

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO GIỮA KÌ MÔN PHÁT TRIỂN TRÒ CHƠI

HÀNH TRÌNH SHURIMA

Người hướng dẫn: **Gv Hà Lê Hoài Trung**

Người thực hiện: **Lê Thành Đăng Khoa - 51900119**

Nguyễn Tiến Phát - 51900167

Lớp: **19050202**

Khoá: **K23**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO GIỮA KÌ MÔN PHÁT TRIỂN TRÒ CHƠI

HÀNH TRÌNH SHURIMA

Người hướng dẫn: **Gv Hà Lê Hoài Trung**

Người thực hiện: **Lê Thành Đăng Khoa - 51900119**

Nguyễn Tiến Phát - 51900167

Lớp: **19050202**

Khoá: **23**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

LỜI CẢM ƠN

Chúng em cảm ơn thầy Hà Lê Hoài Trung đã cung cấp những kiến thức trong các buổi học vừa qua của môn Phát triển trò chơi, nhờ đó mà chúng em có thể hoàn thành được bài báo cáo này. Đồng thời em cũng cảm ơn những sự giúp đỡ và hướng dẫn tận tình của thầy trong quá trình chúng em làm bài để giúp chúng em có thể hoàn thành bài báo cáo này một cách tốt nhất.

TP. Hồ Chí Minh, ngày 5 tháng 12 năm 2021

Tác giả 1

(Ký tên và ghi rõ họ tên)

Lê Thành Đăng Khoa

Tác giả 2

(Ký tên và ghi rõ họ tên)

Nguyễn Tiến Phát

BÁO CÁO ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là công trình nghiên cứu của chúng tôi và được sự hướng dẫn khoa học của thầy Hà Lê Hoài Trung. Các nội dung nghiên cứu, kết quả trong báo cáo là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong báo cáo còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung bài báo cáo của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 5 tháng 12 năm 2021

Tác giả 1

(Ký tên và ghi rõ họ tên)

Tác giả 2

(Ký tên và ghi rõ họ tên)

Lê Thành Đăng Khoa

Nguyễn Tiến Phát

PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN

Tp. Hồ Chí Minh, ngày tháng năm
(Ký tên và ghi rõ họ tên)

MỤC LỤC

LỜI CẢM ƠN	2
BÁO CÁO ĐƯỢC HOÀN THÀNH	3
PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN.....	4
MỤC LỤC	5
DANH MỤC HÌNH.....	7
NỘI DUNG.....	8
Chương 1 - Một số khái niệm cơ bản về Unity	8
1.1 GameObject.....	8
1.2 Component	8
1.3 Scenes	8
1.4 Scripts	9
1.5 Prefabs	9
1.6 Sprite	9
1.7 Camera.....	10
1.8 Light	10
1.9 Animation.....	10
1.10 Sounds	11
1.11 Light, Rigidbody, Collider trong Component	11
1.11.1 Light Component	11
1.11.2 Rigidbody Component	12
1.11.3 Collider Component.....	12
Chương 2 - Giới thiệu game	15
2.1 Cốt truyện	15
2.2 Cơ chế.....	16
Chương 3 - Animation.....	17
3.1 Phương pháp tạo animation.....	17

3.2	Các trạng thái cơ bản của nhân vật.....	17
3.3	Chi tiết trạng thái nhân vật	17
3.3.1	Azir.....	17
3.3.2	Hecarim.....	20
Chương 4 - Sơ đồ trong game		21
4.1	Sơ đồ chơi game	21
4.2	Sơ đồ Animation.....	22
4.2.1	Azir.....	22
4.2.2	Hecarim.....	22
Chương 5 - Scripts.....		23
5.1	Azircotrol: Điều khiển nhân vật xử lý va chạm và âm thanh.....	23
5.2	Aziratk: Hàm gọi và triệu hồi đòn tấn công của nhân vật.....	27
5.3	Skill: Gọi về kỹ năng và phá hủy kỹ năng	28
5.4	Cameracotrol: Điều hướng máy quay theo nhân vật và cố định máy quay	29
5.5	Trieuhoi: Dùng để sinh ra mod trong màn 2	30
5.6	Modcotrol: Xử lý va chạm và âm thanh của mod	31
5.7	Menugame: Dùng để tương tác với Menu Game	34
5.8	Gameplaycotrol: Màn hình khi dừng game.....	35
5.9	Backtomenu: Hiện kết thúc game và quay lại màn hình chính.....	36
5.10	Nextlevel: Chuyển sang màn chơi tiếp theo.....	37
5.11	Wingame: Dùng để kết thúc game và chuyển qua màn hình kết thúc	38
TÀI LIỆU THAM KHẢO		39

