# BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT TRƯỜNG ĐẠI HỌC THỦY LỢI



PHẠM THANH SƠN

PHÁT TRIỂN GAME THE KNIGHT RETURN

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2024

# BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT TRƯỜNG ĐẠI HỌC THỦY LỢI

PHẠM THANH SƠN

PHÁT TRIỂN GAME THE KNIGHT RETURN

Ngành : Công nghệ thông tin

Mã số: 7480201

NGƯỜI HƯỚNG DẪN PGS.TS LÊ ĐỨC HẬU

HÀ NỘI, NĂM 2024

TRƯỜNG ĐẠI HỌC THUỶ LỢI

# KHOA CÔNG NGHỆ THÔNG TIN

BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

**TÊN ĐỀ TÀI**: Phát triển game The Knight Return

*Sinh viên thực hiện*: Phạm Thanh Sơn

*Lớp*: 62TH-NB

*Giáo viên hướng dẫn*: Lê Đức Hậu

## TÓM TẮT ĐỀ TÀI

1. **Bài toán cần giải quyết**

Công nghệ thông tin ngày càng thể hiện được vai trò to lớn và hết sức quan trọng đối với toàn bộ đời sống con người. Cùng với đó, ngành công nghiệp game trên thế giới nói chung và khu vực Đông Nam Á nói riêng đang phát triển mạnh mẽ. Với sự bùng nổ của công nghệ di động và Internet, game trở thành một phần quan trọng của giải trí số và có sức ảnh hưởng rộng lớn. Việc nghiên cứu và phát triển game không chỉ mở ra nhiều cơ hội nghề nghiệp mà còn giúp sinh viên nắm bắt kịp xu hướng phát triển của thế giới.Giải pháp công nghệ. Làm game là một lĩnh vực kết hợp hoàn hảo giữa nghệ thuật và công nghệ. Sinh viên sẽ có cơ hội phát huy khả năng sáng tạo, đồng thời nâng cao kỹ năng kỹ thuật qua việc lập trình, thiết kế đồ họa, âm thanh và trải nghiệm người dùng. Điều này không chỉ giúp họ hoàn thiện bản thân mà còn tạo ra những sản phẩm có giá trị cao. Trong cuộc sống hiện đại đầy áp lực, mọi người thường tìm đến game như một hình thức giải trí để thư giãn và giảm căng thẳng. Game mang lại niềm vui và tạo ra những khoảnh khắc giải trí bổ ích, giúp người chơi tạm quên đi những lo toan hàng ngày. Vì vậy, em quyết định chọn đề tài với tên đề tài: “Phát triển game The Knight Return”.

1. **Giải pháp công nghệ**
   * Công nghệ Unity
   * Ngôn ngữ lập trình và các framework:

+ Unity C#

+ Unity framework.

## CÁC MỤC TIÊU CHÍNH

* Vận dụng kiến thức đã học để phân tích và thiết kế một trò chơi điện tử có cốt truyện và hệ thống nhiệm vụ rõ ràng.
* Vận dụng kiến thức về Unity và các thư viện, framework hỗ trợ để phát triển trò chơi.
* Tiếp cận cách triển khai trò chơi Unity trên nền tảng window và quản lý dữ liệu người chơi bằng JSON Serialization lưu và tải các đối tượng phức tạp bằng cách sử dụng serialization. Unity hỗ trợ JSON và XML để serialize và deserialize các đối tượng..
* Phát triển kỹ năng xây dựng và triển khai trò chơi trên nền tảng window
* Rèn luyện kỹ năng: Khảo sát, phân tích thiết kế hệ thống trò chơi, thiết kế cơ sở dữ liệu, lập trình trò chơi bằng Unity

## KẾT QUẢ DỰ KIẾN

* Thiết kế, xây dựng được Game Play với các chức năng cơ bản để người chơi có thể trải nghiệm:

+ Di chuyển nhân vật.

+ Yếu tô hành dộng như tấn công quái vật.

+ Khám phá bản đồ vượt qua các chướng ngại vật.

+ Thu thập vàng để mua đồ nâng nhân vật

* Thiết kế âm thanh và giao diện người chơi đẹp mắt
* Lưu trữ và quản lý dữ liệu người chơi.



**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập - Tự do - Hạnh phúc**



**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: Phạm Thanh Sơn Hệ đào tạo : Đại học chính quy

Lớp:62TH-NB Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

TÊN ĐỀ TÀI: Phát triển game The Knight Return

1. CÁC TÀI LIỆU CƠ BẢN:
2. NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:

|  |  |
| --- | --- |
| **Nội dung cần thuyết minh** | **Tỉ lệ %** |
| Chương 1: Cơ sở lý thuyết và công nghệ phát triển website   * Bài toán quản lý * Công nghệ phát triển | 25% |
| Chương 2: Mô hình chức năng vào dữ liệu   * Phân tích hệ thống về chức năng * Thiết kế các chức năng * Thiết kế cơ sở dữ liệu | 25% |
| Chương 3 : Mô hình phần mềm   * Backend * Frontend | 25% |

|  |  |
| --- | --- |
| - Mối liên hệ giữa backend và frontend |  |
| Chương 4: Cài đặt ứng dụng   * Môi trường cài đặt * Kết quả đạt được | 25% |

1. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

|  |  |
| --- | --- |
| **Phần** | **Họ và tên giáo viên hướng dẫn** |
| Chương 1: Cơ sở lý thuyết và công nghệ phát triển website   * Bài toán quản lý * Công nghệ phát triển | PGS.TS. Lê Đức Hậu |
| Chương 2: Mô hình chức năng vào dữ liệu   * Phân tích hệ thống về chức năng * Thiết kế các chức năng * Thiết kế cơ sở dữ liệu | PGS.TS. Lê Đức Hậu |
| Chương 3 : Mô hình phần mềm   * Backend * Frontend * Mối liên hệ giữa backend và frontend | PGS.TS. Lê Đức Hậu |
| Chương 4: Cài đặt ứng dụng   * Môi trường cài đặt * Kết quả đạt được | PGS.TS. Lê Đức Hậu |

1. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP Ngày ............ tháng năm 2022

## Trưởng Bộ môn Giáo viên hướng dẫn chính

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

Ngày. . . tháng. năm 2022

## Chủ tịch Hội đồng

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi

Ngày... tháng. năm 2022

**Sinh viên làm Đồ án tốt nghiệp**

# LỜI CAM ĐOAN

Tác giả xin cam đoan đây là Đồ án tốt nghiệp tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào.Việc tham khảo các nguồn tài liệu đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

**Tác giả ĐATN**

**Phạm Thanh Sơn**

LỜI CÁM ƠN

Trong suốt quá trình học tập em luôn được sự quan tâm, hướng dẫn và giúp đỡ tận tình của các giảng viên trong khoa Công nghệ thông tin cùng với sự động viên giúp đỡ của bạn bè.

