Project Title : Lost in Dungeon

Project Group : Team-D

Final Report

Group Members :

Jiabo Zang , SID :200395447

Zhimu Li , SID : 200362915

Program : Software System Engineering

Presented To : Dr. Timothy Maciag

Course : ENSE400/477

Course Title : SSE Capstone

Date : 2023-4-7

University of Regina

Background

Nowadays, most people's daily lives become more and more stressful, so entertainment is an irreplaceable and necessary element in our society. And the different types of games become the most common and effective solution to release pressure for many young people.

Introduction

Our project is about creating a dungeon game with 10-20 minutes of play time, so players can enjoy and pass the game during a short break. And the randomness in the dungeon can make the game more challenging and interesting, so players can have different experiences in every different round.

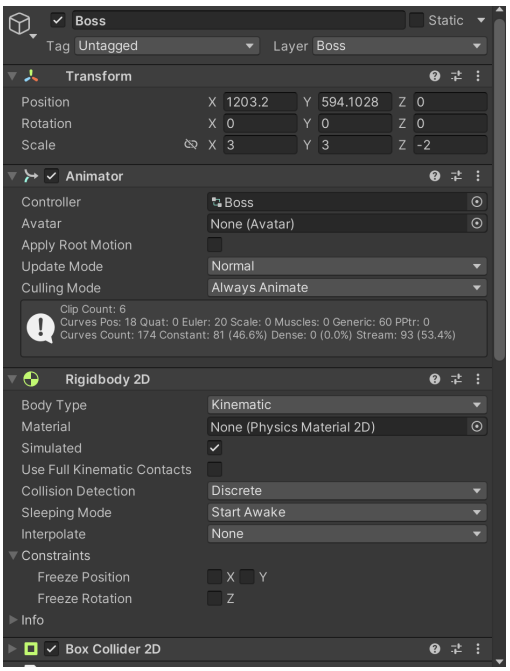
Lost in Dungeon is a rogue-liike game that develops on Unity engine. There are three main types of game scenes, and they are respectively village scene, dungeon scene and battle scenes. The village scene and dungeon scene will be 2 different styles of 2D map models, and battle scenes will be separated as 3 different battle models that implement 3 difficulty levels and AI systems.

Member & Roles

Zhimu Li: Basic UI design, Village Scene, Dungeon Scene, All Triggers of GameObjects, Movement functions.

Jiabo Zhang: Battle Scenes, Battle Animation, Battle AI System, Save and Load function, Menu and End of the Game.

Main Components



Rigidbody 2D: this component can place gameobject under control of the unity physics engine. It means that it turns the asset into an entity.

Different types of collider 2D: this component can define the physical shape of gameobject, and determine how it makes interaction with other GameObjects which are also defined by collider 2D.

Script: this component can combine the codes and asset in game scene, we use C# in unity.

Cinemamachine: this component optimizes the basic camera function, it can make the game camera smoothly follow the main character.

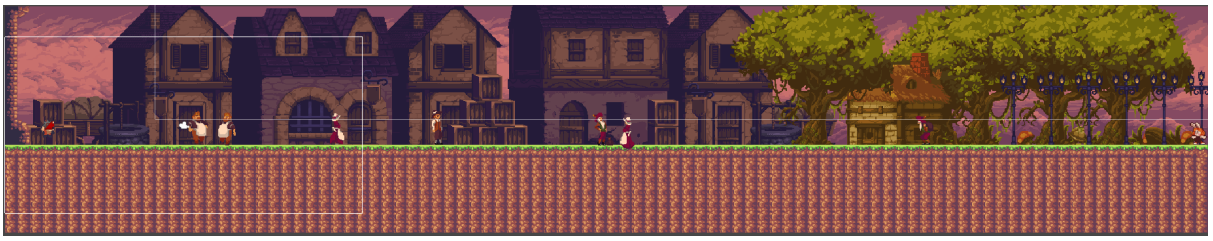
Main Asset Packages

1. Sunny Land Forest Assets
2. Sunnyland

3. Custom Character_Vol3
4. Aekashics Librarium - Megapack III
5. Gothicvania-Town
6. Free_Pack.unitystore

Main Scenes

Village scene: the village scene is the first scene after play starts playing the game, the character can make basic movements and interact with the NPCs. The objective of the village scene is to help players understand part of the story line and gain some information about the dungeon.



