# Milestone 3 submission

**Team name**

FlatDyn Games

**Proposed Level of Achievement:**

Apollo 11

**Motivation**

Nowadays, the game industry is developing rapidly. Its influence cannot be ignored. Video games are no longer just for entertainment, and the technology in this field can even be applied to other fields such as movies, medical care, driverless driving, and urban planning.

As one of the traditional game concepts, 2D platform jumping game has countless masterpieces, such as "Ori and the will of the wisps" and "Dead Cells". We have decided to develop a game of this type as the first step in exploring the game development techniques.

**Scope**

Edge of Adventure is a single-player 2D platform jumping game set in a fictional world.

Players will experience interesting combat systems and diverse enemy modes of action.

Players can gain experience points and collect various props during the adventure.

The complete storyline and beginner guide will navigate players to enjoy the whole world outlook and learn how to control the character.

**User stories**

1. As a player, I want a game with smooth program operation.

2. As a player, I want to be able to control the character to complete movements, attacks, and other commands.

3. As a player, I want a game that can switch between different screens such as the main menu and settings, and freely select levels.

4. As a player, I want to be able to experience the full game visuals and sound effects.

5. As a player, I want to have fun and enjoy the game.

**Storyline**

You are a samurai that is transported to a Wasteland world where you may face dangerous enemy that are aggressive and will try to attack anyone that is not the same as them.

In order to survive, you can either escape or fight your way out and search for the gate that can bring you home.

**Core features(proposed)**

Edge of Adventure will be a 2D platform jumping game which focus on the combat system. We will make the animations look smooth, with natural transitions between each action. The character controlled by players will have various of skills. Players can combine different skills to form combos. Additionally, the character will have some defense measures to deal with the enemy, such as block or evasion. It depends on players to destroy the enemy with stormy attacks or expose their weaknesses with targeted defense. A new map is undergoing creation.

**Features Developed**

By the end of milestone2, the main character in edge of adventure can throw attacks, perform hurt or dead animation when receiving damages. On top of that, our main character can dash which avoids all the attacks and can crouch to jump higher. In the perspective of quality gaming, a camera is designed to follow the player and shake when player gets hurt giving more interactive feedback to the game. Also, a visible health bar to remind player’s health to enhance the experience.

Player will encounter challenges as there are now two new enemies – spear and axe. The Spear will start running around with its weapon pointing forward and the Axe will hunt down player and swing its axe when player is within its attack range.

