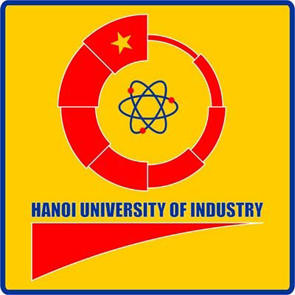
|  |  |
| --- | --- |
| NGUYỄN HỮU TĂNG CÔNG NGHỆ THÔNG TIN | **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **KHOA CÔNG NGHỆ THÔNG TIN**  **======\*\*\*======**  **ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC**  NGÀNH CÔNG NGHỆ THÔNG TIN  **Đề tài: Xây dựng ứng dụng game 2D Sword Hunter bằng Unity**  **CBHD :** ThS. Nguyễn Đức Lưu  **Sinh viên:** Nguyễn Hữu Tăng  **Mã SV :** 2019606645  **Lớp :** CNTT06 **Khóa:** K14  **Hà Nội – Năm 2023** |

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

**ĐỒ ÁN TỐT NGHIỆP/KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC**

CÔNG NGHỆ THÔNG TIN

**Đề tài: Xây dựng ứng dụng game 2D Sword Hunter bằng Unity**

**CBHD :** ThS. Nguyễn Đức Lưu

**Sinh viên:** Nguyễn Hữu Tăng

**Mã SV :** 2019606645

**Lớp :** CNTT06 **Khóa:** K14

**Hà Nội – 2023**

**MÔ TẢ TÓM TẮT ĐỀ TÀI**

**Họ và tên:** Nguyễn Hữu Tăng

**Mã sinh viên:** 2019606645

**Tên đề tài:** Xây dựng ứng dụng game 2D Sword Hunter bằng Unity

**Giới thiệu đề tài**

Thị trường game Việt Nam vẫn còn nhỏ bé và chưa được nhiều tập đoàn lớn quan tâm. Tuy nhiên, Việt Nam đang trong thời điểm chín muồi để khai thác tiềm năng phát triển cả về khách hàng và lực lượng công nghệ game phát triển mạnh mẽ. Chính vì lý do đó mà em quyết định chọn đề tài ***“Xây dựng ứng dụng game 2D Sword Hunter bằng Unity”***

**Mục tiêu đề tài**

* Tìm hiểu phần mềm lập trình game Unity
* Xây dựng được một ứng dụng game với một số chức năng cơ bản
* Cài đặt và triển khai ứng dụng

**Kết quả đạt được**

* Tìm hiểu, vận dụng được các kiến thức để sử dụng phần mềm
* Xây dựng được các thành phần trong ứng dụng game, hoàn thiện được các chức năng cơ bản
* Rút ra được kinh nghiệm khi phát triển ứng dụng
* Hoàn thiện chương trình và báo cáo

**SUMMARY DESCRIPTION OF THEME**

**Full name:** Nguyen Huu Tang

**Student code:** 2019606645

**Subject name:** Build a 2D Sword Hunter game application using Unity

**Topic introduction**

Vietnam's game market is still small and has not been interested by many large corporations. However, Vietnam is at the ripe time to exploit its potential for growth both in terms of customers and a thriving game technology force. That's why I decided to choose the topic ***"Building a 2D Sword Hunter game application with Unity"***.

**Topic goal**

* Learn game programming software Unity
* Build a game application with some basic functions
* Install and deploy the application

**Result**

* Learn and apply knowledge to use software
* Build components in game applications, complete basic functions
* Gain experience when developing applications
* Finalize the program and report

LỜI CẢM ƠN

Để hoàn thành được kháo luận về đề tài: ***“Xây dựng ứng dụng game 2D Sword Hunter bằng Unity”***, em xin chân thành cảm ơn Trường Đại học Công nghiệp Hà Nội, cùng với các thầy, cô khoa Công nghệ thông tin đã day dỗ em trong quá trình học tập và rèn luyện; nhờ công ơn của Nhà Trường và thầy cô đã giúp em có được những kiến thức để hoàn thành khóa luận này. Đặc biệt, em xin gửi lời cảm ơn tới thầy **ThS. Nguyễn Đức Lưu** đã tận tình hướng dẫn, chỉ bảo và giúp đỡ em trong suốt quá trình thực hiện khóa luận này.

Em đã cố gắng rất nhiều để thực hiện đề tài một cách tốt nhất. Em rất mong nhận được sự góp ý của các thầy cô để hoàn thiện đồ án này tốt hơn.

*Em xin chân thành cảm ơn!*

Sinh viên thực hiện

Nguyễn Hữu Tăng

MỤC LỤC

[LỜI CẢM ƠN 5](#_Toc134651762)

[MỤC LỤC 6](#_Toc134651763)

[DANH MỤC HÌNH ẢNH 8](#_Toc134651764)

[MỞ ĐẦU 11](#_Toc134651765)

[1. Lý do chọn đề tài 11](#_Toc134651766)

[2. Mục đích đề tài 12](#_Toc134651767)

[3. Bố cục đề tài 12](#_Toc134651768)

[4. Phương pháp thực hiện 12](#_Toc134651769)

[**CHƯƠNG 1:** LẬP TRÌNH C# TRONG UNITY 13](#_Toc134651770)

[1.1. Kịch bản (Scripting) trong Unity 13](#_Toc134651771)

[1.2. Ngôn ngữ được sử dung trong Unity 14](#_Toc134651772)

[1.3. Biến, Hàm, Lớp và cách sử dụng 15](#_Toc134651773)

[1.4. GameObject – Thao tác với C# Script 23](#_Toc134651774)

[**CHƯƠNG 2:** LẬP TRÌNH GAME UNITY 30](#_Toc134651775)

[2.1. Tổng quan về Unity 30](#_Toc134651776)

[2.2. Các thành phần cơ bản trong Unity 32](#_Toc134651777)

[2.3. Một số bước lập trình game 2D với Unity 38](#_Toc134651778)

[**CHƯƠNG 3:** CÀI ĐẶT ỨNG DỤNG 42](#_Toc134651779)

[3.1. Giời thiệu 42](#_Toc134651780)

[3.2. Một số Scene trong Game 42](#_Toc134651781)

[3.3. Nhân vật Player 49](#_Toc134651782)

[3.4. Một số thành phần trong game 59](#_Toc134651783)

[KẾT LUẬN 62](#_Toc134651784)

[1. Vấn đề đạt được 62](#_Toc134651785)

[2. Hạn chế 62](#_Toc134651786)

[3. Hướng phát triển 62](#_Toc134651787)

[TÀI LIỆU THAM KHẢO 63](#_Toc134651788)

DANH MỤC HÌNH ẢNH

[Hình 1: Một số mã code trong Unity 14](#_Toc134651789)

[Hình 2: Ví dụ về biến (Variables) 16](#_Toc134651790)

[Hình 3: Ví dụ các loại của biến 17](#_Toc134651791)

[Hình 4: Hiển thị các biến public trong Inspector 18](#_Toc134651792)

[Hình 5: Một số hàm cơ bản 19](#_Toc134651793)

[Hình 6: Ví dụ về hàm 20](#_Toc134651794)

[Hình 7: Ví dụ về lớp 21](#_Toc134651795)

[Hình 8: Ví dụ về lớp 22](#_Toc134651796)

[Hình 9: Tạo Project 32](#_Toc134651797)

[Hình 10: Tạo project 32](#_Toc134651798)

[Hình 11: Định dạng file 34](#_Toc134651799)

[Hình 12: Tạo Folder 35](#_Toc134651800)

[Hình 13: Tạo Scene 36](#_Toc134651801)

[Hình 14: Viết kịch bản game 39](#_Toc134651802)

[Hình 15: Ví dụ về đồ họa 2D trong Unity 39](#_Toc134651803)

[Hình 16: Màn hình Logo 43](#_Toc134651804)

[Hình 17: Màn hình Loading 43](#_Toc134651805)

[Hình 18: Màn hình Loading 44](#_Toc134651806)

[Hình 19: Màn hình Menu 44](#_Toc134651807)

[Hình 20: Một số trang bị trong game 45](#_Toc134651808)

[Hình 21: Shop trang bị 45](#_Toc134651809)

[Hình 22: Shop trang bị 46](#_Toc134651810)

[Hình 23: Màn hình Map 1 46](#_Toc134651811)

[Hình 24: Màn hình Map 2 47](#_Toc134651812)

[Hình 25: Màn hình Map 3 47](#_Toc134651813)

[Hình 26: Màn hình Pause 48](#_Toc134651814)

[Hình 27:Màn hình Game Over 48](#_Toc134651815)

[Hình 28: Nhân vật Player 49](#_Toc134651816)

[Hình 29: Player 49](#_Toc134651817)

[Hình 30: Component Player 50](#_Toc134651818)

[Hình 31: Skiill và Coins 51](#_Toc134651819)

[Hình 32: Skill 1: Đánh thường 52](#_Toc134651820)

[Hình 33: Skill 2: Cầu lửa 52](#_Toc134651821)

