**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**NIÊN LUẬN CƠ SỞ NGÀNH KỸ THUẬT PHẦN MỀM**

**Đề tài**

**XÂY DỰNG TRÒ CHƠI THỦ THÀNH**

**SỬ DỤNG THUẬT TOÁN BFS**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn  **ThS. Trương Thị Thanh Tuyền**  **MSCB: 1068** | Sinh viên thực hiện  **Cao Nguyễn Hoài Nhân**  **MSSV: B2110089** |
|  |  |

*Cần Thơ 5/2024*

**Lời cảm ơn**

Trong quá trình hoàn thành quyển báo cáo niên luận cơ sở ngành này, tôi xin gửi lời biết ơn sâu sắc và lòng thành kính đến những người đã luôn ở bên cạnh và hỗ trợ tôi từ đầu đến cuối. Trước hết, tôi muốn gửi lời cảm ơn đặc biệt đến cha và mẹ của mình. Những người đã luôn là nguồn động viên vô bờ bến, luôn tin tưởng và ủng hộ tôi trong mọi quyết định và nỗ lực của cuộc sống. Tôi cũng muốn bày tỏ lòng biết ơn chân thành đến hai người bạn thân của mình, Thịnh và Tấn. Những người đã luôn đồng hành, chia sẻ kinh nghiệm và sẵn lòng giúp đỡ tôi trong mọi tình huống. Đồng thời, tôi không thể quên gửi lời cảm ơn tới giáo viên hướng dẫn của tôi, cô Trương Thị Thanh Tuyền. Cô không chỉ là người đã truyền đạt kiến thức một cách rõ ràng và chuyên nghiệp mà còn là người đã dành thời gian và tâm huyết để hướng dẫn và hỗ trợ tôi trong quá trình thực hiện niên luận này. Cuối cùng, tôi xin gửi lời tri ân đến tất cả những người bạn bè, gia đình và giáo viên đã đóng góp vào sự thành công của dự án này. Sự ủng hộ và động viên từ các bạn và gia đình cũng như thầy cô là nguồn động viên lớn lao để tôi có thể hoàn thành dự án và quyển niên luận này.

**Nhận xét của giảng viên**

**Mục Lục**

[GIỚI THIỆU 5](#_Toc160028177)

[I. Đặt vấn đề 5](#_Toc160028178)

[II. Lịch sử giải quyết vấn đề 5](#_Toc160028179)

[III. Mục tiêu đề tài 6](#_Toc160028180)

[IV. Tổng quan về Godot Engine 6](#_Toc160028181)

[V. Đối tượng và phạm vi nghiên cứu 7](#_Toc160028182)

[1. Đối tượng nghiên cứu 7](#_Toc160028183)

[2. Phạm vi nghiên cứu 7](#_Toc160028184)

[3. Nội dung nghiên cứu 7](#_Toc160028185)

[VI. Những đóng góp chính của đề tài 8](#_Toc160028186)

[VII. Bố cục niên luận 8](#_Toc160028187)

[VIII. Kế hoạch dự án 9](#_Toc160028188)

[IX. Vai trò và nhiệm vụ của các thành viên trong nhóm 9](#_Toc160028189)

[NỘI DUNG 10](#_Toc160028190)

[I. MÔ TẢ HỆ THỐNG 10](#_Toc160028191)

[1. Mô tả chi tiết bài toán 10](#_Toc160028192)

[2. Các yêu cầu giao tiếp 11](#_Toc160028193)

[PHẦN KẾT LUẬN 12](#_Toc160028194)

[II. Kết quả đạt được 12](#_Toc160028195)

[1. Về lý thuyết 12](#_Toc160028196)

[2. Về chương trình 12](#_Toc160028197)

[III. Hạn chế và khó khăn 12](#_Toc160028198)

[1. Hạn chế: 12](#_Toc160028199)

[2. Khó khăn: 13](#_Toc160028200)

[I. Hướng phát triển 13](#_Toc160028201)

[1. Về hệ thống 13](#_Toc160028202)