**Tech stacks(current)**

Unity

- The game engine that our game runs on

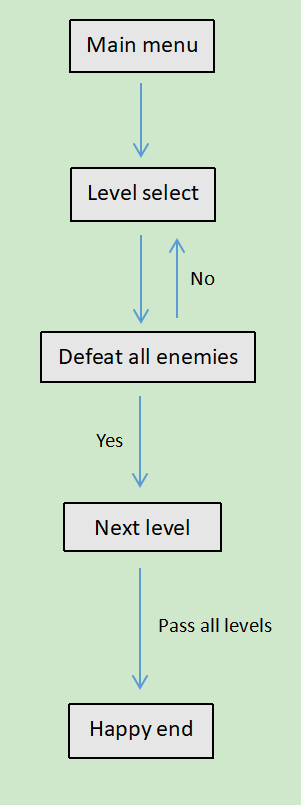
C#

- The language used to code in Unity

Adobe Animate

- The sprite maker

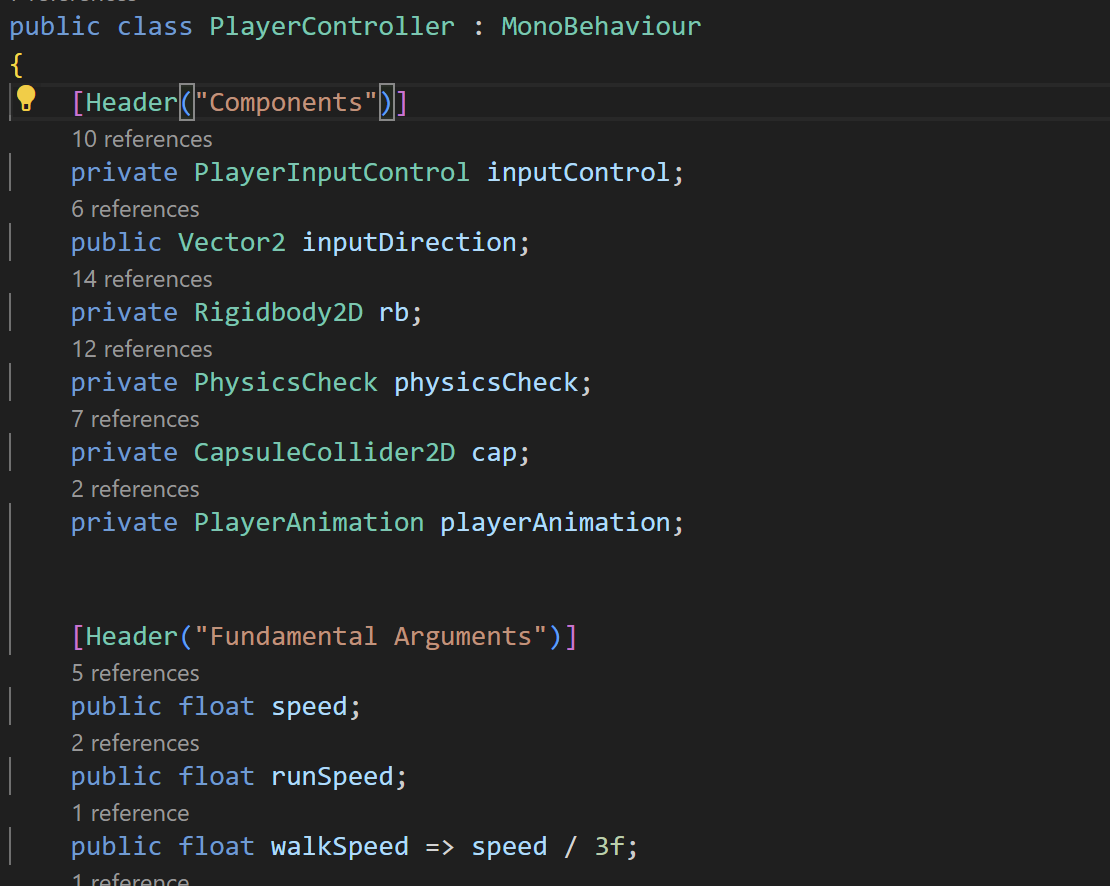
**Design and plan**

Game flow diagram

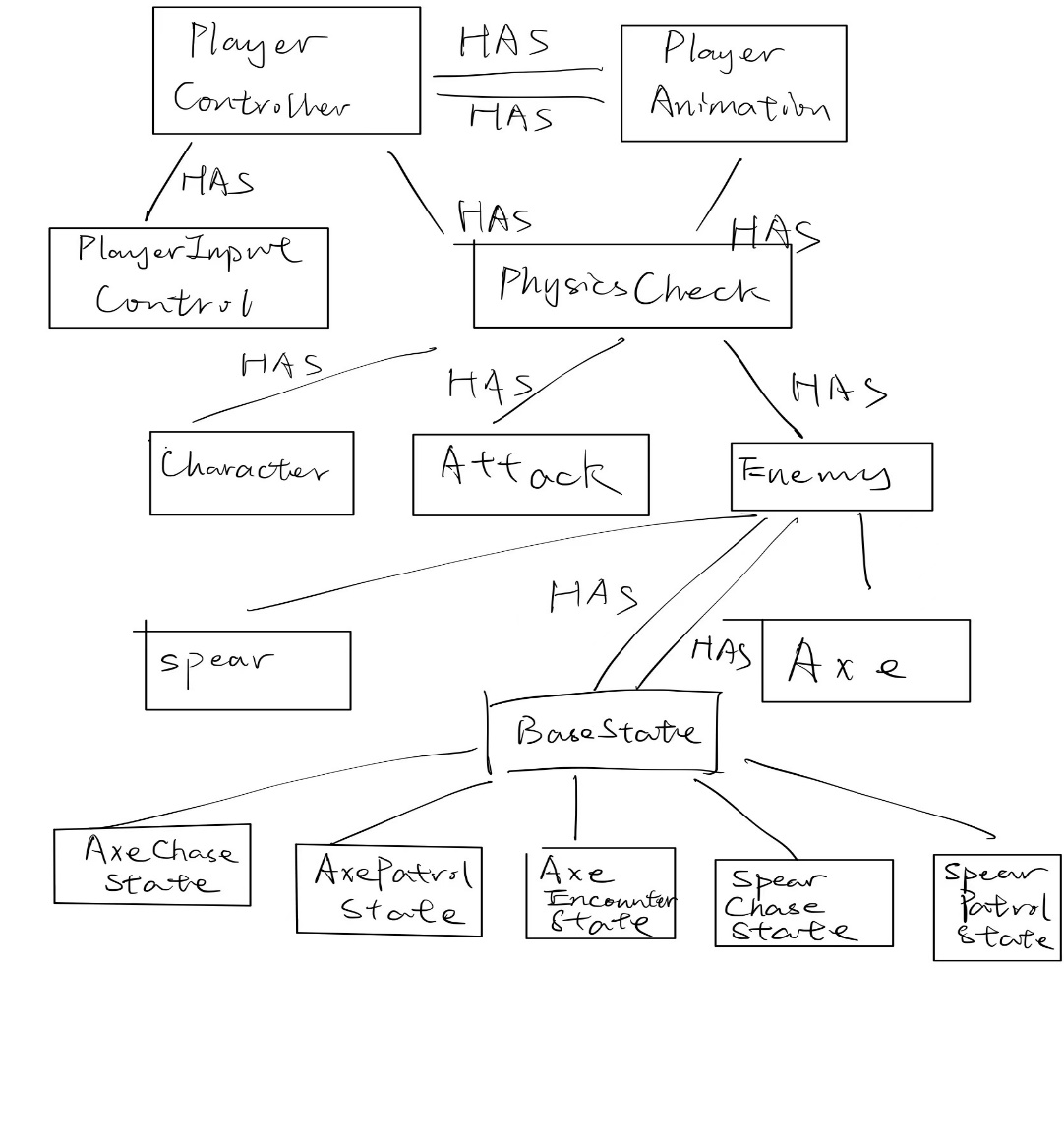
**Software Engineering Practices**

Composition and Inheritance

We have mostly done our coding through composition and inheritance. However, we fail to implement proper tell