

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KHOA HỌC MÁY TÍNH

ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG GAME KNIGHT BATTLE
MATHS SỬ DỤNG UNITY 2D FRAMEWORK

Cán bộ hướng dẫn : TS. Phạm Văn Hà
Sinh viên thực hiện : Nguyễn Tuyển Lâm
Mã sinh viên : 2020606451

HÀ NỘI - 2024

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến Thầy Phạm Văn Hà vì những hỗ trợ và đồng hành trong suốt quá trình thực hiện Đồ án tốt nghiệp ngành Khoa học máy tính. Dưới sự hướng dẫn tận tâm và tâm huyết của Thầy, em đã có cơ hội tiếp cận và khám phá sâu hơn về lĩnh vực này, đồng thời đạt được những thành tựu quan trọng trong dự án của mình.

Thầy là một người giáo viên kiên nhẫn, am hiểu và luôn sẵn lòng chia sẻ kiến thức và kinh nghiệm của mình. Những lời khuyên, góp ý và hỗ trợ từ Thầy đã giúp em xây dựng nền tảng kiến thức vững chắc hơn và phát triển khả năng giải quyết vấn đề một cách hiệu quả.

Em rất tự hào và biết ơn vì đã có cơ hội được học tập và làm việc dưới sự hướng dẫn của Thầy. Đồ án tốt nghiệp không chỉ là kết quả của công sức riêng của em mà còn là sự hỗ trợ và đóng góp đáng kể từ Thầy.

Xin chân thành cảm ơn và kính chúc sức khỏe, thành công và hạnh phúc Thầy và gia đình.

Em xin chân thành cảm ơn !

Sinh viên thực hiện

Nguyễn Tuyển Lâm

MỤC LỤC

| | |
|--|------|
| MỤC LỤC..... | ii |
| DANH MỤC HÌNH ẢNH | v |
| DANH MỤC BẢNG BIỂU | vii |
| MỞ ĐẦU..... | viii |
| CHƯƠNG 1. TỔNG QUAN VỀ ENGINE UNITY VÀ KHÔNG GIAN 2D | 1 |
| 1.1. Tổng quan về Unity..... | 1 |
| 1.1.1 Unity là gì?..... | 1 |
| 1.1.2 Quá trình phát triển của Unity..... | 1 |
| 1.1.3 Lịch sử hình thành..... | 2 |
| 1.1.4 Tính năng của Engine Unity | 3 |
| 1.1.5 Một số thống kê về Unity..... | 4 |
| 1.1.6 Ưu điểm của Unity và kiến trúc tổng quan | 4 |
| 1.2. Các khái niệm trong Unity | 6 |
| 1.2.1 Assets..... | 6 |
| 1.2.2 Scene | 7 |
| 1.2.3 GameObject..... | 7 |
| 1.2.4 Component | 8 |
| 1.2.5 Scripts..... | 9 |
| 1.2.6 Prefabs..... | 9 |
| 1.2.7 Collider..... | 10 |
| 1.2.8 Rigidbody | 12 |

| | |
|--|----|
| 1.2.9 Sprite | 13 |
| 1.2.10 Animator..... | 13 |
| 1.2.11 Audio Source | 14 |
| 1.2.12 Camera | 15 |
| 1.2.13 Transfrom | 16 |
| 1.2.14 Renderer | 16 |
| 1.2.15 Cửa sổ làm việc | 17 |
| CHƯƠNG 2. THIẾT KẾ HỆ THỐNG GAME | 20 |
| 2.1. Ý tưởng và cốt truyện..... | 20 |
| 2.1.1 Ý tưởng..... | 20 |
| 2.1.2 Cốt truyện..... | 20 |
| 2.2. Thiết kế nhân vật và bối cảnh | 21 |
| 2.2.1 Các nhân vật trong trò chơi..... | 21 |
| 2.2.2 Các hành động trong trò chơi..... | 21 |
| 2.3. Thiết kế bối cảnh | 24 |
| 2.4. Thiết kế cơ chế của game:..... | 24 |
| 2.5. Thiết kế nội dung..... | 25 |
| 2.6. Thiết kế âm thanh..... | 26 |
| 2.7. Thiết kế dữ liệu của game | 27 |
| 2.7.1 Bộ dữ liệu của câu hỏi..... | 27 |
| 2.7.2 Bộ dữ liệu của các màn chơi | 28 |
| 2.8. Thiết kế giao diện..... | 28 |
| 2.8.1 Giao diện màn chơi | 28 |

| | |
|--|-----------|
| 2.8.2 Giao diện các màn hình..... | 29 |
| CHƯƠNG 3. XÂY DỰNG GAME VÀ KẾT QUẢ ĐẠT ĐƯỢC | 33 |
| 3.1. Xây dựng game | 33 |
| 3.1.1 Tạo project..... | 33 |
| 3.1.2 Khởi tạo file assets | 33 |
| 3.1.3 Nhập dữ liệu | 34 |
| 3.2. Kết quả đạt được | 36 |
| 3.2.1 Giao diện chính | 36 |
| 3.2.2 Giao diện chọn Level | 37 |
| 3.2.3 Giao diện chơi game | 37 |
| 3.2.4 Giao diện khi hết thời gian..... | 38 |
| 3.2.5 Giao diện khi thắng màn | 39 |
| 3.2.6 Giao diện Setting:..... | 39 |
| CHƯƠNG 4. KIỂM THỬ HỆ THỐNG | 40 |
| 4.1. Kế hoạch kiểm thử | 40 |
| 4.1.1 Mục tiêu kiểm thử | 40 |
| 4.1.2 Phạm vi kiểm thử | 40 |
| 4.1.3 Các ca kiểm thử..... | 40 |
| KẾT LUẬN | 43 |
| TÀI LIỆU THAM KHẢO..... | 44 |

DANH MỤC HÌNH ẢNH

| | |
|--|----|
| Hình 1.1. Hình ảnh Assets Store | 6 |
| Hình 1.2. Scene trong game | 7 |
| Hình 1.3. Component | 8 |
| Hình 1.4 Box Collider | 10 |
| Hình 1.5 Capsule Collider..... | 10 |
| Hình 1.6 Mesh Collider..... | 11 |
| Hình 1.7 Sphere Collider..... | 11 |
| Hình 1.8 Wheel Collider | 12 |
| Hình 1.9 Terrain Collider | 12 |
| Hình 1.10 Transform | 16 |
| Hình 1.11 Renderer | 16 |
| Hình 2.1 Bối cảnh game..... | 24 |
| Hình 2.2 Minh hoạ dữ liệu câu hỏi | 27 |
| Hình 2.3 Minh hoạ dữ liệu màn chơi | 28 |
| Hình 2.4 Thiết kế giao diện màn chơi..... | 28 |
| Hình 2.5 Thiết kế màn hình chính..... | 29 |
| Hình 2.6 Thiết kế màn hình chọn Level..... | 30 |
| Hình 2.7 Màn hình chế độ chơi..... | 32 |
| Hình 3.1 Cửa sổ Projects..... | 33 |
| Hình 3.2 Thư mục Assets | 33 |
| Hình 3.3 Thư mục Scenes | 34 |
| Hình 3.4 Level_Data | 35 |

| | |
|--|----|
| Hình 3.5 Question_Data..... | 35 |
| Hình 3.6 Thư mục Script..... | 36 |
| Hình 3.7 Màn hình Home..... | 36 |
| Hình 3.8 Màn hình chọn Level | 37 |
| Hình 3.9 Màn hình Game Play..... | 37 |
| Hình 3.10 Màn hình thua | 38 |
| Hình 3.11 Màn hình qua màn..... | 39 |
| Hình 3.12 Màn hình Setting của game..... | 39 |

DANH MỤC BẢNG BIỂU

| | |
|---|----|
| Bảng 2.1 Các nhân vật trong trò chơi | 21 |
| Bảng 2.2 Các hành động của người chơi | 22 |
| Bảng 2.3 Các hành động của quái vật trong trò chơi..... | 23 |
| Bảng 2.4 Nội dung câu hỏi..... | 25 |
| Bảng 2.5 Âm thanh trong game | 27 |
| Bảng 2.6 Các đối tượng trong màn chơi | 29 |
| Bảng 2.7 Các đối tượng trong màn hình chính | 30 |
| Bảng 2.8 Các đối tượng trong màn hình chọn Level | 31 |
| Bảng 2.9 Bảng các đối tượng trong màn chế độ chơi | 32 |
| Bảng 4.1 Kiểm thử chức năng..... | 41 |
| Bảng 4.2 Kiểm thử giao diện | 42 |

MỞ ĐẦU

Ngành công nghệ thông tin là một ngành khoa học ngày càng phát triển mạnh mẽ và ứng dụng rộng rãi trên nhiều lĩnh vực khác nhau đã đóng góp phần lớn cho sự phát triển của đất nước và sự phát triển của con người. Hiểu theo nghĩa rộng và tổng quát là việc sử dụng công nghệ hiện đại vào việc tạo ra, lưu trữ, truyền dẫn thông tin, khai thác thông tin.