[2. Về gameplay 13](#_Toc160028203)

[TÀI LIỆU THAM KHẢO 14](#_Toc160028204)

# GIỚI THIỆU

## Tổng quan về thể loại thủ thành

Trò chơi thủ thành (Tower defense) là một nhánh phụ của thể loại chiến lược thời gian thực. Mục tiêu chính của trò chơi này là ngăn chặn kẻ thù tràn qua một bản đồ bằng cách xây dựng các bẫy nhằm làm chậm bước tiến của chúng và sử dụng các tòa tháp để tiêu diệt chúng nếu chúng vượt qua. Kẻ thù và các tòa tháp thường có các đặc điểm khác nhau, cũng như chi phí và khả năng nâng cấp đa dạng.

Khi đánh bại kẻ địch, người chơi kiếm được tiền hoặc điểm, và họ sử dụng nguồn tài nguyên này để mua mới hoặc nâng cấp các tòa tháp hiện có, tăng cường khả năng kiếm tiền hoặc điểm, hoặc thậm chí nâng cấp tốc độ của việc kiếm nguồn tài nguyên.

Việc lựa chọn và đặt vị trí của các tòa tháp là một phần quan trọng của chiến lược trong trò chơi. Một số trò chơi cho phép người chơi xây dựng các mê cung để điều chỉnh con đường di chuyển của kẻ thù, như trong trò chơi Flash Element Tower Defense. Trong khi đó, một số trò chơi khác yêu cầu người chơi xây dựng các mê cung ngoài các tòa tháp của họ, ví dụ như trò chơi Desktop Tower Defense. Có những trò chơi kết hợp cả hai phong cách này, cho phép người chơi điều chỉnh con đường di chuyển của kẻ thù và sửa đổi vị trí của các tòa tháp, tạo ra sự đa dạng và sâu sắc trong chiến lược chơi game.

## Lịch sử giải quyết vấn đề

Platform game đã trở thành một thuật ngữ phổ biến cho thể loại này vào năm 1989, được phổ biến nhờ cách sử dụng nó trên báo chí Vương quốc Anh. Các ví dụ bao gồm việc coi "Super Mario mould" (chẳng hạn như Kato-chan & Ken-chan ) là platform games, và gọi Strider là trò chơi "platform games".

Từ những năm 1980, các game platform dần dần được hình thành và cho ra mắt những tựa game như [Space Panic](https://en.wikipedia.org/wiki/Space_Panic). Đây được xem như là tựa game đầu tiên của thể loại game đi cảnh và là phiên bản [arcade](https://vi.wikipedia.org/wiki/Tr%C3%B2_ch%C6%A1i_arcade) năm 1980 của [Universal](https://vi.wikipedia.org/wiki/Universal_Pictures). Hầu hết các game ra mắt trong thời gian này đều có một màn hình tĩnh và nhân vật sử dụng cơ chế leo trèo là chủ yếu.

Năm 1981, game [Donkey Kong](https://en.wikipedia.org/wiki/Donkey_Kong) - một trò chơi arcade do Nintendo tạo ra đã làm cho game platform có sự thu hút cực kỳ lớn với cộng đồng người chơi game thời bấy giờ. Game đánh dấu cho sự phát triển vì nó cho phép người chơi có thể nhảy qua các chướng ngại vật và khoảng trống. Chính vì điều này đã góp phần đưa tên tuổi của [Nintendo](https://www.nintendo.com/) vươn tầm quốc tế.

Tiwaz’s Adventure được tạo ra nhằm đáp ứng nhu cầu giải trí của người dùng game Platform cũng như đây là một sản phẩm để hoàn thành học phần Niên luận ngành Kỹ thuật phần mềm.

