**TRƯỜNG CAO ĐẲNG THỰC HÀNH**

**FPT POLYTECHNIC**



**PHÁT TRIỂN ỨNG DỤNG GAME**

**“BÓNG TỐI HUYỀN BÍ”**

**Giảng viên hướng dẫn:** Vương Minh Thái

**Sinh viên:** PH55128 – Nguyễn Đắc Huy

Thành phố Hà Nội

Tháng 07/2024

Lời mở đầu

Do nhu cầu giải trí đối với mỗi người là không thể thiếu, ngành công nghiệp game trong những năm gần đây ngày càng khẳng định được vị trí của mình trên thị trường. Công ty Rồng Việt phát hành một tựa game có tên “**Bóng tối huyền bí**”, đây là tựa game dành cho người chơi trong lứa tuổi từ 7 đến 15 tuổi. Người chơi có thể dễ dàng thao tác và nắm bắt rõ ràng chỉ ngay khi vừa vào game và chơi ngay lập tức khi vừa tải xong mà không cần tìm hiểu quá phức tạp hoặc có bất kỳ hướng dẫn nào.

Trong quá trình làm dự án có nhiều khó khăn về thời gian cũng như kinh nghiệm làm việc của em còn yếu kém và điều đó làm em gặp nhiều rủi ro, thiếu sót không đáng có . Kính mong quý thầy cô góp ý để đề tài của em được hoàn thiện hơn. Em xin chân thành cảm ơn sự dạy dỗ và giúp đỡ của thầy cô!

TP. Hà Nội, ngày 7 tháng 6 năm 2024

Sinh viên

Nguyễn Đắc Huy

**MỤC LỤC:**

1. Giới thiệu dự án
2. KHẢO SÁT

2.1 Yêu cầu của khách hàng

2.2 Lập kế hoạch dự án

1. PHÂN TÍCH

3.1 Mô hình triển khai hệ thống

3.2 Sơ đồ Usecase

3.3 Đặc tả hệ thống

3.3.1 Usecase chiến đấu

3.3.2 Usecase nhặt vật phẩm

3.3.3 Usecase điều khiển nhân vật

1. THIẾT KẾ ỨNG DỤNG

4.1 Mô hình công nghệ

4.2 Thiết kế giao diện

4.2.1 Giao diện màn hình level 1

4.2.2 Giao diện màn hình level 2

4.2.3 Giao diện màn hình level 3

1. Thực hiện

5.1 Màn chơi level 1

5.2 Màn chơi level 2

5.3 Màn chơi level 3

5.4 Tổ chức lập trình

5.4.1 Tổ chức dự án

5.4.2 Tổ chức viết mã

5.4.3 Tổ chức animations

5.4.4 Tổ chức levels

5.4.4.1 Chi tiết màn chơi level 1

5.4.4.2 Chi tiết màn chơi level 2

5.4.4.3 Chi tiết màn chơi level 3

1. KIỂM THỬ
2. ĐÓNG GÓI VÀ TRIỂN KHAI
3. KẾT LUẬN
4. **Giới thiệu dự án:**

“Bóng tối huyền bí” là một trò chơi nhập vai vượt chướng ngại vật, qua các ải có độ khó khác nhau, được xây dựng với đồ họa 2D Pixel. Trong trò chơi này, người chơi được hóa thân thành một anh chàng cung thủ bị mắc kẹt trong hang động đầy rẫy sự nguy hiểm

“Bóng tối huyền bí” không chỉ đơn giản là một cuộc khai phá, mà còn là một thế giới đầy rẫy những quái vật, cạm bẫy cũng như các bí mật đang được chờ đợi để khám phá

Hãy sẵn sàng đối mặt với những thách thức nguy hiểm, thu thập các đồng tiền vàng và khám phá những bí mật sâu kín trong cuộc phiêu lưu này.

1. **KHẢO SÁT:**
   1. **Yêu cầu của khách hàng:**

Studio “Rồng việt” có định hướng muốn phát triển một tựa game có tên “Bóng tối huyền bí” dành cho người chơi trong lứa tuổi từ 7-15 tuổi. Game cần có nội dung, cốt truyện và lối chơi đơn giản, có thể dễ dàng thao tác và nắm bắt chỉ ngay khi vừa vào game và chơi ngay lập tức khi tải xong mà không cần tìm hiểu quá phức tạp hoặc hướng dẫn nào.

* Yêu cầu của người chơi:

+ Có thể thao tác 1 cách dễ dàng và nắm bắt khi vào game

+ Chơi ngay lập tức khi tải xong mà không cần tìm hiểu quá phức tạp

* Yêu cầu các chức năng của trò chơi:

+ Giao diện đơn giản, dễ nắm bắt nội dung game.

+ Cần có các thao tác điều khiển nhân vật như chạy, nhảy, tấn công.

+ Xây dựng hệ thống animation cho nhân vật, quái vật trong game.

+ Cần có hệ thống nhiệm vụ trên từng màn chơi.

+ Đồ họa thân thiện, âm thanh nhẹ nhàng.

+ Thông tin kết quả trò chơi cần được ghi nhận lại.