Lời đầu tiên em xin được bày tỏ lòng biết ơn sâu sắc đến Ban giám hiệu Trường Đại học Thủy Lợi, Ban chủ nhiệm khoa Công nghệ thông tin đã tận tình giúp đỡ em suốt thời gian học tại trường.

Đặc biệt em xin được bày tỏ lòng biết ơn chân thành sâu sắc tới thầy Lê Đức Hậu đã trực tiếp giúp đỡ, hướng dẫn em hoàn thành đồ án tốt nghiệp này.

Em xin trân trọng cám ơn.

**CHƯƠNG 1 CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ PHÁT TRIỂN GAME**

* 1. **Bài toán về công nghiệp trò chơi điện tử**
     1. ***Mô tả bài toán***

Công nghệ thông tin ngày càng thể hiện được vai trò to lớn và hết sức quan trọng đối với toàn bộ đời sống con người. Và ngành công nghiệp game trên thế giới nói chung và khu vực Đông Nam Á nói riêng đang phát triển mạnh mẽ. Với sự bùng nổ của công nghệ di động và Internet, game trở thành một phần quan trọng của giải trí số và có sức ảnh hưởng rộng lớn. Việc nghiên cứu và phát triển game không chỉ mở ra nhiều cơ hội nghề nghiệp mà còn giúp sinh viên nắm bắt kịp xu hướng phát triển của thế giới.

Game, hay trò chơi, là các hoạt động giải trí được thiết kế để tạo ra niềm vui, thử thách, và sự tương tác giữa người chơi. Trò chơi điện tử, một dạng phổ biến của game, là các trò chơi được thực hiện trên các thiết bị điện tử như máy tính, console, điện thoại di động, và các thiết bị chơi game chuyên dụng. Trò chơi điện tử thường bao gồm nhiều thể loại khác nhau như hành động, phiêu lưu, chiến lược, thể thao, và mô phỏng, đáp ứng nhu cầu giải trí đa dạng của người chơi. Trò chơi điện tử cung cấp một phương tiện giải trí phong phú và đa dạng, giúp người chơi giảm căng thẳng và thư giãn sau những giờ làm việc và học tập căng thẳng. Với sự phát triển của công nghệ, trò chơi điện tử ngày càng trở nên hấp dẫn và phong phú về nội dung, mang lại những trải nghiệm giải trí sống động và thú vị. Chơi game không chỉ giúp giải trí mà còn giúp phát triển nhiều kỹ năng quan trọng. Ví dụ, các trò chơi chiến lược giúp cải thiện khả năng lập kế hoạch và quản lý tài nguyên, trong khi các trò chơi hành động nhanh giúp tăng cường phản xạ và khả năng tập trung. Nhiều nghiên cứu đã chỉ ra rằng trò chơi điện tử có thể cải thiện kỹ năng giải quyết vấn đề, tư duy logic và sự sáng tạo của người chơi.

Chính vì vậy, em quyết định chọn đề tài: “Phát triển game The Knight Return”. Việc phát triển "The Knight Return" không chỉ là một dự án kỹ thuật mà còn là một cơ hội để em khám phá và thể hiện sự sáng tạo trong thiết kế trò chơi. Qua đề tài này, em hy vọng sẽ góp phần vào sự phát triển của ngành công nghiệp game, đồng thời mang lại những trải nghiệm giải trí tuyệt vời cho người chơi.

* + 1. ***Yêu cầu thực tế***

Để hoàn thiện đề tài: “Phát triển game The Knight Return” thì yêu cầu đặt ra không hề đơn giản. Dưới đây là một số yêu cầu khi thiết kế :

* Thiết kế giao diện game đơn giản dễ đàng sử dụng.
* GamePlay cơ bản dễ dàng tiếp cận.
* Hệ thống điều khiển mượt mà, cấp độ khó tăng dần theo thời gian.
* Game có cốt truyện, nội dung thú vị gây hứng thú với người chơi.
* Tương thích trên nền tảng window đảm bảo hiệu suất tói ưu.
* Game yêu cầu bảo mật dữ liệu tránh người dùng sửa đổi.
* Hệ thống Save & Load lưu trũ tiến trình của người chơi
  + 1. ***Giải pháp thực hiện***

**1.1.3.1 Thiết kế giao diện game đơn giản dễ dàng sử dụng**

* **Giao diện người dùng (UI):** Giao diện người dùng cần được thiết kế trực quan, dễ dàng điều hướng. Các menu, nút bấm và biểu tượng phải rõ ràng và dễ hiểu, giúp người chơi nhanh chóng nắm bắt được cách sử dụng mà không cần phải tìm hiểu quá nhiều.
* **Hướng dẫn người chơi:** Cung cấp các hướng dẫn cụ thể và dễ hiểu trong quá trình chơi, giúp người chơi mới có thể làm quen với cơ chế và cách thức chơi game một cách nhanh chóng. Điều này có thể bao gồm các hướng dẫn bằng hình ảnh, video ngắn hoặc các chỉ dẫn trực tiếp trong game.

**1.1.3.2 GamePlay cơ bản dễ dàng tiếp cận**

* **Hệ thống điều khiển:** Cơ chế điều khiển cần mượt mà, chính xác và dễ dàng tiếp cận cho người chơi ở mọi cấp độ. Đối với phiên bản trên PC, cần hỗ trợ cả bàn phím và chuột. Đối với phiên bản trên console hoặc thiết bị di động, cần tối ưu hóa cho các phím bấm hoặc màn hình cảm ứng.
* **Cấp độ khó phù hợp:** Game nên có các mức độ khó khác nhau, từ dễ đến khó, để người chơi có thể chọn lựa phù hợp với trình độ của mình. Điều này giúp game có thể tiếp cận được đông đảo người chơi với nhiều trình độ khác nhau. Cấp độ khó sẽ tăng dần theo thời gian, giúp người chơi cảm thấy được thử thách và không nhàm chán.

**1.1.3.3 Game có cốt truyện, nội dung thú vị gây hứng thú với người chơi**

* **Xây dựng cốt truyện hấp dẫn:** Phát triển một cốt truyện có chiều sâu với các tình tiết lôi cuốn, nhân vật phát triển và những bất ngờ thú vị. Cốt truyện cần được truyền tải thông qua các đoạn hội thoại, cảnh cắt và các nhiệm vụ trong game.
* **Nội dung phong phú:** Tạo ra nhiều cấp độ và thử thách đa dạng để người chơi luôn cảm thấy mới mẻ và hứng thú khi chơi game. Mỗi nhiệm vụ và thử thách nên liên kết chặt chẽ với cốt truyện chính, giúp người chơi cảm nhận được sự tiến triển và thành tựu.

