|  | **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG** |
| --- | --- |
|  | **KHOA CÔNG NGHỆ THÔNG TIN I**  **-------------------------------------** |
|  |
|  | **ĐỒ ÁN TỐT NGHIỆP** |
|  | **ĐỀ TÀI:**  **XÂY DỰNG HỆ THỐNG CHẤM CODE PHỤC VỤ CHO VIỆC HỌC VÀ LUYỆN TẬP LẬP TRÌNH** |
|  |  |
|  | **Người hướng dẫn : ThS. Đinh Xuân Trường** |
|  | **Họ và tên : Bùi Đức Khanh** |
|  | **Mã sinh viên : B18DCCN303** |
|  | **Lớp : D18CQPM03** |
|  | **HÀ NỘI – 2022** |



| **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG** | |
| --- | --- |
| **KHOA CÔNG NGHỆ THÔNG TIN I**  **---------------------------------------** | |
| **ĐỒ ÁN TỐT NGHIỆP** | |
| **ĐỀ TÀI:**  **XÂY DỰNG HỆ THỐNG CHẤM CODE PHỤC VỤ CHO VIỆC HỌC VÀ LUYỆN TẬP LẬP TRÌNH** | |
| **Người hướng dẫn :** | **ThS. Đinh Xuân Trường** |
| **Họ và tên :** | **Bùi Đức Khanh** |
| **Lớp :** | **D18CQPM03** |
| **Mã sinh viên :** | **B18DCCN303** |

## NHẬN XÉT

**(Của giảng viên hướng dẫn)**

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………...

……………………………………………………………………………………...

……………………………………………………………………………………...

Hà Nội, ngày … tháng … năm 20… Giảng viên hướng dẫn

## NHẬN XÉT

**(Của giảng viên hướng dẫn)**

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………...

……………………………………………………………………………………...

……………………………………………………………………………………...

Hà Nội, ngày … tháng … năm 20… Giảng viên hướng dẫn

## LỜI CẢM ƠN

Em xin cảm ơn các thầy, cô giáo của Học Viện Công nghệ Bưu chính Viễn thông, đặc biệt là các thầy cô trong khoa Công nghệ thông tin 1 đã dạy dỗ và chỉ bảo em trong suốt thời gian học tập tại trường.

Em xin chân thành cảm ơn ThS. Đinh Xuân Trường đã quan tâm, hướng dẫn và tận tình chỉ dạy em trong suốt quá trình thực hiện và viết báo cáo đồ án này.

Mặc dù đã cố gắng hoàn thành báo cáo trong phạm vi cho phép, nhưng bài báo cáo chắc chắn sẽ không tránh được những sự thiếu sót. Em kính mong nhận được sự thông cảm của thầy cô và các bạn.

Em xin chân thành cảm ơn!

Hà Nội, 11/2022 Sinh viên thực hiện **Bùi Đức Khanh**

## MỤC LỤC

## DANH MỤC HÌNH ẢNH

## DANH MỤC BẢNG BIỂU

## DANH MỤC CHỮ VIẾT TẮT

## LỜI NÓI ĐẦU

Hiện nay, với sự phát triển mạnh mẽ của ngành công nghệ thông tin và những ứng dụng của nó ở khắp các lĩnh vực trong đời sống thì nhu cầu về nhân lực trong ngành luôn ở mức cao. Các công việc liên quan đến vị trí lập trình cũng đang được rất nhiều công ty, tổ chức săn đón với mức lương vô cùng hấp dẫn.

Hơn nữa, công nghệ thông tin đang trở thành ngành “hot” trong những năm gần đây. Số lượng các bạn sinh viên yêu thích và theo học ngành này cũng tăng vọt. Khi bước vào cánh cửa của ngành công nghệ thông tin, bên cạnh những kiến thức đại cương nền tảng, việc học lập trình là bắt buộc với những sinh viên theo học ngành này.

Do vậy, nhu cầu học tập và luyện tập lập trình của sinh viên cũng như những người yêu thích ngành công nghệ thông tin cũng tăng cao. Để nâng cao chất lượng của những chương trình máy tính, cũng như nâng cao chất lượng của những lập trình viên tương lai sau này, việc xây dựng và phát triển một hệ thống giúp luyện tập lập trình là rất cần thiết. Đó chính là lý do em mong muốn xây dựng đồ án này với đề tài: **“Xây dựng hệ thống chấm code phục vụ cho việc học và luyện tập lập trình”**.

Với mục tiêu như vậy, em xin trình bày đồ án với 4 chương như sau:

**Chương I Giới thiệu chung:** Trong chương này, đồ án sẽ mô tả đề tài, giới thiệu mục tiêu của đề tài, giới thiệu …

**Chương II Phân tích và thiết kế ứng dụng:** Trong chương này, đồ án sẽ giới thiệu cấu trúc, giao diện,..

**Chương III Cài đặt ứng dụng:** Trình bày các … những yêu cầu cho phần cài đặt phần mềm.

**Chương IV Kết luận và hướng phát triển:** Tổng kết, tóm tắt những kết quả đã thực hiện được để và đề xuất hướng phát triển

# CHƯƠNG I GIỚI THIỆU CHUNG

Trong chương này, đồ án sẽ mô tả và giới thiệu mục tiêu của để tài cùng với đó là giới thiệu các công cụ và công nghệ được sử dụng trong ứng dụng.

Nội dung chính bao gồm:

* + - Giới thiệu đề tài
    - Các công nghệ được sử dụng
    - Các công cụ
    - Tổng kết

## Giới thiệu đề tài

## Tên đề tài

* Tên đề tài: **Xây dựng hệ thống chấm code phục vụ cho việc học và luyện tập lập trình.**

## Mục tiêu đề tài

* + - * Mục tiêu đề tài: Xây dựng một môi trường luyện tập dành cho những người yêu thích công việc lập trình. Cung cấp hệ thống hỗ trợ biên dịch, kiểm tra và chấm code tự động, giúp những người yêu thích lập trình rèn luyện tư duy lập trình.
      * Đối tượng hướng đến: các bạn học sinh, sinh viên và những người yêu thích lập trình.

## Mục tiêu nghiên cứu

* + - * Tìm hiểu về kiến trúc REST và tiêu chuẩn RESTful API trong việc thiết kế ứng dụng web.
      * Tìm hiểu chuyên sâu về ngôn ngữ lập trình Javascript thông qua việc xây dựng ứng dụng web sử dụng Node.Js kết hợp với ReactJs.
      * Tìm hiểu chuyên sâu về cơ sở dữ liệu không quan hệ Mongodb.

## Các công nghệ được sử dụng

## Tổng quan về ngôn ngữ javascript

* Định nghĩa:

Javascript là một ngôn ngữ lập trình an toàn. Ban đầu javascript được tạo ra để giúp cho các trang web trở nên sống động hơn và được sử dụng chủ yếu phía trình duyệt. Các tập lệnh javascript được cung cấp và thực thi dưới dạng văn bản thuần túy, không cần chuẩn bị hay biên dịch để chạy. Hiện nay, javascript không chỉ được thực thi ở phía trình duyệt mà còn trên phía máy chủ hoặc trên thực tế là trên bất cứ thiết bị nào có chương trình đặc biệt được gọi là Javascript Engine. Javascript đang ngày càng phát triển và trở thành một trong những ngôn ngữ được sử dụng nhiều nhất hiện nay.

* Quá trình phát triển:

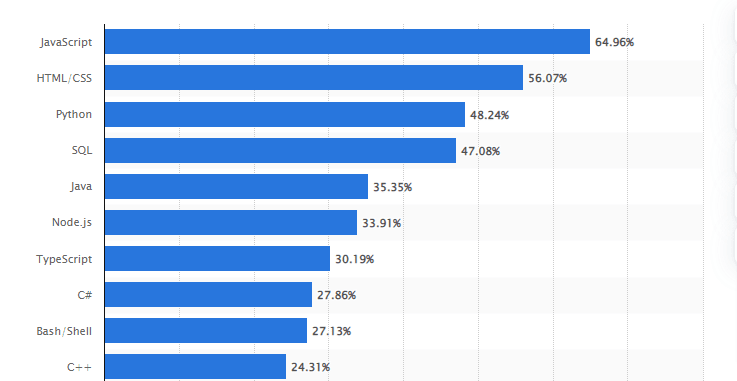
Javascript được phát minh bởi Brendan Eich vào năm 1995.

Ban đầu nó được đặt tên là Mocha, sau đó đổi thành LiveScript và cuối cùng là Javascript

Năm 1996, Netscape 2 được phát hành với javascript 1.0 ( Netscape là tên của trình duyệt web phổ biến nhất thế giới giai đoạn năm 1994, tiền thân của Mozilla FireFox) và trở thành tiêu chuẩn ECMA-262 vào năm 1997.

Trải qua nhiều giai đoạn, Javascript ngày càng phát triển mạnh mẽ. Cho đến năm 2018, Javascript phiên bản ES6 được hỗ trợ trên tất cả các trình duyệt thông dụng: Chrome, FireFox, Safari.

Hiện nay, Javascript nhiều lần được vinh danh là ngôn ngữ lập trình được sử dụng rộng rãi nhất thế giới với 64,9% nhà phát triển sử dụng nó trong năm 2021.



*Hình 1.1 Thống kê nhà phát triển sử dụng javascript*

Một loạt các framework và thư viện JavaScript như Ember, Angular, React và Vue cũng đã được tạo ra để phát triển các ứng dụng web phức tạp và mạnh mẽ. Ngoài ra, bên cạnh phần mềm máy khách và máy chủ, giờ đây thậm chí có thể viết các ứng dụng di động gốc bằng JavaScript.

* Ưu điểm:
* Javascript là ngôn ngữ lập trình có cấu trúc rất đơn giản, mã javascript dễ hiểu và dễ tiếp thu.
* Javascript là ngôn ngữ thông dịch, không cần biên dịch để thực thi nên tốc độ thực thi chương trình nhanh.
* Bất kể JavaScript được lưu trữ ở đâu, nó luôn chạy trong môi trường máy khách để giảm mức sử dụng băng thông và tăng tốc độ thực thi.
* Javascript hoạt động được trên nhiều trình duyệt, nền tảng
* Dễ dàng trong việc giải quyết các vấn đề thắc mắc bởi Javascript được sử dụng bởi rất nhiều nhà phát triển.
* Nhược điểm:
* Khó khăn trong việc gỡ lỗi.
* Người dùng có thể xem được mã javascript phía máy khách, nó có thể bị người khác lạm dụng. Ngoài ra không khó để chèn mã vào trang web làm suy giảm tính bảo mật.
* Kiểu dữ liệu trong javascript ở dạng dynamic type nên khả năng gặp những lỗi tiềm ẩn là rất cao.

