

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN MÔN PHÁT TRIỂN TRÒ CHƠI**

**TÌM HIỂU VỀ GAME ENGINE**  
**KHÔNG CỐT TRUYỆN**

*Giảng viên hướng dẫn:* **ThS. Vũ Đình Hồng**

*Sinh viên thực hiện:* **Tô Vĩnh Khang - 51800408**

**Bùi Quang Khải - 51800785**

**Du Thuận Long - 51800429**

**Lớp: 18050203**

**Khóa: 22**

**Nhóm: 01**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN MÔN PHÁT TRIỂN TRÒ CHƠI**

# **TÌM HIỂU VỀ GAME ENGINE**

## **KHÔNG CỐT TRUYỆN**

*Giảng viên hướng dẫn:* **ThS. Vũ Đình Hồng**

*Sinh viên thực hiện:* **Tô Vĩnh Khang - 51800408**

**Bùi Quang Khải - 51800785**

**Du Thuận Long - 51800429**

**Lớp: 18050203**

**Khóa: 22**

**Nhóm: 01**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

# LỜI CẢM ƠN

Phát triển trò chơi là một môn học rất hay và bổ ích, chúng em xin chân thành cảm ơn thầy ThS.Vũ Đình Hồng đã hướng dẫn và tạo dựng những điều kiện tốt nhất để chúng em hoàn thành bài tập lớn môn Phát triển trò chơi với đề tài là Tìm hiểu game engine không cốt truyện. Giúp chúng em có thêm kiến thức về những giai đoạn từ lên ý tưởng đến hiện thực game, hiểu biết khái quát hơn về cách sử dụng game engine để xây dựng một game là như thế nào.

Trong quá trình thực hiện bài tập này nhóm vẫn khó tránh khỏi những sai sót không mong muốn, kính mong thầy có thể góp ý và giúp đỡ chúng em. Nhóm xin chân thành cảm ơn thầy.

# **BÀI TẬP LỚN NÀY ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Chúng tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của thầy Vũ Đình Hồng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài tập lớn này còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung bài tập lớn của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 20 tháng 10 năm 2020*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Tô Vĩnh Khang*

*(Trưởng nhóm)*

*Bùi Quang Khải*

*Du Thuận Long*

# PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

## Phần xác nhận của GV hướng dẫn

---

---

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm  
(ký và ghi họ tên)

## Phần đánh giá của GV chấm bài

---

---

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm  
(ký và ghi họ tên)

# TÓM TẮT

Trong xã hội ngày càng phát triển với sự bùng phát mạnh mẽ từ các dịch vụ giải trí, nhu cầu về một game thú vị mới lạ, mang lại cảm giác thử thách người chơi với lối chơi mạo hiểm, lôi cuốn ngày càng được chú trọng. Và từ đó các game không cốt truyện ra đời.

Để đáp ứng được nhu cầu trong bối cảnh thị trường hiện nay , KKL Software Team đã triển khai thực hiện dự án game mini 2D không cốt truyện mang tên Jumping Cube. Game được triển khai với mục tiêu giúp người chơi xả stress sau những ngày làm việc căng thẳng, tạo cảm giác muốn chinh phục trò chơi, thử thách bản thân,...

Về công cụ hỗ trợ phát triển game , nhóm sử dụng Godot Engine - một công cụ trò chơi 2D và 3D, đa nền tảng, miễn phí và mã nguồn mở được phát hành theo giấy phép của MIT và được phát triển ban đầu bởi Juan Linietsky và Ariel Manzurfor. Game Jumping Cube được triển khai với giao diện khá là phổ thông, thân thiện với người chơi, dễ dàng trong các thao tác di chuyển, nhảy để vượt chướng ngại vật là các khối vuông sinh ngẫu nhiên với các hình dạng khác nhau cùng với các kẻ thù là những chiếc nắm độc chết chóc khi va chạm.

Với đề tài “Tìm hiểu game engine không cốt truyện”, nhóm mong muốn áp dụng những kiến thức đã học trong trường cùng với việc tìm hiểu nghiên cứu, phân tích tâm lý người chơi, những kiến thức bên ngoài xã hội để xây dựng game mini không cốt truyện này đạt được hiệu quả cao nhất.

# MỤC LỤC

<b>LỜI CẢM ƠN.....</b>	<b>3</b>
<b>TÓM TẮT.....</b>	<b>6</b>
<b>MỤC LỤC.....</b>	<b>7</b>
<b>DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT.....</b>	<b>9</b>
<b>DANH MỤC CÁC BẢNG BIỂU VÀ HÌNH.....</b>	<b>10</b>
Danh mục Hình.....	10
Danh mục Bảng.....	11
<b>CHƯƠNG I: GIỚI THIỆU.....</b>	<b>12</b>
1.1 Giới thiệu đề tài.....	12
1.2 Phương pháp sử dụng.....	12
1.2.1 Game engine - Godot Engine.....	12
1.2.2 Ngôn ngữ lập trình - GDScript.....	13
1.3 Lý do và mục đích chọn đề tài.....	13
1.4 Một số thông tin về game.....	13
1.4.1 Tên game.....	13
1.4.2 Bản quyền.....	13
1.4.3 Phiên bản.....	13
1.4.4 Ý tưởng game.....	14
1.4.5 Thê loại.....	14
1.4.6 Đối tượng nhắm đến.....	14
1.4.7 Tóm tắt dòng game.....	14
1.4.8 Giao diện.....	15
1.4.9 Phạm vi dự án.....	15
1.4.9.1 Cấp độ.....	15
1.4.9.2 Số đối tượng.....	15
<b>CHƯƠNG II: PHÂN TÍCH.....</b>	<b>16</b>
2.1 Xác định các use case và các tác nhân.....	16
2.1.1 Xác định các tác nhân.....	16

2.1.2 Xác định các use case.....	16
2.2 Đặc tả game.....	17
<b>CHƯƠNG III: THIẾT KẾ.....</b>	<b>18</b>
3.1 Sơ đồ use case tổng quát.....	18
3.2 Các use case.....	18
3.2.1 Use case Xem menu game.....	18
3.2.2 Use case Xem hướng dẫn game.....	20
3.2.3 Use case Bắt đầu game.....	21
3.2.4 Use case Thoát game.....	22
3.2.5 Use case Điều chỉnh âm lượng.....	23
3.2.6 Use case Xem điểm số.....	24
3.2.7 Use case Tính điểm số.....	26
3.3 Sơ đồ hoạt động.....	27
3.4 Sơ đồ lớp.....	30
<b>CHƯƠNG IV: HIỆN THỰC.....</b>	<b>31</b>
4.1 Hiện thực bằng code.....	31
4.1.1 Lớp tác nhân.....	31
4.1.2 Lớp người chơi.....	31
4.1.3 Lớp kẻ thù.....	33
4.1.4 Lớp giao diện khi tạm dừng.....	33
4.1.5 Lớp cổng dịch chuyển.....	34
4.1.6 Lớp xu tính điểm.....	35
4.2 Hình ảnh trong game.....	36
4.2.1 Menu chính.....	36
4.2.2 Hướng dẫn chơi.....	36
4.2.3 Khung cảnh trong game.....	37
4.2.4 Giao diện tạm dừng khi người chơi chết hoặc muốn tạm dừng game...	37
4.2.5 Giao diện kết thúc game.....	38
<b>CHƯƠNG V: KẾT LUẬN.....</b>	<b>39</b>
<b>PHÂN CÔNG CÔNG VIỆC NHÓM.....</b>	<b>40</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>41</b>



# **DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT**

KKL: Tên nhóm

PC: Máy tính cá nhân

CSDL: Cơ sở dữ liệu

# DANH MỤC CÁC BẢNG BIỂU VÀ HÌNH

## Danh mục Hình

Hình 1 . Sơ đồ Use case tổng quát	18
Hình 2 . Use case Xem menu game	18
Hình 3 . Use case Xem hướng dẫn game	19
Hình 4 . Use case Bắt đầu game	20
Hình 5 . Use case Thoát game	21
Hình 6 . Use case Điều chỉnh âm lượng	22
Hình 7 . Use case Xem điểm số	23
Hình 8 . Use case Tính điểm số	24
Hình 9 . Sơ đồ hoạt động của usecase Xem menu game	26
Hình 10 . Sơ đồ hoạt động của usecase Bắt đầu game	26
Hình 11 . Sơ đồ hoạt động của usecase Thoát game	27
Hình 12 . Sơ đồ hoạt động của usecase Điều chỉnh âm lượng	27
Hình 13 . Sơ đồ hoạt động của usecase Xem điểm số	28
Hình 14 . Sơ đồ hoạt động của usecase Tính điểm số	28
Hình 15 . Sơ đồ lớp	29
Hình 16 . Giao diện menu chính	35
Hình 17 . Giao diện hướng dẫn	35
Hình 18 . Giao diện trò chơi	36
Hình 19 . Giao diện menu thua cuộc	36
Hình 20 . Giao diện menu kết thúc trò chơi	37

## **Danh mục Bảng**

Bảng 1 . Danh sách các Actor	15
Bảng 2 . Danh sách các Use case	15
Bảng 3 . Đặc tả Use case Xem menu game	18
Bảng 4 . Đặc tả Use case Xem hướng dẫn game	19
Bảng 5 . Đặc tả Use case Bắt đầu game	20
Bảng 6 . Đặc tả Use case Thoát game	21
Bảng 7 . Đặc tả Use case Điều chỉnh âm lượng	22
Bảng 8 . Đặc tả Use case Xem điểm số	23
Bảng 9 . Đặc tả Use case Tính điểm số	24

# CHƯƠNG I: GIỚI THIỆU

## 1.1 Giới thiệu đề tài

Trong xã hội ngày càng phát triển với sự bùng phát mạnh mẽ từ các dịch vụ giải trí, nhu cầu về một game thú vị mới lạ, mang lại cảm giác thử thách người chơi với lối chơi mạo hiểm, lôi cuốn ngày càng được chú trọng. Và từ đó các game không cốt truyện ra đời.

Để đáp ứng được nhu cầu trong bối cảnh thị trường hiện nay, KKL Software Team đã triển khai thực hiện dự án game mini 2D không cốt truyện mang tên Jumping Cube bằng Godot Engine. Game được triển khai với mục tiêu giúp người chơi xả stress sau những ngày làm việc căng thẳng, tạo cảm giác muốn chinh phục trò chơi, thử thách bản thân,... Jumping Cube được triển khai với giao diện khá là phổ thông, thân thiện với người chơi, dễ dàng trong các thao tác di chuyển, nhảy để vượt chướng ngại vật là các khối vuông sinh ngẫu nhiên với các hình dạng khác nhau cùng với các nắm độc đầy nguy hiểm.

## 1.2 Phương pháp sử dụng

*Nhóm sử dụng nền tảng ngôn ngữ GDScript và Godot Engine để hiện thực game.*

### 1.2.1 Game engine - Godot Engine

Godot Engine là một game engine 2D và 3D, đa nền tảng và miễn phí với mã nguồn mở. Nhằm mục đích cung cấp một môi trường phát triển game tích hợp đầy đủ cho phép các nhà phát triển tạo game từ đầu mà không cần công cụ nào khác ngoài những công cụ được sử dụng để tạo nội dung.