[Hình 34: Skill 3: Lửa bùng nổ 52](#_Toc134651822)

[Hình 35: Moving Player 53](#_Toc134651823)

[Hình 36: Change Health 54](#_Toc134651824)

[Hình 37: Animation 55](#_Toc134651825)

[Hình 38: Parameters 56](#_Toc134651826)

[Hình 39: Tạo Blend Tree 56](#_Toc134651827)

[Hình 40: Blend Tree 57](#_Toc134651828)

[Hình 41: Script Animation 58](#_Toc134651829)

[Hình 42: Một số NPC 59](#_Toc134651830)

[Hình 43: Sound 59](#_Toc134651831)

[Hình 44: Quái và Boss 59](#_Toc134651832)

[Hình 45: Enemy Moving 60](#_Toc134651833)

[Hình 46: Enemy Skill 61](#_Toc134651834)

MỞ ĐẦU

1. Lý do chọn đề tài

Báo cáo của Newzoo tính đến tháng 11.2022 cho thấy quy mô ngành game toàn cầu đạt 184,4 tỉ USD, với doanh thu ngành game di động ước tính đạt 92,2 tỉ USD và chiếm khoảng 50% thị phần. Trong số này, một số khu vực nổi bật có tốc độ tăng trưởng thị trường game di động nhanh trong năm 2022 gồm Mỹ La tinh, Trung Đông, Đài Loan & Hồng Kong, Đông Nam Á. Doanh thu game di động tại thị trường Đông Nam Á đạt 4,5 tỉ USD, chỉ sau Mỹ La tinh.

Mặc dù xếp thứ ba về doanh thu trong năm 2022 khi đạt 507 triệu USD từ 54,6 triệu người chơi, nhưng quy mô doanh thu game di động ở thị trường Việt Nam vẫn còn duy trì khoảng cách khá xa so với Thái Lan - vốn đạt doanh thu 763 triệu USD với 38,3 triệu người chơi.

Nếu nhìn vào số dân và lượng tải game ở Việt Nam có thể thấy trung bình số lượng người chơi game ở Việt Nam là khá lớn. Việt Nam có nội lực và tiềm năng cần khai phá trong lĩnh vực game.

Thị trường game Việt Nam vẫn còn nhỏ bé và chưa được nhiều tập đoàn lớn quan tâm. Tuy nhiên, Việt Nam đang trong thời điểm chín muồi để khai thác tiềm năng phát triển cả về khách hàng và lực lượng công nghệ game phát triển mạnh mẽ, trong đó có lĩnh vực thể thao điện tử. Thể thao điện tử đang được cộng đồng đánh rất cao. Game không chỉ là trò chơi giải trí thông thường mà còn có tính thể thao.

Chính vì lý do đó mà em quyết định chọn đề tài***“Xây dựng ứng dụng game 2D Sword Hunter bằng Unity”***

2. Mục đích đề tài

Đề tài ***“Xây dựng ứng dụng game 2D Sword Hunter bằng Unity”*** nhằm xây dựng một ứng dụng game 2D dựa trên công cụ có sẵn là Unity.

3. Phạm vi nghiên cứu

Đề tài tập trung nghiên cứu tìm hiểu về thị trường ngành công nghiệp Game hiện nay đồng thời cài đặt thử nghiệm xây dựng một sản phẩm Game bằng Unity cùng với các công cụ hỗ trợ khác trong việc lập trình Game.

4. Bố cục đề tài

Nội dung chính được chia làm 3 chương:

**Chương 1: Giới thiệu tổng quan về thị trường Game:** Chương này sẽ giới thiệu tổng quan về thị trường Game hiện nay và các công nghệ, công cụ cần thiết để xây dựng một sản phẩm Game

**Chương 2: Phân tích và thiết kế hệ thống trò chơi:** Chương này phân tích, thiết kế hệ thống Game ***“Sword Hunter”***.

**Chương 3: Xây dựng và cài đặt chương trình:** Chương này sẽ đưa ra chi tiết về nội dung của ứng dụng trò chơi ***“Sword Hunter”***. Trình bày các kết quả thu được trong quá trình nghiên cứu lần này.

5. Phương pháp thực hiện

* Thu thập tài liệu
* Tìm hiểu, phân tích, chọn lọc, tổng hợp, thống kê.
* Lập trình và kiểm thử.

1. TỔNG QUAN THỊ TRƯỜNG GAME
   1. Giới thiệu thị trường Game

Game là một ứng dụng phần mềm giải trí được thiết kế để cung cấp cho người chơi một trải nghiệm giải trí và thư giãn. Trong thế giới hiện đại, Game không chỉ đơn thuần là một phương tiện giải trí, mà còn là một lĩnh vực kinh doanh có giá trị thương mại cao. Những năm gần đây, ngành công nghiệp Game và thị trường Game đã phát triển nhanh chóng và đầy biến động, đặc biệt là với sự xuất hiện của các thiết bị di động và kết nối Internet đang ngày càng phổ biến.

Thị trường Game hiện nay là một ngành công nghiệp phát triển mạnh mẽ và đa dạng. Với sự phổ biến của các nền tảng chơi Game di động, Console và PC. Người chơi trên toàn cầu có nhiều lựa chọn hơn bao giờ hết.

* 1. Điểm nổi bật của thị trường Game

Dưới đây là một tổng quan về thị trường Game hiện nay:

* **Kích thước thị trường:** Gần 3,2 tỷ người sẽ chơi trò chơi vào năm 2022, chi tiêu tổng cộng 196,8 tỷ đô la. Và dự báo cho thấy thị trường sẽ tiếp tục tăng trưởng trong những năm tiếp theo. Thể thao điện tử sẽ tạo ra doanh thu gần 1,38 tỷ đô la trên toàn cầu vào cuối năm 2022. Trung Quốc chiếm gần một phần ba doanh thu thể thao điện tử trên toàn thế giới.
* **Đa dạng hóa nền tảng**: Ngày nay, Game không chỉ xuất hiện trên các máy chơi Game Console và PC, mà còn trên điện thoại di động, máy tính bảng và các thiết bị khác. Thị trường Game di động đặc biệt đang trở thành một phần quan trọng với số lượng lớn người chơi và doanh thu đáng kể.
* **Phân loại**: Thị trường Game hiện nay đa dạng với nhiều thể loại khác nhau như hành động, phiêu lưu, chiến thuật, nhập vai, thể thao,...
* **Sự thay đổi trong mô hình kinh doanh**: Cùng với sự phát triển của Game miễn phí và Game tính phí, mô hình kinh doanh trong ngành Game cũng đang thay đổi. Ngoài doanh thu từ việc bán Game và DLC (nội dung tải về), các nhà phát triển còn tìm kiếm các nguồn thu khác như quảng cáo trong Game, giao dịch trong Game và dịch vụ liên quan đến Game.
* **Công nghệ mới**: Công nghệ như thực tế ảo (VR), thực tế tăng cường (AR) và trò chơi điện tử thể thao (eSports) đang thúc đẩy sự phát triển và đổi mới trong ngành Game. Các công nghệ này mở ra những trải nghiệm mới cho người chơi và tạo ra cơ hội kinh doanh mới cho các nhà phát triển Game.
  1. Công nghệ và công cụ sử dụng

Game Engine (Công cụ game) là một phần mềm dùng để xây dựng và thiết kế các trò chơi điện tử. Các nhà lập trình game phải kết hợp các ngôn ngữ lập trình và các công cụ game phù hợp để có thể tạo ra một sản phẩm game hoàn chỉnh.

Hiện nay, trên thế giới có rất nhiều công cụ game nổi tiếng như Unreal Engine, CryEngine, id Tech, Source, Godot,.. Trong số đó, không thể kể đến cái tên nổi bật Unity đã và đang chiếm ưu thế khá lớn trong ngành lập trình Game hiện nay.

* + 1. Giới thiệu Unity

Unity là một công cụ trò chơi đa nền tảng được phát triển bởi Unity Technologies, chủ yếu được sử dụng để phát triển trò chơi điện tử và mô phỏng cho máy tính, bảng điều khiển và thiết bị di động.

Với các tính năng đa nền tảng, Unity phổ biến với cả các nhà phát triển game tự do và trong các studio game. Nó được sử dụng để tạo các trò chơi như Pokemon Go, Heathstone, Rimworld, Cuphead và nhiều trò chơi khác.

Lập trình Unity 2D và 3D được lập trình dựa vào 3 ngôn ngữ chính là C#, Boo và UnityScript. Thông thường, ngôn ngữ chính mà lập trình viên Unity sử dụng phổ biến nhất hiện nay là C#.

Bởi vì sự có mặt lâu từ năm 2005, Unity đã phát triển một lượng lớn người dùng và một thư viện tài nguyên khổng lồ. Unity không chỉ có tài liệu tuyệt vời, mà còn có vô số video và hướng dẫn trực tuyến đáng kinh ngạc cho người dùng.

