[Unity3D Engine을 이용한 3D게임 제작]

학과 : 컴퓨터공학부

학번 : 201958053

이름 : 최지연

1. **게임 설명**

* 제목 : Rocket Raccoon
* 시나리오 : 우주를 돌아다니던 주인공 Raccoon의 우주선에 Alien이 침입하여 공격을 받게 되고, 주인공은 우주선을 지키기 위해 남은 100발의 총알로 적을 소탕한다.
* 플레이 방법

1. 움직임 : 방향키(W,A,S,D)를 이용하여 움직일 방향을 결정한다.
2. 방향 : 마우스의 회전에 따라 캐릭터가 바라보는 방향을 조절한다.

3. 공격

마우스 왼쪽 키 : 연사 공격 (4발이 연속으로 발사)

마우스 오른쪽 키 : 단사 공격 (1발만 발사)

-> 최대 100발까지 발사할 수 있다.

-> 100발이 모두 소진되면, 공격 불가 상태로 game over 된다.

공격 방향 : 캐릭터가 바라보고 있는 방향으로 공격한다.

1. **게임월드 구성 및 주요 기능**

* player

1. 단사와 연사 공격이 가능하며 최대 100발 발사 가능함
2. 마우스 오른쪽 키 클릭 시 단사 공격을 하며 단사 모션을 취함
3. 마우스 왼쪽 키 클릭 시 연사 공격을 하며 연사 모션을 취함
4. 기본 체력은 5이며 적과 부딪히면 -1 (단, 한번 부딪히면 2초동안 무적상태가 됨)
5. 전후좌우 움직임이 가능 (W,A,S,D 버튼을 통해 해당 위치로 움직임)
6. 마우스 움직임으로 회전 가능 (회전을 통해 캐릭터 움직임 및 공격 방향 결정)
7. Alien과 부딪히면 공격을 당한 모션을 취함

/\*PlayerController.cs\*/

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerController : MonoBehaviour

{

public float movementSpeed = 10;

public float jumpForce = 10;

public float timeBeforeBehav = 0.5f;

public float canBehav = 0f;

public int maxHealth = 10;

public int fireInput = 0;

public int health { get { return currentHealth; } }

int currentHealth;

Animator animator;

Rigidbody rgbody;

public Camera mainCamera;

public GameObject player;

public GameObject bulletPrefab;

public float timeInvincible = 2.0f;

bool isInvincible; //true=무적

float invincibleTimer; //남은 무적 시간

void Fire()

{

if(Input.GetMouseButtonDown(0))

{

fireInput = 0;

Vector3 firePos = transform.position + animator.transform.forward + new Vector3(0f, 0.5f, 0f);

Vector3 target = new Vector3(transform.position.x, transform.position.y, transform.position.z);

for(int i=0; i<4; i++)

{

target.x += i;

target.y += i;

target.z += i;

var bullet = Instantiate(bulletPrefab, target, Quaternion.identity).GetComponent<Bullet>();

bullet.Fire(animator.transform.forward);

}

ChangeBullet(fireInput);

}

else if(Input.GetMouseButtonDown(1))

{

fireInput = 1;

Vector3 firePos = transform.position + animator.transform.forward + new Vector3(0f, 0.5f, 0f);

Vector3 target = new Vector3(transform.position.x, transform.position.y, transform.position.z);

var bullet = Instantiate(bulletPrefab, target, Quaternion.identity).GetComponent<Bullet>();

bullet.Fire(animator.transform.forward);

ChangeBullet(fireInput);

}

}

// Start is called before the first frame update

void Start()

{

animator = GetComponent<Animator>();

rgbody = GetComponent<Rigidbody>();

currentHealth = maxHealth;

}

// Update is called once per frame

void Update()

{

controllerPlayer();

LookMouseCursor();

Fire();

if (isInvincible)

{

invincibleTimer -= Time.deltaTime;

if (invincibleTimer < 0) //남은 무적 시간이 없으면

{

isInvincible = false; //무적 해지

}

}

}

public void ChangeHealth(int amount)