## Tổng quan về Node.js

* + - * Định nghĩa:

Node.js là một Javascript runtime mã nguồn mở được xây dựng dựa trên V8 Javascript engine của Chrome ( engine này chịu trách nhiệm cho việc biên dịch javascript sang mã máy tính để máy tính có thể hiểu và chạy được mã javascript ).

Node.js ra mắt vào năm 2009, bởi Ryan Dahl và nhận sự bảo trợ từ công ty Joyent (Hoa Kỳ).

* + - * Đặc điểm của Node.js

**Bất đồng bộ và phát sinh sự kiện** (Non-blocking and Event Driven): Tất các các APIs của thư viện Node.js đều bất đồng bộ (non-blocking), NodeJS không cần đợi một API trả về dữ liệu. Server chuyển sang một API khác sau khi gọi nó và có cơ chế riêng để gửi thông báo và nhận phản hồi về các hoạt động của Node.js và API đã gọi.

**Tốc độ nhanh**: Phần core phía dưới được viết gần như toàn bộ bằng C++ kết hợp Chrome V8 Engine nên tốc độ xử lý công việc của Node.js cực nhanh, nhưng vẫn đảm bảo được tính chuẩn xác.

**Đơn giản – Hiệu năng cao**: Node.js sử dụng một mô hình luồng đơn luồng (single thread) và các sự kiện lặp (event-loop). Cơ chế sự kiện cho phép phía Server trả về phản hồi theo non-blocking, đồng thời tăng hiệu quả sử dụng. Các luồng đơn cung cấp dịch vụ cho nhiều request hơn hẳn Server truyền thống.

* + - * Ưu điểm:

Phần core bên dưới của Node.js được viết chủ yếu bằng C++ nên hiệu năng và tốc độ xử lý cao.

Node.js thích hợp cho việc tạo ra ứng dụng có tốc độ xử lý nhanh với thời gian thực (real-time).

Thích hợp để tạo các sản phẩm có lượng truy cập lớn, mở rộng nhanh và đổi mới công nghệ… hoặc tạo dự án startup tiết kiệm thời gian nhất.

Nodejs sử dụng javascript, là ngôn ngữ lập trình phổ biến nhất trong thời điểm những năm gần đây nên cộng đồng hỗ trợ lớn. Dễ dàng giải đáp những vướng mắc.

* Nhược điểm:

Không thích hợp cho những ứng dụng gây tốn tài nguyên và CPU

Nền tảng Node.js không có vùng nhớ đệm, tức không cung cấp khả năng lưu trữ dữ liệu buffer.

## 1.2.3 Tổng quan về ReactJs

## 1.2.4 Tổng quan về REST và RESTful API

## 1.2.5 Tổng quan về NoSQL và MongoDB

## Các công cụ

## Tổng kết

Trong chương này đồ án đã trình bày được các thông tin chi tiết về đề tài **Xây dựng hệ thống chấm code phục vụ cho việc học và luyện tập lập trình** và thông tin sơ bộ về các công cụ và công nghệ được sử dụng trong ứng dụng.

Dựa vào bài toán đã được đặt ra ở chương này, chương tiếp theo đồ án sẽ thực hiện phân tích và thiết kế ứng dụng.

# CHƯƠNG II PHÂN TÍCH VÀ THIẾT KẾ ỨNG DỤNG

Trong chương này, đồ án sẽ giới thiệu sâu hơn về ứng dụng. Qua đó sẽ đi đến các bước phân tích và thiết kế để tạo nên một ứng dụng hoàn chỉnh. Nội dung chương bao gồm:

* Giới thiệu về ứng dụng
* Unit 1: Save Earth
* Unit 2: Game Christmas Eggs
* Unit 3: Math Game
* Unit 4: Game Dungeon
* Chức năng cài đặt âm thanh
* Tổng kết

## Giới thiệu về ứng dụng

* + - Tên ứng dụng: **DrimLearn.**
    - Đối tượng hướng đến: các bạn nhỏ với trình độ lớp 3. Các câu hỏi có nhiều dạng như: câu hỏi trắc nghiệm trọn một trong bốn đáp án hoặc trả lời tự luận để người học có thể củng cố kiến thức về môn toán.
    - Người học sẽ lần lượt trải qua các bài học với những chủ đề khác nhau, các bài học kết hợp với dạng game một chạm. Mỗi khi vượt qua được một câu hỏi người học sẽ nhận được nhận phần thưởng tùy theo chủ đề từng bài học và đến với câu hỏi tiếp theo.
    - Ứng dụng được chia làm các Unit nhỏ, mỗi Unit sẽ có 1 game và 1 nội dung bài học khác nhau. Cụ thể như sau:

- Unit 1: Game Save Earth

+ Chủ đề: bảo vệ môi trường, bảo vệ trái đất.

+ Câu hỏi: cộng trừ số có 3 chữ số (không nhớ).

+ Game: với mỗi câu trả lời đúng, rác sẽ được đưa vào thùng rác và trồng thêm được một cây xanh.

- Unit 2: Game Christmas Eggs

+ Chủ đề: yêu cầu nhặt trứng từ ông già nô-en.

+ Câu hỏi: Cộng trừ số có 3 chữ số (có nhớ một lần).

+ Game: với mỗi câu trả lời đúng người học sẽ nhận được một quả trứng, nhận được càng nhiều trứng thì sẽ càng mở ra rồng cấp cao.

- Unit 3: Math Game

+ Chủ đề: luyện tập tính nhanh cùng bạn sóc.

+ Câu hỏi: ngẫu nhiên về các chủ đề phép cộng, trừ, nhân, chia.

+ Game: sau khi trả lời đúng câu hỏi trong thời gian quy định thì sẽ đến câu hỏi kế tiếp.

- Unit 4: Game Dungeon

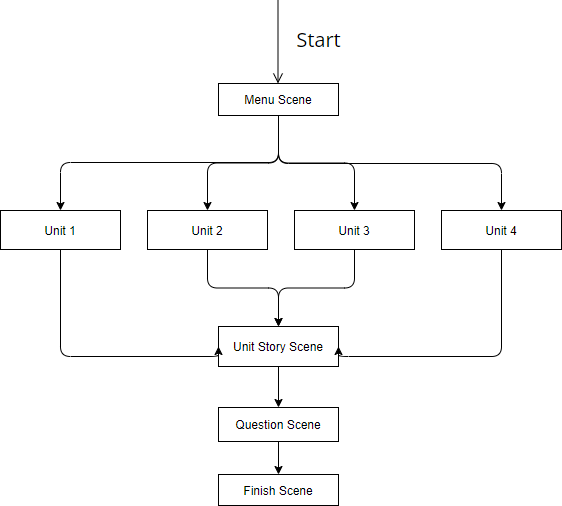
+ Chủ đề: cứu người khỏi mê cung.

+ Câu hỏi: chủ đề về hình học.

+ Game: cứu người lạc trong mê cung bằng cách vượt qua những câu hỏi trên đường thoát hiểm.

## Sơ đồ hoạt động chung của ứng dụng

Ứng dụng hoạt động theo một luồng chung như sau:



*Hình 2. 1. Sơ đồ hoạt động của ứng dụng*

## Xác định tác nhân và xây dựng use case tổng quan

### *Xác định các actor của hệ thống*

Ứng dụng chỉ có duy nhất một actor là người sử dụng(user). User có thể thực hiện các chức năng: tham gia các bài học(unit 1, unit 2, unit 3, unit 4), cài đặt âm thanh, thoát.

### *Xây dựng use case hệ thống*



*Hình 2. 2. Biểu đồ use case tổng quan của hệ thống*

## Unit 1: Game Save Earth

## Thể loại

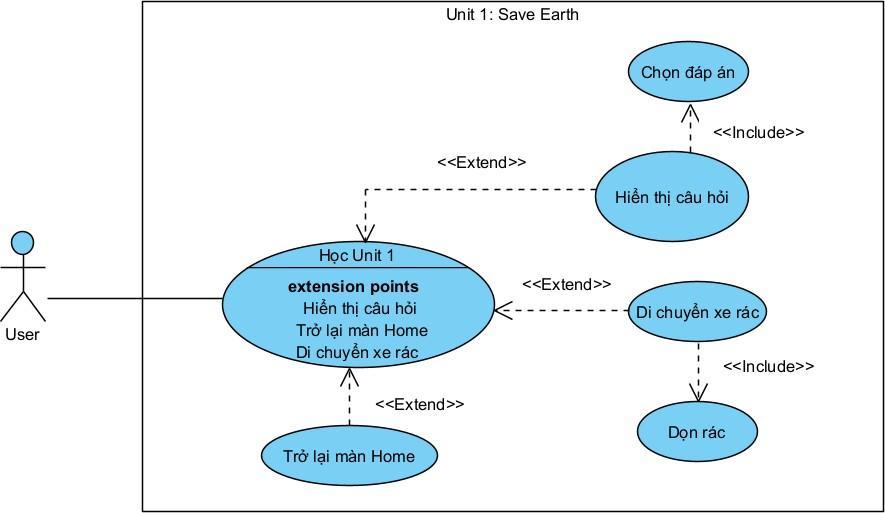
Game trắc nghiệm.

## Kịch bản

Trong sự phát triển của nhân loại, trái đất càng ngày càng ô nhiễm. Lắng nghe lời cầu cứu của trái đất, người học sẽ dọn sạch rác thải và trồng cây xanh bằng cách trả lời đúng những câu hỏi trắc nghiệm được đưa ra. Với mỗi câu trả lời đúng, rác thải sẽ được dọn và trái đất sẽ có thêm một cây xanh.

## Thiết kế chi tiết[1]

### *Use case chi tiết*



*Hình 2. 3. Use case chi tiết Unit 1 Save Earth*

Mô tả use case:

* + - * Trở lại màn hình Home: UC này cho phép user kết thúc bài học và trở về lại màn hình chính.
      * Chọn đáp án: UC này cho phép user lựa chọn đáp án.
      * Hiển thị câu hỏi: UC này cho phép user xem nội dung câu hỏi.
      * Di chuyển xe rác: UC cho phép di chuyển xe rác đến vị trí để dọn rác.
      * Dọn rác: UC cho phép dọn rác thải.

