**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----



# BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI: XÂY DỰNG GAME SỬA TÀU BẰNG PUZZLE SỬ DỤNG GODOT ENGINE

Giảng viên hướng dẫn: THS. TRẦN PHONG NHÃ

Sinh viên thực hiện: LÊ THÀNH MINH KHOA

Lớp: CQ.60.CN.CNTT

MSSV: 6051071059

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----



# BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI: ỨNG DỤNG GODOT ĐỂ PHÁT TRIỂN TRÒ CHƠI 2D HỖ TRỢ HỌC LẬP TRÌNH

Giảng viên hướng dẫn: THS. TRẦN PHONG NHÃ

Sinh viên thực hiện: LÊ THÀNH MINH KHOA

Lớp: CQ.60.CN.CNTT

MSSV: 6051071059

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

\*\*\*

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

Độc lập – Tự do – Hạnh phúc

## NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP BỘ MÔN: CÔNG NGHỆ THÔNG TIN

**-------\*\*\*-------**

**Mã sinh viên**: 6051071059 **Họ tên SV**: Lê Thành Minh Khoa

**Khóa**: 60 **Lớp**: CQ.60.CN.CNTT

1. **Tên đề tài** : Game sửa tàu dựa trên puzzle bằng Godot Engine

### Mục đích, yêu cầu

* 1. **Mục đích**

Mục tiêu chính của đề tài là phát triển một trò chơi 3D đơn giản bằng công cụ Godot Engine, nhằm hỗ trợ người mới bắt đầu làm quen với lập trình một cách dễ hiểu và trực quan. Thông qua các thử thách và tình huống trong game, người chơi sẽ tiếp cận được các khái niệm lập trình cơ bản như biến, vòng lặp, điều kiện,... theo hình thức “vừa học vừa chơi”. Trò chơi không chỉ giúp tăng hứng thú học tập mà còn góp phần rèn luyện tư duy logic và khả năng giải quyết vấn đề cho người học. Bên cạnh đó, đề tài cũng giúp người thực hiện làm quen với quy trình phát triển game 3D, từ thiết kế đến triển khai sản phẩm hoàn chỉnh.

### Yêu cầu

Để thực hiện đề tài, cần nghiên cứu và nắm vững các yếu tố cơ bản của dòng game giải đố (Puzzle), bao gồm hệ thống mô phỏng, vật phẩm, nhiệm vụ và tương tác nhân vật. Đây là những thành phần quan trọng để tạo chiều sâu và sự hấp dẫn cho trò chơi, đồng thời tạo cơ hội tích hợp nội dung học lập trình một cách tự nhiên.

Ngoài ra, việc sử dụng Godot Engine 4 – một công cụ mã nguồn mở mạnh mẽ – đòi hỏi người thực hiện phải tìm hiểu cách xây dựng bản đồ, xử lý sự kiện, thiết kế giao diện người dùng và lập trình hành vi nhân vật bằng GDScript.

Game khi hoàn thiện cần đảm bảo ổn định, dễ thao tác, tối ưu trải nghiệm người chơi và hạn chế lỗi kỹ thuật.

### Nội dung và phạm vi đề tài

* 1. **Nội dung đề tài**

Đề tài tập trung vào việc xây dựng một trò chơi 3D đơn giản bằng Godot Engine với mục tiêu hỗ trợ người mới học lập trình tiếp cận kiến thức cơ bản một cách trực quan, dễ hiểu và sinh động. Trò chơi được thiết kế xoay quanh hành trình của một nhân vật chính trong thế giới ảo, nơi người chơi cần điều khiển nhân vật di chuyển, tương tác với môi trường, đối mặt với nhiều vấn đề khác nhau và hoàn thành nhiệm vụ.

Điểm đặc biệt của trò chơi là các thử thách lập trình được lồng ghép khéo léo vào quá trình chơi dưới dạng câu đố logic hoặc nhiệm vụ cần giải bằng tư duy thuật toán. Người chơi không chỉ giải trí mà còn có cơ hội rèn luyện kiến thức lập trình thông qua những tình huống thực tế trong game. Các chủ đề kiến thức là lập trình căn bản.

### Phạm vi đề tài

Trò chơi được xây dựng với quy mô nhỏ, mang tính giáo dục kết hợp giải trí, phù hợp với đối tượng là người mới học lập trình. Nội dung trò chơi tập trung vào việc điều khiển một nhân vật duy nhất trong môi trường 3D, với lối chơi thiên về ghi nhớ và lặp lại.

Người chơi sẽ cần tiêu diệt các quái vật xuất hiện ngẫu nhiên, né tránh đòn tấn công và sống sót càng lâu càng tốt để hoàn thành nhiệm vụ. Hệ thống vật phẩm trong trò chơi được thiết kế theo cơ chế rơi ngẫu nhiên khi người chơi tiêu diệt kẻ địch hoặc phá hủy vật thể trong môi trường, tạo nên yếu tố bất ngờ và khuyến khích khám phá.

### Công nghệ, công cụ, ngôn ngữ lập trình

Đồ án sử dụng Godot Engine 4 – một công cụ phát triển trò chơi mã nguồn mở mạnh mẽ, hỗ trợ lập trình game 2D và 3D trên nhiều nền tảng với hiệu suất tối ưu. Godot cung cấp một hệ sinh thái hoàn chỉnh cho việc phát triển game, bao gồm trình chỉnh sửa trực quan (Godot Editor) cho phép thiết kế cảnh, lập trình logic, và quản lý tài nguyên một cách hiệu quả.

Để quản lý mã nguồn và theo dõi quá trình phát triển, dự án tích hợp GitHub như một công cụ quản lý phiên bản, giúp dễ dàng kiểm soát, cập nhật và cộng tác trong quá trình lập trình.

Ngôn ngữ lập trình chính được sử dụng là GDScript – một ngôn ngữ kịch bản cấp cao có cú pháp gần giống Python, được thiết kế riêng cho Godot. GDScript có khả năng tích hợp chặt chẽ với các thành phần của Godot Engine, cho phép lập trình nhanh

chóng, linh hoạt và tối ưu hiệu suất khi phát triển game. Đây là lựa chọn lý tưởng cho các dự án game quy mô nhỏ đến trung bình nhờ sự đơn giản, dễ học và hiệu quả mà nó mang lại.

### Giáo viên và cán bộ hướng dẫn

Họ tên: TRẦN PHONG NHÃ

Học vị: Thạc sĩ

Đơn vị công tác: Bộ môn Công nghệ Thông tin – Trường Đại học Giao thông vận tải Phân hiệu

Điện thoại: 0906761014 Email: [tpnha@utc2.edu.vn](mailto:ttdung@utc2.edu.vn)

**Ngày tháng năm 2026 Trưởng BM Công nghệ Thông tin**

**Đã giao nhiệm vụ TKTN Giáo viên hướng dẫn**

**THS. Trần Phong Nhã THS. Trần Phong Nhã**

*TP. Hồ Chí Minh, ngày tháng năm 2026*

### Sinh viên thực hiện

**Lê Thành Minh Khoa**

Đã nhận nhiệm vụ TKTN

Sinh viên: Lê Thành Minh Khoa Ký tên: Khoa

Điện thoại: 0905349132 Email: [6051071059@st.utc2.edu.vn](mailto:6251071082@st.utc2.edu.vn)

# LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành nhất đến Ban Giám hiệu nhà trường, Khoa công nghệ thông tin , Bộ môn công nghệ thông tin đã tạo điều kiện thuận lợi cho em trong quá trình học tập và nghiên cứu.Trước hết em gửi tới các thầy cô Bộ môn Công nghệ thông tin Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Đặc biệt, em xin gửi lời cảm ơn đến thầy Trần Phong Nhã, thầy đã tận tình hướng dẫn chỉ bảo để em có thể tiếp cận với nhiều bài toán thực tế để hoàn thành thực tập chuyên môn. Nhờ thầy hướng dẫn mà em có thể khắc phục nhiều khuyết điểm của bản thân trong quá trình xây dựng báo cáo.Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay em đã có thể có đủ kiến thức để có thể thực hiện một dự án.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của mình, thế nên đề tài đồ án tốt nghiệp này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Sự nhiệt tình, tâm huyết và những lời khuyên bổ ích của thầy cô đã giúp em rất nhiều trong việc hoàn thành đồ án, từ việc lựa chọn đề tài, nghiên cứu tài liệu, lập kế hoạch thực hiện, cho đến việc hoàn thiện sản phẩm cuối cùng. Nhờ sự hướng dẫn của thầy cô, em đã có thể áp dụng kiến thức đã học vào thực tế, đồng thời trau dồi thêm kỹ năng sử dụng Godot engine và kỹ năng lập trình của em.

Em xin chân thành cảm ơn !

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

*Tp. Hồ Chí Minh, ngày tháng năm 2025*

**Giảng viên hướng dẫn**

**THS. Trần Phong Nhã**

# MỤC LỤC

[NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP i](#_bookmark0)

[LỜI CẢM ƠN iv](#_bookmark1)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN v](#_bookmark2)

[MỤC LỤC vi](#_bookmark3)

[DANH MỤC HÌNH ẢNH viii](#_bookmark4)

[DANH MỤC BẢNG BIỂU x](#_bookmark5)

[DANH MỤC CHỮ VIẾT TẮT xi](#_bookmark6)

[CHƯƠNG 1: MỞ ĐẦU 1](#_bookmark7)

* 1. [Tình hình nghiên cứu 1](#_bookmark8)
  2. [Lý do chọn đề tài 2](#_bookmark9)
  3. [Mục tiêu, nội dung, phương pháp nghiên cứu đề tài 2](#_bookmark10)
     1. [Mục tiêu nghiên cứu 3](#_bookmark11)
     2. [Nội dung nghiên cứu 3](#_bookmark12)
     3. [Phương pháp nghiên cứu 4](#_bookmark13)
  4. [Đối tượng và phạm vi nghiên cứu 5](#_bookmark14)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 8](#_bookmark15)

* 1. [Tổng quan về công nghệ godot 8](#_bookmark16)
  2. [Tổng quan về các thành phần trong godot 12](#_bookmark18)
  3. [Tổng quan về ngôn ngữ GdScript 20](#_bookmark34)
  4. [Tổng quan về thiết kế trò chơi 21](#_bookmark35)
     1. [Khái niệm game Puzzle và RPG 21](#_bookmark36)
     2. [Cách chơi 22](#_bookmark37)
     3. [Các yêu cầu về trò chơi 22](#_bookmark38)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ 26](#_bookmark40)

* 1. [Mô tả bài toán 26](#_bookmark41)
  2. [Phân tích hệ thống 27](#_bookmark42)
     1. [Sơ đồ Class 27](#_bookmark43)
     2. [Sơ đồ Phân rã chức năng 29](#_bookmark45)
     3. [Sơ đồ Usecase của Player 30](#_bookmark47)
     4. [Sơ đồ Tuần tự 32](#_bookmark49)
     5. [Sơ đồ Hoạt dộng 35](#_bookmark54)
  3. [Hệ thống tạo ra câu hỏi 39](#_bookmark59)
  4. [Kiểm thử 39](#_bookmark60)

[CHƯƠNG 4: XÂY DỰNG TRÒ CHƠI 41](#_bookmark62)

* 1. [Thiết kế giao diện 41](#_bookmark63)
  2. [Thiết kế player 46](#_bookmark75)
  3. [Thiết kế Tàu 47](#_bookmark77)
  4. [Thiết kế Puzzle 48](#_bookmark80)
  5. [Thiết kế hệ thống lưu, tải lại game và nhiệm vụ 49](#_bookmark82)
  6. [Các vật phẩm và trang bị 50](#_bookmark84)

[KẾT LUẬN VÀ KIẾN NGHỊ 52](#_bookmark89)

1. [Kết luận 52](#_bookmark90)
2. [Hạn chế 52](#_bookmark91)
3. [Kiến nghị 52](#_bookmark92)

[PHỤ LỤC 54](#_bookmark93)

[TÀI LIỆU THAM KHẢO 55](#_bookmark94)

KẾ HOẠCH THỰC HIỆN VÀ TIẾN ĐỘ NGHIÊN CỨU 56

## DANH MỤC HÌNH ẢNH

## DANH MỤC BẢNG BIỂU

## DANH MỤC CHỮ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Chữ viết tắt | Tên đầy đủ đầy đủ | Ý nghĩa |
| 1 | 3D | Three-Dimensional | Bachiều |
| 2 | UI/UX | User Interface / User Experience | Giao diện , trải nghiệm người dùng |
| 3 | GDScript | Godot Script | Ngôn ngữ lập trình của Godot |
| 4 | FPS | Frames Per Second | Số khung hình trên giây |
| 5 | BFD | Business Function Diagram | Sơ đồ phân rã chức năng |
| 6 | RPG | Role-playing games | Trò chơi nhập vai |
| 7 | GPU | Graphic Processing Unit | Bộ xử lý chuyển tăng tốc đồ họa |
| 8 | CPU | Central Processing Unit | Bộ xử lý trung tâm |

## CHƯƠNG 1: MỞ ĐẦU

### Tình hình nghiên cứu

Trong những năm gần đây, việc kết hợp trò chơi vào giáo dục đã trở thành một hướng đi tiềm năng nhằm tạo ra môi trường học tập hấp dẫn, chủ động và sáng tạo hơn cho người học. Sự thay đổi trong cách tiếp cận giáo dục – từ truyền thống sang lấy người học làm trung tâm – đòi hỏi các phương pháp giảng dạy mới mẻ và linh hoạt hơn, trong đó game-based learning (học qua trò chơi) đang nhận được sự quan tâm đặc biệt từ cả giới nghiên cứu và thực tiễn giáo dục.

Đặc biệt trong lĩnh vực tin học và lập trình – vốn thường bị xem là khô khan, trừu tượng và khó tiếp cận đối với người mới bắt đầu – việc học thông qua trò chơi không chỉ giúp giảm bớt sự căng thẳng mà còn góp phần nâng cao hiệu quả tiếp thu kiến thức. Những khái niệm như vòng lặp, biến, cấu trúc điều kiện hay thuật toán thường khiến người học cảm thấy choáng ngợp nếu được truyền đạt theo cách hàn lâm. Tuy nhiên, khi các yếu tố này được tích hợp khéo léo vào cơ chế trò chơi, người học có thể tiếp cận chúng một cách tự nhiên, dần dần hình thành tư duy lập trình thông qua các thao tác mang tính tương tác và giải trí.

Nhiều nghiên cứu và sản phẩm thực tế đã cho thấy rằng các trò chơi có yếu tố tương tác cao, phản hồi nhanh và phần thưởng rõ ràng có thể kích thích động lực học tập rất tốt. Khi người học đóng vai trò là người chơi, họ không chỉ được rèn luyện tư duy logic, khả năng giải quyết vấn đề mà còn học cách phân tích tình huống, phản xạ nhanh và làm việc có mục tiêu. Những trò chơi dạng giải đố (puzzle), mô phỏng (simulation) hoặc hành động kết hợp logic đang ngày càng được ưa chuộng trong việc giảng dạy lập trình cơ bản.

Trong bối cảnh đó, công cụ Godot Engine 4 đang nổi lên như một giải pháp lý tưởng cho việc phát triển các trò chơi phục vụ giáo dục. Với ưu điểm là mã nguồn mở, miễn phí, nhẹ và dễ tiếp cận, Godot cho phép cả những lập trình viên mới làm quen cũng có thể xây dựng trò chơi của riêng mình mà không cần đầu tư phần cứng đắt đỏ. Giao diện trực quan, hệ thống scene linh hoạt và tài liệu phong phú giúp quá trình học trở nên mạch lạc và hiệu quả. Đặc biệt, ngôn ngữ lập trình GDScript – có cú pháp tương tự Python – không chỉ dễ học mà còn đủ mạnh để hiện thực hóa các ý tưởng game phức tạp, từ đó giúp người học dần nắm vững cả kỹ năng lập trình lẫn thiết kế trò chơi.

