khung3khung4khung1

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙟🕮🙝**

**BÀI TẬP LỚN MÔN HỌC LẬP TRÌNH GAME**

**TRÒ CHƠI BẮN XE TĂNG**

Giảng viên hướng dẫn: Nguyễn Việt Hưng

Sinh viên thực hiện:

Vũ Sỹ Ngọc Duy

Lương Doãn Thủy

Nguyễn Trung Cường

Lớp: CNTT1 K59

khung2Hà Nội**, tháng** 6 **năm** 2021

# **Phân tích và thiết kế trò chơi**

## **Kịch bản**

Nhân vật chính của trò chơi là chiếc xe tăng được điều khiển bởi người chơi.

Người chơi phải nhắm bắn vào các xe tăng kẻ thù.

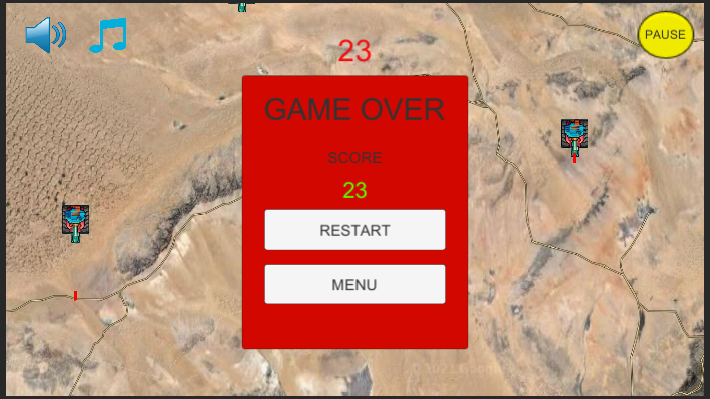


Với mỗi lần bắn trúng thì sẽ được điểm và điểm càng cao thì độ khó của trò chơi càng cao, các xe tăng kẻ thù sẽ di chuyển nhanh hơn, bắn đạn nhiều hơn và số lượng cũng nhiều hơn.



Trong quá trình đó xe tăng của người chơi va chạm với các xe tăng khác, hay là bị xe tăng khác bắn laze trúng thì trò chơi sẽ kết thúc.

Điểm mà người chơi đạt được sẽ hiện lên và đây cũng là điểm thú vị của trò chơi. Người chơi sẽ được thử sự khéo léo nhanh nhẹn và tinh mắt thì với những người chơi khác xem ai được điêm cao hơn.



Các tính năng khác: trong quá trình chơi game, người chơi có thể tắt mở tiếng, đổi nhạc nền và tạm dừng trò chơi.



* Hướng dẫn chơi:

Xe tăng người chơi sẽ được điều khiển bằn các phím mũi tên và bắn ra đạn laze bằng phím spacebar.

## **Phân tích và phần mềm hỗ trợ**

### **Phần mềm**

* Phần mềm xây dựng và thiết kế trò chơi: Unity 3D
* Phần mềm hỗ trợ: Visual Studio 2013, Photoshop CS6

### **Phân tích trò chơi**

Trò chơi sẽ xây dựng với các đối tượng như sau: Xe tăng người chơi, xe tăng kẻ thù, viên đạn từ xe tăng người chơi và xe tăng kẻ thù.

Mỗi đối tượng có các thuộc tính:

* Xe tăng người chơi: di chuyển qua lại lên xuống khắp màn hình.
* Xe tăng kẻ thù: được sinh ra một cách ngẫu nhiên, ở những vị trí ngẫu nhiên phía trên màn hình di chuyển xuốn phía dưới màn hình với tốc độ khác nhau tùy vào số điểm của người chơi.
* Viên đạn xe tăng người chơi: sẽ được bắn ra từ xe tăng người chơi với tốc độ bay cố định và số lần bắn ra từ xe tăng sẽ được điều khiển bởi người chơi qua phím spacebar.
* Viên đạn xe tăng kẻ thù: được bắn ra từ xe tăng kẻ thù với số lần và tốc độ tùy thuộc vào điềm của người chơi.
* Âm thanh: gồm có các tiếng bắn đạn của xe tăng, xe tăng nổ, nhạc nền màn hình menu, nhạc nền man hình chơi trò chơi.