### *Kịch bản*

| **Tên use case** | Học Unit 1. |
| --- | --- |
| **Tác nhân** | User |
| **Mục đích** | Vào bài học Unit 1. |
| **Tiền điều kiện** | User đang ở màn hình chính. |
| **Mô tả luồng sự kiện chính** | 1. User nhấn chọn nút Unit 1. 2. Hệ thống hiển thị chủ đề của bài học và nút bắt đầu. |

|  | 1. User click chọn nút bắt đầu. 2. Hệ thống hiển thị: có 5 câu hỏi, 4 bốn đáp án và phần game gồm xe rác và các loại rác thải. 3. User click chọn một đáp án. 4. Hệ thống hiển thị đáp án đúng và nút câu hỏi kế tiếp.    1. User trả lời đúng: hệ thống sẽ di chuyển xe rác tới vị trí rác thải để dọn.    2. User trả lời sai: hệ thống bôi đỏ đáp án sai và bôi xanh đáp án đúng. 5. User click chọn nút câu hỏi kế tiếp. 6. Hệ thống quay lại bước 4. 7. User trả lời hết câu hỏi. 8. Hệ thống hiển thị kết quả gồm: số điểm, thời gian và nút quay lại màn hình chính. 9. User click chọn nút quay về màn hình chính. |
| --- | --- |
| **Hậu điều kiện** | Học xong và hoàn thành hết câu hỏi ở Unit 1. |
| **Ngoại lệ** | Không. |

1. ***Xây dựng các lớp thực thể***

*Bảng 2. 1. Bảng mô tả lớp QuestInfor.cs*

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | id | int | Mã câu hỏi. |
| 2 | result | int | Đáp án đúng. |
| 3 | quest | string | Nội dung câu hỏi. |
| 4 | A | string | Đáp án A. |
| 5 | B | string | Đáp án B. |
| 6 | C | string | Đáp án C. |

| 7 | D | string | Đáp án D. |
| --- | --- | --- | --- |

*Bảng 2. 2. Bảng mô tả lớp UI\_QuestMng.cs*

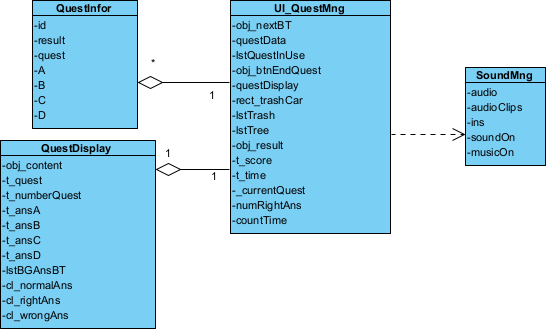
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | obj\_nextBT | GameObject | Nút chuyển câu hỏi. |
| 2 | questData | Quest | Dữ liệu câu hỏi. |
| 3 | lstQuestInUse | List<QuestInfor> | Danh sách những câu hỏi được sử dụng. |
| 4 | obj\_btnEndQuest | GameObject | Nút kết thúc bài học. |
| 5 | questDisplay | QuestDisplay | Lớp thực hiện hiển thị câu hỏi. |
| 6 | rect\_trashCar | RectTransform | Vị trí của xe trở rác. |
| 7 | lstTrash | List<RectTransform> | Vị trí của các rác thải. |
| 8 | lstTree | List<RectTransform> | Vị trí của các cây xanh. |
| 9 | obj\_result | GameObject | Kết quả của bài học. |
| 10 | t\_score | Text | Hiển thị số điểm. |
| 11 | t\_time | Text | Hiển thị số thời gian của người chơi. |
| 12 | \_currentQuest | int | Số câu hỏi hiện tại. |
| 13 | numRightAns | int | Số lượng câu hỏi đúng. |
| 14 | countTime | float | Biến đếm thời gian. |

*Bảng 2. 3. Bảng mô tả lớp QuestDisplay.cs*

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | obj\_Content | GameObject | Giao diện câu hỏi. |
| 2 | t\_quest | Text | Hiển thị câu hỏi. |
| 3 | t\_numberQuest | TextMeshProUGUI | Hiển thị câu hỏi hiện tại và tổng số câu hỏi. |
| 4 | t\_ansA | Text | Hiển thị đáp án A. |
| 5 | t\_ansB | Text | Hiển thị đáp án B. |
| 6 | t\_ansC | Text | Hiển thị đáp án C. |
| 7 | t\_ansD | Text | Hiển thị đáp án D. |
| 8 | lstBGAnsBT | List<Image> | Danh sách hình ảnh các nút đáp án. |
| 9 | cl\_normalAns | Color32 | Màu nút đáp án ở trạng thái bình thường. |
| 10 | cl\_rightAns | Color32 | Màu nút đáp án khi đúng. |
| 11 | cl\_wrongAns | Color32 | Màu nút đáp án khi sai. |

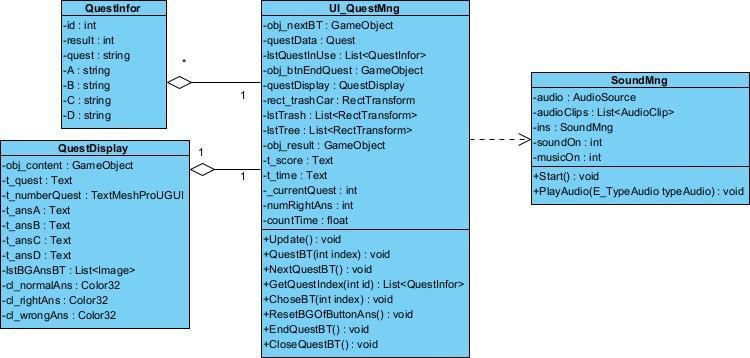
### *Biều đồ lớp thực thể pha phân tích*

* + - * Mỗi một UI\_QuestMng quản lý nhiều QuestInfor và một QuestInfor được quản lý bởi một UI\_QuestMng nên quan hệ giữa UI\_QuesrMng và QuestInfo là 1-n.
      * Mỗi một UI\_QuestMng quản lý một QuestDisplay và một QuestDisplay được quản lý bởi một UI\_QuestMng nên quan hệ giữa UI\_QuestMng và QuestInfo là 1-1.



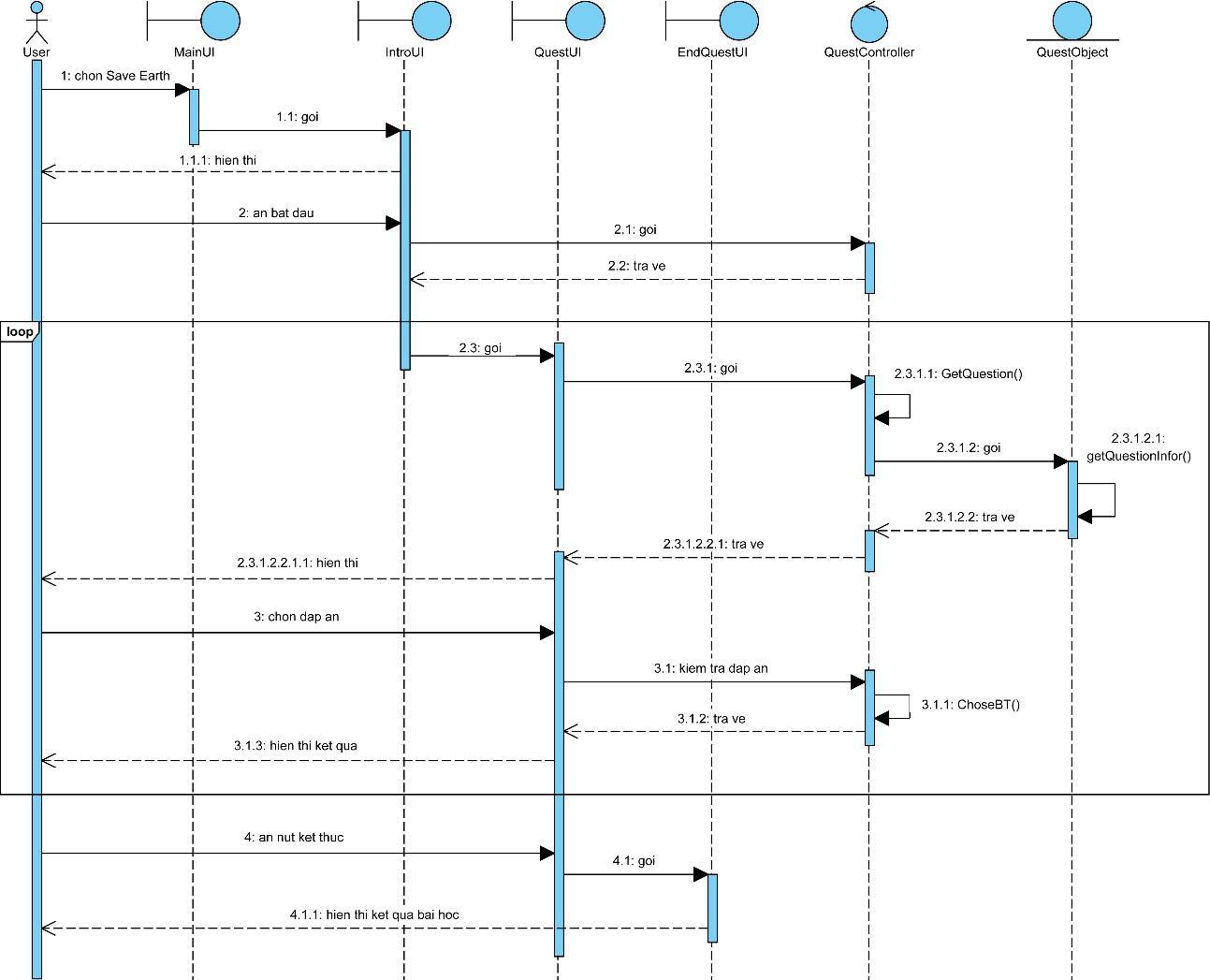
*Hình 2. 4. Biểu đồ lớp thực thể Unit 1*

### *Biểu đồ lớp thực thể pha thiết kế*



*Hình 2. 5. Biểu đồ lớp thiết kế Unit 1*

### *Biểu đồ tuần tự*



*Hình 2. 6. Biểu đồ tuần tự Unit 1*

### *Giao diện chơi*

* + - * Phần mở đầu giới thiệu về câu chuyện và chủ đề bài học.