DANH MỤC HÌNH

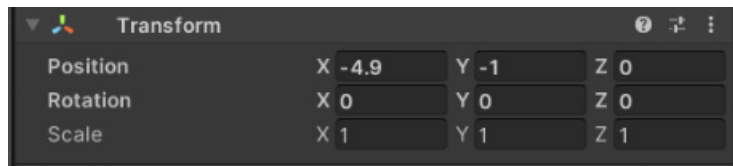
Hình 1 Transform trong Unity.....	8
Hình 2 Scenes trong game Hành trình Shurima	9
Hình 3 Sprite Hecarim trong game Hành trình Shurima.....	10
Hình 4 Animation Azir trong game Hành trình Shurima	11
Hình 5 Rigidbody trong Unity.....	12
Hình 6 Azir attack.....	18
Hình 7 Azir death	18
Hình 8 Azir idle	19
Hình 9 Azir run.....	19
Hình 10 Azir jump.....	20
Hình 11 Hecarim death.....	20
Hình 12 Hecarim run	21
Hình 13 Sơ đồ chơi trong game.....	21
Hình 14 Sơ đồ animation của Azir	22
Hình 15 Sơ đồ animation của Hecarim	22
Hình 16 Nhân vật trong trò chơi.....	23
Hình 17 Màn hình khi thua game	27
Hình 18 Kỹ năng của nhân vật	28
Hình 19 Khu vực sinh quái.....	30
Hình 20 Hình ảnh mod trong game	32
Hình 21 Menu game	34
Hình 22 Màn hình dừng game	35
Hình 23 Màn hình kết thúc game	36
Hình 24 Khu vực chuyển màn chơi.....	37
Hình 25 Thính chén chạm để kết thúc game	38

NỘI DUNG

Chương 1 - Một số khái niệm cơ bản về Unity

1.1 GameObject

Game Object là một đối tượng thể trong game, nó có thể là nhân vật, cây cỏ, công cụ, camera, hiệu ứng, ... Các Game Object chứa thuộc tính Transform, được dùng để thiết lập vị trí (Position), phương hướng (Rotation) và kích thước (Scale) cho Game Object.



Hình 1 Transform trong Unity

1.2 Component

Component được gán các Game Object là những thành phần có chức năng riêng biệt. Một Game Object gồm có ba thành phần chính: Light Component, Rigidbody Component, Collider Component. Các thành phần này sẽ tạo nên một Game Object hoàn chỉnh theo yêu cầu của game.

1.3 Scenes

Scenes là một cảnh, không gian. Nó thiết lập bố cục các Game Object và chứa các thiết lập giao diện trong game. Ngoài ra, Scenes quản lý tất cả các đối tượng trong một màn chơi của game.



Hình 2 Scenes trong game Hành trình Shurima

1.4 Scripts

Đây cũng là một Component trong Unity dùng để tương tác các hành động với người chơi. Scripts chứa các mã nguồn như dùng C#, Java Script, BOO để khởi tạo và xử lý các đối tượng trong Unity.

1.5 Prefabs

Prefabs là một Game Object, được dùng để tái sử dụng các đối tượng tương tự nhau trong game mà ta chỉ cần thiết lập lại vị trí, phương hướng và kích thước từ một đối tượng đầu tiên. Các GameObject được sao chép từ Prefabs hoàn toàn giống nhau.

1.6 Sprite

Là hình ảnh 2D của Game Object. Sprite có thể là hình ảnh hay một bộ phận nào đó của nhân vật.



Hình 3 Sprite Hecarim trong game Hành trình Shurima

1.7 Camera

Camera cũng là một Game Object, dùng để xác định tầm nhìn và quan sát các đối tượng trong game.

1.8 Light

Light dùng để xác định màu sắc và độ sáng của mọi vật thể, việc kết hợp nhiều light sẽ giúp game có được hiệu ứng ánh sáng phù hợp và bắt mắt.

1.9 Animation

Là một yếu tố không thể thiếu trong game. Animation giúp game diễn tả được hành động của từng nhân vật trên sự thay đổi liên tục của các hình ảnh khác nhau.



Hình 4 Animation Azir trong game Hành trình Shurima

1.10 Sounds

Sounds là những âm thanh được sử dụng trong game.

1.11 Light, RigidBody, Collider trong Component

1.11.1 Light Component

1.11.1.1 Directional Light Component

Là nguồn ánh sáng được phát ra từ nguồn xa vô tận, các tia sáng được phát ra song song, được hiểu như ánh sáng của mặt trời.

1.11.1.2 Point Light Component

Point Light Component hoạt động như một bóng đèn, phát ra từ một điểm đến mọi hướng, vật thể càng xa Point Light Component thì càng ít nhận được ánh sáng mà nó chiếu đến.

1.11.1.3 Sky Light Component

Được hiểu là thứ nằm ngoài tầm mắt của con người và biểu thị nó dưới dạng ánh sáng (bầu trời) và ảnh phản xạ sẽ hoàn toàn khớp với chính nó.