Hiện nay, đã có một số dự án cá nhân và nhóm nhỏ ứng dụng Godot để xây dựng các trò chơi hỗ trợ học lập trình cơ bản, chẳng hạn như game điều khiển robot qua lệnh code, trò chơi dạy cấu trúc điều khiển hoặc dạng mini-game giải đố logic. Tuy nhiên, phần lớn các sản phẩm hiện tại vẫn còn mang tính thử nghiệm, thiếu sự đầu tư bài bản

về nội dung giáo dục hoặc chưa được đánh giá đầy đủ về mặt sư phạm. Mặt khác, vẫn còn ít tài liệu và công trình học thuật tại Việt Nam nghiên cứu chuyên sâu về việc kết hợp Godot với giáo dục lập trình cho người mới bắt đầu.

Do đó, việc tiếp tục nghiên cứu và phát triển một trò chơi đơn giản bằng Godot nhằm hỗ trợ học lập trình cơ bản – đặc biệt dành cho học sinh, sinh viên không chuyên hoặc người mới bắt đầu – vẫn là một hướng đi đầy triển vọng. Không chỉ giúp củng cố nền tảng tư duy lập trình theo cách tự nhiên và thú vị, hướng đi này còn góp phần làm phong phú thêm nguồn học liệu mở, thúc đẩy phong trào tự học và khuyến khích đổi mới phương pháp giáo dục tin học trong thời đại số.

### Lý do chọn đề tài

Học lập trình, đặc biệt là giai đoạn mới bắt đầu, có thể khiến nhiều người cảm thấy bối rối và mất phương hướng. Cá nhân em cũng từng trải qua cảm giác đó – khi lần đầu tiếp xúc với các khái niệm như biến, vòng lặp, điều kiện… em thấy mọi thứ thật trừu tượng và khó hình dung. Dù cố gắng học theo giáo trình, nhưng việc chỉ đọc và ghi nhớ khiến em không thực sự hiểu bản chất vấn đề.

Một lần tình cờ, em được giới thiệu một trò chơi trong đó người chơi có thể viết những khối lệnh để điều khiển hầu như mợi thứ trong trò chơi đó. Em nhận ra rằng lập trình cũng có thể rất vui, nếu mình được thực hành chứ không chỉ ngồi đọc. Từ đó, em bắt đầu hứng thú hơn với việc học, và ý tưởng kết hợp trò chơi với việc học lập trình cũng dần hình thành.

Em chọn Godot Engine 4 để thực hiện đề tài này vì nó phù hợp với khả năng hiện tại của em: dễ học, miễn phí và đặc biệt là có ngôn ngữ GDScript thân thiện, gần giống Python – ngôn ngữ em đã quen sử dụng. Việc không cần cài đặt quá nhiều công cụ phức tạp cũng giúp em tiết kiệm thời gian và tập trung vào việc phát triển nội dung game và trải nghiệm người chơi.

Điều em hướng tới không phải là một trò chơi phức tạp hay đồ họa bắt mắt, mà là một game đơn giản, dễ tiếp cận – nơi người chơi có thể học lập trình cơ bản qua những tình huống cụ thể, sinh động, và có cảm giác như đang chơi một trò chơi thực thụ chứ không phải đang “học bài”. Em tin rằng một sản phẩm như vậy không chỉ giúp người học cảm thấy dễ tiếp thu hơn, mà còn khơi dậy được sự tò mò và yêu thích lập trình một cách tự nhiên.

Chọn đề tài này vừa là cách để em học hỏi và thử thách bản thân trong việc thiết kế và lập trình trò chơi, vừa là mong muốn tạo ra một công cụ học tập thú vị cho những người mới bắt đầu, giống như em đã từng cần.

### Mục tiêu, nội dung, phương pháp nghiên cứu đề tài

### Mục tiêu nghiên cứu

Mục tiêu lớn nhất của đề tài là xây dựng một trò chơi đơn giản nhưng mang tính giáo dục cao, kết hợp giữa yếu tố giải trí và học tập nhằm hỗ trợ người chơi – đặc biệt là những người mới làm quen với lập trình – tiếp cận kiến thức một cách trực quan, sinh động và dễ ghi nhớ hơn. Thay vì tiếp thu kiến thức qua những bài giảng lý thuyết nặng tính hàn lâm, trò chơi sẽ cung cấp các tình huống thực tiễn dưới dạng thử thách để người chơi có thể “học qua hành” và rèn luyện tư duy lập trình một cách tự nhiên.

Thông qua việc chơi game, người học sẽ tiếp cận các kiến thức cơ bản như khai báo biến, sử dụng câu điều kiện, vòng lặp, và giải quyết các vấn đề logic mà không cảm thấy bị áp lực. Trò chơi sẽ đóng vai trò như một cầu nối giữa lý thuyết và thực hành, giúp củng cố kiến thức một cách nhẹ nhàng, hiệu quả và có phần thú vị hơn.

Ngoài ra, đề tài còn đặt ra mục tiêu về mặt kỹ thuật và cá nhân: việc lựa chọn Godot Engine 4 làm công cụ phát triển không chỉ là cơ hội để tiếp cận một nền tảng mã nguồn mở phổ biến, mà còn giúp người thực hiện đề tài rèn luyện kỹ năng lập trình thực tế, tổ chức logic phần mềm, quản lý nội dung, thiết kế tương tác và phát triển một sản phẩm hoàn chỉnh từ đầu đến cuối.

Cuối cùng, một mục tiêu quan trọng khác là tính ứng dụng: đề tài không chỉ là một bài tập học thuật mang tính cá nhân, mà còn hướng đến khả năng mở rộng, tham khảo hoặc phát triển tiếp bởi các sinh viên khác, các thầy cô, hoặc những người quan tâm đến việc đổi mới phương pháp dạy và học lập trình cơ bản. Hy vọng rằng đây có thể là bước đầu cho nhiều ý tưởng sáng tạo khác trong lĩnh vực giáo dục và công nghệ sau này.

### Nội dung nghiên cứu

Nội dung nghiên cứu trong đề tài được chia thành các nhóm chính, triển khai tuần tự từ lý thuyết đến thực hành, cụ thể như sau:

Tìm hiểu lý thuyết liên quan đến học lập trình: Em bắt đầu bằng việc nghiên cứu các phương pháp học lập trình hiện nay, đặc biệt là những khó khăn phổ biến mà người mới học thường gặp như: ngại viết code, không hiểu rõ khái niệm trừu tượng, hoặc thiếu động lực khi học thuần lý thuyết. Việc phân tích các yếu tố này giúp xác định được hướng tiếp cận phù hợp để đưa kiến thức vào trò chơi một cách tự nhiên, dễ hiểu.

Nghiên cứu mô hình học thông qua trò chơi (Game-based Learning): Đây là một hướng đi đang được nhiều nhà giáo dục quan tâm và áp dụng. Em tìm hiểu lý do tại sao trò chơi lại có thể tạo động lực học tập, các đặc điểm của một trò chơi giáo dục hiệu quả, và cách để cân bằng giữa tính giải trí và giá trị học thuật trong nội dung trò chơi.

Nghiên cứu công cụ Godot Engine 4: Đây là phần kỹ thuật quan trọng. Em học cách xây dựng game 3D, hiểu rõ về cấu trúc dự án trong Godot, các khái niệm như node, scene, và script, cũng như cách sử dụng ngôn ngữ GDScript để phát triển tính năng và tương tác trong game.

Thiết kế gameplay gắn với kiến thức: Đây là bước quan trọng nhằm tích hợp nội dung học lập trình vào trò chơi. Các màn chơi được thiết kế không chỉ để thử thách kỹ năng chơi mà còn khéo léo đưa người chơi tiếp cận và thực hành các khái niệm lập trình. Chẳng hạn, để vượt qua một chướng ngại vật, người chơi có thể cần viết một đoạn lệnh đúng, hoặc giải một câu đố logic liên quan đến cấu trúc điều kiện hoặc vòng lặp.

Kiểm thử và điều chỉnh sản phẩm: Sau khi xây dựng xong bản demo của trò chơi, em tiến hành kiểm thử với một nhóm nhỏ người chơi, ghi nhận phản hồi và đánh giá về mức độ dễ hiểu, tính hấp dẫn và hiệu quả truyền đạt kiến thức. Dựa vào kết quả này, em thực hiện các điều chỉnh phù hợp để nâng cao chất lượng trò chơi.

### Phương pháp nghiên cứu

Để đảm bảo đề tài được thực hiện một cách có hệ thống và hiệu quả, em đã vận dụng kết hợp nhiều phương pháp nghiên cứu khác nhau:

Phương pháp nghiên cứu tài liệu: Đây là bước đầu tiên giúp em có cái nhìn nền tảng và định hướng rõ ràng hơn cho đề tài. Em tìm hiểu các tài liệu học thuật, giáo trình lập trình cơ bản, sách chuyên ngành về thiết kế game, cũng như các bài báo và nghiên cứu liên quan đến giáo dục bằng trò chơi. Ngoài ra, em còn tham khảo các video, tài nguyên mở và diễn đàn của cộng đồng Godot để học hỏi kinh nghiệm từ những người đi trước.

Phương pháp thực nghiệm: Em trực tiếp thực hiện các bước xây dựng trò chơi – từ thiết kế ý tưởng, xây dựng kịch bản, tạo nhân vật và môi trường, cho đến lập trình chức năng và thử nghiệm gameplay. Trong suốt quá trình này, em không ngừng kiểm tra, điều chỉnh và khắc phục lỗi, qua đó tích lũy thêm kỹ năng lập trình, quản lý dự án nhỏ và xử lý vấn đề thực tiễn.

Phương pháp quan sát và đánh giá thực tiễn: Khi bản thử nghiệm trò chơi hoàn thành, em mời một số người chơi – chủ yếu là bạn bè – tham gia trải nghiệm trò chơi và ghi nhận phản hồi của họ. Thông qua việc quan sát cách họ chơi, những chỗ họ gặp khó khăn hoặc không hiểu, cùng với những góp ý trực tiếp, em có cơ sở thực tiễn để đánh giá hiệu quả truyền đạt kiến thức của trò chơi, từ đó điều chỉnh cho phù hợp hơn với mục tiêu ban đầu.

### Đối tượng và phạm vi nghiên cứu

1. **Đối tượng nghiên cứu**

Đề tài tập trung vào việc nghiên cứu và xây dựng một trò chơi 3D đơn giản với mục tiêu hỗ trợ người chơi – đặc biệt là những người mới bắt đầu học lập trình – tiếp cận các kiến thức cơ bản một cách dễ hiểu, nhẹ nhàng và mang tính thực hành cao. Thay vì học qua các bài giảng lý thuyết khô khan, người học sẽ được tiếp cận kiến thức thông qua các tình huống cụ thể trong game, nơi họ phải vận dụng tư duy lập trình để vượt qua thử thách và hoàn thành nhiệm vụ.

Đối tượng nghiên cứu chính của đề tài là cách thiết kế một trò chơi mang tính giáo dục, không chỉ nhằm mục đích giải trí mà còn có giá trị hỗ trợ học tập rõ ràng. Trò chơi sẽ được xây dựng với các màn chơi xoay quanh những nội dung lập trình cơ bản như: khai báo và sử dụng biến, câu điều kiện (if – else), vòng lặp (for, while), cũng như các tình huống giải quyết vấn đề logic đơn giản. Các khái niệm này sẽ không được trình bày dưới dạng lý thuyết thông thường, mà sẽ được lồng ghép khéo léo vào gameplay để người chơi tiếp cận và áp dụng một cách tự nhiên.

Bên cạnh nội dung kiến thức, đề tài còn nghiên cứu và áp dụng Godot Engine 4 **–** một trong những công cụ phát triển game mã nguồn mở phổ biến hiện nay. Godot có ưu điểm là dễ học, nhẹ, miễn phí và hỗ trợ mạnh mẽ cho phát triển game 3D. Việc sử dụng công cụ này giúp quá trình xây dựng trò chơi diễn ra thuận lợi hơn, đồng thời giúp người thực hiện đề tài rèn luyện được tư duy lập trình thực tế, khả năng tổ chức nội dung và triển khai tính năng trong một sản phẩm phần mềm hoàn chỉnh.

Ngoài hai yếu tố chính là nội dung học lập trình và công cụ phát triển game, đề tài cũng chú trọng đến việc nghiên cứu trải nghiệm của người chơi trong quá trình tương tác với trò chơi. Cụ thể, cần quan sát và phân tích cách người chơi tiếp nhận kiến thức, phản ứng với các tình huống học tập lồng ghép trong game, cũng như mức độ hứng thú và khả năng ghi nhớ kiến thức sau khi chơi. Những yếu tố này sẽ là cơ sở để đánh giá hiệu quả giáo dục của trò chơi, đồng thời giúp điều chỉnh, cải tiến sản phẩm theo hướng thân thiện và dễ tiếp cận hơn cho người học.

Tóm lại, đề tài không chỉ nghiên cứu cách xây dựng một trò chơi có yếu tố giáo dục mà còn hướng đến việc khai thác tiềm năng của game như một công cụ học tập sáng tạo, dễ tiếp cận và hiệu quả. Đối tượng nghiên cứu vì vậy bao gồm cả nội dung kiến thức được truyền tải, phương pháp truyền tải thông qua gameplay, công cụ hỗ trợ phát triển trò chơi, cũng như phản hồi và tương tác của người chơi trong suốt quá trình trải nghiệm game.

### Phạm vi nghiên cứu

Do đề tài được thực hiện trong khuôn khổ một môn học cụ thể, với thời gian và nguồn lực có hạn, nên phạm vi nghiên cứu được thu hẹp lại để đảm bảo tính khả thi và phù hợp với năng lực thực hiện. Thay vì hướng đến một sản phẩm phức tạp hay đa nền tảng, trò chơi sẽ được xây dựng tập trung vào tính đơn giản, dễ hiểu, dễ tiếp cận và mang tính giáo dục rõ ràng.

Trò chơi sẽ bao gồm một số màn chơi cơ bản, mỗi màn được thiết kế xoay quanh một khái niệm lập trình cụ thể như: khai báo biến, sử dụng câu điều kiện (if – else), viết vòng lặp đơn giản hoặc giải quyết các bài toán logic theo kiểu “học mà chơi, chơi mà học”. Những khái niệm này được chọn lọc từ các nội dung lập trình nhập môn, dành cho người mới bắt đầu, và được truyền tải thông qua cách tiếp cận trực quan, sinh động thay vì chỉ đơn thuần là lý thuyết khô khan.

Về nền tảng triển khai, trò chơi chỉ được phát triển để chạy trên máy tính cá nhân, cụ thể là các hệ điều hành phổ biến như Windows hoặc Linux. Các nền tảng khác như thiết bị di động**,** trình duyệt web hay máy chơi game không nằm trong phạm vi nghiên cứu ở thời điểm hiện tại. Điều này nhằm đảm bảo tính ổn định và kiểm soát được các yếu tố kỹ thuật trong quá trình phát triển và thử nghiệm.

Ngoài ra, các kỹ thuật lập trình phức tạp hoặc yêu cầu chuyên môn cao như trí tuệ nhân tạo, hệ thống mạng đa người chơi, hệ thống lưu trữ dữ liệu người dùng hay xây dựng thuật toán điểm số nâng cao. Cũng sẽ không được đề cập trong phạm vi đề tài. Trò chơi được xây dựng chủ yếu với mục tiêu học tập và minh họa, thay vì hướng đến một sản phẩm thương mại hoàn chỉnh.

Kiến thức lập trình được đề cập trong game cũng giới hạn ở mức nhập môn, chỉ bao gồm những nội dung cốt lõi và phổ biến dành cho người mới học. Các thuật ngữ chuyên ngành, kiến thức chuyên sâu về cấu trúc dữ liệu, hướng đối tượng, hoặc xử lý đa luồng sẽ không nằm trong phạm vi nội dung chính của trò chơi.

Mục tiêu của đề tài không phải là tạo ra một trò chơi hoàn thiện về mọi mặt, mà là xây dựng một sản phẩm có tính ứng dụng, dễ sử dụng, và có thể đóng vai trò như một công cụ hỗ trợ học lập trình hiệu quả, giúp người mới học tiếp cận kiến thức một cách trực quan, sinh động và chủ động hơn so với các phương pháp truyền thống như đọc sách hay nghe giảng đơn thuần.