*Hình 2. 7. Phần mở đầu game Save Earth*

* + - * Giao diện màn học chính.



*Hình 2. 8. Giao diện chính game Save Earth*

Trong giao diện bao gồm các rác thải, khi người học trả lời đúng thì rác sẽ được đưa vào trong xe rác và đồng thời sẽ có cây xanh mọc lên.

## Các thành phần chính

### *Lớp QuestInfor.cs*

Class này đóng vai trò là model chứa thông tin về câu hỏi: id, đáp án(result), câu hỏi(quest), các sự lựa chọn: A, B, C, D.

### *Lớp QuestDisplay.cs*

Lớp này thực hiện hiển thị thông tin dữ liệu câu hỏi ra màn hình và hiển thị đáp án đúng, sai.



### *Lớp UI\_QuestMng.cs*

* + - * Phương thức: NextQuestBT()

Phương thức này sẽ thực hiện lấy dữ liệu của câu hỏi tiếp theo và thực hiện khởi tạo lại hiển thị của các đáp án và câu hỏi.

* + - * Phương thức: ChoseBT(int index)

Phương thức này thực hiện xác nhận người chơi đã nhấn vào đáp án nào và thực hiện so sánh đáp án người chơi chọn với đáp án đúng. Nếu người chơi chọn đúng, nút đáp án sẽ chuyển sang màu xanh, rác sẽ được di chuyển vào thùng và đồng thời sẽ mọc thêm một cây xanh trên trái đất. Ngược lại nếu chọn sai, đáp án đúng sẽ hiển thị màu xanh và đáp án người chơi chọn sẽ được hiển thị thành màu đỏ.



* + - * Phương thức: EndQuestBT()

Thực hiện tổng kết lại điểm số và thời gian của người chơi và hiển thị ra màn hình.



## Unit 2: Game Christmas Eggs

## Thể loại

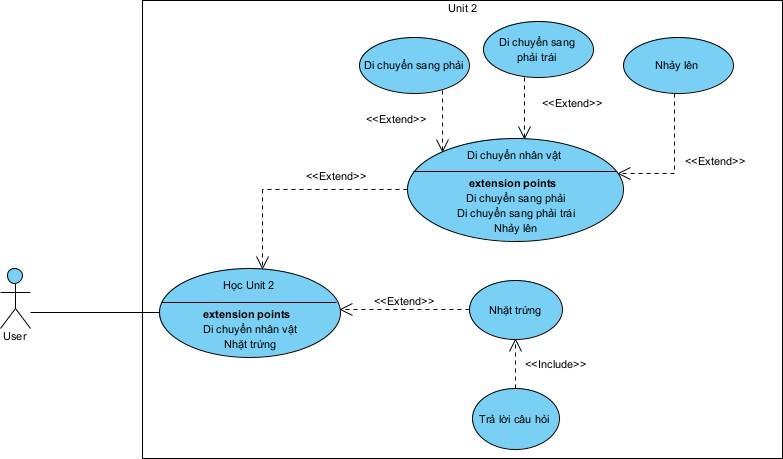
Game giải đố, đi cảnh.

## Kịch bản

* + - * Thực hiện theo yêu cầu của ông già noel, người chơi sẽ thực hiện việc thu hoạch trứng rồng để mở ra rồng theo cấp độ khác nhau.
      * Người chơi thực hiện điều khiển nhân vật vượt qua các chướng ngại vật. người chơi sẽ có 3 trái tim, với mỗi lần trả lời sai câu hỏi hoặc đâm dính chướng ngại vật thì sẽ mất một trái tim.
      * Để thực hiện thu hoạch được trứng rồng người chơi sẽ phải điều khiển nhân vật tới các quả trứng sau đó trả lời câu hỏi. nếu trả lời đúng thì sẽ được nhận một quả trứng rồng. Sau khi đã đến cuối hành trình, người chơi sẽ tới mở rương báu vật và kết thúc hành trình.

## Thiết kế chi tiết[1]

### *Use case chi tiết*



*Hình 2. 9. Use case chi tiết Unit 2 Christmas Eggs*

Mô tả use case:

* + - * + Di chuyển nhân vật: UC cho phép user điều khiển nhân vật.
        + Nhặt trứng: UC cho phép user thu thập trứng.
        + Trả lời câu hỏi: UC cho phép user trả lời câu hỏi.

### *Kịch bản*

| **Tên use case** | Học Unit 2. |
| --- | --- |
| **Tác nhân** | User |
| **Mục đích** | Vào bài học Unit 2. |
| **Tiền điều kiện** | User đang ở màn hình chính. |
| **Mô tả luồng sự kiện chính** | 1. User nhấn chọn nút Unit 2. 2. Hệ thống hiển thị chủ đề của bài học và nút bắt đầu. 3. User click chọn nút bắt đầu. 4. Hệ thống hiển thị: nhân vật, map chơi, trứng rồng,... 5. User di chuyển nhân vật chạm vào trứng trên đường đi. 6. Hệ thống hiển thị câu hỏi và bốn đáp án. 7. User click chọn đáp án. 8. Hệ thống hiển thị đáp án đúng.    1. User trả lời đúng: hệ thống sẽ tăng thêm số lượng trứng rồng lên một.    2. User trả lời sai: hệ thống bôi đỏ đáp án sai, bôi xanh đáp án đúng và trừ một trái tim của nhân vật. 9. User quay lại bước 5 cho đến khi user mở được hòm kho báu ở cuối map. 10. Hệ thống hiển thị kết quả gồm: số trứng đã nhặt được và nút mở trứng. 11. User click chọn nút mở trứng. 12. Hệ thống hiển thị rồng thu được tương ứng với số trứng và nút đóng. |

|  | 1. User click chọn nút đóng. 2. Hệ thống hiển thị màn hình chính. |
| --- | --- |
| **Hậu điều kiện** | Học xong Unit 2 |
| **Ngoại lệ** | * 1. User chạm vào gai thì mất một trái tim.   2. User mất hết trái tim thì kết thúc hành trình.   9.1 User không nhặt được quả trứng nào thì sẽ không mở được rồng. |

1. ***Xây dựng các lớp thực thể***

*Bảng 2. 4. Bảng mô tả lớp CharacterMng.cs*

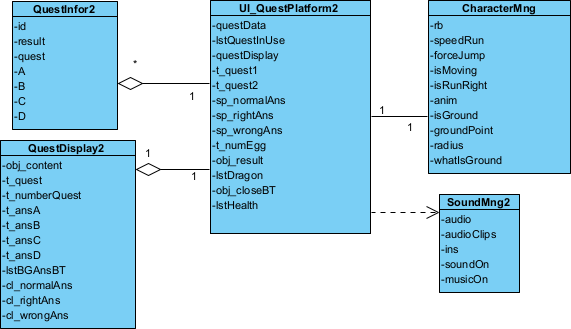
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | rb | Rigidbody2D | Điều khiển vị trí nhân vật thông qua hệ thống mô phỏng vật lý. |
| 2 | speedRun | float | Tốc độ di chuyển. |
| 3 | forceJump | float | Lực nhảy. |
| 4 | isMoving | bool | Kiểm tra nhân vật có đang di chuyển không. |
| 5 | isRunRight | bool | Kiểm tra nhân vật đi sang phải. |
| 6 | anim | Animator | Hoạt ảnh hiển thị hành động. |
| 7 | isGround | bool | Kiểm tra nhân vật có đang đứng dưới đất không. |
| 8 | groundPoint | Transform | Điểm kiểm tra đất. |
| 9 | radius | float | Bán kính kiểm tra chạm đất. |
| 10 | whatIsGround | LayerMask | Xác định đâu là nền đất. |

*Bảng 2. 5. Bảng mô tả lớp UI\_QuestPlatform.cs*

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | questData | Quest | Dữ liệu câu hỏi. |
| 2 | lstQuestInUse | List<QuestInfor> | Danh sách những câu hỏi được sử dụng. |
| 3 | t\_quest1 | Text | Hiển thị thành phần thứ nhất của câu hỏi. |
| 4 | t\_quest2 | Text | Hiển thị thành phần thứ hai của câu hỏi. |
| 5 | questDisplay | QuestDisplay | Thực hiện hiển thị câu hỏi. |
| 6 | sp\_normalAns | Sprite | Hình ảnh nút đáp án ở trạng thái bình thường. |
| 7 | sp\_rightAns | Sprite | Hình ảnh đáp án khi đúng. |
| 8 | sp\_wrongAns | Sprite | Hình ảnh nút đáp án khi sai. |
| 9 | t\_numEgg | Text | Hiển thị số trứng đã thu hoạch được. |
| 10 | obj\_result | GameObject | Kết quả của bài học. |
| 11 | t\_numEggResult | Text | Hiển thị tổng số trứng thu hoạch được sau khi kết thúc hành trình. |
| 12 | obj\_EggResult | GameObject | Đối tượng chứa giao diện kết quả trứng thu được. |
| 13 | lstDragon | List<GameObject> | Dánh sách rồng. |
| 14 | obj\_closeBT | GameObject | Nút kết thúc bài học. |

| 15 | lstHealth | List<GameObject> | Danh sách trái tim nhân vật có. |
| --- | --- | --- | --- |

### *Biểu đồ lớp thực thể pha phân tích*



*Hình 2. 10. Biểu đồ lớp thực thể Unit 2*

### *Biểu đồ lớp thực thể pha thiết kế*

*Hình 2. 11. Biểu đồ lớp thiết kế Unit 2*

### *Biểu đồ tuần tự*Chart, box and whisker chart Description automatically generated

*Hình 2. 12. Biểu đồ tuần tự Unit 2*

### *Giao diện chơi*

* Phần mở đầu giới thiệu về câu chuyện và chủ đề bài học.