{

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

if (currentHealth == 0)

{

animator.SetInteger("GunShot",1);

//Destroy(this.gameObject);

GameController.instance.GameOver();

}

else if (amount < 0)

{

animator.SetTrigger("Dive");

if (isInvincible) return; //무적이면 리턴

isInvincible = true;

invincibleTimer = timeInvincible;

}

}

public void ChangeBullet(int input)

{

if(GameController.instance.leftBullet<=0)

{

GameController.instance.GameOver();

}

if(input==0)

{

GameController.instance.leftBullet -= 4;

GameController.instance.UpdateBullet();

}

else if(input==1)

{

GameController.instance.leftBullet--;

GameController.instance.UpdateBullet();

}

}

void controllerPlayer()

{

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

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

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

if (Input.GetKeyDown(KeyCode.Space) && Time.time > canBehav)

{

rgbody.AddForce(0, jumpForce, 0);

canBehav = Time.time + timeBeforeBehav;

animator.Play("Jump");

}

if (Input.GetMouseButtonDown(0))

{

animator.Play("FourShot");

}

if (Input.GetMouseButtonDown(1))

{

animator.Play("SingleShot");

}

if (movement != Vector3.zero)

{

animator.Play("Run");

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

}

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

}

public void LookMouseCursor()

{

Ray ray = mainCamera.ScreenPointToRay(Input.mousePosition);

RaycastHit hitResult;

if(Physics.Raycast(ray, out hitResult))

{

Vector3 mouseDir = new Vector3(hitResult.point.x, transform.position.y, hitResult.point.z) - transform.position;

animator.transform.forward = mouseDir;

}

}

private void OnCollisionEnter(Collision collision)

{

NPCCollision npc = collision.gameObject.GetComponent<NPCCollision>();

if(npc!=null)

{

animator.Play("Dive");

}

}

}

* Alien

1. Player 위치를 파악하고 해당 위치로 이동함
2. player와 충돌하면 player의 체력이 -1됨
3. Player의 공격인 총알 공격을 받으면 죽으며 사라짐
4. Nav Mesh에 따라 이동이 제한됨
5. 게임 시작 시 Spawn Manager 스크립트로 미리 지정된 위치에 랜덤 생성됨

/\*NPCMove.cs\*/

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.AI;

public class NPCMove : MonoBehaviour

{

GameObject target;

NavMeshAgent navAgent;

public void Start()

{

target = GameObject.Find("Player");

navAgent = GetComponent<NavMeshAgent>();

}

public void Update()

{

Vector3 navPos = new Vector3(target.transform.position.x, transform.position.y, target.transform.position.z);

if(navAgent.destination != target.transform.position)

{

navAgent.SetDestination(navPos);

}

else

{

navAgent.SetDestination(transform.position);

}

}

}

/\*NPCCollision.cs\*/

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class NPCCollision : MonoBehaviour

{

void OnCollisionEnter(Collision collision)

{

PlayerController player = collision.gameObject.GetComponent<PlayerController>();

if (player != null)

{

player.ChangeHealth(-1);

Debug.Log(player.health);

}

}

}

/\*SpawnManager.cs\*/

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class SpawnManager : MonoBehaviour

{

public GameObject[] prefabs;

private BoxCollider area;

public int count = 1;

private List<GameObject> gameObject1 = new List<GameObject>();

void Start()

{

area = GetComponent<BoxCollider>();

for (int i = 0; i < count; ++i)//count 수 만큼 생성한다

{

Spawn();//생성 + 스폰위치를 포함하는 함수

}

area.enabled = false;

}

private Vector3 GetRandomPosition()

{

Vector3 basePosition = transform.position;

Vector3 size = area.size;

float posX = basePosition.x + Random.Range(-size.x / 2f, size.x / 2f);

float posY = basePosition.y + Random.Range(-size.y / 2f, size.y / 2f);

float posZ = basePosition.z + Random.Range(-size.z / 2f, size.z / 2f);