Kiến trúc của Godot Engine được xây dựng xung quanh khái niệm cây nút (nodes). Các nút được tổ chức bên trong cảnh (scene), là các nhóm nút có thể sử dụng lại, kế thừa hay lồng vào nhau. Tài nguyên game 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 CSDL nhằm tạo điều kiện hợp tác giữa các nhóm phát triển game sử dụng hệ thống kiểm soát phiên bản phần mềm.

Môi trường phát triển chạy trên nhiều hệ điều hành bao gồm Windows, macOS, Linux. Godot Engine có thể tạo trò chơi nhắm mục tiêu nền tảng PC, Web và di động.

### **1.2.2 Ngôn ngữ lập trình - GDScript**

GDScript là một ngôn ngữ lập trình bậc cao. Nó sử dụng cú pháp tương tự như Python với các khối dựa trên thụt lề và nhiều từ khóa tương tự. Mục tiêu của nó là được tối ưu hóa và tích hợp chặt chẽ với Godot Engine, cho phép tạo và tích hợp nội dung một cách linh hoạt nhất.

## **1.3 Lý do và mục đích chọn đề tài**

Thứ nhất, chúng em chọn đề tài này là vì muốn có thêm kiến thức về việc tìm hiểu thêm các game engine khác trên thị trường (cụ thể là Godot engine), cách thức hoạt động của game engine và cấu trúc ngôn ngữ của nó (GDScript).

Thứ hai, loại hình game không cốt truyện này đang ngày càng phát triển và có chỗ đứng trong xã hội. Khi mà nhu cầu đòi hỏi về những game mang tính chinh phục , thú vị, lôi cuốn.

## **1.4 Một số thông tin về game**

### **1.4.1 Tên game**

Jumping Cube - Hành trình của khối vuông

### **1.4.2 Bản quyền**

Bản quyền thuộc về KKL Software Team

### **1.4.3 Phiên bản**

Phiên bản 1.0 - Ban hành ngày 31/10/2020

Tác giả: KKL Software Team

#### **1.4.4 Ý tưởng game**

Một hình vuông 2D di chuyển vượt qua các chướng ngại vật và các nấm độc đầy nguy hiểm tiềm tàng trên đường. Càng di chuyển được nhiều và nhặt các đồng xu thì điểm số của người chơi càng cao và có thể lưu lại điểm số cao nhất trong những lần chơi. Khi người chơi va chạm nấm độc thì thua cuộc. Trong quá trình di chuyển thì độ khó cũng ngày một tăng dần qua các vòng thông qua cổng xuyên không. Ngoài ra, còn có sự hỗ trợ bật nhảy đến từ thiết bị bật nhảy - Phụộc nhún, giúp người chơi có thể vượt qua những chướng ngại vật khó. Các hình khối vuông dị dạng được sinh ra phức tạp hơn cũng như các nấm độc ngày một nhiều dần lên trên đường đi.

#### **1.4.5 Thể loại**

Phiêu lưu, thử thách, chinh phục.

#### **1.4.6 Đối tượng nhắm đến**

Độ tuổi: Mọi lứa tuổi

#### **1.4.7 Tóm tắt dòng game**

Mở đầu game người chơi sẽ loading vào màn hình hiển thị main menu của game với các tính năng “Start Game”, “Exit Game”, “Volume”, “Guide”. Đối với nút “Start Game” người chơi sẽ bắt đầu game bằng nhân vật là một hình khối vuông 2D di chuyển lần lượt vượt các chướng ngại vật hình khối. Đồng thời phải tránh né mối nguy hiểm chết chóc là các nấm độc trên đường di chuyển. Độ khó của game sẽ tăng dần (về hình dạng các chướng ngại vật sinh ra lẫn các nấm độc ngày càng nhiều) đến khi người chơi va phải nấm độc và thua cuộc. Đối với nút “Exit Game” , người chơi sẽ thoát khỏi ứng dụng. “Volume” sẽ giúp người chơi thay đổi âm lượng cho phù hợp với sở thích. Và cuối cùng là “Guide” để hiển thị bảng hướng dẫn luật chơi.

1.4.8 Giao diện

Giao diện cơ bản của game: Góc nhìn thứ ba

Hình ảnh khi chơi: Khối hình vuông 2D di chuyển trong khung cảnh các chướng ngại vật hình khối sinh ra các hình dạng ngẫu nhiên, đan xen vào đó là nấm độc trên đường di chuyển. Có các cổng xuyên không để chuyển đến cấp độ tiếp theo. Sự hỗ trợ bật nhảy đến từ Phụộc nhún.

1.4.9 Phạm vi dự án

1.4.9.1 Cấp độ

Cấp độ khó dần dựa trên hình dạng của Chướng ngại vật và số lượng Nấm độc ngày càng nhiều.

1.4.9.2 Số đối tượng

Đối tượng	Mô tả
Khối vuông (Player)	Người chơi chính
Khối dị dạng (Obstacle)	Chướng ngại vật
Cổng xuyên không (Gateway)	Sang cấp độ tiếp theo
Nấm độc (Poison Mushroom)	Kẻ thù

## CHƯƠNG II: PHÂN TÍCH

### 2.1 Xác định các use case và các tác nhân

#### 2.1.1 Xác định các tác nhân

Sau khi đã tìm hiểu, thu thập thông tin từ một số loại Game mini 2D không cốt truyện, chúng em đã đúc kết được tác nhân của game Jumping Cube như sau:

STT	Tác nhân	Mô tả
1	Người chơi	Truy cập vào ứng dụng và chơi game.
2	Hệ thống	Thực hiện một số thao tác khác trong hệ thống

*Bảng 1. Danh sách các Actor*

#### 2.1.2 Xác định các use case

Dựa vào yêu cầu đặc ra, chúng em xác định use case cho game này gồm:

ID	Use case	Mô tả
VM	Xem menu game	Xem các chức năng của game
VH	Xem hướng dẫn game	Xem các hướng dẫn của game
SG	Bắt đầu game	Chọn và bắt đầu chơi game
EG	Thoát game	Chọn và thoát game.
AV	Điều chỉnh âm lượng	Chọn và kéo để thay đổi âm lượng nhạc nền
VS	Xem điểm số	Xem điểm số đạt được và điểm số cao nhất
CS	Tính điểm số	Hệ thống tính điểm đạt được và điểm cao nhất

*Bảng 2. Danh sách các Use case*



## 2.2 Đặc tả game

Game mini 2D không cốt truyện - Jumping Cube cung cấp dịch vụ cho đối tượng là mọi lứa tuổi người chơi. Do đó, bối cảnh game phải có màu sắc tươi vui, nhạc nền sống động. Các chức năng hệ thống phải được tổng hợp tại phần menu game. Giao diện game dễ sử dụng, luật chơi đơn giản dễ hiểu. Người chơi sau khi nhấn vào ứng dụng Jumping Cube sẽ vào trực tiếp menu game mà không cần tài khoản đăng nhập.

Game cho phép xem menu game về các chức năng của game như Xem hướng dẫn game (Guide), Bắt đầu game (Start Game), Thoát game (Exit Game), Điều chỉnh âm lượng (Volume). Riêng chức năng Xem điểm số sẽ được hiển thị sau khi người chơi thua cuộc.

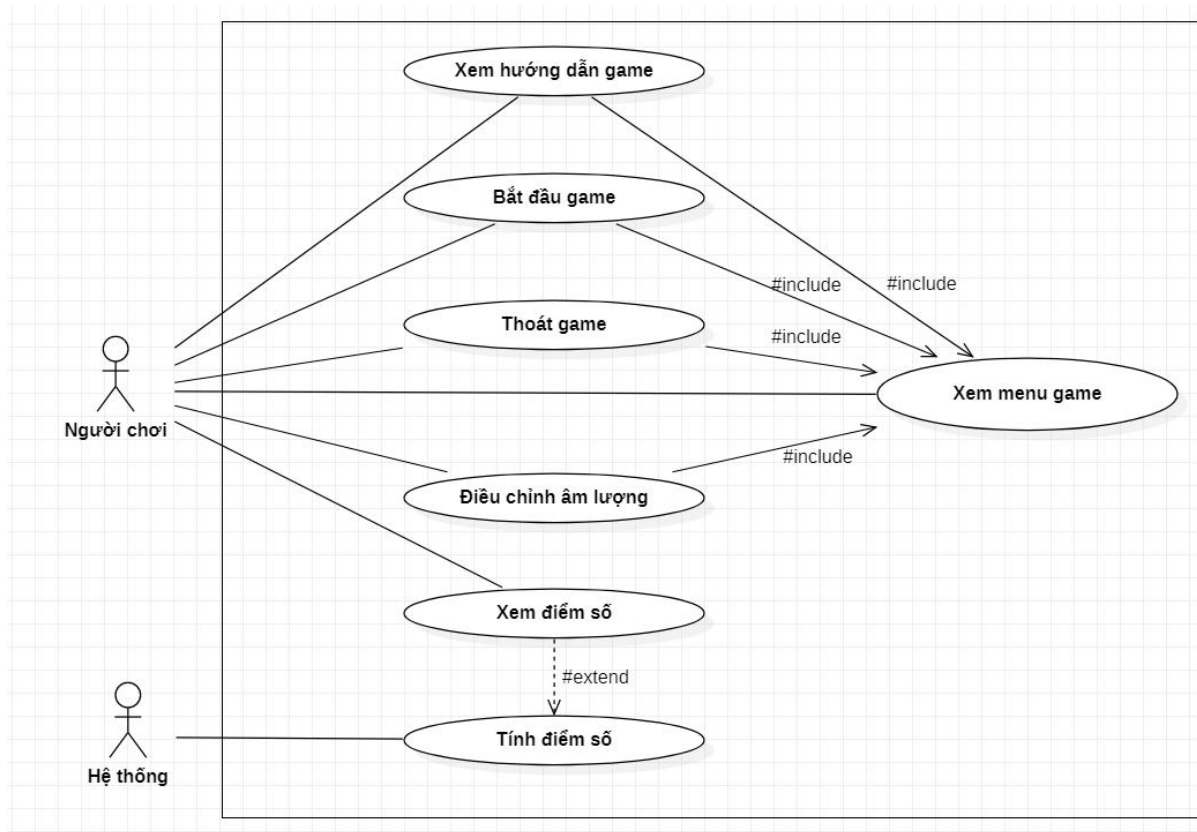
Đối với chức năng Xem hướng dẫn game sẽ hiển thị một bảng hướng dẫn luật chơi, các sự kiện trong game và các thao tác di chuyển. Đối với chức năng Bắt đầu game, người chơi sẽ vào trực tiếp game và thông tin về điểm số và cấp độ sẽ được cập nhật lại và hiển thị cho người chơi. Đối với chức năng Thoát game, người chơi sẽ thoát hoàn toàn ứng dụng và thông tin về điểm số và cấp độ cũng sẽ được lưu lại. Đối với chức năng Điều chỉnh âm lượng sẽ được hiển thị dưới dạng thanh kéo ngang trong khoảng từ 0 đến 100 tương ứng với số phần trăm âm lượng nhạc nền game. Cuối cùng là chức năng Xem điểm số, điểm được cộng dồn liên tục và chỉ hiển thị khi người chơi va chạm với nấm độc và thua cuộc. Đồng thời so sánh và trích xuất điểm số cao nhất trong các lần chơi của người chơi.