Cùng với xu hướng phát triển của các phương tiện truyền thông như Radio, Báo,... thì việc sử dụng Internet ngày càng phổ biến hơn, chúng ta có một kho thông tin khổng lồ phục vụ mọi nhu cầu, mục đích mà ta cần dùng. Và công nghệ thông tin sẽ ngày càng được phát triển hơn vì những lợi ích nó đã đem lại cho tất cả người dùng và cho xã hội.

Kể từ năm 2020, khi cả thế giới thực hiện biện pháp “lockdown”, hạn chế tối đa mọi sự gặp mặt trực tiếp đã vô tình mở ra cơ hội cho ngành công nghiệp Game tăng trưởng mạnh mẽ. Thị trường Game toàn cầu tạo ra mức doanh thu 159,3 tỷ USD vào năm 2020, tăng 9,3% so với cùng kỳ năm trước và dự kiến sẽ vượt mốc 200 tỷ USD vào năm 2025.

Số liệu khảo sát từ Newzoo về doanh thu toàn cầu của ngành công nghiệp Game đã cho thấy tiềm năng phát triển vô cùng mạnh mẽ của thị trường Game tại khu vực Châu Á - Thái Bình Dương, trong đó có cả Việt Nam. Theo số liệu khảo sát năm 2020 từ kênh Statista, Việt Nam là quốc gia có tỷ lệ người trưởng thành chơi game trên thiết bị điện tử cao nhất thế giới với 94% người trưởng thành cho biết họ thỉnh thoảng có chơi game và 85% trong số đó sử dụng thiết bị di động để chơi game.

Bên cạnh việc trở thành một trong những quốc gia có tỷ lệ người trưởng thành chơi game cao trên thế giới, Việt Nam cũng là nguồn "outsourcing" chất lượng cho nhiều công ty phát triển Game nổi tiếng như:

Vinagame - VNG, Glass Egg, Digital Works, NCSoft, SPARX* - A Virtuos Studio, Bombus, Vietnam Esports,...

Hiện nay, các studio sản xuất game trong nước và nước ngoài đang mở ra rất nhiều cơ hội cho các bạn trẻ được tự do phát triển nghề nghiệp trong lĩnh vực game yêu thích của mình với nhiều vị trí tuyển dụng đa dạng như: Game Developer, Game Designer, Systems Designer, UI Designer,... Trong đó, hai vị trí được tuyển dụng nhiều nhất là Game Developer (nhà phát triển Game), Game Designer (nhà thiết kế Game) và Game Programmer (lập trình Game) với mức lương và chế độ đãi ngộ hấp dẫn. Theo thống kê mới nhất từ trang SalaryExpert, mức lương trung bình cho vị trí Game Developer là khoảng 187.000.000 VNĐ/năm (tương đương khoảng 90.000 VNĐ/giờ), mức lương của Game Artist khoảng 389.000.000 VNĐ/năm (tương đương khoảng 187.000 VNĐ/giờ), mức lương trung bình cho vị trí Game Designer và Game Programmer gần bằng nhau, khoảng 449.000.000 VNĐ/năm (tương đương 215.000 VNĐ/giờ).

Phát triển trò chơi là một sự thử thách nhưng rất vui. Theo như những chia sẻ của các chuyên gia trong ngành, phát triển game là một công việc không hề đơn giản chút nào. Nó không phải là cái gì đó được thực hiện chỉ bằng việc chơi thật nhiều game. Trên thực tế nó được thực hiện bởi 1 quá trình bao gồm nhiều đêm không ngủ, sự nỗ lực hết mình, và sự cống hiến của mọi thành viên. Tuy nhiên, tất cả những đam mê, năng lượng và những đêm trắng đó đều rất đáng công khi bạn thấy sản phẩm của mình được ra đời với sự hoàn thiện và nhận được sự hưởng ứng nhiệt tình của game thủ.

Nhận được sự hưởng ứng nhiệt tình của game thủ. Qua quá trình tìm hiểu về phần mềm này và sự hướng dẫn của thầy Phạm Văn Hà, em đã chọn đề tài Xây dựng trò chơi bằng Unity để làm đồ án chuyên ngành. Do mới tiếp xúc với phần mềm nên còn nhiều thiếu sót, mong thầy cô góp ý để em có sự tiến bộ hơn.

1. Lý do chọn đề tài

Trong thời đại công nghệ phát triển vượt bậc thì nhu cầu của con người về dịch vụ và giải trí cũng tăng cao. Đặc biệt là ngành giải trí, chỉ cần một dự án game có đầu tư, có chất xám và đánh được vào nhu cầu giải trí cần thiết của đại đa số mọi người thì số tiền thu được từ dự án game đây là một con số khổng lồ lên tới hàng trăm triệu, thậm chí là vài chục tỷ. Đơn cử là tựa game Flappy Bird của Nguyễn Hà Đông, một nhà thiết kế game người Việt Nam, nhờ tựa game Flappy Bird mà Nguyễn Hà Đông một ngày có thể kiếm được vài tỷ đồng. Qua đó, thiết kế game là một công việc rất thú vị, tương tác trực tiếp với người phát triển game, làm cho những đoạn code không còn khô khan mà còn đem lại trải nghiệm thú vị cho người trải nghiệm. Đồng thời qua việc làm game có thể giúp chúng ta có thể trau dồi thêm kiến thức về cách sử dụng các công cụ làm game hay cũng như những ngôn ngữ để làm game như Unity, C#, ...

Ứng dụng Game được thực hiện dựa trên hai mục đích sau:

- ✓ Thứ nhất là nghiên cứu các công nghệ lập trình như: lập trình ứng dụng với Visual studio
- ✓ Thứ hai là phát triển ứng dụng có tính thực tiễn cao, có khả năng triển khai ứng dụng vào thực tế, giúp những người có nhu cầu giải trí sau những giờ làm việc mệt mỏi.
- ✓ Thứ ba là để học hỏi kỹ năng làm game, hiểu các quy trình vận hành của game, từ đó trau dồi kỹ năng của bản thân để sẵn sàng bước chân vào làm việc lại các công ty.

Em hi vọng dựa trên nền tảng lý thuyết đã được thầy cô truyền thụ lại, kết quả với sự tìm hiểu công nghệ của cá nhân, báo cáo đồ án chuyên ngành của em sẽ đạt được mục đích mong đợi.

2. Mục tiêu của đề tài

- Làm chủ công nghệ Unity Engine.
- Phân tích, thiết kế, phát triển một ứng dụng Game 2D hoàn chỉnh chạy trên hệ điều hành Windows
- Đáp ứng các yêu cầu cơ bản của một ứng dụng Game
- Xây dựng đồ án đúng quy định

3. Đối tượng và phạm vi nghiên cứu

❖ Đối tượng nghiên cứu

- Quy trình thiết lập đồ họa và viết chương trình liên kết giữa unity và C#.
- Nguyên lý hoạt động của các trình biên dịch của các ngôn ngữ lập trình.

❖ Phạm vi nghiên cứu

- Sử dụng phần mềm Unity.
- Sử dụng ngôn ngữ C# để viết code.

4. Hướng tiếp cận và phương pháp nghiên cứu

❖ Cách tiếp cận

Từ nhu cầu giải trí hằng ngày của con người. Cùng với đó, sự bùng nổ của internet, mạng xã hội giúp con người tiếp nhận được nhiều thông tin hơn. Ngoài xem các chương trình giải trí, thì chơi game cũng là một phần rất được quan tâm trong thời gian hiện nay.

❖ Phương pháp nghiên cứu

Nghiên cứu các lý thuyết về Unity.

Phương pháp nghiên cứu lý luận: Nghiên cứu tìm hiểu các tài liệu nghiên cứu, giáo trình, báo viết và các nguồn thông tin có chọn lọc trên Internet có liên quan, cơ sở là ngôn ngữ lập trình C#.

Phương pháp phân tích và tổng kết kinh nghiệm: Từ việc nghiên cứu tài liệu, giáo trình và tìm hiểu các nguồn thông tin, rút ra được các thuật toán và các kiến thức liên quan.

Phương pháp lấy ý kiến chuyên gia: Lấy ý kiến giảng viên trực tiếp hướng dẫn và các chuyên gia khác để có thể hoàn thiện về mặt nội dung cũng như hình thức của đề tài.

5. Nội dung nghiên cứu

- Tìm hiểu và sử dụng ngôn ngữ lập trình C# cũng như Unity Framework trong việc xây dựng 1 tựa game 2D
- Thiết kế hệ thống game như giao diện, nhân vật, các hành động của nhân vật, các màn chơi ,...
- Tiến hành nghiên cứu chạy game trên Unity
- Tựa game đáp ứng các yêu cầu như:
 - Game đa dạng câu hỏi
 - Có nhiều màn chơi
 - Bối cảnh đa dạng
 - Giao diện thân thiện

6. Kết quả dự kiến

- Bản thiết kế game: Thiết kế nhân vật, thiết kế kịch bản, thiết kế giao diện game
- Ứng dụng đáp ứng các yêu cầu sau:
 - Game gồm tối thiểu 3 câu hỏi trong 1 màn chơi với trên 10 levels
 - Nhân vật game gồm: Knight và quái vật
 - Câu hỏi đa dạng
 - Giao diện trò chơi thân thiện, dễ sử dụng
- Báo cáo tổng kết đề tài
- Các tài liệu liên quan đến đồ án tốt nghiệp

7. Kịch bản game

Knight Battle Maths là trò chơi giải toán cho những thiên tài thực sự, nó bắt đầu rất dễ dàng nhưng thử thách được tăng lên nhanh chóng, chơi Knight Battle Maths để luyện tập trí não và sự nhanh nhẹn, hoàn thành các thử thách bạn sẽ trở thành người phá đảo.