Vector3 spawnPos = new Vector3(posX, posY, posZ);

return spawnPos;

}

private void Spawn()

{

int selection = Random.Range(0, prefabs.Length);

GameObject selectedPrefab = prefabs[selection];

Vector3 spawnPos = GetRandomPosition();//랜덤위치함수

GameObject instance = Instantiate(selectedPrefab, spawnPos, Quaternion.identity);

gameObject1.Add(instance);

}

}

* Bullet

1. player 공격 시 발사되는 bullet
2. alien과 부딪히면 alien과 함께 사라짐
3. 캐릭터와 마우스의 방향을 통해 발사 방향이 결정 됨
4. 10초간 alien과 부딪히지 않으면 해당 오브젝트가 사라짐 (용량을 줄이기 위함)

/\*Bullet.cs\*/

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Bullet : MonoBehaviour

{

bool isFire;

Vector3 direction;

[SerializeField]

float speed = 10f;

// Start is called before the first frame update

void Start()

{

Destroy(gameObject, 10f);

}

// Update is called once per frame

void Update()

{

if(isFire)

{

transform.Translate(direction \* Time.deltaTime \* speed);

}

}

public void Fire(Vector3 dir)

{

direction = dir;

isFire = true;

}

private void OnTriggerEnter(Collider other)

{

NPCMove npc = other.GetComponent<NPCMove>();

if(npc!=null)

{

Destroy(gameObject);

Destroy(other.gameObject);

GameController.instance.leftAlien --;

Debug.Log(GameController.instance.leftAlien);

GameController.instance.UpdateScore();

}

}

}

* Camera

1. Player의 전후좌우 움직임을 따라다님
2. 마우스의 좌우 움직임에 따라 회전을 하며 캐릭터의 좌, 우를 보여줌

/\*MainCamera\_Action.cs\*/

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;

// Use this for initialization

void Start()

{

player = GameObject.FindGameObjectWithTag("Player");

}

// Update is called once per frame

void Update()

{

moveObjectFunc();

}

// 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);

}

private float speed\_rota = 2.0f;

void moveObjectFunc()

{

//float mouseX = Input.GetAxis("Mouse X") % 180;

float mouseY = Input.GetAxis("Mouse Y") % 180;

//transform.Rotate(Vector3.up \* speed\_rota \* mouseX);

transform.Rotate(Vector3.left \* speed\_rota \* mouseY);

}

}

* Game control

1. Player의 체력이 0이 되면 game over 화면을 띄워 줌
2. Player의 남은 총알이 0이 되면 game over 화면을 띄워 줌
3. 사용자가 Game over 화면을 클릭하면 다시 게임을 시작해 줌
4. Player가 최대 총알 내에서 모든 alien을 처치하면, you win 화면을 띄워 줌
5. 사용자가 You win 화면을 클릭하면 다시 게임을 시작해 줌

/\*GameController.cs\*/

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 totalAlien = 36;

public float leftAlien = 36;

public Text scoreText;

public Text bulletText;

public int leftBullet = 100;

public int totalBullet = 100;

private void Awake()

{

if (instance == null) instance = this;

else if (instance != this) Destroy(gameObject);

}

// Start is called before the first frame update

void Start()

{

leftAlien = totalAlien;

leftBullet = totalBullet;

UpdateScore();

UpdateBullet();

}

// Update is called once per frame

void Update()

{

if (gameOver && Input.GetMouseButtonDown(0))

{

SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);

}

if (leftAlien <= 0) PlayerWin();

}

public void UpdateScore()

{

scoreText.text = "left alien :" + (int)leftAlien + " / " + totalAlien;

}

public void UpdateBullet()

{

bulletText.text = "left bullet :" + leftBullet + " / " + totalBullet;

}

public void GameOver()

{

gameOverText.SetActive(true); gameOver = true;

}

public void PlayerWin()

{

youWinText.SetActive(true); gameOver = true;

}

}