* Công nghệ:

+ Unity Engine 2D

+ Visual Studio C#

* Nhiệm vụ trong game:

+ Vượt chướng ngại vật

+ Ăn xu xuất hiện trong mỗi màn chơi

+ Về đích

* Nền tảng hỗ trợ: PC (Window, macOS) ; Mobile (Android, IOS)
  1. **Lập kế hoạch dự án:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Cô**ng việc | **Bắt đầu** | **Kết thúc** | **Thực hiện** | **Trạng thái** |
| **1** | **Ph**ân tích yêu cầu khách hàng | | | | |
| 1.1 | Vẽ sơ đồ tổng quan hệ thống |  |  |  |  |
| 1.2 | Vẽ sơ đồ use case |  |  |  |  |
| 1.3 | Xây dựng bản đặc tả yêu cầu hệ thống |  |  |  |  |
| 1.4 | Mô tả nghiệp vụ |  |  |  |  |
| **2** | **Thi**ết kế hệ thống | | | | |
| 2.1 | Thiết kế hình ảnh dùng cho game |  |  |  |  |
| 2.2 | Thiết kế giao diện bản đồ trong game |  |  |  |  |
| 2.3 | Thiết kế hình ảnh nhân vật, quái vật, vật phẩm |  |  |  |  |
| 2.4 | Thiết kế animation cho nhân vật |  |  |  |  |
| 2.5 | Thiệt kế hệ thống nhiệm vụ trong từng bản đồ |  |  |  |  |
| 2.6 | Thiết kế âm thanh cho game |  |  |  |  |
| **3** | **Th**ực hiện dự án | | | | |
| 3.1 | Xây dựng bản đồ cho từng cấp độ |  |  |  |  |
| 3.2 | Xây dựng hình ảnh nhân vật |  |  |  |  |
| 3.3 | Xây dựng hình ảnh quái vật, vật phẩm |  |  |  |  |
| 3.4 | Xây dựng hiệu ứng hoạt hình cho nhân vật, quái vật |  |  |  |  |
| 3.5 | Xây dựng giao diện menu, bảng xếp hạng |  |  |  |  |
| 3.6 | Lập trình điều khiển chuyển động của nhân vật |  |  |  |  |
| 3.7 | Lập trình xử lý tương tác giữa nhân vật và các đối tượng trong game |  |  |  |  |
| 3.8 | Thiết kế quái vật |  |  |  |  |
| 3.9 | Thiết kế màn chơi |  |  |  |  |
| 3.10 | Lập trình các thao tác |  |  |  |  |
| 3.11 | Lập trình xử lý menu |  |  |  |  |
| 3.12 | Lập trình thao tác nhân vật |  |  |  |  |
| 3.14 | Lập trình cơ chế quái vật |  |  |  |  |
| 3.15 | Lập trình xử lý ghi nhận kết quả màn chơi |  |  |  |  |
| 3.16 | Lập trình xử lý âm thanh |  |  |  |  |
| **4** | **Ki**ểm thử | | | | |
| 4.1 | Xây dựng kịch bản kiểm thử game |  |  |  |  |
| 4.2 | Xây dựng kịch bản kiểm thử game |  |  |  |  |
| 4.3 | Thực hiện kiểm thử game |  |  |  |  |
| 4.4 | Lập trình sửa lỗi |  |  |  |  |
| **5** | **Đó**ng gói và triển khai | | | | |
| 5.1 | Đóng gói sản phẩm |  |  |  |  |
| 5.2 | Viết tài liệu hướng dẫn sử dụng |  |  |  |  |

1. **Phân tích:**
2. **Mô hình triển khai hệ thống:** 
   1. **Mô hình triển khai:**

Trò chơi được triển khai theo mô hình Client-Only, vì đây là 1 tựa game offline nên không yêu cầu kết nối mạng

* 1. **Thiết kế hệ thống của Game:**
* Game Manager: Quản lý trạng thái tổng thế của trò chơi, điều khiển chuyển đối giữa các màn hình chơi ( menu chính, màn hình chơi game, màn hình kết thúc )

Thành phần:

+ Victory ( Game over ) Screen: Giao diện hiển thị khi người chơi hoàn thành hoặc thất bại trong một màn chơi

* Level Manger: Quản lý các màn chơi, bao gồm cả dữ liệu màn chơi, khởi tạo đối tượng trong màn chơi cũng như xử lý các sự kiện trong màn chơi

Thành phần:

+ Level Data: Dữ liệu màn chơi được lưu trữ thông tin về bố cục và vị trí các đối tượng

+ Object Pool: Quản lý tái sử dụng các đối tượng để tối ưu hóa hiệu suất

* Player Controller: Quản lý hành vi và điều khiển của nhân vật chính, bao gồm di chuyển, nhảy và thu thập xu
* Thành phần:

+ Input Handler: Xử lý đầu vào từ người chơi như bàn phím, chuột, cảm ứng

+ Movement Controller: Quản lý chuyển động và vật lý của nhân vật chính

+ Collision Handler: Xử lý va chạm của nhân vật chính với các đối tượng trong màn chơi.

* UI Manager: Quản lý giao diện người dùng, hiển thị các thông tin như số xu thu thập, các thông báo

Thành phần:

+ Score Display: Hiển thị số xu đã thu thập

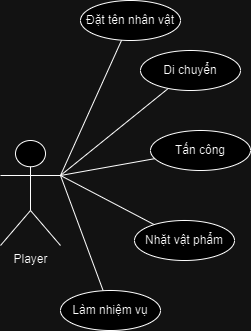
+ Notification: Hiển thị các thông báo ( ví dụ: hoàn thành màn chơi )

* Audio Manager: Quản lí âm thanh và nhạc nền khi chơi
* Thành phần:

+ Background music: Quản lí nhạc nền khi chơi

+ Sound effect: Quản lí các âm thanh khi ăn xu

1. **Sơ đồ Usecase:**



1. **Đặc tả hệ thống:**

* Player: người chơi sẽ điều khiển nhân vật trong game để vượt chướng ngại vật, nhặt xu và về đích

+ Coin: số xu Player có

+ Character\_name: tên nhân vật mà người chơi đặt

+ Position in game: vị trí của nhân vật khi người chơi di chuyển

+ Life: mạng sống của nhân vật

* Levels: các màn chơi mà người sẽ chọn

+ Coin: số xu trong các màn chơi

+ Trap: bẫy trong màn chơi, gây khó dễ cho nhân vật

+ Enemy: quái vật trong màn chơi, giúp người chơi khó khăn hơn trong việc vượt ải