Nói cách khác, đây là một bước thử nghiệm nhỏ nhưng mang tính thực tiễn cao, với mong muốn góp phần vào việc đổi mới phương pháp học lập trình cơ bản theo hướng trực quan và trải nghiệm hơn, qua đó tạo nền tảng để phát triển các sản phẩm tương tự trong tương lai nếu có điều kiện mở rộng.

Kết luận chương: Chương 1 đã trình bày rõ các nội dung nền tảng cho đồ án tốt nghiệp, bao gồm lý do chọn đề tài, mục tiêu chính – cụ thể, kết quả kỳ vọng cũng như phương pháp tiếp cận được sử dụng trong quá trình thực hiện. Việc lựa chọn đề tài làm game puzzle 3D bằng Godot không chỉ là sự lựa chọn mang tính cá nhân mà còn là định hướng học tập thực tiễn, phù hợp với xu thế phát triển của ngành công nghệ và game hiện nay.Các nội dung được trình bày trong chương này sẽ là cơ sở để phát triển các chương tiếp theo của đồ án, bắt đầu từ chương 2 – Cơ sở lý thuyết, trong đó sẽ làm rõ hơn về công cụ Godot Engine, ngôn ngữ lập trình GDScript và các thành phần kỹ thuật hỗ trợ việc phát triển game.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### Tổng quan về công nghệ godot

Godot là một phần mềm phát triển trò chơi miễn phí và mã nguồn mở, ngày càng được ưa chuộng trong cộng đồng lập trình game trên toàn thế giới. Được phát triển từ những năm 2007 và chính thức ra mắt công khai vào năm 2014, Godot không ngừng cải tiến qua từng phiên bản để phục vụ tốt hơn cho cả người mới bắt đầu lẫn các lập trình viên chuyên nghiệp. Điểm đặc biệt là Godot không chỉ hỗ trợ lập trình game 2D mà còn có thể tạo ra các trò chơi 3D với chất lượng cao – trong khuôn khổ đề tài này, trọng tâm sẽ là phát triển trò chơi 3D đơn giản để phù hợp với mục tiêu hỗ trợ học lập trình.



*Hình 2.1: Godot Engine*

Phiên bản được sử dụng trong đề tài là Godot Engine 4, một bản nâng cấp đáng kể so với các phiên bản trước đó. Với Godot 4, khả năng xử lý hình ảnh 3D đã được tối ưu hóa, hiệu suất hoạt động cũng tốt hơn, giao diện làm việc được cải thiện, đồng thời bổ sung thêm nhiều tính năng hữu ích giúp quá trình làm game trở nên thuận tiện hơn. Người dùng có thể dễ dàng kéo – thả các đối tượng, tổ chức chúng theo cấu trúc cây node rõ ràng, và lập trình hành vi cho từng thành phần trong game mà không cần mất quá nhiều thời gian làm quen.

Một trong những lý do khiến Godot trở nên gần gũi với người học là việc sử dụng ngôn ngữ lập trình GDScript. Đây là một ngôn ngữ được chính đội ngũ Godot phát triển, với cú pháp đơn giản, dễ hiểu, rất giống với Python – một ngôn ngữ phổ biến trong giảng dạy lập trình hiện nay. Điều này giúp người mới học dễ dàng tiếp cận, đọc hiểu và viết mã mà không cảm thấy quá khó khăn. Ngoài GDScript, Godot cũng hỗ trợ thêm các ngôn ngữ khác như C#, C++, và cả visual scripting (lập trình không cần viết mã, chỉ kéo – thả khối lệnh), giúp người dùng có nhiều lựa chọn hơn tùy theo nhu cầu.

Một điểm mạnh khác của Godot là khả năng xuất bản trò chơi ra nhiều nền tảng

khác nhau như Windows, Linux, macOS, Android, iOS, hoặc thậm chí là chạy trực tiếp trên trình duyệt web thông qua HTML5. Điều này rất có lợi cho các nhà phát triển khi muốn chia sẻ sản phẩm của mình đến nhiều người dùng mà không cần tốn công chỉnh sửa quá nhiều.

Không giống như nhiều phần mềm làm game khác yêu cầu trả phí hoặc mua bản quyền, Godot hoàn toàn miễn phí và không tính phí thương mại nếu người dùng phát hành sản phẩm ra thị trường. Điều này đặc biệt phù hợp với sinh viên, những người đang học lập trình, hoặc các nhà phát triển độc lập có ít kinh phí. Hơn nữa, Godot có một cộng đồng người dùng rất tích cực, thường xuyên chia sẻ kiến thức, hướng dẫn, tài nguyên và giải đáp thắc mắc. Điều đó giúp việc học và phát triển game bằng Godot trở nên nhẹ nhàng và có định hướng hơn, nhất là khi người dùng còn chưa có nhiều kinh nghiệm.

Tóm lại, Godot Engine 4 là một công cụ làm game hiện đại, dễ tiếp cận, linh hoạt và hoàn toàn miễn phí. Với ưu điểm mạnh về phát triển game, hỗ trợ học lập trình, và cộng đồng hỗ trợ lớn, Godot là lựa chọn lý tưởng để sử dụng trong đề tài nghiên cứu và xây dựng một trò chơi học lập trình thân thiện, dễ dùng, nhưng vẫn mang lại trải nghiệm thú vị và hữu ích cho người chơi

### Vì sao lại chọn Godot?

Hiện tại, có khá nhiều Engine đã tạo được danh tiếng của mình trong việc hỗtrợ thiết kế và xây dựng game. Nó rút ngắn quá trình, tiết kiệm thời gian và công sức cho người sản xuất. Hãy cùng điểm qua một số engine nổi tiếng mà em tìm hiểu được.

Unity Engine là một môi trường phát triển trò chơi cực kì mạnh mẽ và đa dụng đã được sử dụng rộng rãi trong cả ngành công nghiệp thiết kế game, thậm chí còn được mang vào giảng dạy lập trình [1]. Unity có những điểm cộng to lớn làm nên sức hút mạnh mẽ của nó:

Đa nền tảng: Unity cho phép người sản xuất phát triển trò chơi của mình trên nhiều nền tảng khác nhau như PC, smartphone, tablet, các thiết bị kĩ thuật số thực tế ảo (VR) và nhiều hệ thống máy móc dành riêng cho việc chơi game khác nữa. Điều này giải quyết xuất sắc vấn đề xung đột hay không tương thích giữa game với thiết bị của người dùng. Khi game có chất lượng, ưu điểm này của Unity sẽ giúp nó được phủ sóng nhanh chóng và mạnh mẽ, tổng hợp được số lượng người chơi khủng.

Cộng đồng lớn: Nó như thể là một điều tất yếu! Sự nổi tiếng và tínhnăng hỗ trợ của Unity giúp nó xây dựng một cộng đồng lập trình viên,nhà thiết kế và game thủ rất lớn, ưu điểm này giúp những người mớitiếp cận dễ dàng tìm kiếm sự hỗ trợ, tài liệu và giải đáp những câu hỏi, thắc mắc trong quá trình sử dụng và vận hành.

Asset Store: Asset Store của Unity cung cấp cho người xây dựng game một thư

viện lớn các tài nguyên, script và plug-in giúp họ tăng tốc quá trình thiết kế và phát triển.

Dễ học: theo như đánh giá từ các lập trình viên, Unity có giao diệndễ sử dụng và hỗ trợ nhiều ngôn ngữ lập trình như C# và JavaScript,điều này hỗ trợ cho những người mới bắt đầu có thể nắm bắt nhanh chóng.

Render chất lượng cao: 1 game chất lượng phải thu hut game thủ ngay từ phần nhìn. Unity cung cấp hệ thống render mạnh mẽ giúp người thiết kế game dễ dàng hơntrong việc tạo ra hình ảnh chất lượng cao, cho dù họ đang xây dựng trò chơi 2D hay 3D.

Unreal Engine cũng là một trong những engine phát triển trò chơi và ứng dụng 3D thuộc top đầu trên thị trường. Không kém cạnh Unity, Unreal cũng có những ưu điểm lớn của nó.

Đồ họa và hiệu ứng chất lượng cao: một trong những điểm cộng đầu tiên tạo nên danh tiếng của Unreal Engine là khả năng tạo ra đồ họa và hiệu ứng 3D chất lượng cao, nổi tiếng là trong trò chơi AAA. Hệ thống render của Unreal cho phép người thiết kế tạo ra hình ảnh đẹp mặt với hiệu ứng ánh sáng sắc nét và đổ bóng phức tạp.

Cộng đồng lớn và tài liệu phong phú: cũng giống như Unnity, Unreal Engine xây dựng được một cộng đồng lập trình viên rất lớn, thư viện hướng dẫn với rất nhiều tài liệu và khóa học, nhóm hỗ trợ được xây dựng giúp những người mới sử dụng có thể dễ dàng tiếp cận để học và giải quyết vấn đề

Đa nền tảng: Unreal hỗ trợ xây dựng game tương thích với nhiều nền tảng bao gồm PC, console, điện thoại di động, VR và rất nhiều hệ thống khác.

Blueprint Visual Scripting: tính năng nổi bật này là một công cụ được Unreal cung cấp cho phép người xây dựng lập trình mà không cần phải viết mã, nó thích hợp cho người mới bắt đầu hoặc các nghệ sĩ đồ họa dễ dàng tạo nên kết quả tốt hơn và nhanh hơn mà không cần can thiệp quá nhiều.

Có thể nói, những Engine được nêu ở trên hiện đang rất thịnh hành và được ưa chuộng sử dụng nhiều, hỗ trợ đắc lực trong việc thiết kế và xây dựng một trò chơi điện tử.

Nhưng đối với đề tài này, chúng tôi quyết định xây dựng “game 2D Action & Adventure” với sự hỗ trợ của Godot! Bởi lẽ:

Miễn phí và mã nguồn mở: Godot Engine là một dự án mã nguồn mở hoàn toàn miễn phí, giúp tiết kiệm chi phí phát triển và phù hợp cho cả những dự án cá nhân lẫn doanh nghiệp. Với một nhóm các cá nhân là sinh viên Công nghệ Thông tin – đặc điểm phù hợp này là điểm cộng đầu tiên.

Dễ học và dễ sử dụng: Theo những đánh giá từ các lập trình viên, giao diện của Godot Engine được thiết kế trực quan, ngôn ngữ lập trình GDScript tương tự như Python, dễ học và dễ sử dụng, nó rất phù hợp với những người mới bắt đầu làm quen với việc thiết kế và xây dựng game.

Hỗ trợ nhiều nền tảng: Gdot hỗ trợ nhiều nền tnagr như Windows, macOS, Linux, Android, iOS, HTML5 và khá nhiều nền tảng khác, giúp người sử dụng phát triển trò chơi dễ dàng mà không quá lo ngại vấn đề xung đột [2].

Kích thước nhẹ và hiệu năng tốt: Godot có kích thước nhẹ, phù hợp cho quá trình sử dụng, nó làm việc tốt trên nhiều loại máy tính và thiết bị di dộng. Đồng thời nó có khả năng tối ưu hóa hiệu suất để đảm bảo cho người chơi game một trải nghiệm mượt mà.

Cộng đồng tài liệu phong phú: dù chưa thể nói là có không giạn lớn bằng cộng đồng Unreal hay Unity, nhưng Godot cũng đã kịp tạo cho mình một cộng đồng sáng tạo và nhiệt tình, cùng với tài liệu ví dụ trực quan hỗ trợ tốt nhất cho người phát triển.

### So sánh với các Engine khác

Mặc dù là các bậc đàn anh trong số những engine hiện hành được ưu tiên nhất, nhưng để so sánh về độ phù hợp dành cho cá nhân – cụ thể là nhóm sinh viên chúng tôi, Godot lại là engine tạo được bước đà tốt nhất.

Kích thước ứng dụng: Unity – những ứng dụng hay game được phát triển từ engine này có kích thước lớn hơn so với với engine khác, đặc biệt là khi bạn sử dụng nhiều tài nguyên và plug-in từ Asset Store. Unreal cũng đòi hỏi ở người phát triển một hệ thống máy tính đủ mạnh và hao tốn tài nguyên hệ thống cao để phát triển và chạy các trò chơi cũng như ứng dụng 3D. Trong khi đó, Godot lại giải quyết khá ổn vấn đề này. Với kích thước nhẹ, dung lượng thấp, phù hợp để làm việc tốt trên nhiều loại máy tính cũng như thiết bị di động mà chúng tôi có thể đầu tư được.

Hiệu năng: Hiện tại, mặc dù Unity có khả năng tối ưu hóa hiệu năng, tuy nhiên, việc thiết kế và phát triển trò chơi từ engine này đòi hỏi người phát triển có kiến thức và một trình độ tốt về tối ưu hóa để giảm thiểu tối đa hiện tượng giật lag cũng như tính toán để tối thiểu mức tiêu tốn tài nguyên. Unreal Engine cũng không khác biệt hơn Unity về vấn đề này. Trong khi đó, Godot lại có thể đảm bảo trải nghiệm tốt và mượt mà hơn.

Cấu trúc dự án: đối với những người mới bắt đầu, cấu trúc dự án của Unity hay Unreal thực sự là một vấn đề khó khăn bởi nó không rõ ràng, đặc biệt là trong các dự án lớn, gây trở ngại trong việc quản lý mã nguồn.

Ngược lại, Godot với một giao diện trực quan và dễ sử dụng, dễ quản lý, sẽ là

điểm cộng phù hợp để lựa chọn.

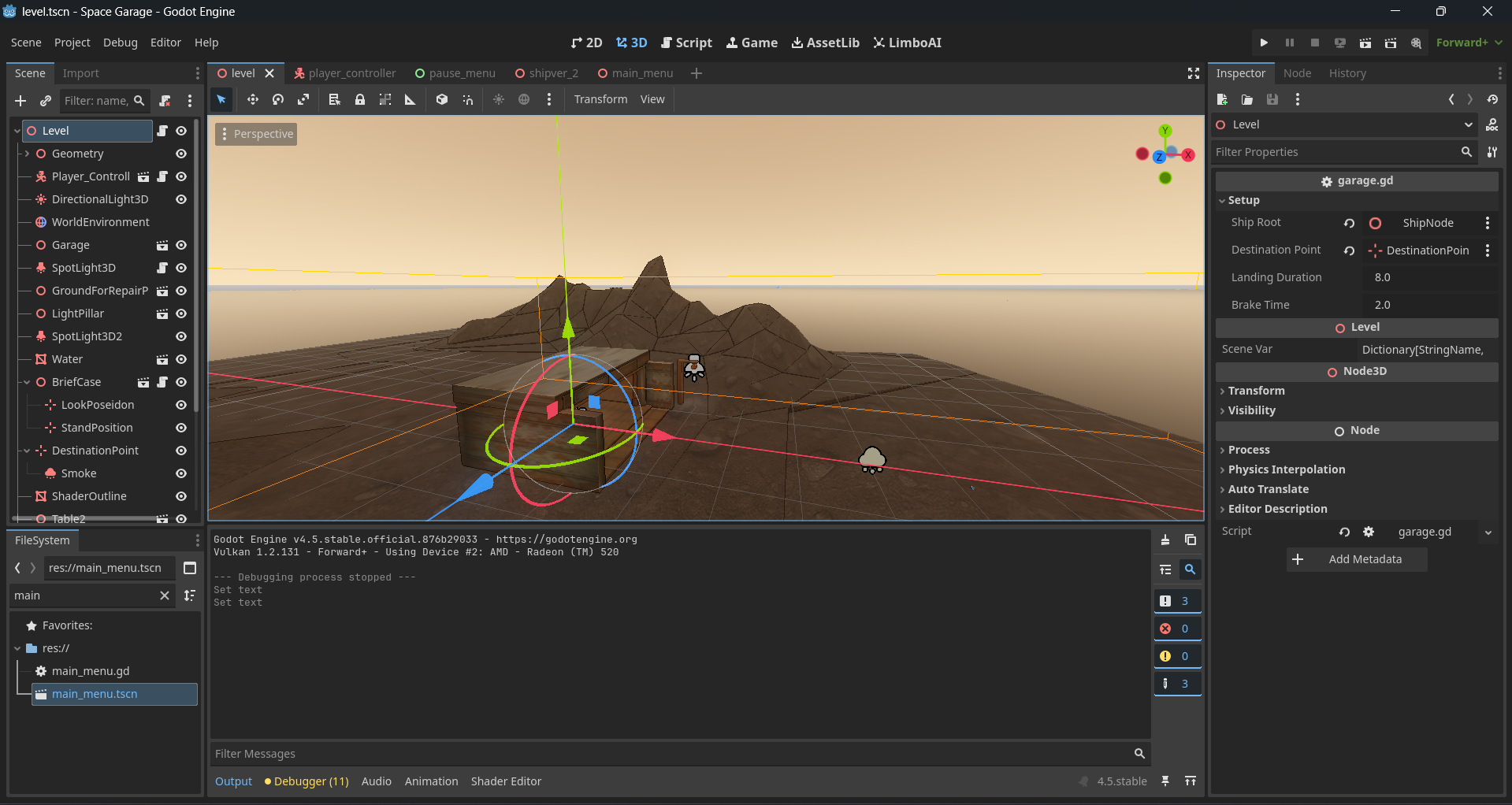
Giấp phép và kinh phí: Unity và Unreal đều là các engine có thể cung cấp phiên bản miễn phí, tuy nhiên, muốn khai thác tốt mọi chức năng để tạo ra một sản phẩm hoàn chỉnh là có giới hạn. Để loại bỏ những hạn chế và sử dụng các tính năng mạnh mẽ hơn, người phát triển game cần phải trả chi phí cho các điều khoản này. Nhưng Godot lại là một engine miễn phí hoàn toàn.

