Hacker Frank

Team 11

2022.06.19

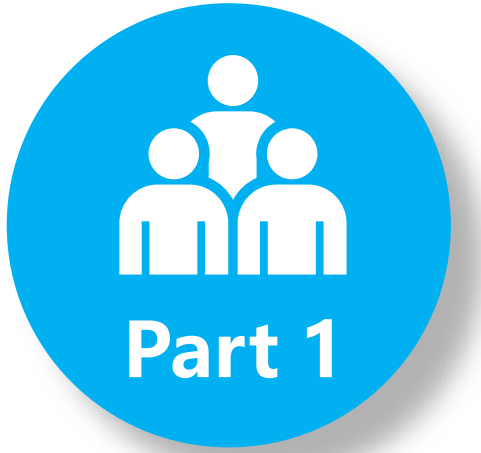
1 Introduction

2 Game Design

3 Challenges & Implementation

4 Game Test

5 Reflective Evaluation



Introduction

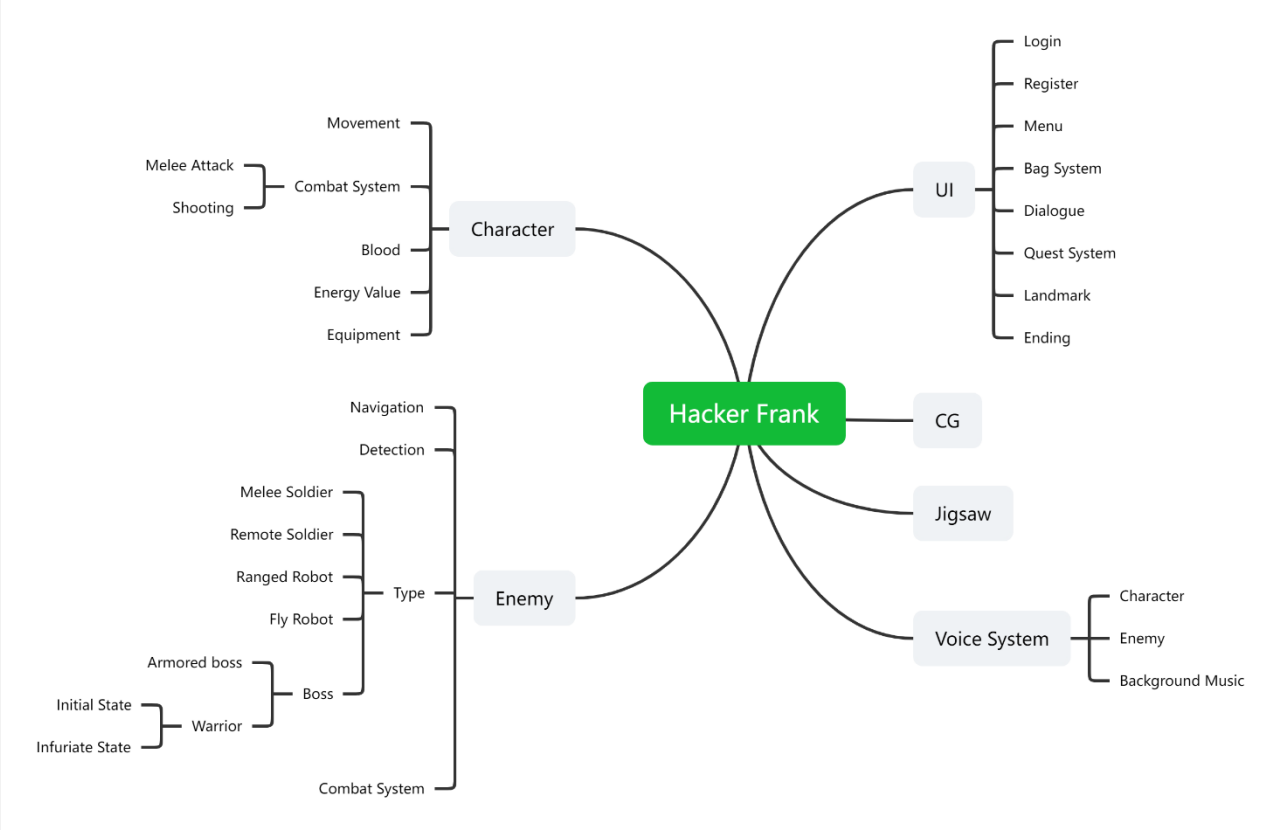
Introduction

Game Type

- RPG
- Third-person perspective
- Console game
- High quality

Story Line

- AI vs Human
- Chip drawing – key to defeat AI
- Defeat robots to get chip drawing



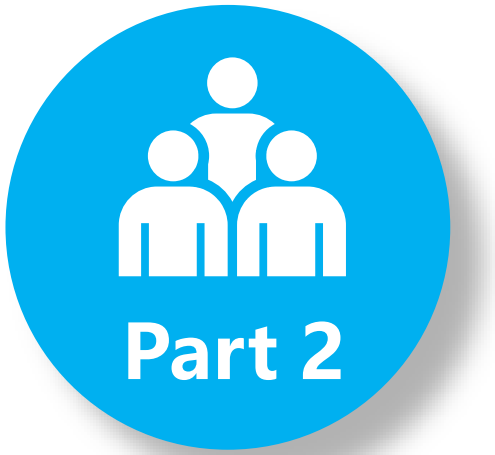
Game Rule

Final Task

- Get the chip drawing

Steps

- Use sword and gun to defeat enemies
- Collect items drop from the enemies for recovering
- Pick up fragments in the map
- Defeat boss to get better equipment and fragments
- Piece up the fragments to get the chip drawing