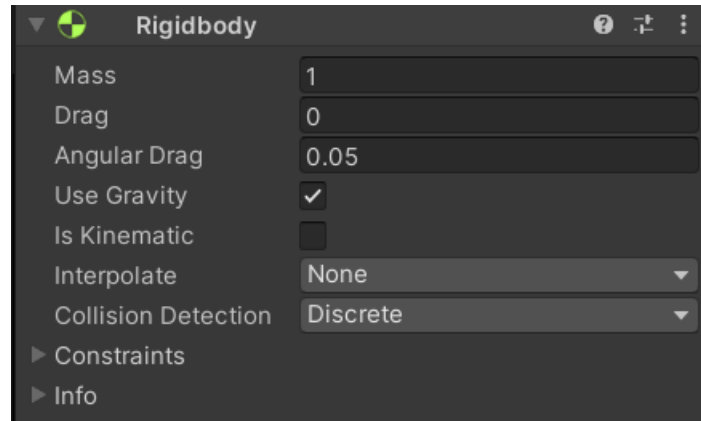
1.11.1.4 Spot Light Component

Spot Light Component phát ra ánh sáng từ một điểm dưới dạng hình nón.

1.11.2 Rigidbody Component

1.11.2.1 Khái niệm:

Rigidbody là thành phần có tác dụng điều khiển vị trí của vật thể, không cần sử dụng Script nhưng nó vẫn có thể di chuyển và xử lý hướng và chạm khi có Collider.



Hình 5 Rigidbody trong Unity

1.11.2.2 Các thành phần của Rigidbody:

- Mass : Trọng lượng
- Drag : Lực cản không khí
- Angular Drag : Lực cản không khí khi xoay
- Use Gravity : Áp dụng trọng lực cho vật thể
- Is Kinematic : Dừng việc áp dụng trọng lực cho vật thể ngay lập tức
- Freeze Position : Đóng băng Tọa độ
- Freeze Rotation : Đóng băng Góc xoay

1.11.3 Collider Component

Collider là thành phần được sử dụng để nhận biết sự va chạm. Phần lớn các Collider có hình dạng đơn giản để dễ tính toán dễ dàng hơn.

1.11.3.1 Colliders được Unity cung cấp những API sau để nhận biết va chạm:

- OnCollisionEnter được gọi khi GameObject này bắt đầu chạm vào GameObject khác.
- OnCollisionStay được gọi khi GameObject này đang chạm vào GameObject khác.
- OnCollisionExit được gọi khi GameObject này đã ngừng chạm vào GameObject khác.
- Contacts: điểm va chạm giữa 2 vật.
- GameObject: GameObject va chạm với Object gốc.
- RelativeVelocity: vận tốc tương đối.

1.11.3.2 Collider với thuộc tính As Trigger

Để sử dụng Trigger trong Collider, ta chỉ cần tick vào As Trigger trong Component Collider. Khi đó Collider sẽ không bị va chạm bởi bất cứ Object nào và được sử dụng để phát hiện những va chạm trên nó tạo ra những sự kiện sau:

- OnTriggerEnter được gọi khi GameObject này bắt đầu chạm vào GameObject khác.
- OnTriggerStay được gọi khi GameObject này đang chạm vào GameObject khác.
- OnTriggerExit được gọi khi GameObject này đã ngừng chạm vào GameObject khác.

1.11.3.3 Ví dụ về Scripts cho các loại va chạm:

❖ Kiểm tra va chạm của nhân vật nhân vật:

```
private void OnCollisionEnter2D(Collision2D coll2D)
{
    if (coll2D.gameObject.CompareTag("Mod"))
    {
```

```

        Chet();
    }
    else if (coll2D.gameObject.CompareTag("Nendat"))
    {
        Vitri = true;
    }
}
void Chet()
{
    if (ado && Adie)
    {
        ado.PlayOneShot(Adie);
    }
    if (Chuyendong)
        Chuyendong.SetTrigger("Death");

    if (r2d)
    {
        r2d.velocity = new Vector2(0f, r2d.velocity.y);
    }
    Losepanel.SetActive(true);
    Time.timeScale = 0f;

}

```

❖ Xử lý va chạm khi kẻ thù:

```

private void OnCollisionEnter2D(Collision2D col2)
{
    if (col2.gameObject.CompareTag("New"))
    {
        Chet();
    }
    else if (col2.contacts[0].normal.x > 0)
    {

```

```

        Quaydau();
        Dichuyentrai = false;

    }
    else
    {

        Quaydau();
        Dichuyentrai = true;
    }
}
void Quaydau()
{
    Dichuyentrai = !Dichuyentrai;
    transform.Rotate(0f, 180f, 0f);
}
void Chet()
{
    Dachet = true;
    ChuyendongMod.SetTrigger("Die");
    Destroy(gameObject, 0.5f);
}
}