Hạn chế kiểm soát và giới hạn nền tảng: Trong một số trường hợp, Unity có thể sẽ hạn chế sự kiểm soát của bạn trong việc xử lý nhiều khía cạnh của trò chơi, đặc biệt là trong việc xử lý sâu các đối tượng và sự tương tác phức tạp. Trong khi đó, Unreal lại giới hạn khi phát triển trò chơi cho một số thiết bị di dộng có cấu hình thấp. Với tình hình sử dụng thiết bị di dộng cực kỳ nhiều như hiện nay, giới hạn này là một bất lợi. Godot sẽ xử lý ổn thỏa vấn đề này mà không có sự hạn chế kiểm soát hay giới hạn nền tảng.

### Tổng quan về các thành phần trong godot

Scene trong Godot không chỉ đơn giản là một “màn chơi” như cách hiểu truyền thống trong game, mà thực chất là một đơn vị cấu trúc độc lập, có thể đại diện cho bất kỳ thành phần nào trong trò chơi – từ một nhân vật, một vũ khí, một đoạn UI, cho đến một bản đồ hoàn chỉnh. Một scene có thể bao gồm nhiều node và có thể được lồng ghép vào các scene khác, giúp tạo nên cấu trúc phân tầng rõ ràng và dễ quản lý [2].

Chính nhờ cơ chế này mà quá trình phát triển game trở nên cực kỳ linh hoạt: người lập trình có thể tạo ra một scene “nhân vật”, sau đó chèn vào bất kỳ map nào mà không cần viết lại logic. Điều này tối ưu hóa khả năng tái sử dụng và tăng tốc độ phát triển game.



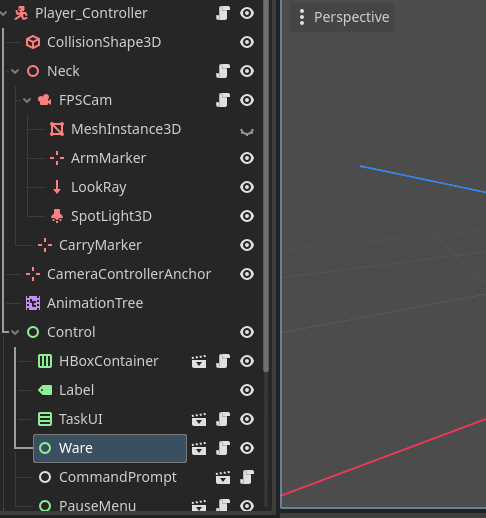
*Hình 2.2: Thành phần của Scenes*

Bạn cũng có thể lồng các Scene, tức là đưa một scene này làm thành phần con trong một scene khác. Đây là một tính năng rất mạnh của Godot, giúp tái sử dụng tài nguyên, tiết kiệm thời gian phát triển và quản lý dự án hiệu quả hơn.

Ví dụ cụ thể, bạn có thể xây dựng một scene riêng cho nhân vật chính, bao gồm sprite, animation, logic di chuyển. Sau đó bạn tạo một scene bản đồ cấp độ. Thay vì thiết kế lại nhân vật trong từng cấp độ, bạn chỉ cần kéo scene nhân vật vào trong scene cấp độ như một node con. Như vậy, nhân vật sẽ xuất hiện đầy đủ với toàn bộ chức năng đã thiết kế sẵn.

Việc lồng scene không chỉ áp dụng cho nhân vật mà còn cho mọi thành phần khác như: quái vật, cửa ra vào, hộp thoại, giao diện người dùng. Điều này mang lại sự linh hoạt cực kỳ lớn

Node là khối xây dựng cơ bản nhất trong Godot. Mỗi node đảm nhiệm một vai trò nhất định, như hiển thị hình ảnh (Sprite), xử lý âm thanh (AudioStreamPlayer), kiểm tra va chạm (CollisionShape), điều khiển logic (Script) hoặc phục vụ điều hướng người dùng (Button, Label,...). Mỗi scene là một cây node (scene tree), trong đó các node con được tổ chức theo dạng phân cấp, tạo nên các mối quan hệ cha – con. Mối quan hệ này cho phép node cha kiểm soát hành vi của các node con, đồng thời đơn giản hóa việc di chuyển, xoay, hoặc ẩn/hiện cả nhóm node chỉ bằng thao tác trên node cha.



*Hình 2.3: Thành phần Nodes*

Lưu ý cách các nodes và scene trông giống nhau trong trình chỉnh sửa. Khi bạn lưu một cây nodes làm scene, nó sẽ hiển thị dưới dạng một nodes duy nhất, với cấu trúc

bên trong của nó được ẩn trong trình chỉnh sửa. Godot cung cấp một thư viện phong phú về các loại nodes cơ sở mà bạn có thể kết hợp và mở rộng để xây dựng những loại nodes mạnh hơn. Giao diện 2D, 3D hoặc người dùng, bạn sẽ thực hiện hầu hết mọi việc với các nút này.

The scene tree là tất cả các scenes trong trò chơi của bạn đều tập hợp lại trong scene tree, nghĩa đen là một scene tree . Và vì scene là cây của nút, nên cây scene cũng là cây của nút. Nhưng sẽ dễ dàng hơn khi nghĩ về trò chơi của bạn dưới dạng các cảnh vì chúng có thể đại diện cho các nhân vật, vũ khí, cánh cửa hoặc giao diện người dùng của bạn.

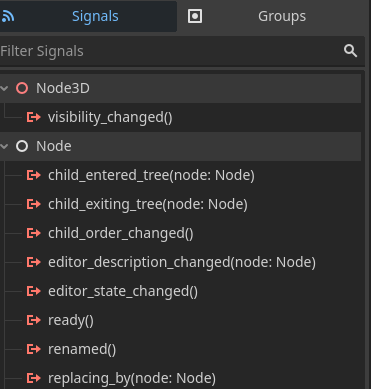
Trong quá trình chạy game, Godot sẽ kết hợp tất cả các scene được sử dụng để tạo nên một “Scene Tree toàn cục” – chính là không gian hoạt động chính của trò chơi. Từ đây, các node có thể tương tác lẫn nhau thông qua các tín hiệu (signals), gọi phương thức hoặc chia sẻ tài nguyên. Việc theo dõi và quản lý scene tree một cách có tổ chức là yếu tố quan trọng giúp trò chơi hoạt động ổn định, hạn chế lỗi logic, và dễ mở rộng về sau.



*Hình 2.4: Thành phần The scene tree*

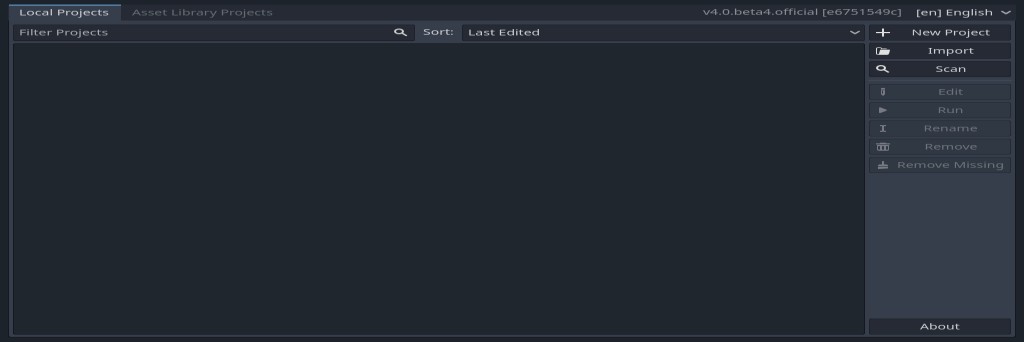
Signals là hệ thống “giao tiếp không dây” giữa các node. Thay vì viết mã gọi trực tiếp giữa các node (gây phụ thuộc chặt và khó bảo trì), signals cho phép một node phát ra thông báo khi có sự kiện xảy ra (ví dụ: khi người chơi nhấn nút, khi kẻ địch chết, khi nhặt được vật phẩm,...), và các node khác có thể “lắng nghe” tín hiệu này để phản hồi lại. Cách tiếp cận này giúp code linh hoạt hơn, dễ kiểm soát hơn, và đặc biệt phù hợp với kiến trúc hướng sự kiện trong game.

Signals là Các node phát ra tín hiệu khi một số sự kiện xảy ra. Tính năng này cho phép bạn làm cho các node giao tiếp mà không cần nối cứng chúng bằng mã. Nó mang lại cho bạn rất nhiều sự linh hoạt trong cách bạn cấu trúc các cảnh của mình.



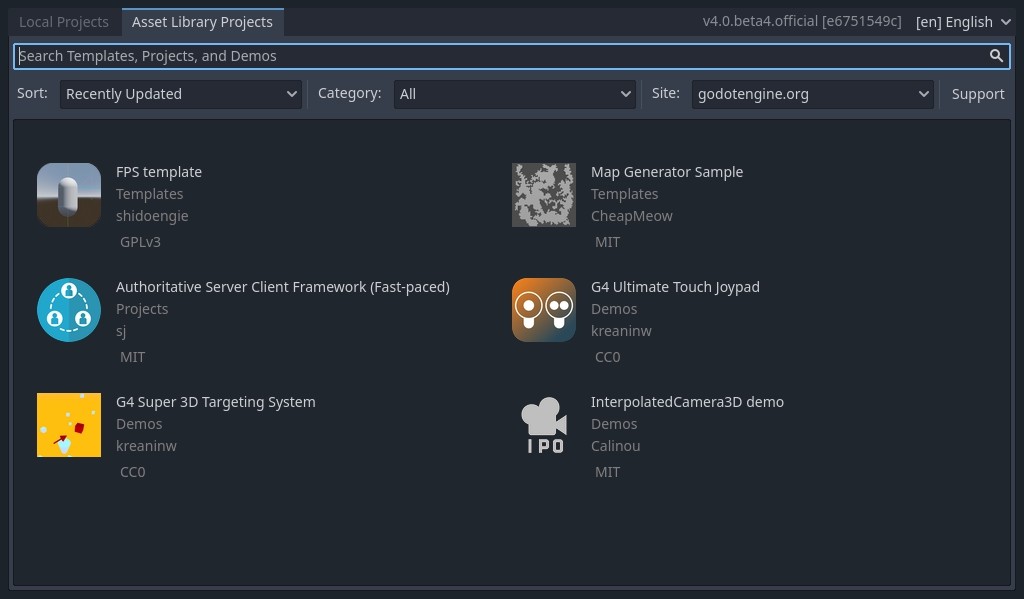
*Hình 2.5: Thành phần Signals*

The Project Manager(Quản lý dự án) :Khi bạn khởi chạy Godot, cửa sổ đầu tiên bạn nhìn thấy là Trình quản lý dự án. Trong tab mặc định Local Project, bạn có thể quản lý các dự án hiện có, nhập hoặc tạo dự án mới, v.v.



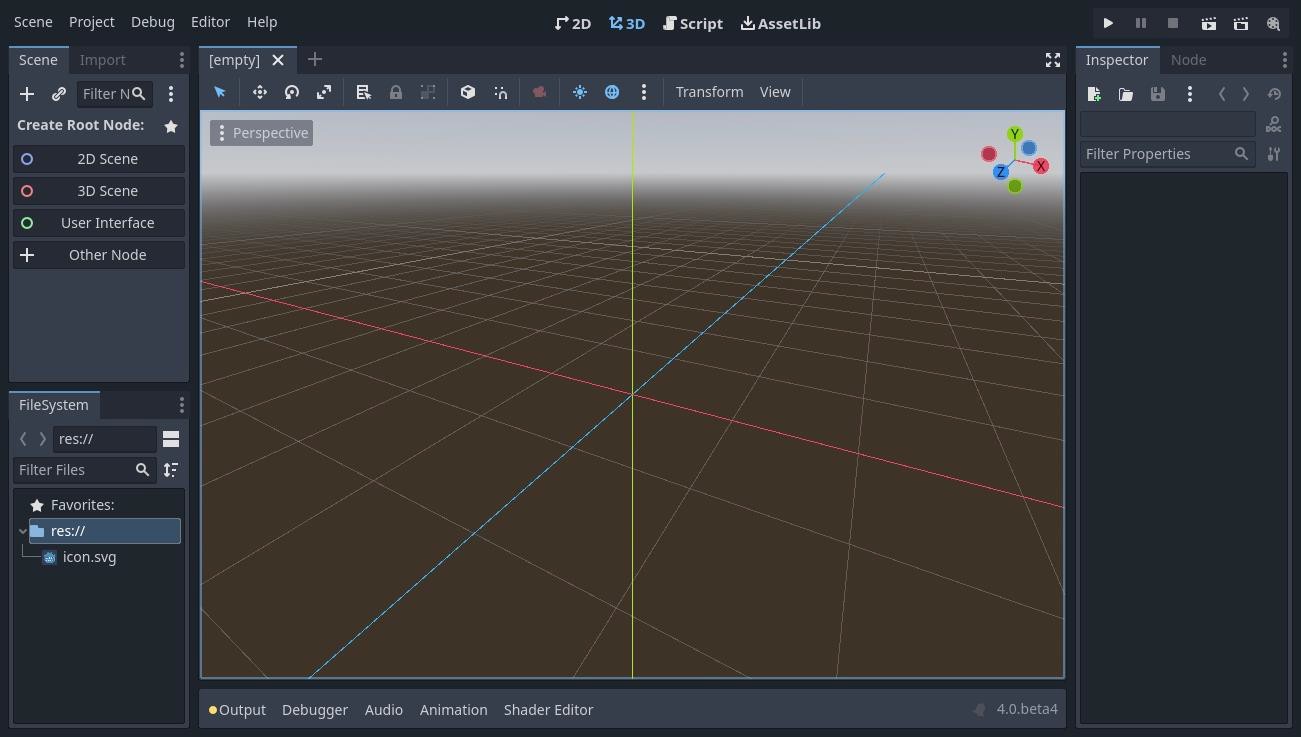
*Hình 2.6: Thành phần The Project Manager (a)*

Ở đầu cửa sổ, có một tab khác có tên “Asset Library”. Bạn có thể tìm kiếm các dự án demo trong thư viện tài sản nguồn mở, bao gồm nhiều dự án do cộng đồng phát triển.



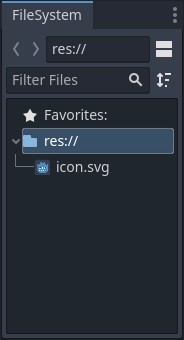
*Hình 2.7: Thành phần The Project Manager (b)*

Trình soạn thảo của GoDot Là khi bạn mở một dự án mới hoặc một dự án hiện có, giao diện của trình soạn thảo sẽ xuất hiện. Hãy nhìn vào các lĩnh vực chính của nó.



