**TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI**

**Khoa Công nghệ Thông tin**

—o0o—



**BÁO CÁO ĐỒ ÁN**

**TẠO GAME 3d: BẮN ZOMBIE**

**Lớp 63IT3**

**Giảng viên hướng dẫn: Nguyễn Văn Hải**

**Sinh viên thực hiện:**

**NGUYỄN ANH BẰNG**

**Hà Nội, 2022**

**MỤC LỤC**

[**CHƯƠNG I: MÔ TẢ THIẾT KẾ GAME** 4](#_Toc97972846)

[I. Tổng quan: 4](#_Toc97972847)

[II. Nền tảng thiết kế: 4](#_Toc97972848)

[III. Thể loại: 4](#_Toc97972849)

[IV. Tiến trình của game: 4](#_Toc97972850)

[V. Hoạt động và cảm nhận: 4](#_Toc97972851)

[**Chương II GAME PLAY VÀ CÁC CHỨC NĂNG CƠ BẢN** 5](#_Toc97972852)

[I. Game play: 5](#_Toc97972853)

[II. Các chức năng cơ bản: 6](#_Toc97972854)

[**Chương III. ĐỊA HÌNH BẢN ĐỒ GAME** 9](#_Toc97972855)

[I. Địa hình và bản đồ Game 9](#_Toc97972856)

[II. tài nguyên: 10](#_Toc97972857)

[1. Nhà, trường học, bệnh viện, ATM 10](#_Toc97972858)

[2. Đèn giao thông: 10](#_Toc97972859)

[3. Đường đi, vạch kẻ đường, sân chơi: 10](#_Toc97972860)

[4. Súng: 11](#_Toc97972861)

[5. Zombie: 11](#_Toc97972862)

[**Chương IV. Code C#:** 11](#_Toc97972863)

[1. Điều khiển lệnh bàn phím để nhân vật góc nhìn thứ nhất di chuyện theo phím: 11](#_Toc97972864)

[2. Góc nhìn xoay theo con trỏ chuột tạo cảm giác chân thật: 12](#_Toc97972865)

[3. Súng xoay theo con trỏ chuột: 12](#_Toc97972866)

[4. Spam zombie ra liên tục: 13](#_Toc97972867)

[5. Zombie đuổi theo người chơi: 14](#_Toc97972868)

[6. Xử lý bắn đạn: 15](#_Toc97972869)

[7. Xử lí âm thanh game khi hạ Zombie: 18](#_Toc97972870)

[8. Xử lí menu: 20](#_Toc97972871)

[**Chương V. Âm thanh** 21](#_Toc97972872)

[**1.** **Âm thanh:** 21](#_Toc97972873)

[**Chương VI. Link Demo game trên YTB:** 22](#_Toc97972874)

[1. Link: https://www.youtube.com/watch?v=EyW1yBgcCAs 22](#_Toc97972875)

# 

# **CHƯƠNG I: MÔ TẢ THIẾT KẾ GAME**

1. **Tổng quan:**

Game bắn súng zombie là một game bắn súng góc nhìn thứ nhất, nhiệm vụ của người chơi là cố gắng sống sót lâu nhất có thể và đạt được mạng hạ gục zombie cao nhất có thể. Ngoài mục tiêu ganh đua điểm cao với những người chơi khác, người chơi còn có thể kết thúc trận đấu bằng cách hạ Boss chính.

1. **Nền tảng thiết kế:**

-phần mềm: Unity 2020.3.19f1

-nền tảng: windows

-ngôn ngữ lập trình: C#

1. **Thể loại:**

-Game bắn súng góc nhìn thứ nhất tạo cảm giác chân thật nhất.

-Zombie: thây ma với hiệu ứng di chuyển cào người chơi và tiếng game dồn dập kinh dị.

1. **Tiến trình của game:**

Người chơi sẽ phải trải qua những thử thách đặc biệt về tính kiến nhẫn cũng như đưa ra chiến thuật hợp lý ở trong trận đấu.

Trên tay một khẩu súng, người chơi phải “tiệu diệt” zombie để ghi điểm và tăng “độ cảm hứng”, sau khi tiêu diệt hết zombie thì sẽ xuất hiện zombie chúa (Boss).

Để có thể win, người chơi phải đạt được những chỉ tiêu mức định và thời gian nào đó Boss chính xuất hiện với máu và độ sát thương cao hơn.