Dungeon scene: the whole dungeon scene is based on a script “RoomGenerator.cs”. This script can randomly generate a dungeon map with over 30 rooms. And every room contains a monster, or an item, or a healer. Players can gain gold by defeating monsters, and unlock items to strengthen himself or ask healer to heal up with gold. The goal of the game is to find the boss room and defeat the boss.

Battle scene: All battle scenes will be 2D scroll-rolling scenes (like KOF). There will be three levels of difficulty for monster battles, easy, medium and hard. The different difficulty levels of monsters are respectively loaded with different abilities (HP, ATK, and DEF), gold reward, status (phase 2) and AI system. If a player does not want to explore, he can choose to challenge the boss without any buff.

```
public class Boss_Run : StateMachineBehaviour
{

    public float speed = 2.5f;
    public float attackRange = 3f;

    Transform player;
    Rigidbody2D rb;
    Boss boss;

    // OnStateEnter is called when a transition starts and the state machine starts to evaluate this state
    override public void OnStateEnter(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
    {
        player = GameObject.FindGameObjectWithTag("Player").transform;
        rb = animator.GetComponent<Rigidbody2D>();
        boss = animator.GetComponent<Boss>();

    }

    // OnStateUpdate is called on each Update frame between OnStateEnter and OnStateExit callbacks
    override public void OnStateUpdate(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
    {
        boss.LookAtPlayer();

        Vector2 target = new Vector2(player.position.x, rb.position.y);
        Vector2 newPos = Vector2.MoveTowards(rb.position, target, speed * Time.fixedDeltaTime);
        rb.MovePosition(newPos);

        if (Vector2.Distance(player.position, rb.position) <= attackRange)
        {
            animator.SetTrigger("Attack");
        }
    }

    // OnStateExit is called when a transition ends and the state machine finishes evaluating this state
    override public void OnStateExit(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
    {
        animator.ResetTrigger("Attack");
    }
```

Problems and solutions

Problem #1: The frame of movement makes the animation look strange.

Solution: Using FixedUpdate function to replace Update function.

Problem #2: There is no header file in C#, and we need to define global variables and call them in other C# files.

Solution: Using Singleton for all variables. For example:

```
public static int Gold = 0;
   public static int globalGold
   {
     get { return Gold; }
      set { globalGold = value; }
   }
```

Then developers can use “ClassName.Gold” to call the “Gold” without worrying about repeated variable names.

Problem #3: Save and load problem. This is the most troublesome and tricky issue during the whole development process. Because we need to change scenes between dungeon scene and battle scenes when players trigger the battle, and we have to use the LoadScene and GetActiveScene functions of UnityEngine.SceneManagement, it means when every time players get back from battle scene the whole dungeon map will be reloaded and refreshed, and character will be back to the initial room of a new dungeon and monsters will be reborn.

```
public class MapsChange : MonoBehaviour
{
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.E))
        {
            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
        }
    }
}
```

Solution: Using DontDestroyOnLoad() function to save those changes that the player made.

For Example:

```
void Awake()
{   if (instance == null)
    {
        instance = this;
        Debug.Log("empty");
    }
    else if (instance != null)
    {
        Destroy(gameObject);
        //gameObject.SetActive(false);
        Debug.Log("not empty");
    }
 DontDestroyOnLoad(gameObject);}
```

However, because our dungeon map will be generated randomly, we have to create another empty GameObject to define a GameManager class and the whole dungeon map and every entity in the dungeon.

User experience & feedback

User #1 comment: The control setting and input management could be a bit confusing, and the game instruction and guide is not completed and clear enough. In most games, players used to use space key to jump and “a”, “s”, “d” and “w” to control direction movement. But in this dungeon game, the space key presents a magic attack and “w” is used to jump. It is kind of unnatural for many gamers.

Answer to User #1: Because when our group was creating the game, we used the default 2D game input management of unity, “a”, “s”, “d”, “w” and arrow keys all are used to change character’s position and transform. We used a similar movement function in both the dungeon scene and battle scene, that is the reason why we use “w” to change a character's y-axis and add rigidbody’s gravity setting to make the character jump in a side-scrolling scene. So in the village scene and dungeon scene, we will create canvas IU and battle tutorial scenes to clarify the guide of the game, and disable arrow keys and use the space key to replace “w” in the battle scene.