Một số sự kiện trong game như đồng xu (Coin), Nấm độc (Poison Mushroom), Phục nhún (Shrug), Cổng xuyên không (Gateway) sẽ được bố trí theo từng cấp độ.

Đối với đồng xu tương ứng với 10 điểm (Score). Đối với Nấm độc sẽ tự động di chuyển qua lại và tính toán va chạm với Khối vuông (Player). Đối với Phục nhún sẽ giúp người chơi bật nhảy trong một khoảng chiều cao nhất định. Đối với Cổng xuyên không sẽ điều hướng qua vòng tiếp theo.

# CHƯƠNG III: THIẾT KẾ

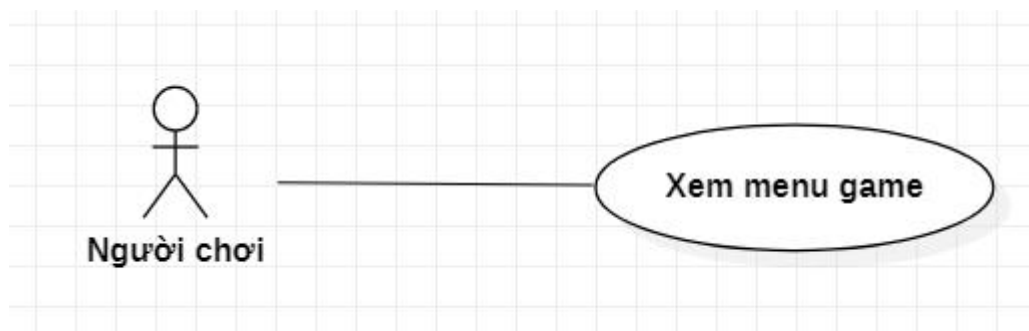
## 3.1 Sơ đồ use case tổng quát



Hình 1. Sơ đồ Use case tổng quát

## 3.2 Các use case

### 3.2.1 Use case Xem menu game

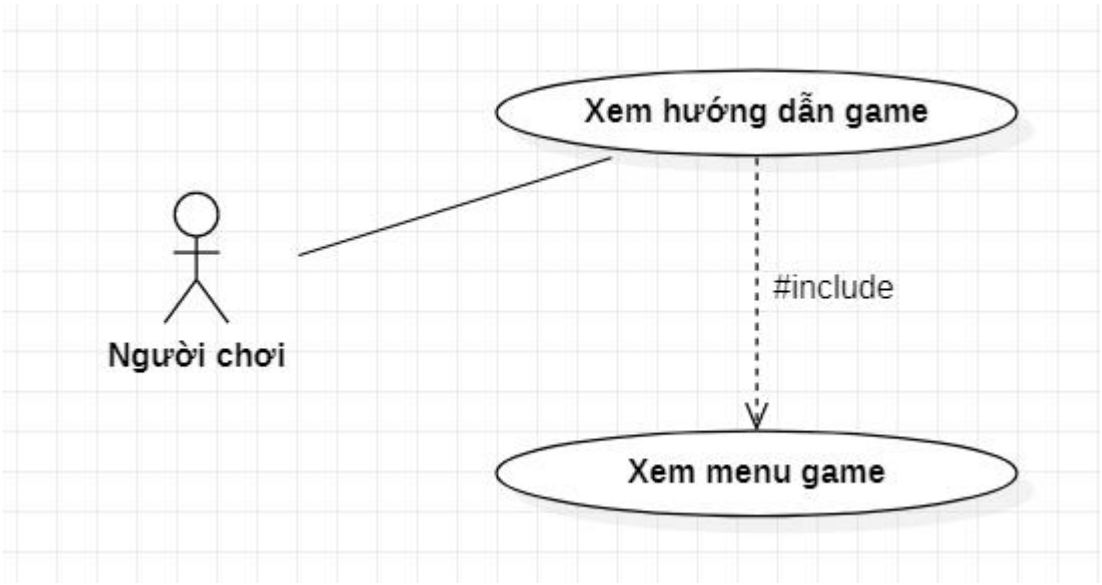


Hình 2. Use case Xem menu game

Use Case Name	Xem menu game	
Scenario	Ứng dụng Jumping Cube được mở	
Triggering event	Truy cập ứng dụng Jumping Cube	
Brief description	Người chơi dùng chức năng này để thao tác chọn lựa cho các chức năng khác trong game.	
Actors	Người chơi	
Related Use Case	Xem hướng dẫn game, Bắt đầu game, Thoát game.	
Stakeholder	Không	
Precondition	Truy cập được vào ứng dụng game	
Postcondition	Không	
Flow of activities	Người dùng	Hệ thống
	1. Người chơi nhấn vào nút ứng dụng game Jumping Cube	1.1 Hệ thống hiển thị lên giao diện menu để lựa chọn
Exception condition	Không	

***Bảng 3. Đặc tả Use case Xem menu game***

3.2.2 Use case Xem hướng dẫn game



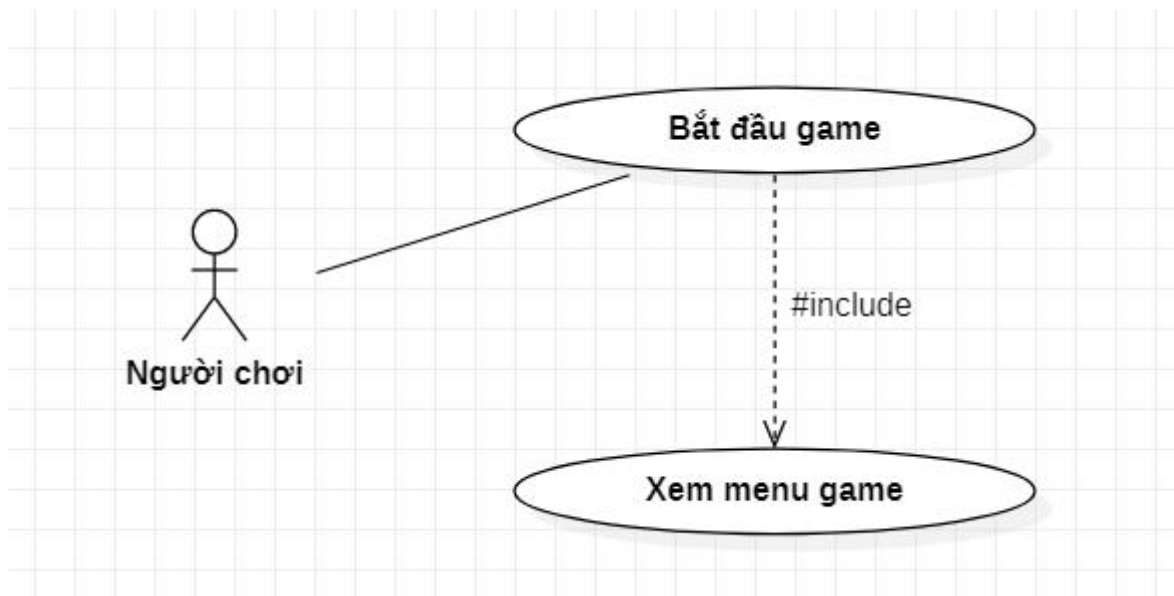
Hình 3. Use case Xem hướng dẫn game

Use Case Name	Xem hướng dẫn game	
Scenario	Nhấn vào nút Guide trong menu	
Triggering event	Trang Xem menu game đã được hiển thị	
Brief description	Người chơi dùng chức năng này để xem các hướng dẫn về luật chơi và hiểu được các thao tác di chuyển trong game	
Actors	Người chơi	
Related Use Case	Xem menu game	
Stakeholder	Không	
Precondition	Menu game đã load lên đầy đủ	
Postcondition	Chỉ nhân xem được hướng dẫn khi vào giao diện menu game	
Flow of activities	Người dùng	Hệ thống
	1. Người chơi nhấn vào nút	1.1 Hệ thống hiển thị một

	Guide	bảng các hướng dẫn về game
Exception condition	Không	

**Bảng 4. Đặc tả Use case Xem hướng dẫn game**

### 3.2.3 Use case Bắt đầu game



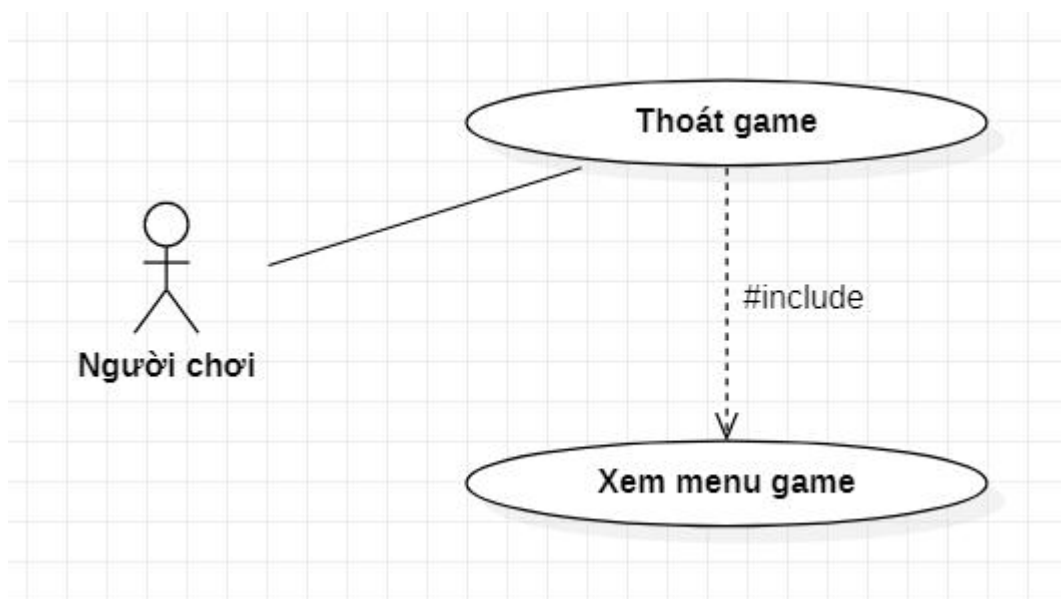
**Hình 4. Use case Bắt đầu game**

Use Case Name	Bắt đầu game
Scenario	Nhấn vào nút Start Game trong menu
Triggering event	Trang Xem menu game đã được hiển thị
Brief description	Người chơi dùng chức năng này để truy cập vào game
Actors	Người chơi
Related Use Case	Xem menu game
Stakeholder	Không
Precondition	Menu game đã load lên đầy đủ