Chính vì vậy, Unity là một sự lựa chọn thông minh cho người mới bắt đầu tiếp cận với các công cụ game. Nằm trong danh sách các công cụ trò chơi điện tử , Unity đóng vai trò như một cổng thông tin kiến ​​thức và tài nguyên được xây dựng chỉ dựa trên cộng đồng rộng lớn của họ.

**Ưu điểm khi sử dụng Unity**

* Đối với người mới bắt đầu, Unity là một công cụ toàn diện tốt có thể đáp ứng hầu hết các nhu cầu của bất kỳ ai. Unity cũng tuyệt vời cho các trò chơi tạo mẫu.
* Lý do chính để chọn Unity là thư viện tài nguyên khổng lồ dành cho mọi người. Ngay cả những nhà phát triển có kinh nghiệm cũng có thể tiết kiệm thời gian và học hỏi được nhiều điều từ cộng đồng.
* Unity cũng cung cấp một bộ công cụ dựa trên điện toán đám mây (Cloud Computing) mạnh mẽ để dễ dàng kiếm tiền từ trò chơi và thêm khả năng nhiều người chơi.
* Với Unity Analytics , Unity Ads, Unity Collaborate và Unity Multiplayer, người dùng có quyền truy cập vào một bộ công cụ để tạo các trò chơi động (dynamic games). Rất ít công cụ trò chơi khác cung cấp nhiều tính năng tập trung này.
  + 1. Ngôn ngữ sử dụng trong Unity

Một Tập lệnh (Script) phải được đính kèm vào một GameObject trong cảnh (scene) để được gọi ra bởi Unity. Các tập lệnh được viết bằng một ngôn ngữ đặc biệt mà Unity có thể hiểu được. Và, thông qua ngôn ngữ này, chúng ta có thể nói chuyện với máy tính và đưa ra hướng dẫn cho nó hiểu.

Ngôn ngữ được sử dụng trong Unity được gọi là C# (phát âm là C-sharp). Tất cả các ngôn ngữ mà Unity hoạt động đều là ngôn ngữ kịch bản hướng đối tượng. Giống như bất kỳ ngôn ngữ nào, ngôn ngữ kịch bản có cú pháp hoặc các phần của ngôn ngữ và các phần chính được gọi là biến, hàm và lớp (variables, functions, and classes).

Các phiên bản Unity cho đến trước 2017.3, sẽ có một trình soạn thảo văn bản có tên là MonoDevelop: nó có thể giúp chúng ta hoàn thành mã của mình, nó sẽ cho chúng ta biết nếu chúng ta viết sai một đoạn mã tại ví trị cụ thể, và cho phép chúng ta đi đường tắt đến vị trí sai đó. Kể từ phiên bản 2018.1, cũng có thể sử dụng trình soạn thảo mã như Visual Studio cho Unity Community hoặc các trình soạn thảo văn bản khác như Visual Studio, Notepad hoặc Sublime text.

* + 1. Một số bước trong lập trình Game 2D với Unity

**Các nguyên tắc cơ bản**

* GameObjects: Là các đối tượng cơ bản trong Unity đại diện cho các nhân vật, đạo cụ, phong cảnh, v.v. Mọi đối tượng trong trò chơi là một GameObject.

GameObjects đại diện cho các mục trong trò chơi; không gian mà đặt chúng để xây dựng cấp độ được gọi là bối cảnh. Khi lập trình Game 2D, có thể bỏ qua trục thứ 3 (trục z).

Hành vi của GameObjects được xác định bởi các khối chức năng được gọi là các thành phần. Các thành phần sau là cơ bản cho trò chơi 2D:

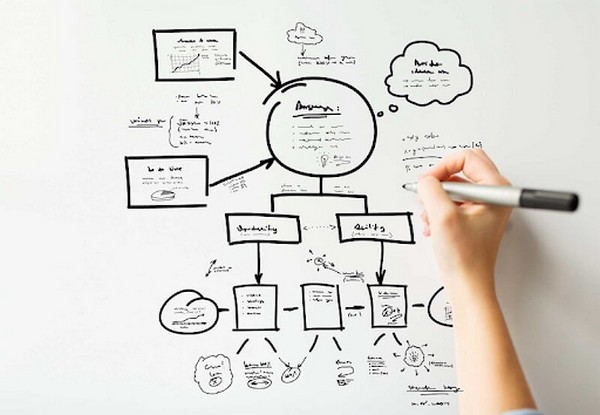
* Transform: thành phần Transform sẽ xác định Vị trí, Xoay và Tỷ lệ của từng GameObject trong cảnh. Mỗi GameObject đều có một thành phần Transform.
* Sprite Renderer: thành phần Sprite Renderer kết xuất Sprite và điều khiển nó trông như thế nào trong một cảnh.
* Máy ảnh: thiết bị thu và hiển thị thế giới cho người chơi. Đánh dấu Máy ảnh là Orthographic sẽ xóa tất cả góc nhìn khỏi chế độ xem của Máy ảnh. Điều này chủ yếu hữu ích cho việc tạo các trò chơi isometric hoặc 2D.
* Collider 2D: thành phần này xác định hình dạng của GameObject 2D cho các mục đích va chạm vật lý

Các thành phần là các đại diện giao diện người dùng của các lớp C #; có thể sử dụng các tập lệnh để thay đổi và tương tác với các thành phần hoặc tạo các tập lệnh mới.

**Viết kịch bản Game**

Tất cả các trò chơi 2D đều cần kịch bản. Các kịch bản phản hồi thông tin đầu vào từ người chơi và sắp xếp các sự kiện trong trò chơi xảy ra khi cần thiết.

Kịch bản được đính kèm với GameObjects và bất kỳ kịch bản nào tạo ra đều kế thừa từ lớp MonoBehaviour.



Hình 1: Viết kịch bản game

**Sprites**

Sprites là các đối tượng đồ họa 2D. Sử dụng Sprites cho tất cả các loại trò chơi 2D. Ví dụ: có thể nhập hình ảnh của nhân vật chính dưới dạng Sprite.



Hình 2: Ví dụ về đồ họa 2D trong Unity

Có thể sử dụng một bộ sưu tập Sprites để xây dựng một nhân vật. Điều này cho phép kiểm soát tốt hơn chuyển động và hoạt ảnh của các nhân vật.

**Xây dựng môi trường trong trò chơi**

Thiết kế môi trường đề cập đến quá trình xây dựng các cấp độ và môi trường trong trò chơi. Có thể kết hợp các công cụ thiết kế môi trường trong phần này theo bất kỳ cách nào phù hợp nhất với trò chơi.

**Hoạt hình nhân vật**

Có ba cách khác nhau để có thể tạo hoạt ảnh cho các nhân vật 2D:

|  |  |
| --- | --- |
| **Loại hoạt hình 2D** | **Mục đích** |
| **Frame-by-frame** | Lý do nghệ thuật, nếu muốn trò chơi của mình có phong cách nghệ thuật hoạt hình cổ điển. Hoạt ảnh từng khung hình tương đối tốn nhiều tài nguyên, cả để tạo và chạy. |
| **Cutout** | Hình ảnh động khung xương mượt mà, khi các nhân vật không yêu cầu khớp nối thực tế. |
| **Skeletal** | Hoạt ảnh khung xương mượt mà trong đó Sprites uốn cong theo cấu trúc xương. Sử dụng điều này khi các nhân vật cần một cảm giác hữu cơ hơn. |

**Đồ họa**

Phần này mô tả các tùy chọn đồ họa khi sử dụng UniversalRender Pipeline (URP) – Một đường dẫn kết xuất thực hiện một loạt các hoạt động lấy nội dung của Bối cảnh và hiển thị chúng trên màn hình.

**Vật lý 2D**

Cài đặt Vật lý 2D xác định giới hạn về độ chính xác của mô phỏng vật lý trong trò chơi 2D.

**Âm thanh**

Có thể thêm nhạc nền và hiệu ứng âm thanh vào trò chơi trong Unity. Sử dụng phần mềm của bên thứ ba để tạo âm thanh và nhập nó vào Unity với các cài đặt được đề xuất.

**Giao diện người dùng**

Nếu muốn thêm menu hoặc trợ giúp vào trò chơi, cần thiết lập giao diện người dùng. Để thiết lập giao diện người dùng, hãy sử dụng Unity UI

**Lập hồ sơ, tối ưu hóa và thử nghiệm**

Việc lập hồ sơ cho phép xem các phần khác nhau trong trò chơi sử dụng nhiều tài nguyên như thế nào. Nên luôn lập hồ sơ trò chơi trên nền tảng phát hành mục tiêu của nó.

Sau khi lập hồ sơ, sử dụng kết quả để cải thiện và tối ưu hóa hiệu suất.

Kiểm tra trò chơi và mã với Khung kiểm tra thống nhất.

**Xuất bản**

Khi đã hoàn thành, đã sẵn sàng xuất bản trò chơi.