don’t ask principle, for the reason of finding proper arguments for player and enemies ,which will be fixed in the next milestone once settled down the final arguments.



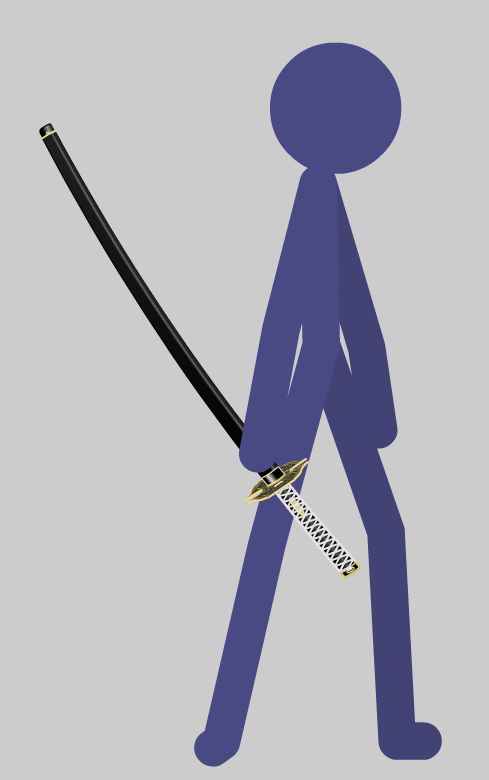
Class Diagram



**Github Documentation**

Because there is only two of us doing the project. Our work is all done in the main branch. The test branch is created because there was a warning due to changing directories of a file but turns out harmless to the whole project.