* Trap và Enemy ảnh hưởng đến Life của người chơi, nếu người chơi đụng phải 1 trong 2 cái trên, màn hình Game Over sẽ hiện ra

+ FinishGoal: vạch đích trong màn chơi, giúp người chơi qua màn

* Người chơi điều khiển nhân vật tiến tới vạch đích 1 cách an toàn, khi chạm tới FinishGoal thì màn hình Level Complete sẽ hiện ra

1. **Thiết kế ứng dụng:**
2. **Mô hình công nghệ:**

* Engine phát triển: Unity
* Ngôn ngữ lập trình: Microsoft Visual Studio C#

1. **Thiết kế giao diện:**
2. Giao diện Level 1:

**Mô tả hoạt động:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Điều khiển** | **Sự kiện** | **Mô tả hoạt động** |
| **1** | [A,D]  [Mũi tên trái, phải ] | Nhấn nút | Di chuyển nhân vật trái phải |
| **2** | [Spacebar] | Nhấn nút | Giúp nhân vật nhảy vượt chướng ngại vật |

Bảng 4.1: Mô tả hoạt động của màn chơi 1

1. Giao diện Level 2:

**Mô tả hoạt động:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Điều khiển** | **Sự kiện** | **Mô tả hoạt động** |
| **1** | [A,D]  [Mũi tên trái, phải ] | Nhấn nút | Di chuyển nhân vật trái phải |
| **2** | [Spacebar] | Nhấn nút | Giúp nhân vật nhảy vượt chướng ngại vật |
| **3** | [Q] | Nhấn nút | Tấn công tiêu diệt quái |

Bảng 4.2: Mô tả hoạt động của màn chơi 2

1. Giao diện Level 3:

**Mô tả hoạt động:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Điều khiển** | **Sự kiện** | **Mô tả hoạt động** |
| **1** | [A,D]  [Mũi tên trái, phải ] | Nhấn nút | Di chuyển nhân vật trái phải |
| **2** | [Spacebar] | Nhấn nút | Giúp nhân vật nhảy vượt chướng ngại vật |
| **3** | [Q] | Nhấn nút | Tấn công tiêu diệt quái |

Bảng 4.3: Mô tả hoạt động của màn chơi 3

1. **Thực hiện:**
2. **Màn chơi Level 1:**

Hình 5.1: Màn chơi Level 1

**Mô tả hoạt động:**

* Dùng phím A, D hoặc Mũi tên trái, mũi tên phải để di chuyển nhân vật
* Dùng phím Space để điều khiển nhân vật nhảy lên cao né chướng ngại vật
* Trên bản đồ có các đồng xu, nhân vật thu thập xu sẽ được ghi nhận điểm, điểm sẽ hiển thị trên góc màn hình bên phải
* Nhân vật chỉ có 1 mạng chơi, nếu chạm phải bẫy, nhân vật sẽ chết và sẽ phải chơi lại màn chơi đó
* Ở cuối màn chơi sẽ có chỗ thoát hiểm, khi nhân vật đến điểm này sẽ thắng và tiếp tục tiến đến màn tiếp theo

1. **Màn chơi Level 2:**

Hình 5.2: Màn chơi Level 2

**Mô tả hoạt động:**

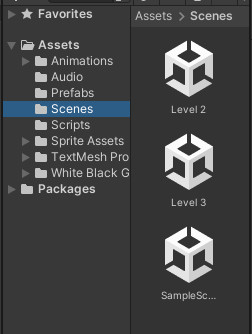
* Dùng phím A, D hoặc Mũi tên trái, mũi tên phải để di chuyển nhân vật
* Dùng phím Space để điều khiển nhân vật nhảy lên cao né chướng ngại vật
* Trên bản đồ có thang, nhân vật tiếp xúc với thang và bấm nút Mũi tên lên để leo
* Trên bản đồ có các đồng xu, nhân vật thu thập xu sẽ được ghi nhận điểm, điểm sẽ hiển thị trên góc màn hình bên phải
* Ngoài các đồng xu còn có các quái vật, người chơi bấm phím Q để bắn ra mũi tên tiêu diệt quái, quát chạm vào mũi tên sẽ biến mất. Nhưng nếu người chơi chạm vào quái sẽ chết và chơi lại màn chơi đó
* Nhân vật chỉ có 1 mạng chơi, nếu chạm phải bẫy, nhân vật sẽ chết và sẽ phải chơi lại màn chơi đó
* Ở cuối màn chơi sẽ có chỗ thoát hiểm, khi nhân vật đến điểm này sẽ thắng và tiếp tục tiến đến màn tiếp theo

1. **Màn chơi Level 3:**

Hình 5.3: Màn chơi Level 3

**Mô tả hoạt động:**

* Dùng phím A, D hoặc Mũi tên trái, mũi tên phải để di chuyển nhân vật
* Dùng phím Space để điều khiển nhân vật nhảy lên cao né chướng ngại vật
* Trên bản đồ có thang, nhân vật tiếp xúc với thang và bấm nút Mũi tên lên để leo
* Trên bản đồ có các đồng xu, nhân vật thu thập xu sẽ được ghi nhận điểm, điểm sẽ hiển thị trên góc màn hình bên phải
* Nhân vật chỉ có 1 mạng chơi, nếu chạm phải bẫy, nhân vật sẽ chết và sẽ phải chơi lại màn chơi đó
* Ở cuối màn chơi sẽ có chỗ thoát hiểm, khi nhân vật đến điểm này sẽ thắng và tiếp tục tiến đến màn tiếp theo