## Mục tiêu đề tài

2D Platform Game có một số tiêu chí chung:

* Nhân vật chính có đủ các yêu cầu: di chuyển, nhảy, tấn công, …
* Nhân vật có các chi số riêng: máu, độ cao khi nhảy, …
* Hệ thống bản đồ có độ tương tác
* Số lượng kẻ địch đa dạng
* Hệ thống UI hiển thị gọn gàng
* …

## Tổng quan về Godot Engine

Godot là một công cụ trò chơi mã nguồn mở , miễn phí và đa nền tảng được phát hành theo giấy phép MIT cho phép. Ban đầu nó được phát triển bởi các nhà phát triển phần mềm người Argentina Juan Linietsky và Ariel Manzur cho một số công ty ở Mỹ Latinh trước khi phát hành ra công chúng vào năm 2014. Môi trường phát triển chạy trên nhiều nền tảng và có thể xuất sang nhiều nền tảng khác. Nó được thiết kế để tạo cả trò chơi 2D và 3D nhắm mục tiêu vào nền tảng PC, thiết bị di động và web, đồng thời cũng có thể được sử dụng để phát triển phần mềm không phải trò chơi, bao gồm cả phần mềm chỉnh sửa.

Godot cho phép các nhà phát triển trò chơi điện tử tạo trò chơi 3D và 2D bằng nhiều ngôn ngữ lập trình, chẳng hạn như C++, C# và GDScript. Nó sử dụng hệ thống phân cấp để tạo điều kiện thuận lợi cho các nhà phát triển. Các lớp có thể được bắt nguồn từ một node để tạo ra các loại node chuyên biệt hơn kế thừa hành vi. Các nút được tổ chức bên trong các "scenes", là các nhóm node có thể tái sử dụng, có thể thực hiện được, có thể kế thừa và có thể lồng nhau. Tất cả tài nguyên trò chơi, bao gồm tập lệnh và nội dung đồ họa, được lưu dưới dạng một phần của hệ thống tệp của máy tính (chứ không phải trong cơ sở dữ liệu ). Giải pháp lưu trữ này nhằm tạo điều kiện thuận lợi cho sự cộng tác giữa các nhóm phát triển trò chơi sử dụng hệ thống kiểm soát phiên bản phần mềm .

Godot hỗ trợ nhiều ngôn ngữ lập trình để tạo trò chơi, bao gồm ngôn ngữ tích hợp GDScript, C++ và C#. Ngoài ra, công cụ này còn bao gồm GDNative, một phương tiện để tạo liên kết với các ngôn ngữ khác. Các ngôn ngữ GDNative được hỗ trợ chính thức bao gồm C và C++ . Các ngôn ngữ được cộng đồng hỗ trợ bao gồm Rust, Nim, Haskell, Clojure, Swift và D. Visual Coding cũng được hỗ trợ, thông qua ngôn ngữ tích hợp VisualScript, được thiết kế để tương đương trực quan với GDScript. Visual Scripting đã bị xóa khỏi core engine trong bản Godot 4.0. Các trò chơi Godot chạy trên trình duyệt có thể giao tiếp với mã JavaScript của trình duyệt. Trình soạn thảo Godot bao gồm một trình soạn thảo văn bản với tính năng tự động thụt lề , tô sáng cú pháp , hoàn thiện và gấp mã . Nó cũng có tính năng gỡ lỗi với khả năng thiết lập các điểm dừng và bước chương trình.

## Đối tượng và phạm vi nghiên cứu

### Đối tượng nghiên cứu

Đối tượng người dùng: là người có biết về các trò chơi, có niềm yêu thích với thể loại của game

### Phạm vi nghiên cứu

* Engine: Godot.
* Thể loại: platformer.
* Các công cụ hỗ trợ: Visual Studio, ngôn ngữ C#

### Nội dung nghiên cứu

* Tìm hiểu platformer game
* Phân tích hệ thống và thiết kế các lớp đối tượng
* Xây dựng hệ thống tương tác của các đối tượng.
* Nghiên cứu xây dựng hệ thống bản đồ
* Thiết kế giao diện, chức năng.

## Những đóng góp chính của đề tài

Xây dựng hệ thống tương tác chặt chẽ giữa các đối tượng, mang đến trải nghiệm thú vị cho người chơi.

Xây dựng giao diện đơn giản, trực quan, logic kế thừa các ý tưởng trước đây.