**1.1.3.4. Tương thích trên nền tảng Windows đảm bảo hiệu suất tối ưu**

* **Tối ưu hóa hiệu suất:** Đảm bảo game chạy mượt mà trên các thiết bị với cấu hình khác nhau, từ những thiết bị cao cấp đến những thiết bị có cấu hình trung bình và thấp. Sử dụng các công cụ và kỹ thuật tối ưu hóa như giảm bớt các hiệu ứng không cần thiết, tối ưu hóa mã nguồn và sử dụng các công nghệ đồ họa phù hợp.
* **Kiểm tra và thử nghiệm:** Thực hiện các quá trình kiểm tra và thử nghiệm kỹ lưỡng trên nhiều cấu hình phần cứng khác nhau để đảm bảo game hoạt động ổn định và không gặp lỗi.

**1.1.3.5. Game yêu cầu bảo mật dữ liệu tránh người dùng sửa đổi**

* **Bảo vệ dữ liệu người chơi:** Áp dụng các biện pháp bảo mật để bảo vệ dữ liệu cá nhân và thông tin tiến độ chơi của người chơi. Sử dụng mã hóa dữ liệu và lưu trữ trên máy tính

1.2 Tổng quan về game engine

1.2.1 Định nghĩa Game Engine

Game Engine là một nền tảng phần mềm được thiết kế để đơn giản hóa việc phát triển trò chơi điện tử. Nó cung cấp một bộ công cụ và thư viện hỗ trợ các nhà phát triển tạo ra, quản lý và tối ưu hóa các yếu tố khác nhau của trò chơi, bao gồm đồ họa, âm thanh, vật lý, và trí tuệ nhân tạo.

1.2.2 Thành phần cơ bản của Game Engine

Graphics Engine: Chịu trách nhiệm hiển thị đồ họa 2D hoặc 3D của trò chơi. Nó bao gồm các công cụ để tạo ra và quản lý các mô hình, kết cấu, và hiệu ứng ánh sáng.

Physics Engine: Quản lý các quy luật vật lý trong trò chơi, như va chạm, trọng lực, và chuyển động của các đối tượng.

Audio Engine: Xử lý âm thanh trong trò chơi, bao gồm âm nhạc nền, hiệu ứng âm thanh và âm thanh 3D.

Scripting Engine: Hỗ trợ viết mã lệnh điều khiển hành vi của các đối tượng trong trò chơi. Ngôn ngữ lập trình thường được sử dụng bao gồm C++, C#, Lua, và Python.

Animation Engine: Quản lý các chuyển động của nhân vật và đối tượng trong trò chơi, bao gồm cả hoạt cảnh (animation) và điều khiển khung hình (frame control).

Networking Engine: Hỗ trợ các tính năng chơi đa người và trực tuyến, bao gồm kết nối mạng, đồng bộ hóa dữ liệu và quản lý máy chủ.

AI Engine: Quản lý trí tuệ nhân tạo trong trò chơi, bao gồm các hành vi của NPC (Non-Player Characters) và các thuật toán học máy.

1.2.3 Lợi ích của việc sử dụng Game Engine

Tiết kiệm thời gian và công sức: Các nhà phát triển không cần phải viết lại từ đầu tất cả các thành phần cơ bản của trò chơi. Họ có thể tận dụng các công cụ và thư viện có sẵn trong Game Engine.

Khả năng tái sử dụng: Game Engine cho phép tái sử dụng mã nguồn và tài nguyên giữa các dự án khác nhau, giúp tối ưu hóa quá trình phát triển.

Hỗ trợ đa nền tảng: Nhiều Game Engine hỗ trợ xuất bản trò chơi trên nhiều nền tảng khác nhau như PC, console, di động và web.

Cộng đồng và tài liệu hỗ trợ: Nhiều Game Engine có cộng đồng người dùng lớn và tài liệu phong phú, giúp dễ dàng học tập và giải quyết các vấn đề gặp phải trong quá trình phát triển.

1.2.4 Một số Game Engine phổ biến

Unity: Một trong những Game Engine phổ biến nhất, hỗ trợ đa nền tảng và được sử dụng rộng rãi trong cả phát triển trò chơi và ứng dụng tương tác khác.

Unreal Engine: Được phát triển bởi Epic Games, nổi tiếng với khả năng đồ họa cao cấp và thường được sử dụng trong các trò chơi AAA.

Godot: Một Game Engine mã nguồn mở, dễ học và sử dụng, thích hợp cho cả phát triển 2D và 3D.

CryEngine: Nổi bật với đồ họa chất lượng cao, thường được sử dụng trong các trò chơi yêu cầu hiệu suất đồ họa mạnh mẽ.

RPG Maker: Chuyên dùng để phát triển các trò chơi nhập vai 2D, rất phổ biến trong cộng đồng phát triển indie.

Game Engine là một công cụ quan trọng và không thể thiếu trong việc phát triển trò chơi điện tử hiện đại. Nó giúp các nhà phát triển tiết kiệm thời gian, tối ưu hóa quy trình làm việc, và tập trung vào việc sáng tạo nội dung và cải thiện trải nghiệm người chơi. Với sự phát triển nhanh chóng của công nghệ, các Game Engine ngày càng mạnh mẽ và linh hoạt hơn, mở ra nhiều cơ hội cho cả các nhà phát triển chuyên nghiệp và những người mới bắt đầu.

**1.3 Công nghệ phát triển** **Unity Engine**

Trong thời đại công nghệ thông tin như hiện nay, sản phẩm công nghệ ngày càng chịu sự đánh giá khắt khe hơn từ phía những người dùng, đặc biệt là về sản phẩm Game được nhận rất nhiều sự đánh giá từ phía các Game thủ, hay chỉ là những người chơi bình thường. Ngành công nghiệp Game hiện nay có thể nói là bùng nổ, với tốc độ phát triển đến chóng mặt, rất nhiều những Game hay và hấp dẫn đã được ra đời trong thời gian qua. Phía sau những Game phát triển và nổi tiếng như vậy đều có một Game Engine. Game Engine là một công cụ hỗ trợ, một Middleware giúp người phát triển viết Game một cách nhanh chóng và đơn giản, đồng thời cung cấp khả năng tái sử dụng các tài nguyên và mã nguồn cao do có thể phát triển nhiều Game từ một Game Engine.