Postcondition	Chỉ nhấn bắt đầu game được khi vào giao diện menu game	
Flow of activities	Người dùng	Hệ thống
	1. Người chơi nhấn vào nút Start Game	1.1 Hệ thống sẽ điều hướng sang scene chính (main scene)
Exception condition	Không	

***Bảng 5. Đặc tả Use case Bắt đầu game***

### 3.2.4 Use case Thoát game



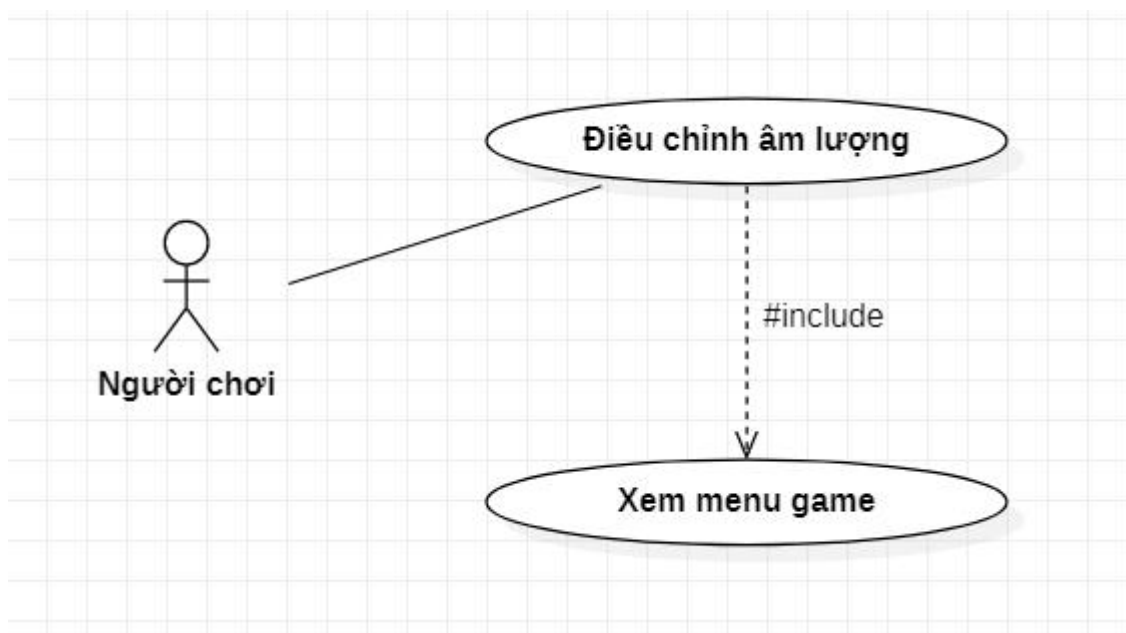
***Hình 5. Use case Thoát game***

Use Case Name	Thoát game
Scenario	Nhấn vào nút Exit Game trong menu
Triggering event	Trang Xem menu game đã được hiển thị
Brief description	Người chơi dùng chức năng này để thoát khỏi ứng dụng
Actors	Người chơi

Related Use Case	Xem menu game	
Stakeholder	Không	
Precondition	Menu game đã load lên đầy đủ	
Postcondition	Chỉ nhấn thoát game được khi vào giao diện menu game	
Flow of activities	Người dùng	Hệ thống
	1.Người chơi nhấn vào nút Exit Game	1.1 Hệ thống sẽ chuyển người chơi ra khỏi ứng dụng
Exception condition	Không	

***Bảng 6. Đặc tả Use case Thoát game***

### 3.2.5 Use case Điều chỉnh âm lượng



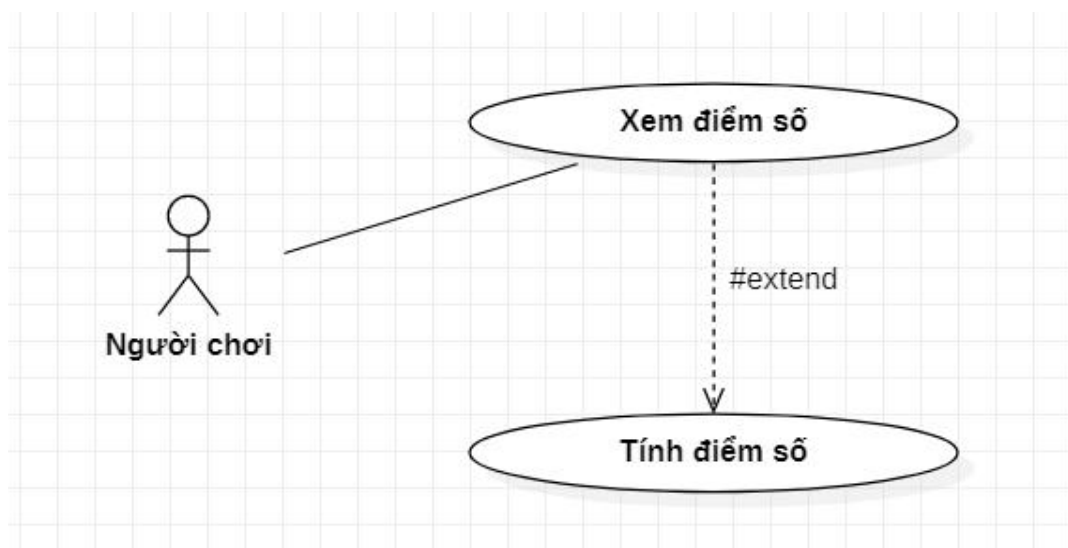
***Hình 6. Use case Điều chỉnh âm lượng***

Use Case Name	Điều chỉnh âm lượng
Scenario	Nhấn vào nút Volume trong menu và thanh cuộn ngang được hiển thị

Triggering event	Trang Xem menu game đã được hiển thị	
Brief description	Người chơi dùng chức năng này để điều chỉnh tăng giảm âm lượng nhạc nền game	
Actors	Người chơi	
Related Use Case	Xem menu game	
Stakeholder	Không	
Precondition	Menu game đã load lên đầy đủ	
Postcondition	Chỉ có thể điều chỉnh âm lượng được khi vào giao diện menu game và hiển thị thanh cuộn ngang	
Flow of activities	Người dùng	Hệ thống
	1. Người chơi nhấn vào nút Volume	1.1 Hệ thống sẽ hiển thị thanh cuộn ngang để điều chỉnh âm lượng trong khoảng từ 0 đến 100 tương ứng với số phần trăm âm lượng nhạc nền.
Exception condition	Không	

**Bảng 7. Đặc tả Use case Điều chỉnh âm lượng**

### 3.2.6 Use case Xem điểm số



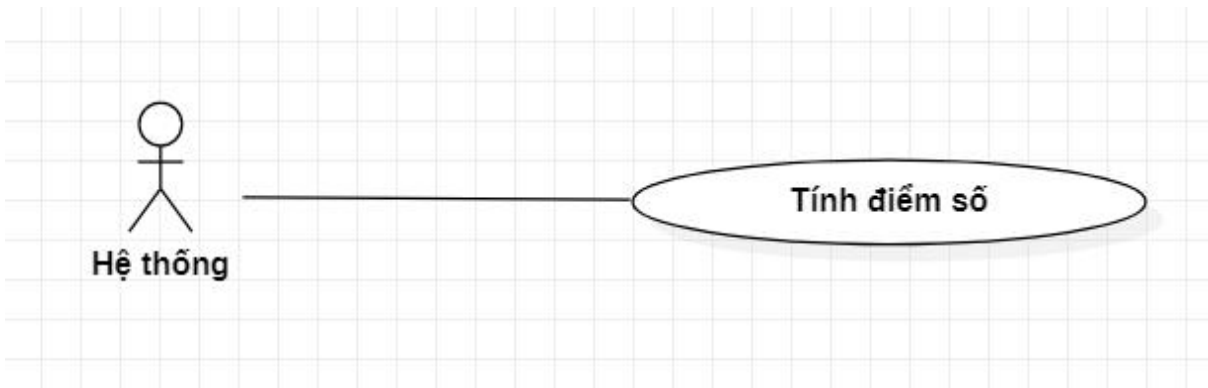
**Hình 7. Use case Xem điểm số**



Use Case Name	Xem điểm số	
Scenario	Khi người chơi chết thì hiển thị bảng điểm số	
Triggering event	Khi người chơi va chạm với nấm độc	
Brief description	Người chơi dùng chức năng này để điều chỉnh xem điểm số đạt được và điểm số cao nhất trong các lần chơi	
Actors	Người chơi	
Related Use Case	Tính điểm số	
Stakeholder	Hệ thống	
Precondition	Va chạm với nấm độc	
Postcondition	Hiển thị bảng điểm đã được thống kê từ Tính điểm số	
Flow of activities	Người dùng	Hệ thống
	1. Người chơi va chạm với nấm độc	1.1 Hệ thống sẽ tự động thu thập và thống kê điểm số đạt được, tính toán điểm số cao nhất từ Tính điểm số và truy xuất dữ liệu sang Xem điểm số và hiển thị lên bảng điểm số cho người chơi.
Exception condition	Không	