Game Design

Game Manager

UI System

Log in

Register

Menu

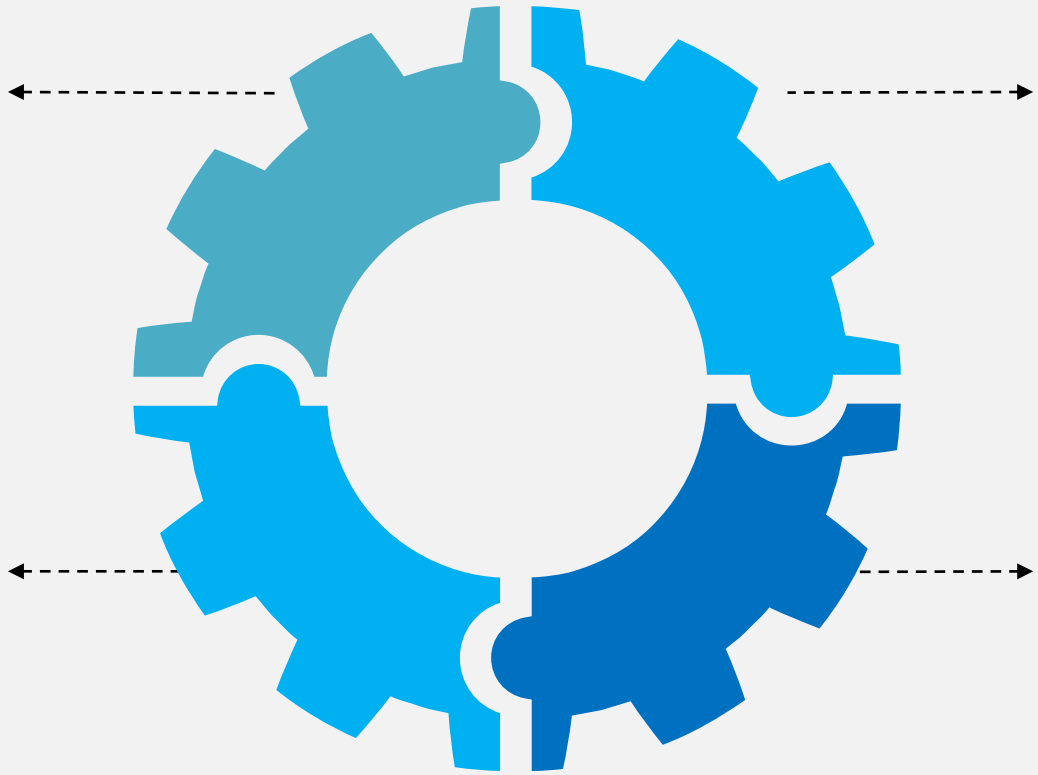
Status Information

Action System

Movement

Melee Attack

Shooting



Voice System

Background Music

Environmental Sound

Character sound

Game Process

Log in

CG

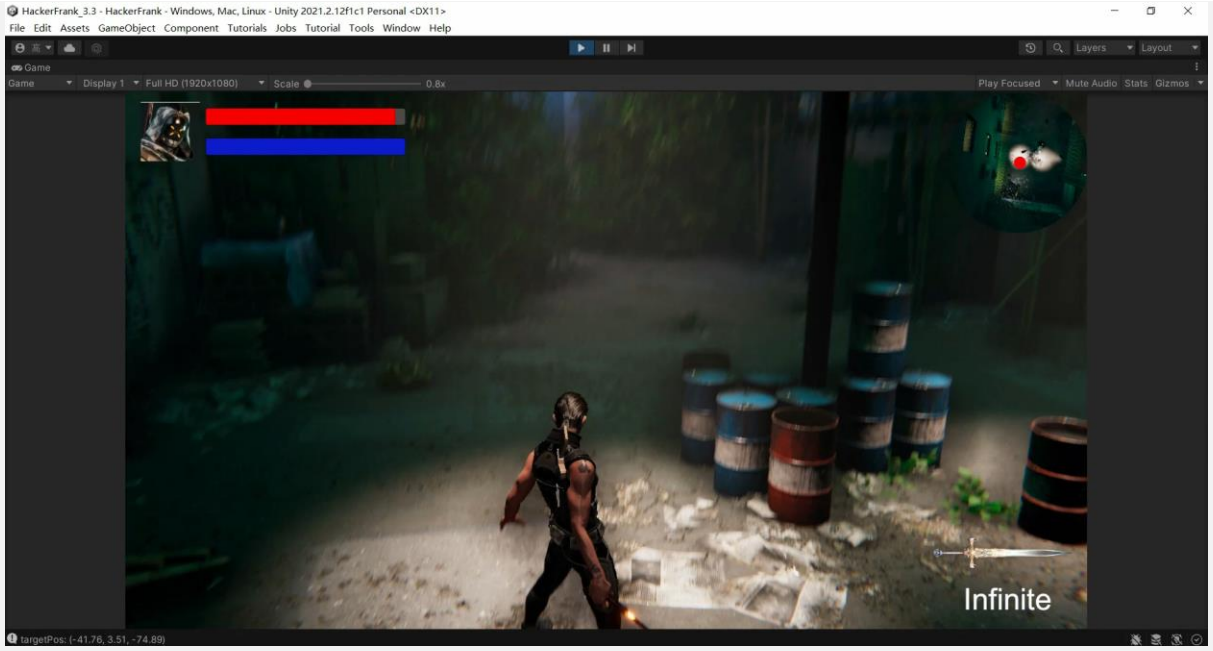
Main Game

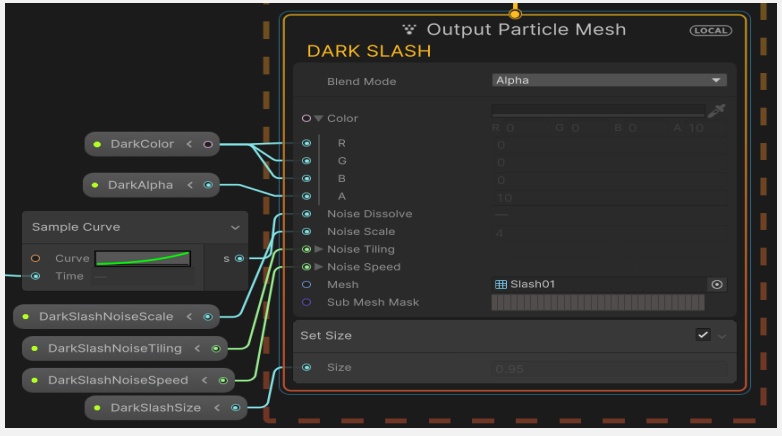
Ending

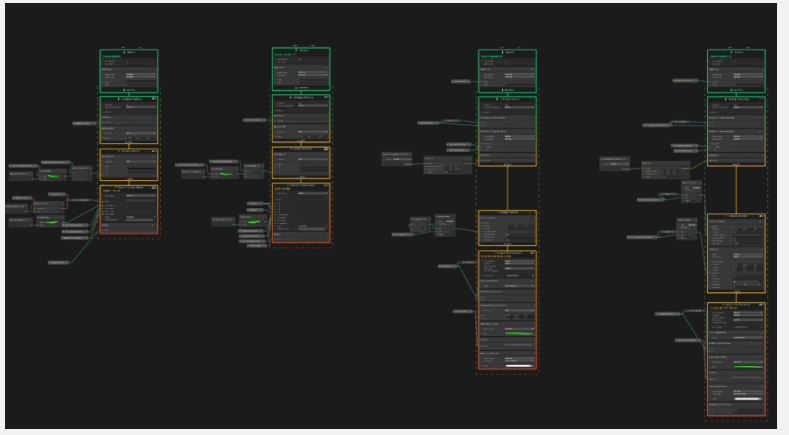
Melee Attack

Three combo

- Fluency
  - “CorssFade” method
- Visual Effects
  - “VFX” effects



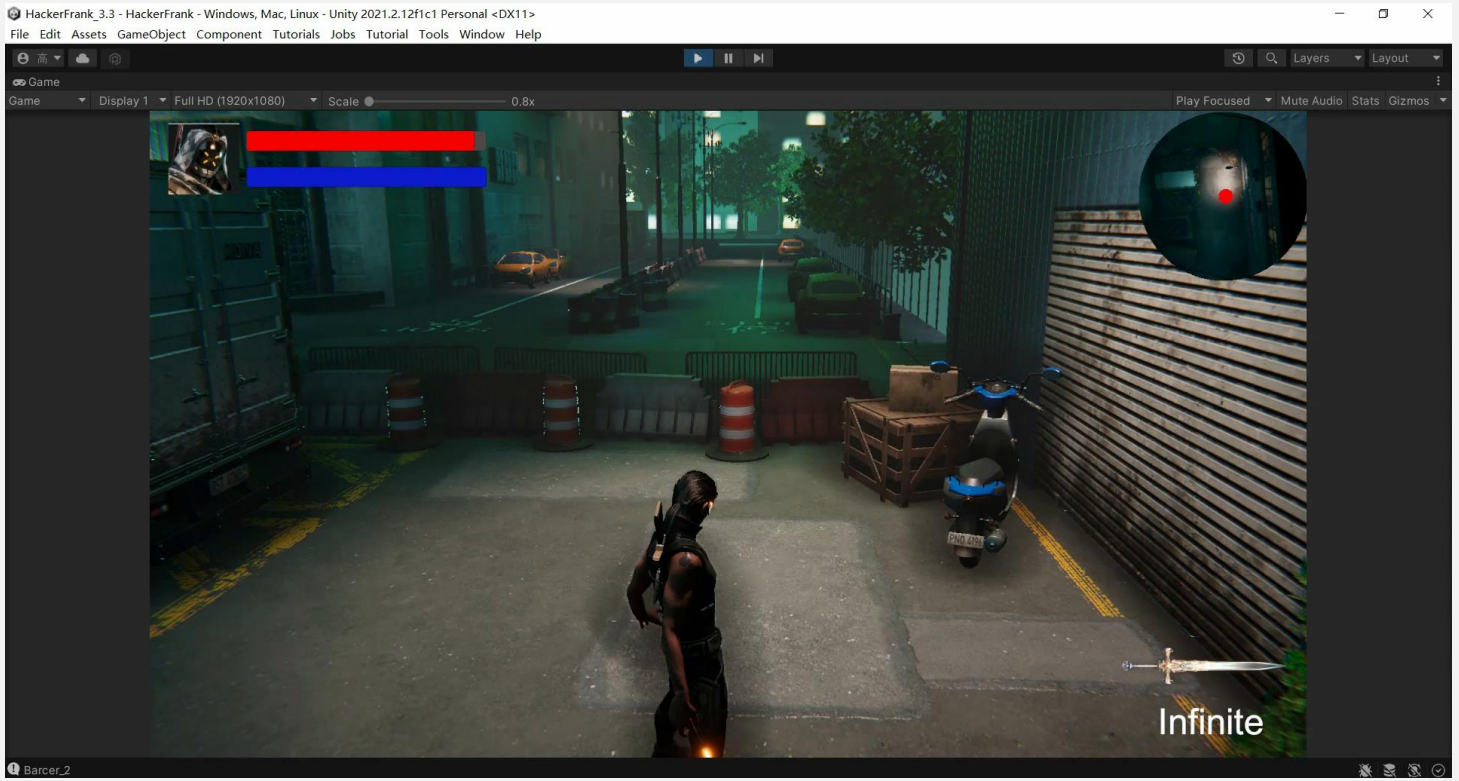




Melee Attack

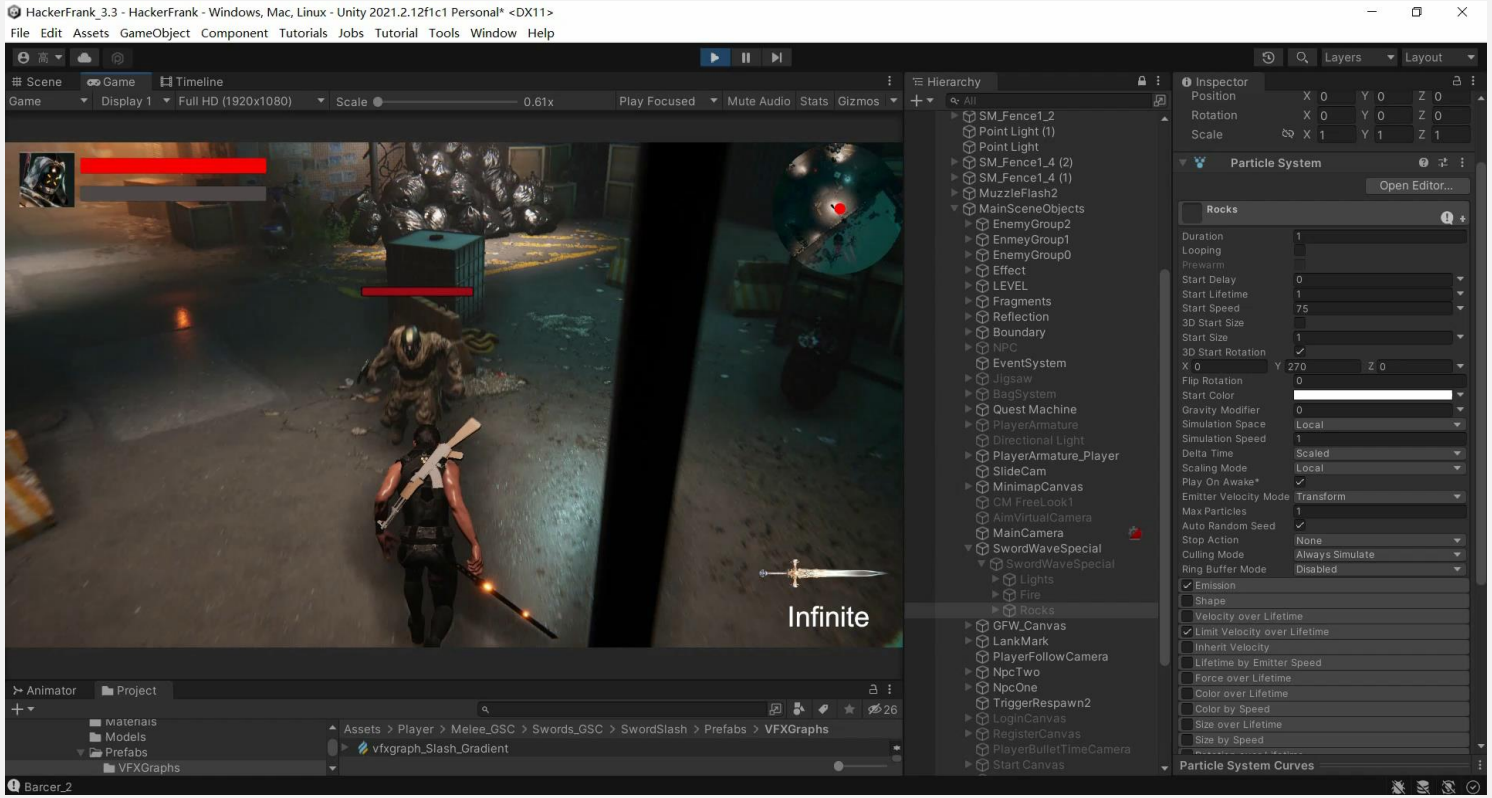
Special Skill

- Particle System
  - Sword wave
  - Light
  - Fire
  - Rock



Positive Feedback

- Energy Value
- Item drop
  - First aid kit
  - Ammunition box
  - Better weapons