```

Chương 2 - Giới thiệu game

2.1 Cốt truyện

Đã từ lâu, Shurima là một vùng đất phì nhiêu màu mỡ, nằm ở phía Nam Valoran. Khi đó vị thần Azir đã hợp nhất các dân tộc lại làm một, thành lập binh đoàn bất bại và ông đã dẫn quân hùng hậu của mình đi chinh phục khắp nơi trên thế giới. Với nguồn năng lượng đến từ trái tim Mặt trời, ông đã thành công tiêu diệt Urgot – thủ lĩnh vùng sa mạc Florida. Vì một phút mềm Azir đã tha chết cho con trai Urgot, nhận con hấn – Sion làm đệ tử duy nhất và coi Sion như con ruột của mình. Cũng chính sự nhân hậu đầy đã khiến ông phải trả giá đắt sau này. Đến lúc trưởng thành, Sion đã học được tất cả các nhẫn thuật lẫn thể thuật từ Azir, hấn nhân lúc sơ hở để âm độc chiếm trái tim Mặt trời từ tay Azir, giết chết ông và

biến vùng đất này thành vùng hoang mạc như lãnh thổ Florida đã từng. Lúc ấy, linh hồn Azir đi qua dòng sông Noxus và gặp vị thần cai quản Thăng Hoa Đoàn (thần bảo hộ vùng đất Shurima), nhờ trái tim nhân hậu và quăng đại mà ngài được hồi sinh tại đây. Nhiệm vụ của Azir là quay lại vùng đất Shurima, chiến đấu với Sion cùng những tên thuộc hạ của hắn, giành lại trái tim Mặt trời và khôi phục lại đế chế Shurima như thời hoàng kim. Liệu Azir có làm được điều này!?

2.2 Cơ chế

Ở các màn chơi, sẽ có những quái thú khác nhau, ngăn cản ông ấy tìm đến chén thánh. Người chơi sẽ điều khiển Azir chiến đấu với lũ quái thú, né được các cạm bẫy và tìm được cánh cửa để có thể qua màn. Nếu tìm lại được chén thánh, bạn sẽ giành được chiến thắng. Ngược lại, nếu trong quá trình chơi game, người chơi để quái thú đụng trúng người hoặc vô tình dẫm phải bẫy, người chơi sẽ thua và phải chơi lại từ đầu.

Cơ chế di chuyển:

Sử dụng các phím A,D để di chuyển xung quanh.

Sử dụng phím W để nhảy.

Sử dụng phím Space tấn công.

Chương 3 - Animation

3.1 Phương pháp tạo animation

B₁: Tìm kiếm hoạt ảnh (Gif, video, ...) thích hợp từng hoạt động, nhân vật trong game.

B₂: Chia video thành nhiều ảnh nhỏ hay còn gọi là FPS (frame per second), ít nhất 10 ảnh/giây. Với mật độ ảnh như vậy nhân vật sẽ di chuyển mượt mà hơn.

B₃: Sử dụng các phần mềm để loại bỏ phần sau của từng khung ảnh (Transparent).

3.2 Các trạng thái cơ bản của nhân vật

- Attack : trạng thái nhân vật khi tấn công.
- Death : trạng thái nhân vật khi bị hạ gục.
- Idle : trạng thái nhân vật khi đang rảnh.
- Run : trạng thái nhân vật khi chạy.
- Jump : trạng thái nhân vật khi nhảy.

3.3 Chi tiết trạng thái nhân vật

- Azir: attack, death, idle, run, Jump.
- Hecarim: death, run.