**Feature implemented**

Hero

HP: 40

Avg Damage: 7

Enemy

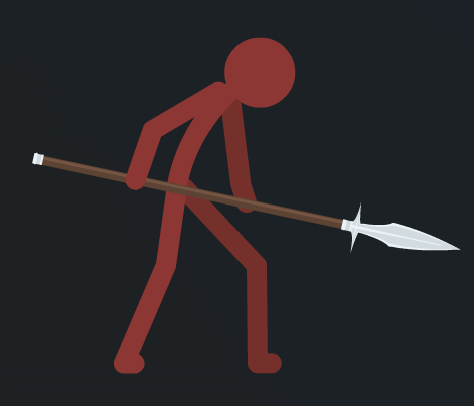
 Axe

HP:60

Damage: 5

Attack method: Chase down player and performs attack

Enemy Spear

HP: 60

Damage: 5

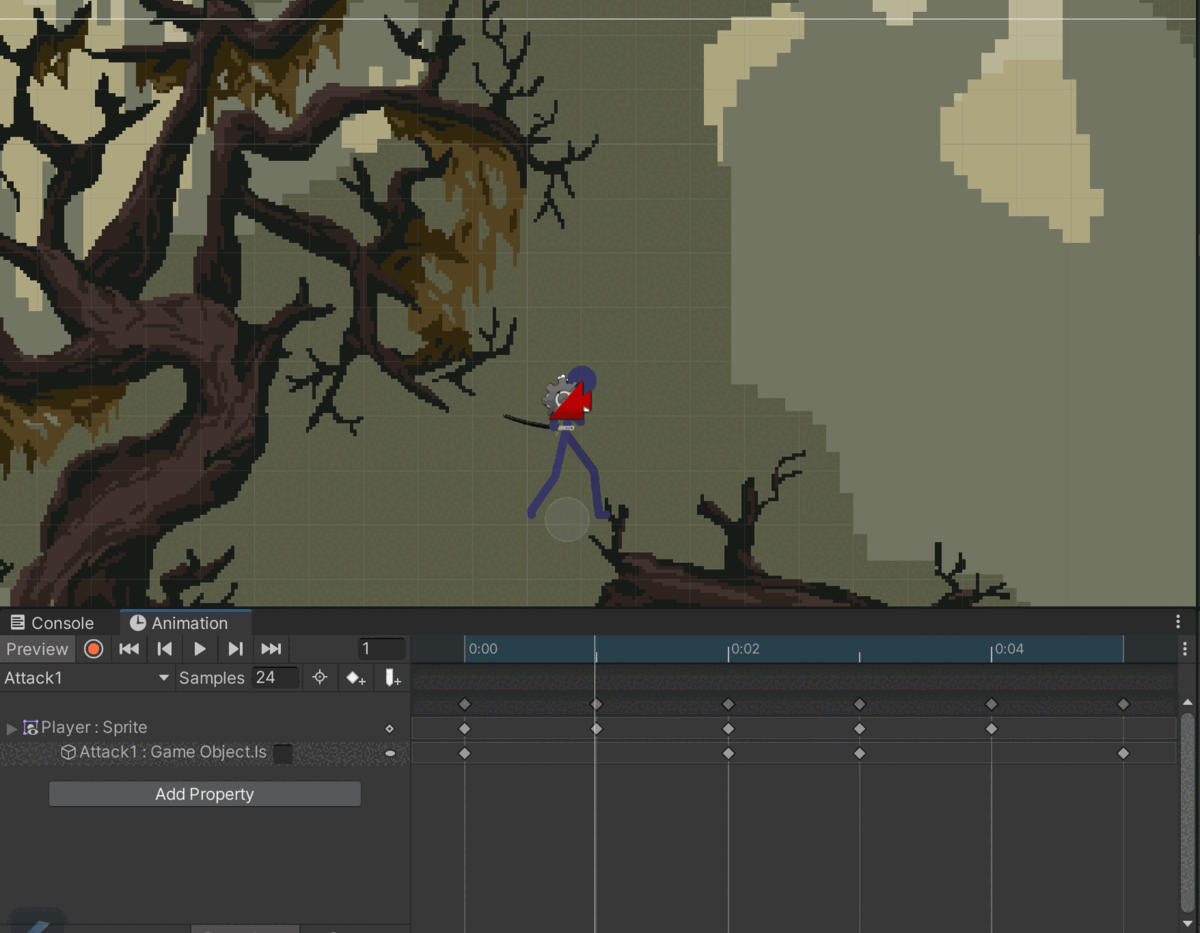
Attack method: running around with spear pointed forward

**State Switching system**

The enemies’ logic is controlled by state machine. We define a base state as an ancestor and all states from the enemies inherits from the base state. The states switch under certain conditions.