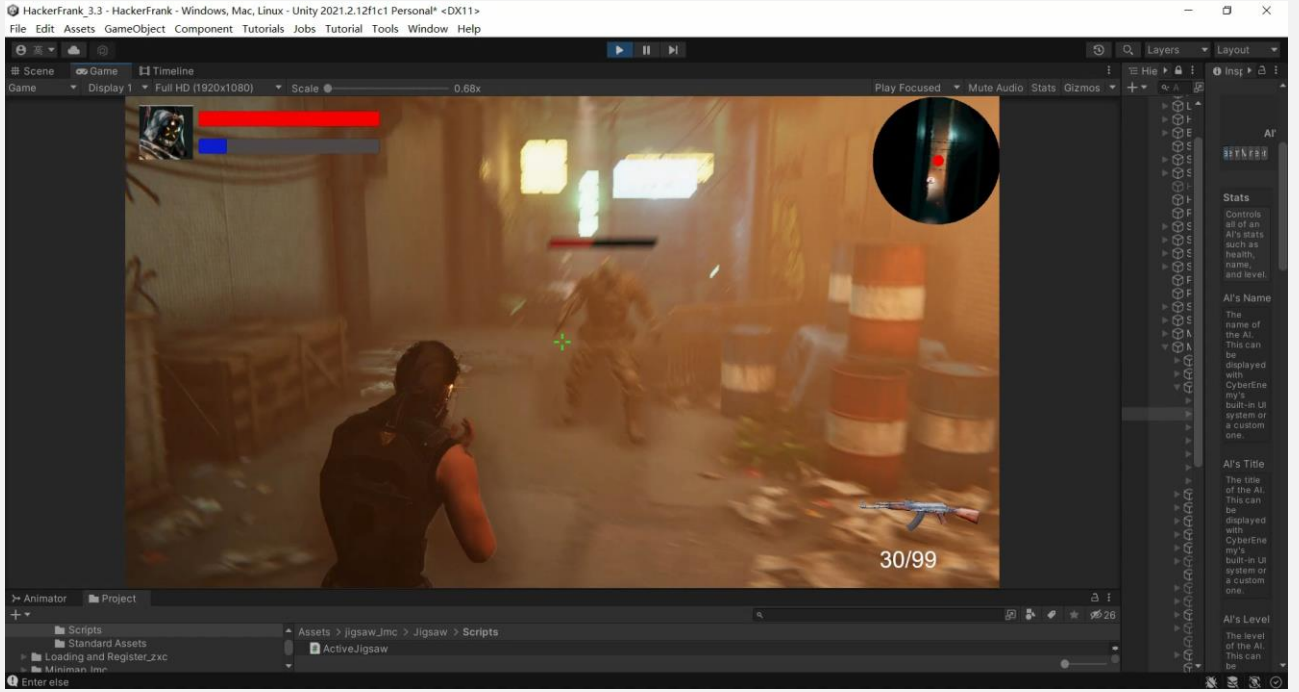


Third Person Shooting

Key Technology

- Cinemachine
  - 3 virtual cameras
  - Smooth switching of 3 state
  - Holding a gun, aiming and shooting

- 2 Animation layers
  - 2 layers of animation
  - Combination of 2 animations
  - Fluent action



