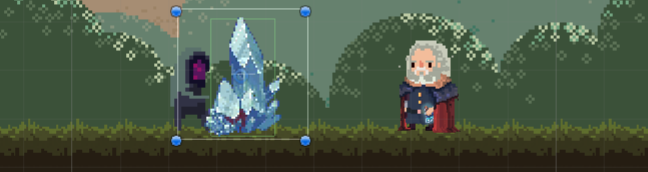
#### Fire Flame

The fire flame is a prefab object that will be instantiated when the fire potion’s collider collides with the ground’s collider and the fire potion will then be destroyed. The fire flame will then apply certain amount of damage to the enemy when the enemy’s collider enters or stays inside the triggered collider of fire flame. The fire flame is AOE attack type, which mean that it can damage multiple enemy at the same time as long as the enemy inside its triggered collider. However, fire flame was set that will not damage the Alchemist, which is the player. The figure below shown that the fire flame is damaging the normal enemy.



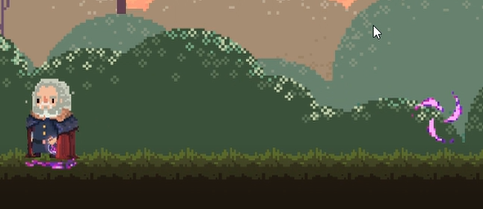
#### Ice Wall

The ice wall is a prefab object that will be instantiated when the ice potion’s collider collides with the ground’s collider and the ice potion will then be destroyed. The ice wall rigidbody2D is set to be static as it will not be push by other game object’s rigidbody2D. As it’s static characteristic, it acts as the Alchemist’s defensive technique as it can block the enemy way toward the player. It also can act as bait to enemy as the enemy will chase and attack the nearest target they detected. However, ice wall can be destroyed by the enemy attack but fire flame has no effect to the ice wall. In basic word, the ice wall can be destroyed by the enemy but will not destroy by fire flame. On the other hands, the ice wall has a maximum limit of instance, whenever the number of instances of ice wall exceeds the limit set, the earliest ice wall will be destroyed. So, the number of ice wall’s instances will be between 0 to 5 only. The figure below show the normal enemy is blocked by the ice wall.



#### Space Teleportation

The space teleportation is more complex compare to previous two potion’s effect. The space teleportation effect is achieved by two prefab objects, which is the starting effect and ending effect. After the space potion was throwed out from the Alchemist, the *“current transform position”* and *“previous transform position”* of the space potion at one time will be stored for every frame update. When the space potion’s collider collides with the ground’s collider or wall’s collider, the player, which is the Alchemist’s position will be translate to the *“previous transform position”* of space potion that stated above. The reason of why doing this is to reduce the chances of the player teleport into the collider of the ground or wall game object as the *“current transform position”* of space potion may be inside the collider of ground or wall. As the same time, the starting effect of the space teleportation will be instantiated at the player original position while the ending effect will be instantiated at the player new translated position. For example, that stated below:



Original Position

New Position

### Enemy AI Implementation

There are two type of living enemy and non-living enemy in this game, which called as “Normal Enemy”, “Boss” and “Space Portal”.

#### Normal Enemy

The normal enemy had the highest population in this game. It has two mode which is patrol mode and chasing mode. When a normal enemy is in patrol mode, a random x-coordinate will be generated in range set and the normal enemy will patrol to that point of coordinate. Then, an idle time will be randomly generated within a range that set. After the idle time, a new random x-coordinate will be generated again, the patrolling loop will continuously ongoing until enemy detect its target.

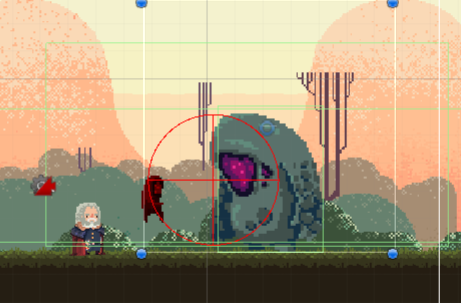
The detection of target of normal enemy is achieved by a triggered box collider that attached on the normal enemy game object. The box collider defined the normal enemy’s detection area. Whenever the Alchemist or ice wall entered the enemy’s triggered box collider, the Alchemist or ice wall will be added into enemy target list. Enemy will chase toward the nearest target in its target list. For examples, the figure below shows the enemy’s triggered box collider by yellow rectangle.