Từ xu hướng phát triển và những bất cập trên, đồ án này sẽ khảo sát và nghiên cứu về Engine Unity – một Game Engine rất phổ biến và không kém mạnh mẽ hiện nay, nhằm thực nghiệm việc phát triển một trò chơi (Demo) bắn máy bay 2D. Chuẩn bị kiến thức và kỹ năng cho định hướng nghề nghiệp (phát triển Game) sau này của chúng em, góp phần vào sự phát triển của ngành công nghiệp Game nước nhà. Các chương đầu trong tài liệu sẽ trình bày lần lượt các khái niệm chung về Game Engine và thế giới 2D.

1.3.1 Tổng quan về engine Unity.

Unity là một Game Engine được phát triển bởi Unity Technologies, được ra mắt lần đầu vào năm 2005. Unity hỗ trợ lập trình đa nền tảng, cho phép các nhà phát triển tạo ra các trò chơi và ứng dụng tương tác không chỉ trên máy tính để bàn mà còn trên các thiết bị di động, máy chơi game, và các nền tảng thực tế ảo (VR) và thực tế tăng cường (AR). Sau đây là những tính năng chính của engine Unity.

**Đa nền tảng:** Unity hỗ trợ phát triển game cho nhiều nền tảng khác nhau, bao gồm Windows, macOS, Linux, iOS, Android, WebGL, và các hệ máy console như PlayStation, Xbox và Nintendo Switch. Điều này giúp game "The Knight Return" có thể tiếp cận được đa dạng người chơi trên nhiều thiết bị khác nhau.

**Giao diện trực quan:** Unity cung cấp một giao diện người dùng (UI) trực quan và dễ sử dụng, với các công cụ kéo-thả giúp nhà phát triển dễ dàng tạo ra và chỉnh sửa các đối tượng trong game. Giao diện này giúp tối ưu hóa quy trình phát triển và giảm bớt thời gian học tập cho người mới.

**Công cụ đồ họa mạnh mẽ:** Unity hỗ trợ các công nghệ đồ họa tiên tiến như đồ họa 2D và 3D, ánh sáng động, đổ bóng, và hiệu ứng hậu kỳ (post-processing). Điều này giúp tạo ra hình ảnh chất lượng cao và trải nghiệm thị giác hấp dẫn cho người chơi.

**Hệ thống vật lý và hoạt ảnh:** Unity tích hợp sẵn hệ thống vật lý mạnh mẽ và công cụ tạo hoạt ảnh (animation), giúp tạo ra các hiệu ứng chuyển động và tương tác vật lý chân thực trong game. Các công cụ này rất hữu ích cho việc phát triển cơ chế điều khiển và hoạt động của nhân vật trong "The Knight Return".

**Hệ thống script:** Unity sử dụng ngôn ngữ lập trình C# cho việc viết script, giúp nhà phát triển dễ dàng tạo ra các tính năng và logic cho game. C# là ngôn ngữ phổ biến và mạnh mẽ, hỗ trợ phát triển game một cách hiệu quả.

**Unity Editor**: Đây là công cụ chính của Unity, nơi người dùng có thể thiết kế, lập trình và kiểm tra trò chơi. Unity Editor bao gồm nhiều cửa sổ và công cụ hỗ trợ cho việc chỉnh sửa scene, quản lý asset, và lập trình.

**Cộng đồng và tài liệu phong phú:** Unity có một cộng đồng nhà phát triển lớn và sôi động, cùng với tài liệu hướng dẫn chi tiết và nhiều tài nguyên học tập. Nhà phát triển có thể dễ dàng tìm thấy các giải pháp cho vấn đề của mình hoặc học hỏi kinh nghiệm từ cộng đồng.

**Asset Store:** Unity Asset Store cung cấp một kho tài nguyên phong phú, bao gồm các mẫu đồ họa, âm thanh, script, và các plugin bổ sung. Nhà phát triển có thể tận dụng các tài nguyên này để tiết kiệm thời gian và công sức trong quá trình phát triển game.

Unity là một công cụ phát triển game mạnh mẽ và đa năng, mang lại nhiều lợi ích cho nhà phát triển game từ cá nhân đến các studio lớn. Với các tính năng mạnh mẽ, hỗ trợ đa nền tảng, và cộng đồng phát triển sôi động, Unity là lựa chọn hàng đầu cho việc phát triển game "The Knight Return" và nhiều dự án game khác. Việc sử dụng Unity không chỉ giúp tối ưu hóa quy trình phát triển mà còn mang lại trải nghiệm tuyệt vời cho người chơi.

**1.3.2 Tổng quan về C# trong Unity**

1.3.2.1. Giới thiệu về C# trong Unity

C# là ngôn ngữ lập trình chính được sử dụng trong Unity, một engine phát triển game phổ biến. Unity hỗ trợ C# vì tính mạnh mẽ, hiệu suất cao và cú pháp dễ hiểu, phù hợp cho việc phát triển game. C# được phát triển bởi Microsoft và là một phần của nền tảng .NET, giúp nó có một cộng đồng phát triển lớn và nhiều tài nguyên hỗ trợ.

1.3.2.2. Lý do chọn C# cho Unity

Tính mạnh mẽ và hiệu suất cao: C# cung cấp các tính năng lập trình hiện đại như lambda expressions, LINQ, và async/await, giúp viết mã nguồn hiệu quả và tối ưu. Điều này đặc biệt quan trọng trong phát triển game, nơi hiệu suất là yếu tố then chốt.

Cú pháp dễ hiểu: C# có cú pháp dễ học và dễ hiểu, đặc biệt với những người đã quen thuộc với các ngôn ngữ lập trình hướng đối tượng khác như Java hoặc C++. Điều này giúp các nhà phát triển nhanh chóng làm quen và bắt đầu phát triển game trong Unity.

Tích hợp tốt với Unity: Unity cung cấp các API và thư viện phong phú bằng C#, giúp dễ dàng tạo và quản lý các thành phần trong game. Sự tích hợp chặt chẽ giữa Unity và C# giúp tối ưu hóa quy trình phát triển và giảm bớt sự phức tạp.

Tài liệu và cộng đồng hỗ trợ: C# có một cộng đồng lớn và nhiều tài liệu hướng dẫn, từ các khóa học trực tuyến đến các diễn đàn hỗ trợ. Nhà phát triển có thể dễ dàng tìm thấy giải pháp cho các vấn đề của mình và học hỏi kinh nghiệm từ cộng đồng.