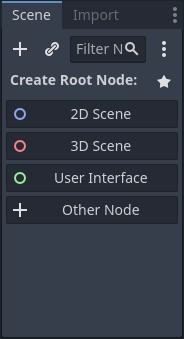
*Hình 2.8: Trình Soạn Thảo Của GoDot (a)*

Thẻ hệ thống tập tin liệt kê các tệp dự án của bạn, bao gồm tập lệnh, hình ảnh, mẫu âm thanh.



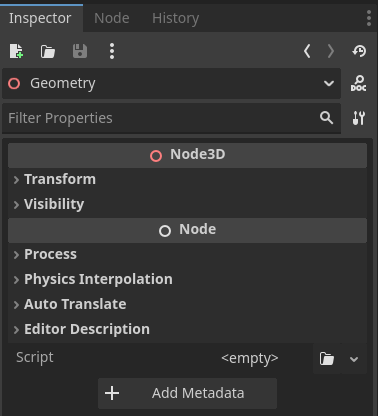
*Hình 2.9: Hệ thống tập tin*

Thẻ Scene liệt kê các nút của scenes đang hoạt động.



*Hình 2.10: Thẻ Scene*

Thẻ Inspector cho phép bạn chỉnh sửa các thuộc tính của nút đã chọn.



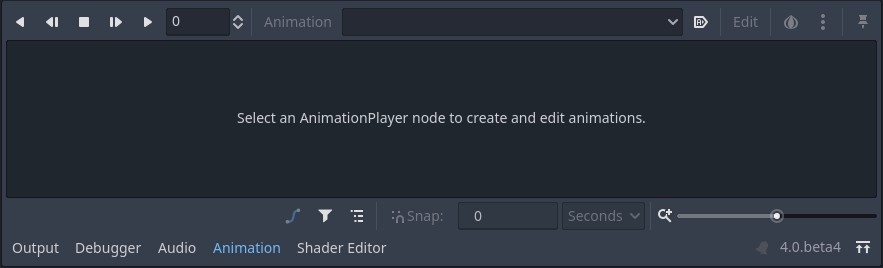
*Hình 2.11: Thẻ Inspector*

Bảng điều khiển phía dưới, nằm bên dưới khung nhìn, là máy chủ dành cho bảng điều khiển gỡ lỗi, trình chỉnh sửa hoạt ảnh, bộ trộn âm thanh, v.v. Chúng có thể chiếm không gian quý giá, đó là lý do tại sao chúng được gấp lại theo mặc định.



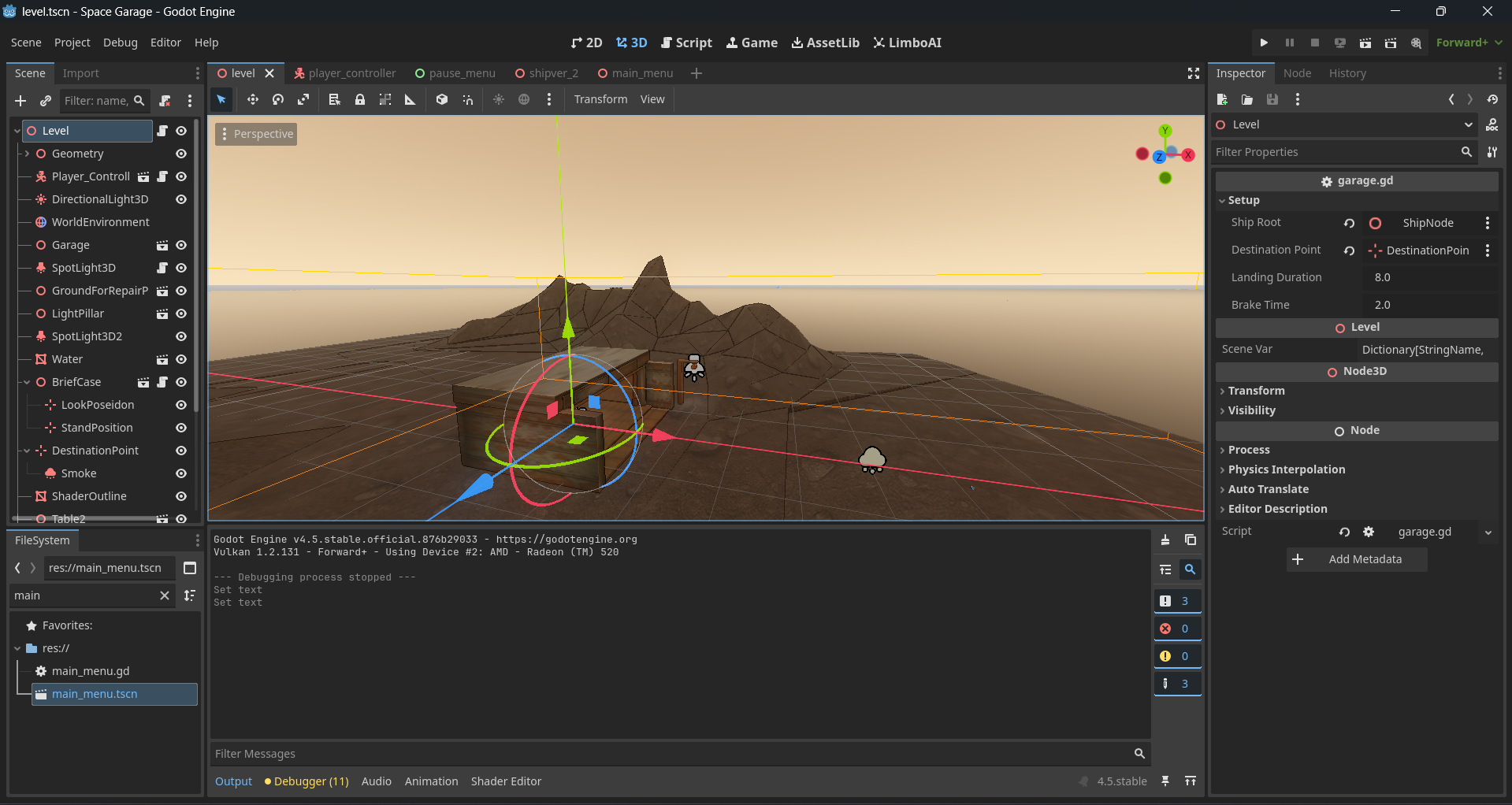
*Hình 2.12: Thẻ Control*

Khi bạn nhấp vào một cái, nó sẽ mở rộng theo chiều dọc. Dưới đây, bạn có thể thấy trình chỉnh sửa hoạt ảnh được mở.



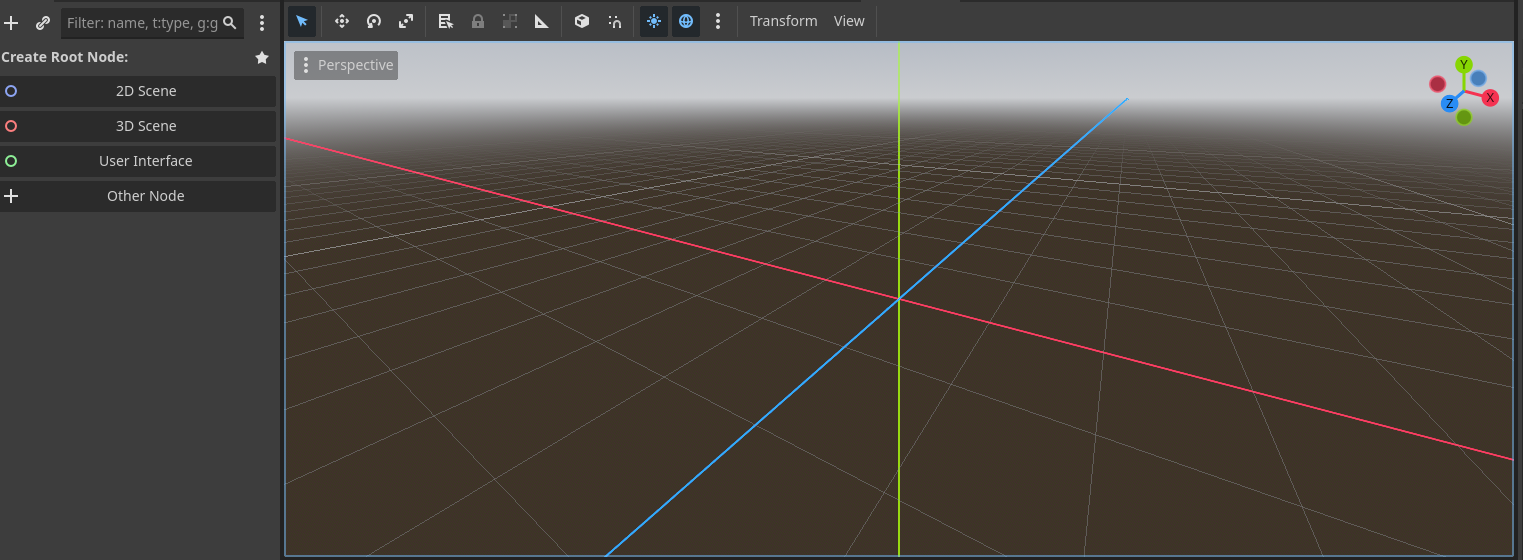
*Hình 2.13: Thẻ Control*

Màn hình chính gồm có bốn nút màn hình chính được đặt chính giữa ở đầu trình chỉnh sửa: 2D, 3D, Script và AssetLib.



*Hình 2.14: Màn hình chính*

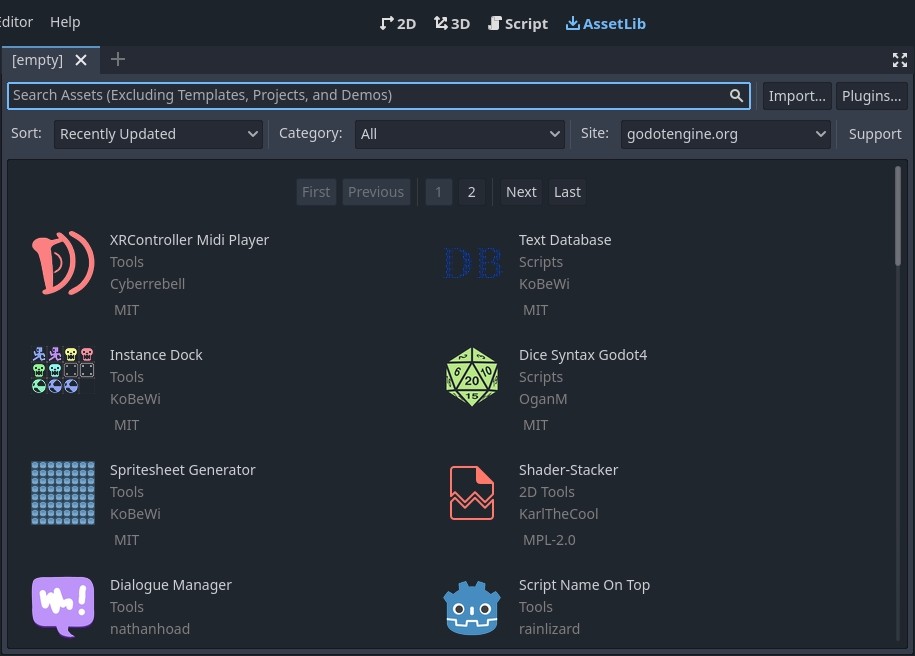
Trong màn hình 3D , bạn có thể làm việc với các mắt lưới, ánh sáng và cấp độ thiết kế cho trò chơi 3D.



*Hình 2.15: Màn hình thiết kế*

Script screen là trình chỉnh sửa mã hoàn chỉnh với trình gỡ lỗi, tính năng tự động hoàn thành phong phú và tham chiếu mã tích hợp.

Cuối cùng, AssetLib là một thư viện gồm các tiện ích bổ sung, tập lệnh và tài sản nguồn mở và miễn phí để sử dụng trong các dự án của bạn.



*Hình 2.16: Thư viện tài nguyên*

### Tổng quan về ngôn ngữ GdScript

GDScript là một ngôn ngữ lập trình kịch bản được phát triển chuyên biệt cho Godot Engine – một trong những công cụ làm game mã nguồn mở phổ biến nhất hiện nay. Khác với nhiều ngôn ngữ lập trình được tích hợp vào game engine một cách gián tiếp, GDScript được tạo ra nhằm phục vụ duy nhất cho Godot, nên mức độ tương thích và tối ưu hóa là rất cao. Việc Godot có một ngôn ngữ “cây nhà lá vườn” không phải là điểm trừ, mà trái lại, đó là một trong những lý do khiến Godot trở nên dễ tiếp cận và thân thiện hơn với người mới bắt đầu.

Điểm khiến GDScript trở nên hấp dẫn, đặc biệt đối với người học lập trình lần đầu, là ở sự đơn giản trong cú pháp và cấu trúc ngôn ngữ. GDScript được lấy cảm hứng từ Python – một ngôn ngữ vốn nổi tiếng vì sự ngắn gọn, rõ ràng và dễ hiểu. Nhờ đó, người mới học không bị choáng ngợp bởi những cú pháp phức tạp hay quy tắc khắt khe. Việc đọc và viết một đoạn mã trở nên nhẹ nhàng hơn, không quá khác biệt so với cách con người suy nghĩ logic trong đời sống hằng ngày. Đây chính là yếu tố quan trọng giúp GDScript trở thành công cụ lý tưởng để dạy và học lập trình trong môi trường trực quan như trò chơi.

GDScript còn được thiết kế để tận dụng tối đa cấu trúc node-based (dựa trên các nút) đặc trưng của Godot. Trong Godot, mọi đối tượng – từ nhân vật, vật thể, giao diện, cho đến các yếu tố âm thanh và hiệu ứng – đều được tổ chức dưới dạng các node có thể lồng ghép với nhau. Nhờ GDScript, người phát triển có thể dễ dàng kiểm soát từng node một cách trực tiếp, như điều khiển hành vi của nhân vật, xử lý va chạm, phản hồi sự

kiện, thay đổi hiệu ứng hiển thị, hoặc chuyển đổi giữa các màn chơi. Tất cả đều có thể thực hiện một cách mạch lạc và nhanh chóng mà không cần phải viết quá nhiều mã hay cấu hình rườm rà.

Một điểm cộng nữa của GDScript là khả năng hỗ trợ lập trình hướng đối tượng, cho phép người dùng tổ chức mã nguồn một cách khoa học, dễ mở rộng và bảo trì. Điều này rất cần thiết khi quy mô của trò chơi bắt đầu lớn hơn, với nhiều nhân vật, vật phẩm, quy tắc và giao diện khác nhau. Ngoài ra, GDScript còn hỗ trợ kiểm tra lỗi ngay trong quá trình viết mã và có khả năng cập nhật tức thì trong khi trò chơi đang chạy, giúp việc thử nghiệm và tinh chỉnh sản phẩm trở nên linh hoạt và tiết kiệm thời gian.

Không chỉ dừng lại ở mặt kỹ thuật, GDScript còn góp phần tạo ra một môi trường học tập mang tính khám phá và trải nghiệm. Người học khi tiếp cận GDScript thường không chỉ học các khái niệm lập trình khô khan mà còn được thấy ngay kết quả trực quan của mình trong trò chơi – từ việc một nhân vật di chuyển, một nút bấm xuất hiện, đến những thay đổi phức tạp hơn như điều kiện thắng thua hay các hiệu ứng đặc biệt. Điều này giúp tạo ra cảm giác hứng thú, giúp người học duy trì động lực và phát triển tư duy lập trình một cách tự nhiên, thay vì cảm thấy chán nản như khi chỉ học qua lý thuyết.

Sự phát triển liên tục của cộng đồng Godot cũng góp phần làm cho GDScript ngày càng hoàn thiện hơn. Các phiên bản mới liên tục được cập nhật để cải thiện hiệu năng, bổ sung tính năng mới, tăng cường sự ổn định và mở rộng khả năng tùy biến. Người dùng có thể dễ dàng tìm thấy tài liệu, ví dụ, hướng dẫn chi tiết hoặc trao đổi trực tiếp với những người dùng khác để học hỏi và giải quyết các vấn đề thực tế trong quá trình phát triển game bằng GDScript.

