LOGBOOK 9

Team 1 : Hand Ball

Artist :

* Melanjutkan pekerjaan membuat elemen-elemen UI yang lain
* Hasil ada di github

Programmer :

GameManager.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine.UI;

using UnityEngine;

public class GameManager : MonoBehaviour {

//Declare Game Attribute

private static GameManager gameManager;

private float healthPoint;

private int score;

public bool gameOver = false;

private bool gameStart = false;

[HideInInspector]

public static List<GameObject> listEnemies = new List<GameObject>();

public Wave[] waves;

public int timeBetweenWaves = 5;

private float lastSpawnTime;

private int currentEnemiesSpawn = 0;

private int enemiesSpawned = 0;

private static int index = 0;

private static int indexPos = 0;

int currentWave;

//End of Declare Game Attribute

// Declare Game Object

public Player mPlayer;

// End of Declare Game Object

private void Awake()

{

gameManager = this.gameObject.GetComponent<GameManager>();

}

// Get Instance of Game Manager

public static GameManager GetInstanceOfGameManager()

{

return gameManager;

}

// Use this for initialization

void Start()

{

mPlayer = GameObject.Find("Player").GetComponent<Player>();

HealthPoint = 100;

Score = 0;

initSpawnEnemies();

waves[currentWave].updateMaxEnemies();

}

//Spawn Enemies

void initSpawnEnemies()

{

lastSpawnTime = Time.time;

currentWave = 0;

gameStart = true;

Debug.Log("Game Start: " + gameStart);

}

void spawnEnemies()

{

if (currentWave < waves.Length)

{

if (enemiesSpawned != waves[currentWave].maxEnemies)

{

if (index < waves[currentWave].enemyNums.Length)

{

if (enemiesSpawned == 0 || (enemiesSpawned < waves[currentWave].maxEnemies &&

currentEnemiesSpawn < waves[currentWave].enemyNums[index]))

{

if (indexPos > 3)

{

indexPos = 0;

}

//waves[currentWave].enemiesPrefab.GetComponent<EnemiesPrefabs>().setCurrentType(waves[currentWave].types[index]);

GameObject newEnemy = (GameObject)Instantiate(waves[currentWave].enemiesPrefab, waves[currentWave].position[indexPos], Quaternion.identity);

newEnemy.GetComponent<EnemiesPrefabs>().setCurrentType(waves[currentWave].types[index]);

listEnemies.Add(newEnemy);

enemiesSpawned++;

currentEnemiesSpawn++;

indexPos++;

}

else if (currentEnemiesSpawn == waves[currentWave].enemyNums[index])

{

currentEnemiesSpawn = 0;

index++;

}

}

}

else if (enemiesSpawned == waves[currentWave].maxEnemies && GameObject.FindGameObjectWithTag("Enemy") == null)

{

enemiesSpawned = 0;

currentEnemiesSpawn = 0;

index = 0;

indexPos = 0;

currentWave++;

if (currentWave < waves.Length)

waves[currentWave].updateMaxEnemies();

}

}else{

gameOver = true;

}

}

//End of Spawn Enemies

// Counter Attack Mechanic

public void CounterAttack\_Engage()

{

if (CounterAttack\_Treshold.collidedCT != null)

{

CounterAttack\_Treshold.collidedCT.GetComponent<BulletBehavior>().CounterAttack();

}

}

// End of Counter Attack Mechanic

// Update is called once per frame

void Update () {

if (gameStart)

{

if (!gameOver)

spawnEnemies();

if (HealthPoint <= 0)

gameOver = true;

}

}

//Setter-Getter

public float HealthPoint

{

get { return healthPoint; }

set { healthPoint = value; }

}

public int Score

{

get { return score; }

set { score = value; }

}

public int CurrentWave

{

get { return currentWave; }

}

//End of Setter-Getter

}

[System.Serializable]

public class Wave

{

[HideInInspector]

public Vector3[] position = new Vector3[3];

public GameObject enemiesPrefab;

public int[] types;

public int[] enemyNums;

public float interval;

[HideInInspector]

public int maxEnemies;

public Wave()

{

position[0] = new Vector3() {x = 0f, y = 1.5f, z = -2f};

position[1] = new Vector3() { x = -4.19f, y = 1.5f, z = -2f };

position[2] = new Vector3() { x = 4f, y = 1.5f, z = -2f };

}

public void updateMaxEnemies()

{

for (int i = 0; i < this.enemyNums.Length; i++)

{

maxEnemies += this.enemyNums[i];

}

}

}

Game Design :

* Game Stage Flow Details  
  File in : ../Game Design Document/Game Stage Design – Chapter 1
* Game Monetization Plan

File in : ../Game Design Document/Game Monetization Model

To do :

* Chapter 1 Stages progressive story.
* Game Monetization Details.
* Game Publishing Plan.