## Bố cục niên luận

Niên luận được tổ chức với 3 phần chính cùng với phần kết luận, tài liệu tham khảo.

Phần giới thiệu: Đặt ra vấn đề cần giải quyết, trong quá khứ đã có các hệ thống, website nào tương tự đã được xây dựng qua đó nêu lên mục tiêu của đề tài, nêu lên các nội dung nghiên cứu cho đề tài và những đóng góp chính mà đề tài mang lại.

Phần nội dung: gồm có các chương

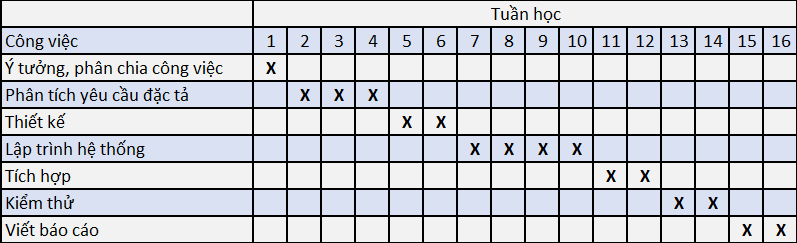
* + - * Mô tả hệ thống: mô tả bài toán, liệt kê chi tiết các chức năng cũng như diễn đạt lại gameplay, các tính chất của phần mềm trò chơi
      * Các yêu cầu giao tiếp:
  + Các yêu cầu giao tiếp bên ngoài: giao diện bên ngoài, giao tiếp phần cứng, giao tiếp phần mềm và truyền thông tin
  + Các yêu cầu chức năng: Sơ đồ trường hợp sử dụng, danh sách trường hợp sử dụng, bảng trường hợp sử dụng
    - Các yêu cầu phi chức năng: Yêu cầu về hiệu suất, Yêu cầu về tính tin cậy, Yêu cầu về an toàn thông tin, Yêu cầu về tính duy trì, Yêu cầu về tính khả dụng
      * Thiết kế và cài đặt giải pháp: đưa ra mô hình kiến trúc đã dùng, tài liệu thiết kế dữ liệu

Phần kiểm thử: bao gồm kế hoạch kiểm thử và các kế quá kiểm thử

Phần tổng kết: chỉ ra các nội dung đã hoàn thành được, cũng như các hạn chế và khó khăn trong quá trình hoạt động, đưa ra thêm hướng phát triển sau này cho dự án

Phần tài liệu tham khảo

## Kế hoạch dự án



## Vai trò và nhiệm vụ của các thành viên trong nhóm

|  |  |
| --- | --- |
| **Tên thành viên** | **Vai trò** |
| **Phan Phước Thịnh** | * Thiết kế Enemy * Thiết kế Game Manager * Thiết kế Player   + Kế hoạch kiểm thử (Test plan)  + Kịch bản thử nghiệm (Test Scenario)  + Kịch bản kiểm thử (Test case)   * Thiết kế sơ đồ CDM, LDM |
| **Huỳnh Thanh Nguyễn** | * Thiết kế các màn chơi * Thiết kế giao diện chính   + Kế hoạch kiểm thử (Test plan)  + Thiết kế sơ đồ PDM   * Sơ đồ PDM |
| **Phạm Đăng Khoa** | * Đặc tả LDM, PDM |