Cách chơi rất đơn giản, với mỗi màn chơi, bạn chỉ cần trả lời đúng 3 câu hỏi và giữ cho vị hiệp sĩ của chúng ta vẫn còn máu.

8. Bố cục của đề tài

Bao gồm 1 phần mở đầu và 4 chương chính:

Mở đầu: Những nội dung gồm lý do chọn đề tài nghiên cứu, mục tiêu nghiên cứu của đề tài, đối tượng và phạm vi nghiên cứu, hướng tiếp cận và phương pháp nghiên cứu, nội dung nghiên cứu, kết quả dự kiến, kịch bản game được trình bày trong phần này nhằm cung cấp 1 góc nhìn tổng quát đề tài nghiên cứu.

Chương 1. Tổng quan về Unity engine: Nội dung của chương này cung cấp một góc nhìn tổng quát về unity engine, về các tính năng và ưu điểm của Unity.

Chương 2. Thiết kế hệ thống game: Nội dung của chương là những thiết kế đã sử dụng trong tựa game 2D được xây dựng.

Chương 3. Xây dựng game và kết quả đạt được: Chương này trình bày quá trình xây dựng và kết quả đạt được của đồ án trong thời gian qua. Cùng với việc kiểm thử các chức năng và hệ thống của game.

Chương 4. Kiểm thử hệ thống: Chương này sẽ gồm kế hoạch kiểm thử, ca kiểm thử và báo cáo kiểm thử

CHƯƠNG 1. TỔNG QUAN VỀ ENGINE UNITY VÀ KHÔNG GIAN 2D

1.1. Tổng quan về Unity

1.1.1 Unity là gì?

Đã qua khoảng thời gian mà tất cả mọi người phải làm game trên nền Flash cơ bản. Unity đã thổi một làn gió mới trong ngành làm game, Unity mang lại sức mạnh kỳ diệu giúp cho nhân vật mà ta muốn thể hiện trở nên sống động hơn trong không gian 3 chiều đầy huyền ảo. Công nghệ này tạo ra một bước đột phá mới về sự khác biệt trong cộng nghệ làm game hiện nay, mang đến cho người chơi một trải nghiệm khác, hiện tại công nghệ này đã được áp dụng vào game Việt Nam. Đã có rất nhiều game của Việt Nam sử dụng Unity và thành công cùng với những tựa game 2D, 3D.

Unity là một nền tảng phát triển trò chơi đa nền tảng được phát triển bởi Unity Technologies, ra đời với mục đích cung cấp cho các nhà phát triển một công cụ mạnh mẽ và linh hoạt để tạo ra các trò chơi và ứng dụng tương tác. Với Unity, các nhà phát triển có thể xây dựng trò chơi và ứng dụng chạy trên nhiều hệ điều hành và thiết bị khác nhau, bao gồm cả máy tính để bàn (Windows, macOS), thiết bị di động (Android, iOS), và các hệ máy chơi game như PlayStation, Xbox, và Nintendo Switch. Ngoài ra, Unity còn hỗ trợ phát triển cho các nền tảng thực tế ảo (VR) và thực tế tăng cường (AR).

1.1.2 Quá trình phát triển của Unity

Lần đầu tiên nó được công bố chạy trên hệ điều hành OS X, tại Apple's Worldwide Developers Conference vào năm 2005, đến nay đã mở rộng 27 nền tảng.

Ra mắt đầu tiên vào năm 2005 tại sự kiện Apple's Worldwide Developer Conference bởi nhà sáng lập David Helgason, trải qua nhiều năm phát triển, Unity đã có version 5.5 hoàn thiện hơn về rất nhiều mặt. Tháng 5-2012 theo cuộc khảo sát Game Developer Megazine được công nhận là Game engine tốt nhất cho mobile. Năm 2014 Unity thắng giải “Best Engine” tại giải UK's annual Develop Industry Excellence. Phiên bản mới nhất hiện nay là 2019.2 với nhiều tính năng nổi trội

1.1.3 Lịch sử hình thành

Phần lõi của Unity ban đầu được viết bởi Joachim Ante vào năm 2001. Sau đó công ty được hình thành vào năm 2005 và bắt đầu với phiên bản 1.0. Đến năm 2007, Unity được nâng lên phiên bản 2.0. Unity bắt đầu hỗ trợ iPhone vào năm 2008. Vào tháng 6/2010, unity chính thức hỗ trợ Android và cho ra đời phiên bản 3.0 có hỗ trợ Android vào tháng 9/2010. Có thể thấy tốc độ phát triển của Unity khá nhanh. Vào năm 2011, Unity đã thực hiện một bước tiến quan trọng bằng việc tích hợp với các công cụ phát triển khác, đáng chú ý nhất là Adobe Flash. Điều này cho phép các nhà phát triển dễ dàng xuất bản trò chơi và ứng dụng Unity trực tiếp lên nền tảng Flash, mở rộng phạm vi tiếp cận người dùng và tăng tính linh hoạt trong việc phát triển nội dung đa phương tiện. Đến năm 2014, Unity đã ra mắt phiên bản Unity 5, mang lại nhiều cải tiến lớn về đồ họa và hiệu suất. Phiên bản này giới thiệu các công cụ mới như Enlighten (một hệ thống chiếu sáng thời gian thực) và WebGL, cho phép trò chơi chạy trực tiếp trên trình duyệt mà không cần plugin. Những cải tiến này không chỉ nâng cao chất lượng hình ảnh mà còn cải thiện hiệu suất tổng thể, làm cho Unity trở nên hấp dẫn hơn đối với các nhà phát triển.

Đến năm 2021, Unity tiếp tục mở rộng và cải tiến với sự tập trung mạnh mẽ vào các công nghệ thực tế ảo (VR) và thực tế tăng cường (AR). Các tính năng mới giúp các nhà phát triển dễ dàng tạo ra các trải nghiệm VR và AR chất lượng cao, phù hợp với nhu cầu ngày càng tăng của thị trường. Hơn nữa, Unity đã mở rộng hỗ trợ cho các nền tảng mới như Google Stadia, dịch vụ chơi game dựa trên đám mây của Google. Sự hỗ trợ này cho phép các nhà phát triển sử dụng Unity để tạo ra các trò chơi có thể chơi trên nhiều thiết bị mà không cần phần cứng mạnh mẽ, thúc đẩy sự phát triển và phân phối trò chơi trên đám mây. Những bước tiến này không chỉ củng cố vị thế của Unity trong ngành công nghiệp game mà còn mở ra nhiều cơ hội mới cho sự sáng tạo và đổi mới.

1.1.4 Tính năng của Engine Unity

Môi trường phát triển được tích hợp với tính năng kế thừa, khả năng chỉnh sửa đồ họa, chức năng kiểm tra chi tiết, và đặc biệt tính năng xem trước game ngay trong lúc xây dựng.

Triển khai được trên nhiều nền tảng:

- Chương trình độc lập trên Windows và Mac OS.
- Trên web, thông qua Unity Web Player plugin cho internet Explorer, Firefox, Safari, Opera, Chrome, cho cả Windows và Mac OS.
- Trên Mac OS Dashboard widget.
- Cho Nintendo wii.
- Cho iPhone, iPad application.

Tài nguyên được tải vào trong unity và tự động cập nhật nếu tài nguyên có sự thay đổi. Unity hỗ trợ các kiểu định dạng từ 3DS Max, Maya, Blender. Graphics engine sử dụng Direct3D, OpenGL, OpenGL

ES, và các API khác trên Wii. Hỗ trợ bump mapping, reflection mapping, ...

Unity Asset Server: Đây là một tính năng khá mới của unity, theo đó unity sẽ cung cấp một hệ thống quản lý theo dạng phiên bản cho tất cả asset và cả script. Đây là một kho chứa các tài nguyên cần thiết cho việc làm game. Khi import cũng như sửa chữa, trạng thái của asset ngay lập tức được cập nhật. Server chạy trên database opensource PostgreSQL và có thể truy cập trên cả

Mac lẫn Windows, Linux. Asset Server đòi hỏi một khoản phí phụ trội là \$499 cho mỗi bản copy unity, và một license Unity Pro.

1.1.5 Một số thống kê về Unity

Sự thay đổi trong cách thức chơi game của người chơi hay nói cách khác là xu hướng mọi người tập trung vào game trên di động nhiều hơn. 2,4 tỉ thiết bị di động đã từng tải ít nhất 1 ứng dụng bởi unity.

Số lượng người dùng (gamer) của Unity đạt tới con số 770 triệu, trong khi đó số người thường xuyên sử dụng Twitter là 310 triệu người.