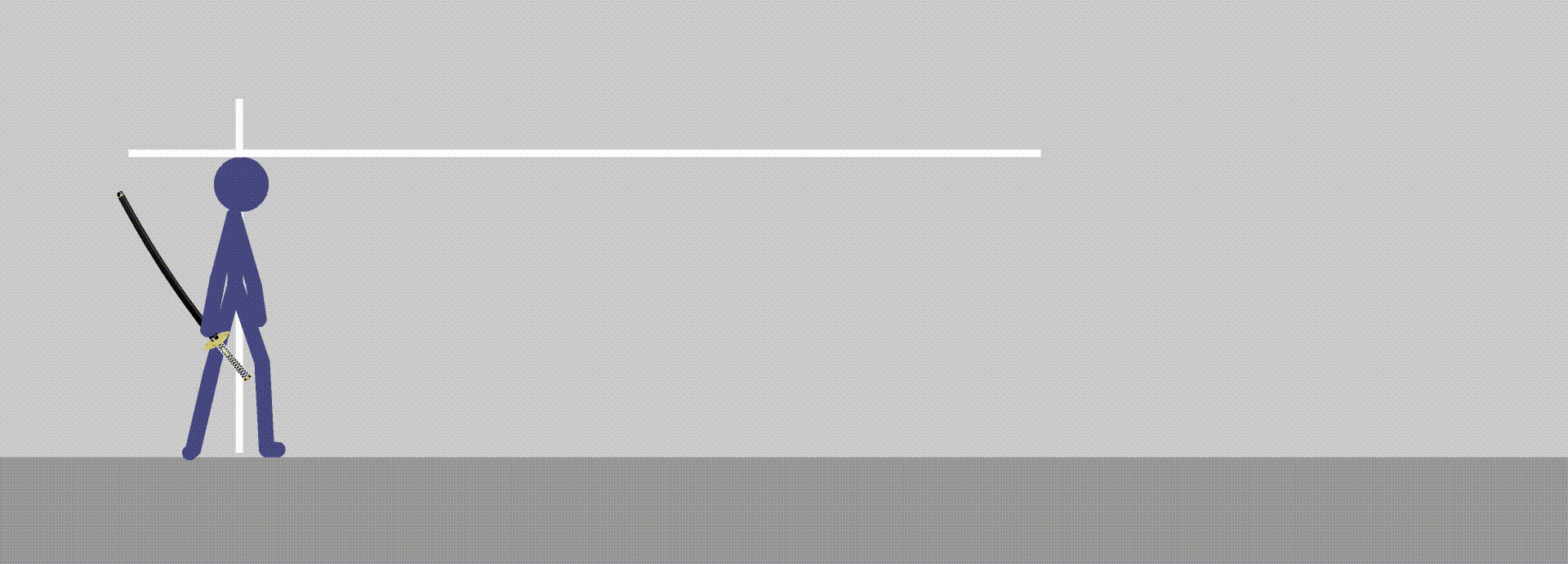
For example: from patrol to chase, the enemies have to find player. We define a method under enemy script so when player is 6 unity in front of enemy, player is considered as spotted.

**Attack detection system**

We first define child objects (attack) under the main characters in the game. After that, we added attack script to each character and choose proper colliders as attack detection. Then, we inactivate the colliders only in certain frames of the attack animation. The player’s attack child is also attached with rigidbody in order to give more precise detections.

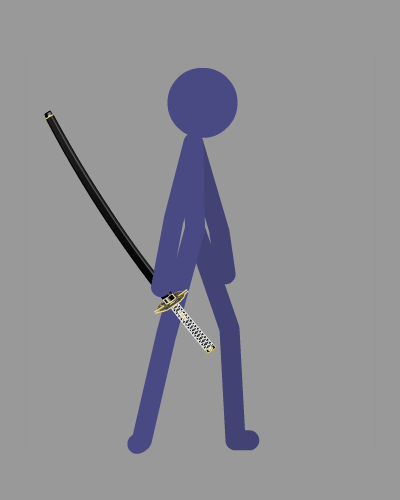
**Animation of Player and enemies**

We have implemented very smooth animations where player performs different movements.