3.3.1 Azir

3.3.1.1 Attack



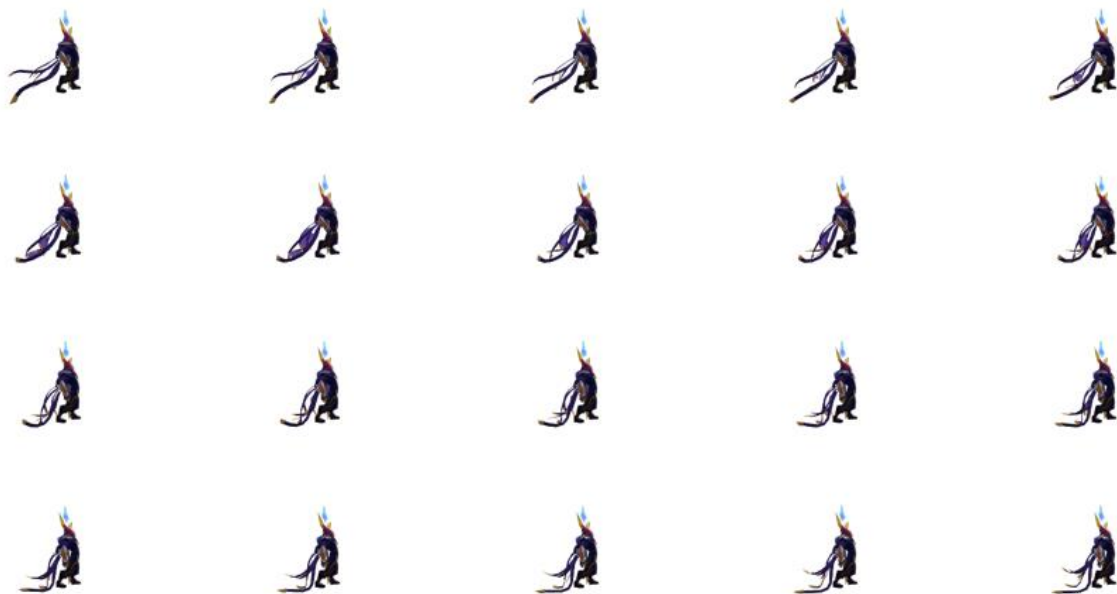
Hình 6 Azir attack

3.3.1.2 Death



Hình 7 Azir death

3.3.1.3 Idle



Hình 8 Azir idle

3.3.1.4 Run



Hình 9 Azir run

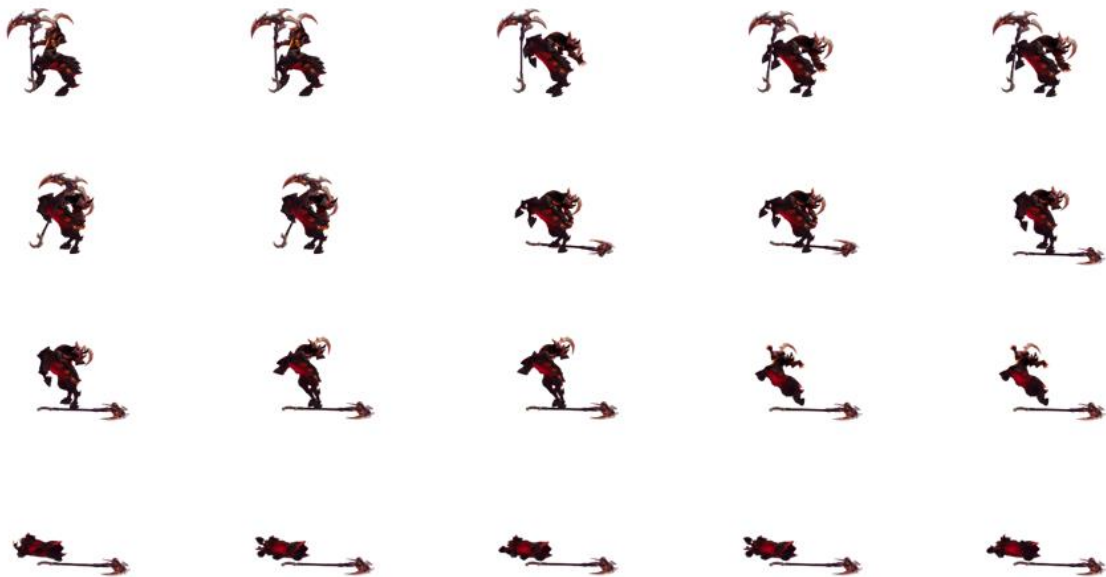
3.3.1.5 Jump



Hình 10 Azir jump

3.3.2 *Hecarim*

3.3.2.1 Death



Hình 11 Hecarim death

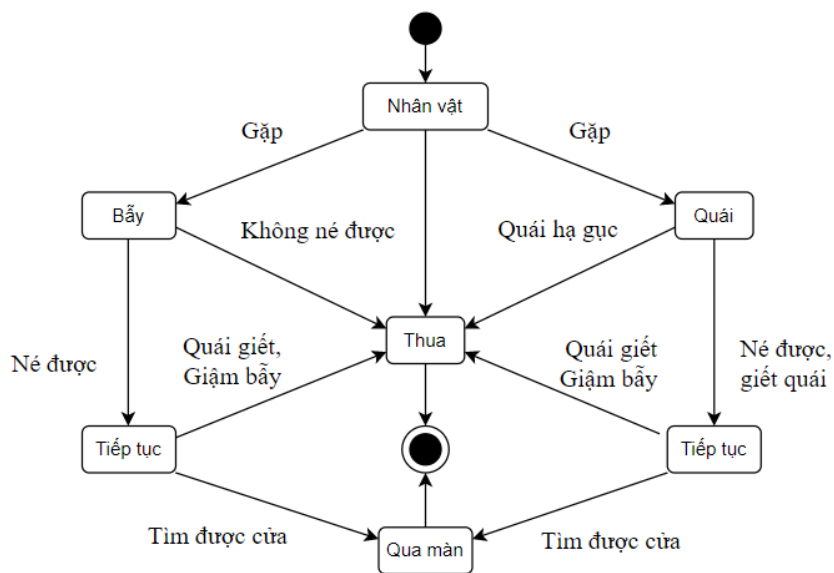
3.3.2.2 Run



Hình 12 Hecarim run

Chương 4 - Sơ đồ trong game

4.1 Sơ đồ chơi game

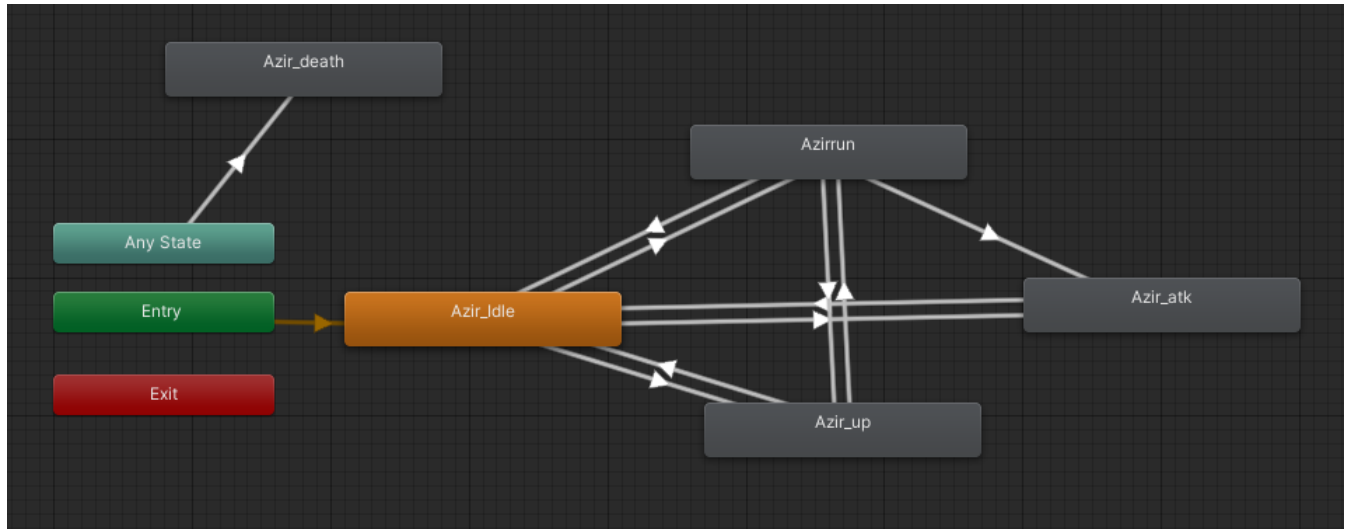


Hình 13 Sơ đồ chơi trong game

Cách chơi: sử dụng các nút A, D để di chuyển qua lại W để nhảy và Space để tấn công
 Nhiệm vụ người chơi là tránh các chương ngại vật và thu về chén thánh.

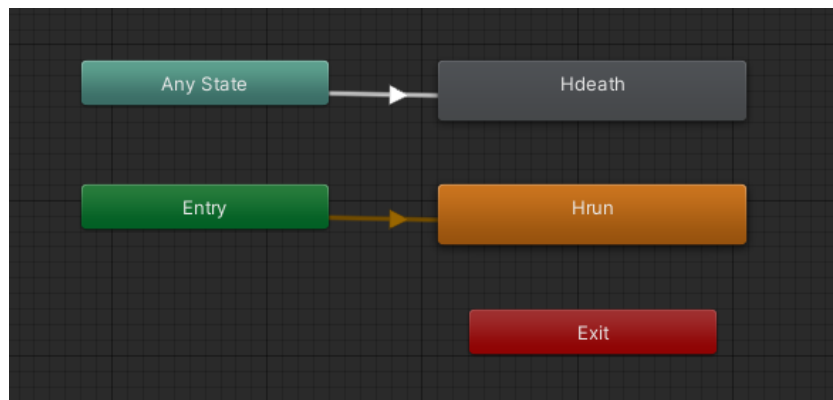
4.2 Sơ đồ Animation

4.2.1 Azir



Hình 14 Sơ đồ animation của Azir

4.2.2 Hecarim



Hình 15 Sơ đồ animation của Hecarim

Chương 5 - Scripts

5.1 Azircotrol: Điều khiển nhân vật xử lý va chạm và âm thanh



Hình 16 Nhân vật trong trò chơi

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class Azircotroler : MonoBehaviour
{
    public float Vantoc;
    private float Tocdo = 0f; // toc do nhan vat
    private bool Vitri = false; // vi tri dung