(Hiệu ứng nổ của xe tăng địch)

## **Thiết kế**

### **Xe tăng người chơi**



**Xử lý bắn đạn cho xe tăng**

IEnumerator Shoot()

{

canShoot = false;

// de dan ban ra tu dau nong xe tang

Vector3 temp = transform.position;

temp.y += 0.6f;

Instantiate(bullet, temp, Quaternion.identity);

AudioSource.PlayClipAtPoint(weaponClip, temp); // them am thanh khi ban dan

yield return new WaitForSeconds(0.2f); //0,2 giay ban ra vien dan 1 lan

canShoot = true;

}  
**Xử lý di chuyển**

void FixedUpdate() // di chuyen 1 doi tuong co vat ly

{

PlayerMovement();

}

void PlayerMovement()

{

float xAxis = Input.GetAxisRaw("Horizontal") \* playerSpeed;

float yAxis = Input.GetAxisRaw("Vertical") \* playerSpeed;

myBody.velocity = new Vector2(xAxis, yAxis);

}

**Xử lý va chạm với các vật khác**

void OnTriggerEnter2D(Collider2D target)

{

if (target.tag == "Enemy" || target.tag == "RedBullet")

{

Destroy(gameObject);

Destroy(target.gameObject);

//hieu ung no

explosionPlayer = (GameObject)Instantiate(explosionPlayer, target.transform.position, Quaternion.identity);

Destroy(explosionPlayer, 1);

if (GamePlayController.instance != null)

{

GamePlayController.instance.PlayerDiedShowPanel();

}

//am thanh no

AudioSource.PlayClipAtPoint(explosionPlayerClip, transform.position);

}

}

### **Xe tăng kẻ thù**



**Xử lý tốc độ bay theo điểm**

void Start()

{

if (GamePlayController.instance.playerScore < 10)

{

enemySpeed = Random.Range(2, 4);

};

if (10 <= GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 20)

{

enemySpeed = Random.Range(3, 5);

};

if (20 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 30)

{

enemySpeed = Random.Range(4, 6);

};

if (30 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 40)

{

enemySpeed = Random.Range(5, 7);

};

if (40 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 50)

{

enemySpeed = Random.Range(6, 8);

};

if (50 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 60)

{

enemySpeed = Random.Range(7, 9);

};

if (60 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 70)

{

enemySpeed = Random.Range(8, 10);

};

if (70 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 80)

{

enemySpeed = Random.Range(9, 11);

};

if (80 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 90)

{

enemySpeed = Random.Range(10, 12);

};

if (90 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 100)

{

enemySpeed = Random.Range(11, 13);

};

if (GamePlayController.instance.playerScore > 100)

{

enemySpeed = Random.Range(12, 14);

};

StartCoroutine(EnemyShoot());

}

Xử lý tốc độ bắn theo điểm

IEnumerator EnemyShoot()

{

if (GamePlayController.instance.playerScore <= 30)

{

yield return new WaitForSeconds(Random.Range(0.7f, 1f));

};

if (30 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 60)

{

yield return new WaitForSeconds(Random.Range(0.4f, 0.7f));

};

if (GamePlayController.instance.playerScore > 60)

{

yield return new WaitForSeconds(Random.Range(0.2f, 0.5f));

};

Vector3 temp = transform.position;

if (GamePlayController.instance.playerScore <= 20)

{

temp.y -= 0.4f;

};

if (20 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 40)

{

temp.y -= 0.45f;

};

if (40 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 60)

{

temp.y -= 0.5f;

};

if (60 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 80)

{

temp.y -= 0.55f;

};

if (GamePlayController.instance.playerScore > 80)

{

temp.y -= 0.6f;

};

Instantiate(bullet, temp, Quaternion.identity);

AudioSource.PlayClipAtPoint(weaponEnemyClip, temp);

StartCoroutine(EnemyShoot());

}

### **Viên đạn xe tăng người chơi**

D:\Downloads\Space_MayBay\Assets\Sprites\Yellow Bullet.png

**Xử lý va chạm khi bắn trúng các vật thể**

void OnTriggerEnter2D(Collider2D target) //xu ly va cham voi trigger

{

if (target.tag == "Enemy")

{

Destroy(target.gameObject);

Destroy(gameObject);

explosionEnemy = (GameObject)Instantiate(explosionEnemy, target.transform.position, Quaternion.identity);

Destroy(explosionEnemy, 1);

GamePlayController.instance.playerScore++; //cong diem

AudioSource.PlayClipAtPoint(explosionEnemyClip, target.transform.position);

}

if (target.tag == "Border")

{

Destroy(gameObject);

}

}

### **Viên đạn Xe tăng địch**

D:\Downloads\Space_MayBay\Assets\Sprites\Red Bullet.png

**Xử lý tốc độ đạn bay theo Xe tăng**

void Start()