1.3.2.3. Các thành phần chính trong lập trình C# cho Unity

Script: Scripts trong Unity được viết bằng C# để định nghĩa hành vi và logic cho các đối tượng trong game. Một script cơ bản thường bao gồm các phương thức như Start(), Update(), và FixedUpdate() để quản lý các sự kiện trong vòng đời của đối tượng.

GameObject và Component: Trong Unity, mọi thứ đều là một GameObject, và các Component được gắn vào GameObject để thêm chức năng. Các script C# thường được gắn vào GameObject dưới dạng Component, giúp dễ dàng quản lý và tổ chức mã nguồn.

API và Thư viện Unity: Unity cung cấp một bộ API và thư viện phong phú cho việc quản lý các yếu tố như vật lý, âm thanh, hình ảnh, và đầu vào. Các API này giúp nhà phát triển dễ dàng tạo ra các hiệu ứng và tương tác phức tạp trong game.

Sự kiện và Hệ thống sự kiện: Unity sử dụng hệ thống sự kiện để quản lý các tương tác giữa người chơi và game. Các sự kiện như nhấp chuột, va chạm, và kích hoạt trigger đều có thể được xử lý bằng các script C#, giúp tạo ra trải nghiệm chơi game động và phản hồi nhanh.

C# là ngôn ngữ lập trình mạnh mẽ và hiệu quả, được sử dụng rộng rãi trong phát triển game với Unity. Với sự hỗ trợ từ Unity Editor, các API phong phú, và cộng đồng phát triển lớn, C# giúp nhà phát triển dễ dàng tạo ra các trò chơi chất lượng cao. Việc nắm vững C# không chỉ giúp bạn phát triển game một cách hiệu quả mà còn mở rộng cơ hội nghề nghiệp trong lĩnh vực công nghệ thông tin và phát triển phần mềm.

**1.3.3 Các Unity Framework**

1.3.3.1. UnityEngine Framework

UnityEngine là nền tảng chính của Unity, cung cấp các lớp và API cần thiết để phát triển game. Nó bao gồm các module quan trọng sau:

Physics: Xử lý vật lý cho các đối tượng trong game, bao gồm va chạm, trọng lực, và lực.

Rendering: Quản lý việc hiển thị các đối tượng 2D và 3D trên màn hình.

Animation: Tạo và quản lý các hoạt ảnh của nhân vật và đối tượng.

Audio: Quản lý âm thanh, bao gồm phát và xử lý âm thanh.

Networking: Hỗ trợ tạo các ứng dụng mạng và game nhiều người chơi.

1.3.3.2. Unity Editor Framework

UnityEditor cung cấp các công cụ và API để mở rộng và tùy chỉnh Unity Editor. Các nhà phát triển có thể tạo ra các công cụ tùy chỉnh, menu, và cửa sổ riêng để tối ưu hóa quy trình phát triển.

Custom Editor Windows: Tạo các cửa sổ và công cụ tùy chỉnh trong Unity Editor.

Editor Scripting: Viết các script để tự động hóa các tác vụ trong Editor, như xây dựng asset bundles hoặc kiểm tra lỗi.

1.3.3.3. Unity UI (uGUI)

Unity UI là framework để tạo giao diện người dùng trong game. Nó bao gồm các công cụ để tạo và quản lý các phần tử UI như nút bấm, thanh trượt, và hộp thoại.

Canvas: Là nền tảng để vẽ các phần tử UI.

Event System: Quản lý các sự kiện tương tác người dùng như nhấp chuột và chạm màn hình.

Layout Groups: Sắp xếp và căn chỉnh các phần tử UI một cách tự động.

1.3.3.4. Unity 2D Framework

Unity 2D cung cấp các công cụ và API để phát triển game 2D, bao gồm:

Sprite Renderer: Hiển thị hình ảnh 2D (sprite) trên màn hình.

Tilemap: Tạo và quản lý bản đồ ô vuông (tile-based map) trong game.

Physics2D: Xử lý vật lý cho các đối tượng 2D.

1.3.3.5. Unity 3D Framework: Unity 3D cung cấp các công cụ và API để phát triển game 3D, bao gồm:

Mesh Renderer: Hiển thị các mô hình 3D trên màn hình.

Terrain: Tạo và quản lý địa hình 3D.

NavMesh: Quản lý điều hướng AI cho các đối tượng trong môi trường 3D.

1.3.3.6. Unity ECS (Entity Component System)

Unity ECS là một kiến trúc phát triển game hiệu suất cao, tách biệt dữ liệu và hành vi của các đối tượng game. ECS cho phép tối ưu hóa việc xử lý dữ liệu và cải thiện hiệu suất.

Entities: Các đơn vị dữ liệu cơ bản trong ECS.

Components: Chứa dữ liệu liên quan đến các entities.

Systems: Xử lý logic và hành vi dựa trên dữ liệu từ các components.

1.3.3.7. Unity Addressable Assets

Addressable Assets là hệ thống quản lý tài nguyên giúp dễ dàng tải và quản lý tài nguyên trong game, đặc biệt là các game có quy mô lớn hoặc nội dung động.

Addressable: Đánh dấu tài nguyên để quản lý chúng dễ dàng.

Profile: Quản lý các cấu hình tải tài nguyên khác nhau cho các nền tảng hoặc môi trường khác nhau.

1.3.3.8. Unity Timeline

Timeline là công cụ để tạo các đoạn cắt cảnh, hoạt cảnh phức tạp và điều khiển các sự kiện trong game.

Tracks: Các dòng thời gian chứa các đối tượng và sự kiện.

Clips: Các đoạn thời gian cụ thể chứa các hành động hoặc hoạt cảnh.

Signals: Các sự kiện kích hoạt trong quá trình chạy Timeline.

1.3.3.9. Unity Cinemachine

Cinemachine là công cụ quản lý camera mạnh mẽ, giúp tạo các chuyển động camera mượt mà và chuyên nghiệp mà không cần viết nhiều mã.

Virtual Cameras: Các camera ảo có thể được thiết lập và điều chỉnh tự động.

Cinemachine Brain: Quản lý các camera ảo và chuyển đổi giữa chúng một cách mượt mà.

Confiner: Giữ camera trong một khu vực giới hạn.

1.3.3.10. Unity Jobs System và Burst Compiler

Jobs System và Burst Compiler là các công cụ tối ưu hóa hiệu suất giúp tận dụng tối đa sức mạnh của CPU đa lõi.

Jobs System: Tạo và quản lý các công việc song song để tối ưu hóa hiệu suất.

Burst Compiler: Tăng hiệu suất của mã C# bằng cách biên dịch nó xuống mã máy được tối ưu hóa.