GDScript không đơn thuần là một ngôn ngữ lập trình dành cho việc viết kịch bản game, mà còn là một phương tiện hiệu quả để đưa người học tiếp cận với tư duy lập trình, kỹ năng giải quyết vấn đề và khả năng sáng tạo. Với sự kết hợp giữa tính trực quan, dễ học và tích hợp sâu với Godot, GDScript đang ngày càng chứng minh vai trò quan trọng trong việc thúc đẩy giáo dục lập trình thông qua trò chơi, đặc biệt là đối với người mới bắt đầu. Việc lựa chọn GDScript trong dự án trò chơi 2D hỗ trợ học lập trình không chỉ là lựa chọn về công nghệ, mà còn là lựa chọn mang tính sư phạm, gắn liền với sự phát triển của tư duy logic và khả năng tự học trong một môi trường tương tác sinh động.

### Tổng quan về thiết kế trò chơi

### Khái niệm game Puzzle và RPG

Sự kết hợp giữa thể loại Giải đố (Puzzle) và Nhập vai (RPG) đã tạo ra một dòng game vô cùng hấp dẫn. Thay vì chỉ đơn thuần là sắp xếp các khối hình hay giải mã con số, việc tích hợp yếu tố nhập vai giúp các hành động giải đố trở nên có ý nghĩa và sống động hơn.

Đối với loại trò chơi này, người chơi sẽ điều khiển nhân vật chính sửa tàu. Góc máy quay trong game có thể đặt ở nhiều vị trí (thứ nhất, thứ ba, hoặc góc nhìn 2D), tùy vào thiết kế của trò chơi.

Điểm đặc trưng của game Puzzl và RPG là sự kết hợp giữa gameplay nhập vai lành mạnh và những đoạn chơi mang tính tư duy, khám phá (như giải đố, tương tác với môi trường, thu thập vật phẩm). Thể loại này thường yêu cầu người chơi vừa có kỹ năng thao tác nhanh, vừa cần khả năng quan sát và tư duy logic.

### Cách chơi

Mục tiêu của trò chơi là người chơi phải hòa mình vào nhân vật “Thợ sửa tàu”, yêu cầu di chuyển nhân vật chính bằng các nút nhấn trên bàn phím để tiếp cận tàu, đồng thời sử dụng chuột máy tính để điều khiển hướng nhìn, nhấn V để quét tàu và tìm chỗ hư hại.

Người chơi tương tác bản request để tiếp nhận sửa tàu, sau đó tương tác với chỗ hỏng để puzzle hiện lên, người chơi phải hoàn thành và sửa hết các chỗ hư hại sau đó tàu sẽ bay đi và request hiện lên và vòng lặp lặp lại, người chơi sẽ phải thực hiện sửa tàu cho đến khi hết thời gian hoặc chọn ngừng trò chơi.

### Các yêu cầu về trò chơi

1. **Yêu cầu về kĩ thuật**

Đầu tiên và quan trọng nhất, nhóm cần nắm vững kỹ thuật sử dụng Godot Engine**.** Điều này bao gồm khả năng làm việc thành thạo với giao diện, hệ thống node, scene, quản lý tài nguyên, cũng như hiểu biết sâu sắc về luồng hoạt động của Godot. Khả năng tận dụng tối đa các công cụ tích hợp sẵn của Godot sẽ giúp tăng tốc độ phát triển và tối ưu hóa quy trình làm việc.

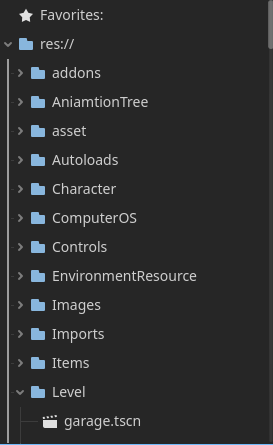
Song song đó, kỹ thuật lập trình ngôn ngữ GDScript là yếu tố cốt lõi. GDScript, với cú pháp tương tự Python, yêu cầu nhóm phải có khả năng viết mã sạch, hiệu quả và có cấu trúc. Việc thành thạo GDScript không chỉ giúp triển khai logic game phức tạp như hệ thống giải đố, quản lý vật phẩm mà còn đảm bảo tương tác mượt mà giữa các thành phần trong game, tận dụng tối đa API của Godot.

Bên cạnh lập trình, kỹ thuật xử lý hình ảnh và thiết kế đồ họa cơ bản cũng rất cần thiết. Việc hiểu biết về nguyên tắc thiết kế đồ họa, khả năng tạo và chỉnh sửa tài nguyên hình ảnh (như sprite, background, hiệu ứng) ở mức cơ bản sẽ giúp em chủ động hơn trong việc xây dựng thế giới game và các yếu tố trực quan.

Một kỹ thuật đặc biệt quan trọng khác là kỹ thuật thiết kế về Procedural Generation. Đây là một yếu tố then chốt để tạo ra các tàu đa dạng và có khả năng chơi lại cao mà không cần thiết kế thủ công từng cấp độ. Cần có khả năng xây dựng các thuật toán để tự động sinh ra cấu trúc puzzle và điểm hư hại một cách ngẫu nhiên nhưng vẫn đảm bảo tính cân bằng và hấp dẫn của gameplay.

### Yêu cầu về tài nguyên

Dự án "Sửa tàu vũ trụ bằng puzzle" được tổ chức theo cấu trúc thư mục rõ ràng và hợp lý, giúp việc phát triển, bảo trì và mở rộng trò chơi được dễ dàng hơn. Cấu trúc này phản ánh cách tiếp cận mô-đun – chia nhỏ thành từng thành phần chuyên biệt, mỗi phần đảm nhiệm một vai trò trong hệ thống tổng thể.



*Hình 2.17: Hệ thống phân cấp tệp*

### Yêu cầu về âm thanh đồ họa

Thiết kế đồ họa cần đảm bảo nổi bật; dễ quan sát các đối tượng trong màn chơi; đảm bảo độ phân giải không bị mờ, lỗi; kết hợp phát triển vào đó yếu tố mĩ thuật giúp thu hút người chơi với giao diện đẹp, hình ảnh đẹp.

Âm thanh nền, âm thanh thao tác, âm thanh khi nhân vật di chuyển, âm thanh khi nhân vật tấn công, âm thanh khi tàu đáp xuống,... đảm bảo rằng: có đủ âm thanh đối với mọi tương tác trong game, âm thanh có thể điều chỉnh độ lớn, nhỏ, bật hoặc tắt âm thanh [3].

Các hiệu ứng âm thanh được nhóm tìm từ các nguồn trên Internet và ghép với tương tác trong game.

### Yêu cầu về hiệu suất

Nguyên tắc 1: Thiết kế giao diện game chặt chẽ và dễ sử dụng

Cách thiết kế giao diện cho game là một trong những vấn đề rất quan trọng, điều này thu hút người chơi ngay từ những lần tiếp xúc đầu tiên. Ngoài ra, còn cần phải chú trọng đến bố cục của game sao cho hợp lý để người dùng dễ dàng tiếp cận các thông tin cần thiết khi sử dụng. Bởi khi game được thiết kế không tốt thì người chơi sẽ khó tiếp cận được những nội dung theo đúng ý muốn của người chơi. Từ đó, sẽ nhanh chóng gây ra sự nhàm chán.

Về cách phân chia và tổ chức, trước khi Game được thiết kế, em đã định hình một số khung giao diện thường gặp và thiết kế chúng trở thành giao diện.

Nguyên tắc 2: Game phải được sử dụng một cách mượt mà

Các cử chỉ, hành động của nhân vật chính cũng như kẻ thù phải được thực thi một cách trơn tru, do đó nhóm thực hiện đã quyết định tận dụng để xây dựng cách điều khiển bằng nút bấm trên bàn phím kết hợp với thao tác chuột nhằm giúp cho người chơi có thể thao tác dễ dàng nhất.

Nguyên tắc 3: Cách cài đặt game phải dễ dàng

Nhờ công cụ Godot, như những gì đã liệt kê về ưu điểm của nó, việc cài đặt và chơi trên đa nền tảng là hết sức dễ dàng, người chơi chỉ cần chọn nền mà mình mong muốn game được chạy trên đó và build.

Hệ thống sẽ tự động build cho người chơi một file .exe, chỉ việc thực hiện click double vào và game sẽ tự động được build và hoạt động.

Kết luận chương: Chương 2 không chỉ đóng vai trò cung cấp kiến thức lý thuyết mà còn là bước đệm quan trọng giúp người thực hiện định hình tư duy và hướng đi trong quá trình phát triển game. Việc lựa chọn Godot Engine thay vì các công cụ nổi tiếng khác không đơn thuần là vì yếu tố “miễn phí” hay “nhẹ”, mà bởi vì sự phù hợp với điều kiện thực tế của sinh viên, cũng như khả năng học hỏi và ứng dụng linh hoạt mà nó mang lại. Thông qua việc tìm hiểu các thành phần trong Godot, từ hệ thống node – scene cho đến giao diện và ngôn ngữ lập trình GDScript, người thực hiện không chỉ nắm được cách một trò chơi hoạt động mà còn hiểu sâu hơn về kiến trúc của một sản phẩm phần mềm tương tác.

Không dừng lại ở mặt công nghệ, chương này cũng nhấn mạnh đến những yêu cầu mang tính thực tiễn như hiệu suất, tài nguyên, âm thanh – đồ họa, cho thấy rằng làm game không chỉ là viết mã, mà còn là một quá trình kết hợp giữa tư duy kỹ thuật và cảm quan thẩm mỹ. Nhờ sự chuẩn bị kỹ lưỡng về mặt lý thuyết, các phần sau của đồ án sẽ có được nền tảng vững chắc để đi vào triển khai và hiện thực hóa ý tưởng trò chơi một cách hiệu quả hơn.

## CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ

Tóm tắt chương: Chương 3 trình bày quá trình xây dựng trò chơi từ phân tích đề tài đến thiết kế chi tiết các thành phần trong game. Nội dung bao gồm thiết kế môi trường, đồ họa, âm thanh, gameplay và các màn chơi nhằm tạo trải nghiệm hấp dẫn cho người chơi. Ngoài ra, chương cũng đề cập đến việc xây dựng nhân vật chính, hệ thống tàu, puzzle và hệ thống lưu – tải game. Cuối cùng là phần thiết kế link kiện và trang bị, giúp tăng chiều sâu và chiến thuật trong quá trình chơi.

### Mô tả bài toán

Trong việc học lập trình ngày càng trở nên phổ biến và cần thiết, đặc biệt với những người mới bắt đầu, một trong những khó khăn lớn nhất là làm sao để tiếp cận được kiến thức một cách nhẹ nhàng, dễ tiếp thu và không quá áp lực. Lập trình vốn dĩ là một lĩnh vực mang tính logic cao, khô khan và đòi hỏi khả năng tư duy trừu tượng. Tuy nhiên, không phải ai cũng có thể ngay lập tức thích nghi với cách học truyền thống – vốn thường tập trung vào lý thuyết, cú pháp, và các ví dụ trên sách vở hoặc màn hình đen trắng của các môi trường lập trình. Chính vì thế, bài toán được đặt ra là: “làm thế nào để biến quá trình học lập trình cơ bản trở nên sinh động, hấp dẫn hơn, đặc biệt là đối với những người chưa từng tiếp xúc với nó? ”.

Từ thực tế đó, đề tài hướng đến việc xây dựng một trò chơi đơn giản, sử dụng công cụ Godot Engine, nhằm hỗ trợ người học làm quen với những khái niệm lập trình nền tảng như biến, vòng lặp, điều kiện, và xử lý logic. Mục tiêu không phải là thay thế hoàn toàn phương pháp học truyền thống, mà là bổ sung thêm một cách tiếp cận mới – học thông qua chơi, hay còn gọi là học bằng trải nghiệm. Trong môi trường game, người chơi không chỉ giải trí mà còn vô tình được tiếp cận với các khái niệm lập trình thông qua cách giải quyết các puzzle, thử thách được thiết kế lồng ghép trong gameplay. Mỗi tàu sẽ đặt ra một tình huống cụ thể, yêu cầu người chơi vận dụng tư duy logic để có thể vượt qua.

Bài toán đặt ra không chỉ nằm ở việc trò chơi phải hấp dẫn mà còn phải đảm bảo yếu tố sư phạm – nghĩa là người chơi sau khi hoàn thành một màn chơi phải học được điều gì đó, dù là khái niệm đơn giản nhất như điều kiện “nếu – thì”, hay cách lặp lại một hành động nhất định. Ngoài ra, trò chơi cũng cần phải dễ điều khiển, giao diện thân thiện, có hướng dẫn rõ ràng, và nội dung phải phù hợp với người mới bắt đầu – tránh gây cảm giác quá tải hoặc nhàm chán.

Bên cạnh đó, bài toán cũng bao gồm việc lựa chọn và khai thác hiệu quả công cụ phát triển game – trong trường hợp này là Godot Engine 4. Đây là một nền tảng mã nguồn mở, có khả năng hỗ trợ lập trình bằng ngôn ngữ GDScript – vốn phù hợp với

người học mới. Việc sử dụng Godot không chỉ giúp rút ngắn thời gian phát triển mà còn tạo điều kiện cho người phát triển – đồng thời cũng là người học – được trải nghiệm toàn bộ quy trình tạo ra một trò chơi, từ ý tưởng đến sản phẩm hoàn chỉnh.

### 

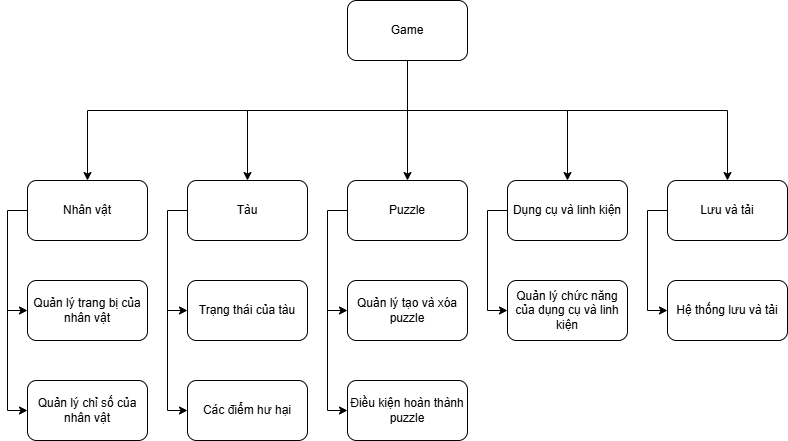
### 3.2 Phân tích hệ thống

### Sơ đồ Phân rã chức năng

Sơ đồ phân rã chức năng (BFD - Business Function Diagram**)** là một công cụ trực quan quan trọng trong phân tích và thiết kế hệ thống, cho phép mô tả hệ thống dưới dạng cấu trúc các chức năng chính cùng với các chức năng con tương ứng. Thông qua sơ đồ này, toàn bộ hệ thống được phân tích theo hướng từ tổng thể đến chi tiết, giúp chia nhỏ hệ thống thành các quy trình con một cách có tổ chức, logic và dễ hiểu. Đây là bước nền tảng để xác định cấu trúc chức năng, hỗ trợ việc lập kế hoạch và tổ chức triển khai các tác vụ phát triển hệ thống.

BFD mang lại cái nhìn tổng quan về hệ thống thông qua việc trình bày rõ ràng mối quan hệ giữa các chức năng, cách các chức năng tương tác với nhau và cách chúng trao đổi dữ liệu. Điều này cho phép các nhà phân tích hệ thống, kỹ sư phần mềm và các bên liên quan hiểu rõ hơn về cách hoạt động nội tại của hệ thống, cũng như mối liên kết giữa các bộ phận chức năng khác nhau. Việc xác định ranh giới hệ thống và phạm vi của từng chức năng trong sơ đồ giúp giảm thiểu sự nhầm lẫn và đảm bảo rằng các yêu cầu người dùng được phản ánh chính xác trong thiết kế.