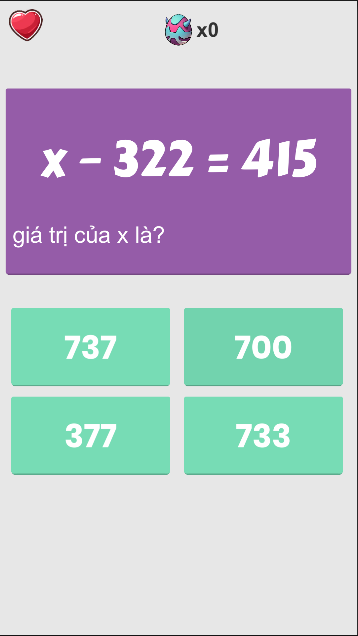
*Hình 2. 13. Phần mở đầu game Christmas Eggs*

* + - * + Giao diện game.



*Hình 2. 14. Giao diện game Christmas Eggs*

* + - * + Giao diện câu hỏi.



*Hình 2. 15. Giao diện câu hỏi game Christmas Eggs*

## Các thành phần chính

### *Lớp CharacterMng.cs*

* Lớp này dùng để điều khiển khiển nhân vật khi có sự kiện đến từ người chơi cũng như bắt các sự kiện khi nhân vật tương tác với các đối tượng trong game.
* Một số phương thức quan trọng:

- IsGrounded(): phương thức thực hiện kiểm tra xem người chơi có đang đứng ở trên mặt đất hay không.



- MoveRightBT(), MoveLeftBT(), JumpBT(): phương thức thực hiện điều hướng nhân vật di chuyển sang phải, trái, nhảy lên khi có sự kiện nhấn tương ứng từ người chơi.



- OnTriggerEnter2D(Collider2D other): hàm bắt sự kiện khi nhân vật chạm vào gai hoặc trứng rồng.

### *Lớp ChestObject.cs*

* Lớp này được gắn với đối tượng là chiếc rương ở cuối cuộc hành trình, nhiệm vụ chính sẽ là bắt sự kiện va chạm với người chơi sau đó hiển thị kết quả của người chơi và rồng thu được sau cuộc hành trình.

### *Lớp UI\_QuestPlatform.cs*

* Lớp này thực hiện nhiệm vụ hiển thị câu hỏi, hiển thị số trứng rồng người chơi thu được và rồng được ấp ra sau khi kết thúc hành trình.

## Unit 3: Math Game

## Thể loại

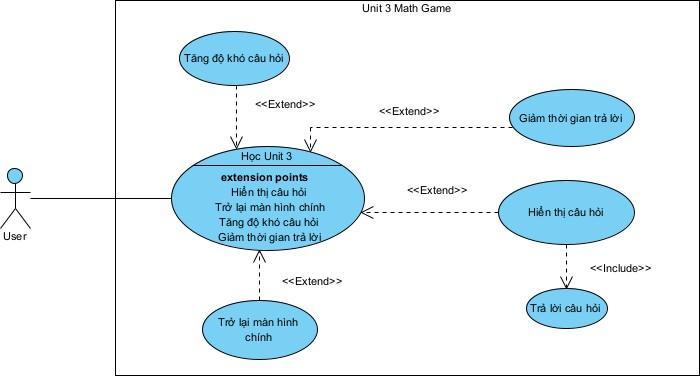
Game giải đố, endless.

## Kịch bản

* + - * Người chơi thực hiện tính toán và lựa chọn một trong 4 đáp án cho thành phần còn thiếu trong câu hỏi. Mỗi câu hỏi chỉ được trả lời trong một khoảng thời gian nhất định.
      * Cho đến khi trả lời đúng câu hỏi hiện tại thì người chơi mới được trả lời câu hỏi tiếp theo. Khi người chơi trả lời sai thì sẽ bị trừ một khoảng vào thời gian đang đếm ngược. Sau mỗi 5 câu trả lời đúng, độ khó của câu hỏi sẽ tăng lên.

## Thiết kế chi tiết[1]

### *Use case chi tiết*



*Hình 2. 16. Use case chi tiết Unit 3 Math Game*

Mô tả use case:

* + Trở lại màn hình chính: UC này cho phép user kết thúc bài học và trở về lại màn hình chính.
  + Trả lời câu hỏi: UC này cho phép user lựa chọn đáp án.
  + Hiển thị câu hỏi: UC này cho phép user xem nội dung câu hỏi.
  + Tăng độ khó câu hỏi: UC này cho phép tăng độ khó của câu hỏi khi user lên mức độ cao dần.
  + Giảm thời gian trả lời: UC này cho phép thực hiện trừ số thời gian đếm ngược của user khi user trả lời sai

### *Kịch bản*

| **Tên use case** | Học Unit 3. |
| --- | --- |
| **Tác nhân** | User |
| **Mục đích** | Vào bài học Unit 3. |
| **Tiền điều kiện** | User đang ở màn hình chính. |
| **Mô tả luồng sự kiện chính** | 1. User nhấn chọn nút Unit 3. 2. Hệ thống hiển thị chủ đề của bài học và nút bắt đầu. 3. User click chọn nút bắt đầu. 4. Hệ thống hiển thị: khung thời gian đếm ngược, cấp độ hiện tại của câu hỏi, câu hỏi và bốn đáp án lựa chọn. 5. User click chọn đáp án. 6. Hệ thống kiểm tra đáp án.    1. User trả lời đúng: hệ thống sẽ tăng thêm 1 điểm cho người chơi và hiển thị câu hỏi tiếp theo.    2. User trả lời sai: hệ thống hiển thị animation chọn sai và trừ đi một khoảng thời gian đếm ngược của người chơi. 7. User quay lại bước 5 cho đến khi user mở được hòm kho báu ở cuối map. 8. Hệ thống hiển thị kết quả gồm: số trứng đã nhặt được và nút mở trứng. 9. User click chọn nút mở trứng. 10. Hệ thống hiển thị rồng thu được tương ứng với số trứng và nút đóng. |

|  | 1. User click chọn nút đóng. 2. Hệ thống hiển thị màn hình chính. |
| --- | --- |
| **Hậu điều kiện** | Học xong Unit 3. |
| **Ngoại lệ** | * 1. User chạm vào gai thì mất một trái tim.   2. User mất hết trái tim thì kết thúc hành trình.   9.1 User không nhặt được quả trứng nào thì sẽ không mở được rồng. |

1. ***Xây dựng các lớp thực thể***

*Bảng 2. 6. Bảng mô tả lớp GameManager.cs*

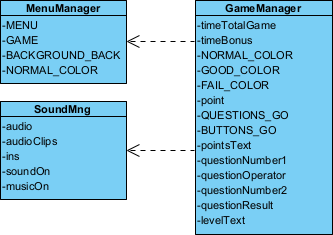
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | timeTotalGame | int | Tổng số thời gian cho một câu hỏi. |
| 3 | timeBonus | int | Thời gian được cộng thêm. |
| 4 | NORMAL\_COLOR | Color | Màu ở chế động bình thường. |
| 5 | GOOD\_COLOR | Color | Màu khi trả lời đúng. |
| 6 | FAIL\_COLOR | Color | Màu khi trả lời sai. |
| 7 | BACKGROUND\_BACK | Image | Ảnh nền. |
| 8 | point | point | Số điểm chơi. |
| 9 | QUESTIONS\_GO | GameObject | Đối tượng chứa câu hỏi. |
| 10 | BUTTONS\_GO | GameObject | Nút bắt đầu. |
| 11 | POINTS | GameObject | Đối tượng chứa điểm. |

| 12 | pointsText | Text | Hiển thị số điểm. |
| --- | --- | --- | --- |
| 13 | questionNumber1 | Text | Hiển thị số thứ nhất trong câu hỏi. |
| 14 | questionOperator | Text | Hiển thị phép toán trong câu hỏi. |
| 15 | questionNumber2 | Text | Hiển thị số thứ 2 trong câu hỏi. |
| 16 | questionResult | Text | Hiển thị kết quả trong câu hỏi. |
| 17 | level | int | Số level hiện tại. |
| 18 | \_score | int | Biến đếm số điểm hiện tại. |
| 19 | GOODANSWER | int | Số lượng câu hỏi trả lời tốt. |
| 20 | levelText | Text | Hiển thị level hiện tại. |

*Bảng 1 Bảng mô tả lớp MenuManager.cs*

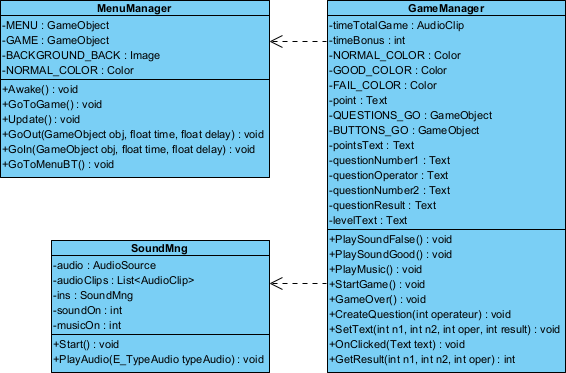
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | MENU | GameObject | Giao diện menu. |
| 2 | GAME | GameObject | Giao diện chơi . |
| 3 | BACKGROUND\_BACK | Image | Ảnh nền. |
| 4 | NORMAL\_COLOR | Color | Màu các nút ở trạng thái bình thường. |

### *Biểu đồ lớp thực thể pha phân tích*



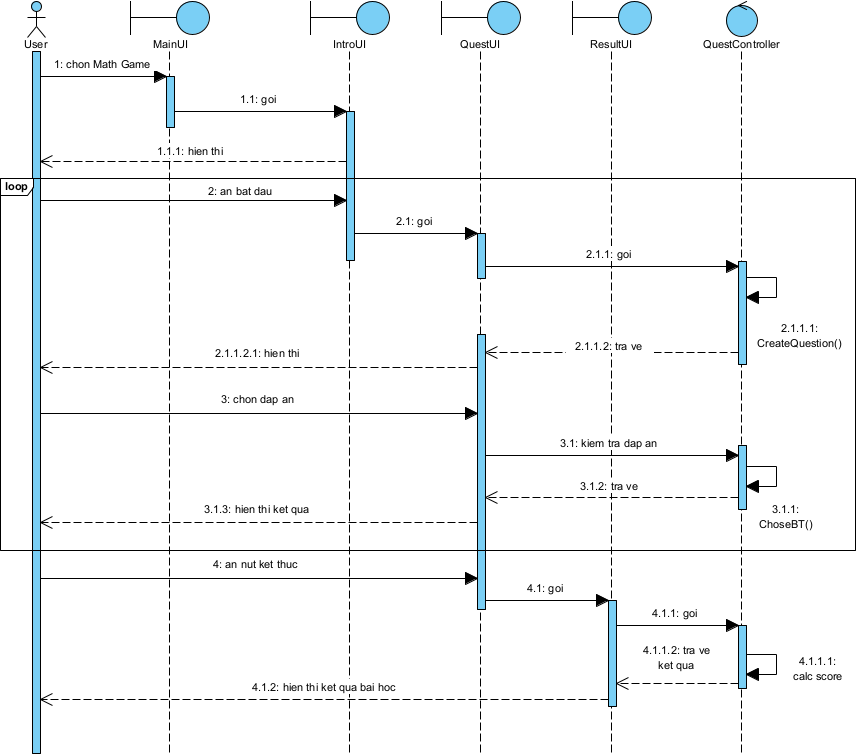
*Hình 2. 17. Biểu đồ lớp thực thể Unit 3*