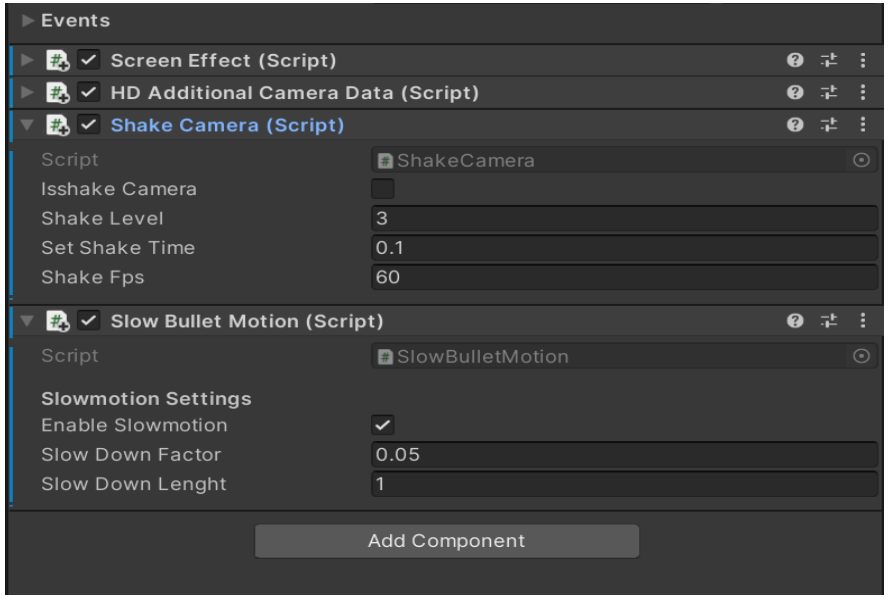
Third Person Shooting

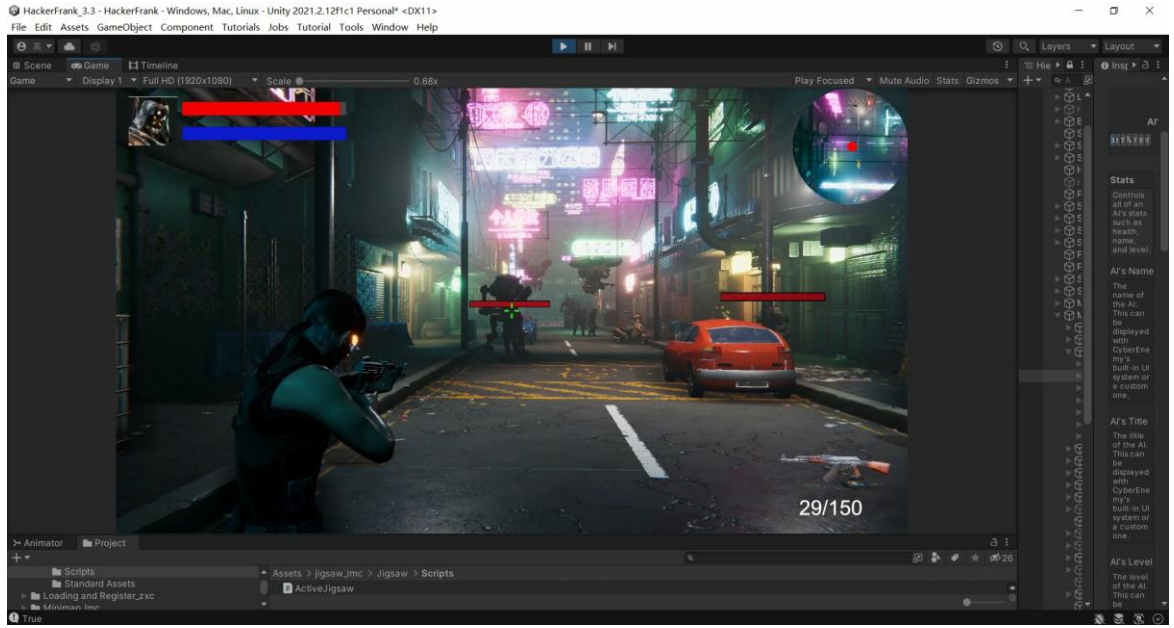
Skill – Bullets Time

Press Q button

Slow down the time

Increase the damage value





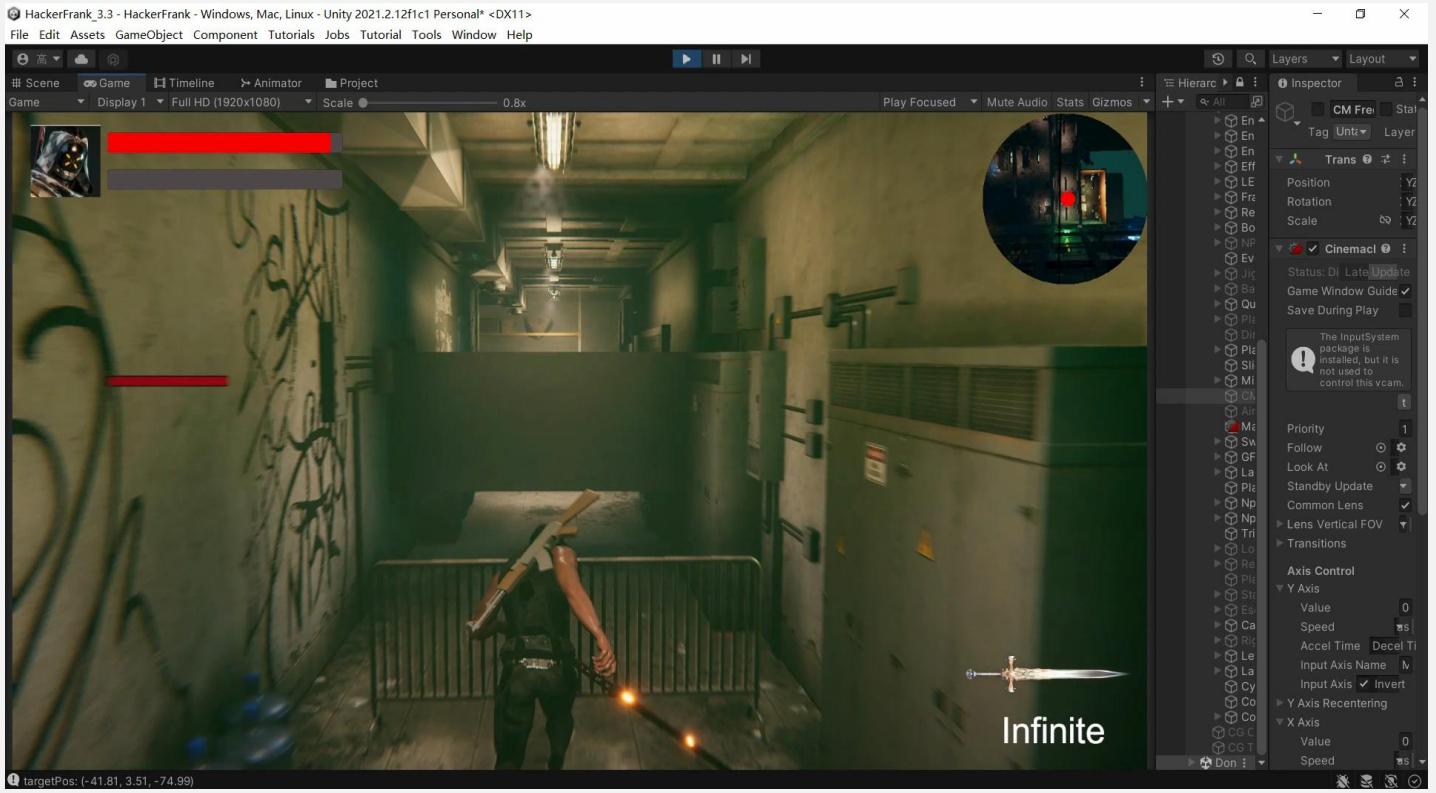
Camera Shake

Simulate the shaking of a gun as it fires

Increase the user's experience

Parkour System

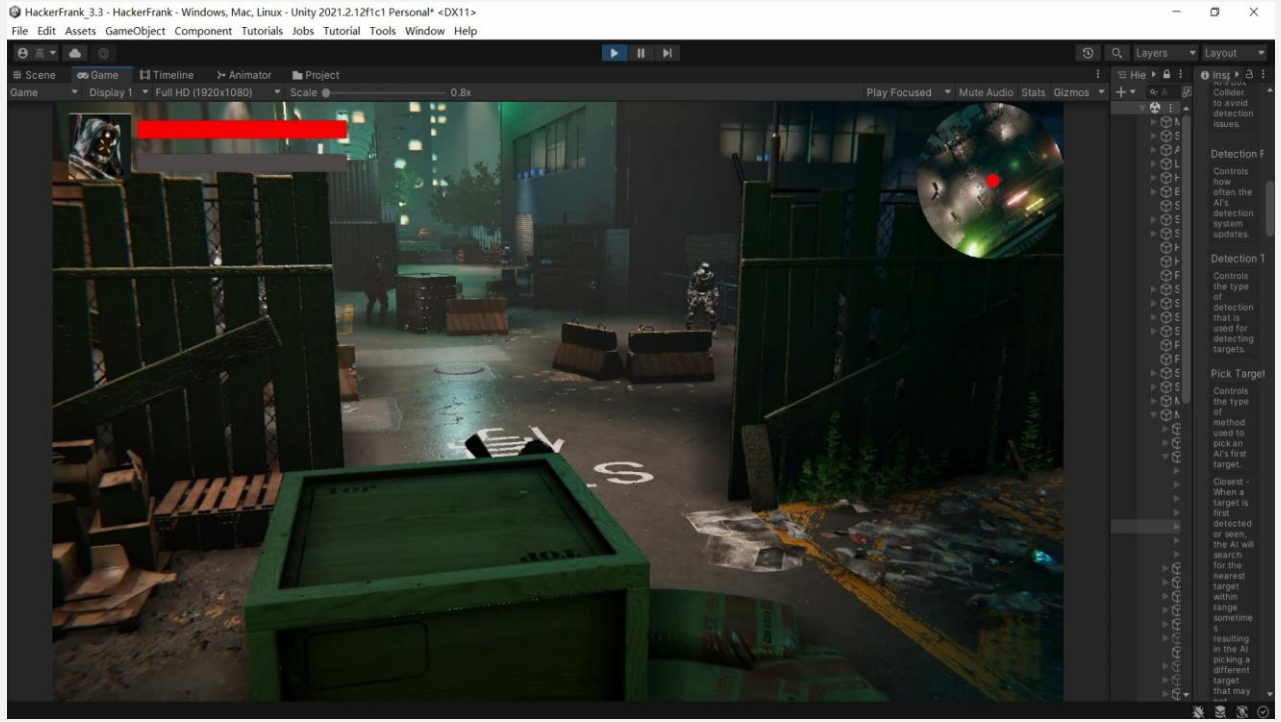
- Vaulting
  - Use IK system to adjust the position of hand
  - the Animation can be suitable for Obstacles of different size
- Sliding
  - Use cinemachine and camara switch



Enemy AI System

Default Behaviors

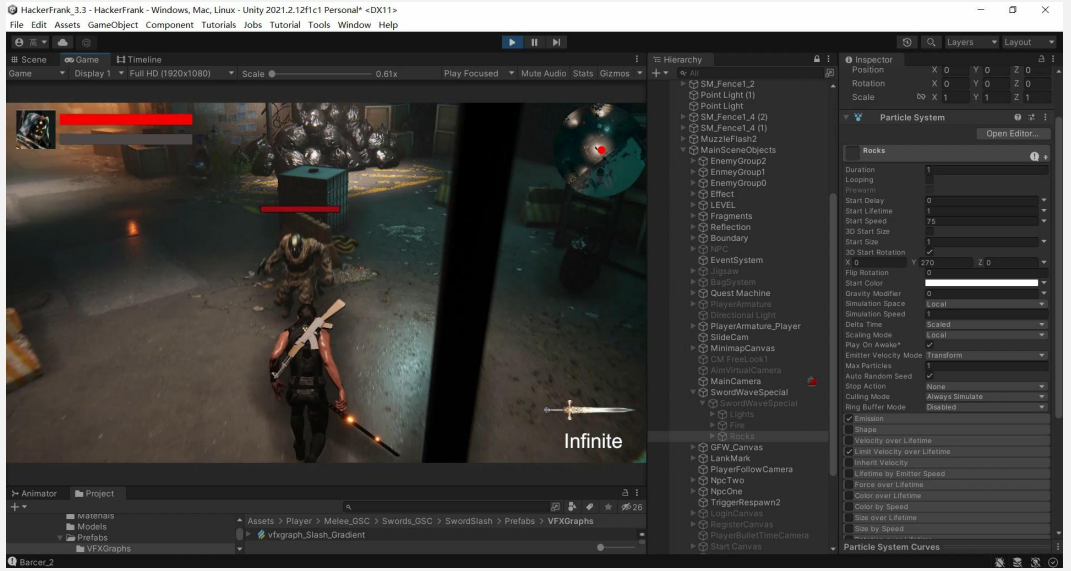
- Patrolling
  - Randomly spawn waypoints in an area
  - Patrol between waypoints by navigation system
- Detecting
  - The enemy keep detecting player.
  - Once detect player tag, perform attack behavior
- Chasing
  - Use unity navigation to follow the player target
  - Once lose target, the enemy will return to their territory

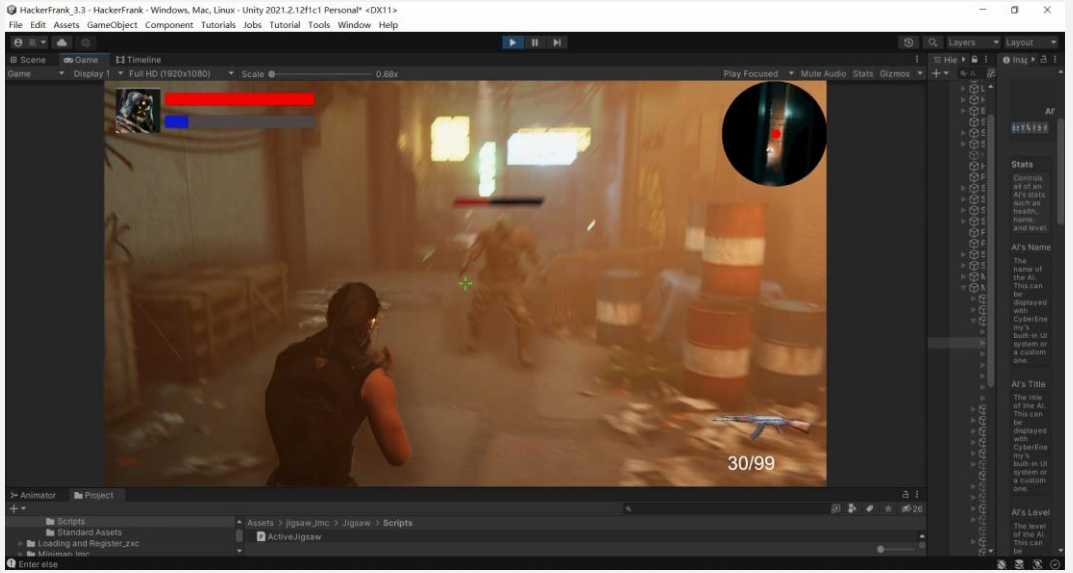


Enemy AI System

Combat Behaviors

- Attack
  - Once catch player, the enemy will play attack animation and hurt player
- Back up
  - Once being attacked or too close to player, the enemy will back up to avoid damage
- Hit recover
  - The duration of being unable to move or attack after being attacked.
  - Implemented by interruption