Một điểm nổi bật của BFD là khả năng biểu diễn các luồng dữ liệu giữa các chức năng trong hệ thống cũng như giữa hệ thống và các tác nhân bên ngoài (như người dùng, hệ thống khác, cơ quan tổ chức...). Nhờ đó, BFD không chỉ là công cụ dành cho kỹ sư phần mềm mà còn là phương tiện truyền đạt hiệu quả thông tin cho các bên liên quan không chuyên về kỹ thuật, như khách hàng, nhà quản lý dự án hay các phòng ban nghiệp vụ. Điều này góp phần tạo nên sự thống nhất trong nhận thức và phối hợp giữa các bên trong quá trình phát triển hệ thống.



*Hình 3.2: Sơ đồ phân rã chức năng*

### Sơ đồ Usecase của Player

Hệ thống trò chơi nhập vai 2D (RPG 2D) được xây dựng với mục tiêu mang đến một trải nghiệm chơi game thú vị, linh hoạt và trực quan cho người chơi. Trong quá trình phát triển hệ thống, các chức năng chính được xác định và mô hình hóa thông qua sơ đồ Use Case dưới đây.

Người chơi khi khởi động trò chơi sẽ được đưa đến giao diện chính với các lựa chọn: bắt đầu trò chơi mới, tiếp tục trò chơi đã lưu hoặc thoát khỏi game. Khi chọn "Chơi game", hệ thống sẽ khởi tạo các thành phần cần thiết và đưa người chơi vào bản đồ đầu tiên - thường là khu vực an toàn. Nếu người chơi đã từng lưu trạng thái chơi trước đó, họ có thể chọn "Tiếp tục" để tải lại tiến trình cũ. Đây là bước khởi đầu cho toàn bộ hành trình khám phá và chiến đấu trong game.

Trong game, người chơi sẽ được trải nghiệm nhiều bản đồ khác nhau, bao gồm khu vực an toàn (không có quái vật, có thể tương tác với NPC) và khu vực chiến đấu (có quái vật và thử thách). Mỗi màn chơi được thiết kế với địa hình, quái vật và mục tiêu riêng biệt, giúp tăng tính đa dạng và hấp dẫn. Trong quá trình chơi, người chơi có thể khám phá bản đồ, chiến đấu với quái vật, nhận nhiệm vụ hoặc thu thập vật phẩm.

Khám phá bản dồ là chức năng này cho phép người chơi tự do di chuyển trong thế giới game, tương tác với các vật thể và NPC. Trong quá trình khám phá, người chơi có thể phát hiện các bí mật, chạm trán với kẻ địch hoặc tìm được vật phẩm hỗ trợ. Việc khám phá bản đồ đóng vai trò quan trọng trong việc mở khóa nhiệm vụ mới và tiến xa hơn trong hành trình.

Tấn công quái vật đây là một trong những chức năng trọng tâm của trò chơi. Khi phát hiện kẻ địch, người chơi có thể sử dụng phím chuột để điều khiển hướng tấn công, sử dụng vũ khí như kiếm, boomerang, hoặc các kỹ năng đặc biệt để tiêu diệt quái. Mỗi khi tiêu diệt thành công một quái vật, người chơi sẽ nhận được điểm kinh nghiệm, vật phẩm ngẫu nhiên hoặc vàng. Ngược lại, nếu bị tấn công quá nhiều, người chơi sẽ mất máu và có thể thất bại, buộc phải chơi lại từ đầu màn.

Người chơi có thể tương tác với các NPC thông qua phím chức năng để nhận nhiệm vụ. Mỗi nhiệm vụ yêu cầu người chơi thực hiện một số hành động cụ thể như tiêu diệt một số lượng quái nhất định, thu thập vật phẩm hoặc mở một khu vực bản đồ. Khi hoàn thành nhiệm vụ, người chơi sẽ được nhận phần thưởng như vàng, trang bị hoặc nâng cấp chỉ số.

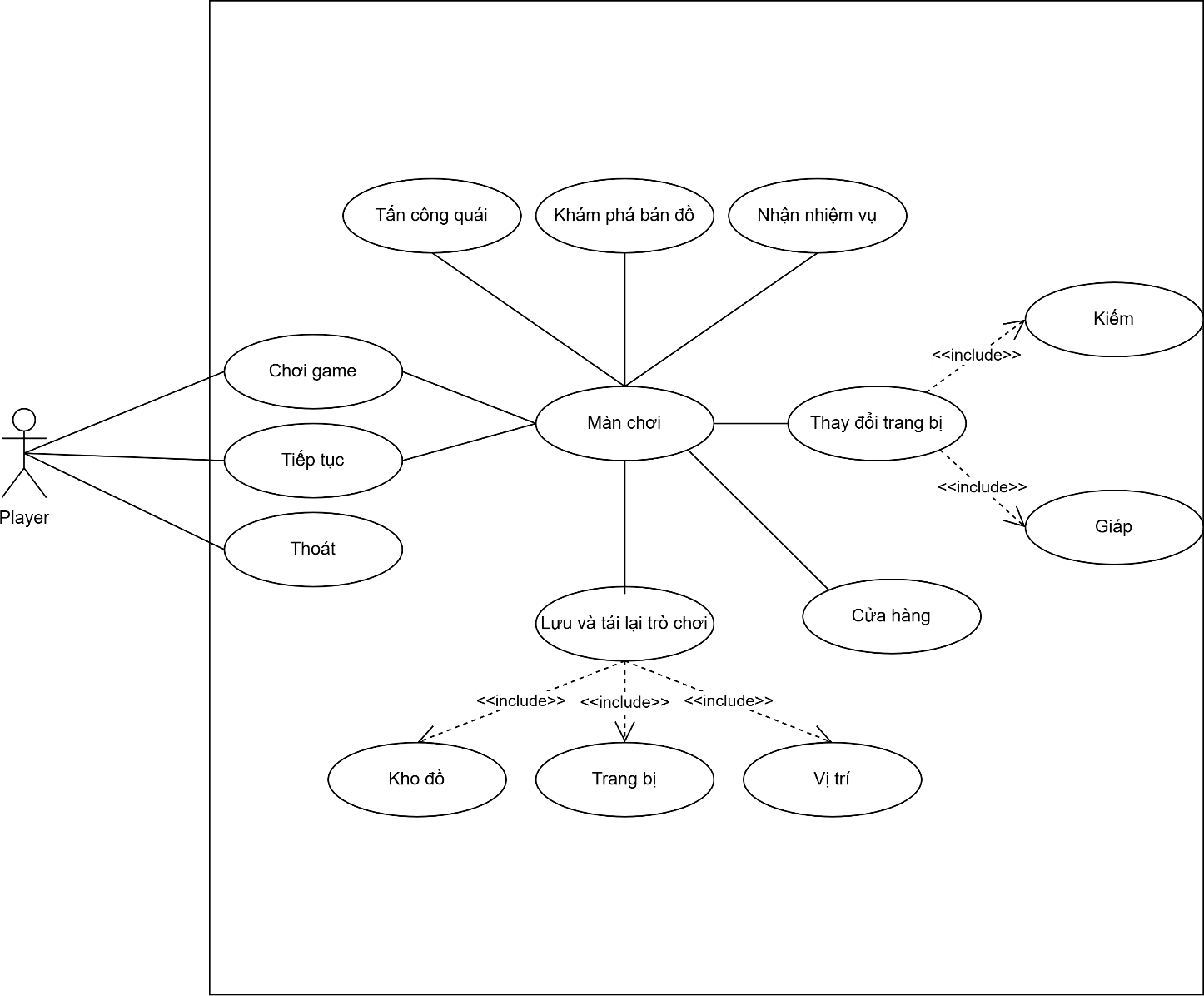
Khi thu thập được các vật phẩm mới, người chơi có thể vào giao diện "Trang bị" để thay đổi hoặc nâng cấp các loại vũ khí, giáp trụ đang sử dụng. Việc trang bị phù hợp sẽ giúp tăng sức mạnh, khả năng phòng thủ hoặc tốc độ di chuyển của nhân vật, góp

phần nâng cao cơ hội chiến thắng trong các trận đấu.

Hệ thống cho phép người chơi lưu lại tiến trình hiện tại thông qua chức năng "Save", và có thể tải lại trò chơi từ nơi đã dừng trước đó bằng chức năng "Load". Dữ liệu lưu bao gồm vị trí nhân vật, chỉ số, trang bị và nhiệm vụ đã hoàn thành. Điều này giúp người chơi dễ dàng quay lại quá trình chơi mà không bị mất dữ liệu, đặc biệt hữu ích với các trò chơi có độ dài và độ khó cao như RPG.

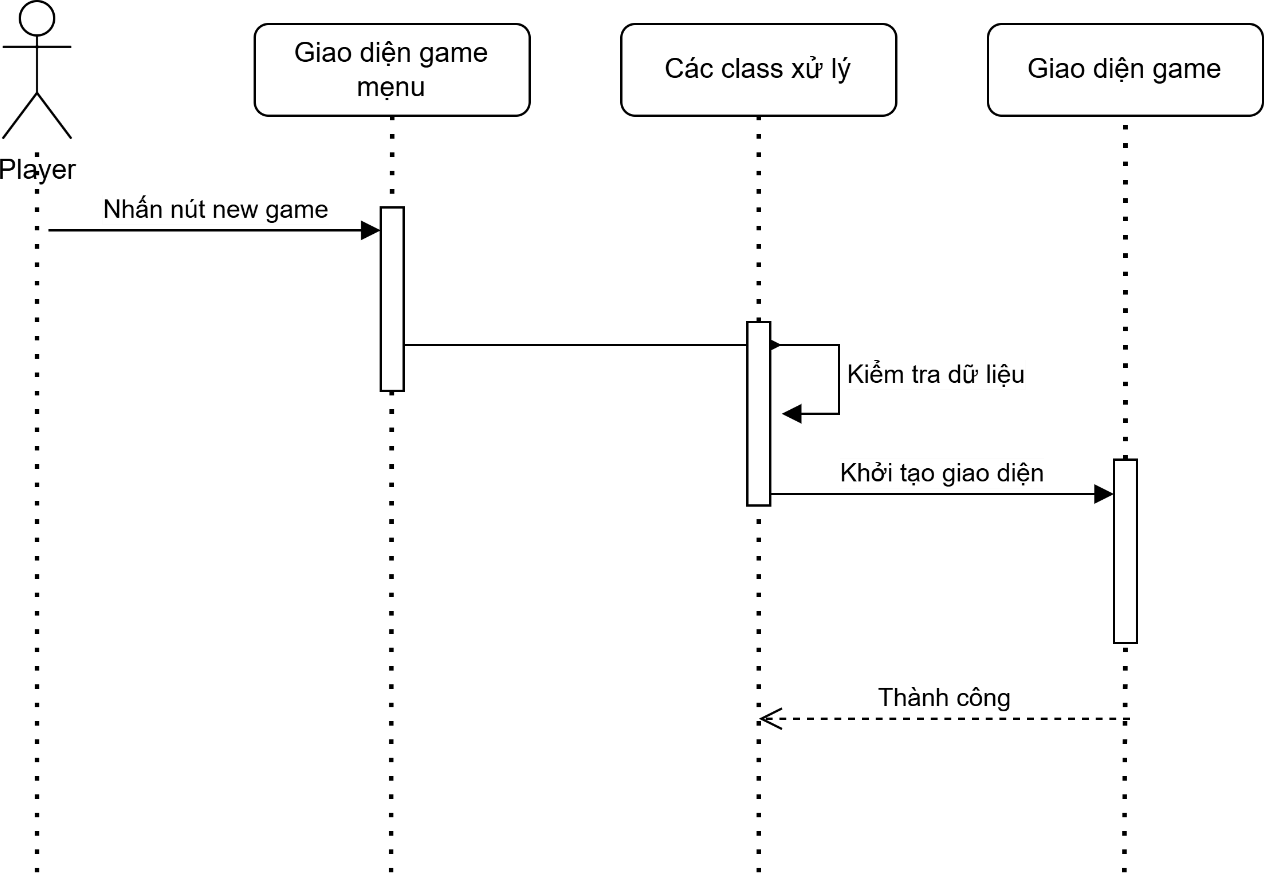
Trong game, người chơi có thể sử dụng tài nguyên nhặt được để mua vật phẩm tại cửa hàng. Các vật phẩm này bao gồm bình máu, trang bị mới, hoặc vật phẩm hỗ trợ. Sau khi mua, vật phẩm sẽ được lưu trữ trong kho đồ của người chơi, nơi họ có thể quản lý, sắp xếp và sử dụng khi cần thiết.

Hệ thống còn hỗ trợ chức năng xác định vị trí hiện tại của người chơi trên bản đồ và cho phép tương tác với các vật thể như rương báu, hòm đồ hoặc các vật phẩm có thể bị phá hủy để lấy phần thưởng. Ngoài ra, một số vật phẩm khi mở còn yêu cầu người chơi trả lời câu hỏi – đây là điểm sáng tạo kết hợp yếu tố giáo dục trong game.



*Hình 3.3: Sơ đồ Usecase của Player*

### Sơ đồ Tuần tự

****

*Hình 3.4: Sơ đồ tuần tự vào game*

Mô tả các bước mà người chơi tương tác với hệ thống từ lúc mở game, chọn “Start”, hệ thống khởi tạo nhân vật, bản đồ, cho đến khi người chơi bắt đầu di chuyển, chiến đấu.

### 

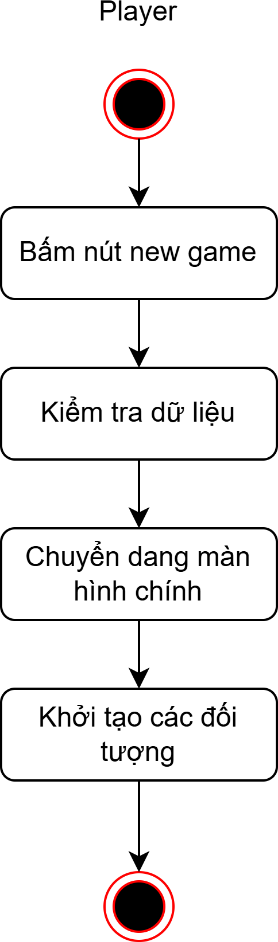
### Sequence_ship_fix_diagram

*Hình 3.5: Sơ đồ tuần tự sửa tàu*

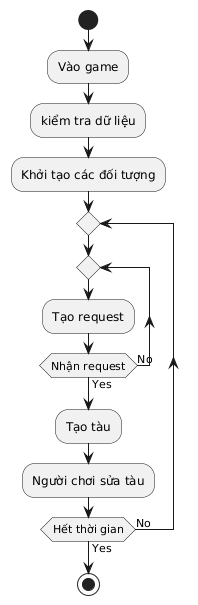
### Sequence_ware_and_restock_diagram

*Hình 3.6: Sơ đồ tuần tự restock*

### 3.2.5 Sơ đồ Hoạt dộng

****

*Hình 3.7: Sơ đồ hoạt dộng vào game*



*Hình 3.8: Sơ đồ hoạt động đầy đủ game*

### 3.3 Hệ thống tạo ra puzzle

Trong trò chơi 3D hỗ trợ học lập trình được phát triển bằng Godot, hệ thống tạo ra puzzle từ việc sử dụng vật phẩm đóng vai trò quan trọng trong việc kết hợp giữa yếu tố giải trí và nội dung giáo dục. Khi người chơi tương tác với một vật phẩm nhất định trong môi trường trò chơi , hệ thống sẽ kích hoạt một cơ chế tự động tạo ra câu đố liên quan đến kiến thức lập trình. Cơ chế này được thiết kế nhằm giúp người chơi tiếp cận và ôn tập các khái niệm như biến, kiểu dữ liệu, câu lệnh điều kiện, vòng lặp, hàm, v.v. thông qua tình huống cụ thể trong game.

Puzzle được trình bày dưới dạng giao diện UI trong game hay vật 3D. Sau khi người chơi giải puzzle, chỗ hư hại đang tương tác sẽ được sửa.

Thiết kế này không chỉ tạo ra sự liên kết tự nhiên giữa gameplay và nội dung học lập trình, mà còn giúp củng cố kiến thức thông qua sự lặp lại trong ngữ cảnh cụ thể.