1. **Tổ chức lập trình:**
2. **Tổ chức dự án:**

Hình 5.4: Tổ chức của dự án

|  |  |  |
| --- | --- | --- |
| **TT** | **Thư mục** | **Mô tả chức năng** |
| **1** | Animations | Chứa các animations của nhân vật, quái vật, xu |
| **2** | Audio | Chứa các tệp tin âm thanh |
| **3** | Prefabs | Chứa các tài nguyên dùng chung của game |
| **4** | Scenes | Chứa các màn chơi của game |
| **5** | Scripts | Chứa các code điều khiển trong game |
| **6** | Sprite Assets | Chứa các hình ảnh dùng trong game |
| **7** | TextMesh Pro | Chứa các tài nguyên của thư viện TMP |
| **8** | White and Black Icon | Chứa các Icon của game |

Bảng 5.4: Mô tả tổ chức dự án

1. **Tổ chức viết mã:**

* **Mã điều khiển nhân vật:**

|  |
| --- |
| using System;  using System.Collections;  using System.Collections.Generic;  using TMPro;  using UnityEditor.SearchService;  using UnityEngine;  using UnityEngine.SceneManagement;  using UnityEngine.SocialPlatforms.Impl;  using UnityEngine.Timeline;  using UnityEngine.UI;  public class Player\_Ctrl : MonoBehaviour  {  [SerializeField] private float atkSpeed, countDown = 0;  [SerializeField] private float \_speedMove = 450;  [SerializeField] private float jumpHigh = 6;  private SpriteRenderer spd;  public float climbSpeed = 3f;  private bool isClimbing = false;  [SerializeField] private GameObject bullet;  [SerializeField] Transform firerate;  public Slider \_slide;  private Rigidbody2D rb;  private Animator ani;  private string currentAnim;  public float maxHearlt;  public GameObject Panel\_L, Button\_L, Text\_L;  public GameObject Panel\_W, Button\_W, Text\_W, Button\_WR;  public TMP\_Text Score;  private int countScore = 0;  // Start is called before the first frame update  void Start()  {  Score.text = " " + countScore;  rb = GetComponent<Rigidbody2D>();  ani = GetComponent<Animator>();  spd = GetComponent<SpriteRenderer>();  }  // Update is called once per frame  void Update()  {  Move();  Jump();  Attack();  }  void Move()  {  float horizontal = Input.GetAxis("Horizontal");  float vertical = Input.GetAxis("Vertical");  Vector2 movement = new Vector2(horizontal, vertical).normalized;  if (Math.Abs(horizontal) > 0.1f || Math.Abs(vertical) > 0.1f)  {  changeAnim("run");  transform.rotation = Quaternion.Euler(new Vector3(0, (horizontal > 0.1f) ? 0 : -180, 0));  rb.velocity = movement \* \_speedMove \* Time.deltaTime;  }  else  {  changeAnim("idle");  rb.velocity = Vector2.zero;  }  }  private void changeAnim(string AnimName)  {  if (currentAnim != AnimName)  {  ani.ResetTrigger(AnimName);  currentAnim = AnimName;  ani.SetTrigger(currentAnim);  }  }  void Jump()  {  if (Input.GetKey(KeyCode.Space))  {  rb.AddForce(Vector2.up \* jumpHigh, ForceMode2D.Impulse);  }  }  private void OnTriggerEnter2D(Collider2D collision)  {  if (collision.CompareTag("Ladder"))  {  isClimbing = true;  rb.gravityScale = 0f;  changeAnim("climb");  }  if (collision.gameObject.CompareTag("Coin"))  {  countScore++;  Audio\_Manager.Instance.PlaySFX("Coin");  Destroy(collision.gameObject);  Score.SetText(countScore.ToString());  }  }  private void OnTriggerExit2D(Collider2D collision)  {  if (collision.CompareTag("Ladder"))  {  isClimbing = false;  rb.gravityScale = 25f;  changeAnim("idle");  }  }  private void FixedUpdate()  {  if(isClimbing)  {  float climbInput = Input.GetAxisRaw("Vertical");  rb.velocity = new Vector2(rb.velocity.x, climbInput \* climbSpeed);  }  }  void Attack()  {  countDown -= Time.deltaTime;  if (countDown > 0)  {  return;  }  if (Input.GetKeyDown(KeyCode.Q))  {  changeAnim("attack");  Instantiate(bullet, firerate.position, transform.rotation);  countDown = atkSpeed;  }  }  private void OnCollisionEnter2D(Collision2D col)  {  if (col.gameObject.CompareTag("trap"))  {  Time.timeScale = 0;  Panel\_L.SetActive(true);  Button\_L.SetActive(true);  Text\_L.SetActive(true);  Audio\_Manager.Instance.musicSource.Stop();  }  if (col.gameObject.CompareTag("enemy"))  {  Time.timeScale = 0;  Panel\_L.SetActive(true);  Button\_L.SetActive(true);  Text\_L.SetActive(true);  Audio\_Manager.Instance.musicSource.Stop();  }  if (col.gameObject.CompareTag("Finish"))  {  Time.timeScale = 0;  Panel\_W.SetActive(true);  Button\_W.SetActive(true);  Text\_W.SetActive(true);  Button\_WR.SetActive(true);  Audio\_Manager.Instance.musicSource.Stop();  }  }  } |

* **Mã điều khiển của mũi tên:**

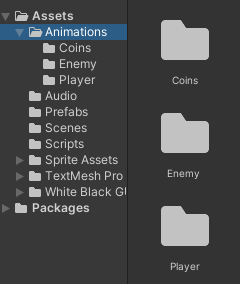
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Bullet : MonoBehaviour  {  public float speed;  public float lifeTime;  public GameObject effect\_bullet;  Rigidbody2D rb;  // Start is called before the first frame update  void Start()  {  rb = GetComponent<Rigidbody2D>();  Destroy(gameObject, lifeTime);  }  // Update is called once per frame  void Update()  {  rb.velocity = transform.right \* speed \* Time.deltaTime;  }  private void OnTriggerEnter2D(Collider2D other)  {  if (other.gameObject.CompareTag("enemy"))  {  Destroy(gameObject);  GameObject Explode = Instantiate(effect\_bullet, transform.position, Quaternion.identity);  Destroy(Explode, 0.1f);  var name = other.attachedRigidbody.name;  Destroy(GameObject.Find(name));  }  }  } |