# NỘI DUNG

## MÔ TẢ HỆ THỐNG

### Mô tả chi tiết bài toán

* Hệ thống gồm có:
* User Interface (giao diện người dùng): hiển thị tương tác giữa người dùng và máy tính thông qua các nút (button), văn bản (text).
* Giao diện chính: gồm các nút tương tác như bắt đầu, tiếp tục, tùy chỉnh, quay lại, thoát.
* Giao diện chọn màn chơi.
* Giao diện dừng trò chơi: dùng để tạm dừng / tạm hoãn trong quá trình chơi.
* Level (màn chơi):
* Màn khởi động (tutorial): hướng dẫn người chơi cách điều khiển nhân vật trong trò chơi.
* Màn chơi bình thường: có các level được đánh số từ 1-4, quái đơn giản, độ khó Dễ,.
* Màn đánh boss: có một boss với độ thông minh cao hơn so với các loại quái thông thường, độ khó Trung Bình.
* Người chơi (Player): nhân vật do người chơi điều khiển từ bàn phím, có các trạng thái như:
* Di chuyển: Nhân vật sẽ có thể di chuyển thông qua thao tác của người chơi từ bàn phím: sang trái, sang phải.
* Nhảy: Nhân vật sẽ nhảy được thông qua thao tác của người chơi từ bàn phím: ấn hoặc ấn liên tục để nhảy theo từng độ cao.
* Tấn công kẻ địch: Nhân vật có thể tấn công kẻ địch khi người chơi thao tác trên bàn phím: ấn hoặc ấn giữ để tấn công bình thường hoặc tấn công liên tục.
* Máu: Nhân vật sẽ có mức máu riêng. Máu sẽ mất khi bị tấn công và sẽ chết khi không còn máu
* Enemy (quái): có 3 loại quái bình thường, tinh anh, trùm và các chỉ số như máu, chỉ số tấn công, khu vực tuần tra, phát hiện người chơi.
* Game Manager: là singleton quản lý các scene, lưu trữ vị trí nhân vật, vị trí bắt đầu và kết thúc của màn chơi. Bắt sự kiện đầu vào của người chơi và xử lý các chức năng Bắt đầu, Tiếp tục, Tạm dừng, Tùy chỉnh, Quay lại, Thoát.

### Các yêu cầu giao tiếp

#### Các yêu cầu giao tiếp bên ngoài

##### Về giao diện bên ngoài

* Giao diện được thiết kế với sự thân thiện để đáp ứng đa dạng người dùng một cách tự nhiên và dễ dàng.
* Màu sắc được sử dụng một cách cân đối, tạo ra sự thống nhất giữa các thành phần, mang đến một trải nghiệm thị giác hài hòa và thú vị.
* Quá trình tương tác được tối ưu hóa thông qua việc giảm thiểu các bước thao tác, giúp người dùng dễ dàng thực hiện các tính năng mà không gặp phải sự phức tạp không cần thiết.

##### Giao tiếp phần cứng

##### Kết nối phần cứng ổn định, không xuất hiện lỗi.

##### Số lượt truy cập không bị hạn chế, đảm bảo sự thuận tiện và linh hoạt cho người dùng.

##### Giao tiếp phần mềm và truyền thông tin

* Hệ điều hành cần hỗ trợ cài đặt trình duyệt web để đảm bảo sự tương thích và khả năng chạy mượt mà.

#### Các yêu cầu chức năng

##### Sơ đồ trường hợp sử dụng

* User:

# PHẦN KẾT LUẬN

## Kết quả đạt được

### Về lý thuyết

Biết thêm về lịch sử của thể loại game platform, tìm hiểu được thị trường trò chơi ở Việt Nam và thế giới cũng như nắm bắt được thị hiếu của người tiêu dùng.

Tiếp thu các kiến thức mới về Godot Engine, công cụ hỗ trợ các game developer. Hiểu được các hàm sẵn có, các công cụ hỗ trợ đến từ engine này.

Mặt khác, để phát triển thêm về đồ họa, nhóm đã có tìm hiểu một ít về các dòng pixel art và các công cụ hỗ trợ chỉnh sửa, sáng tạo đồ họa.

Về mặt âm thanh cũng đã học hỏi được các cách thức xử lý âm thanh và một số công cụ hỗ trợ

Nắm bắt được cách làm việc nhóm để có thể hoàn thành dự án

### Về chương trình

Tiếp thu và áp dụng các thư viện có sẵn do engine cung cấp cũng như các thuật toán quan trọng vào phần mềm trò chơi.

Thống nhất được ý kiến hướng đồ họa và đồng bộ nó vào phần mềm để tăng sự hài hòa, vừa mắt người dùng. Đồng nhất về phong cách âm nhạc, hiệu ứng âm thanh cho game.