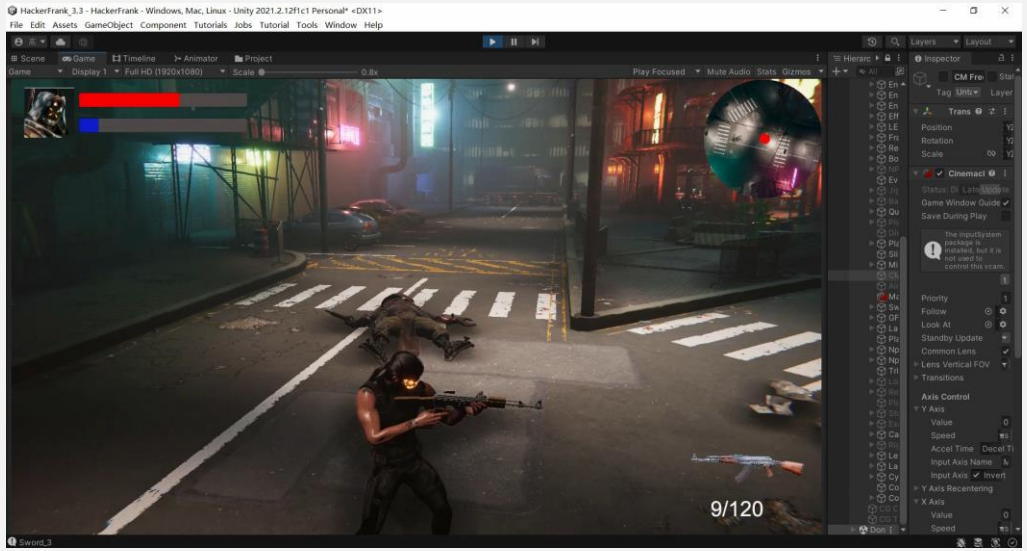


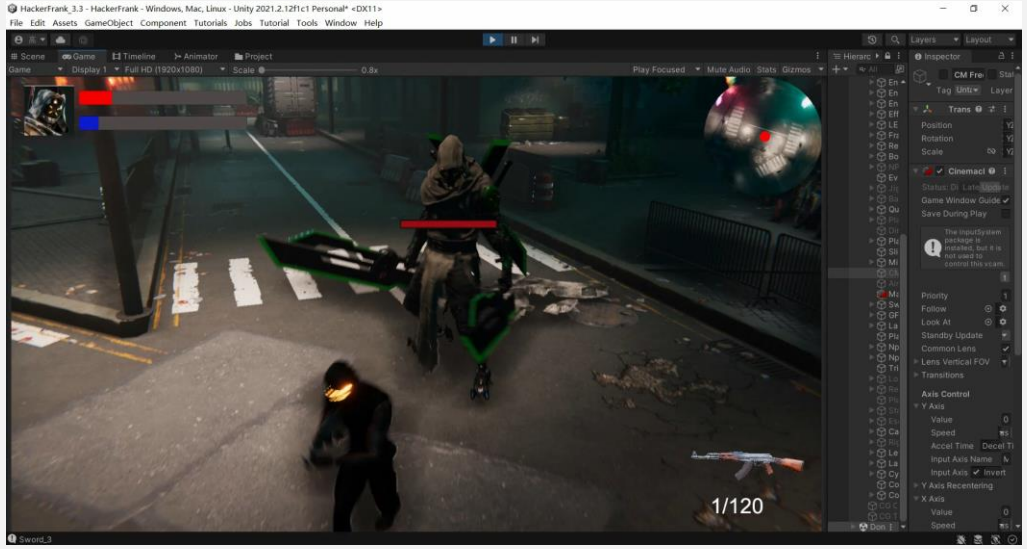


Enemy AI System

Boss Features

- Different stages
  - Different forms have different characteristics
  - Stage1 is bulky and stage 2 is aggressive
- Special skills
  - Add “VFX” effects for better visual effects
  - Use effects to inform player so that they can dodge the damage





Improve Efficiency

Unity Coroutines and Object Pool

- Coroutine
  - Coroutines are lightweight threads with almost no performance overhead

```
//Return the AI to its starting destination to continue wandering based on it WanderType.
StartCoroutine(DelayReturnToDestination(0));
```

```
IEnumerator DelayReturnToDestination (float DelaySeconds)
{
    yield return new WaitUntil(() => CyberEnemyComponent.CyberEnemyLookAtComponent.BodyWeight == 0);
    yield return new WaitForSeconds(DelaySeconds);
```

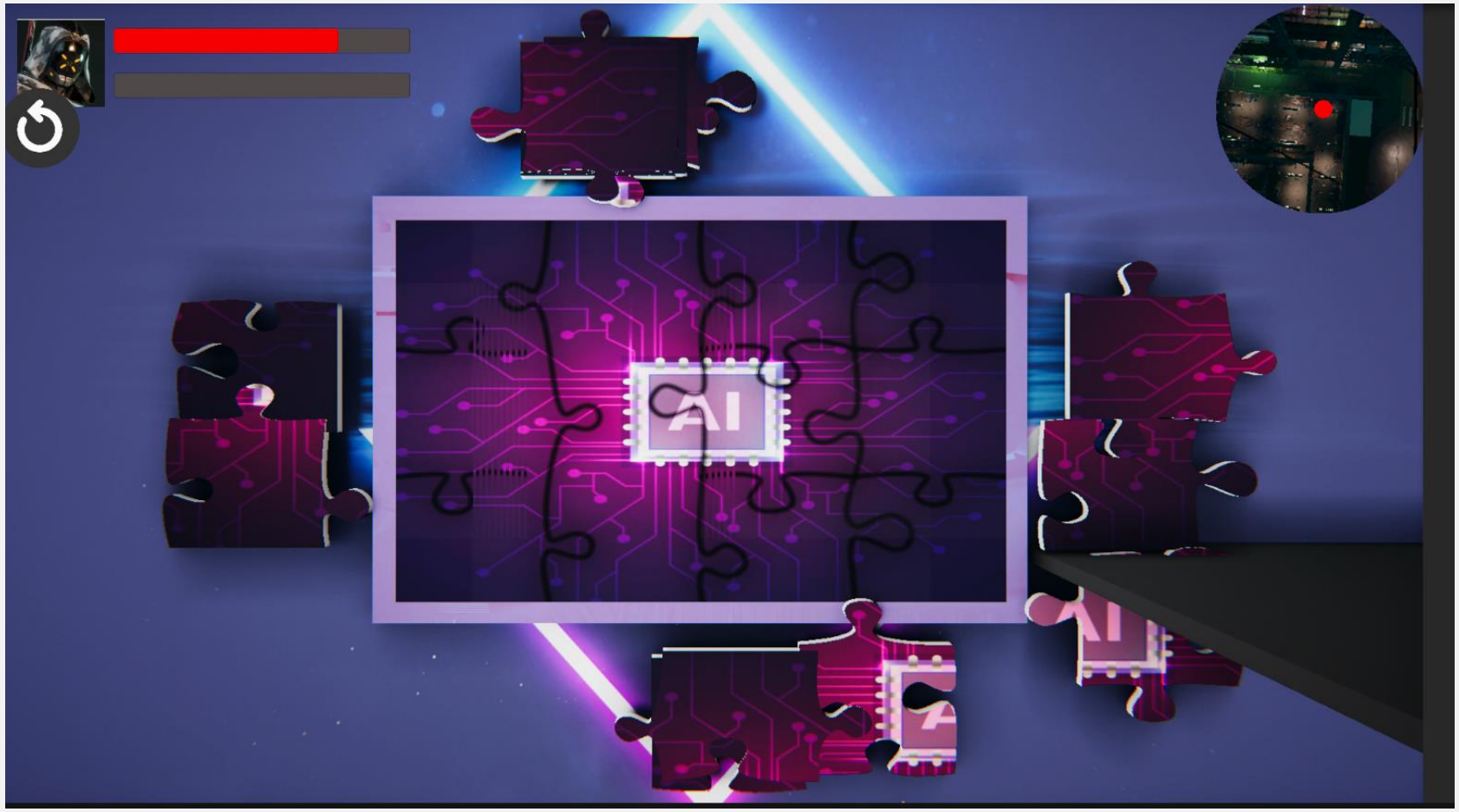
- CyberEnemyAIObjectPool
  - Singleton object pool, reduce the memory overhead

```
public static class CyberEnemyAIObjectPool
{
    const int DEFAULT_POOL_SIZE = 3;

    /// <summary>
    /// The Pool class represents the pool for a particular prefab.
    /// </summary>
    5 个引用
    class Pool...


    /// <summary>
    /// Added to freshly instantiated objects, so we can link back
    /// to the correct pool on despawn.
    /// </summary>
    Unity 脚本|4 个引用
    class PoolMember ...

    // All of our pools
    static Dictionary<GameObject, Pool> pools;
```

Jigsaw

Why add it

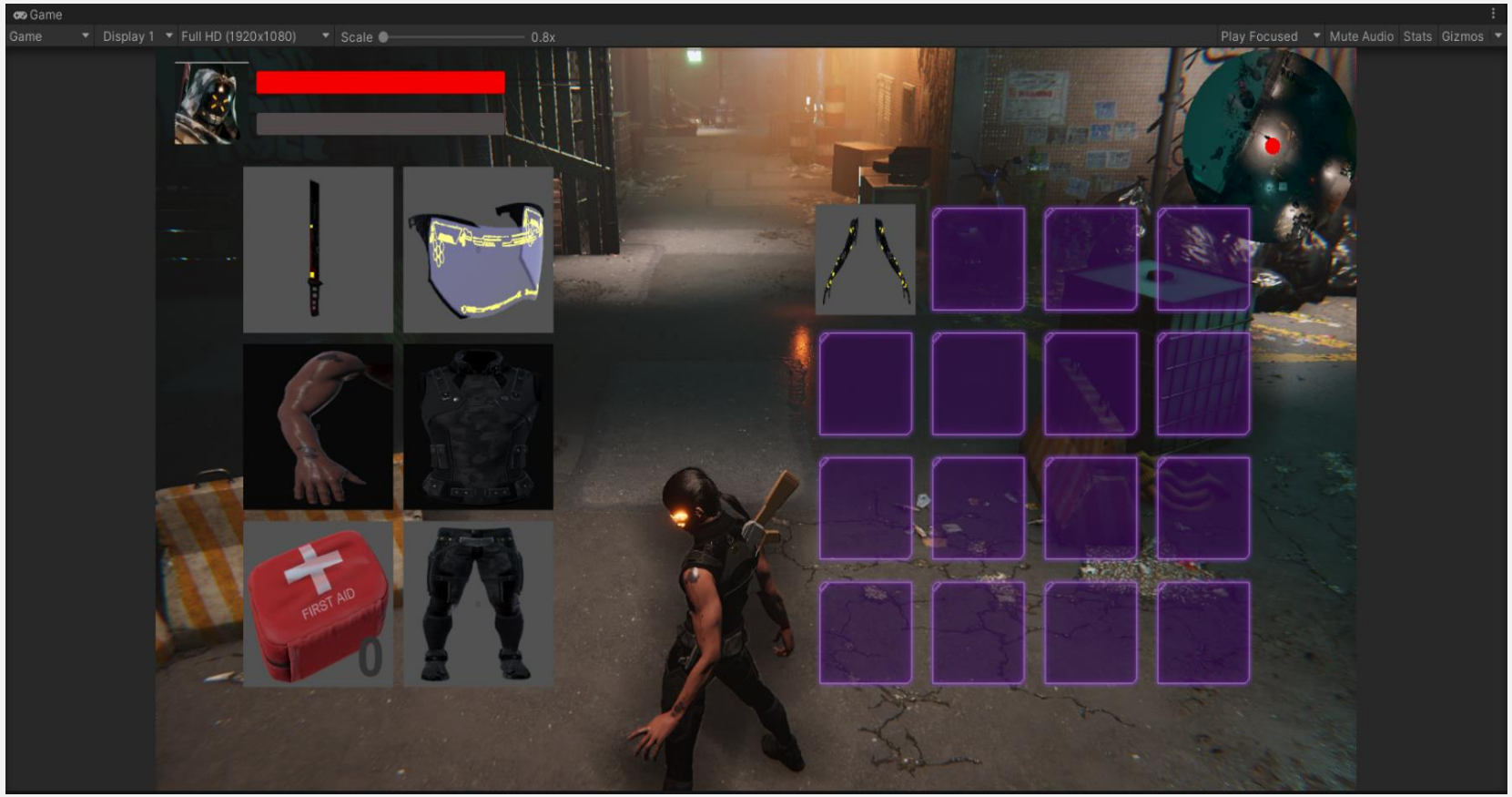
- As clue of game
  - Collect fragments around the word
- Restate theme of game