The attack type of normal enemy is only melee attack. The red circle above shows the enemy’s attack area. While the normal enemy is in chasing mode, it will continuously be moving toward the target when the target still inside the enemy’s detection area. When the distance between target and normal enemy is within the normal enemy’s attack range, enemy will attack the target. The red circle will capture all the colliders inside the red circle area and a list of colliders will be produced. If player or ice wall is inside the list of colliders, damage will be applied to the player or ice wall.

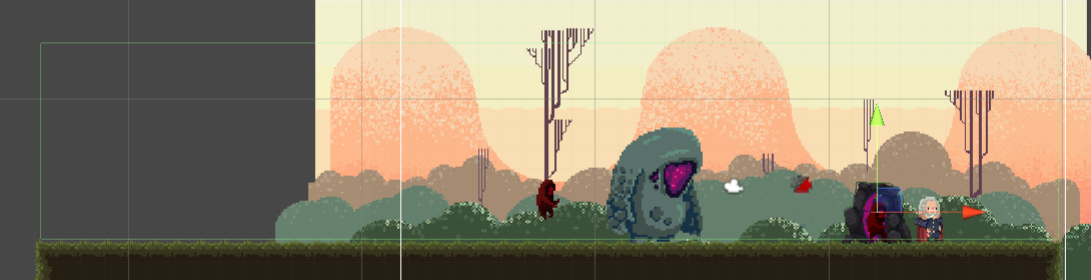
#### Boss

The boss had the similar component and mechanics as like normal enemy but one additional ability, which is summon normal enemy same like space portal. The boss had attached with a triggered collider that similar to the detection area’s triggered collider. When the Alchemist entered or staying inside the summon area of boss which defined by a triggered collider, the boss will summons one normal enemy and then this ability will go into cool down. The boss has a summoners amount limit such as five, if the summoned normal enemy amount had reached five unit, then the boss was unable to summon anymore. The summoned normal enemy’s game object will add into an ArrayList of game object, this specific ArrayList used to track the number of normal enemy summoned. When a normal enemy die, it will be removed from the ArrayList and the boss can now ready to summon a new normal enemy as the length of the ArrayList had been reduced. As the figure shown below, the blue rectangle labels the boss triggered collider that indicate the boss’s summon area.



#### Space Portal

The space portal is a non-living enemy that have a static rigidbody2D. It’s summons ability had the same mechanics just like the boss’s summons ability (same number of maximum summoners) but have larger cover area, which cover the whole boss platform. For example, the figure below labels out the summon triggered area of space portal:



### Statues and Hints Implementation

There are two type of statues and four type of hints constructed in this game, which is:

* Lava Statue Obstacle
* Heal Fountain Statue
* Fire Hint
* Ice Hint
* Space Hint
* Map Hint.

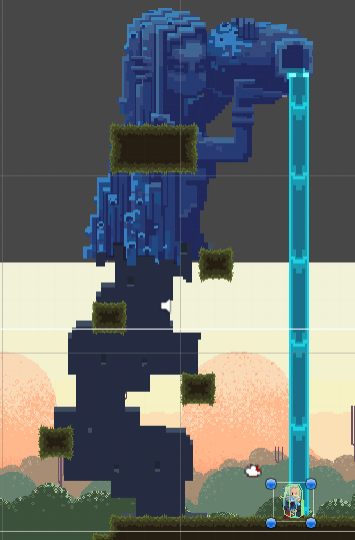
#### Lava Statue Obstacle

Lava status is an obstacle that block the journey of the Alchemystic and the player need to overcome it in order to pass through it. It mainly consists of two part, which is the lava outlet and flowing lava. The flowing lava consist of the box collider that block the way of the player by instantly kill the Alchemist when Alchemist’s collider collides with it. In addition, potion like ice potion and space potion will also be destroy by the flowing lava. However, only the fire potion can pass through the flowing lava as they are same element. The solution to the obstacle is throw an ice potion to the lava outlet, then the ice potion’s collider will collide with the lava outlet’s collider, the flowing lava then can be turned off. After turned off, the flowing lava can also be turned on again by throwing a fire potion to the lava outlet. There is another way to pass through this obstacle, which is throw a space with maximum force and at specific angle, there will be slightly chance to pass through it. The figure below labeled the lava outlet with red rectangle, and flowing lava with the blue rectangle.