Tính đến quý 3 năm 2016 đã có 5 tỉ lượt download game và ứng dụng được phát triển bởi Unity.

Trong top 1000 game Mobile miễn phí thì số lượng game tạo ra bởi Unity chiếm tới 34%.

1.1.6 Ưu điểm của Unity và kiến trúc tổng quan

a. Ưu điểm của Unity:

Unity có một cộng đồng rất lớn về asset và plugin – trong đó có rất nhiều resources free và có nhiều thứ rất đáng bỏ tiền.

Unity có bộ công cụ rất trực quan và editor có thể mở rộng bằng plugins.

Unity hỗ trợ rất nhiều định dạng asset khác nhau và có thể tự động chuyển đổi đến định dạng phù hợp nhất với nền tảng thích hợp.

Unity hỗ trợ nhiều nền tảng: di động, desktop, web và console.

Việc triển khai đến các nền tảng khác nhau cũng khá dễ quản lý.

Bạn có thể dễ dàng xây dựng một game 3D mà không cần cấu hình quá phức tạp.

Unity bản free có hầu hết những tính năng quan trọng nhất.

Unity bản trả phí phù hợp với các developer chuyên nghiệp.

b. Kiến trúc:

Engine Unity hỗ trợ cho chúng ta UnityAPI để viết script game. UnityAPI là API lập trình game trong Unity rất mạnh. UnityAPI chứa các đối tượng và phương thức hỗ trợ hầu hết các đối tượng và các loại thành phần trong Unity.

Trong một scene thường có nhiều đối tượng game. Mỗi đối tượng này có thể có hoặc không có đoạn script nào gắn lên đó. Nếu muốn gắn script vào đối tượng, ta bắt buộc phải kế thừa class đó từ lớp MonoBehaviour của UnityAPI và tên class phải trùng với tên file script. Mỗi script khi gắn lên đối tượng game đều được đối tượng game xem như một thành phần bên trong và được cấp phát vùng nhớ khi chạy game.

Bên trong UnityAPI có chứa rất nhiều lớp hỗ trợ lập trình game, trong đó có một số lớp quan trọng như:

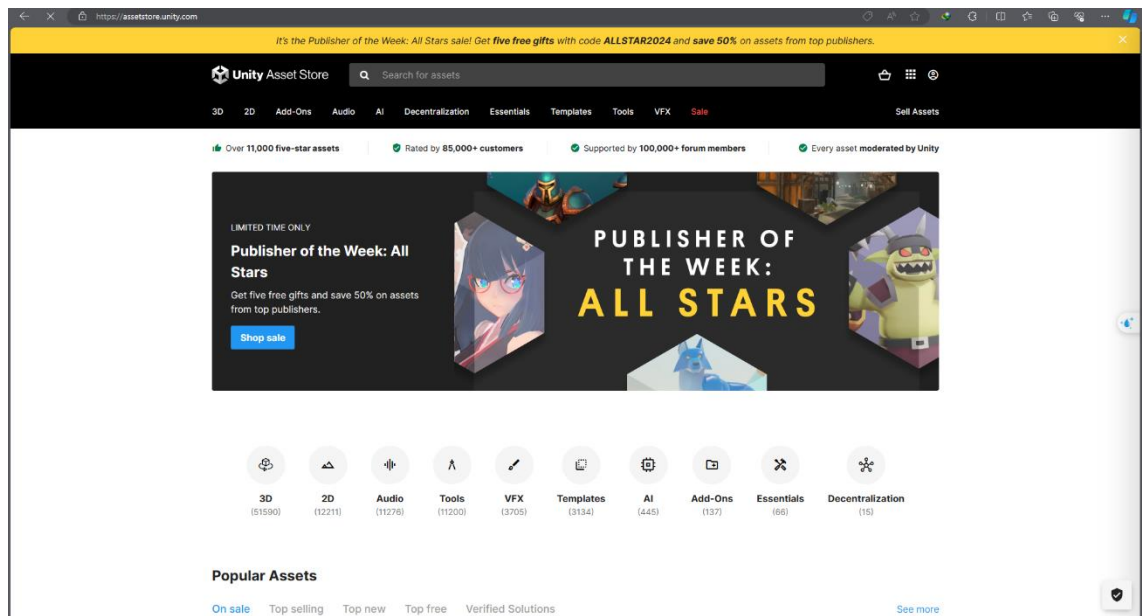
- **MonoBehaviour**: tất cả các script muốn gắn vào một đối tượng game bắt buộc phải kế thừa từ lớp này.
- **GameObject**: lớp cha của tất cả các thực thể trong scene.
- **Component**: lớp cha của tất cả các thành phần có thể gắn vào đối tượng.

1.2. Các khái niệm trong Unity

1.2.1 Assets

Assets là tài nguyên xây dựng nên một dự án trên Unity. Những tài nguyên có thể là hình ảnh, âm thanh, mô hình 2D 3D, chất liệu (material), texture vv hoặc cả một project hoàn chỉnh

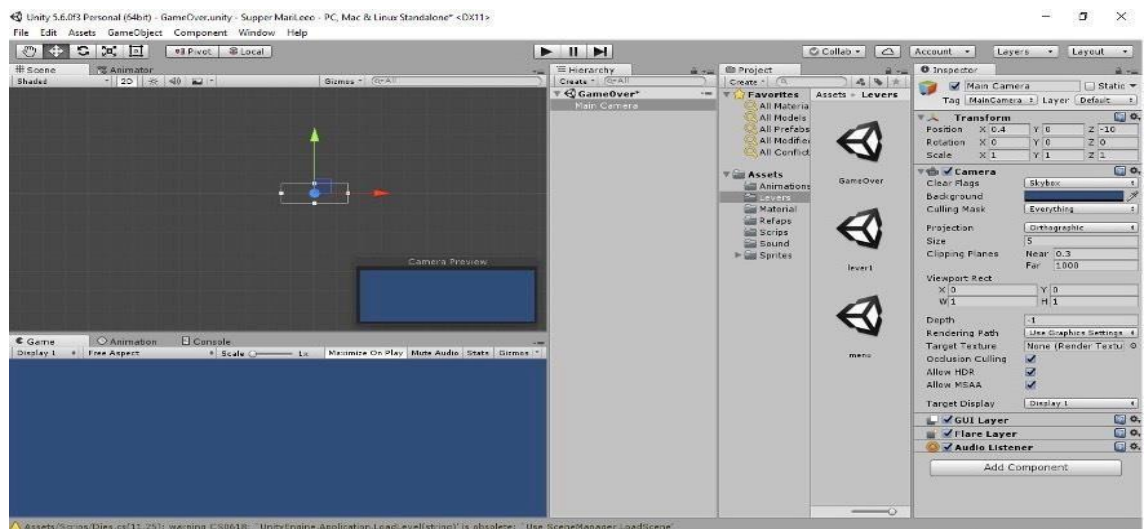
Các asset do chính những nhà phát triển game tạo ra và có thể được download miễn phí hoặc trả phí trên Unity Asset Store. Đây là một trong những tính năng rất hay của Unity. Các asset này sẽ giúp giảm thiểu rất nhiều thời gian cho việc thiết kế và lập trình game.



Hình 1.1. Hình ảnh Assets Store

1.2.2 Scene

Trong Unity, một cảnh chơi (hoặc một phân đoạn) là những màn chơi riêng biệt, một khu vực trong game hoặc thành phần có trong nội dung của trò chơi (các menu). Các thành phần này được gọi là Scene. Bằng cách tạo ra nhiều Scenes, chúng ta có thể phân phối thời gian và tối ưu tài nguyên, kiểm tra các phân đoạn trong game một cách độc lập.