### *Biểu đồ lớp thực thể pha thiết kế*



*Hình 2. 18. Biểu đồ lớp thiết kế Unit 3*

### *Biểu đồ tuần tự*



*Hình 2. 19. Biểu đồ tuần tự Unit 3*

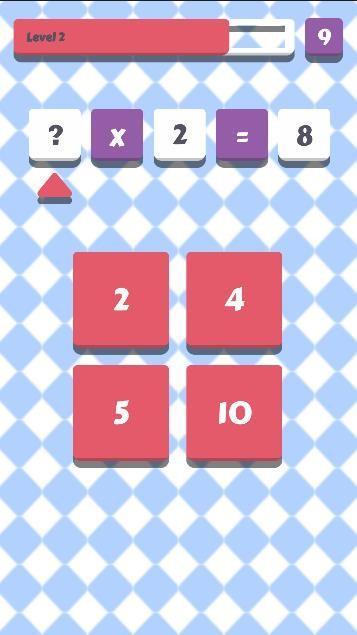
### *Giao diện chơi*

Giao diện mở đầu game.



*Hình 2. 20. Mở đầu Math Game*

Math game có giao diện trả lời câu hỏi như sau.



*Hình 2. 21. Giao diện trả lời câu hỏi Math game*

## Các thành phần chính

### *Lớp AnswerInGameButton.cs*

* + Lớp này thực hiện xử lý sự kiện khi người chơi nhấn chọn kết quả và hiển thị các hoạt ảnh phản hồi trở lại để người chơi nhận biết mình đang trả lời sai hay đúng.





### *Lớp GameManager.cs*

* + Lớp này thực hiện quản lý mọi hoạt động trong game như: sinh ra câu hỏi, thực hiện đếm ngược thời gian, kiểm tra đáp án người chơi.
  + Các phương thức để phát nhạc trong game: PlaySoundFalse(): phát nhạc khi trả lời sai. PlaySoundGood(): phát nhạc khi trả lời đúng. PlayMusic(): phát nhạc nền.

StopMusic(): dừng phát nhạc nền.



* Phương thức TimerStart(): đếm thời gian trong game, nếu hết thời gian mà người chơi chưa trả lời đúng thì sẽ kết thúc game.



* Phương thức CreateQuestion(int operateur): thực hiện tự động tạo câu hỏi dựa theo cấp độ hiện tại của người chơi và truyền dữ liệu câu hỏi để hiển thị ra màn hình.

## Unit 4: Game Dungeon

## Thể loại

Game giải đố, platform.

## Kịch bản

* + - * Giải cứu người khỏi hang động, người chơi sẽ thực hiện việc di chuyển người chơi và vượt qua các chướng ngại vật trong hang động.
      * Trong hang động sẽ có nhiều hướng đi khác nhau. Khi người chơi vượt chướng ngại vật, nếu trả lời đúng câu hỏi thì sẽ được đi tiếp nếu trả lời sai thì con đường đang đi sẽ bị chặn lại và người chơi phải chọn một con đường khác để đi.
      * Khi bắt đầu chơi, người chơi sẽ chỉ được nhìn thấy mọi thứ xung quanh trong một bán kính nhỏ, sau khi trả lời đúng câu hỏi thì vùng nhìn thấy sẽ càng rộng hơn.

## Thiết kế chi tiết[1]

### *Use case chi tiết*

*Hình 2. 22. Use case chi tiết unit 4*

Mô tả UC:

* + Di chuyển nhân vật: UC cho phép user di chuyển nhân vật.
  + Bắt đầu lại bài học: UC cho phép user bắt đầu lại bài học ở thời điểm đang học.
  + Nhặt vật phẩm: UC cho phép user thực hiện tương tác với vật phẩm trong game.
  + Trả lời câu hỏi: UC cho phép user thực hiện trả lời câu hỏi trong bài học.

### *Kịch bản*

| **Tên use case** | Học Unit 4. |
| --- | --- |
| **Tác nhân** | User |
| **Mục đích** | Vào bài học Unit 4. |
| **Tiền điều kiện** | User đang ở màn hình chính. |
| **Mô tả luồng sự kiện chính** | 1. User nhấn chọn nút Unit 4. 2. Hệ thống hiển thị chủ đề của bài học và nút bắt đầu. |

|  | 1. User click chọn nút bắt đầu. 2. Hệ thống hiển thị: nhân vật, map chơi, các vật phẩm, các nút di chuyển... 3. User di chuyển nhân vật chạm vào vật phẩm trên đường đi. 4. Hệ thống hiển thị câu hỏi, khung điền đáp án và nút trả lời. 5. User điền đáp án và click nút trả lời. 6. Hệ thống hiển thị đáp án đúng.    1. User trả lời đúng: hệ thống sẽ tăng khoảng nhìn thấy của người chơi lên và người chơi có thể đi tiếp.    2. User trả lời sai: hệ thống sẽ chặn đường hiện tại và đẩy người chơi về vị trí cũ. 7. User quay lại bước 5 cho đến khi đi đến đích. 8. Hệ thống hiển thị kết quả thông báo hoàn thành bài học. 9. User click chọn nút đóng. 10. Hệ thống hiển thị màn hình chính. |
| --- | --- |
| **Hậu điều kiện** | Học xong Unit 4. |
| **Ngoại lệ** | 6.1 User không điền đáp án.  9.1 User nhấn nút chơi lại. |

1. ***Xây dựng các lớp thực thể***

*Bảng 2. 7. Bảng mô tả lớp CharDungeonMng.cs*

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | rb | Rigidbody2D | Điều khiển vị trí nhân vật thông qua hệ thống mô phỏng vật lý. |
| 2 | speed | float | Tốc độ di chuyển của nhân vật. |
| 3 | whatIsWall | LayerMask | Biến xác định đâu là tường. |
| 4 | distance | float | Biến kiểm tra khoảng cách người chơi và các tường xung quanh. |

| 5 | canMoveUp | bool | Biến kiểm tra có thể di chuyển lên trên hay không. |
| --- | --- | --- | --- |
| 6 | canMoveDown | bool | Biến kiểm tra có thể di chuyển xuống dưới hay không. |
| 7 | canMoveLeft | bool | Biến kiểm tra có thể di chuyển sang trái hay không. |
| 8 | camMoveRight | bool | Biến kiểm tra có thể di chuyển sang phải hay không. |
| 9 | isFaceRight | bool | Biến kiểm tra hướng quay mặt của nhân vật. |
| 10 | moveStatus | E\_Move | Biến enum kiểm soát trạng thái di chuyển của nhân vật. |
| 11 | obj\_hide | GameObject | Đối tượng màn che tầm nhìn người chơi. |

*Bảng 2. 8. Bảng mô tả lớp UIDungeonQuestMng.cs*

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | obj\_questContent | GameObject | Chứa giao diện câu hỏi. |
| 2 | t\_numQuest | Text | Hiển thị số thứ tự câu hỏi hiện tại. |
| 3 | lstPics | List<GameObject> | Chứa danh sách các hình. |
| 4 | inputAns | InputField | Chứa thông tin câu hỏi người chơi điền vào. |
| 5 | rightAns | Text | Hiển thị câu trả lời đúng. |
| 6 | picQuest | PicQuest | Hiển thị ảnh câu hỏi. |
| 7 | objClosebtn | GameObject | Nút đóng câu hỏi. |

| 8 | objEndDungeon | GameObject | Chứa giao diện kết thúc bài học. |
| --- | --- | --- | --- |
| 9 | lstIDQuest | List<int> | Chứa danh sách định danh câu hỏi. |
| 10 | backPoint | ItemDungeon | Chứa thông tim điểm quay lại khi trả lời sai. |

*Bảng 2. 9. Bảng mô tả lớp ItemDungeon.cs*

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | boxCollider2D | BoxCollider2D | Thành phần để kiểm tra va chạm. |
| 2 | closeWaySprite | Sprite | Hình ảnh hiển thị khi item đóng lại. |
| 3 | pointBack | GameObject | Điểm quay trở lại của người chơi khi trả lời sai. |

### *Biểu đồ lớp thực thể pha phân tích*

*Hình 2. 23. Biểu đồ lớp thực thể Unit 4*

### *Biểu đồ lớp thực thể pha thiết kế*

*Hình 2. 24. Biểu đồ lớp phân tích Unit 4*

### *Biểu đồ tuần tự*Box and whisker chart Description automatically generated with low confidence

*Hình 2. 25. Sơ đồ tuần tự Unit 4.*

### *Giao diện chơi*

* + Giao diện mở đầu.



*Hình 2. 26. Giao diện mở đầu game Dungeon*

* + Giao diện màn chơi.



*Hình 2. 27. Giao diện màn chơi game Dungeon*

## Các thành phần chính

### *Lớp CharDungeonMng.cs*

* + Lớp này thực hiện điều khiển người chơi, nhận các sự kiện tương tác từ người chơi và xử lý tương tác của nhân vật với các đồ vật trong game.
  + Hàm Update(): thực hiện cập nhật trạng thái và di chuyển nhân vật dựa theo trạng thái và kiểm tra va chạm của nhân vật với các thành tường bốn phía(trên, dưới, trái, phải).



* + Các hàm nhận sự kiện tương tác di chuyển.



### *Lớp UIDungeonQuestMng.cs*

* + Lớp này quản lý các hoạt động trong game liên quan đến việc hiển thị như: hiển thị câu hỏi, hiển thị kết quả.
  + Hàm **ShowQuest():** thực hiện việc lấy ngẫu nhiên câu hỏi trong bộ câu hỏi và hiển thị câu hỏi lên màn hình.