{

if (GamePlayController.instance.playerScore <= 10)

{

speed = 5;

};

if (10 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 20)

{

speed = 6;

};

if (20 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 30)

{

speed = 7;

};

if (30 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 40)

{

speed = 8;

};

if (40 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 50)

{

speed = 9;

};

if (50 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 60)

{

speed = 10;

};

if (60 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 70)

{

speed = 11;

}

if (70 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 80)

{

speed = 12;

}

if (80 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 90)

{

speed = 13;

}

if (90 < GamePlayController.instance.playerScore && GamePlayController.instance.playerScore <= 100)

{

speed = 14;

}

if (GamePlayController.instance.playerScore > 100)

{

speed = 16;

}

}

**Xử lý va chạm khi trúng Xe tăng người chơi**

void FixedUpdate()

{

myBody.velocity = new Vector2(0f, -speed);

}

void OnTriggerEnter2D(Collider2D target) //xu ly va cham voi trigger

{

if (target.tag == "Border")

{

Destroy(gameObject);

}

}

### **Hình nền**



* Hình map game (sa mạc)



### **Xử lý âm thanh**

**Xứ lý tắt mở nhạc**

void Start() {

btnNhac.GetComponent<Button>();

audioSource = gameObject.GetComponent<AudioSource>();

audioSource.Play();

Unmute();

}

**Xử lý chọn nhạc nền**

public void ChangeButtonAudio(){

if (counter == 3)

counter = 0;

else

counter++;

switch (counter)

{

case 0:

btnNhac.image.overrideSprite = MoNhac;

audioSource.clip = NhacNen3;

audioSource.Play();

break;

case 1:

btnNhac.image.overrideSprite = MoNhac;

audioSource.clip = NhacNen2;

audioSource.Play();

break;

case 2:

btnNhac.image.overrideSprite = MoNhac;

audioSource.clip = NhacNen1;

audioSource.Play();

break;

case 3:

btnNhac.image.overrideSprite = TatNhac;

audioSource.Stop();

break;

}

}

//Tắt Nhạc

public void Mute(){

btnTatNhac.SetActive(true);

//audioSource.Stop();

AudioListener.volume = 0f;

btnMoNhac.SetActive(false);

// Time.timeScale = 0;

}

//Mở Nhạc

public void Unmute()

{

btnTatNhac.SetActive(false);

audioSource.clip = NhacNen3;

AudioListener.volume = 1f;

audioSource.Play();

btnMoNhac.SetActive(true);

}

//Mở List nhac

public void ListNhac()

{

PanelListNhac.SetActive(true);

btnListNhac.SetActive(false);

}

//Mở nhạc nền 1

public void MoNhacNen1()

{

PanelListNhac.SetActive(false);

btnListNhac.SetActive(true);

audioSource.clip = NhacNen1;

audioSource.Play();

}

//Mở nhạc nền 3

public void MoNhacNen2()

{

PanelListNhac.SetActive(false);

btnListNhac.SetActive(true);

audioSource.clip = NhacNen2;

audioSource.Play();

}

//Mở nhạc nền 3

public void MoNhacNen3()

{

PanelListNhac.SetActive(false);

btnListNhac.SetActive(true);

audioSource.clip = NhacNen3;

audioSource.Play();

}

## **Mục tiêu**

* Giải trí, rèn luyện khả năng lập trình.
* Tìm hiểu thêm phần mềm mới và mảng mới là viết trò chơi.
* Có thể áp dụng được mô hình vừa học vào viết trò chơi thực tế.
* Hoàn thiện các kỹ năng xây dựng phần mềm

# **Thuận lợi và khó khăn**

## **Thuận lợi**

* Phần mềm Unity 3D hỗ trợ 80% trong thiết kế trò chơi
* Có nền tảng lập trình hướng đối tượng
* Tài liệu tham khảo dễ hiểu, dễ áp dụng
* Thành viên nhóm làm việc tích cực

## **Khó khăn**

* Công cụ viết game hoàn toàn mới
* Phát sinh lỗi trò chơi khi xây dựng và thiết kế

# **Tài liệu tham khảo**

* Khóa học lập trình Unity 3D cua Khoa Phạm
* Wikipedia – lập trình Unity

# **Phân công**

* Vũ Sỹ Ngọc Duy: xử lý xe tăng
* Lương Doãn Thủy: xử lý phần đạn và chướng ngại vật
* Nguyễn Trung Cường: xử lý hình ảnh và âm thanh