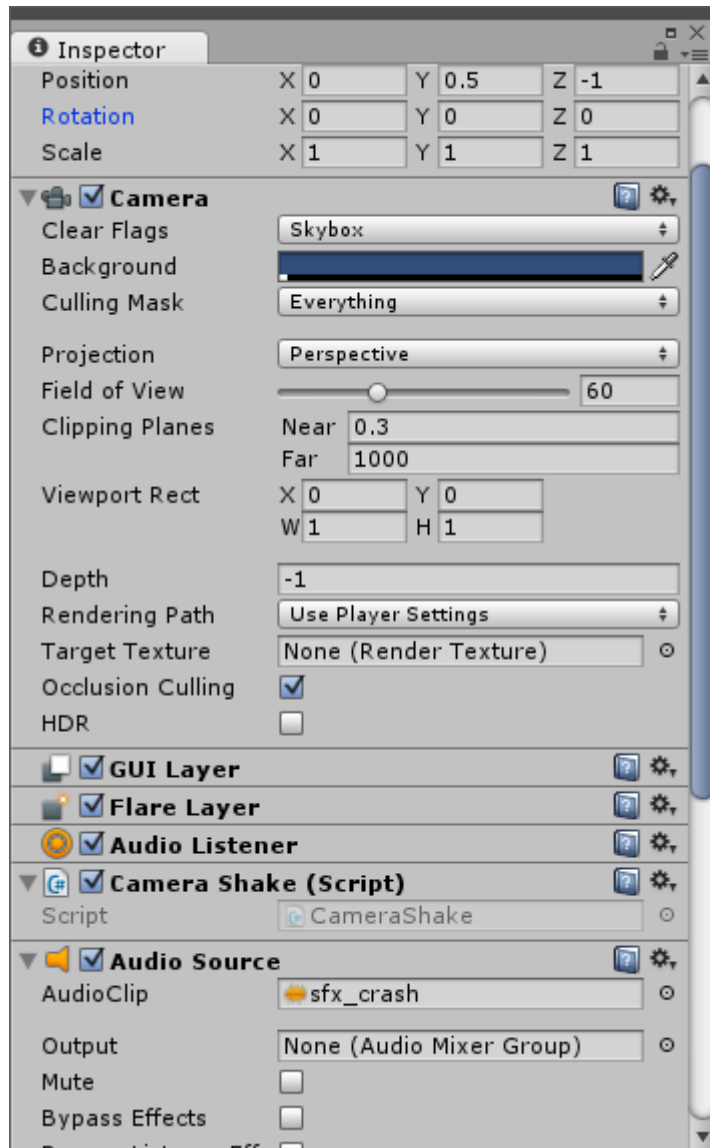


Hình 1.2. Scene trong game

1.2.3 GameObject

Khi Asset được sử dụng trong các Scene, Unity định nghĩa đó là Game Object. Đây là một thuật ngữ thông dụng, đặc biệt trong mảng lập trình. Tất cả các Game Object đều chứa ít nhất một thành phần cơ bản là Transform, lưu trữ thông tin về vị trí, góc xoay và tỉ lệ của Game Object. Thành phần Transform có thể được tùy biến và chỉnh sửa trong quá trình lập trình.

1.2.4 Component



Hình 1.3. Component

Components là các thành phần trong game, bổ sung tính năng cho các Game Object. Mỗi Component có chức năng riêng biệt. Đa phần các Component phụ thuộc vào Transform, vì nó lưu trữ các thông số cơ bản của Game Object. Bản chất của

Game Object là không có gì cả, các đặc tính và khả năng của Game Object nằm hoàn toàn trong các Component. Do đó chúng ta có thể xây dựng nên bất kỳ Game Object nào trong game mà chúng ta có thể tưởng tượng được.

1.2.5 Scripts

Scripts được Unity xem như một Component. Đây là thành phần thiết yếu trong quá trình phát triển game. Bất kỳ một game nào, dù đơn giản nhất đều cần đến Scripts để tương tác với các thao tác của người chơi, hoặc quản lý các sự kiện để thay đổi chiều hướng của game tương ứng với kịch bản game.

Unity cung cấp cho lập trình viên khả năng viết Script bằng các ngôn ngữ: JavaScript, C#. Unity không đòi hỏi lập trình viên phải học cách lập trình trong Unity, nhưng trong nhiều tình huống, chúng ta cần sử dụng Script trong mỗi phần của kịch bản game.

Để viết Script, chúng ta có thể làm việc với một trình biên tập Script độc lập của Unity, hoặc làm việc trên Mono Developer được tích hợp vào Unity trong những phiên bản gần đây. Mono Developer là một IDE khá tốt, cung cấp nhiều chức năng tương tự Visual Studio chúng ta cũng có thể dùng Visual Studio để viết file C# như bình thường. Mã nguồn viết trên Mono Developer sẽ được cập nhật và lưu trữ trong dự án trên Unity

1.2.6 Prefabs

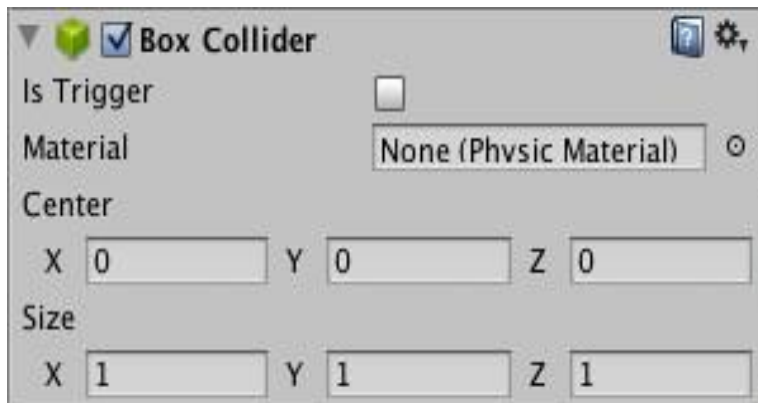
Prefabs thực chất là Game Object được lưu trữ lại để tái sử dụng. Các Game Object được nhân bản từ một prefab sẽ giống nhau hoàn toàn, ngoại trừ thành phần Transform để phân biệt và quản lý được tốt hơn.

Để tạo ra một prefab, ta đơn giản chỉ cần kéo một Game Object vào cửa sổ Project

1.2.7 Collider

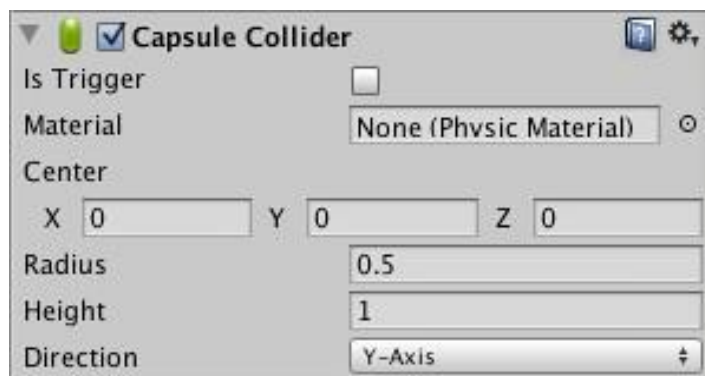
Hệ thống xử lý va chạm bao gồm 2D và 3D. Được chia ra các va chạm của các hình cơ bản:

- Box Collider: Va chạm cho các vật hình hộp chữ nhật với 3D và hình chữ nhật với 2D



Hình 1.4 Box Collider

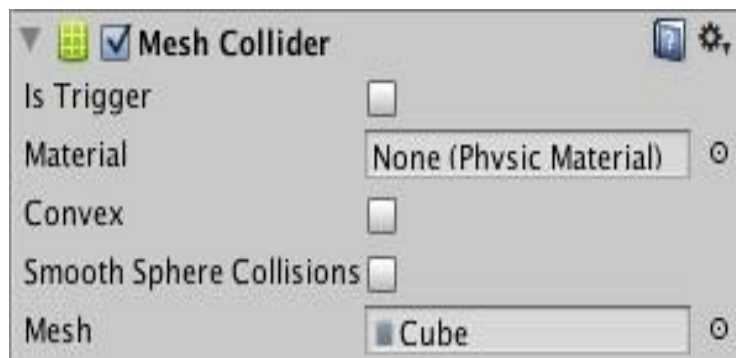
- Is Trigger: dạng true/false, cho phép va chạm có đi xuyên qua không?
- Material: tham chiếu đến physic material.
- Center: điều chỉnh thông số tâm của va chạm theo các trục tương ứng.
- Size: điều chỉnh kích thước theo các trục tương ứng.
- Capsule Collider: Dành cho va chạm khối trụ



Hình 1.5 Capsule Collider

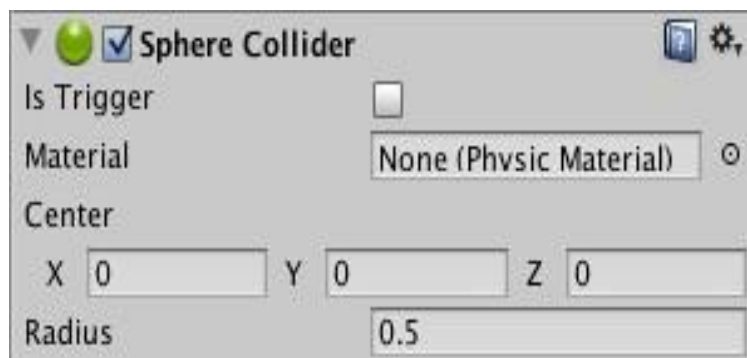
- Is Trigger: dạng true/false, cho phép va chạm có đi xuyên qua không?