Inventory System

Functions:

- Store objects that player picked and do a count
- Switch weapons



Task System

Giving Task

- NPC gives task to player

Task Journal

- Show all tasks that the player has accepted
- Show task process

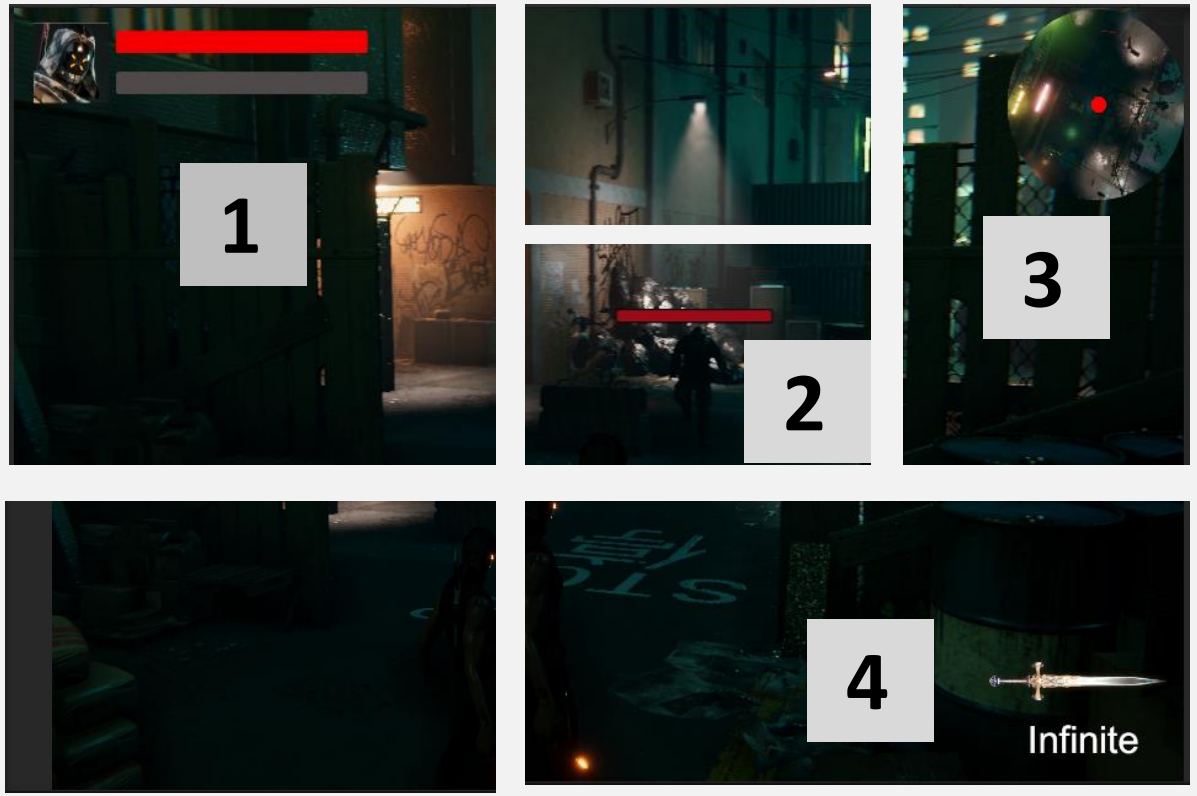
Game Senses

Status information

- Health point and energy value of player 1
- Health point of enemies 2
- Mini map 3
- Current weapon in use 4

Limit range of movement

- Avoid player leaving game sense

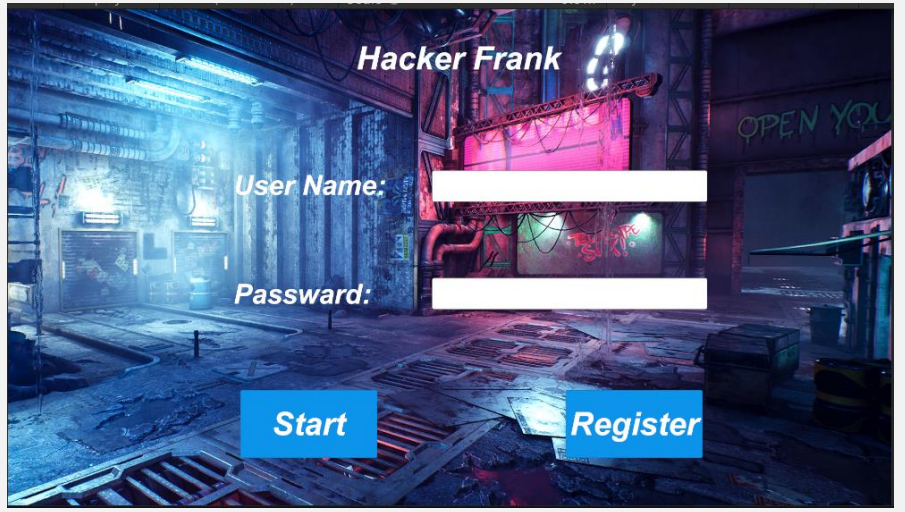


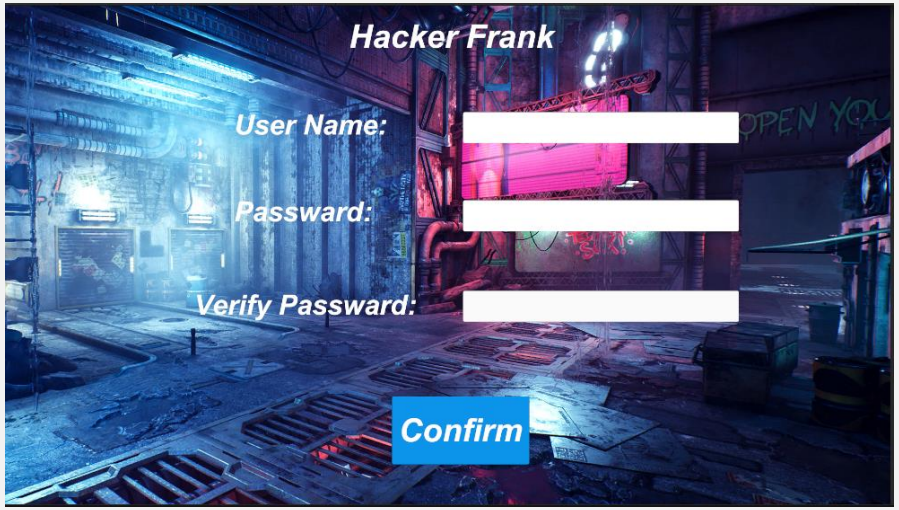
Start Menu and Control Panel

- Start Menu
  - Click the Start button to enter the login interface
- Control Panel
  - Press “ESC” button to pop the key manual

Login and Register System

- Collect user names and passwords



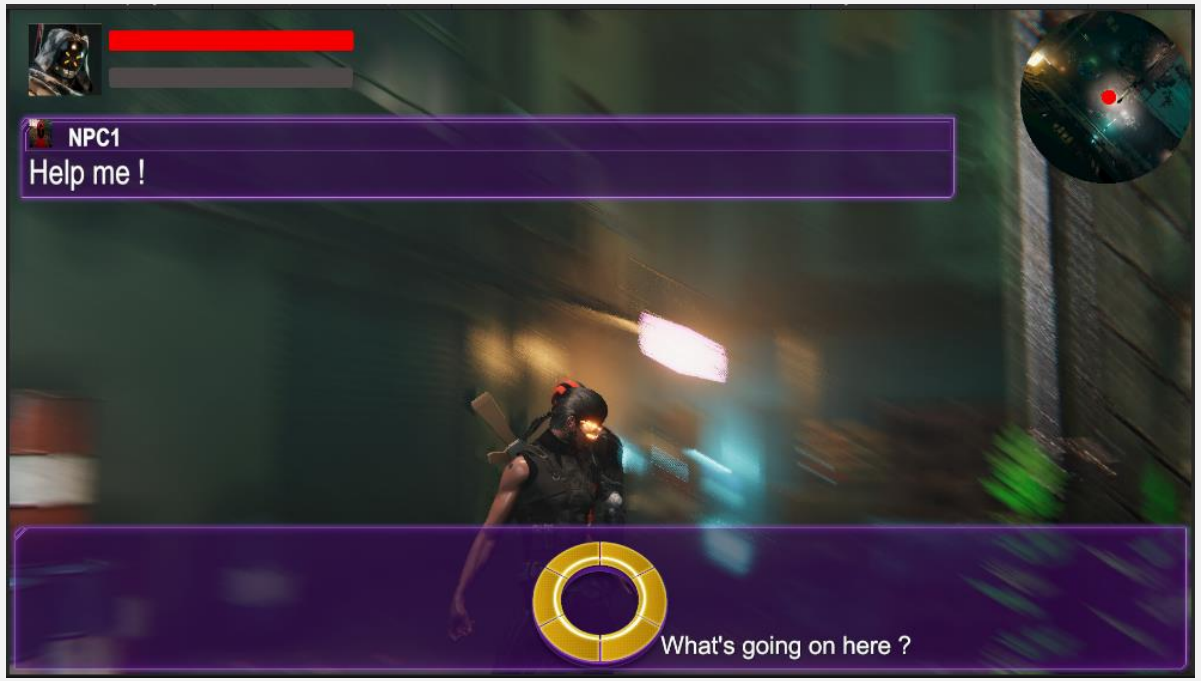


- Use a local database for storage

Dialogue System

Functions:

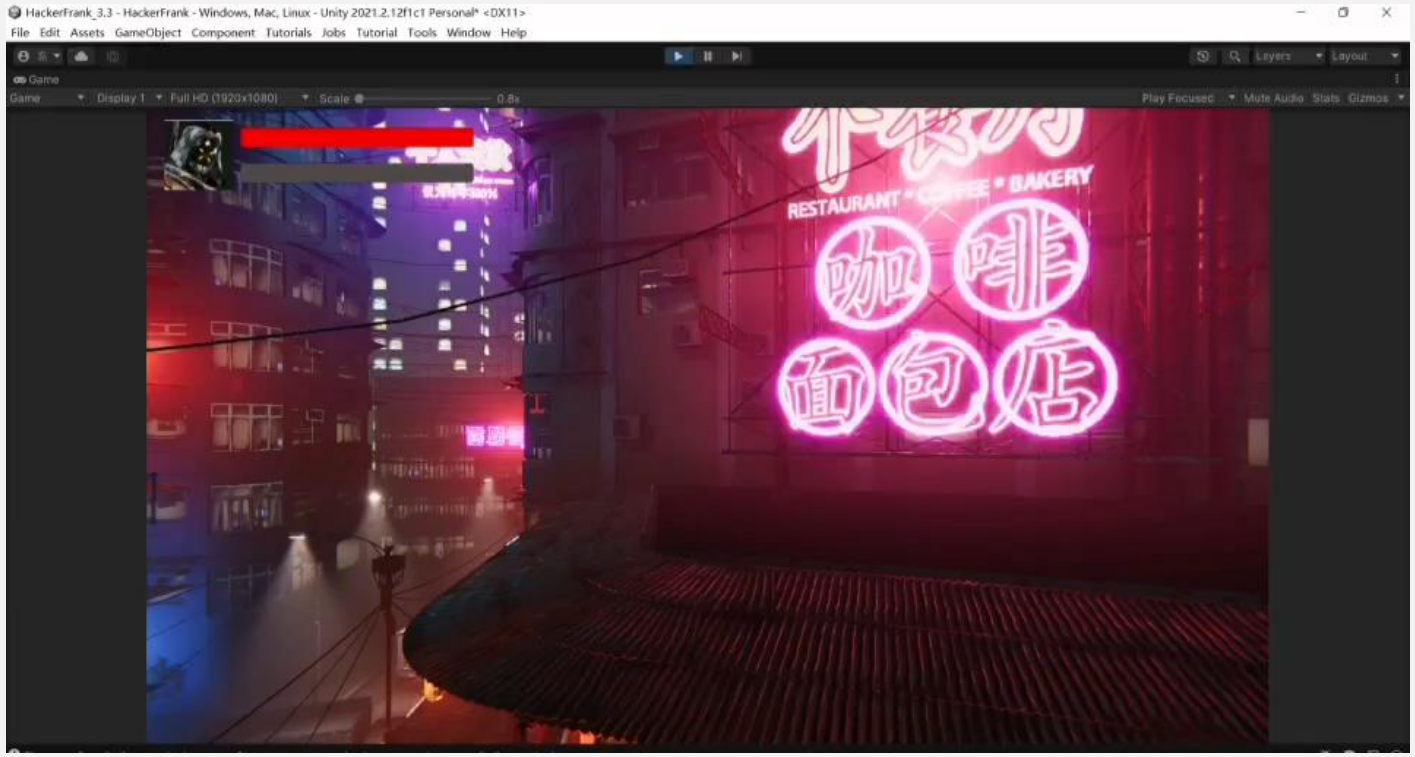
- Select conversations with NPC
- Publish tasks through interaction

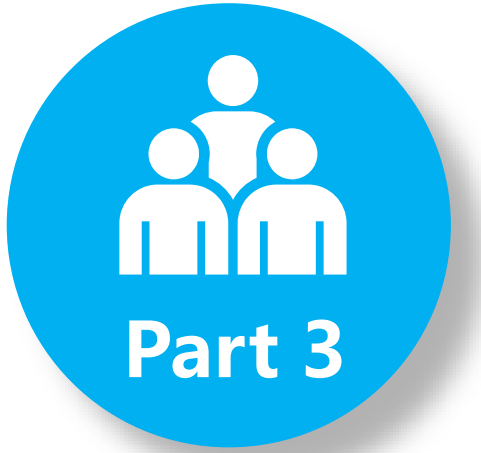


CG

Functions:

- Introduce the background
- Show game scene
- Enhance the fun

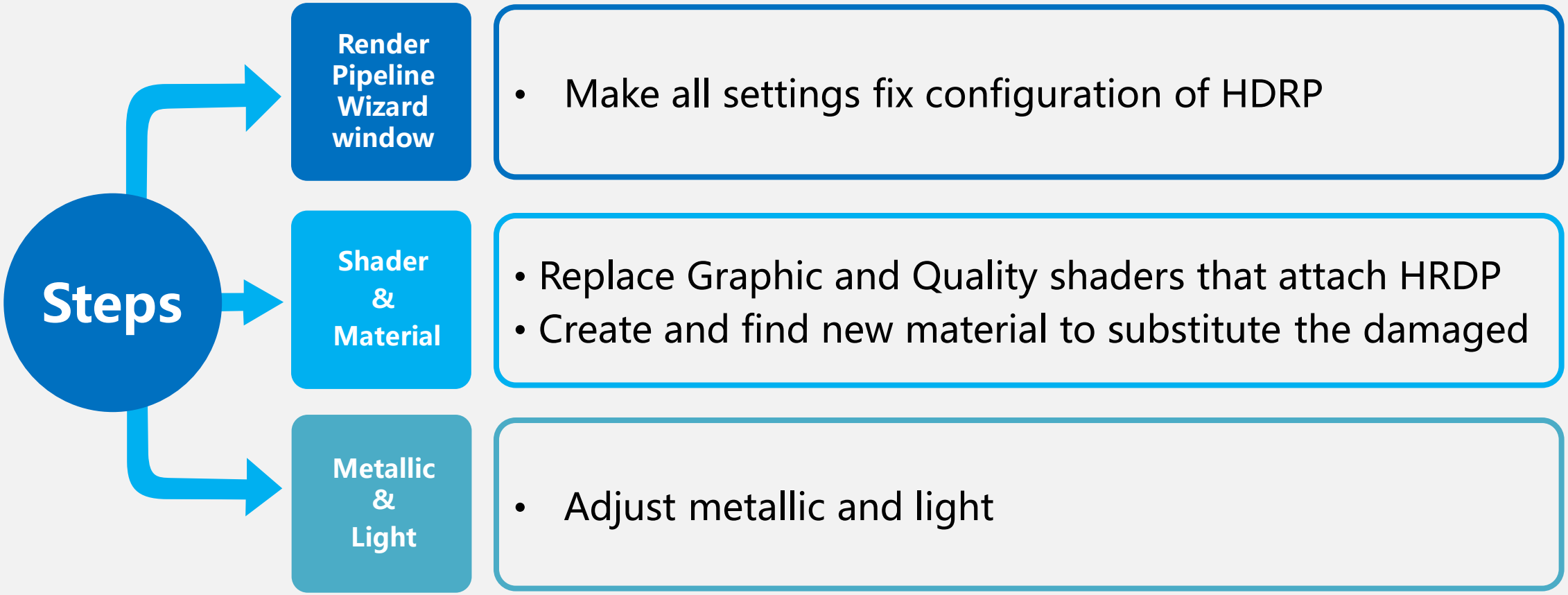




Challenges and Implementation

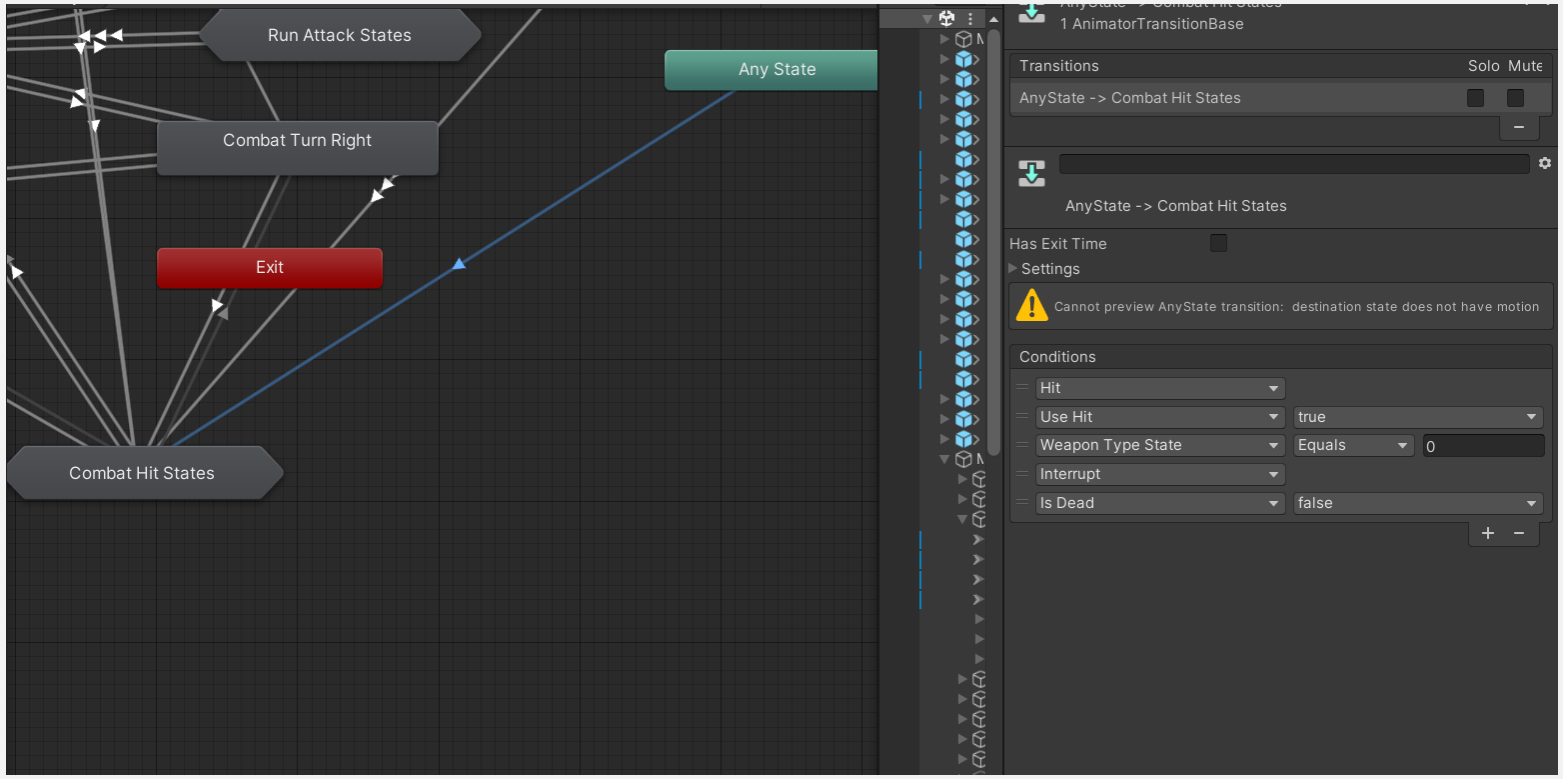
Challenge

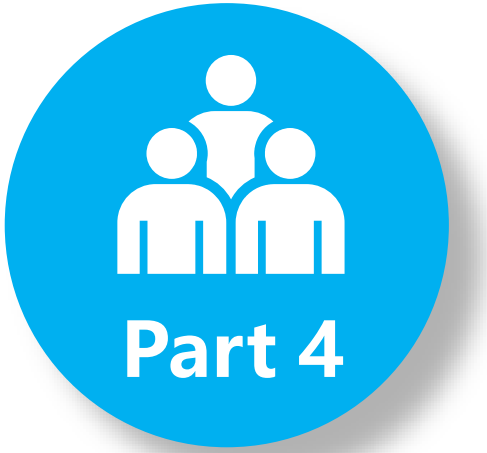
- Update to HDRP



Challenge

- Implementation of Hit recover:
  - Using interruption in animator
  - When enemy being attacked in hit states
  - Interrupt animation with any state