* **Mã điều khiển nhặt xu:**

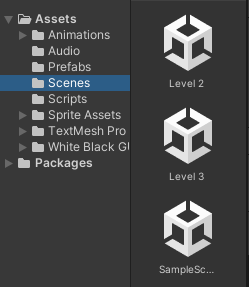
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Coin\_Collect : MonoBehaviour  {  // Start is called before the first frame update  void Start()  {  }  // Update is called once per frame  void Update()  {    }  public void OnTriggerEnter2D(Collider2D other)  {  if (other.gameObject.CompareTag("Coin"))  {  Destroy(gameObject);  }  }  } |

* **Mã điều khiển quái vật:**

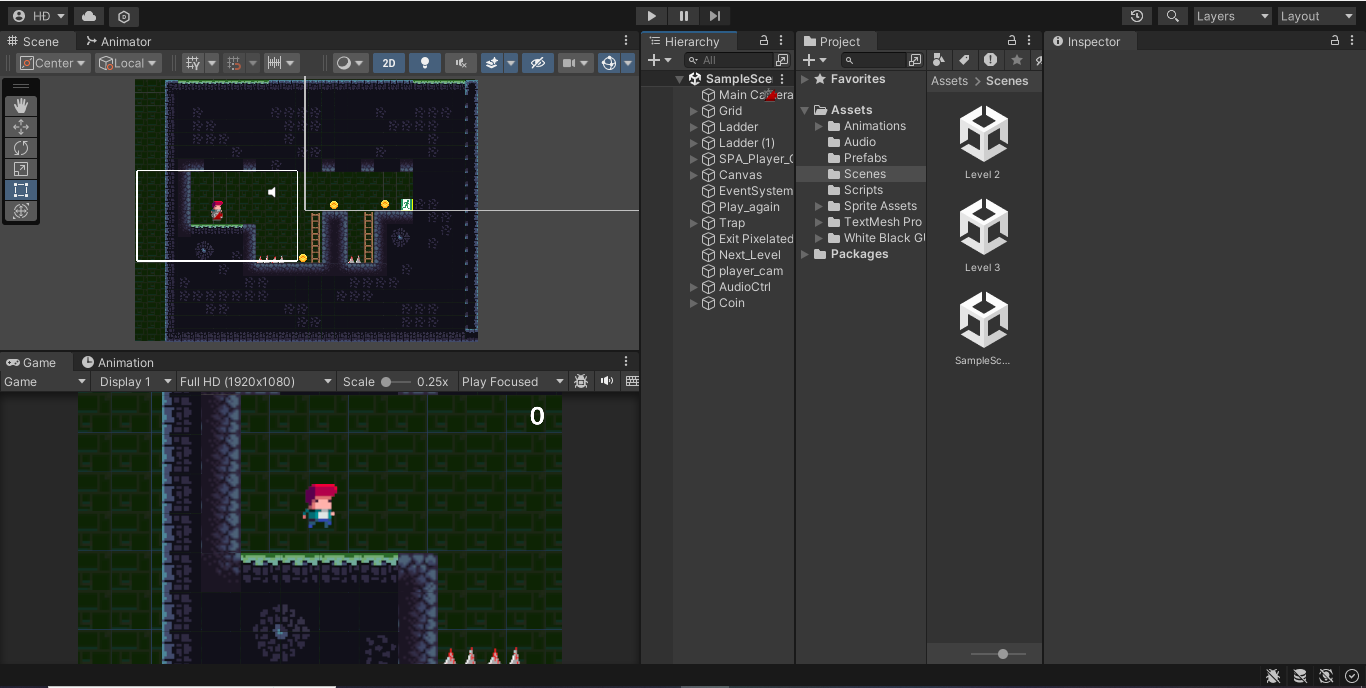
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using Unity.VisualScripting;  using UnityEngine;  public class Enemy\_Ctrl : MonoBehaviour  {  public float speed;    public Transform \_player;  public SpriteRenderer enemySR;  private Rigidbody2D rb;  private string currentAnim;  private Animator ani;  private int direction = 0;  // Start is called before the first frame update  void Start()  {  GameObject playerObject = GameObject.FindGameObjectWithTag("Player");  if (playerObject == null)  {  playerObject = FindAnyObjectByType<GameObject>();  }  if (playerObject != null)  {  \_player = playerObject.transform;  }  ani = GetComponent<Animator>();  rb = GetComponent<Rigidbody2D>();  }  // Update is called once per frame  void Update()  {  if (\_player != null)  {  Vector2 dir = (\_player.position - transform.position).normalized;  Vector3 faceEnemy = dir \* speed \* Time.deltaTime;  transform.Translate(faceEnemy);  if (faceEnemy.x != 0)  {  if (faceEnemy.x < 0)  {  direction \*= 1;  enemySR.transform.localScale = new Vector3(1, 1, 1);  }  else  {  direction \*= -1;  enemySR.transform.localScale = new Vector3(-1, 1, 1);  }  }  }  }  private void changeAnim(string AnimName)  {  if (currentAnim != AnimName)  {  ani.ResetTrigger(AnimName);  currentAnim = AnimName;  ani.SetTrigger(currentAnim);  }  }  private void OnTriggerEnter2D(Collider2D collision)  {  if (collision.CompareTag("arrow"))  {  rb.AddForce(Vector2.up \* 15, ForceMode2D.Impulse);  }  }  } |

1. **Tổ chức animations:**

Ở đây mình khai báo các animations controller dành Coins, Enemy và Player. Các animations clip của các đối tượng đều được chia ra trong từng thư mục như ảnh trên.