Đây chỉ là các bước cơ bản để **lập trình game 2D bằng Unity.**

Tóm lại, thị trường Game hiện nay là một ngành công nghiệp phát triển mạnh mẽ và đa dạng. Với kích thước lớn và doanh thu đáng kể, ngành Game thu hút sự quan tâm của nhiều nhà phát triển và người chơi trên toàn thế giới. Thị trường Game đa dạng với nhiều thể loại và nền tảng chơi Game. Sự phổ biến của Game di động và xu hướng mới như VR, AR và eSports đang thúc đẩy sự đổi mới và tạo ra cơ hội kinh doanh mới. Thị trường Game hiện nay không chỉ tạo ra giải trí cho người chơi mà còn mang lại lợi ích kinh tế đáng kể cho các nhà phát triển và nhà phân phối Game. Unity rất linh hoạt và có thể được sử dụng để tạo bất kỳ loại trò chơi điện tử nào, vì vậy đây là một nền tảng tuyệt vời cho các nhà lập trình game mới bắt đầu hoặc các nhà phát triển muốn thử xây dựng trò chơi mẫu cho mình.

1. PHÂN TÍCH THIẾT KẾ HỆ THỐNG
   1. Phân tích và thiết kế hệ thống Game

Phân tích thiết kế hệ thống (System Analysis and Design) là quá trình xác định và hiểu rõ các yêu cầu, chức năng và cấu trúc của hệ thống trong quá trình phát triển phần mềm hoặc ứng dụng.

Trong lĩnh vực Game, phân tích thiết kế hệ thống đóng vai trò quan trọng để xác định cấu trúc tổ chức và cách thức hoạt động của Game. Quá trình này bao gồm các bước sau:

* **Xác định yêu cầu hệ thống:** Đầu tiên, nhà phát triển phải tìm hiểu và xác định rõ yêu cầu của hệ thống Game, bao gồm chức năng, hiệu năng, giao diện người dùng, tính bảo mật và các yêu cầu khác từ phía người dùng.
* **Thiết kế kiến trúc hệ thống:** Sau khi xác định yêu cầu, nhà phát triển sẽ tạo ra một bản thiết kế kiến trúc hệ thống Game. Điều này bao gồm xác định các thành phần chính của hệ thống, quyết định cách chúng tương tác với nhau và thiết kế cấu trúc dữ liệu phù hợp.
* **Thiết kế cơ sở dữ liệu:** Trong một số trò chơi, việc quản lý dữ liệu là một phần quan trọng của hệ thống. Nhà phát triển sẽ thiết kế cơ sở dữ liệu để lưu trữ thông tin về người chơi, màn chơi, điểm số và các yếu tố khác.
* **Xác định luồng công việc:** Nhà phát triển sẽ xác định các bước, quy trình và luồng công việc trong hệ thống Game. Điều này bao gồm xác định các trạng thái, sự kiện và hành động của người chơi để đảm bảo trải nghiệm chơi Game liền mạch.
* **Xác định giao diện người dùng:** Giao diện người dùng là một phần quan trọng của hệ thống Game. Nhà phát triển sẽ thiết kế các giao diện người dùng thân thiện, dễ sử dụng và hợp lý để người chơi có thể tương tác và điều khiển Game một cách dễ dàng.

Phân tích thiết kế hệ thống giúp nhà phát triển có cái nhìn toàn diện về cấu trúc và hoạt động của hệ thống. Quá trình này mang lại các lợi ích sau:

* **Hiểu rõ yêu cầu:** Phân tích thiết kế hệ thống giúp nhà phát triển hiểu rõ yêu cầu của hệ thống Game. Điều này đảm bảo rằng các chức năng và tính năng cần thiết được xác định một cách chính xác và đáng tin cậy.
* **Xác định cấu trúc hệ thống:** Qua quá trình phân tích thiết kế, nhà phát triển có thể xác định cấu trúc tổ chức của hệ thống Game. Điều này bao gồm các thành phần chính, quan hệ giữa chúng và cách chúng hoạt động cùng nhau.
* **Đảm bảo tính mô-đun và tái sử dụng:** Phân tích thiết kế hệ thống giúp phân tách hệ thống thành các mô-đun độc lập và tái sử dụng được. Điều này tạo điều kiện cho việc phát triển linh hoạt, nhanh chóng và dễ dàng mở rộng hệ thống Game trong tương lai.
* **Đảm bảo tính linh hoạt và mở rộng:** Phân tích thiết kế hệ thống giúp nhà phát triển dự đoán và chuẩn bị cho các nhu cầu mở rộng và phát triển trong tương lai. Điều này đảm bảo rằng hệ thống Game có thể linh hoạt thay đổi, thích ứng và mở rộng để đáp ứng nhu cầu của người chơi và thị trường.
* **Đảm bảo hiệu suất và tối ưu hóa:** Phân tích thiết kế hệ thống giúp nhà phát triển định rõ các yêu cầu về hiệu suất và tối ưu hóa. Điều này đảm bảo rằng hệ thống Game chạy mượt mà, không bị giật lag và đáp ứng tốt các yêu cầu về tốc độ xử lý và bộ nhớ.
* **Đảm bảo trải nghiệm người dùng tốt:** Phân tích thiết kế hệ thống giúp đảm bảo rằng trải nghiệm người chơi được đặt lên hàng đầu.
  1. Phân tích và thiết kế hệ thống trong Game Sword Hunter

**Các chức năng trong Game:**

* Di chuyển và né tránh sự tấn công của quái vật
* Tấn công quái vật bằng các kỹ năng
* Mua trang bị
* Sử dụng trang bị
* Bật tắt âm thanh Game

**Hệ thống điều khiển nhân vật:**  Người dùng sẽ sử dụng các phím mũi tên để di chuyển nhân vật.

**Hệ thống cửa hàng trong Game:** Game xây dựng cửa hàng có thể mua 2 loại trang bị là vũ khí và giáp. Trong cửa hàng xây dựng 3 loại vũ khí và 3 loại giáp. Trong giao diện cửa hàng gồm có: hình ảnh vũ khí, nút mua vật phẩm hiển thị cùng các chỉ số và giá của trang bị, số lượng coins hiện có, nút thoát khỏi của hàng, nút trang bị vũ khí.

**Giao diện người dùng:** Giao diện người dùng được xây dựng lên bằng các công cụ phần mềm của Unity nên giao diện người dùng khá thân thiện và dễ dàng sử dụng.

**Tính năng Single-Player:** Chế độ một người chơi.

* 1. Biểu đồ Use case
     1. Biểu đồ Use-case tổng quan

Ảnh có chứa biểu đồ, hàng, vòng tròn, văn bản

Mô tả được tạo tự động

Hình 3: Biểu đồ Use-case tổng quan

Biểu đồ Use case tổng quan bao gồm 1 tác nhân: Người chơi. Người chơi cần chọn map, di chuyển và tấn công quái vật. Sau khi quái vật bị tiêu diệt sẽ nhận được coins người chơi sẽ mua các trang bị. Người chơi sẽ sử dụng trang bị để tăng chỉ số của nhân vật. Trong màn chơi có thể dừng trò chơi và bật/tắt âm thanh.

* + 1. Use case phân ra gameplay

Ảnh có chứa văn bản, biểu đồ, bản đồ

Mô tả được tạo tự động

Hình 4: Biểu đồ Use case phân rã Gameplay

Người chơi có thể chọn map, di chuyển nhân vật, tấn công quái vật, tạm dừng trò chơi và tiếp tục trò chơi.

* + 1. Use case phân rã mua trang bị

Ảnh có chứa hàng, biểu đồ, vòng tròn, Sơ đồ

Mô tả được tạo tự động

Hình 5: Biểu đồ Use case phân rã mua trang bi

Người chơi mở shop trang bị, xem thông tin trang bị và mua trang bị

* + 1. Use case phân rã sử dụng trang bị

Ảnh có chứa hàng, biểu đồ, vòng tròn, Sơ đồ

Mô tả được tạo tự động

Hình 6: Biểu đồ Use case phân rã sử dụng trang bị

Người chơi xem thông tin trang bị và sử dụng trang bị.

* 1. Mô tả chi tiết Use case
     1. Mô tả use case GamePlay

Use case này cho phép Người chơi chơi game.

**Luồng sự kiện:**

* Luồng cơ bản:

1. Use case này bắt đầu khi người chơi chọn map trên màn hình Menu. Hiển thị lên màn hình map.
2. Chơi game: Người chơi di chuyển nhân vật, tiêu diệt quái vật trên màn hình chơi game.
3. Người chơi chọn tạm dừng game trên màn hình chơi game. Hiển thị màn hình tạm dừng.
4. Người chơi chọn bật/tắt âm thanh trên màn hình tạm dừng.
5. Hệ thống bật/tắt âm thanh trò chơi.
6. Người chơi chọn tiếp tục game trên màn hình tạm dừng. Hiển thị lên màn hình chơi game.
7. Người chơi tiêu diệt quái vật.
8. Hệ thống thêm coins cho nhân vật, hiển thị lên màn hình.

Use case kết thúc.

* Luồng rẽ nhánh:

**Các yêu cầu đặc biệt**

Không có.

**Tiền điều kiện**

Người chơi cần mở trò chơi và chọn map.

**Hậu điều kiện**

Không có.