### 3.4 Kiểm thử

1. **Mục tiêu kiểm thử**

Mục tiêu của việc kiểm thử là đánh giá mức độ ổn định, độ chính xác, hiệu năng và tính tương tác của game trong các tình huống sử dụng thực tế. Kiểm thử cũng giúp phát hiện lỗi để sửa chữa, từ đó nâng cao chất lượng sản phẩm trước khi đưa vào sử dụng hoặc trình bày bảo vệ đồ án.

### Kịch bản kiểm thử

Kiểm thử chức năng (Functional Testing): Đảm bảo các chức năng như di chuyển, tương tác, lưu/ tải game, hoạt động đúng như thiết kế.

Kiểm thử UI/UX: Đánh giá khả năng hiển thị và mức độ thân thiện với người dùng.

Kiểm thử hiệu suất (Performance Testing): Đo FPS trung bình, độ mượt khi chơi, khả năng tải nhanh cảnh mới.

Kiểm thử lỗi (Bug Testing): Ghi nhận các lỗi trong quá trình chơi thử (ví dụ: va chạm không chính xác, quái vật không phản ứng).

### Kết quả kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tính năng kiểm thử | Kết quả mong đợi | Trạng thái |
| 1 | Di chuyển nhân vật | Nhân vật di chuyển đúng hướng | Thành công |
| 2 | Quét tàu | Hiển thị chỗ hư hại của tàu | Chưa thành công |
| 3 | Tạo puzzle | Tạo puzzle cho tàu | Thành công |
| 4 | Cập nhật tàu | Cập nhật lại status của tàu sau khi sửa | Chưa hoàn thành |
| 5 | Mở kho | Hiển thị kho hàng | Thành công |
| 6 | Restock | Bổ sung kho hàng đầu ngày | Chưa hoàn thành |
| 7 | Lưu game | File đươck ghi lại | Chưa hoàn thành |
| 8 | Tải lại game | Game khôi phục trạng thái trước đó | Chưa hoàn thành |

*Bảng 3.1: Kết quả kiểm thử*

Kết luận chương: Chương 3 đã trình bày toàn bộ quá trình xây dựng trò chơi từ khâu thiết kế giao diện, nhân vật, môi trường, đến triển khai các chức năng gameplay và hệ thống tương tác. Mỗi thành phần trong game đều được phát triển dựa trên định hướng rõ ràng, đảm bảo tính logic, thẩm mỹ và trải nghiệm người dùng. Việc tích hợp các yếu tố như hệ thống vật phẩm, lưu – tải tiến trình đã góp phần hoàn thiện trò chơi về cả mặt kỹ thuật lẫn nội dung. Đây là bước then chốt giúp chuyển đổi các ý tưởng từ lý thuyết sang sản phẩm thực tế có thể trải nghiệm được.

## CHƯƠNG 4: XÂY DỰNG TRÒ CHƠI

### Thiết kế giao diện

****

*Hình 4.1: Giao diện màn hình chính*

Giao diện chính khi vào game cho phép người chơi lựa chọn dựa vào hai nút (button) được thiết kế sẵn, là “Start” và “Exit” cho phép người chơi ấn “Start” để bắt đầu chơi và “Continue” để load lại cái file đã lưu.



*Hình 4.3: Màn hình khi bấm tạm dừng*

Người chơi ấn phím Esc trên bàn phím để tạm dừng trò chơi. Và nhấn Esc để tiếp tục trò chơi

*Hình 4.7: Giao diện khi vào kho*

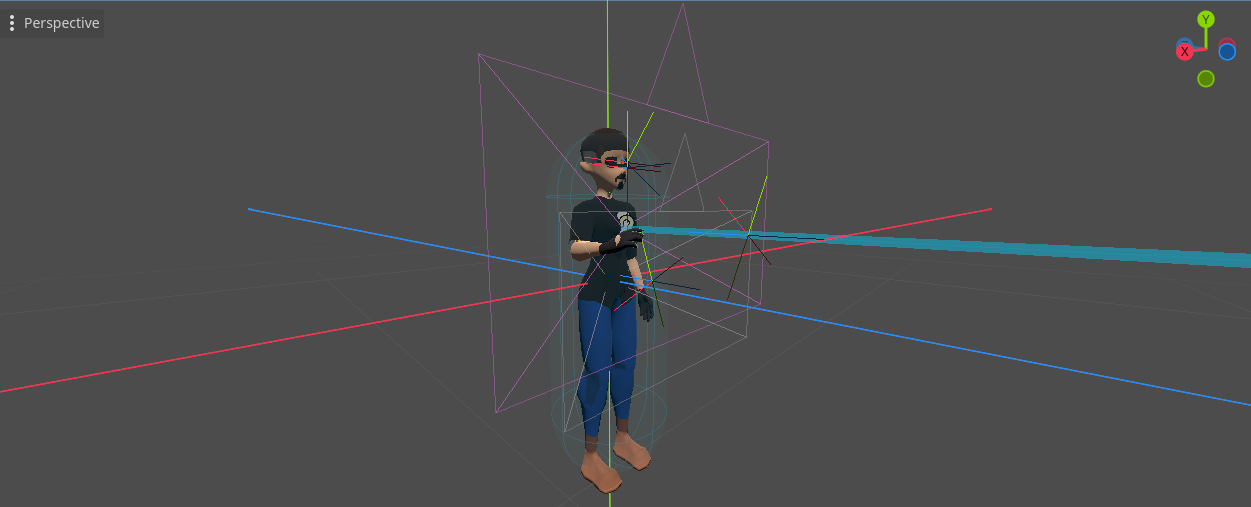
Khi ta nhấn phím Tab ta sẽ thấy được kho hàng hiện tại.



*Hình 4.8: Giao diện khi mở kho*

Sau khi kiểm tra xong kho nhấn Tab một lần nữa để đóng giao diện kho

### Thiết kế player



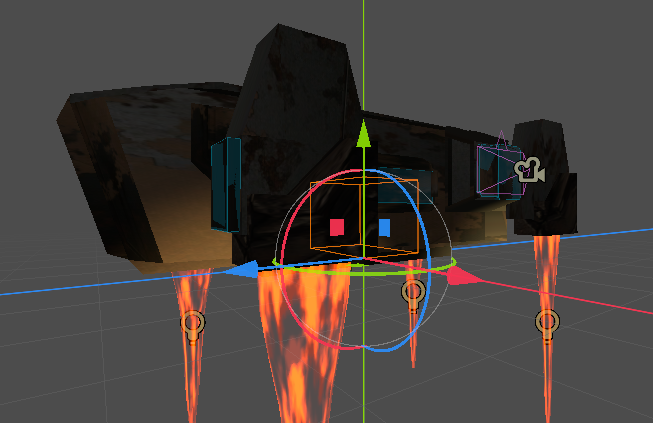
*Hình 4.12: Hình ảnh của nhân vật*

Thiết kế nhân vật Player (người chơi) đóng vai trò trung tâm trong trò chơi, là đối tượng mà người chơi sẽ trực tiếp điều khiển trong suốt quá trình trải nghiệm. Nhân vật được xây dựng dưới dạng một Scene riêng biệt trong Godot Engine, sử dụng node

chính là CharacterBody3D, kết hợp với node CollisionShape3D, AnimationPlayer, AnimationTree, RayCast3D để quản lý hiển thị, va chạm, hoạt ảnh và phạm vi tương tác [4].

Player được thiết kế để có khả năng di chuyển linh hoạt theo bốn hoặc tám hướng, tương tác với các đối tượng trong môi trường, và thực hiện các hành động sinh tồn như ăn uống hay nghỉ ngơi. Việc xây dựng logic điều khiển được thực hiện thông qua ngôn ngữ GDScript vả AnimationTree, đảm bảo tính phản hồi nhanh và linh hoạt trong xử lý chuyển động, hoạt ảnh và các hành vi tương tác.

### Thiết kế tàu



*Hình 4.14: Hình ảnh tàu*

Để tạo ra hệ thống di chuyển cho tàu, trong game này sử dụng phương thức tween, một công cụ hoạt ảnh mạnh mẽ sử dụng các trị số 3D của một vật thể 3D (vị trí, kích thước, độ xoay) và thay đổ các thông số đó theo thời gian tạo nên cảm giác mượt mà của hoạt ảnh.

### Thiết kế hệ thống lưu, tải lại game



*Hình 4.16: Màn hình hệ thống*

Khi người chơi kích hoạt menu hệ thống (thường thông qua một phím tắt hoặc sự kiện trong game), màn hình sẽ tạm thời được dừng lại và hiển thị giao diện gồm hai lựa chọn chính: “Save and quit” và “Settings”. Trong đó nút “Save and quit” đóng vai trò lưu lại tiến trình game hiện tại và thoát game, quay lại trang menu chính.

Khi xác nhận thoát game, hệ thống sẽ lưu lại trạng thái hiện tại của trò chơi vào một tệp có tên là save.sav. Việc lưu được thực hiện tại đường dẫn ảo user://, được định nghĩa trong biến SAVE\_PATH. Đây là đường dẫn đặc biệt trong Godot, đảm bảo tính tương thích đa nền tảng và cho phép ghi dữ liệu vào thư mục người dùng (sandbox) của từng hệ điều hành mà không cần quyền truy cập hệ thống.



*Hình 4.16: Màn hình hệ thống*

Sau đó ở main menu, nếu hệ thống tìm được file save.sav của người thì nút “Start” sẽ chuyển thành nút “Continue”, trò chơi sẽ đọc dữ liệu từ file save.sav, tái tạo lại toàn bộ trạng thái mà người chơi đã lưu trước đó, bao gồm, vật phẩm, loại tàu, chỗ hư hại và puzzl đã tạo, v.v. Điều này giúp người chơi có thể tiếp tục hành trình của mình mà không cần bắt đầu lại từ đầu.

Cuối cùng, nút “Quit” đóng vai trò đơn giản nhưng quan trọng – cho phép người chơi thoát khỏi game một cách an toàn. Tất cả các thao tác được xử lý thông qua sự kiện tương tác với UI, kết hợp với các hàm đọc/ghi dữ liệu và xử lý dừng game logic, đảm bảo tính ổn định và trải nghiệm người dùng mượt

## KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận

Qua quá trình nghiên cứu và thực hiện, đề tài "Sửa tàu vũ trụ bằng puzzle" đã đạt được các mục tiêu cơ bản đã đề ra. Cụ thể, trò chơi được phát triển bằng Godot Engine 4 đã tích hợp được các yếu tố học tập và giải trí, mang đến một trải nghiệm mới mẻ và thú vị cho người chơi.

Thông qua các tình huống tương tác trong game như điều khiển nhân vật, giải đố, người chơi không chỉ giải trí mà còn tiếp cận được các khái niệm lập trình cơ bản một cách trực quan. Bên cạnh đó, việc sử dụng GDScript – ngôn ngữ kịch bản tương tự Python – cũng giúp người thực hiện hiểu rõ hơn về lập trình hướng đối tượng, xử lý logic và tối ưu hóa hiệu suất trong game.

Đề tài đồng thời giúp người chơi thực hiện nâng cao kỹ năng trong việc tổ chức dự án, làm việc với công cụ Godot, sử dụng GitHub để quản lý mã nguồn và tiếp cận với quy trình phát triển phần mềm từ ý tưởng đến sản phẩm hoàn chỉnh. Có thể nói, đây là một bước khởi đầu tốt để ứng dụng game vào việc hỗ trợ giáo dục trong bối cảnh học tập hiện đại.

1. Hạn chế

Mặc dù đề tài đã hoàn thành mục tiêu xây dựng một trò chơi kết hợp yếu tố học lập trình, tuy nhiên trong quá trình thực hiện vẫn còn tồn tại một số hạn chế nhất định.

Bên cạnh đó, phần gameplay chưa phong phú, thiếu các cơ chế nâng cấp, hệ thống cấp độ hay phần thưởng rõ ràng, dễ dẫn đến cảm giác lặp lại sau một thời gian chơi. Giao diện người dùng trong game còn đơn sơ, chưa có nhiều yếu tố tương tác hoặc đồ họa sinh động để tạo cảm giác hấp dẫn và chuyên nghiệp. Ngoài ra, trò chơi cũng chưa hỗ trợ tính năng lưu tiến trình, chưa có hệ thống bảng điểm hay chế độ chơi đa người, đồng thời vẫn còn hạn chế khi triển khai trên các nền tảng khác ngoài máy tính cá nhân.

1. Kiến nghị

Để nâng cao chất lượng trò chơi và phát triển sản phẩm ở các giai đoạn tiếp theo, một số hướng cải tiến và kiến nghị được đưa ra như sau: Trước hết, cần mở rộng nội dung học lập trình trong game để bao quát thêm nhiều chủ đề quan trọng như cấu trúc

dữ liệu, hàm, mảng, thuật toán sắp xếp hoặc đệ quy, giúp người chơi có thêm cơ hội rèn luyện và ứng dụng kiến thức đa dạng hơn. Về gameplay, nên bổ sung các yếu tố như cấp độ, hệ thống nhiệm vụ và phần thưởng để tăng độ hấp dẫn và thử thách. Giao diện người dùng cần được thiết kế lại theo hướng hiện đại, thân thiện hơn, đồng thời tối ưu hóa để có thể chạy tốt trên nhiều thiết bị khác nhau như điện thoại, máy tính bảng hoặc nền tảng web.

Ngoài ra, việc tích hợp chức năng lưu trữ tiến trình người chơi, bảng xếp hạng và thu thập phản hồi người dùng sẽ giúp cải thiện trải nghiệm và chất lượng trò chơi. Cuối cùng, nhóm thực hiện cũng nên xem xét khả năng đưa trò chơi vào ứng dụng thực tế trong các khóa học lập trình cơ bản hoặc hoạt động hướng nghiệp công nghệ thông tin để phát huy hiệu quả giáo dục của sản phẩm.

# PHỤ LỤC

### Liên kết mã nguồn(Github)

Mã nguồn trò chơi https://github.com/MinhKhoa21/RepairProtocol

1. **Hướng dẫn cài đặt và chạy project Bước 1**: Cài đặt Godot Engine

Truy cập trang chủ của Godot Engine: https://godotengine.org/

Tải phiên bản Godot phù hợp với hệ điều hành của bạn (Windows, macOS, Linux).

Giải nén và chạy file thực thi (không cần cài đặt).

**Bước 2**: Tải mã nguồn từ GitHub

Truy cập kho mã nguồn tại: https://github.com/MinhKhoa21/RepairProtocol

Nhấn vào nút Code > chọn Download ZIP để tải mã nguồn về máy, hoặc dùng lệnh Git: git clone https://github.com/MinhKhoa21/RepairProtocol.git

**Bước 3**: Mở project trong Godot Mở Godot Engine.

Nhấn vào Import.

Chọn file project.godot trong thư mục mã nguồn vừa tải về. Nhấn Import & Edit để mở project.

**Bước 4**: Chạy project

Trong giao diện Godot, nhấn nút Play (biểu tượng hình tam giác) hoặc nhấn F5 ở góc trên màn hình để chạy chương trình.

Nếu được yêu cầu chọn Main Scene, hãy chọn file scene chính (thường là main\_menu.tscn hoặc tương tự trong thư mục dự án).

Game sẽ được build và chạy ngay trong Godot.

## TÀI LIỆU THAM KHẢO

1. D. Giray, "A Beginner's Complete Guide to Unity," 2024. [Online]. Available: [https://vagon.io/blog/complete-guide-to-unity.](https://vagon.io/blog/complete-guide-to-unity) [Accessed 17 06 2025].
2. G. Docs, "Learn to code with GDScript," 2025. [Online]. Available: <https://docs.godotengine.org/en/3.5/getting_started/introduction/learn_to_co> de\_with\_gdscript.html. [Accessed 12 05 2025].
3. FreeSound, "FreeSound – Collaborative database of Creative Commons Licensed Sounds," 2025. [Online]. Available: [https://freesound.org.](https://freesound.org/) [Accessed 16 05 2025].
4. G.Docs, "Using CharacterBody2D/3D," 2014. [Online]. Available: <https://docs.godotengine.org/en/stable/tutorials/physics/using_character_bod> y\_2d.html. [Accessed 3 06 2025].