Unity cung cấp một loạt các framework mạnh mẽ giúp tối ưu hóa quy trình phát triển game và nâng cao chất lượng sản phẩm cuối cùng. Từ quản lý UI, đồ họa 2D/3D, hệ thống vật lý, đến tối ưu hóa hiệu suất, Unity giúp nhà phát triển tạo ra những trò chơi phong phú và hấp dẫn trên nhiều nền tảng khác nhau. Việc hiểu rõ và tận dụng các framework này sẽ giúp tối ưu hóa quy trình phát triển và nâng cao chất lượng game

**1.3.4 JSON Serialization và Deserialization trong Unity**

1.3.4.1. Giới thiệu về JSON

JSON (JavaScript Object Notation) là một định dạng trao đổi dữ liệu nhẹ, dễ đọc và dễ viết cho con người, đồng thời dễ dàng phân tích và sinh ra bởi máy móc. JSON thường được sử dụng để truyền dữ liệu giữa máy chủ và ứng dụng web, nhưng nó cũng phổ biến trong nhiều ứng dụng khác, bao gồm cả game development.

1.3.4.2. Tại sao sử dụng JSON trong Unity?

JSON được sử dụng rộng rãi trong Unity để lưu trữ và truyền tải dữ liệu, bao gồm cấu hình game, thông tin người chơi, trạng thái game và nhiều thứ khác. Lợi ích của việc sử dụng JSON bao gồm:

Định dạng nhẹ và dễ đọc: JSON dễ đọc và viết, làm cho việc kiểm tra và sửa lỗi trở nên dễ dàng hơn.

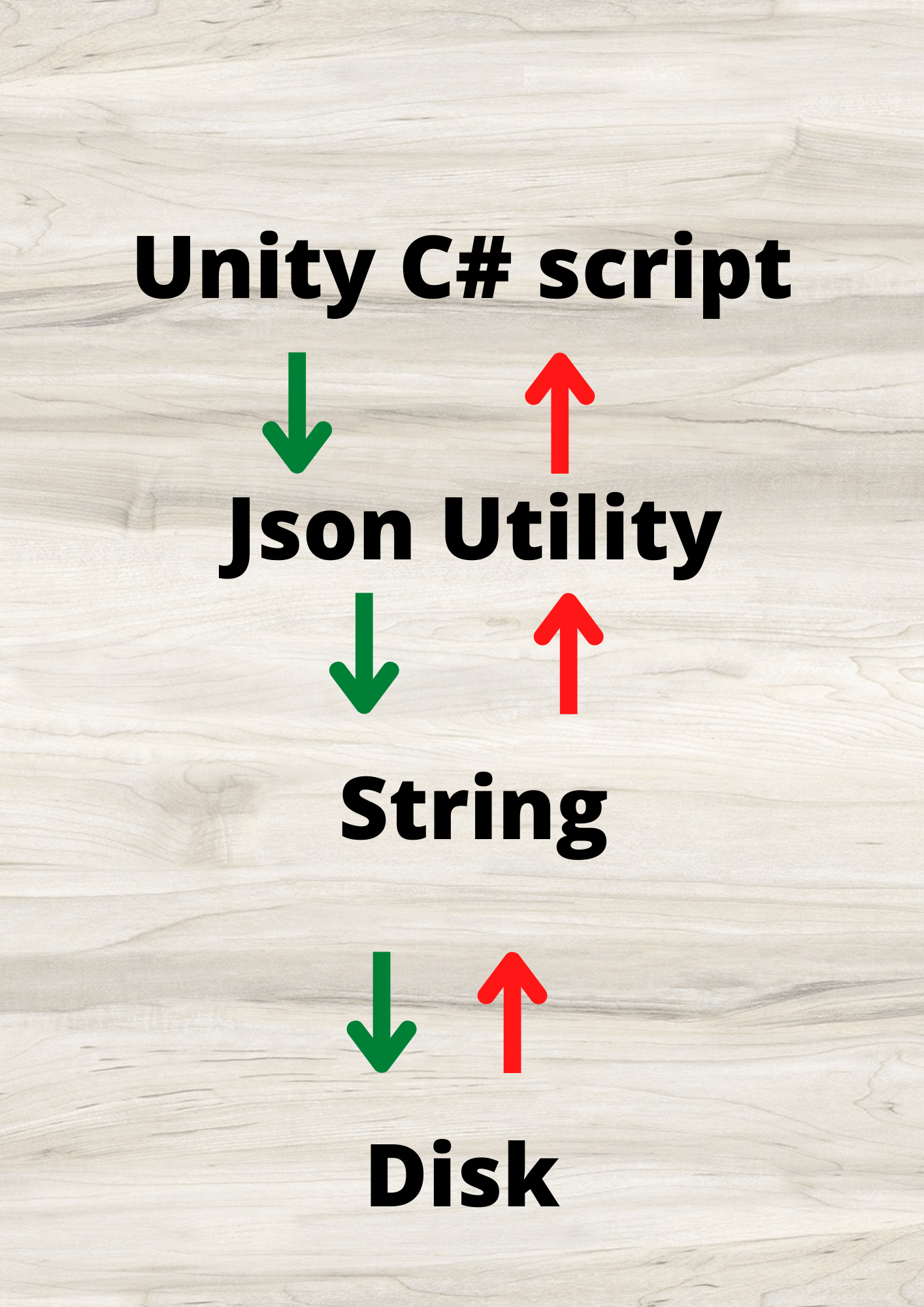
Khả năng tương thích tốt: JSON có thể được sử dụng trên nhiều nền tảng và ngôn ngữ lập trình khác nhau.

Tiện lợi cho việc lưu trữ và truyền tải dữ liệu: JSON là một lựa chọn tốt để lưu trữ dữ liệu tạm thời hoặc truyền tải dữ liệu qua mạng.

1.3.4.3. JSON Serialization và Deserialization trong Unity

Serialization là quá trình chuyển đổi đối tượng (object) thành một định dạng mà có thể lưu trữ hoặc truyền tải được, chẳng hạn như JSON. Deserialization là quá trình ngược lại, chuyển đổi dữ liệu từ định dạng JSON trở lại thành đối tượng.

Unity cung cấp một số phương pháp để thực hiện JSON Serialization và Deserialization như JsonUtility



1.3.4.4. Sử dụng JsonUtility trong Unity

JsonUtility là lớp được Unity cung cấp để làm việc với JSON. Nó nhanh chóng và dễ sử dụng cho các tác vụ đơn giản.

JsonUtility:

Ưu điểm:

Tích hợp sẵn trong Unity, không cần cài đặt thêm.