    private bool Huongphai = true; // doi huong nhan vat
    private bool Tancong = false;
    public float Azirnhay;
    public float Trongluc;
    private bool Isdeath;

    private bool canatk = true;
    [SerializeField]
    private GameObject Losepanel;

    public AudioSource ado;
    public AudioClip Adie;
```



```

private Animator Chuyendong;
private Rigidbody2D r2d;
// Start is called before the first frame update
void Start()
{
    r2d = GetComponent<Rigidbody2D>();
    Chuyendong = GetComponent<Animator>();
    Time.timeScale = 1f;
}

// Update is called once per frame
void Update()
{
    Chuyendong.SetFloat("Tocdo", Tocdo);
    Chuyendong.SetBool("Vitri", Vitri);
    Chuyendong.SetBool("Tancong", Tancong);
    Nhay();
    Danh();
    if (gameObject.transform.position.y < -5f)
    {
        Chet();
    }
}

private void FixedUpdate()
{
    Dichuyen();
}
void Dichuyen()
{
    float traiphai = Input.GetAxis("Horizontal");
    r2d.velocity = new Vector2(Vantoc * traiphai, r2d.velocity.y);
    Tocdo = Mathf.Abs(Vantoc * traiphai);
    if (traiphai > 0 && !Huongphai)
    {
        huongdichuyen();
    }
    if (traiphai < 0 && Huongphai)

```