#### Heal Fountain Statue

Heal fountain is a statue that will heal the Alchemist when his health is not full. The heal fountain has one triggered box collider that defined the healing area. When the Alchemist’s collider enters or stays inside the heal fountain triggered box collider, if the player health is not full, the healing fountain will heal the player with a certain amount of health with a certain amount of cool down defined, for example, 20 amounts of health with 2 second cool down. The figure below labeled the healing area by the red rectangle.



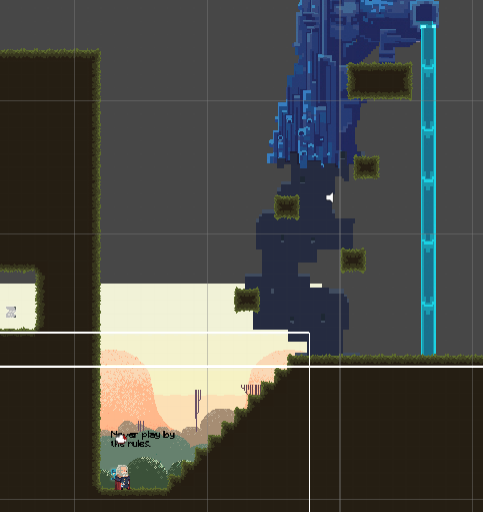
#### Fire Hint

Fire hint is a floating scroll has a collider to detect if the player enters or staying in it. Only the Alchemist that holding fire potion can show up the text hidden. The text, “It’s better for the doer to undo what he has done.” implied a hint that the space portal can be sealed by the space potion that are same elemental characteristic. The figure below specified the hidden location of the fire hint where the player may hard to reached by a red circle.



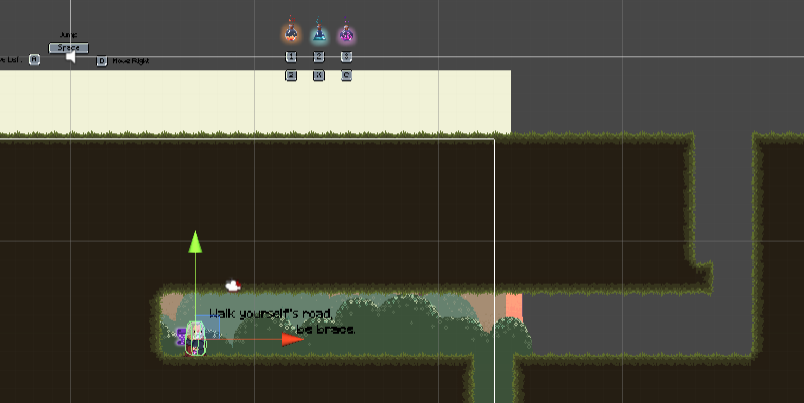
#### Ice Hint

Ice hint had the same detection mechanics and text show up mechanic same like fire hint. The only difference is the text and location of the ice hint scroll. The text, “Never play by the rules.” implied a meaning that the player should not restrict by the intuitive way and should use creativity to find solution. The figure below specified the hidden location of the ice potion where is near to the healing fountain.