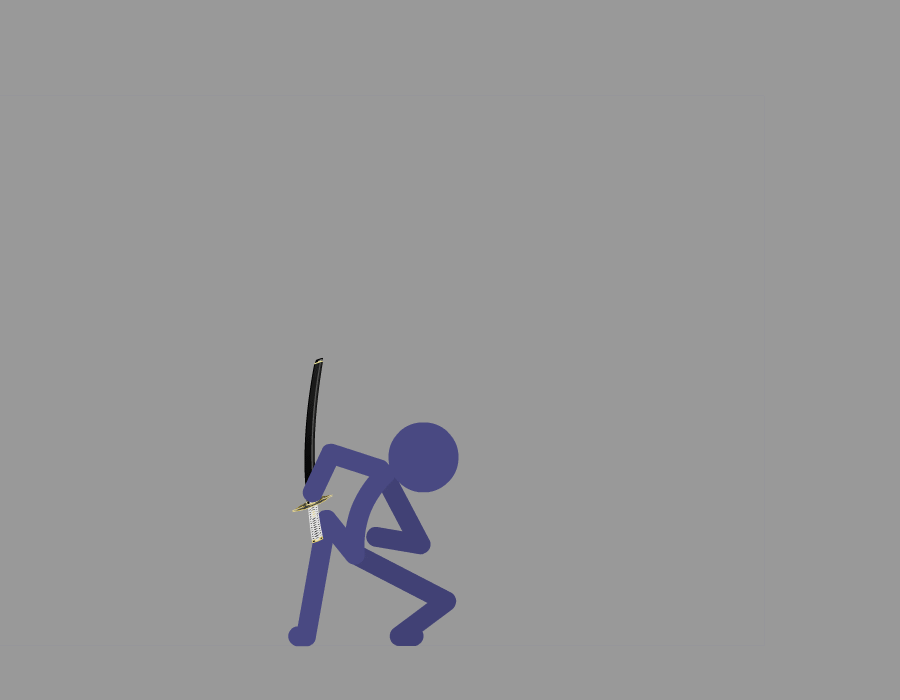


**Additional Player Animations**

**Crouch**

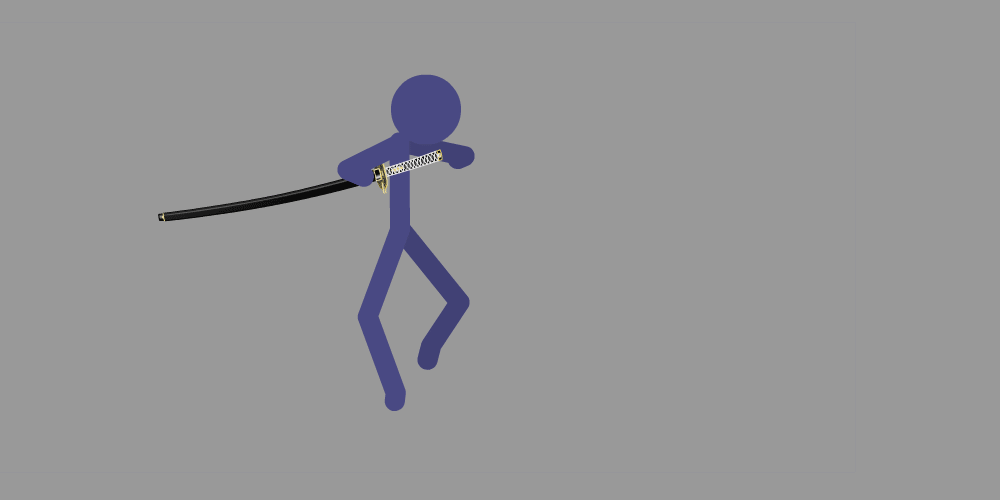
****

Player crouch is a key feature of our game. When Crouching player can perform skill Upward Slash and also jump higher.

**UpWardSlash**

Upward Slash will result in a knock up to the enemies where player can perform air attacks to combo.

**Air Attacks**

****

Players can now attack in air.

**Dodge**



Dodge allows player to escape from enemies’ attack. It also has a judgement of perfect dodge. When succeed, there will be additional judgement attacks to the enemy.

**Judgement Cut**

****

Animation will perform when perfect dodge succeeds.

Enemies has basic animations implemented. For axe, idle, attack, hurt, and walk animations are implemented. For Spear, walk, run, hurt, and idle animations are implemented.

**Test cases**

|  |  |  |
| --- | --- | --- |
| **Case** | **Expected outcome** | **As Expected** |
| Enemy state switching | Enemies can switch from patrol to chase smoothly | True |
| Enemy idle state | Enemies will wait at edges and when hitting walls | True |
| Axe Chase state | Axe will chase down player until lose sight of player or player is at attack range | True |
| Axe Encounter state | Axe will attack player when player is on ground and axe found player | True |
| Spear Chase State | Spear chasing state will be cancelled only when player is lose sight and the counter less than 0 | True |
| Player Attack | Player will be able to cancel attack and has smooth transitions between different stages of attack | True |

Above tests are carried out manually.