***Bảng 8. Đặc tả Use case Xem điểm số***

3.2.7 Use case Tính điểm số



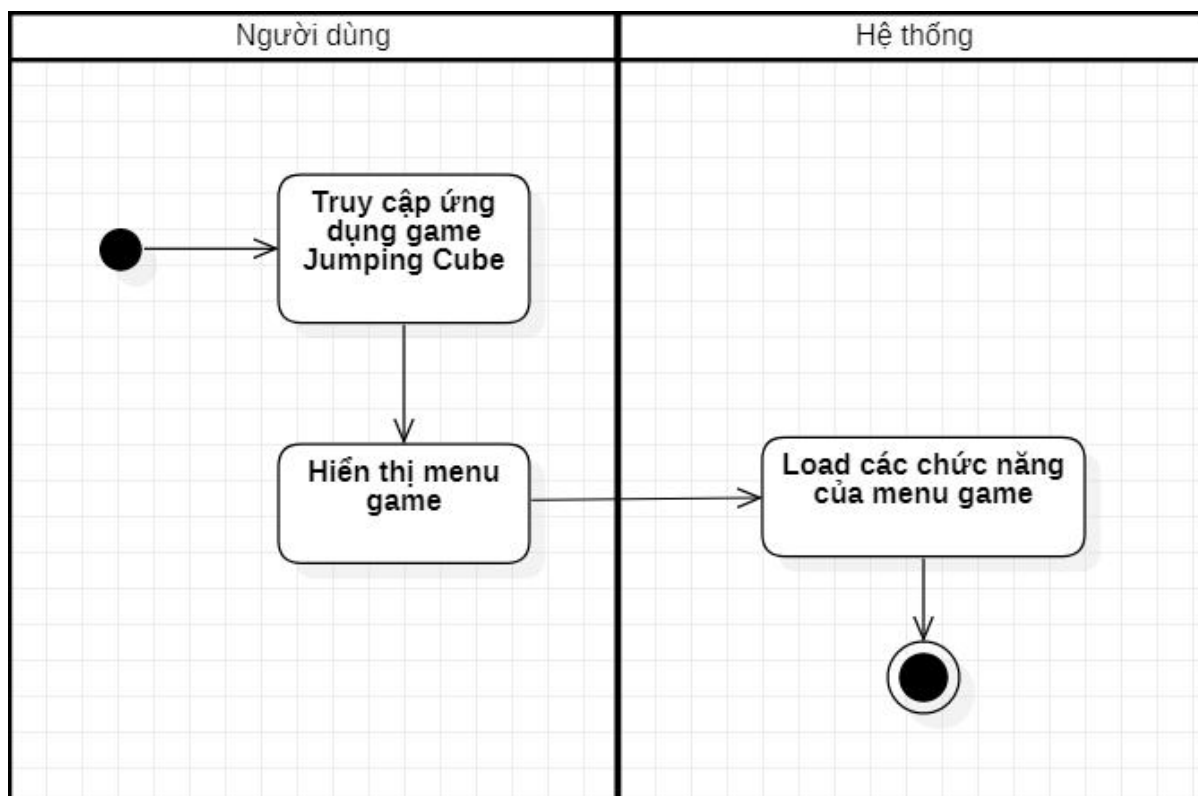
Hình 8. Use case Tính điểm số

Use Case Name	Tính điểm số	
Scenario	Tự động thu thập thông tin điểm số liên tục, tính toán và truy xuất điểm số đạt được và điểm số cao nhất	
Triggering event	Khi va chạm đồng xu, vượt qua các cấp độ trong game	
Brief description	Tác nhân Hệ thống dùng chức năng này để tự động thu thập, tính toán và cập nhật điểm số mới nhất và điểm số cao nhất trong các lần chơi. Khi người chơi va chạm với nắm độc thì sẽ truy xuất điểm số và hiển thị lên bảng Xem điểm số	
Actors	Hệ thống	
Related Use Case	Xem điểm số	
Stakeholder	Không	
Precondition	Không	
Postcondition	Biến Score sẽ trả về 0 khi người chơi bắt đầu lại game và biến highScore sẽ lưu giá trị lớn nhất của Score.	
Flow of activities	Người dùng	Hệ thống
	1.Tác nhân Hệ thống thu thập thông tin điểm số thông qua các va chạm với đồng xu và công xuyên không	1.1 Hệ thống sẽ cộng dồn các điểm số qua các tính toán va chạm. Điểm số được cập nhật liên tục. Điểm số cao nhất là giá trị lớn nhất của các lần thống kê điểm số.

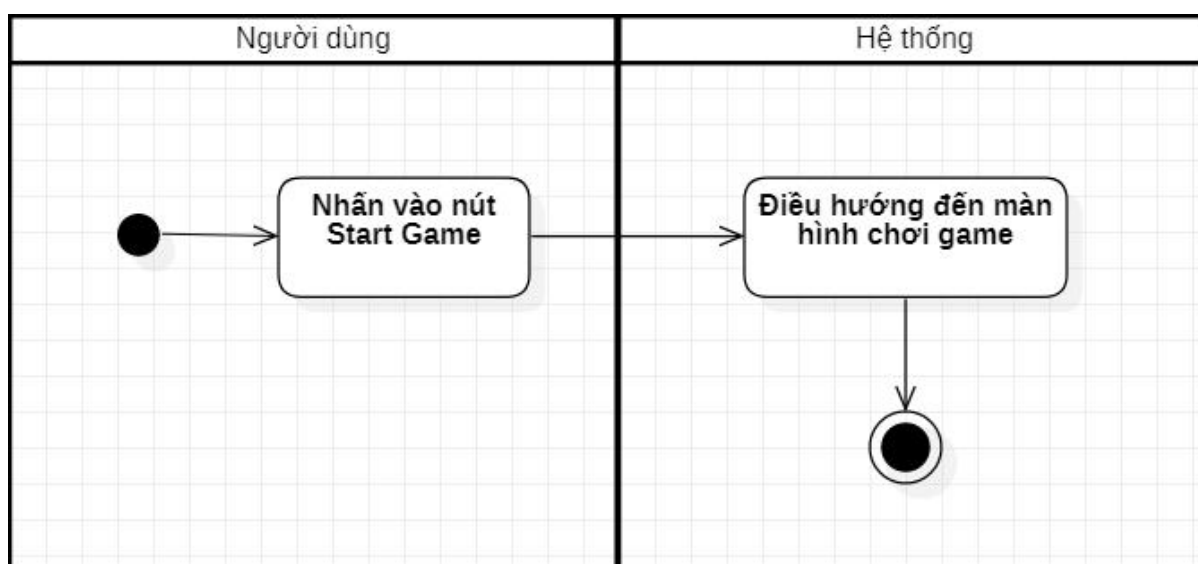
Exception condition	Không
---------------------	-------

**Bảng 9. Đặc tả Use case Tính điểm số**

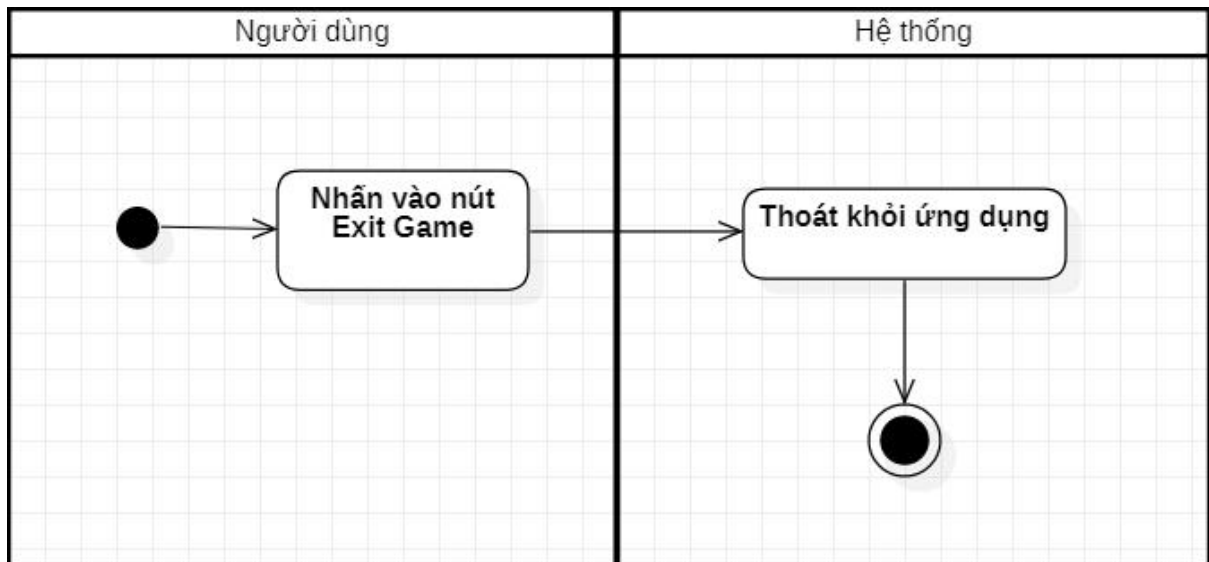
### 3.3 Sơ đồ hoạt động



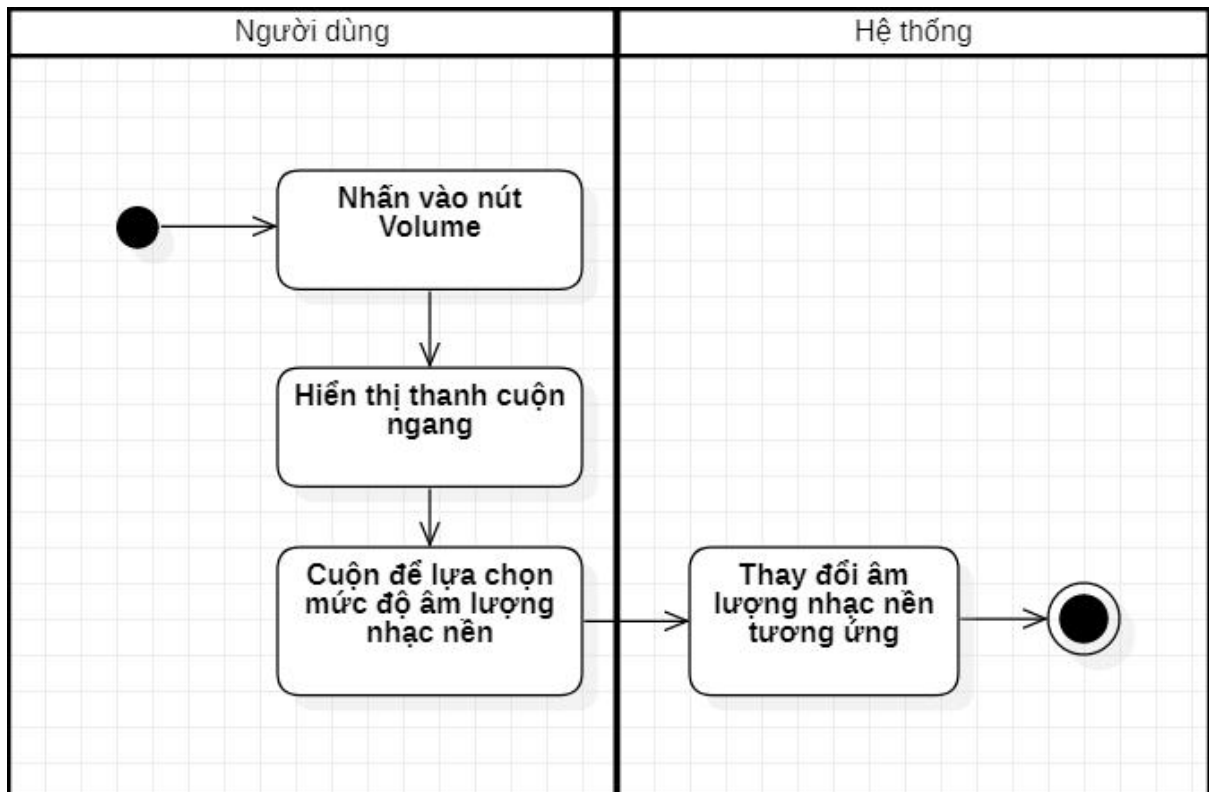
**Hình 9. Sơ đồ hoạt động của usecase Xem menu game**



