TABLE OF CONTENTS

**CHAPTER**

[1 GAME OVERVIEW 4](#_Toc17157149)

[1.1 Game Summary 4](#_Toc17157150)

[1.2 Movement 4](#_Toc17157151)

[1.3 Potion Control 5](#_Toc17157152)

[1.4 Game Flow 5](#_Toc17157153)

[1.5 Target Platform 5](#_Toc17157154)

[2 GAME MECHANICS 6](#_Toc17157155)

[2.1 Technical Specification 6](#_Toc17157156)

[2.1.1 Game Engine and Tools 6](#_Toc17157157)

[2.1.2 Technical Obstacle 6](#_Toc17157158)

[2.2 Architecture 6](#_Toc17157159)

[2.2.1 Start Scene 6](#_Toc17157160)

[2.2.2 Story Scene 7](#_Toc17157161)

[2.2.3 Main Scene 7](#_Toc17157162)

[2.3 Software Pattern 7](#_Toc17157163)

[2.4 Features Implementation 7](#_Toc17157164)

[2.4.1 Potion Projectile Motion 7](#_Toc17157165)

[2.4.2 Fire Flame 8](#_Toc17157166)

[2.4.3 Ice Wall 8](#_Toc17157167)

[2.4.4 Space Teleportation 9](#_Toc17157168)

[2.5 Enemy AI Implementation 10](#_Toc17157169)

[2.5.1 Normal Enemy 10](#_Toc17157170)

[2.5.2 Boss 11](#_Toc17157171)

[The boss had the similar component and mechanics as like normal enemy but one additional skill, which is summon normal enemy. 11](#_Toc17157172)

[2.5.3 Space Portal 11](#_Toc17157173)

[2.6 Obstacle 11](#_Toc17157174)

[2.7 Camera Setup 11](#_Toc17157175)

[2.8 Audio Setup 11](#_Toc17157176)

[3 PROJECT MANAGEMENT 11](#_Toc17157177)

[3.1 Coding Convention 11](#_Toc17157178)

[3.2 Assets Format 11](#_Toc17157179)

[3.3 Gantt Chart 11](#_Toc17157180)

# 

## 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

### 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 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

#### 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”.

#### Story Scene

This scene consists of a text object that display the Alchemystic story to the player. Same as previous scene, this scene also has a script object to detect any input that will load the next scene which is “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.

### Software Pattern

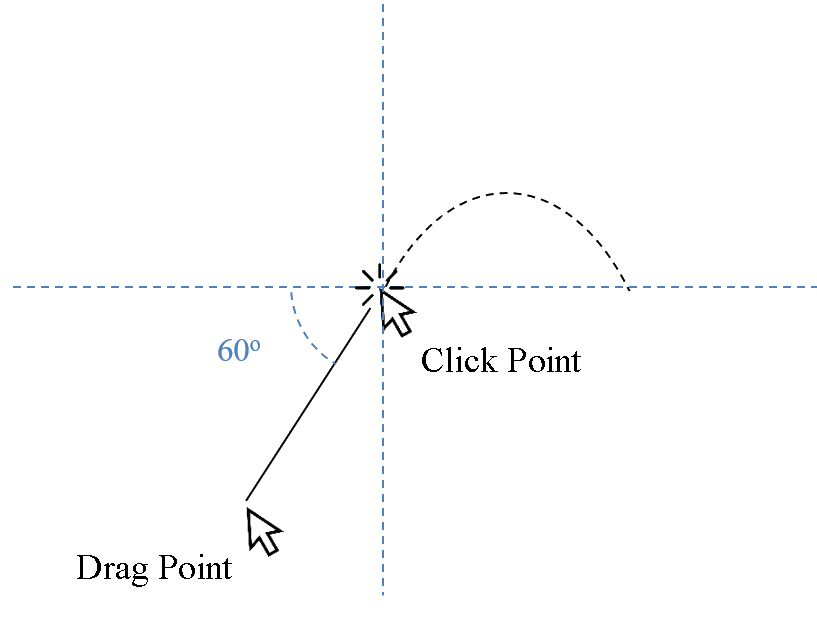
One of the software patterns used in this game is “Singleton pattern”. Singleton pattern is used for the purpose that restricts the instantiation of a class to one single instance. It applied in the “PlayerController” and “AudioManagerController” classes as these two scripts will always have exactly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