* + Hàm **CheckAnsBT():** thực hiện kiểm tra đáp án mà người chơi đã nhập và hiển thị đáp án đúng. Nếu đáp án đúng thì chướng ngại vật sẽ biến mất. Nếu đáp án sai thì sẽ trả người chơi về vị trí cũ và đóng lối đi hiện tại.

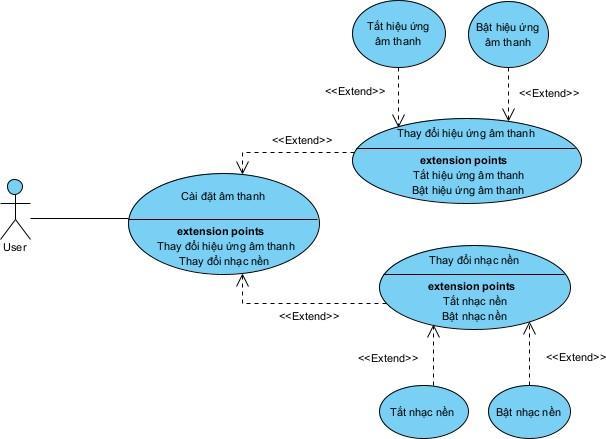
## Chức năng cài đặt âm thanh

## Kịch bản

User muốn bật hoặc tắt âm thanh, nhạc nền của ứng dụng khi thực hiện học các bài học.

## Thiết kế chi tiết[1]

### *Use case chi tiết*



*Hình 2. 28. Use case chi tiết chức năng cài đặt âm thanh*

### *Kịch bản*

| **Tên use case** | Cài đặt âm thanh. |
| --- | --- |
| **Tác nhân** | User |
| **Mục đích** | User có thể tùy ý bật, tắt các loại âm thanh trong ứng dụng. |
| **Tiền điều kiện** | User đang ở màn hình chính. |
| **Mô tả luồng sự kiện chính** | 1. User nhấn chọn nút cài đặt âm thanh. 2. Hệ thống hiển thị giao diện cài đặt âm thanh. Gồm nút cho âm thanh và nhạc nền. 3. User click các nút thay đổi âm thanh. |

|  | 1. Hệ thống hiển thị giao diện theo lựa chọn của user. 2. User click đóng giao diện. 3. Hệ thống hiển thị giao diện màn hình chính. |
| --- | --- |
| **Hậu điều kiện** | Âm thanh được thay đổi cài đặt theo ý của user. |
| **Ngoại lệ** | Không |

1. ***Xây dựng các lớp thực thể***

*Bảng 2. 10. Bảng mô tả lớp UI\_MenuSettingMng.cs*

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Nội dung** |
| --- | --- | --- | --- |
| 1 | obj\_offSoundBtn | GameObject | Nút tắt hiệu ứng âm thanh. |
| 2 | obj\_offMusicBtn | GameObject | Nút bật hiệu ứng âm thanh. |
| 3 | soundOn | int | Biến xác định bật, tắt của hiệu ứng âm thanh. |
| 4 | musicOn | int | Biến xác định bật, tắt của nhạc nền. |

### *Giao diện*

*Hình 2. 29. Giao diện cài đặt âm thanh*

## Tổng kết

Chương này đã hoàn thành việc giới thiệu về ứng dụng và hoàn thành phân tích thiết kế cho 4 Unit ứng với các bài học từ một đến bốn. Ở chương 3 đồ án sẽ tiến hành đi đến các bước cài đặt trong đồ án, khởi tạo dự án.

# CHƯƠNG III CÀI ĐẶT ỨNG DỤNG

Trong chương III, đồ án sẽ trình bày về các yêu cầu hệ thống, cài đặt dự án và xử lý các thành phần trong game. Nội dung bao gồm:

* + - Yêu cầu hệ thống
    - Cài đặt
    - Xử lý các thành phần trong game
    - Tổng kết

## Yêu cầu hệ thống

* + 1. **Yêu cầu lập trình**
* Unity Engine 2020.3.18f1
* Jetbrain Rider 2020.1
* Android SDK, NDK Tools
* Open JDK

## Yêu cầu cài đặt

* Cấu hình yêu cầu

Game được thiết kế cho các thiết bị di động có cấu hình:

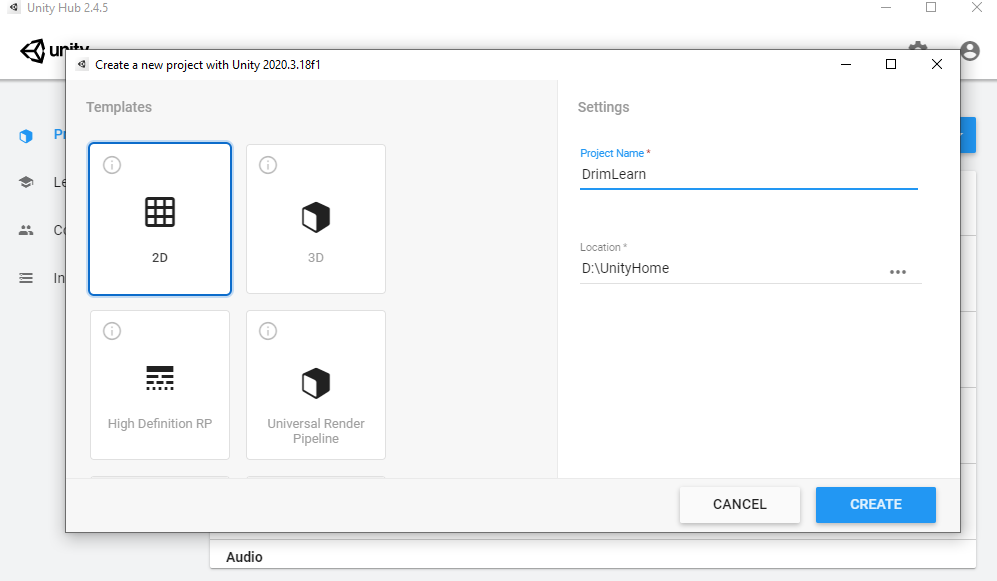
* + Hệ điều hành Android 4.4 KitKat trở lên.
  + Độ phân giải màn hình 1920x1080 pixel trở lên.
* Cấu hình đề nghị.

Để hiển thị đầy đủ các thành phần trong game, không làm ảnh hưởng đến hệ thống, cấu hình đề nghị cho thiết bị di động:

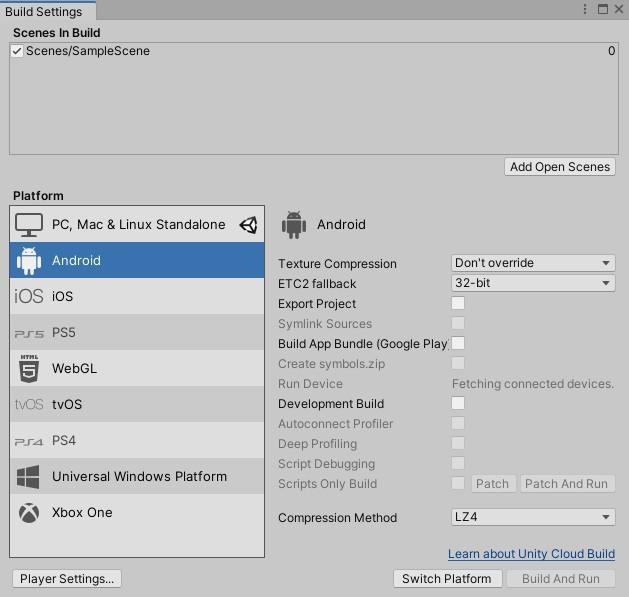
* + Hệ điều hành Android 4.0 trở lên.
  + Độ phân giải màn hình 1920x1080 pixel.

## Cài Đặt

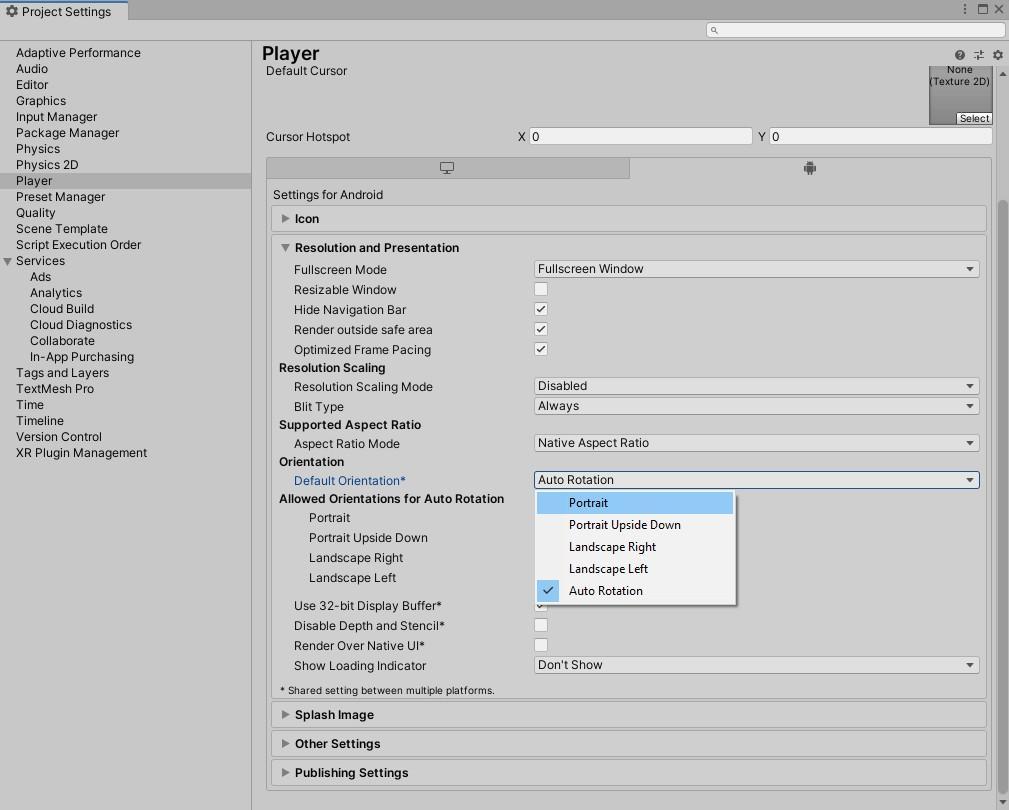
* Bước 1: Tạo một project Unity 2D mới: nhập tên project và nhấn Create.