**Điểm mở rộng**

Không có.

* + 1. Mô tả Use case mua trang bị

Use case này cho phép Người chơi mua trang bị cho nhân vật.

**Luồng sự kiện:**

* Luồng cơ bản:

1. Use case này bắt đầu khi người chơi chọn shop trang bị trên màn hình Menu. Hiển thị lên màn hình shop trang bị.
2. Người chơi chọn trang bị trên màn hình shop. Hiển thị thông tin trang bị: chỉ số trang bị, giá mua trang, nút mua trang bị lên màn hình.
3. Người chơi chọn mua trang bị trên màn hình.
4. Hệ thống kiểm tra coins của nhân vật có đủ không. Hiển thị màn hình Menu.

Use case kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 4 trong luồng cơ bản, nếu nhân vật không đủ coins, hệ thống hiển thị thông báo “Coins không đủ”, nút bấm trở về shop trang bị, ngược lại use case tiếp tục.

**Các yêu cầu đặc biệt**

Không có.

**Tiền điều kiện**

Người chơi cần mở trò chơi.

**Hậu điều kiện**

Không có.

**Điểm mở rộng**

Không có.

* + 1. Mô tả Use case sử dụng trang bị

Use case này cho phép Người chơi mua trang bị cho nhân vật.

**Luồng sự kiện:**

* Luồng cơ bản:

1. Use case này bắt đầu khi người chơi chọn sử dụng trang bị trên màn hình Menu. Hiển thị lên màn hình sử dụng trang bị.
2. Người chơi chọn trang bị trên màn hình. Hiển thị thông tin trang bị: chỉ số trang bị, giá mua trang bị lên màn hình.
3. Hệ thống kiểm tra trang bị đã được mua hay chưa. Hiển thị nút sử dụng trang bị lên màn hình.
4. Người chơi chọn sử trang bị trên màn hình.
5. Hệ thống kiểm tra thông tin nhân vật. Hiển thị màn hình Menu.

Use case kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 3 trong luồng cơ bản, nếu người chơi chưa mua trang bị, hệ thống ẩn nút sử dụng trang bị, ngược lại use case tiếp tục.

**Các yêu cầu đặc biệt**

Không có.

**Tiền điều kiện**

Người chơi cần mở trò chơi.

**Hậu điều kiện**

Không có.

**Điểm mở rộng**

Không có.

* 1. Biểu đồ tuần tự



Hình 7: Biểu đồ tuần tự khởi động game

Người chơi sẽ khởi động game bằng cách kích đúp vào biểu tượng game, hệ thống sẽ hiển thị màn hình logo sau đó là màn hình loading, tiếp đến là màn hình Menu với các map và các trang bị.



Hình 8: Biểu đồ tuần tự mua và sử dụng trang bị

* 1. Thiết kế cơ sở dữ liệu

Để lưu trữ dữ liệu trong Game thì có rất nhiều cách nhưng trong đồ án này sử dụng PlayerPrefs để lưu trữ dữ liệu. Trong Unity, PlayerPrefs là một công cụ được cung cấp để lưu trữ và truy cập dữ liệu cục bộ của người chơi. Nó cho phép lưu trữ các giá trị dưới dạng cặp khóa-giá trị (key-value), tương tự như một bảng băm (hash table) đơn giản. PlayerPrefs sử dụng hệ thống tệp tin cục bộ của hệ điều hành để lưu trữ dữ liệu, do đó dữ liệu được lưu giữ ngay cả sau khi Game đã được tắt và khởi động lại.

Cơ chế hoạt động của PlayerPrefs trong Unity gồm các phương thức chính sau:

* SetInt, SetFloat, SetString: Các phương thức này được sử dụng để lưu trữ giá trị của một khóa (key) cụ thể.

Ví dụ: PlayerPrefs.SetInt("Score", 100) lưu trữ giá trị 100 vào khóa "Score".

* GetInt, GetFloat, GetString: Các phương thức này được sử dụng để truy cập giá trị của một khóa đã lưu trữ trước đó.

Ví dụ, Int score = PlayerPrefs.GetInt("Score") trả về giá trị của khóa "Score" đã lưu trữ.

* HasKey: Phương thức này kiểm tra xem một khóa đã tồn tại trong PlayerPrefs hay chưa.

Ví dụ, bool hasKey = PlayerPrefs.HasKey("Score") kiểm tra xem khóa "Score" đã tồn tại hay không.

* DeleteKey: Phương thức này xóa một khóa khỏi PlayerPrefs.

Ví dụ, PlayerPrefs.DeleteKey("Score") xóa khóa "Score" khỏi PlayerPrefs.

* DeleteAll: Phương thức này xóa tất cả các khóa và giá trị trong PlayerPrefs.

Ví dụ, PlayerPrefs.DeleteAll() xóa toàn bộ dữ liệu đã lưu trữ.

* 1. Giao diện người dùng

**Giao diện người dùng:** giao diện người dùng thân thiện, dễ sử dụng và hợp lý để người chơi có thể tương tác và điều khiển Game một cách dễ dàng.

**Các giao diện người dùng đã xây dựng trong Game:**

Ảnh có chứa ảnh chụp màn hình, màn hình, văn bản, Hình chữ nhật

Mô tả được tạo tự động

Hình 9: Giao diện GamePlay

Ảnh có chứa ảnh chụp màn hình, văn bản, Phần mềm đa phương tiện, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 10: Giao diện Menu

Ảnh có chứa văn bản, phù hiệu, ảnh chụp màn hình, biểu tượng

Mô tả được tạo tự động

Hình 11: Giao diện chiến thắng

Ảnh có chứa ảnh chụp màn hình, biểu tượng, màu vàng, vòng tròn

Mô tả được tạo tự động

Hình 12: Giao diện tạm dừng Game

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, đồng hồ

Mô tả được tạo tự động

Hình 13: Giao diện thua cuộc

Tóm tắt, chương 2 đã làm rõ các use case hệ thống, mô tả chi tiết các use case, xây dựng biểu đồ use case và phân tích thiết kế hệ thống để làm rõ các chức năng, từ đó dễ dàng hơn trong việc xây dựng ứng dụng.

1. CÀI ĐẶT CHƯƠNG TRÌNH
   1. Giời thiệu về chương trình
      1. Giới thiệu về Game Sword Hunter

Tên ứng dụng : **Sword Hunter**

Thể loại : Game 2D

Người phát triển : Nguyễn Hữu Tăng

Hệ điều hành : Windown

Ảnh có chứa biểu tượng, Đồ họa, Phông chữ, thiết kế

Mô tả được tạo tự động

Hình 14: Hình ảnh logo Game

Game có lối chơi khá đơn giản như sau: Điều khiển nhân vật player di chuyển trên map. Có nhiều quái vật đi lại trong thị trấn. Nhiệm vụ của người chơi là di chuyển, tránh né và sử dụng các kỹ năng để tấn công quái vật. Tiêu diệt quái vật để nhận được coin và sử dụng để mua các trang bị nâng cao sức mạnh. Có những con quái vật rất mạnh, nâng cao sức mạnh bản thân để có thể đánh bại chúng và lấy lại tự do cho thị trấn.

**Cách chơi**

* Dùng các phím mũi tên hoặc các phím A, S, D, W để di chuyển nhân vật.
* Hãy khéo léo tránh né sự tấn công của quái vật, và sử dụng các kỹ năng để tiêu diệt chúng.
* Tiêu diệt thật nhiều quái vật để mua trang bị nâng cao các chỉ số của bản thân và tiêu diệt nhiều quái vật hơn.
  + 1. Ý tưởng xây dựng

Game dựa trên cốt truyện về thị trấn yên bình đã bị xâm nhập bởi những quái vật. Hãy ngăn chặn chúng bằng sức mạnh của bản thân. Là thợ săn của võ đường danh tiếng, là anh hùng duy nhất có thể chống lại những làn sóng quái vật vô tận.

* + 1. Đồ họa và tài nguyên

Để tạo ra một sản phẩm Game hấp dẫn và thu hút người chơi, thiết kế đồ họa đóng vai trò rất quan trọng. Các nhà phát triển cần thiết kế nhân vật, cảnh quan và các yếu tố khác trong Game với chất lượng cao và phù hợp với ý tưởng ban đầu.

Trong dự án này, các tài nguyên và đồ họa chính của Game chủ yếu thông qua quá trình tìm kiếm và mua bán các tài nguyên trên mạng dưới đây có cung cấp một số trang để tìm kiếm các tài nguyên để phát triển ứng dụng Game.

* Về phần tài nguyên nhân vật và các vật thể trong Game, dưới đây là một số trang Websites có thể tham khảo:

Unity Asset Store ( <https://assetstore.unity.com/> ): Cung cấp các tài nguyên, bao gồm các gói asset, script, plugin và nhiều hơn nữa cho Unity.

Fsy.itch.io ( https://[fsy - itch.io](https://fsy.itch.io/)/): Cung cấp các Background đẹp và GUI cho Game.

Craftpix ( <https://craftpix.net/> ): Cung cấp các 2D Game Assets store & free.

