[Unity3D Engine을 이용한 3D게임 제작]

학과 : 컴퓨터공학부

학번 : 201958053

이름 : 최지연

1. **게임 설명**

* 제목 : Run Away
* 장르 : 아케이드 게임
* 시나리오 : 우리에 침입한 천적인 동물들을 피해 모이를 모두 먹어야 한다.
* 플레이 방법

1. 움직임 : 방향키를 이용하여 캐릭터의 방향을 조절한다.

2. 점프

Space bar : 높은 곳으로 점프해서 올라간다.

1. **게임월드 구성 및 주요 기능**

* player

1. 앞, 뒤, 좌, 우로 이동이 가능하다.
2. 점프를 통해 높은 곳으로 올라갈 수 있다.
3. 천적인 다른 동물들과 닿으면 피를 뿜으며 죽는다.
4. 먹이인 잡초를 먹을 수 있고, 먹을 때마다 score가 올라간다

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerController : MonoBehaviour

{

public float movementSpeed = 6;

public float jumpForce = 1000;

public float timeBeforeNextJump = 0.5f;

private float canJump = 0f;

public int maxHealth = 1;

public int health { get { return currentHealth; } }

int currentHealth;

Animator anim;

Rigidbody rb;

void Start()

{

anim = GetComponent<Animator>();

rb = GetComponent<Rigidbody>();

currentHealth = maxHealth;

}

void Update()

{

ControllPlayer();

}

public void ChangeHealth(int amount)

{

currentHealth = Mathf.Clamp(currentHealth + amount, 0, maxHealth);

if(currentHealth == 0)

{

GameController.instance.GameOver();

}

}

void ControllPlayer()

{

float moveHorizontal = Input.GetAxisRaw("Horizontal");

float moveVertical = Input.GetAxisRaw("Vertical");

Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);

if (movement != Vector3.zero)

{

transform.rotation = Quaternion.Slerp(transform.rotation, Quaternion.LookRotation(movement), 0.15f);

anim.SetInteger("Walk", 1);

}

else {

anim.SetInteger("Walk", 0);

}

transform.Translate(movement \* movementSpeed \* Time.deltaTime, Space.World);

if (Input.GetButtonDown("Jump") && Time.time > canJump)

{

rb.AddForce(0, jumpForce, 0);

canJump = Time.time + timeBeforeNextJump;

anim.SetTrigger("jump");

}

}

* NPC

1. 앞, 뒤, 좌, 우로 움직일 수 있다.
2. 플레이어의 움직임을 따라 움직인다.
3. 점프는 할 수 없다.
4. 고양이는 2, 개는 3, 사자는 4의 속도를 가지고 있다.
5. 플레이어인 닭과 닿으면 닭이 destroy 된다.

* //NPCMove.cs
* using System.Collections;
* using System.Collections.Generic;
* using UnityEngine;
* using UnityEngine.AI;
* public class NPCMove : MonoBehaviour
* {
* NavMeshAgent \_navMeshAgent; // 네비게이션 에이전트
* Transform objectToChase; // 추격할 위치
* public void Start()
* {
* \_navMeshAgent = this.GetComponent<NavMeshAgent>();
* objectToChase = GameObject.FindGameObjectWithTag("Player").transform; // 플레이어 위치로
* if (\_navMeshAgent == null)
* {
* Debug.LogError("Nav Mesh Agent component not found attached to " + gameObject.name);
* }
* else \_navMeshAgent.SetDestination(objectToChase.position); // 플레이어를 향해 이동
* }
* public void Update()
* {
* objectToChase = GameObject.FindGameObjectWithTag("Player").transform;
* \_navMeshAgent.SetDestination(objectToChase.position);
* }
* }
* //NPCCollision.cs
* using System.Collections;
* using System.Collections.Generic;
* using UnityEngine;
* public class NPCCollision : MonoBehaviour
* {
* public GameObject explosion;
* public int healthValue = 10;
* void OnCollisionEnter(Collision collision)
* {
* PlayerController player = collision.gameObject.GetComponent<PlayerController>();
* if (player != null)
* {
* Instantiate(explosion, collision.transform.position + Vector3.up, collision.transform.rotation);
* player.ChangeHealth(-1);
* Destroy(collision.gameObject);
* }
* else return; // Floor 혹은 다른 물체와의 충돌 무시
* }
* }
* grass

1. 1개 당 score 1점이다.
2. 총 50개까지 먹을 수 있다.
3. 50개를 먹을 시 player가 승리하며 게임이 끝난다.

* using System.Collections;
* using System.Collections.Generic;
* using UnityEngine;
* public class GrassCollectible : MonoBehaviour
* {
* private void OnTriggerEnter(Collider other)
* {
* PlayerController player = other.GetComponent<PlayerController>();
* if(player != null)
* {
* GameController.instance.score++;
* GameController.instance.UpdateScore();
* Destroy(gameObject);
* }
* }
* // Start is called before the first frame update
* void Start()
* {
* }
* // Update is called once per frame
* void Update()
* {
* }
* }
* camera

1. player를 따라다닌다.