Trò chơi kết thúc khi người chơi thoát game hoặc người chơi bị zombie đánh bại, hoặc đánh bại các zombie và Boss chính.

1. **Hoạt động và cảm nhận:**

* Ánh sáng trời sầm kèm với cơn ác mộng nhẹ nhàng ghê rợn.
* Màn chơi là khung cảnh thân thuộc như trường học, bệnh viện, đường phố, ngân hàng…
* Hiệu ứng vật lý ở mức trung bình, kẻ địch với hình dáng thận thuộc, kinh dị.
* Khung cảnh huyền ảo nhưng tối tăm, dáng vẻ đáng sợ.
* Nhạc nền dồn dập tạo cảm giác cho người chơi.
* Nhạc hạ zombie, tiếng súng kêu sẽ tạo cảm giác cho người chơi.
* Laze để người chơi biết súng đang ngắm ở đâu.

**Chương II GAME PLAY VÀ CÁC CHỨC NĂNG CƠ BẢN**

**I. Game play:**

-mục tiêu chính: Sống sót và hạ các zombie và Boss cuối.

-Cấu trúc dự án:

Graphical user interface

Description automatically generated with low confidence

-player:

A screenshot of a computer

Description automatically generated with medium confidence

+di chuyển và chạy nhảy để tránh Zombie và Boss cuối.

+sự dụng vũ khí súng để tiêu diệt zombile và Boss cuối.

* Boss cuối:

Graphical user interface, application, website

Description automatically generated

-zombie tải trên Asset store https://assetstore.unity.com/packages/3d/characters/humanoids/zombie-30232

-súng tải trên Asset store

-nhà, địa hình tại trên Asset store https://assetstore.unity.com/packages/3d/environments/urban/city-package-107224

1. **Các chức năng cơ bản:**

- Di chuyển: Người chơi sử dụng bàn phím để di chuyển với tổ hợp phím “W, A, S, D” (tương ứng với tiến xuống, sáng phải, lên và sang trái). Thanh Space để giúp người chơi nhảy, leo trèo trên các chướng ngại vật.

- Tấn công: Mouse 1 (chuột trái), di chuyển chuột hiện laze màu đỏ để người chơi dễ dàng ngắm trúng mục tiêu cần hạ:

A screenshot of a video game

Description automatically generated

- Thanh máu: Máu là một giá trị quan trọng với người chơi, luôn luôn phải giữ cho máu >=0 nếu không thì trò chơi sẽ kết thúc (Game Over). Nó sẽ được hiển thị ở góc trái phía bên dưới của màn hình và màu đỏ, nếu bị zombie cào trúng thì nó sẽ giảm theo lượng cào của zombie.

A screenshot of a video game

Description automatically generated

**-** Điểm hiển thị ở góc trên của màn hình, nơi người chơi nhận biết được mình đạt được bao nhiêu điểm từ tiêu diệt kẻ địch:

A screenshot of a video game

Description automatically generated

- Khi người chơi hết máu thì trò chơi sẽ kết thúc (game Over) và sẽ hiển thị có chơi lại hay không:

A screenshot of a video game

Description automatically generated

**Chương III. ĐỊA HÌNH BẢN ĐỒ GAME**

1. Địa hình và bản đồ Game

A screenshot of a computer

Description automatically generated with medium confidence

**-**Địa hình game được thiết kế là một thành phố được chia làm 4 khu, mỗi khu có 1 sân vui chơi, bệnh viện, ngân hàng, trường học,... như ngoài đời làm cho người chơi cảm thấy thân quen giống đời thực hơn.

## II. tài nguyên:

1. Nhà, trường học, bệnh viện, ATM

A screenshot of a video game

Description automatically generated with medium confidence

1. Đèn giao thông:

Graphical user interface, text

Description automatically generated

1. Đường đi, vạch kẻ đường, sân chơi:

Graphical user interface

Description automatically generated

1. Súng:

Graphical user interface, application

Description automatically generated

1. Zombie:

Graphical user interface, application, website

Description automatically generated

**Chương IV. Code C#:**

1. **Điều khiển lệnh bàn phím để nhân vật góc nhìn thứ nhất di chuyện theo phím:**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Move : MonoBehaviour

{

void Update()

{

float xDirection = Input.GetAxis("Horizontal");

float zDirection = Input.GetAxis("Vertical");

Vector3 moveDirection = new Vector3(xDirection, 0.0f, zDirection);

transform.position += moveDirection;