Kenney ( <https://www.kenney.nl/> ): Cung cấp một vào model 3D Free và các Assets 3D và 2D khác nữa.

* Về phần tài nguyên âm thanh (Sound Effect):

Freesound.org: cung cấp kho âm thanh miễn phí và được chia sẻ bởi cộng đồng người dùng.

Soundbible.com: cung cấp hàng ngàn hiệu ứng âm thanh miễn phí.

ZapSplat.com: cung cấp âm thanh chất lượng cao và miễn phí cho các dự án tương đối nhỏ.

AudioJungle.net: cung cấp âm thanh chuyên nghiệp và có thể sử dụng cho các dự án thương mại.

Epidemic Sound: cung cấp các bản nhạc và âm thanh tùy chỉnh.

Sonniss.com: cung cấp hàng nghìn âm thanh và hiệu ứng tuyệt vời.

Dưới đây là hình ảnh mô tả các tài nguyên đã sử dụng trong Game:

Ảnh có chứa ảnh chụp màn hình, văn bản, Phông chữ

Mô tả được tạo tự động

Hình 15: Sprite Player

Ảnh có chứa ảnh chụp màn hình, Phần mềm đa phương tiện, Phần mềm đồ họa, phần mềm

Mô tả được tạo tự động

Ảnh có chứa ảnh chụp màn hình, Phần mềm đa phương tiện, Phần mềm đồ họa, phần mềm

Mô tả được tạo tự động Ảnh có chứa ảnh chụp màn hình, Phần mềm đa phương tiện, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 16: Các map trong game

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu tượng

Mô tả được tạo tự động

Hình 17: Các thành phần UI

Ảnh có chứa ảnh chụp màn hình, Phần mềm đa phương tiện, mô hình 3d, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 18: Sprite trang bị

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

Hình 19: Các File Script

Ảnh có chứa ảnh chụp màn hình, Nhiều màu sắc, màu vàng

Mô tả được tạo tự động

Hình 20: Các Sound trong Game

Ảnh có chứa ảnh chụp màn hình, văn bản, thiết kế

Mô tả được tạo tự động

Hình 21: Một số các Animation

* + 1. StoryBoard

Chart, box and whisker chart

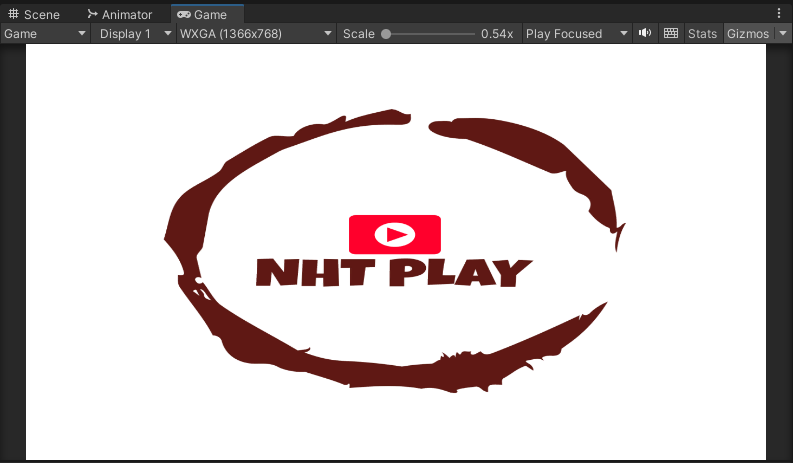
Description automatically generated

Use Weapon

Hình 22: StoryBoard của Game

* 1. Một số Scene trong Game

Khi bắt đầu dự án game xuất hiện logo của ứng dụng:



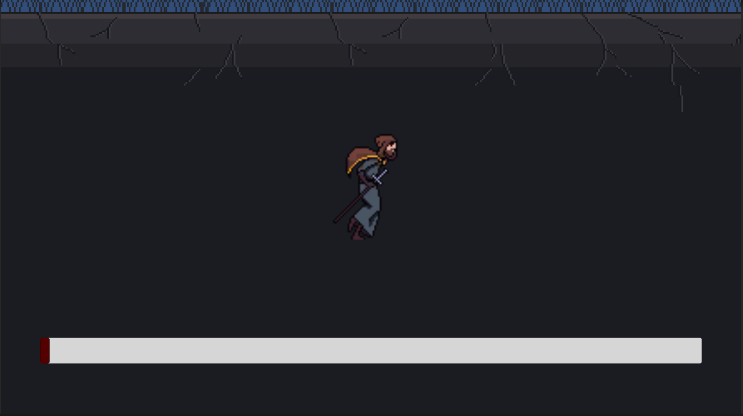
Hình 23: Màn hình Logo

Màn hình khi load game:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 24: Màn hình Loading



Hình 25: Màn hình Loading

Màn hình Menu xuất hiện sau khi loading, có thể chọn các màn chơi, chọn trang bị để sử dụng, cũng như mua các trang bị, hướng dẫn chơi và đóng ứng dụng.

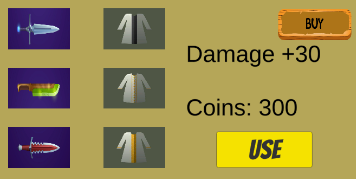
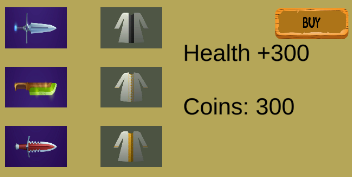
Ảnh có chứa Website

Mô tả được tạo tự động

Hình 26: Màn hình Menu

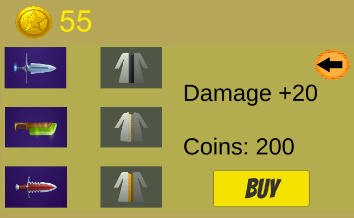
Một số trang bị trong game với các chỉ số và giá coins khác nhau. Trước khi mua trong Shop thì trang bị sẽ không hiện nút Use.

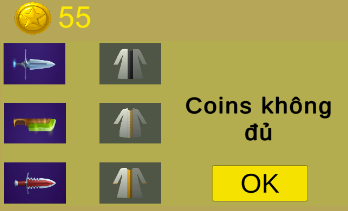
 

Hình 27: Một số trang bị trong game

Khi bấm vào nút Buy, sẽ xuất hiện Shop trang bị. Sau khi mua thành công sẽ quay lại màn hình Menu. Nếu coins không đủ xẽ xuất hiện thông báo.



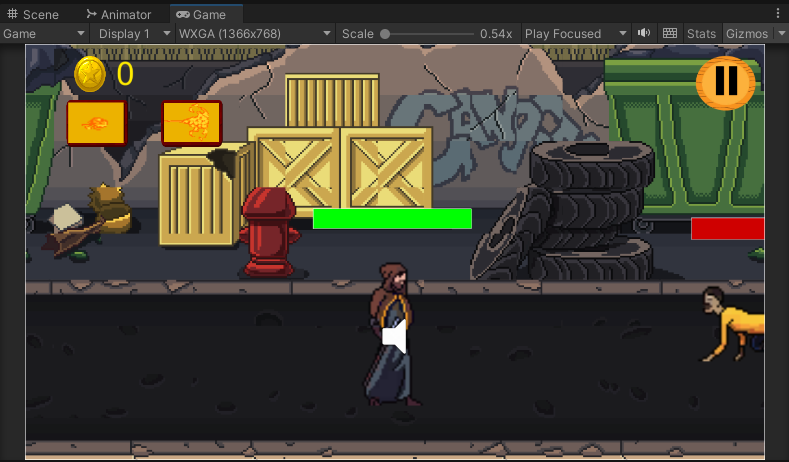
Hình 28: Shop trang bị



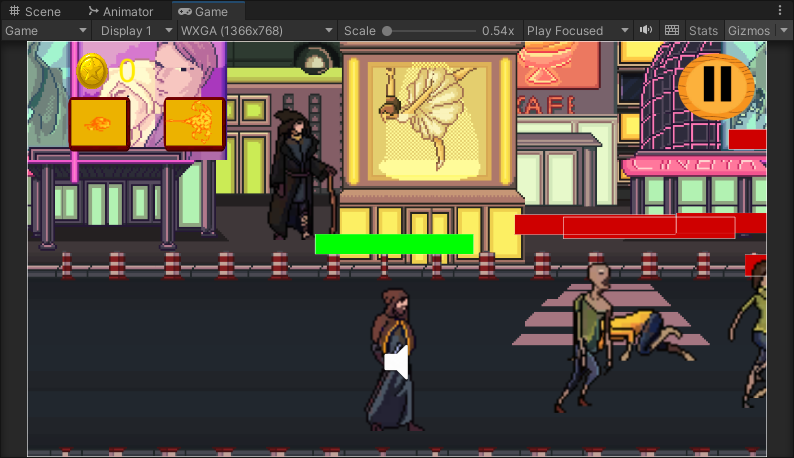
Hình 29: Shop trang bị

Nhấn Ok để quay lại Shop, muốn trở về kho trang bị thì nhấn vào mũi tên.