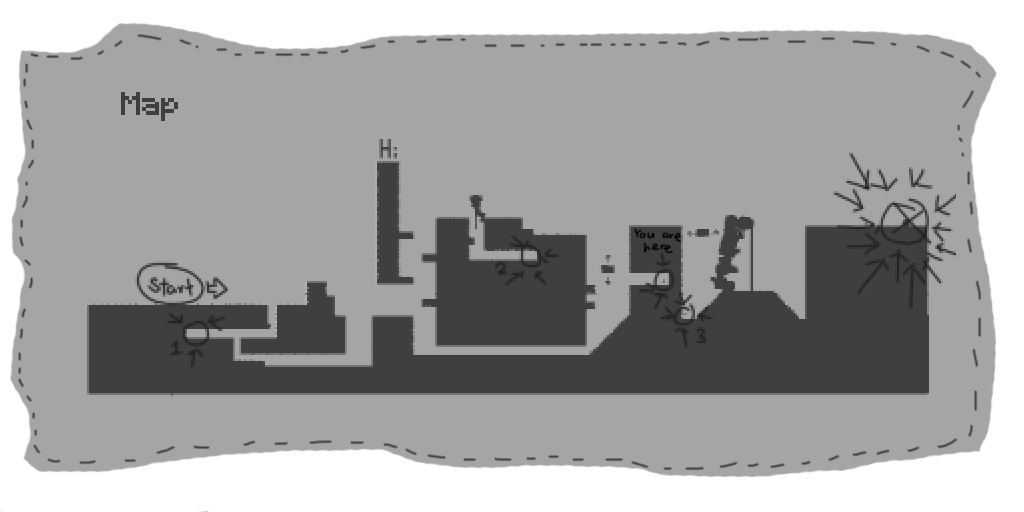
#### Space Hint

As like fire hint and ice hint, space hint had the same detection mechanics and text show up mechanic same like them. The only difference is the text and location of the space hint scroll. The text, “Walk yourself road, be brave.” implied a hint that player should not always follow the obvious way and should brave to explore the map. The figure below specified the hidden location of the space hint where under the tutorial platform by a red circle.



#### Map Hint

As like all previous hint, map hint had the same game mechanics with them. But instead of showing up text, an overview map image will be shown.



## OTHERS IMPLEMENTATION

### Map Design

The map implementation of this game is done by tile map. The tile map sprite sheet is manually draw and inspired by the ground of the background image. There are four tile maps inside the game which is “Ground”, “Wall”, “Vertical Obstacle” and “Horizontal Obstacle”. All the tile map collider’s edge radius had set into 0.1 to reduce the chances of player use the space potion and teleport into the tile map collider.

#### Ground

This tile map serves as the mainly platform that player will stand on, and to collide with the three potions’ collider to produced potion effect.

#### Wall

This tile map exists because the ice potion and fire potion will not instantiate their respective effect when the potions’ colliders collide with wall. But space potion will still have its teleportation effect when space potion’s collider collides with wall.

#### Vertical Obstacle

This tile map is a vertical moving platform that will moving within a range of y-coordinate while its x-coordinate is constant. Whenever the player jumps onto this tile map, the player’s rigidbody2D parent will be set to this game object to prevent the player is not moving with the platform.

#### Horizontal Obstacle

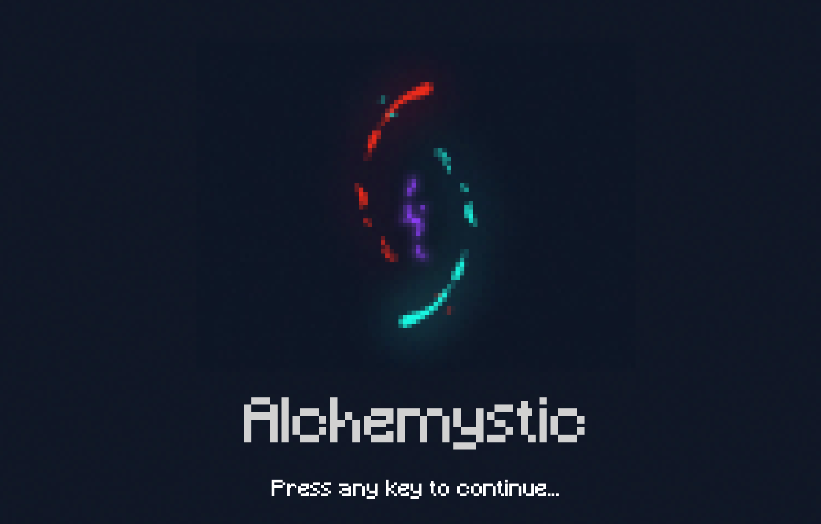
This tile map is a horizontal moving platform that will moving within a range of x-coordinate while its y-coordinate is constant. Whenever the player jumps onto this tile map, the player’s rigidbody2D parent will be set to this game object to prevent the player is not moving with the platform.

### User Interface

There are three scenes had implemented user interface.