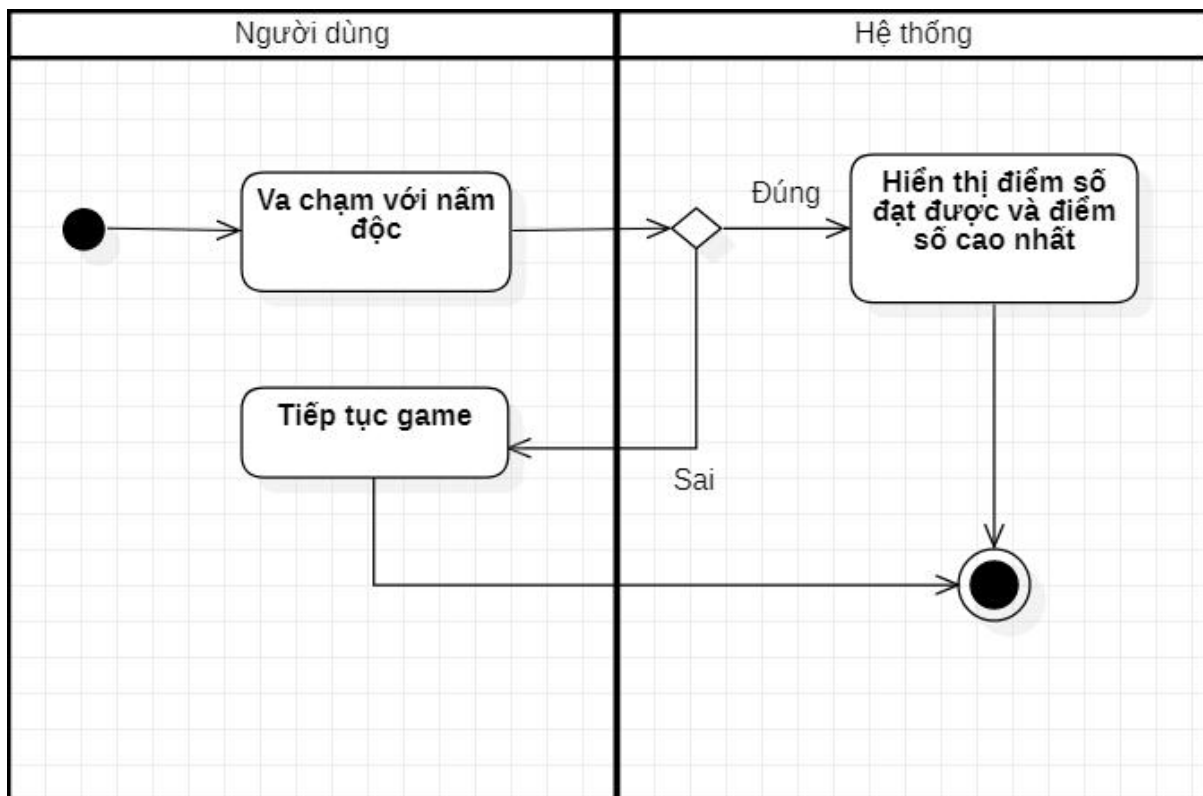
**Hình 10. Sơ đồ hoạt động của usecase Bắt đầu game**



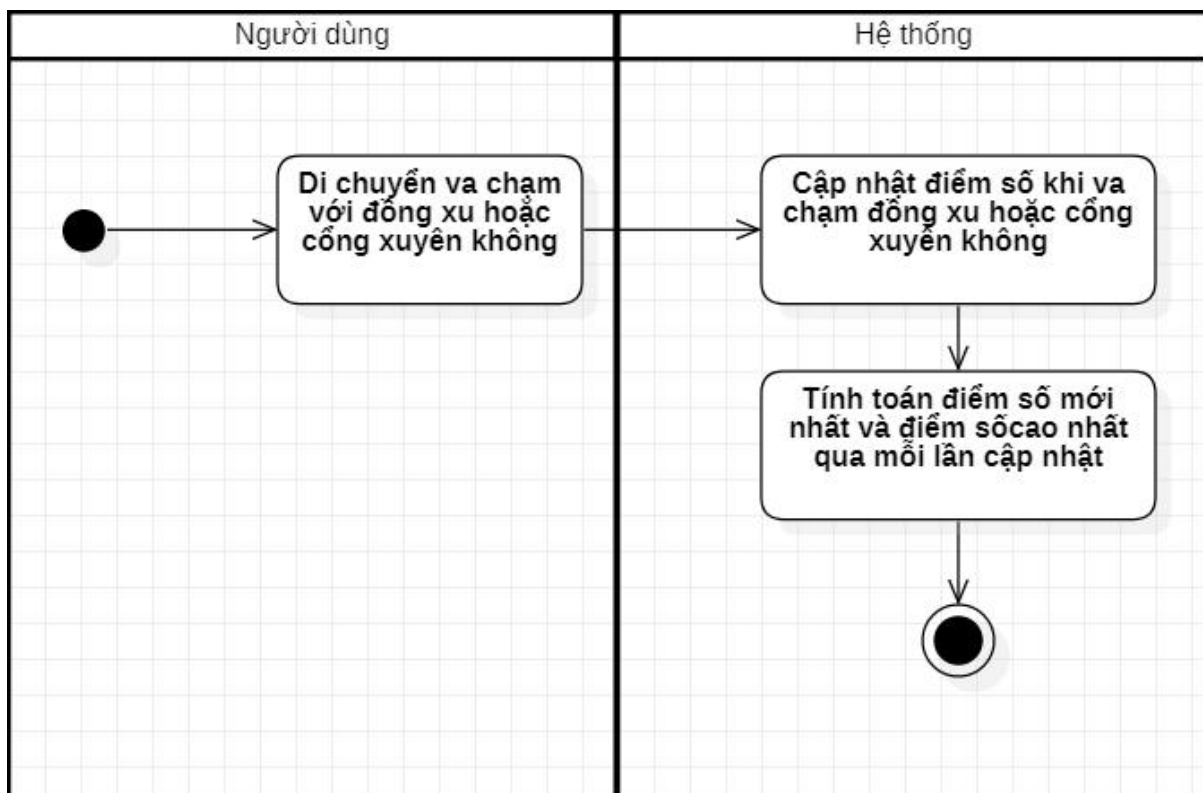
*Hình 11. Sơ đồ hoạt động của usecase Thoát game*



*Hình 12. Sơ đồ hoạt động của usecase Điều chỉnh âm lượng*

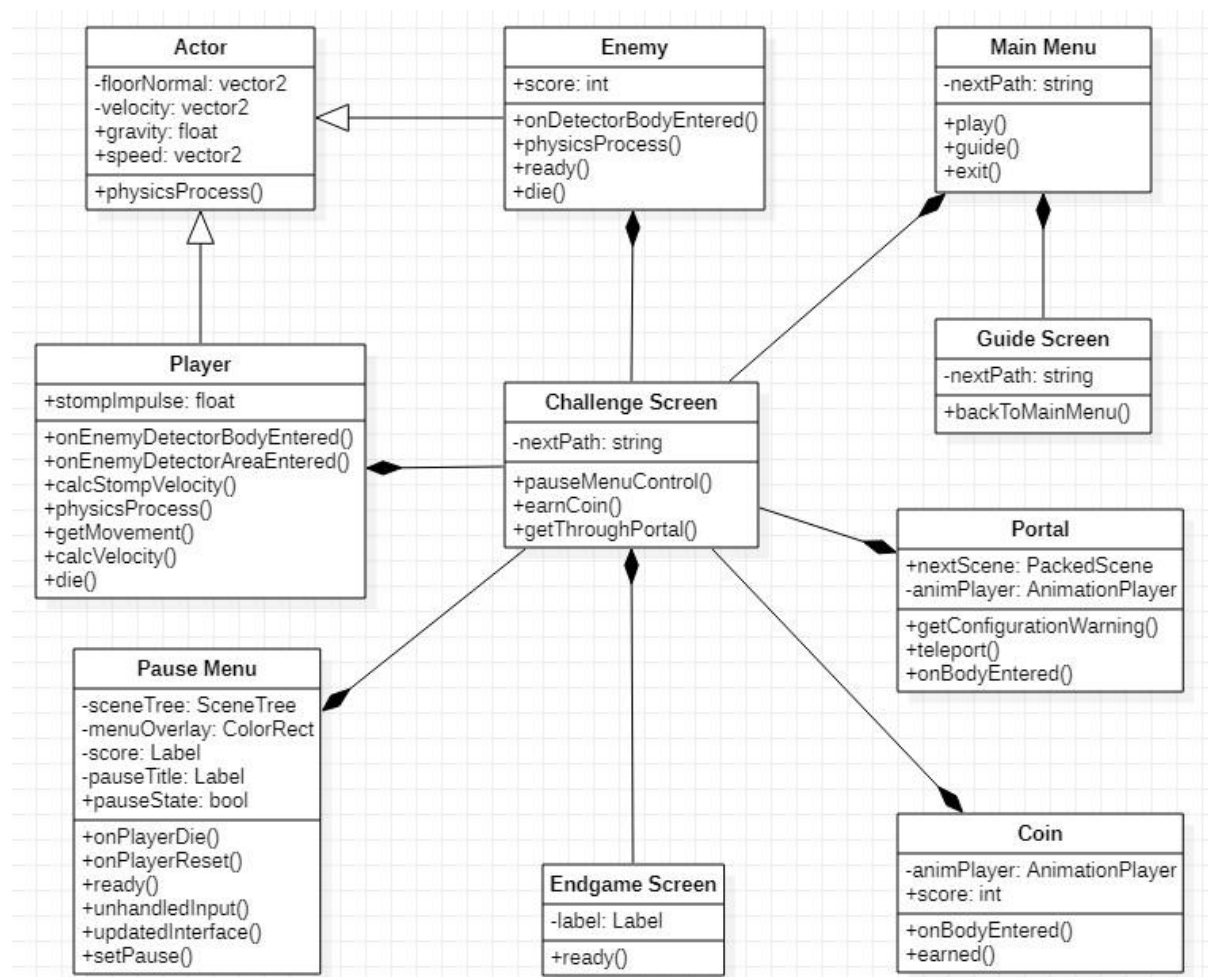


Hình 13. Sơ đồ hoạt động của usecase Xem điểm số



Hình 14. Sơ đồ hoạt động của usecase Tính điểm số

### 3.4 Sơ đồ lớp



Hình 15. Sơ đồ lớp

## CHƯƠNG IV: HIỆN THỰC

### 4.1 Hiện thực bằng code

#### 4.1.1 Lớp tác nhân

```
extends KinematicBody2D  
class_name Actor  
const floor_normal: = Vector2.UP  
var _velocity: = Vector2.ZERO  
export var gravity: = 3000.0  
export var speed: = Vector2(400.0, 500.0)  
func _physics_process(delta: float) -> void:  
    _velocity.y += gravity * delta
```

Lớp tác nhân được thiết kế nhằm áp dụng các chuyển động vật lý như tốc độ di chuyển, trọng lực và xác định mặt đất cho người chơi và kẻ thù trong game.