### Features Implementation

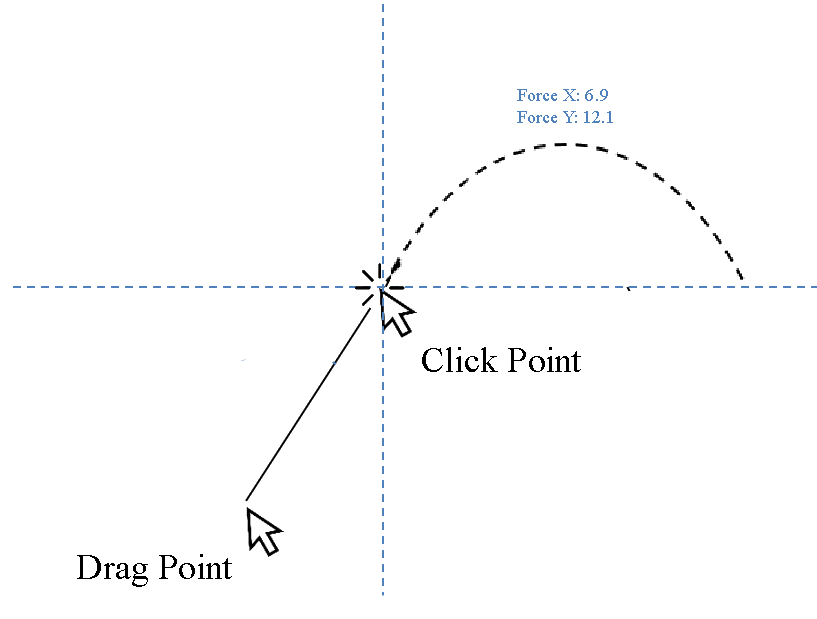
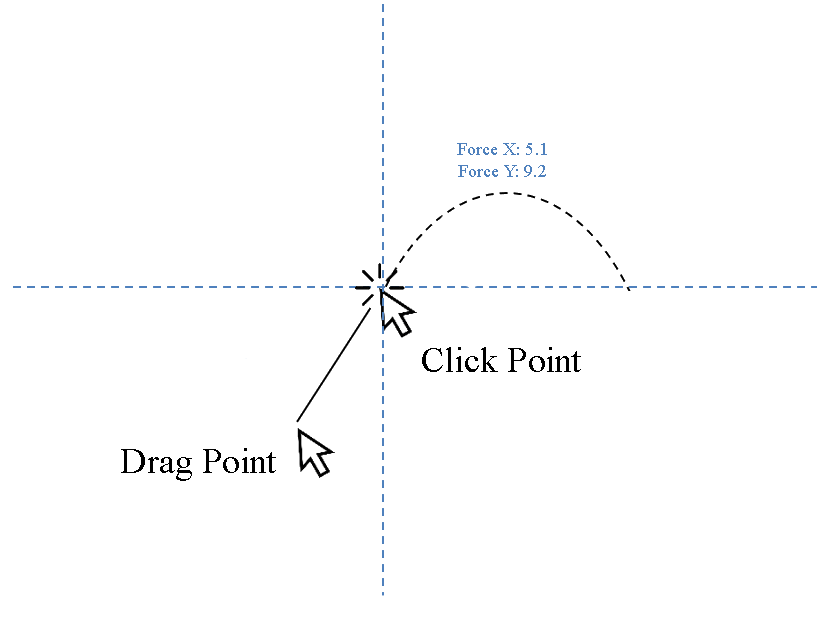
This session will explain the implementation approach of the main feature of the game developed.

#### 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

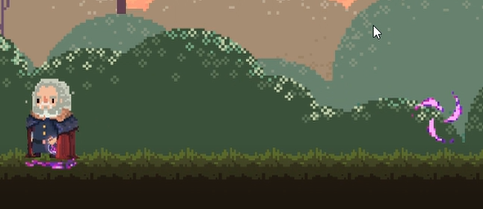
The fire flame is a prefab object that will be instantiated when the fire potion’s collider collides with the ground’s collider and the fire potion will then be destroyed. The fire flame will then apply certain amount of damage to the enemy when the enemy’s collider enters or stays inside the triggered 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.

#### 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.

#### 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 prevent 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 skill, which is summon normal enemy.

#### Space Portal

### Obstacle

### Camera Setup

### Audio Setup

## PROJECT MANAGEMENT

### Coding Convention

### Assets Format

### Gantt Chart