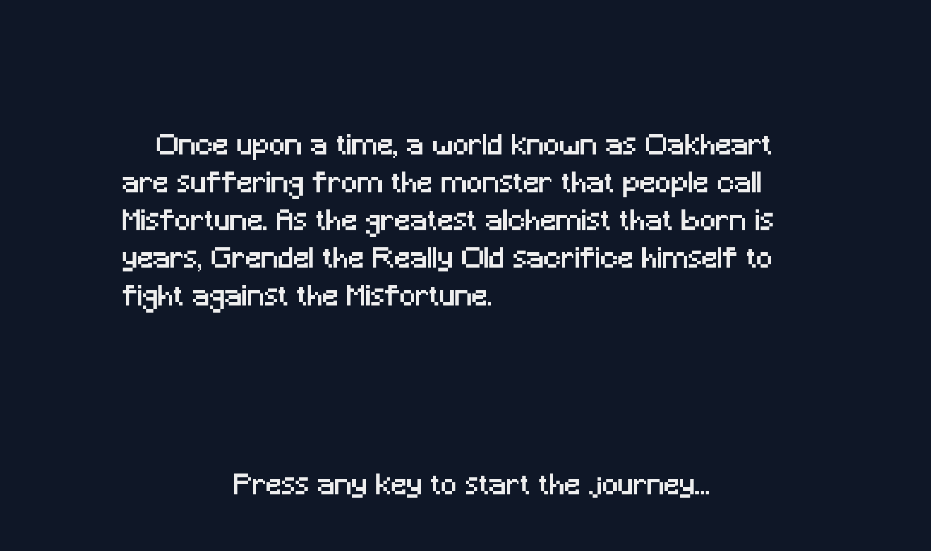
#### Start Scene

This scene consists of a video that looping an animation of 3 elements and a script object to detect any input that will load the next scene which is “Story Scene”. The figure below shows the start scene.



#### Story Scene

This scene consists of a text object that display the Alchemystic story to the player. The text object will have a type writing effect which the word will be seem like type out one by one. Same as previous scene, this scene also has a script object to detect any input, but the first input will skip the type writing effect and the second input will load the next scene which is “Main Scene”. The figure below shows the introduction story of the game in the story scene.

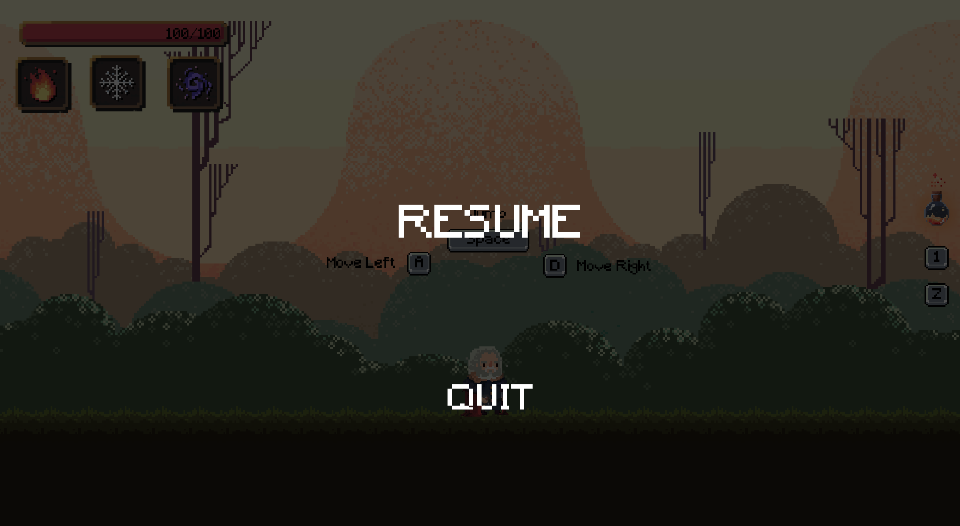


#### Main Scene

This scene is the game scene that the player will spend they time mostly. Player will need to explore the map, defeat enemies, and sealed portal in order to win the game.