- Material: tham chiếu đến physic material.
 - Center: điều chỉnh thông số tâm của va chạm theo các trục tương ứng.
 - radius: điều chỉnh kích thước bán kính.
 - Height: độ cao
 - Direction: xoay hình trụ theo các trục tương ứng.
- Box collider: Va chạm cho các vật hình hộp chữ nhật đối với 3d và hình chữ nhật đối với 2d
- Mesh Collider: Va chạm dạng lưới dành các vật không thể xác định hình thể



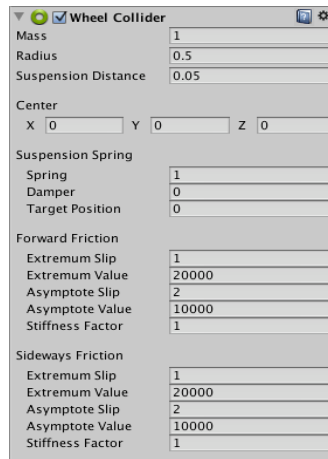
Hình 1.6 Mesh Collider

- Sphere Collider: Va chạm áp dụng cho các vật hình cầu



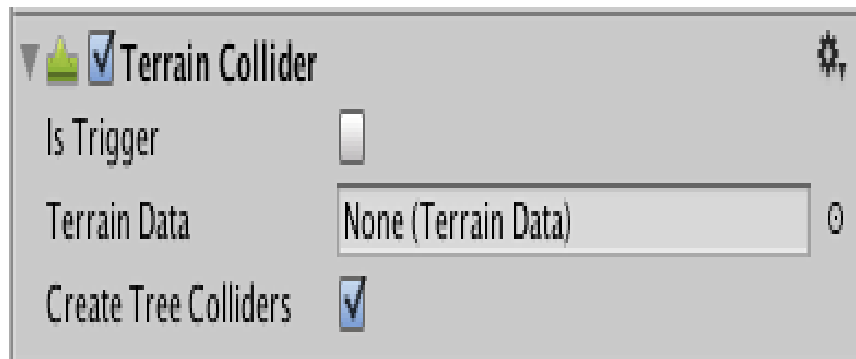
Hình 1.7 Sphere Collider

- Wheel Collider: Dành cho vật thể hình bánh xe. Nó sẽ mô phỏng va chạm giống với các bánh xe



Hình 1.8 Wheel Collider

- Terrain Collider: Dành cho các vật thể địa hình và Terrain Collider sẽ dựa theo địa hình đó



Hình 1.9 Terrain Collider

1.2.8 Rigidbody

Hệ thống mô phỏng vật lý trong game. Cũng chia ra 2D

- Mass: khối lượng (đơn vị tùy ý).
- Drag: Sức cản không khí ảnh hưởng như thế nào với đối tượng khi di chuyển. 0 có nghĩa là không có sức cản không khí, và vô cùng làm cho các đối tượng di chuyển ngay lập tức dừng lại.
- Angular Drag: Sức cản không khí ảnh hưởng đến các đối tượng khi quay từ mô-men xoắn. 0 có nghĩa là không có sức cản không khí. Không thể làm cho vật dừng quay hẳn chỉ bằng cách thiết lập Angular Drag của nó đến vô cùng.

- Use Gravity: Nếu được kích hoạt, các đối tượng bị ảnh hưởng bởi lực hấp dẫn.
- Is Kinematic: Nếu được kích hoạt, các đối tượng sẽ không được thúc đẩy bởi động cơ vật lý.
- Interpolate: giảm xóc.
- Collision detection: Được sử dụng để ngăn chặn các đối tượng chuyển động nhanh qua các đối tượng khác mà không phát hiện va chạm.
- Constraints: Ràng buộc Những hạn chế về chuyển động của Rigidbody

1.2.9 Sprite

Là một hình ảnh 2D của một game object có thể là hình ảnh đầy đủ, hoặc có thể là một bộ phận nào đó. Unity cho phép tùy chỉnh màu sắc, kích thước, độ phân giải của một hình ảnh 2D.

1.2.10 Animator

Trong 2D thì animation là tập một hình ảnh động dựa trên sự thay đổi liên tục của nhiều sprite khác nhau. Trong 3D là một tập hợp các sự thay đổi theo thời gian của đối tượng trong không gian. Mỗi thay đổi là một key frame.-Key Frame hay Frame là một trạng thái của một animation.

- Animator: gồm các thành phần:
- Controller: Bộ điều khiển animation gắn liền với nhân vật này. Nó sẽ quản lý các animation clip, các thông số tốc độ của animation, thứ tự các clip...
- Avatar: thành phần tạo hình ảnh cho object.
- Apply Root Motion: dạng true-false, cho phép thiết lập animation có di chuyển theo không gian đã được tạo khi cấu hình animation.

- Animate Physics: dạng true-false, khi được chọn, các hình ảnh trong animation sẽ có thể tương tác vật lý với nhau.
- Culling mode: chọn chế độ cho hình ảnh động.
- Trong 2D thì animation là tập một hình ảnh động dựa trên sự thay đổi liên tục của nhiều sprite khác nhau. Trong 3d là một tập hợp các sự thay đổi theo thời gian của đối tượng trong không gian. Mỗi thay đổi là một key frame.-Key Frame hay Frame là một trạng thái của một animation.
- Animator: gồm các thành phần:
 - Controller: Bộ điều khiển animation gắn liền với nhân vật này. Nó sẽ quản lý các animation clip, các thông số tốc độ của animation, thứ tự các clip...
 - Avatar: thành phần tạo hình ảnh cho object.
 - Apply Root Motion: dạng true-false, cho phép thiết lập animation có di chuyển theo không gian đã được tạo khi cấu hình animation.
- Animate Physics: dạng true-false, khi được chọn, các hình ảnh trong animation sẽ có thể tương tác vật lý với nhau.
- Culling mode: chọn chế độ cho hình ảnh động.

1.2.11 Audio Source

Âm thanh trong game. Gồm cả âm thanh 2D và 3D.

- Audio clip: tham chiếu đến file âm thanh.
- Mute: chơi ở chế độ như tắt tiếng.
- Bypass effects: bộ lọc hiệu ứng áp dụng cho các nguồn âm thanh.
- Bypass listener effects: Điều này là để nhanh chóng chuyển tắt cả các hiệu ứng Listener on / off.
- Pass Reverb Zones: Điều này là để nhanh chóng chuyển tắt cả các khu Reverb on / off.

- Play on awake: Nếu được kích hoạt, âm thanh sẽ bắt đầu chơi lúc cảnh ra mắt. Nếu vô hiệu hóa, cần phải bắt đầu nó bằng cách sử dụng lệnh Play () từ kịch bản script.
- Loop: Kích hoạt tính năng này để làm cho Clip âm thanh lặp lại.
- Pitch: Xác định ưu tiên của nguồn âm thanh này trong số tất cả những nguồn âm cùng tồn tại trong bối cảnh đó. (Ưu tiên: 0 = quan trọng nhất, 256 = ít quan trọng nhất Mặc định = 128).
- 3D Sound Setting: Cài đặt được áp dụng cho các nguồn âm thanh nếu Audio Clip là một âm thanh 3D.
- 2D Sound Setting: Cài đặt được áp dụng cho các nguồn âm thanh nếu Audio Clip là một âm thanh 2D.

1.2.12 Camera

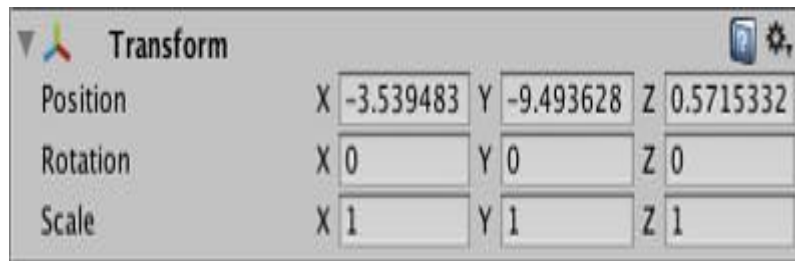
Là một game object đặc biệt trong scene, dùng để xác định tầm nhìn, quan sát các đối tượng khác trong game. Bao gồm các thuộc tính:

- Clear flags: Xác định phần nào của màn hình sẽ bị xóa. Đây là tiện dụng khi sử dụng nhiều máy ảnh để vẽ các yếu tố trò chơi khác nhau.
- Background: Màu áp dụng cho các màn hình còn lại sau khi tất cả các yếu tố trong quan điểm đã được rút ra và không có skybox.
- Culling Mask: Bao gồm hoặc bỏ qua lớp của các đối tượng được đưa ra bởi các Camera.
- Projection: khả năng của máy ảnh để mô phỏng góc nhìn.
- Field of view (thuộc tính chỉ xuất hiện khi chọn Perspective trong mục Projection): Chiều rộng của góc nhìn của Camera, đo bằng độ dọc theo trục Y.

- HDR: Cho phép High Dynamic Range dựng hình cho camera này.

1.2.13 Transform

Transform: quản lý object trong không gian ba chiều, theo ba thông số:

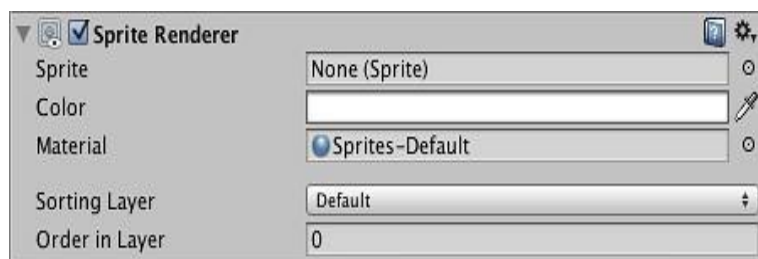


Hình 1.10 Transform

- Position: quản lý vị trí hiện tại của object.
- Rotation: quản lý các thông số quay của object theo các trục x, y, z.
- Scale: quản lý các thông số phóng to, thu nhỏ theo các trục x, y, z.

1.2.14 Renderer

Các SpriteRenderer thành phần cho phép bạn hiển thị hình ảnh như Sprites để sử dụng trong cả hai cảnh 2D và 3D.