Hiệu suất cao cho các tác vụ đơn giản.

Nhược điểm:

Ít tính năng hơn, không hỗ trợ nhiều cấu trúc dữ liệu phức tạp.

JSON Serialization và Deserialization là các kỹ thuật quan trọng trong phát triển game bằng Unity, giúp quản lý và truyền tải dữ liệu một cách hiệu quả. Unity cung cấp sẵn JsonUtility để thực hiện các thao tác này một cách nhanh chóng và dễ dàng.

1.3.5 Quy trình phát triển game bằng Unity

Để hoàn thiện một dự án game trên Unity, cần tuân theo các bước quy trình phát triển sau:

1.3.5.1 Thiết kế và lập kế hoạch:

Xác định ý tưởng và khái niệm game: Lên ý tưởng và xác định mục tiêu của game, bao gồm cốt truyện, nhân vật, và các yếu tố gameplay chính.

Xây dựng cốt truyện và nhân vật: Phát triển cốt truyện chi tiết và thiết kế các nhân vật chính trong game.

Thiết kế các cấp độ và nhiệm vụ: Lên kế hoạch cho các cấp độ và nhiệm vụ, đảm bảo rằng game có sự cân bằng giữa thử thách và niềm vui.

Lập kế hoạch phát triển chi tiết: Xác định các giai đoạn phát triển, phân chia công việc và đặt ra các mốc thời gian cụ thể.

1.3.5.2. Tạo dựng game trong Unity:

Thiết kế giao diện và các màn hình game: Sử dụng Unity Editor để thiết kế các màn hình menu, giao diện người dùng và các yếu tố đồ họa khác.

Xây dựng các mô hình 3D/2D và tạo hoạt ảnh: Tạo ra các mô hình và hoạt ảnh cho nhân vật, môi trường, và các đối tượng khác trong game.

Viết script cho các tính năng và logic game: Sử dụng C# để lập trình các tính năng, tương tác, và logic của game.

Tích hợp âm thanh và nhạc nền: Thêm các hiệu ứng âm thanh và nhạc nền để tăng cường trải nghiệm chơi game.

1.3.5.3. Kiểm tra và tối ưu hóa:

Thực hiện kiểm tra chất lượng và tìm lỗi: Kiểm tra game để phát hiện và sửa các lỗi, đảm bảo game hoạt động mượt mà và không có lỗi nghiêm trọng.

Tối ưu hóa hiệu suất và kích thước file: Tối ưu hóa mã nguồn, đồ họa và các tài nguyên khác để đảm bảo game chạy mượt mà trên các nền tảng khác nhau.

Điều chỉnh các yếu tố gameplay và đồ họa: Dựa trên phản hồi từ thử nghiệm, điều chỉnh gameplay và đồ họa để đảm bảo game hấp dẫn và cân bằng.

1.3.5.4. Phát hành và bảo trì:

Chuẩn bị và triển khai game trên các nền tảng đã chọn: Đóng gói và phát hành game trên các cửa hàng ứng dụng và nền tảng phân phối game.

Theo dõi phản hồi của người chơi và cập nhật game: Lắng nghe phản hồi từ người chơi và thực hiện các cập nhật cần thiết để cải thiện game.

Bảo trì và cải tiến game để duy trì sự hấp dẫn: Liên tục bảo trì và cập nhật game, thêm nội dung mới và cải thiện trải nghiệm chơi game.

1.3.5.5 Công cụ và tài nguyên bổ sung

Visual Studio: Visual Studio là môi trường phát triển tích hợp (IDE) phổ biến, được sử dụng để viết mã C# trong Unity. Nó cung cấp nhiều công cụ hỗ trợ lập trình như gỡ lỗi, quản lý mã nguồn, và tích hợp với Unity Editor.

Photoshop/GIMP: Photoshop và GIMP là các công cụ chỉnh sửa hình ảnh mạnh mẽ, giúp tạo ra các kết cấu và hình ảnh 2D cho game. Photoshop là một phần mềm thương mại, trong khi GIMP là một lựa chọn mã nguồn mở miễn phí.

Unity Collaborate và Git: Unity Collaborate và Git là các công cụ quản lý mã nguồn giúp quản lý và đồng bộ hóa mã nguồn và tài nguyên của dự án giữa các thành viên trong nhóm. Điều này rất hữu ích cho việc làm việc nhóm và theo dõi các thay đổi trong dự án.

**Chương 2: Tổng quan về Game “The Knight Return”**

**2.1 Thể loại game**

2.1.1Phiêu lưu, hành động, khám phá. Thể loại game 2D.

**2.2 Yếu tố**

**2.2.1 Xây dựng nhân vật**: Nhẹ nhàng và tinh tế.

**2.2.2 Phong cách chiến đấu**: Ấn tượng và bắt mắt.

**2.2.3 Nghệ thuật**: Pixel đơn giản nhưng cuốn hút.

**2.2.4 Đối tượng**: Nhắm tới người chơi trẻ từ 14 tuổi trở lên.

**2.2.5 Hệ thống chiến đấu**: Đa dạng về quái vật và mức độ, mang đến sự hào hứng và tránh sự nhàm chán. Người chơi sẽ nhận được kỹ năng mới sau mỗi màn chơi.

**2.2.6 Phong cách phiêu lưu**: Kết hợp với áp lực từ các cạm bẫy và quái vật, lối chơi thử thách tạo nên sự khác biệt và nổi bật hơn so với các game cùng thể loại.

**2.3 Nội dung**

**2.3.1 Bối cảnh**: Nhân vật chính tỉnh dậy trong một khu rừng bí ẩn. Người chơi điều khiển nhân vật tiến lên phía trước, vượt qua cạm bẫy và quái vật trên hành trình của mình.

**2.3.2 Lối chơi**: Người chơi sẽ sử dụng kiếm để đánh vào những con quái vật, chúng ẩn nấp, rải rác khắp nơi không để bị tấn công và mất máu.

**2.3.3 Thử thách**: Lối chơi đòi hỏi sự chú ý và kỹ năng cao của người chơi.

**2.3.4 Âm thanh**: Mỗi bản đồ có âm thanh riêng, tạo cảm giác căng thẳng và thư giãn khác nhau.

**2.3.5 Đồ họa**: 2D đẹp mắt kết hợp với nhạc nền xuất sắc là yếu tố quyết định của trò chơi.

**2.4 Chủ đề**

2.4.1 Game nhập vai hành động tiêu diệt quái vật, khám phá thế giới.

**2.5 Phong cách**

2.5.1 Tăm tối, u ám và chết chóc với những con quái vật ghê rợn.

**2.6 Đối tượng người chơi**

2.6.1 Người chơi từ 14 tuổi trở lên, yêu thích nhập vai vào thế giới u ám đầy quái vật đáng sợ, với sự thay đổi liên tục qua các vùng đất khác nhau.