*Hình 3. 1. Tạo project Unity 2D*

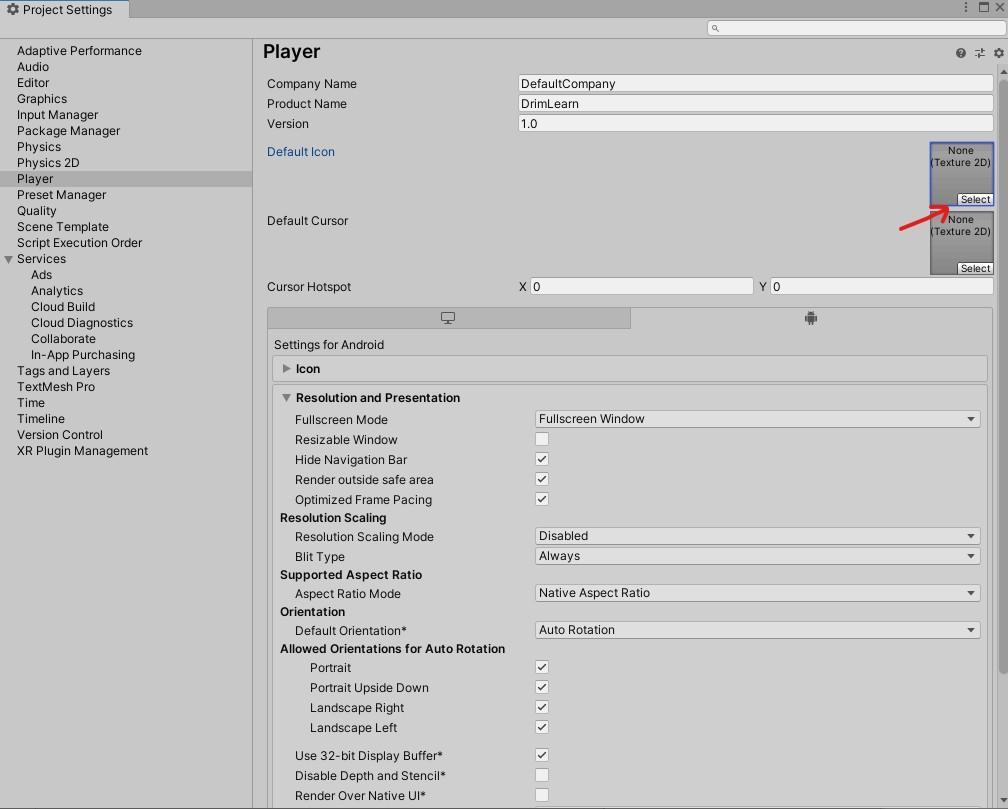
* Bước 2: Cài đặt platform sang android: File => build setting => chọn android => Switch Platform.

*Hình 3. 2. Cài đặt platform*

* Bước 3: cài đặt ứng dụng luôn quay dọc: từ bước 2, ấn Player Settings => Resolution and Presentation => Default Orientation => Portrait.

*Hình 3. 3. Cài đặt chiều xoay ứng dụng*

* Bước 4: Chọn logo ứng dụng.



*Hình 3. 4. Chọn logo ứng dụng*

## Xử lý các thành phần trong game

## Xử lý đồ họa[8]

* + - * Trong game bao gồm rất nhiều yếu tố liên quan đến đồ họa nên việc làm sao cho hình ảnh bắt mắt để có thể thu hút người dùng là 1 điều hết sức quan trọng.
      * Một vài hình ảnh đơn giản ta có thể xử lý thông qua công cụ Adobe Photoshop như thay đổi trạng thái của các nút ấn(trạng thái bình thường, trạng thái khi nhấn, trạng thái khi không có khả năng tương tác).
      * Với một button thông thường, ta sẽ có trạng thái khi kích hoạt và không kích hoạt. Thông qua 1 vài thao tác đơn giản trong photoshop bằng việc đổi màu của bức ảnh có sẵn, ta đã có 2 trạng thái của button đó.

## Xử lý âm thanh

Trong game, các âm thanh hiệu ứng thường ngắn và có kích thước nhỏ và nếu muốn chúng lặp lại thì ta sẽ cho phát lặp lại liên tục. Các file nhạc sử dụng thường có đuôi .wav, .ogg, …

Một vài file âm thanh quá dài hoặc quá ngắn ta có thể xử lý cắt hoặc ghép nhiều file âm thanh lại với nhau 1 cách đơn giản thông qua các trang xử lý âm thanh trực tuyến.

## Tổng kết

Trong chương này đồ án đã hoàn thành việc cài đặt và xử lý các tài nguyên trong ứng dụng. Ở chương tiếp theo, đồ án sẽ đi đến kết luận và hướng phát triển cho ứng dụng.

# CHƯƠNG IV KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong chương này em sẽ trình bày những kết quả đã đạt được sau thời gian xây dựng ứng dụng và tổng kết lại những hạn chế trong đồ án cũng như phương hướng phát triển cho tương lai. Nội dung bao gồm:

* Kết quả đạt được
* Những hạn chế
* Hướng phát triển

## Kết quả đạt được

Đề tài **Xây dựng ứng dụng học tập DrimLearn bằng Unity** tuy vẫn chưa được hoàn thiện một cách xuất sắc và quy mô cũng chưa thật sự lớn nhưng em hy vọng nó sẽ có ích trong việc nâng cao kiến thức cho người học đặc biệt là trong thời gian dịch bệnh covid hoành hành.

Trên cơ sở việc tìm hiểu và xây dựng ứng dụng học tập kết hợp vui chơi trên thiết bị di động thông qua nền tảng Unity Engine. Kết hợp với những kiến thức mà em học và có được trong thời gian vừa qua, em đã đạt được kết quả thông qua đồ án như sau:

* + - Giới thiệu 1 cách khái quát về game trên nền tảng di động.
    - Giới thiệu khái quát về nền tảng Unity Engine, các thành phần của Unity và ứng dụng của nền tảng này trong lĩnh vực làm game.
    - Kỹ năng lập trình và phân tích thiết kế ứng dụng.
    - Giới thiệu sơ lược về những thành phần, các bước cơ bản của 1 game trên nền tảng Unity.
    - Trên cơ sở đó xây dựng 1 số game đơn giản ứng dụng các tính năng của Unity kết hợp với các câu hỏi tạo nên ứng dụng học tập.
    - Luồng chương trình dễ hiểu, các bài học được giới thiệu thông qua các câu chuyện nhỏ. Giao diện thân thiện và dễ sử dụng.

## Những hạn chế

Bên cạnh đó, đồ án cũng như ứng dụng cũng còn một số hạn chế đó là:

* + - Các game có cách chơi đơn giản, chưa thực sự hấp dẫn với người dùng.
    - Gameplay chưa có nhiều mức độ chơi.
    - Tính đa dạng của bộ câu hỏi còn hạn chế do đây chỉ là ứng dụng offline.
    - Hiệu năng của ứng chưa được tốt, đôi lúc còn hiện tượng giật, lag do tối ưu đồ họa chưa tốt.

## Hướng phát triển

Thông qua những kết quả và hạn chế đó, đề xuất ứng một vài hướng phát triển ứng dụng như sau:

* + - Trước hết cần tập trung tối ưu hóa hiệu năng của ứng dụng để giúp người dùng có trải nghiệm tốt nhất bằng cách tối ưu hóa đồ họa sử dụng, giảm thiểu việc đồ họa được sử dụng 1 cách tràn lan gây lãng phí bộ nhớ.
    - Ứng dụng sẽ cải thiện hệ thống gameplay. Xây dựng những gameplay hấp dẫn, cuốn hút hơn với người dùng bằng các tạo ra các game có tương tác nhiều hơn, nhiều màn chơi hơi, độ khó tăng cấp nhiều hơn, nhiều vật dụng, tính năng trong game hơn,…
    - Đưa đến cho người dùng lượng kiến thức chính xác, phong phú hơn thông qua việc xây dựng bộ câu hỏi đa dạng và chất lượng hơn. Mở rộng thêm nhiều bộ môn khác.

## TÀI LIỆU THAM KHẢO

**Danh mục các giáo trình tham khảo:**

1. Trần Đình Quế. *Phân tích và thiết kế hệ thống thông tin*. Học viện Công nghệ Bưu chính Viễn thông. 2018.
2. Trần Hương Giang. *Kỹ năng tạo lập văn bản tiếng Việt*. Học viện Công nghệ Bưu chính Viễn thông. 2016.

## Danh mục các website tham khảo:

1. “Các thành phần cơ bản trong Unity” https://docs.unity3d.com/Manual/index.html
2. “Giới thiệu về Unity Engine”

https://viblo.asia/p/gioi-thieu-ve-unity-engine-game-engine-pho-bien-nhat- hien-nay-V3m5WBj8lO7

1. “Các khái niệm cơ bản trong Unity”

[https://ww](http://www.stdio.vn/unity-game/cac-khai-niem-trong-game-b0WZE)w.stdi[o.vn/unity](http://www.stdio.vn/unity-game/cac-khai-niem-trong-game-b0WZE)-[game/cac](http://www.stdio.vn/unity-game/cac-khai-niem-trong-game-b0WZE)-[khai-niem-trong-game-b0WZE](http://www.stdio.vn/unity-game/cac-khai-niem-trong-game-b0WZE)

1. “Tương tác vật lý trong Unity”

https://viblo.asia/p/co-ban-ve-engine-vat-ly-trong-unity-3OEqGj1PM9bL

1. “Spine animation” [https://ww](http://www.raywenderlich.com/2589)w.raywe[nderlich.com/2589](http://www.raywenderlich.com/2589)
2. Hình ảnh [https://ww](http://www.freepik.com/)w.freepi[k.com/](http://www.freepik.com/)
3. Font chữ [https://ww](http://www.dafont.com/)w.dafon[t.com/](http://www.dafont.com/)
4. “DOTween Unity” <http://dotween.demigiant.com/documentation.php>
5. “Odin - Inspector and Serializer” https://odininspector.com/tutorials
6. “Bài tập toán lớp 3” https://vietjack.com/giai-vo-bai-tap-toan-3/