```

    {
        huongdichuyen();
    }
}
void huongdichuyen()
{
    Huongphai = !Huongphai;
    transform.Rotate(0f, 180f, 0f);

}
void Nhay()
{
    if (Input.GetKeyDown(KeyCode.W) && Vitri == true)
    {

        r2d.AddForce((Vector2.up) * Azirnhay);
        Vitri = false;

    }
    else if (r2d.velocity.y < 0)
    {
        Vitri = true;
        r2d.velocity += Vector2.up * Physics2D.gravity.y * (Trongluc - 1) *
Time.deltaTime;

    }

}

void Danh();//hàm để gọi animation
{
    if (Input.GetKey(KeyCode.Space))
    {
        if (canatk)
        {
            StartCoroutine(cddanh());
        }
    }
}
IEnumerator cddanh()

```

```

{
    Tancong = true;
    canatk = false;

    yield return new WaitForSeconds(0.5f);

    canatk = true;
    Tancong = false;
}

private void OnCollisionEnter2D(Collision2D coll2D)
{
    if (coll2D.gameObject.CompareTag("Mod"))
    {
        Chet();
    }
    else if (coll2D.gameObject.CompareTag("Nendat"))
    {
        Vitri = true;
    }
}

void Chet()
{
    if (ado && Adie)
    {
        ado.PlayOneShot(Adie);
    }
    if (Chuyendong)
        Chuyendong.SetTrigger("Death");

    if (r2d)
    {
        r2d.velocity = new Vector2(0f, r2d.velocity.y);
    }
    Losepanel.SetActive(true);
    Time.timeScale = 0f;
}

```

```

public void Quitgame();//hàm gọi menu khi chết
{
    SceneManager.LoadScene(0);
}
public void Restart1Button()
{
    Losepanel.SetActive(false);
    SceneManager.LoadScene(1);
    Time.timeScale = 1f;
}
public void Restart2Button()
{
    Losepanel.SetActive(false);
    SceneManager.LoadScene(2);
    Time.timeScale = 1f;
}
}

```



Hình 17 Màn hình khi thua game

5.2 Aziratk: Hàm gọi và triệu hồi đòn tấn công của nhân vật

using System.Collections;

```

using System.Collections.Generic;
using UnityEngine;

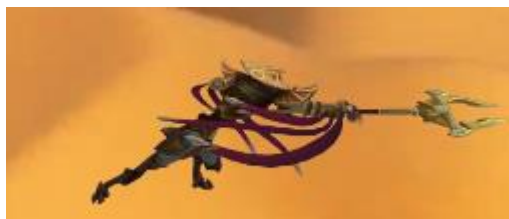
public class Aziratk : MonoBehaviour
{
    public Transform Startatk;
    public GameObject Skill;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            Atk();
        }
    }
    void Atk()
    {
        Instantiate(Skill, Startatk.position, Startatk.rotation);
    }
}

```

5.3 Skill: Gọi về kĩ năng và phá hủy kĩ năng



Hình 18 Kĩ năng của nhân vật

```

using System.Collections;
using System.Collections.Generic;

```

```

using UnityEngine;

public class Skill : MonoBehaviour
{
    public float speed = 10f;
    private Rigidbody2D rd2d;
    // Start is called before the first frame update
    void Awake()
    {
        rd2d = GetComponent<Rigidbody2D>();
    }

    void Start()
    {
        rd2d.velocity = transform.right * speed;
    }

    // Update is called once per frame
    void Update()
    {
        Destroy(gameObject, 1.0f);
    }
}

```

5.4 Cameracotrol: Điều hướng máy quay theo nhân vật và cố định máy quay

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Cameracotrol : MonoBehaviour
{
    private Transform Player;
    private float minX = 0f, maxX = 60f;
    private float minY = 1f, maxY = 32f;
    // Start is called before the first frame update
    private void Start()
    {

```

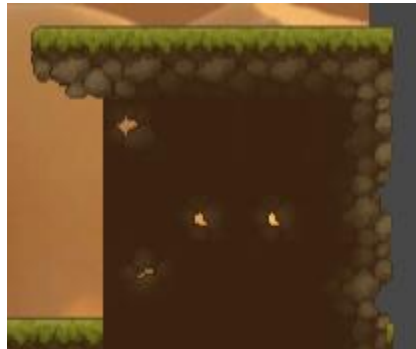
```

    Player = GameObject.FindWithTag("Player").transform;
}

// Update is called once per frame
private void Update()
{
    if(Player != null)
    {
        Vector3 vitri = transform.position; //co dinh may quay
        vitri.x = Player.position.x;
        vitri.y = Player.position.y;
        if (vitri.x < minX) vitri.x = 0;
        if (vitri.x > maxX) vitri.x = maxX;
        if (vitri.y < minY) vitri.y = 1;
        if (vitri.y > maxY) vitri.y = maxY;
        transform.position = vitri;
    }
}
}