* using System.Collections;
* using System.Collections.Generic;
* using UnityEngine;
* public class MainCamera\_Action : MonoBehaviour
* {
* GameObject player;
* public float offsetX = 0f;
* public float offsetY = 3f;
* public float offsetZ = -8f;
* Vector3 cameraPosition;
* public float followSpeed = 2f;
* // Start is called before the first frame update
* void Start()
* {
* player = GameObject.FindGameObjectWithTag("Player");
* }
* // Update is called once per frame
* void LateUpdate()
* {
* cameraPosition.x = player.transform.position.x + offsetX;
* cameraPosition.y = player.transform.position.y + offsetY;
* cameraPosition.z = player.transform.position.z + offsetZ;
* transform.position = Vector3.Lerp(transform.position, cameraPosition, followSpeed \* Time.deltaTime);
* }
* }
* Game control

1. Player의 체력이 0이 되면 game over 화면을 띄워 줌
2. 사용자가 Game over 화면을 클릭하면 다시 게임을 시작해 줌
3. Player가 모은 grass의 개수가 최대(50)개가 되면 you win 화면을 띄워 줌
4. 사용자가 You win 화면을 클릭하면 다시 게임을 시작해 줌

* using System.Collections;
* using System.Collections.Generic;
* using UnityEngine;
* using UnityEngine.UI;
* using UnityEngine.SceneManagement;
* public class GameController : MonoBehaviour
* {
* public static GameController instance;
* public GameObject gameOverText;
* public bool gameOver = false;
* public GameObject youWinText;
* public int score;
* public Text scoreText;
* private void Awake()
* {
  + if (instance == null) instance = this;
  + else if (instance != this) Destroy(gameObject);
* }
* // Start is called before the first frame update
* void Start()
* {
  + score = 0;
  + UpdateScore();
* }
* // Update is called once per frame
* void Update()
* {
  + if (gameOver && Input.GetMouseButtonDown(0))
  + {
    - SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
  + }
  + if (score == 50) PlayerWin();
* }
* public void UpdateScore()
* {
  + scoreText.text = "Score :" + score;
* }
* public void GameOver()
* {
  + gameOverText.SetActive(true); gameOver = true;
* }
* public void PlayerWin()
* {
  + youWinText.SetActive(true); gameOver = true;
* }
* }

1. **게임 실행 화면 캡처**

* 게임 시작 화면

**텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명**

* 점프한 화면

**장난감이(가) 표시된 사진

자동 생성된 설명**

* 다른 동물과 충돌해 죽은 화면 (피가 나옴)

****

* You win 화면
* 아직 저도 못깨봤습니다…

1. **게임 제작 후기**

2차원 게임에서도 그랬듯이, 게임의 스토리나 win/game over의 조건 등을 정하는 것이 가장 고민이었다. 2번째로 게임을 만들며 얻은 노하우는 asset store의 asset들을 구경하며 컨셉을 잡아 나가는 것이다. 2차원 게임 때 여우를 주인공으로 한 점이 마음에 들어서, 이번에도 동물들을 주제로 찾아 나가기 시작했고, 저번에는 공격을 할 수 있는 플레이어를 구상해 보았으니 이번에는 공격은 불가능하고 도망 다니며 미션을 수행하는 게임을 만들어 보고 싶었다. 그렇게 생각해 낸 게임이 술래잡기 형태의 Run Away이다. 고른 asset에는 닭, 개, 고양이, 사자가 있었는데, 이 중 닭만 날개가 있다는 점을 활용하여 닭을 플레이어로 설정했고, 단순히 도망가는 것 보다는 점프까지 가능한 것이 좋을 것 같아 플레이어에게만 점프 기능을 넣었다. 다음으로 가장 고민이었던 것은 카메라였다. 2차원 게임때와는 달리 3가지 축을 생각해야 하니 카메라를 설정하는 것에 시간을 많이 쓴 것 같다. 이 과정 중 화면 밖이 보이는 것을 3D 게임에서는 어떻게 해결해야 되는지에 대한 고민이 생겼는데, 이는 skybox라는 기능을 찾아 해결하게 되었다. 또한 게임 화면의 범위를 설정하는 문제도 고민이 되었는데, 동물들로 이루어진 게임이라서 울타리를 이용해 해결하게 되었다. 기말고사 기간이라 2D 게임만큼 시간을 많이 쓰지 못해서 더 많은 기능을 구현하지 못한 것이 아쉬웠지만, 플레이를 해보니 제작자인 나도 깨기 어려워서 뿌듯했다.