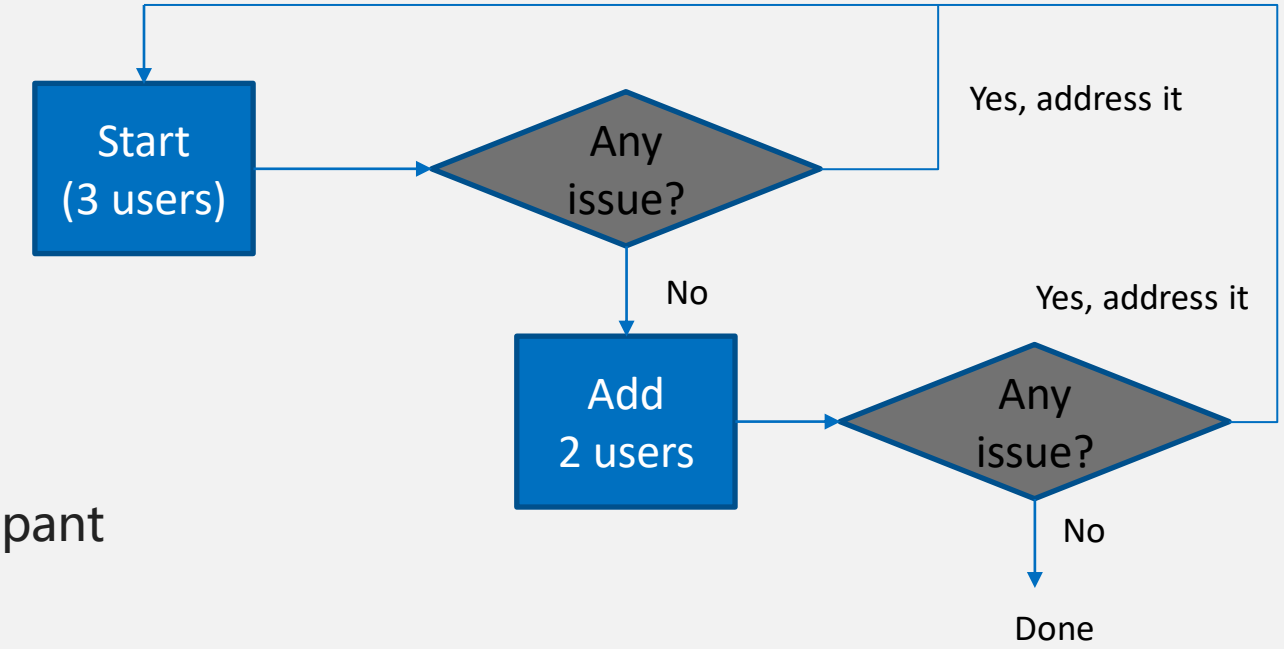


Game Test

RITE Test

Procedure:

- **Step 1: Start with 3 participants**
  - If found issue, address them and repeat Step 1 with 3 new participants
  - If not found, go to Step 2
- **Step 2: Add 2 more participants**
  - If found issue, address them and start at Step 1 again with 3 new participant
  - If not found, finish



RITE Test

Iteration Records

- Start date: 16th May
- Total round: 7
- End date: 10th June

Issue	N1	N2	N3	N4	N5
Player can still shoot after death	X		X		
Sword cannot be taken out correctly if the player is equipped with the gun				X	X
Player cannot move after respawn	X	X			
If player plays attack animation in the air, he cannot play the landing animation correctly.				X	
The damage of enemies is unreasonable. Boss damage is too high, soldier damage is too low.		X	X		
The actual attack range is different from the attack animation.					X
The position of the player in the Timeline returns to the original point.	X		X		
The pieces of Jigsaw will disorder if the bord of Jigsaw placed overlap with other objects				X	

Iteration1 | Iteration2 | Iteration3 | Iteration4 | Iteration5 | Iteration6 | Iteration7

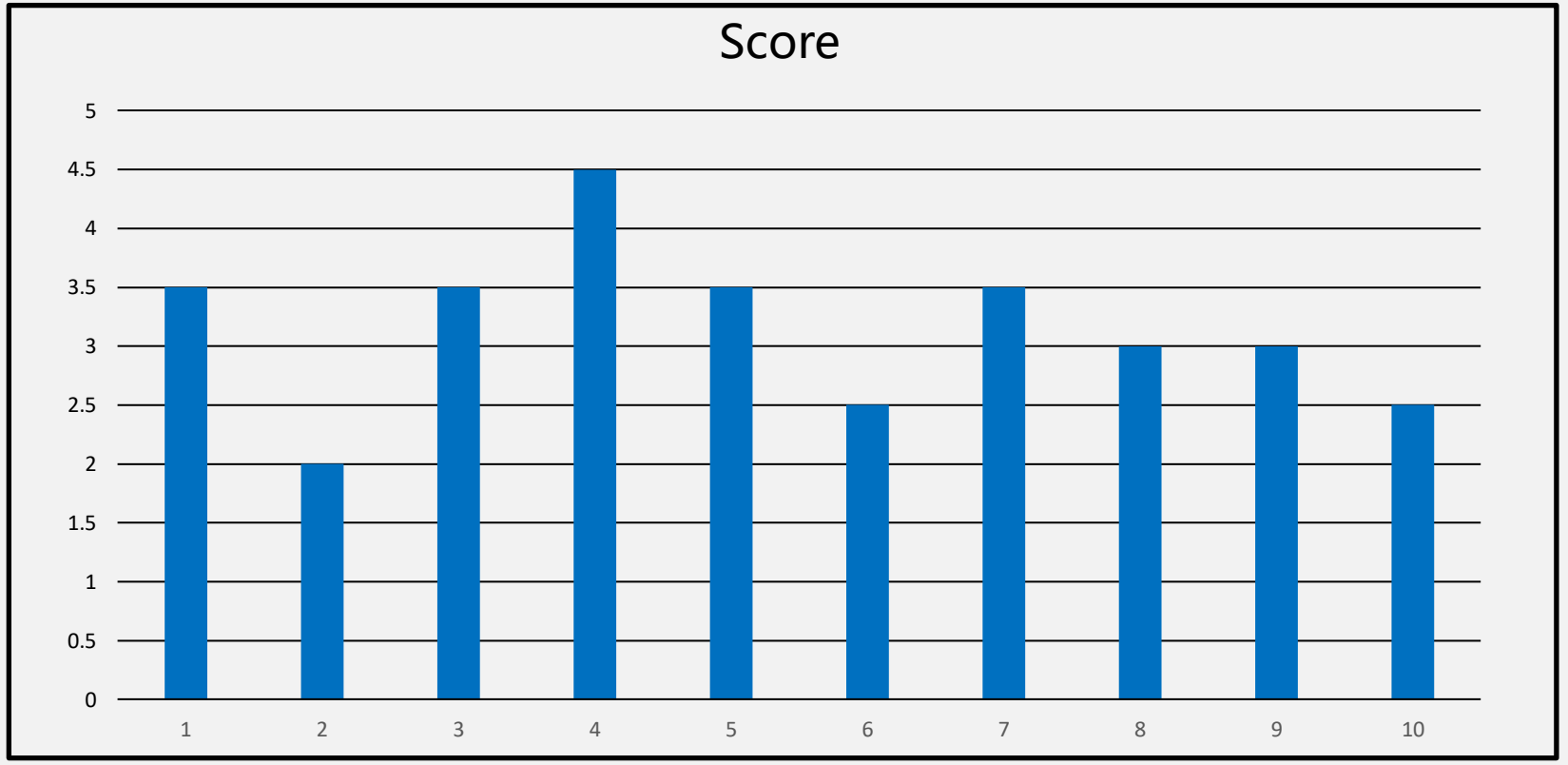
Satisfaction Test

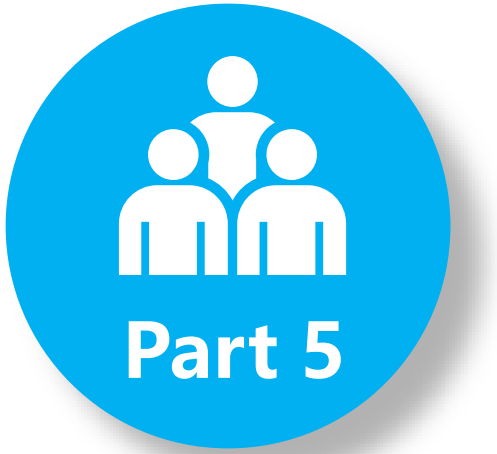
Procedure:

- Live broadcast of the game
- Distribute questionnaire in the broadcast room
- Use SUS to carry out the satisfaction survey

Test Result

- The result of SUS is 78.75





Reflective Evaluation

Bright Spots of Game

- Use HDRP, make the game more realistic
- Improve game efficiency
  - Unity Coroutines and Object Pool
- Login and register system connect to the local database
- Positive feedback mechanism
  - Gain experience when attacking an enemy with general attacks
  - Different items have different attack bonuses
- Diversity of skills
- Have three check points
- Original code is about 10000 lines

Reflection of Cooperation

Reflection

- The lack of experience in game development
- An unreasonable division of labor at the beginning
- Incongruity in material rendering methods

Improvement

- Sufficient group communication
- Two meetings a week at the early stage
- Focus on work at the last stage


THANKS
Team 11