}

}

1. **Góc nhìn xoay theo con trỏ chuột tạo cảm giác chân thật:**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class MouseLookAround : MonoBehaviour

{

float rotationX = 0f;

float rotationY = 0f;

public float sensitivity = 1f;

// Update is called once per frame

void Update()

{

rotationX += Input.GetAxis("Mouse X") \* sensitivity;

rotationY += Input.GetAxis("Mouse Y") \* -1 \* sensitivity;

transform.localEulerAngles = new Vector3(rotationX, rotationY, 0);

}

}

1. **Súng xoay theo con trỏ chuột:**

using UnityEngine;

using System.Collections;

public class RotateGun : MonoBehaviour {

public Vector3 target;

// Use this for initialization

void Start () {

}

// Update is called once per frame

void Update () {

LookAtCursor();

}

void LookAtCursor()

{

Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);

RaycastHit hit;

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

{

target = hit.point;

}

transform.LookAt(target);

}

}

1. **Spam zombie ra liên tục:**

using UnityEngine;

using System.Collections;

public class SpawnEnemy : MonoBehaviour {

GameObject[] spawnPoint;

public GameObject zombie;

public PlayerPrefs z;

public float minSpawnTime = 0.2f;

public float maxSpawnTime = 1;

private float lastSpawnTime = 0;

private float spawnTime = 0;

// Use this for initialization

void Start () {

spawnPoint = GameObject.FindGameObjectsWithTag("Respawn");

UpdateSpawnTime();

}

void UpdateSpawnTime()

{

lastSpawnTime = Time.time;

spawnTime = Random.Range(minSpawnTime, maxSpawnTime);

}

void Spawn()

{

int point = Random.Range(0, spawnPoint.Length);

Instantiate(zombie, spawnPoint[point].transform.position, Quaternion.identity);

UpdateSpawnTime();

}

// Update is called once per frame

void Update () {

if (Time.time >= lastSpawnTime + spawnTime)

{

Spawn();

}

}

}

1. **Zombie đuổi theo người chơi:**

using UnityEngine;

using System.Collections;

public class MoveToPlayer : MonoBehaviour {

float moveSpeed;

public float minMoveSpeed = 0.05f;

public float maxMoveSpeed = 0.3f;

GameObject player;

public float attackRange = 1;

// Use this for initialization

void Start () {

player = GameObject.FindGameObjectWithTag("Player");

UpdateMoveSpeed();

}

void UpdateMoveSpeed()

{

moveSpeed = Random.Range(minMoveSpeed, maxMoveSpeed);

}

void Move()

{

if (player == null)

return;

if (Vector3.Distance(transform.position, player.transform.position) > attackRange)

{

transform.position = Vector3.Lerp(transform.position, player.transform.position, moveSpeed \* Time.deltaTime);

}

else

{

gameObject.GetComponent<Animator>().SetBool("isIdle", true);

gameObject.GetComponent<ZombieController>().isAttack = true;

gameObject.GetComponent<MoveToPlayer>().enabled = false;

}

}

// Update is called once per frame

void Update () {

Move();

}

}

1. **Xử lý bắn đạn:**

using UnityEngine;

using System.Collections;

using UnityEngine.UI;

public class PlayerController : MonoBehaviour {

public int damge = 1;

public float fireTime = 0.3f;

public GameObject smoke;

public GameObject gunHead;

public float playerHealth = 10;

public AudioClip playerDeathSound;

public Slider healthBar;

private float playerCurrentHealth = 10;

private float lastFireTime = 0;

private Animator anim;

private AudioSource audioS;

private GameObject gameController;

// Use this for initialization

void Start () {

anim = gameObject.GetComponent<Animator>();

UpdateFireTime();

audioS = gameObject.GetComponent<AudioSource>();

gameController = GameObject.FindGameObjectWithTag("GameController");

healthBar.maxValue = playerHealth;

healthBar.value = playerCurrentHealth;

healthBar.minValue = 0;

}

void UpdateFireTime()

{

lastFireTime = Time.time;

}

void SetFireAnim(bool isFire)

{

anim.SetBool("isShoot", isFire);

}

public void GetHit(float damge)

{

audioS.Play();

playerCurrentHealth -= damge;

healthBar.value = playerCurrentHealth;

if (playerCurrentHealth <= 0)

{

Dead();

}

}

void Dead()

{

audioS.clip = playerDeathSound;

audioS.Play();

gameController.GetComponent<GameController>().EndGame();