Một số màn hình khi chơi game:



Hình 30: Màn hình Map 1



Hình 31: Màn hình Map 2

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 32: Màn hình Map 3

Màn hình Pause xuất hiện khi bấm vào nút Ảnh có chứa văn bản

Mô tả được tạo tự động trên góc màn hình. Nút bật âm thanh và tắt âm thanh trong game, nút resume và nút trở lại Menu.



Hình 33: Màn hình Pause

Màn hình Game Over xuất hiện khi nhân vật hết máu.



Hình 34:Màn hình Game Over

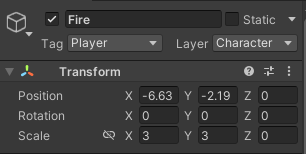
* 1. Nhân vật Player



Hình 35: Nhân vật Player

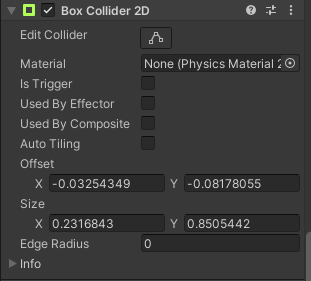
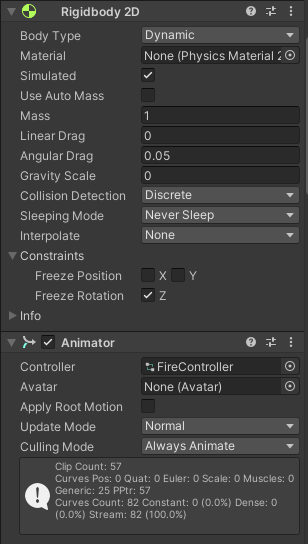
Khi tạo mới một nhân vật cần làm một số thao tác sau:

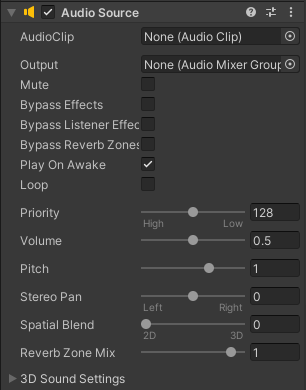
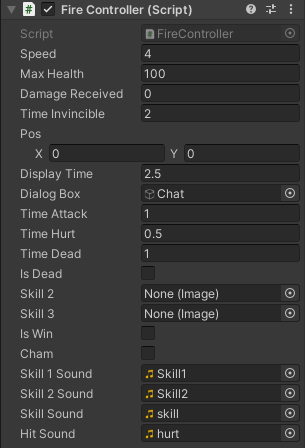
* Kéo thả hình của player vào trong Scene
* Tùy chỉnh kích thước và vị trí sao cho phù hợp
* Thêm các component như Box Collider 2D để nhận biết được sự va chạm giữa các phần trên map.
* Tạo Script để điều khiển player.
* Tạo Animation cho nhân vật chuyển động



Hình 36: Player

**Các Component có trong Inspector của Player**





Hình 37: Component Player

* + 1. Script

Script có tên là FireController.cs và được gắn vào player.

Cơ bản player sẽ có damage là 10, health là 100 và speed là 4.0f.

int health = 100;

int damage = 10;

float speed = 4.0f;

Dưới đây là một số phần cơ bản trong script:

**Skill**

Skill trong game và coins, hiển thị thời gian hồi skill và khi giết được quái vật, coins sẽ ngẫu nhiên trong một khoảng theo từng loại quái vật.

 Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 38: Skiill và Coins

Skill thường: Đánh thường, thời gian hồi chiêu 1s, Key: Space.

Skill 1: Cầu lửa, thời gian hồi chiêu 2s, Key: Q.

Skill 2: Lửa bùng nổ, thời gian hồi chiêu 3s, Key: E.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 39: Skill 1: Đánh thường

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 40: Skill 2: Cầu lửa

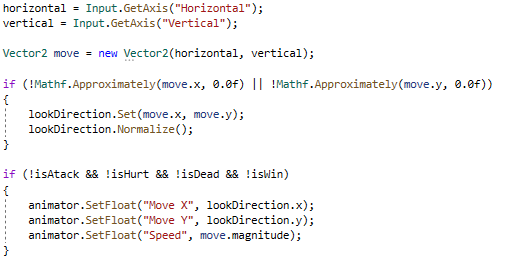
Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 41: Skill 3: Lửa bùng nổ

**Moving**

Để nhân vật có thể di chuyển cần bắt sự kiện từ bàn phím, có nhiều cách để di chuyển nhân vật trong Unity tùy theo kịch bản game.



Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 42: Moving Player

Khi nhân vật đang sử dụng skill hoặc khi nhân vật đang bị tấn công sẽ không thể di chuyển.

**Health**

Health sẽ hiển thị phía trên nhân vật. Khi quái bị quái vật tấn công máu của nhân vật sẽ bị thay đổi.

Ảnh có chứa văn bản

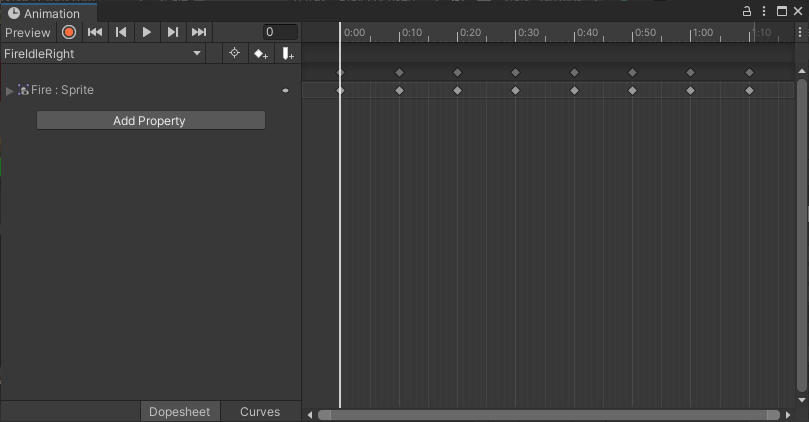
Mô tả được tạo tự động

Hình 43: Change Health

* + 1. Thay đổi trạng thái của nhân vật

**Đầu tiên, tạo một Animator Controller có tên là FireController và gán vào nhân vật.**

Chọn Window -> Animation -> Animation, xuất hiện cửa sổ Animation. Tạo các Animation Clip tương ứng theo từng hành động.



Hình 44: Animation

Trong Animator, chọn Parameters, click vào dấu +, để tạo các trạng thái của nhân vật.

Float: dùng để tạo Move x, Move y, Speed nhằm nhận biết được trạng thái khi nhân vật di chuyển.

Trigger: dùng để nhận biết các trạng thái khi bị quái vật tấn công, khi sử dụng skill và khi nhân vật chết.

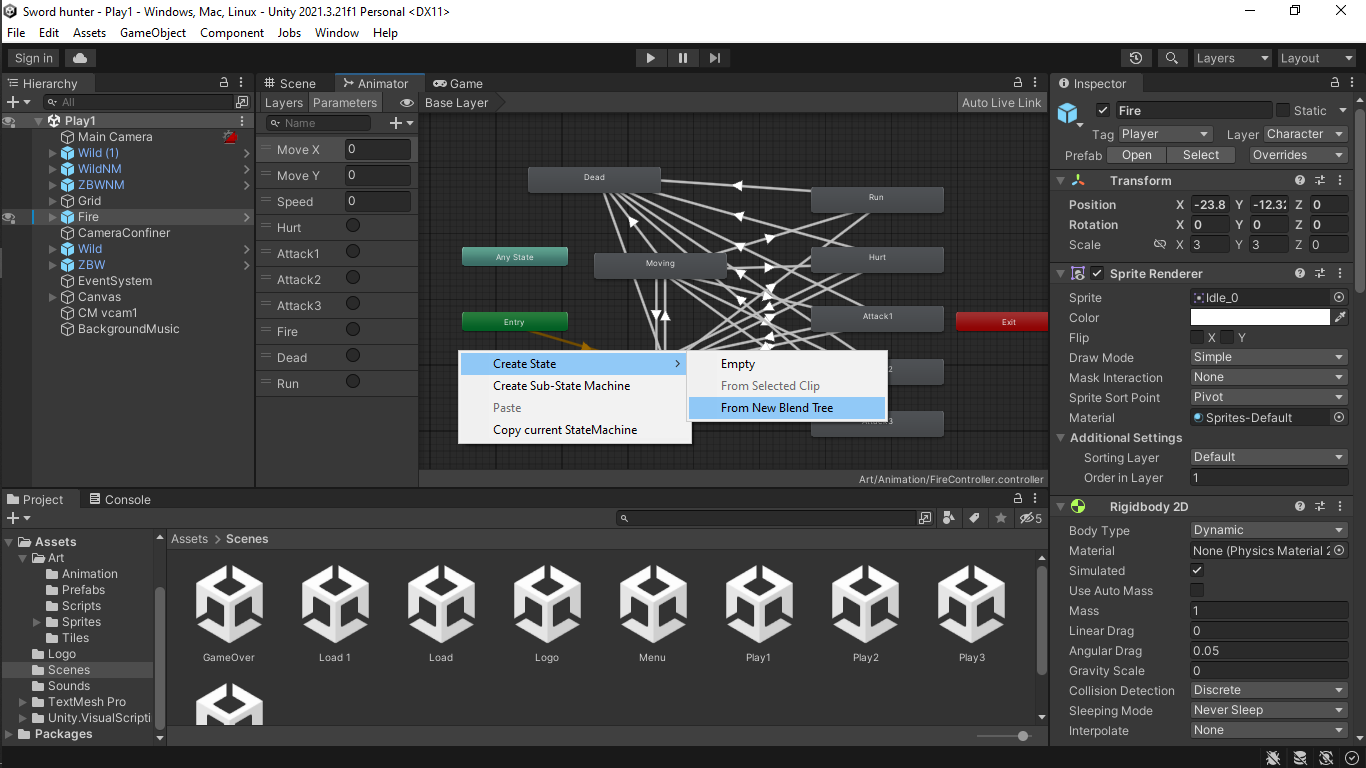
Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, màn hình