**2.7 Lối chơi**

**2.7.1 Menu chính**: Người chơi sẽ chọn các mục tương ứng:

Play game: Vào các màn chơi.

Setting: Cài đặt game.

Quit: Thoát khỏi game.

**2.7.2 Điều khiển**:

Phím mũi tên: Di chuyển nhân vật.

Phím X: Tấn công.

Phím C: Lướt.

Phím Z: Nhảy

Phím ESC: Vào menu Pause, ấn lần nữa để thoát.

**2.8. Giao diện và cảm nhận**

Bối cảnh: Thế giới rộng lớn với những con quái vật ở khắp mọi nơi. Người chơi phải tiến lên phía trước qua các vùng đất để đến được lâu đài.

Phong cách: Tăm tối, kinh dị cả về hình ảnh lẫn âm thanh, tạo cảm giác rùng rợn và kích thích sự tò mò, ham khám phá của người chơi.

Âm thanh: Nền âm thanh vừa nhẹ nhàng vừa dồn dập, âm thanh từ quái vật và các pha tấn công mạnh mẽ gây kích thích.

**2.9 Khía cạnh tác động đến người chơi**

Hình ảnh và âm thanh: Tác động mạnh mẽ, tạo sự hấp dẫn.

Cách điều khiển và tấn công: Đòi hỏi kỹ năng và sự cẩn trọng.

Cốt truyện: Khiến người chơi tò mò và muốn tìm hiểu thêm.

**2.10 Mục tiêu trải nghiệm**

* **Nhập vai và cảm xúc**: Giúp người chơi hòa nhập vào thế giới trong game, mang lại cảm xúc mà khó có thể trải nghiệm ngoài đời thực. Tạo ra những phút giây thư giãn và giải trí sau những giờ học tập và làm việc căng thẳng.

**PHẦN 3: GAMEPLAY & MECHANICS**

**1. Game Player**

Trò chơi hành động mang yếu tố chiến lược. Mục tiêu duy nhất là tiêu diệt kẻ thù vượt qua cạm bẫy cho đến khi tìm ra lối thoát, đánh bại boss khiến chiếc rương hiện ra chạm vào và qua àn. Trò chơi bắt đầu ở một khu rừng tăm tối. Người chơi trải nghiệm thế giới trò chơi qua con mắt của một nhân vật chính. Điều này tạo cảm giác nhập vai cho người chơi, người chơi điều khiển nhân vật chính tên là “Javan”, một vị hoàng tử bị mất trí nhớ đi trên hành trình của mình để tìm ra mục tiêu của bản thân, hướng về lâu đài của mình và đánh bại quỷ vương.

Nhân vật di chuyển tự do trong màn chơi, sử dụng kiếm để đánh quái vật. Trò chơi có thiết kế chỉ có chế độ chơi đơn. Người chơi bắt đầu trò chơi với một vũ khí và số lượng máu nhất định. Theo mặc định, người chơi có 3 HP, khi bị quái vật thì bị giảm 1 HP, khi chạm vào bẫy người chơi sẽ chết. Hiệu quả của sát thương sẽ thay đổi tùy thuộc vào loại quái vật khác nhau. Tuy nhiên, người chơi có thể nhặt lọ máu để hồi phục HP và trái tim để tăng lượng máu tối đa.

Người chơi chỉ có một mạng. Khi HP của nhân vật giảm xuống 0, nhân vật chính chết. Khi nhân vật chính chết, người chơi sẽ quay lại điểm checkpoint gần nhất. Nếu người chơi tiêu diệt tất cả kẻ thù và nhặt được chiếc rương để qua màn.

Quái tấn công người chơi khi nhân vật đến gần. Chúng được sinh ra tại các vị trí trong bản đồ. Số lượng quái vật được giới hạn hữu hạn. Mỗi loại quái vật có một mức HP khác nhau. Quái vật bị tấn công, sẽ bị giảm HP. Khi HP của quái vật giảm xuống 0, quái vật sẽ chết và biến mất. Nhân vật chính giết chết quái vật cuối cùng (boss) để lấy được rương báu để qua màn.

**2. Level design**

Game sẽ có 4 map khác nhau

Map 1: Dark Forest

Map 2: Dead village

Map 3: Under The Castle

Map 4: Devil’s Castle

**3. Game mode**

Trò chơi có thiết kế chỉ có chế độ chơi đơn. Trong chế độ chơi đơn, người chơi vượt qua từng map, số lượng kẻ thù và lượng máu bị mất khi bị tấn công sẽ khác nhau. Khi trò chơi ở map cao hơn, số lượng kẻ thù sẽ tăng lên. Bảng dưới đây cung cấp thông tin về số lượng mạnh và kẻ thù trong mỗi cấp độ.

Player (chỉ số giữ nguyên): Khi va chạm với quái sẽ bị -1 HP,

Enemy: Có số lượng máu khác nhau. Tăng dần qua từng map

|  |  |  |  |
| --- | --- | --- | --- |
| Map | Quái thường | Boss | Khi bị tấn công |
| 1 | 3 HP | 10 HP | -1 HP |
| 2 | 4 HP | 15 HP | -1 HP |
| 3 | 5 HP | 20 HP | -1 HP |
| 4 | 6 HP | 30 HP | -1 HP |
|  |  |  |  |
|  |  |  |  |

**4. Game Control**

Trò chơi sẽ sử dụng chuột và bàn phím để điều khiển

Movement

* Left - A
* Right - D
* Jump – W, Space

Actions

* Attack Weapon - Left Mouse
* Dash - Right Mouse

**5. Winning and Losing**

Mục tiêu của trò chơi là tiêu diệt tất cả kẻ thù trước khi chúng giết người chơi.

Người chơi chỉ có 1 mạng với số lượng HP nhất định. Khi HP của player giảm xuống 0, người chơi sẽ chết => Sẽ dịch chuyển đến checkpoint gần nhất (Tượng Thần)

Ngưởi chơi cần phải đánh bại boss màn để rương báu hiện ra (không hiện khi boss còn sống)

Khi boss còn sống

Khi boss chết

Nếu người chơi vượt qua map 4 thì sẽ giành chiến thắng.

**6. Heart potion and Heart**

Heart potion: giúp hồi máu đã mất.

Heart: Giúp tăng số lượng máu tối đa.