#### 4.1.2 Lớp người chơi

```
extends Actor  
export var stomp_impulse: = 600.0  
func _on_EnergyDetector_area_entered(area: Area2D) -> void:  
    _velocity = calc_stomp_velocity(_velocity, stomp_impulse)  
func _on_EnergyDetector_body_entered(body: PhysicsBody2D) -> void:  
    die()  
func get_movement() -> Vector2:  
    return Vector2(  
        Input.get_action_strength("move_right") -  
        Input.get_action_strength("move left"),  
        -Input.get_action_strength("jump") if  
        Input.is_action_just_pressed("jump") and is_on_floor() else 0.0  
    )
```

```

func calc_velocity(linear_velocity: Vector2, direction: Vector2, speed: Vector2,
is_jump_condition: bool) -> Vector2:
    var velocity: = linear_velocity
    velocity.x = speed.x * direction.x
    if direction.y != 0.0:
        velocity.y = speed.y * direction.y
    if is_jump_condition:
        velocity.y = 0.0
    return velocity

func calc_stomp_velocity(linear_velocity: Vector2, impulse: float) -> Vector2:
    var stomp_jump: = -speed.y if Input.is_action_pressed("jump") else -
stomp_impulse
    return Vector2(linear_velocity.x, stomp_jump)

func _physics_process(delta: float) -> void:
    var is_jump_condition: = Input.is_action_just_pressed("jump") and
_velocity.y < 0.0
    var direction: = get_movement()
    var snap: Vector2 = Vector2.DOWN * 60.0 if direction.y == 0.0 else
Vector2.ZERO
    _velocity = calc_velocity(_velocity, direction, speed, is_jump_condition)
    _velocity = move_and_slide_with_snap(_velocity, snap, floor_normal,
true)

func die() -> void:
    PlayerProcess.death_numbers += 1
    queue_free()

```

Lớp người chơi sẽ kế thừa các chức năng từ lớp tác nhân, bên cạnh đó lớp người chơi cũng sẽ có thêm các chức năng khác như tiêu diệt kẻ thù, điều khiển chuyển động và bật nhảy, người chơi sẽ chết nếu chạm vào phần thân của kẻ thù đồng thời game cũng sẽ đếm số lần chết của người chơi.



### 4.1.3 Lớp kẻ thù

```
extends Actor

export var score: = 100

func _ready() -> void:
    set_physics_process(false)
    _velocity.x = -speed.x

func _on_Detector_body_entered(body: PhysicsBody2D) -> void:
    if body.global_position.y > get_node("Detector").global_position.y:
        return

    die()

func _physics_process(delta: float) -> void:
    _velocity.x *= -1 if is_on_wall() else 1
    _velocity.y = move_and_slide(_velocity, floor_normal).y

func die() -> void:
    queue_free()
    PlayerProcess.score += score
```

Lớp kẻ thù sẽ kế thừa các chức năng từ lớp tác nhân, bên cạnh đó lớp kẻ thù cũng sẽ có thêm các chức năng khác như tiêu diệt người chơi, tự động chuyển động qua lại khi va chạm với chướng ngại vật, khi kẻ thù bị người chơi tiêu diệt thì đồng thời game cũng sẽ tính điểm tiêu diệt kẻ thù cho người chơi.

### 4.1.4 Lớp giao diện khi tạm dừng

```
extends Control

onready var scene_tree: = get_tree()
onready var menu_overlay: ColorRect = get_node("MenuOverlay")
onready var score: Label = get_node("Label")
onready var pause_title: Label = get_node("MenuOverlay/Title")
#onready var main_menu_buttons: Button =
    get_node("MenuOverlay/VBoxContainer/MainMenuButton")
var pause_state: = false setget set_pause

func _ready() -> void:
    PlayerProcess.connect("score_updated", self, "updated_interface")
```

```

        PlayerProcess.connect("player_died", self, "_on_player_died")
        PlayerProcess.connect("reset", self, "_on_player_reset")
        updated_interface()
func _on_player_died() -> void:
    self.pause_state = true
    pause_title.text = "You died!"
func _on_player_reset() -> void:
    self.pause_state = false
func _unhandled_input(event: InputEvent) -> void:
    if event.is_action_pressed("pause") and pause_title.text != "You died!":
        self.pause_state = not self.pause_state
func updated_interface() -> void:
    score.text = "Score: %s" % PlayerProcess.score
func set_pause(value: bool) -> void:
    pause_state = value
    scene_tree.paused = value
    menu_overlay.visible = value

```

Lớp giao diện tạm dừng được thiết kế để người chơi có thể tạm dừng game nếu muốn thông qua phím tắt Esc, đồng thời giao diện tạm dừng còn bao gồm các tính năng như chơi lại, quay về menu chính và thoát game. Bên cạnh đó, nếu người chơi chết thì giao diện tạm dừng cũng sẽ được hiển thị.

#### 4.1.5 Lớp cổng dịch chuyển

```

tool
extends Area2D
export var next_scene: PackedScene
onready var anim_player: AnimationPlayer = $AnimationPlayer
func _get_configuration_warning() -> String:
    return "The next scene can't be empty" if not next_scene else ""
func teleport() -> void:
    anim_player.play("fade_in")
    yield(anim_player, "animation_finished")

```

```

        get_tree().change_scene_to(next_scene)
func _on_body_entered(body: PhysicsBody2D) -> void:
    teleport()

```

Lớp cổng dịch chuyển được thiết kế cho người chơi mỗi khi muốn di chuyển qua các màn chơi mới, đồng thời tạo hiệu ứng chuyển cảnh.

#### 4.1.6 Lớp xu tính điểm

```

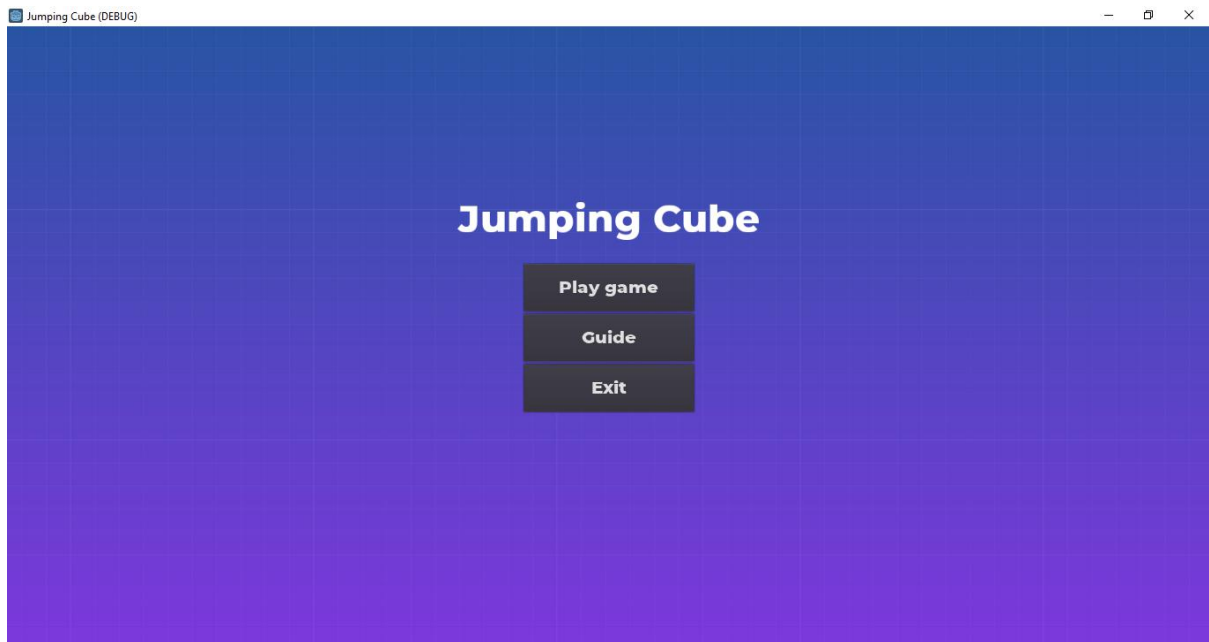
extends Area2D
onready var anim_player: AnimationPlayer = get_node("AnimationPlayer")
export var score: = 100
func _on_body_entered(body: PhysicsBody2D) -> void:
    earned()
func earned() -> void:
    PlayerProcess.score += score
    anim_player.play("fade_out")

```

Lớp xu tính điểm được thiết kế để khi người chơi va chạm cũng như ăn nó thì sẽ được cộng thêm điểm thưởng, đồng thời cũng kèm theo hiệu ứng biến mất đột ngột.