Hình 1.11 Renderer

- Sprite: Các đối tượng Sprite để render. Đối tượng Sprite có thể được tạo ra từ textures bằng cách sử dụng các thiết lập Sprite.
- Color: Vertex màu.

- Material: Chất liệu được sử dụng để làm sprite.
- Sorting Layer: Các layer được sử dụng để xác định các ưu tiên của sprite này trong khi hiển thị.
- Order in Layer: Các ưu tiên của sprite trong layer của nó. Con số thấp hơn được kết xuất đầu tiên và con số tiếp theo phủ bên dưới.

1.2.15 Cửa sổ làm việc

a. Project:

Mỗi Project của Unity đều chứa một thư mục Assets. Nội dung của thư mục này được hiển thị trong Project View. Đây là nơi chứa tất cả các assets để tạo Game như Scenes, Script, 3D models, Textures, Audio, Prefabs. Chúng ta không nên di chuyển assets bằng cách sử dụng Window Explorer (hay Finder trong MAC), vì nó sẽ phá vỡ cấu trúc siêu dữ liệu của assets với nhau, và luôn sử dụng Project View để cấu trúc assets.

Để thêm assets vào project, chúng ta có thể kéo thả bất kỳ file nào vào trong project view hoặc vào công cụ Assets – Import new Assets (Click chuột phải vào Project View). Scenes cũng được lưu trữ trong Project view, và đây là một level độc lập mang tính cá nhân. Chúng ta dễ dàng tạo một assets game trong Unity bằng cách chọn hình tam giác nhỏ nằm bên phải Create trong cửa sổ Project hoặc click chuột phải trong Project View, sau đó chọn assets tương ứng.

b. Hierarchy:

Trong Hierarchy chứa các GameObject hiện thời, một số có thể trở trực tiếp tới những file assets như 3D models, một số khác đại diện cho Prefabs – những đối tượng đã được tùy biến, dùng làm các công việc khác nhau sau này trong Game. Chúng ta có thể chọn và parenting Object trong Hierarchy. Một Object có thể được thêm vào

hay loại bỏ trong scene và có thể thấy nó mất đi hay xuất hiện trong Hierarchy.

c. Parenting:

Tức là thư mục chứa hay thư mục gốc, bất kỳ một game object nào muốn là đối tượng con(child) thì ta chỉ việc kéo thả đối tượng đó vào trong đối tượng dự tính làm Parenting trong Hierarchy và nó sẽ kế thừa chuyển động và quay của parenting.

d. Toolbar:

Toolbar chứa 5 loại điều khiển cơ bản, mỗi loại giữ một vai trò quan trọng trong Editor:

- Transform Tool: được dùng với Scene view, như quay trái , phải, lên trên, xuống dưới, phóng to thu nhỏ đối tượng.
- Transform Gizmo Toggles: dùng cho việc thể hiện Scene view.
- Play/Pause/Step Buttons: dùng cho view game, chý game ngay trong Editor để kiểm tra.
- Layer Drop-down kiểm soát đối tượng nào đang được thực hiện trong Scene view
- Layout Drop-down kiểm soát sự sắp xếp của các Views.

e. Scene View

Là nơi Design Game, đối tượng Maneuvering và Importanting trong Scene view (chuyển động và điều khiển) là hai trong số các chức năng quan trọng của Unity, ở góc bên phải của Scene là Scene Gizmo, nó thể hiện hướng nhìn trong không gian của camera trong Scene View hiện thời, cho phép thay đổi góc nhìn trực quan và nhanh chóng.

Click lên các nhánh hình nón để chuyển qua các góc nhìn khác nhau có thể xem ở chế độ Isometric Model (tức ở dạng mặt cắt hai

chiều), để chuyển qua chế độ 3D, click vào hình vuông ở giữa hay giữ phím Shift + Click để chuyển đổi chế độ nhìn.

f. Game View

Game View được rendered từ những Camera trong Game. Đó là những gì được nhìn thấy khi hoàn tất, khi Game được xuất bản. Chúng ta sẽ cần ít nhất là một hoặc nhiều hơn số lượng các Camera để quyết định những gì mà người chơi sẽ nhìn thấy khi họ chơi Game.

g. Play Mode

Sử dụng những nút trên Toolbar để điều khiển Editor Play Mode, và xem trước Game sẽ như thế nào khi chơi. Trong chế Play, mọi giá trị thay đổi sẽ được lưu tạm, và bị xóa khi thoát khỏi chế độ play.

h. Inspector

Inspector sẽ hiển thị mọi thông tin về đối tượng đang làm việc một cách chi tiết, kể cả những Components được đính kèm và những thuộc tính của nó. Tại đây ta có thể điều chỉnh, thiết lập mọi thông số chức năng của những mối liên kết GameObject-Component.

Mọi thuộc tính thể hiện trong Inspector đều có thể được tùy biến một cách trực tiếp. Ngay cả với những biến trong script cũng có thể được hiệu chỉnh mà không cần xem mã. Trong script, nếu chúng ta định nghĩa một giá trị là public cho một kiểu đối tượng (như GameObject hay Transform), ta có thể drag-drop một GameObject hay một Prefab vào trong Inspector để gán giá trị cho nó.

Chúng ta có thể click lên icon hình bánh răng nhỏ bên phải hay click chuột phải lên tên của Component để xuất hiện context menu dành cho những thiết lập của Component.

Inspector cũng sẽ thể hiện mọi thông số Import Setting của assets đang làm việc.

CHƯƠNG 2. THIẾT KẾ HỆ THỐNG GAME

2.1. Ý tưởng và cốt truyện

2.1.1 Ý tưởng


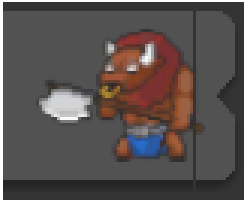
- Tên game: “Knight Maths Battle”
- Thể loại: Casual, Giải đố
- Nền tảng: Unity 2D
- Mục tiêu chính: Người chơi sẽ vào vai là một chàng kỵ sĩ, người chơi sẽ phải trả lời các câu hỏi về toán học để đánh bại quái vật và băng qua các khu vực để có thể quay lại lâu đài

2.1.2 Cốt truyện

Cốt truyện của game xoay quanh hành trình của nhân vật chính, một chiến binh dũng cảm tiêu diệt thế lực hắc ám đang lan tràn. Chàng kỵ sĩ dũng cảm, sau khi nhận được tin khẩn về cuộc tấn công của quái vật vào lâu đài của mình, lập tức lên đường trở về để bảo vệ quê hương. Hành trình của chàng bắt đầu từ một khu rừng rậm rạp, nơi ẩn chứa vô số hiểm nguy và thử thách. Trong khu rừng, chàng phải đối mặt với những sinh vật hung dữ, giải quyết các câu đố phức tạp, và vượt qua những chướng ngại vật đầy thử thách. Mỗi bước đi trong rừng đều mang đến cho chàng những bài học quý giá và giúp chàng trở nên mạnh mẽ hơn. Cuối cùng, sau một hành trình dài và gian nan, chàng kỵ sĩ cũng đã vượt qua khu rừng, sẵn sàng đối đầu với kẻ thù và bảo vệ lâu đài của mình, mang lại hòa bình cho vương quốc.

2.2. Thiết kế nhân vật và bối cảnh


2.2.1 Các nhân vật trong trò chơi


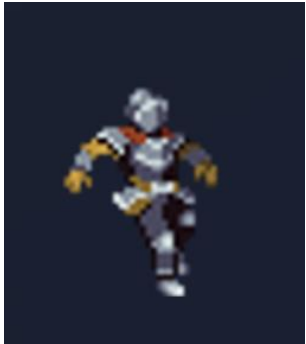
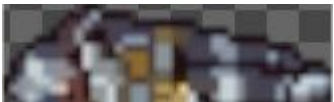
| Đối tượng | Đặc trưng | Thông số | Hình ảnh |
|------------|--|---|--|
| Người chơi | Chàng kĩ sĩ mạnh mẽ, cùng với bộ áo giáp sắt | Nhân vật sẽ có 3 máu, khi trả lời sai 1 câu hỏi sẽ mất đi 1 máu |  |
| Quái vật | Một quái vật hình trâu, cầm trên tay chiếc rìu | Sẽ dần dần tiến lại gần người chơi, chỉ bị tiêu diệt khi mà người chơi trả lời đúng câu hỏi |  |

Bảng 2.1 Các nhân vật trong trò chơi

2.2.2 Các hành động trong trò chơi




- **Người chơi:**

| Đối tượng | Tên hành động | Miêu tả hành động | Hình ảnh minh họa |
|------------|-----------------|---|---|
| Người chơi | Chạy, di chuyển | Người chơi sẽ di chuyển đến nơi mà quái vật đang đứng |  |

| | | | |
|------------|---------|---|--|
| Người chơi | Chém | Khi người chơi đã di chuyển tới vị trí chỉ định, người chơi sẽ thực hiện động tác chém. Người chơi sẽ vung thanh kiếm chém vào quái vật |  |
| Người chơi | Bị đánh | Sau khi người chơi bị chém bởi quái vật, người chơi sẽ thực hiện động tác lùi về sau như biểu hiện việc mình bị đánh |  |
| Người chơi | Chết | Sau khi người chơi bị chết vì hết 3 thanh máu của mình. Người chơi sẽ từ từ ngã xuống đất |  |