Mô tả được tạo tự động

Hình 45: Parameters

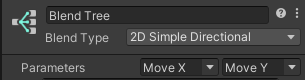
**Tiếp theo là tạo Blend Tree để kết hợp các animation clip phù hợp theo từng trạng thái.**

Đầu tiên, để tạo Blend Tree click phải chuột -> Create State -> From New Blend Tree.



Hình 46: Tạo Blend Tree

Click đúp vào Blend Tree, thay đổi type thành 2D Simple Directional, click chọn dấu + để thêm các Motion

 Ảnh có chứa văn bản, đồ điện tử, máy tính

Mô tả được tạo tự động

Hình 47: Blend Tree

**Kế tiếp, cần viết code trong Script để thay đổi trạng thái nhân vật.**

Khai báo đối tượng Animator để có thể thay đổi trạng thái nhân vật.

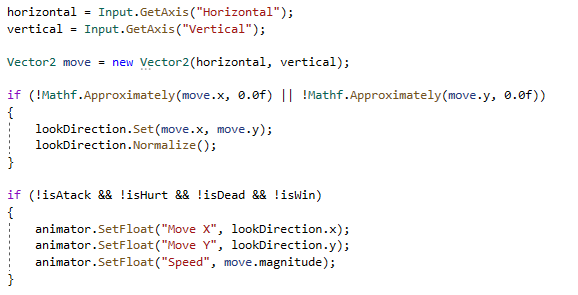
Animator animator;

Vector2 lookDirection = new Vector2(1, 0);

Trong void Start()

animator = GetComponent<Animator>();

Kế tiếp, khi nhấn phím di chuyển cần thêm câu lệnh chuyển trạng thái tương ứng.



Hình 48: Script Animation

Như vậy, bằng đối tượng Animator, chúng ta có thể thay đổi trạng thái nhân vật, kết hợp với di chuyển nhân vậy, từ đó xây dựng được nhân vật game di chuyển kèm Animation trùng khớp.

* 1. Một số thành phần trong game
     1. NPC

Ảnh có chứa văn bản, bầu trời, biển hiệu

Mô tả được tạo tự động Ảnh có chứa văn bản, bầu trời, biển hiệu

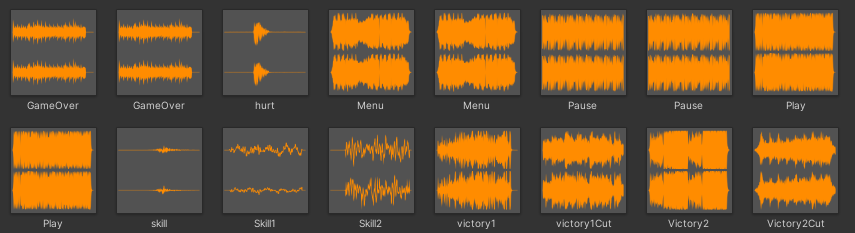
Mô tả được tạo tự động Ảnh có chứa văn bản, biển hiệu, bầu trời

Mô tả được tạo tự động

Hình 49: Một số NPC

Các NPC sẽ xuất hiện với một lời nhắc nhở khi bắt đầu màn chơi. Lời nhắc nhở về sự nguy hiểm trong từng màn chơi sẽ tăng lên.

* + 1. Sound

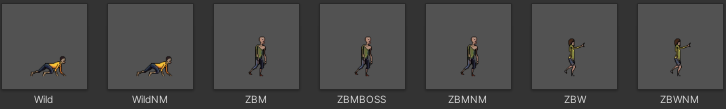


Hình 50: Sound

Sound là một thành phần không thể thiếu trong các trò chơi ví dụ như âm thanh trong màn chơi, âm thanh khi sử dụng skill hay âm thanh khi bị tấn công,..

* + 1. Quái vật

Các quái vật khác nhau sẽ có một số sự khác nhau về damage, health, speed, … Nhưng nhìn chung có một số sự tương đồng sau:



Hình 51: Quái và Boss

Quái vật sẽ di chuyển từ trái sang phải theo thời gian. Khi bị tấn công, quái vật sẽ di chuyển với tốc độ gấp đôi và đuổi theo player. Riêng boss sau khi xuất hiện sẽ di chuyển về phía player.

Ảnh có chứa văn bản

Mô tả được tạo tự động Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 52: Enemy Moving

Khi tấn công quái vật sẽ ngẫu nhiên 1 trong 2 skill.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 53: Enemy Skill

Trên đây là một số thành phần cơ bản trong game. Để hoàn thành được ứng dụng game này cần rất nhiều thành phần lớn nhỏ khác như tạo map, các canvas,…

Tóm tắt, Tại chương 3 đã đưa ra kết quả thực nghiệm và kiểm thử chức năng hệ thống, từ đó tìm ra được các lỗi, và khắc phục chất lượng của website.

KẾT LUẬN

1. Vấn đề đạt được

Thông qua quá trình xây dựng ứng dụng game 2D Sword Hunter bằng Unity, em đã vận dụng các kiến thức đã được học trên trường và kiến thức đã học được trong quá trình hoàn thiện ứng dụng, xây dựng được một ứng dụng game với các chức năng cơ bản, hoàn thiện báo cáo. Qua đó, em rút ra được nhiều kinh nghiệm cho bản thân mình hơn, chuyên môn cũng được cải thiện hơn. Bên cạnh các kiến thức chuyên môn, em còn học hỏi được nhiều kỹ năng làm việc như:

* Cách làm việc cá nhân
* Cách lên kế hoạch cho công việc
* Cách quản lý thời gian,…

2. Hạn chế

Mặc dù còn gặp một số khó khăn trong quá trình hoàn thiện sản phẩm, nhưng em đã cố gắng rất nhiều để thực hiện đề tài một cách tốt nhất. Thời gian làm không được nhiều nên em có một số vấn đề chưa hoàn thiện được: Ngoài di chuyển như bình thường quái vật có thể nhảy tới chỗ nhân vật, thêm skill cho nhân vật như bắn đạn,…

3. Hướng phát triển

Ngoài hoàn thiện những phần còn thiếu em sẽ phát triển một số phần như: nhân vật có thể lướt, có thêm nhiều chỉ số hơn, thêm các trang bị mới, và các trang phục cho nhân vật, cải thiện hơn về đồ họa,…

Em nghĩ bản thân cần cố gắng nỗ lực hơn nữa trong tương lai để hoàn thiện bản thân.

TÀI LIỆU THAM KHẢO

**Website tham khảo**

[1] RYE NGUYEN. (2020, September 25). *GameObject - Thao Tác với C# Script — Lập Trình Game*. (n.d.-b). IO Stream. https://www.iostream.vn/lap-trinh-game/gameobject-thao-tac-voi-c-script-y11g3U

[2] Hùng, & Hùng. (2022, April 25). Lập trình game Unity là gì? Cách biến ý tưởng thành sản phẩm Game. *VTC Academy | Học viện Công nghệ thông tin và Thiết kế VTC (VTC Academy)*. https://vtc.edu.vn/lap-trinh-game-unity/

[3] Cường T. M. (2022, February 24). *Cơ bản về lập trình game Unity 2D - Phần 1*. techmaster.vn. https://techmaster.vn/posts/33356/hoc-lap-trinh-game-unity-co-ban

[4] Anh, M. (2022, February 24). *Các câu lệnh cơ bản với Script trong lập trình Game Unity - Cao Đẳng FPT Polytechnic xét tuyển*. Cao Đẳng FPT Polytechnic Xét Tuyển. https://caodang.fpt.edu.vn/tin-tuc-poly/cac-cau-lenh-co-ban-voi-script-trong-lap-trinh-game-unity.html

[5] Technologies, U. (n.d.). *Learning C sharp in Unity for beginners*. Unity. https://unity.com/how-to/learning-c-sharp-unity-beginners