## Hạn chế và khó khăn

### Hạn chế:

Làm việc trên một môi trường mới nên có một số vấn đề phải tìm hiểu kỹ lưỡng.

Do nhóm là lập trình viên nên các khá hạn chế trong quy trình trích xuất, phát triển đồ họa cũng như âm thanh.

Kỹ năng và kiến thức chuyên sâu về engine không đồng đều dẫn đến thời gian hoàn thành các quy trình hơi lệch ra khỏi kế hoạch.

### Khó khăn:

Lần đầu làm việc nhóm để hoàn thành một phần mềm trò chơi nên khi khởi động dự án đã có nhiều ý kiến dẫn đến xung đột, chưa thống nhất, phân chia công việc cũng gặp nhiều khó khăn.

Phải tiếp cận thêm nhiều công cụ nhỏ lẻ để có thể chỉnh chu hơn về âm thanh và hình ảnh cũng gây mất thời gian cho nhóm.

Tài liệu và các hướng dẫn hoàn toàn bằng ngoại ngữ dẫn đến việc thời gian bị kéo dài thêm để xử lý vấn đề.

Do là dự án nhỏ, dành cho học phần niên luận ngành nên việc bỏ chi phí để mua Asset nhằm giảm tải công việc dường như là không thể. Mặc khác để tìm các nguồn tài nguyên mở trên mạng cũng là một mặt khó khăn trong quá trình làm dự án.

## Hướng phát triển

### Về hệ thống

Tối ưu hóa thêm nữa về tốc độ xử lý cũng như các thao tác cần thiết để trãi nghiệm game.

Làm chuyển cảnh, animation mượt mà hơn nữa, cùng đó nâng cao thêm về nghe và nhìn cho phần mềm trò chơi.

### Về gameplay

Tạo thêm các màn chơi mới với các tính năng tương tác lạ hơn trước như dưới nước, trên thuyền, …

Sáng tạo thêm các chức năng, hành động mới cho người chơi như lướt, du dây, bắn, ném, …

Thêm nhiều nữa cốt truyện chính cho game cũng như các câu chuyện ngoài lề khác để góp phần xây dựng sự đa dạng, đặc sắc tổng thể.

# TÀI LIỆU THAM KHẢO

[1]. Wikipedia tiếng Việt, *“Trò chơi platform”.* Available: <https://vi.wikipedia.org/wiki/Trò_chơi_platform>, Access 2/11/2023.

[2]. Wikipedia English, *“Platform”* Available: [https://en.wikipedia.org/wiki/Platformer, Access](https://en.wikipedia.org/wiki/Platformer,%20Access): 02/11/2023.

[3]. Wikipedia English, *“Godot (game engine)”*. Available: <https://en.wikipedia.org/wiki/Godot_(game_engine)> , Access 15/11/2023.

[4]. Pixel Frog, *“Asset Pack Kings and Pigs”*, Available: <https://pixelfrog-assets.itch.io/kings-and-pigs> , Access 15/08/2023.

[5]. Christian Nguyễn, *“TƯ DUY THIẾT KẾ CƠ CHẾ ĐIỀU KHIỂN TRONG GAME”*, Available: <https://thietkegame.com/thiet-ke-game/tu-duy-thiet-ke-co-che-dieu-khien-trong-game/> , Access: 16/10/2023.

[6]. Yos Riady, *“Entity Component Systems in Elixir”*, Available: <https://yos.io/2016/09/17/entity-component-systems/> , Access 05/11/2023.

[7]. John Terra, *“Entity Component System: An Introductory Guide”*, Available:

<https://www.simplilearn.com/entity-component-system-introductory-guide-article> ,

Access 05/11/2023.

[7]. Trần Cao Đệ, Nguyễn Công Danh, Nxb. Đại học Cần Thơ, *“Giáo trình Đảm bảo chất lượng phần mềm”.*

[8]. Trần Cao Đệ, Đỗ Thanh Nghị, Nxb. Đại học Cần Thơ, *“Giáo trình Kiểm thử phần mềm”.*