```

5.5 Trieuhoi: Dùng để sinh ra mod trong màn 2



Hình 19 Khu vực sinh quái

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Trieuhoi : MonoBehaviour

```

```

{
    [SerializeField]
    private GameObject Mod;

    private BoxCollider2D Box;
    // Start is called before the first frame update
    void Awake()
    {
        Box = GetComponent<BoxCollider2D>();

    }
    void Start()
    {
        StartCoroutine(Thmod());
    }

    // Update is called once per frame
    void Update()
    {

    }

    IEnumerator Thmod()
    {
        yield return new WaitForSeconds(5f);
        Vector3 temp = transform.position;
        Instantiate(Mod, temp, Quaternion.identity);

        StartCoroutine(Thmod());
    }
}

```

5.6 Modcotrol: Xử lý va chạm và âm thanh của mod



Hình 20 Hình ảnh mod trong game

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Modcotroler : MonoBehaviour
{
    public float Tocdomod;
    public bool Dichuyentrai = true;
    public bool Dachet = false;
    private Animator ChuyendongMod;
    private Rigidbody2D r2d;
    public AudioClip Hrun;
    public AudioSource ado;

    void Start()
    {
        r2d = GetComponent<Rigidbody2D>();
        ChuyendongMod = GetComponent<Animator>();
    }
    private void FixedUpdate()
    {
        Chay();
        if (ado && Hrun)
        {
            ado.PlayOneShot(Hrun);
        }
    }
}

```

```

    }

}

void Chay()
{
    Vector2 Dichuyen = transform.localPosition;
    if (Dichuyentrai)
    {
        Dichuyen.x -= Tocdomod * Time.deltaTime;

    }

    else
    {
        Dichuyen.x += Tocdomod * Time.deltaTime;
    }

    transform.localPosition = Dichuyen;
}

private void OnCollisionEnter2D(Collision2D col2)
{
    if (col2.gameObject.CompareTag("New"))
    {
        Chet();
    }
    else if (col2.contacts[0].normal.x > 0)
    {
        Quaydau();
        Dichuyentrai = false;

    }
    else
    {
        Quaydau();
        Dichuyentrai = true;
    }
}

void Quaydau()

```

```

{
    Dichuyentrai = !Dichuyentrai;
    transform.Rotate(0f, 180f, 0f);
}
void Chet()
{
    Dachet = true;
    ChuyendongMod.SetTrigger("Die");
    Destroy(gameObject, 0.5f);
}
}

```

5.7 Menugame: Dùng để tương tác với Menu Game



Hình 21 Menu game

```

using UnityEngine;
using UnityEngine.SceneManagement;

public class Menugame : MonoBehaviour
{
    public void StartGame()
    {
        SceneManager.LoadScene(1);
    }
    public void Quitgame()

```

```

{
    Application.Quit();
}
}

```

5.8 Gameplaycotrol: Màn hình khi dừng game



Hình 22 Màn hình dừng game

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class Gameplaycotrol : MonoBehaviour
{
    [SerializeField]
    private GameObject Pausepanel;
    public void PauseGame()
    {
        Pausepanel.SetActive(true);
        Time.timeScale = 0f;
    }
}

```

```
public void ResumeButton()
{
    Pausepanel.SetActive(false);
    Time.timeScale = 1f;
}
```

```
public void Quitgame()
{
    SceneManager.LoadScene(0);
}
}
```

5.9 Backtomenu: Hiện kết thúc game và quay lại màn hình chính



Hình 23 Màn hình kết thúc game

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class Backtomenu : MonoBehaviour
{
    public void MenuButton()
    {
        SceneManager.LoadScene(0);
    }
}
```

```
}  
}
```

5.10 Nextlevel: Chuyển sang màn chơi tiếp theo



Hình 24 Khu vực chuyển màn chơi

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.SceneManagement;  
  
public class Nextlevel : MonoBehaviour  
{  
    // Start is called before the first frame update  
    void Start()  
    {  
  
    }  
  
    private void OnCollisionEnter2D(Collision2D collevel)  
    {  
        if (collevel.gameObject.CompareTag("Player"))  
        {  
            SceneManager.LoadScene(2);  
        }  
    }  
}
```

5.11 Wingame: Dùng để kết thúc game và chuyển qua màn hình kết thúc



Hình 25 Thánh chén chạm để kết thúc game

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class Wingame : MonoBehaviour
{
    // Start is called before the first frame update
    private void OnCollisionEnter2D(Collision2D collevel)
    {
        if (collevel.gameObject.CompareTag("Player"))
        {
            SceneManager.LoadScene(3);
        }
    }
}
```

TÀI LIỆU THAM KHẢO

1. <https://www.stdio.vn/unity-game/cac-thanh-phan-co-ban-trong-unity-Cf3Uw1>
2. <https://viblo.asia/p/co-ban-ve-engine-vat-ly-trong-unity-3OEqGj1PM9bL>
3. <https://lmss.info/3d-skin>