User #2 comment: Players might want more types of monsters and items. The rogue-like dungeon game needs to decrease the repetition rate of monster design and battle system. Sometimes, the sound effect also can make battles more fun and vibrant.

Answer to User #2: About sound effects, this is the easiest component ro edit, we did consider adding background music into the game, but copyright issue is a potential concern. We also tried to use the Recorder to record some sound effects for the game, but it did not work well so we deleted those components. As with other asset packages, because of the budget limitation, we only used some free asset packages from the Unity asset store and bought a couple cheap packages from other websites.

User #3 comment: Why allow players to pass the room without defeating the monster or finishing the event?

Answer to User #3: Balancing the difficulty of monsters is always the primary concern in the games. We do not expect all players to perfectly pass all events without mistakes, so the tolerance of mistakes is a necessary element. At the same time, we do not want players experience situation of dead game at the middle of game, because there is possibility of the dilemma like no money and no health at the same time and player have to defeat boss with dying status.

Future prospect

1. Redesigning the UI of the game. So far the UI of character's status (HP, MP, ATK, DEF, Maximum HP, Maximum MP and Gold amount) is very simple, and it is not well-done and conspicuous enough. Then there are two solutions for that. First one is replacing the text of the string with game maps or png assets. Second one is creating two more canvases to display character's status (Character page), and items and gold remains (Baggage page). At the ordinary game time those two canvases will be disabled, when players press a particular key, the canvas will be set active and one more time it will be turned off.
2. Replacing the asset packages. Just like we mentioned before, most asset packages we used in the game are free or cheap. For this reason, we need better or superior asset packages to make the dungeon and battle system diversified and satisfied. Because of the limitation of animators of our assets, the battle scene might make players feel humdrum. At the same time, we actually considered the suggestion from presentation day that creating characters and monsters by ourselves, but it is an impractical

solution for a two-man team. Because if we want to create one our own character, firstly we need a foundation of art, and secondly character is not a single png picture, it is made by at least ten or more pieces of body parts. Then we need to add a timeline into every part to complete the animation.

3. Add a difficulty selection at the beginning. So far, the play time of the game is about 5 to 10 minutes. We hope players can control play time of the game by selecting different models. Because of the “RoomGenerator” function, we can easily generate a dungeon and control the amount of rooms. In the easy model, players will only need to explore 1 or 2 floors with 5 to 10 minutes at maximum, and the maximum room amount will be less than 30. Then, the rest of difficulty models can be done for the same reason, in the most difficult level players need to pass 5 or 6 floors with over 100 rooms, and players need to prudently select items and healing timing with limitation of gold.

Conclusion

For the 400/477 capstone project, our team chose to build the game on unity. However, we do not have any former experience about unity engine and C# language. For this reason, our team had to learn the foundation of the Unity engine to make a fresh start. Although we experienced lots of conflicts and confusion during the whole development process, the Unity community is mature enough and effective communication between group members also solves many problems for designing game. According to our MVP, we made three huge changes at least during the whole process. So we got more chances to learn more about game development and C#. When the first time we figured out how to set the trigger and load the new scene, we felt that we are able to actually successfully develop our game. In

the future, we might try more development work on other different game engines which are more widely used, like Unreal Engine or RE engine. At the same time, 3D games also will be on our to-do list.

Reference

https://www.youtube.com/watch?v=AD4JIXQDw0s&list=PLRRnET3ZAhEzkrCdxn9Ik1xvwuwrFpPDp

https://assetstore.unity.com/packages/2d/characters/2d-simple-monster-orc-133919

https://assetstore.unity.com/packages/2d/characters/aekashics-librarium-megapack-iii-130410

https://assetstore.unity.com/packages/2d/characters/custom-character-2d-vol-3-65471

https://assetstore.unity.com/packages/2d/characters/free-animated-warrior-242585

https://assetstore.unity.com/packages/2d/environments/pixel-art-platformer-village-props-166114

https://assetstore.unity.com/packages/2d/characters/sunny-land-103349

https://assetstore.unity.com/packages/2d/characters/sunny-land-forest-108124

https://assetstore.unity.com/packages/2d/characters/gothicvania-town-101407

https://docs.unity3d.com/cn/2021.3/Manual/bakemultiplescenes.html