##### Menu

Pause menu and restart menu were implemented in this scene. The pause menu will appear when the user pressed the “Escape” key. In the pause menu, there are two button that allow player to click, which is the resume button and quit button. When player pause the game, the time scale had set to zero and some input for the game had been disable to stop the game. The resume button will restore the time scale back to value of one and continue the game. The quit button is basically to exit and close the game. The figure below shows the pause menu.



For the restart menu, it only appears when the player died. When the player died, the time scale will be set to 0.2 and this will slow down the game motion. Restart button is just loaded again the “Main Scene”, so the player will be respawned and back to starting point. The figure below shows the restart menu.



##### Health Bar and Cool Down Button

The top left panel of the main scene will display the basic information about the Alchemist which include the health and each potion’s cool down. The health bar will indicate the current health amount of the Alchemist, it consists of three part, the border frame, red filled bar and a text object. The changing of length of red filled bar is achieved by the editing the image filled amount of the red horizontal filled bar. The text object will also keep updating the health amount. Besides, a potion’s cool down icon is made from two part, which is the background icon and grey cover in front of it. The cool down effect is done by editing the image filled amount of the grey radial cover image. When the grey cover disappears, it means that the potion is ready to use again. The figure below is the example of a top left panel of Main Scene.



### Camera Setup

The camera setup is done by the Cinemachine package. Cinemachine is a smart camera tools for Unity game development and it delegate the camera manual setup for developer. By utilizing Virtual Camera of Cinemachine, the camera will follow the target set which in this case is the Alchemist. It provided a smooth camera movement and does not jitters for the graphical view. Besides, a component called Cinemachine Confiner also be used to restrict the camera frame stay inside the boundary defined by the colliders. This prevent the camera capture any unwanted objects or graphics into it.

### Audio Setup

There are two type of audio setup used in this game, which is component type and script type. The component type is simply attached the audio source to the specific game object. This kind of audio setup is used on the static game object such as heal fountain, lava obstacle etc. For the script type, two script are used to control the audio. Firstly, a class script to store the audio information is created and has a name called “Sound”. Secondly, a script called “AudioManagerController” is created to manage the instance of the “Sound” class such as play or stop the audio sound. The script type audio setup is convenience as some sound effect is required to play in the run time. A centralized audio manager is easier to control and manage. The script type is used to achieve the background music transition between scene, control the sound volume etc.

## PROJECT MANAGEMENT

### Coding Convention

The goal of the coding convention of this game development are readability and simplicity.

**Class Naming Standard**

PascalCase is used.

For example: AudioManagerController

**Variable Naming Standard**

camelCase is used.

For example: spaceStartEffectPrefab

**Method Naming Standard**

PascalCase is used.

For example: StartMainSceneBackgroundMusic

**Parameter Naming Standard**

camelCase is used.

For example: targetList

### Assets Format

Assets is a representation of any items that are used for building the game. Assets may come from a file created outside of Unity, including audio file, images and any type of the file that supported by Unity. For example, most of the image format that used in this game is .png, and audio format is .mp3.

Majority of the sprite source of Alchemystic are come from Pinterest and then edited to tailor into the game, including the character, maps and the obstacle. Besides, a part of the game audio file is come from SoundSnap. On the other hands, some sprite sheet and audio file such as normal enemy, boss, portal and background music are extracted from a game call Kingdom: New Lands. All the related file will be stored in respectively folder, such as boss animation will be stored in Assets>Animations>Boss. Below is the directory of the assets file:

Assets

|-- Animations

| |-- AfterSealed

| |-- Boss

| |-- Enemy

| |-- Hints

| |-- Obstacles

| |-- Player

| |-- Potion Effect

| `-- UI

|-- Audio

| |-- Background Music

| `-- Other Sound

|-- Fonts

|-- Gizmos

|-- Scenes

|-- Scripts

|-- Sprites

| |-- AfterSealed

| |-- Boss

| |-- Enemy

| |-- Environments

| |-- Hints

| |-- Hurting Effect

| |-- Player

| | |-- Death

| | |-- Fire

| | |-- Heal

| | |-- Hurt

| | |-- Ice

| | `-- Space

| |-- Potions & Effect

| | |-- Fire

| | |-- Ice

| | `-- Space

| `-- UI

| |-- Button

| |-- CD

| |-- Click and Drag

| |-- Health Bar

| `-- Potion

`-- Video

### Gantt Chart