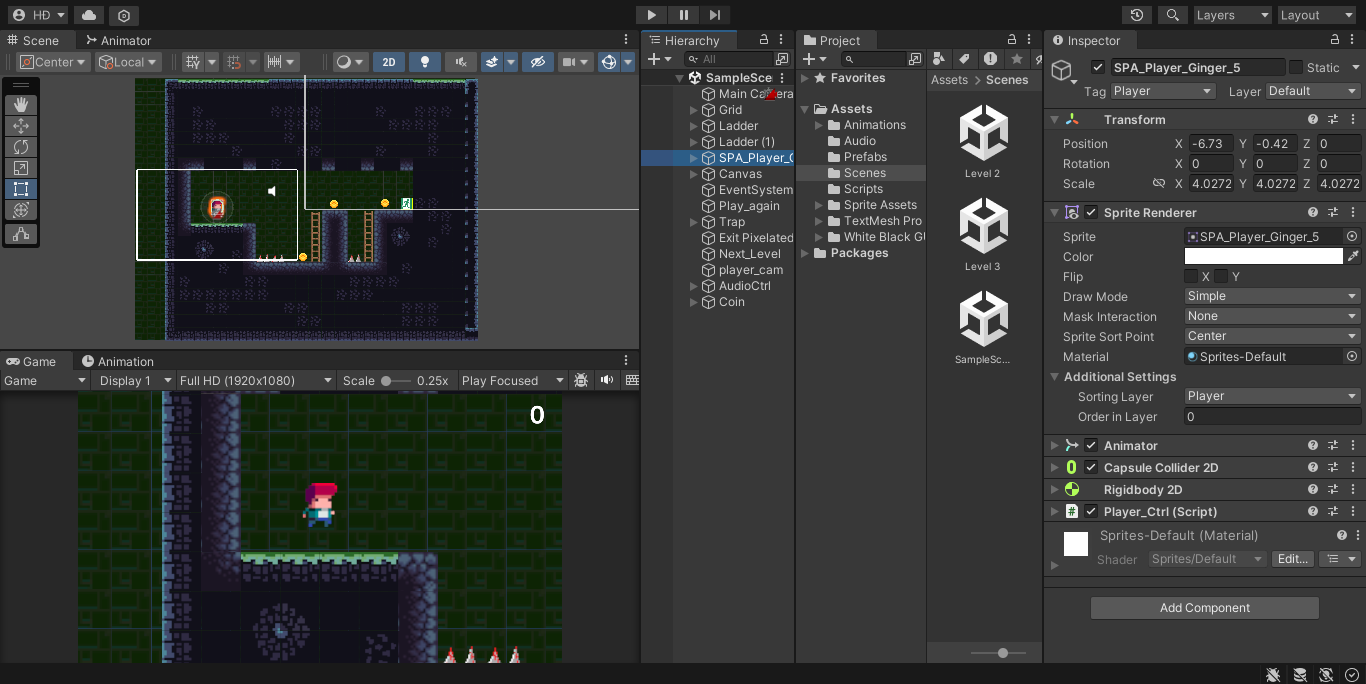
1. **Tổ chức Levels:**

Các màn chơi đều ở trong thư mục này, hiện tại trò chơi của em có 3 màn chơi tất cả

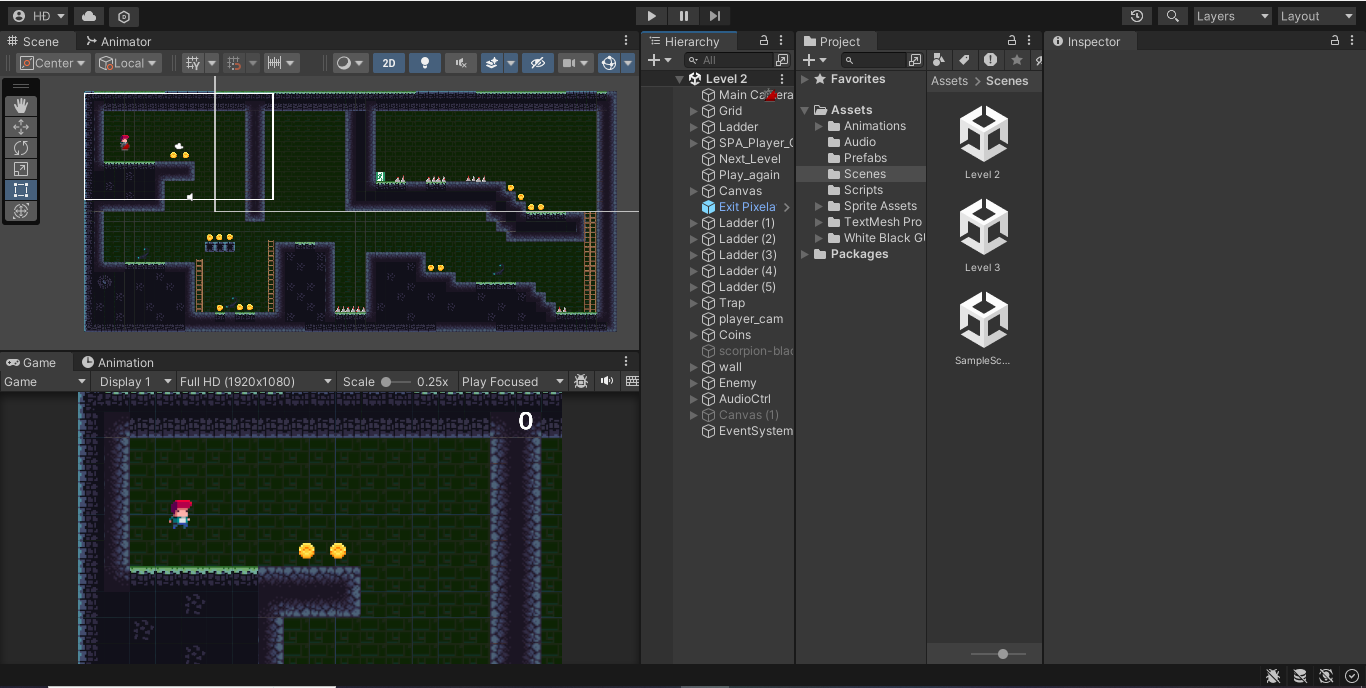
1. Chi tiết màn chơi Level 1:

Màn chơi Level 1 sẽ có:

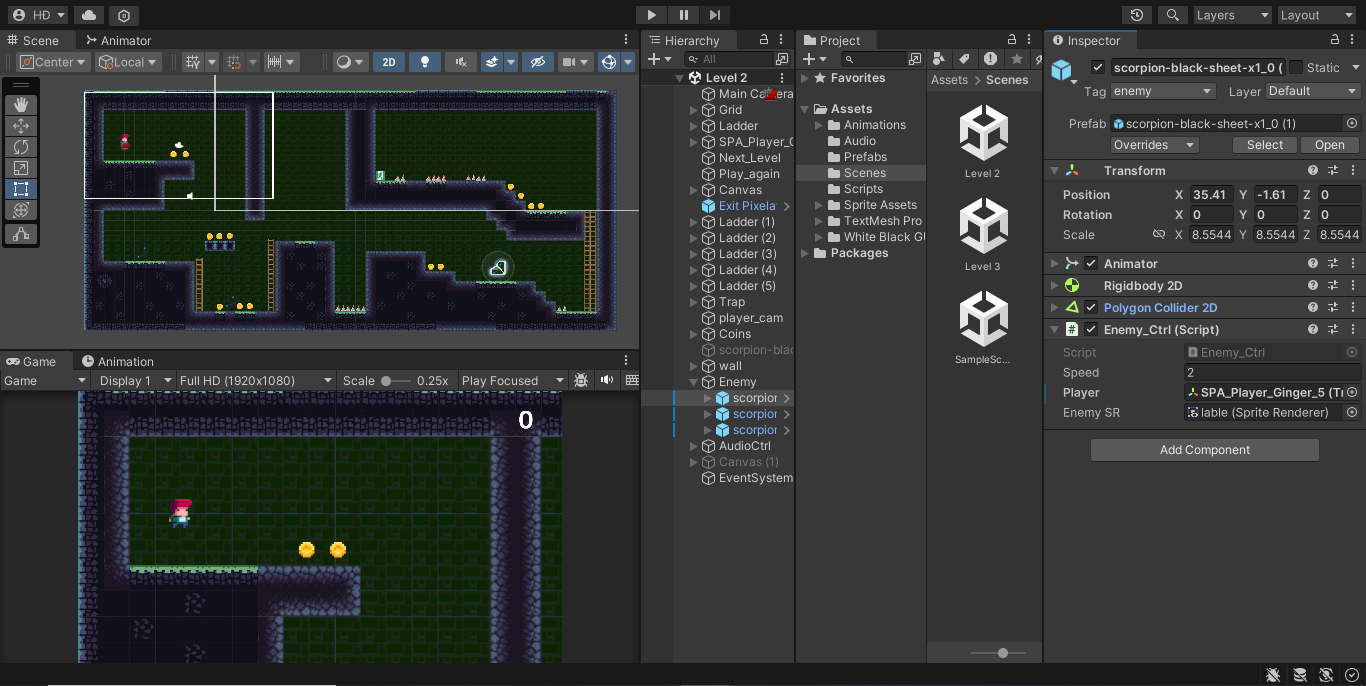
* Player: nhân vật chính của trò chơi, có thể chạy, nhảy, tấn công
* Xu: vật phẩm trong game mà nhân vật có thể thu thập
* Bẫy: chướng ngại vật, nếu nhân vật chạm vào sẽ chết
* Escape: Kết thúc 1 màn chơi

Thông tin chi tiết của Player:

* Animator: Điều khiển animation cho nhân vật
* Capsule Collider 2D: Dùng để tương tác với các đối tượng trong game
* Rigidbody 2D: tương tác vật lý
* Player\_Ctrl: mã code giúp cho nhân vật có thể chạy, nhảy, tấn công

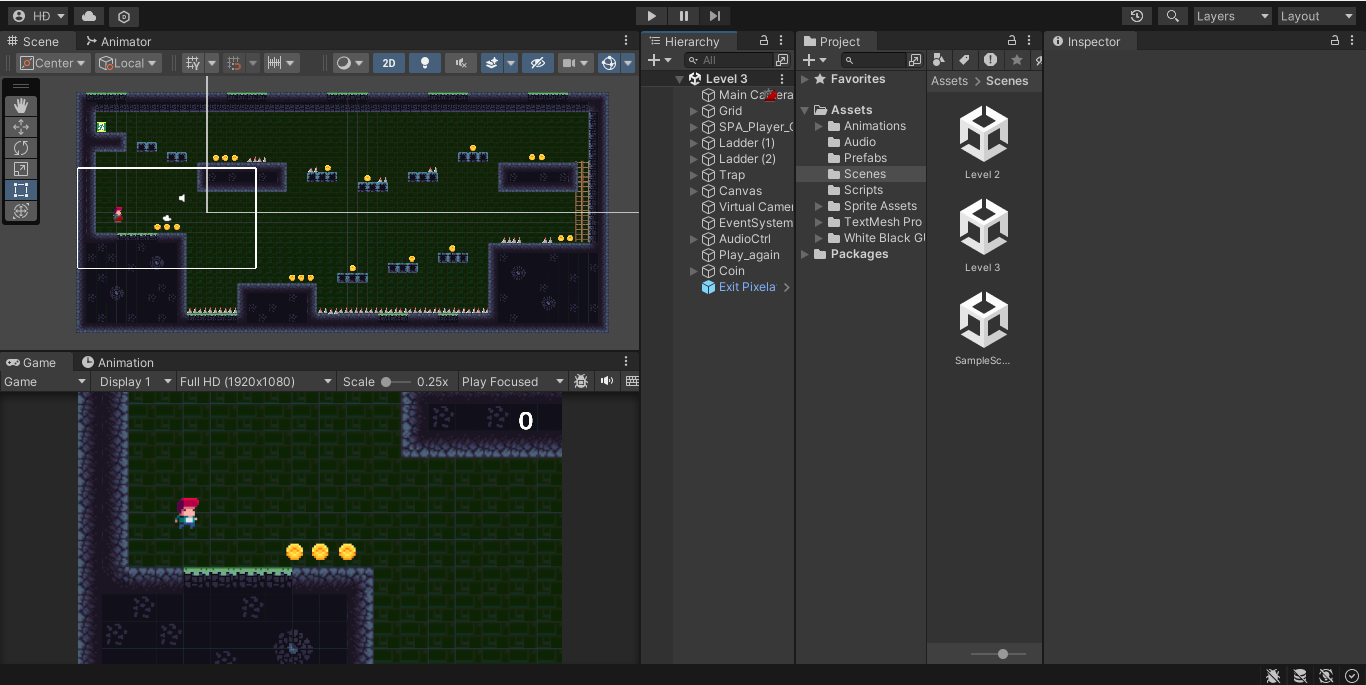
1. Chi tiết màn chơi Level 2:

Màn chơi Level 2 sẽ có:

* Player: nhân vật chính của trò chơi, có thể chạy, nhảy, tấn công
* Xu: vật phẩm trong game mà nhân vật có thể thu thập
* Bẫy: chướng ngại vật, nếu nhân vật chạm vào sẽ chết
* Quái: luôn đuổi theo nhân vật, nếu chạm vào sẽ chết
* Escape: Kết thúc 1 màn chơi

Thông tin chi tiết của quái vật:

* Animator: Điều khiển animation cho quái vật
* Rigidbody 2D: tương tác vật lý
* Polygon Collider 2D: Dùng để xác định các đối tượng khác trong game
* Enemy\_Ctrl: code giúp quái đuổi theo nhân vật

1. Chi tiết màn chơi Level 3:

Màn chơi Level 3 sẽ có:

* Player: nhân vật chính của trò chơi, có thể chạy, nhảy, tấn công
* Xu: vật phẩm trong game mà nhân vật có thể thu thập
* Bẫy: chướng ngại vật, nếu nhân vật chạm vào sẽ chết
* Escape: đây là màn cuối cùng, chạm vào sẽ kết thúc game

1. Kiểm thử:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case Title** | **Expected Result** | **Actual Result** | **Run type** | **Test Step Details** | **Note** |
| **1** | Di chuyển trái | Thành công | Thành công | Manual | Nhấn ( giữ ) phím mũi tên trái hoặc phím A | Pass |
| **2** | Di chuyển phải | Thành công | Thành công | Manual | Nhấn ( giữ ) phím mũi tên phải hoặc phím D | Pass |
| **3** | Di chuyển lên trên ( leo thang ) | Thành công | Thành công | Manual | Chạm vào thang và bấm mũi tên lên hoặc phím W để bắt đầu leo | Pass |
| **4** | Nhảy | Nhảy thấp và khi giữ bị bay cao | Thất bại |  | Nhấn ( giữ ) nút SpaceBar | Fail |
| **5** | Tấn công | Bắn đạn tại vị trí của nhân vật khi bấm bắn | Bắn đạn tại vị trí của nhân vật khi bấm bắn | Manual | Nhấn Q | Pass |
| **6** | Thu thập xu | Thu thập xu trên đường đi | Thu thập xu trên đường đi | Manual | Di chuyển nhân vật chạm vào xu, chạm vào thì thu thập và tăng điểm | Pass |
| **7** | Hoạt ảnh nhân vật | Hoạt ảnh bình thường, leo thang thi thoảng bị lỗi | Hoạt ảnh bình thường, leo thang thi thoảng bị lỗi | Manual | Vào game, di chuyển nhân vật | Fail |
| **8** | Hoạt ảnh quái vật | Hoạt ảnh bình thường, thay đổi khi chạm đạn và chết | Hoạt ảnh bình thường, thay đổi khi chạm đạn và chết | Manual | Vào game, di chuyển tới vị trí của quái, tấn công quái | Pass |
| **9** | Va chạm với các lề của bản đồ | Nhân vật bị cản, không đi qua trực tiếp | Nhân vật bị cản, không đi qua trực tiếp | Manual | Di chuyển đến bên lề của bản đồ | Pass |
| **10** | Va chạm với đồng xu | Thu thập xu | Nhân vật thu thập được xu | Manual | Di chuyển đến đồng xu | Pass |
| **11** | Va chạm với quái, bẫy | Nhân vật bị chết | Nhân vật bị chết | Manual | Di chuyển đến quái | Pass |

**Kết quả kiểm thử:**

Hình 6.1: Biểu đồ kiểm tra chức năng trò chơi

Đối với trò chơi:

* PASS: 9
* FAIL: 2

1. **Đóng gói và triển khai:**

* **Đường dẫn cài đặt:**

<https://drive.google.com/file/d/1J5O7c8Cp7s6d67Q6PoY_keusbwH0Mieb/view?usp=sharing>

* **Hướng dẫn cài đặt:**

Người chơi tải game về thiết bị ở link tải game. Sau đó vào game và bắt đầu chơi

1. **Kết luận:**
2. **Những điều đã làm trong trò chơi:**

* Hiển thị thông tin trò chơi: số điểm
* Điều khiển chuyển động của nhân vật
* Xây dựng animation cho nhân vật, quái vật
* Sử dụng Tilemap

1. **Điều còn chưa hoàn thành được:**

* Lưu trữ thông tin trò chơi
* Chưa xây dựng được màn hình Pause Game
* Còn thiết kỹ năng xây dựng, thiết kế