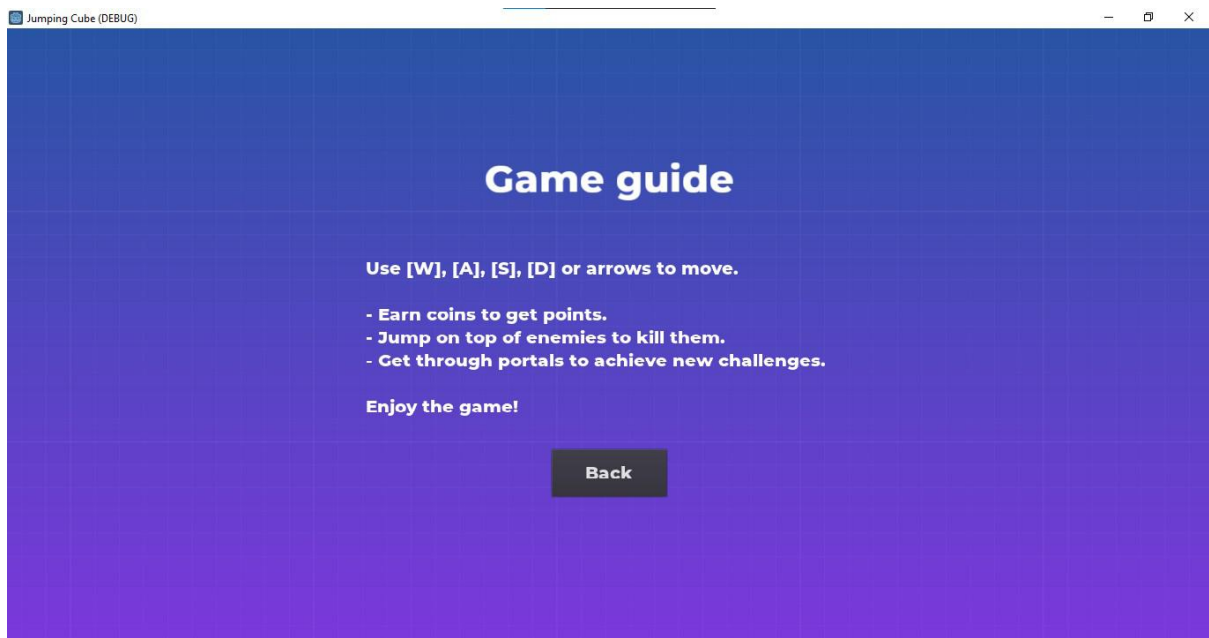
## 4.2 Hình ảnh trong game

### 4.2.1 Menu chính



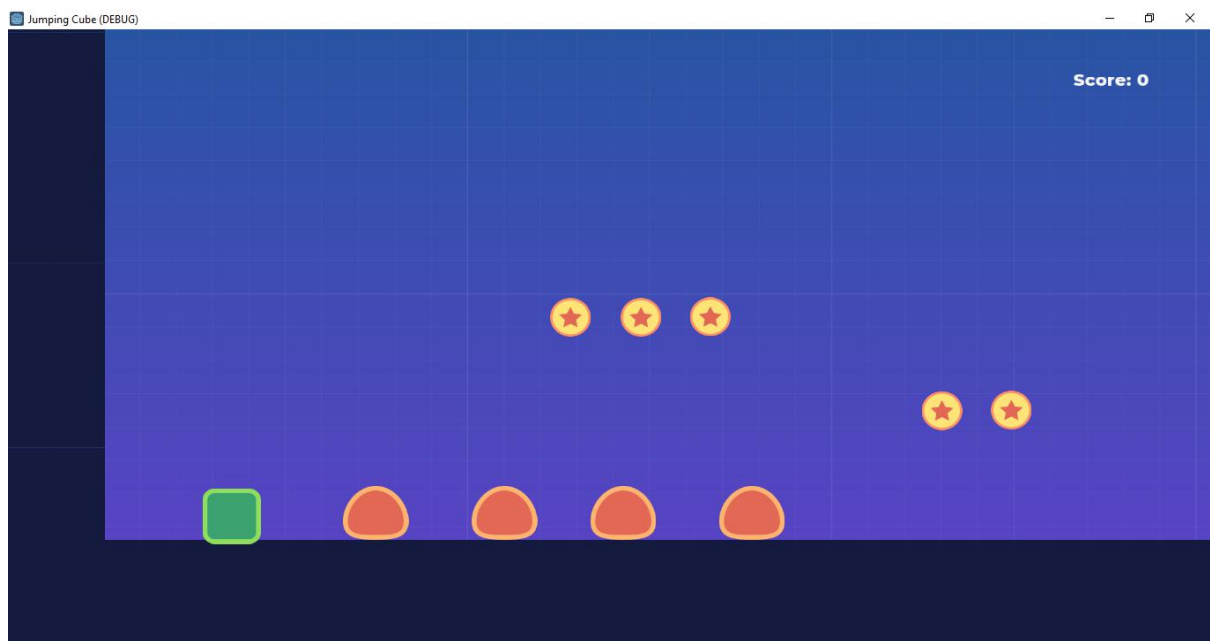
*Hình 16. Giao diện menu chính*

### 4.2.2 Hướng dẫn chơi



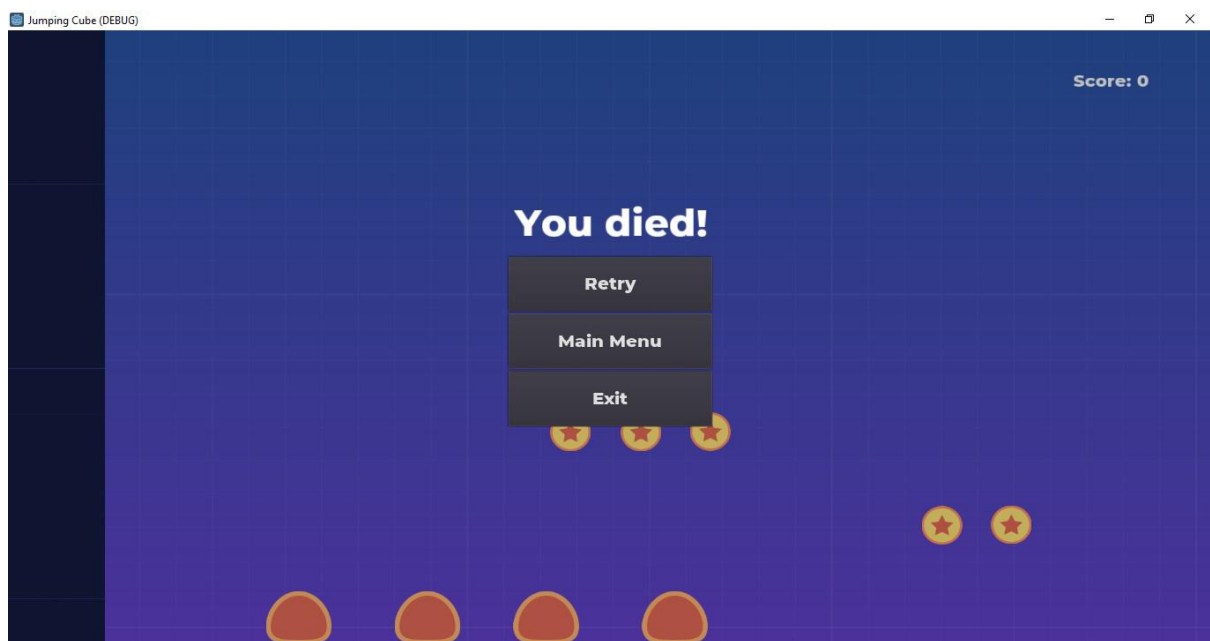
*Hình 17. Giao diện hướng dẫn*

### 4.2.3 Khung cảnh trong game



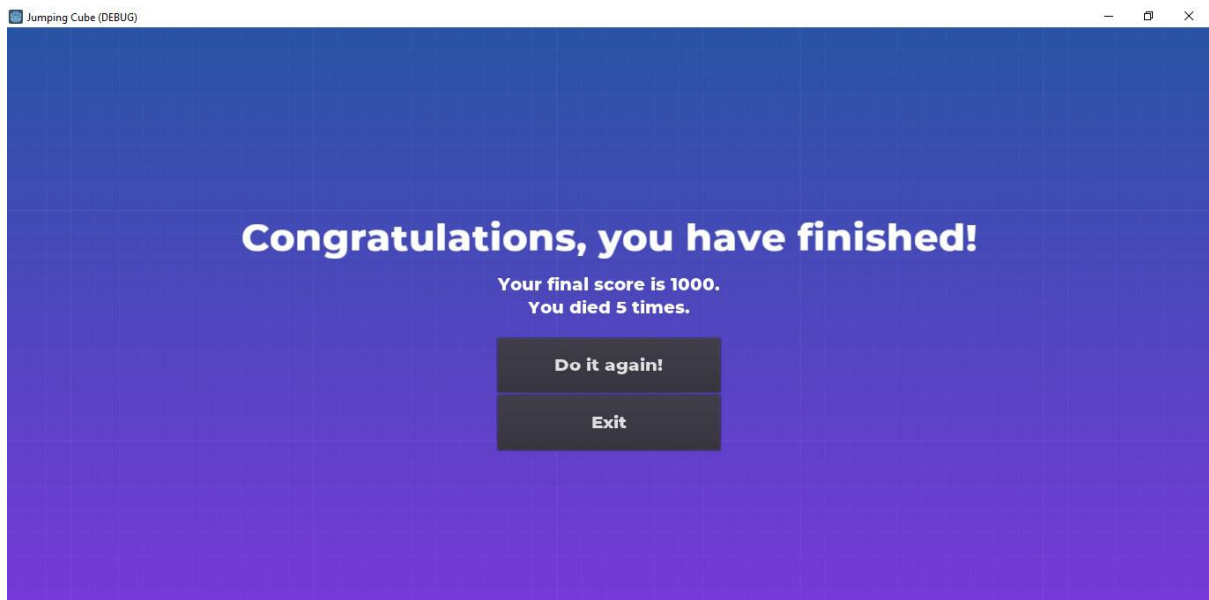
*Hình 18. Giao diện trò chơi*

### 4.2.4 Giao diện tạm dừng khi người chơi chết hoặc muốn tạm dừng game



*Hình 19. Giao diện menu thua cuộc*

## 4.2.5 Giao diện kết thúc game



*Hình 20. Giao diện kết thúc trò chơi*

## CHƯƠNG V: KẾT LUẬN

Bài báo cáo này đã giới thiệu tổng quát nhất về game 2D không cốt truyện - Jumping Cube. Thông qua các use case, đặc tả usecase cùng với các sơ đồ hoạt động, sơ đồ lớp đã minh họa cho việc hiện thực một ứng dụng game mini 2D một cách đơn giản nhất, dễ hiểu nhất. Đồng thời cũng giới thiệu qua về một game engine rất có ích trong việc phát triển game - Godot Engine, gọn nhẹ, dễ sử dụng, hỗ trợ nhiều thư viện có sẵn giúp tiết kiệm thời gian rất nhiều trong việc thiết kế và triển khai thực hiện một game mini 2D và build ra một ứng dụng thực tế. Cú pháp ngôn ngữ GDScript tương tự ngôn ngữ Python, cho phép tạo và tích hợp nội dung một cách linh hoạt, người phát triển game dễ dàng đọc hiểu và thao tác nhanh chóng.

Trong tương lai , nhóm sẽ phát triển một số game mini 2D mới lạ, đánh thẳng nhu cầu thị trường hiện nay, tìm hiểu chi tiết hơn về tâm lý người chơi để đưa ra những trải nghiệm tốt nhất cho người chơi.

## PHÂN CÔNG CÔNG VIỆC NHÓM

Họ và tên	Thời gian	Phân công nhiệm vụ
<b>Tô Vĩnh Khang</b>	5/10/2020-12/10/2020	Thảo luận nhóm và Phân công công việc.  Tìm hiểu về Godot Engine và ngôn ngữ GDScript.
	13/10/2020-30/10/2020	Trình bày báo cáo Chương I, Chương II, Chương III (3.1,3.2), Chương V.  Hiện thực bằng code GDScript.
<b>Bùi Quang Khải</b>	5/10/2020-12/10/2020	Thảo luận nhóm.  Tìm hiểu về Godot Engine và ngôn ngữ GDScript.
	13/10/2020-30/10/2020	Trình bày báo cáo Chương IV.  Hiện thực bằng code GDScript.
<b>Du Thuận Long</b>	5/10/2020-12/10/2020	Thảo luận nhóm.  Tìm hiểu về Godot Engine và ngôn ngữ GDScript.
	13/10/2020-30/10/2020	Trình bày báo cáo Chương III (3.3,3.4).  Hiện thực bằng code GDScript.

**Bảng phân công nhiệm vụ**



# TÀI LIỆU THAM KHẢO

## **Tài liệu Tiếng Anh:**

- [1] Vũ Đình Hồng, 2019, Slide Bài giảng môn Phát triển Trò chơi, Đại học Tôn Đức Thắng
- [2] Steve Rabin, 2010, Introduction to Game Development, 2<sup>nd</sup> Edition, Cengage Learning.
- [3] Jesse Schell, 2008, The Art of Game Design - A Book of Lenses, 1<sup>st</sup> Edition, Morgan Kaufmann.

## **Tài liệu Internet:**

- [4] <https://www.youtube.com/watch?reload=9&v=b6pbtmj1O4k>
- [5] <https://godotengine.org/features>
- [6] <https://gdscript.com/tutorials>