}

void Fire()

{

if (Time.time >= lastFireTime + fireTime)

{

Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);

#if UNITY\_IOS || UNITY\_ANDROID

ray = Camera.main.ScreenPointToRay(Input.mousePosition);

RaycastHit hit;

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

{

if (hit.transform.tag.Equals("Zombie"))

{

SetFireAnim(true);

InsSmoke();

hit.transform.gameObject.GetComponent<ZombieController>().GetHit(damge);

}

}

#else

RaycastHit hit;

if (Physics.Raycast(gunHead.transform.position, gunHead.transform.forward, out hit))

{

if (hit.transform.tag.Equals("Zombie"))

{

SetFireAnim(true);

InsSmoke();

hit.transform.gameObject.GetComponent<ZombieController>().GetHit(damge);

}

}

#endif

UpdateFireTime();

}

else

{

SetFireAnim(false);

}

}

void InsSmoke()

{

GameObject sm = Instantiate(smoke, gunHead.transform.position, gunHead.transform.rotation) as GameObject;

Destroy(sm, 0.5f);

}

// Update is called once per frame

void Update () {

if (Input.GetMouseButton(0))

{

Fire();

}

}

}

1. **Xử lí âm thanh game khi hạ Zombie:**

using UnityEngine;

using System.Collections;

public class ZombieController : MonoBehaviour {

public int zombieHealth = 3;

public float shootTime = 0.5f;

public bool isAttack = false;

public float attackTime = 1;

public AudioClip zombieDeathSound;

public float damge = 1;

public bool IsShooten

{

get { return isShooten; }

set

{

isShooten = value;

ShootenAnim(isShooten);

UpdateShootenTime();

}

}

private Animator anim;

private bool isShooten;

private bool isDead = false;

private float lastAttackTime = 0;

private AudioSource audioS;

private float lastShootenTime = 0;

private GameObject player;

private GameObject gameController;

// Use this for initialization

void Start () {

anim = gameObject.GetComponent<Animator>();

IsShooten = false;

anim.SetBool("isDead", false);

audioS = gameObject.GetComponent<AudioSource>();

player = GameObject.FindGameObjectWithTag("Player");

gameController = GameObject.FindGameObjectWithTag("GameController");

}

void UpdateShootenTime()

{

lastShootenTime = Time.time;

}

void UpdateAttackTime()

{

lastAttackTime = Time.time;

}

void ShootenAnim(bool isShooten)

{

anim.SetBool("isShooten", isShooten);

}

void AttackAnim(bool isAttack)

{

anim.SetBool("isAttack", isAttack);

}

public void GetHit(int damge)

{

if (isDead)

return;

audioS.Play();

IsShooten = true;

zombieHealth -= damge;

if (zombieHealth <= 0)

{

Dead();

}

}

void Dead()

{

isDead = true;

audioS.clip = zombieDeathSound;

audioS.Play();

anim.SetBool("isDead", true);

gameController.GetComponent<GameController>().GetPoint(1);

Destroy(gameObject, 2f);

}

void Attack()

{

if (Time.time >= lastAttackTime + attackTime)

{

AttackAnim(true);

UpdateAttackTime();

player.GetComponent<PlayerController>().GetHit(damge);

}

else

{

AttackAnim(false);

}

}

// Update is called once per frame

void Update () {

if (IsShooten && Time.time >= lastShootenTime + shootTime)

{

IsShooten = false;

}

if (isAttack)

{

Attack();

}

}

}

1. **Xử lí menu:**

using UnityEngine;

using System.Collections;

using UnityEngine.SceneManagement;

using UnityEngine.UI;

public class GameController : MonoBehaviour {

public GameObject menu;

public Text txtPoint;

private int currentPoint = 0;

// Use this for initialization

void Start () {

Time.timeScale = 1;

menu.SetActive(false);

}

// Update is called once per frame

void Update () {

}

public void GetPoint(int point)

{

currentPoint++;

txtPoint.text = "Zombie killed: " + currentPoint.ToString();

}

public void RestartGame()

{

SceneManager.LoadScene(0);

}

public void EndGame()

{

menu.SetActive(true);

Time.timeScale = 0;

}

}

**Chương V. Âm thanh**

1. Âm thanh:

**Graphical user interface

Description automatically generated**

**Chương VI. Link Demo game trên YTB:**

1. Link: https://www.youtube.com/watch?v=EyW1yBgcCAs