Bảng 2.2 Các hành động của người chơi

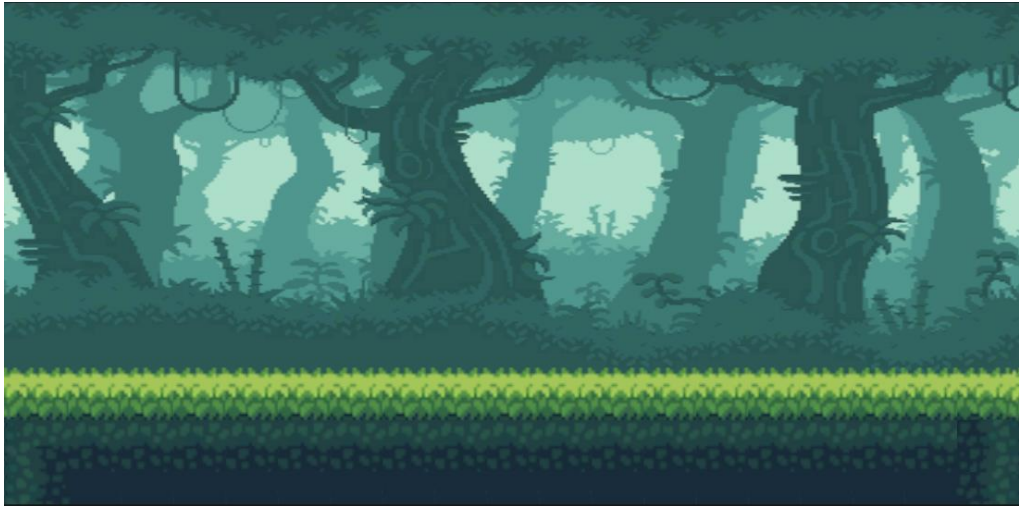
- **Quái vật:**

| Đối tượng | Tên hành động | Miêu tả hành động | Hình ảnh minh họa |
|-----------|-----------------|--|---|
| Quái vật | Chạy, di chuyển | Quái vật sẽ đi từng bước tới vị trí của người chơi đang đứng |  |
| Quái vật | Chém | Quái vật sẽ thực hiện hành động vung chiếc rìu của mình về phía người chơi |  |
| Quái vật | Chết | Sau khi bị người chơi chém trúng, quái vật sẽ từ từ ngã xuống mặt đất |  |

Bảng 2.3 Các hành động của quái vật trong trò chơi

2.3. Thiết kế bối cảnh

Bối cảnh trong game được diễn ra ở trong 1 khu rừng khi mà nhân vật chính là chàng kị sĩ phải băng qua khu rừng để có thể trở về toà lâu đài của mình.



Hình 2.1 Bối cảnh game

2.4. Thiết kế cơ chế của game:

- Nhiệm vụ chính: Người chơi sẽ phải lần lượt trả lời qua 3 câu hỏi mà mỗi màn chơi đặt ra.
- Câu hỏi trong mỗi màn chơi: Trong mỗi câu hỏi đều có giới hạn thời gian nếu như người chơi không thể trả lời câu hỏi hoặc khi hết thời gian trả lời của câu hỏi người chơi sẽ bị mất đi 1 máu. Số lượng máu của người chơi là 3 và sẽ hồi lại khi người chơi qua màn.
- Câu trả lời: Trong mỗi câu hỏi người chơi sẽ có 4 câu trả lời và sẽ chỉ có 1 câu trả lời đúng
- Các trường hợp khi người chơi trả lời câu hỏi: Nếu như người chơi trả lời sai quái vật sẽ tiến tới vị trí của người chơi nhanh hơn và khi quái vật tới vị trí của người chơi, quái vật sẽ thực hiện hành động chém. Còn trong trường hợp người chơi trả lời đúng thì quái vật sẽ đứng im và kị sĩ sẽ tiến lại gần vị trí của quái vật để tiêu diệt quái vật.

- Mỗi 1 câu hỏi sẽ có 15 giây để cho người chơi có thể suy nghĩ trả lời, mỗi khi trả lời đúng hoặc hết thời gian sẽ chuyển qua câu hỏi tiếp theo và làm mới lại thời gian

- Số sao khi người chơi hoàn thành màn chơi: Số sao sẽ tương ứng với số máu của người chơi còn lại khi hoàn thành màn chơi

+ 1 sao: Tương ứng với 1 máu

+ 2 sao: Tương ứng với 2 máu

+ 3 sao: Tương ứng với 3 máu

- Hoàn thành màn chơi: Màn chơi chỉ được hoàn thành khi mà người chơi trả lời hết 3 câu hỏi của màn chơi mà vẫn còn máu trên màn hình.

2.5. Thiết kế nội dung

- Chủ đề chính:

Chủ đề chính của câu hỏi sẽ là các câu hỏi các hỏi về toán học dành cho các em học sinh từ lớp 1 đến lớp 3

- Các dạng câu hỏi:



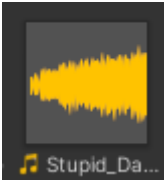
| STT | Dạng câu hỏi | Miêu tả | Ví dụ minh hoạ |
|-----|----------------|--|---------------------|
| 1 | Giải phép tính | Là kết quả các phép tính nhân, chia, cộng, trừ cơ bản | $3 + 5 = ?$ |
| 2 | Tìm x | Là phần tử của các phép tính nhân, chia, cộng, trừ. Đã cho trước kết quả và bắt đi tìm x | Tìm x: $5 + x = 10$ |

Bảng 2.4 Nội dung câu hỏi

- Câu trả lời : Gồm 4 câu trả lời và chỉ có 1 câu trả lời đúng

2.6. Thiết kế âm thanh

| Tên âm thanh | Mô tả | Hình ảnh |
|--|--|---|
| Hiệu ứng âm thanh khi chọn 1 button | Âm thanh sẽ chạy khi người chơi click vào button |  |
| Hiệu ứng âm thanh khi người chơi chọn sai câu trả lời | Âm thanh sẽ chạy khi người chơi chọn sai câu trả lời |  |
| Hiệu ứng âm thanh khi người chơi chọn đúng câu trả lời | Âm thanh sẽ chạy khi người chơi chọn đúng câu trả lời |  |
| Hiệu ứng âm thanh chém | Âm thanh sẽ phát khi người chơi hoặc quái vật thực hiện hành động chém |  |
| Hiệu ứng âm thanh khi chết | Âm thanh sẽ phát khi người chơi hoặc quái bị chết |  |
| Hiệu ứng âm thanh khi chịu sát thương | Âm thanh sẽ phát khi người chơi chịu sát thương từ quái vật |  |

| | | |
|--|--|---|
| Hiệu ứng âm thanh khi hoàn thành màn chơi | Âm thanh sẽ phát khi hoàn thành màn chơi |  |
| Hiệu ứng âm thanh khi người chơi thua cuộc | Khi người chơi hết máu sẽ có 1 âm thanh vang lên báo hiệu người chơi đã thua |  |
| Âm thanh nền | Âm thanh sẽ chạy xuyên suốt trong trò chơi |  |

Bảng 2.5 Âm thanh trong game

2.7. Thiết kế dữ liệu của game

Bộ dữ liệu của game em sử dụng để ở dạng của file Json và em sử dụng code để import dữ liệu vào game

2.7.1 Bộ dữ liệu của câu hỏi

Dữ liệu câu hỏi là 1 list sẽ gồm có nhiều phần tử và mỗi phần tử dạng như sau:

```
"question": "17 + 5 = ?",
"correctAnswer": "22",
"wrongAnswer": ["2", "20", "23"]
```

Hình 2.2 Minh họa dữ liệu câu hỏi

- “question” là phần tử chứa câu hỏi của màn chơi
- “correctAnswer” là phần tử chứa kết quả đúng của câu hỏi
- “wrongAnswer” là 1 mảng chứa các giá trị của câu trả lời sai

2.7.2 Bộ dữ liệu của các màn chơi

Đối với bộ dữ liệu của màn chơi cũng là 1 list các phần tử và mỗi phần tử sẽ có dạng như sau:

```
{
  "levelNumber": 1,
  "numberStar": 1,
  "isCompleted": false,
  "isPlaying": true
}
```

Hình 2.3 Minh hoạ dữ liệu màn chơi

- “levelNumber” : Chỉ số màn chơi
- “numberStar”: Là số sao của màn chơi
- “isCompleted”: Là màn chơi đã được người chơi chơi qua hay không. Giá trị false là người chơi chưa chơi qua, giá trị true là người chơi đã qua màn
- “isPlaying”: Là màn chơi đang chơi và đang mở khoá gần nhất. Giá trị true là đang chơi, giá trị false là không

2.8. Thiết kế giao diện

2.8.1 Giao diện màn chơi



Hình 2.4 Thiết kế giao diện màn chơi

| Đối tượng | Hình ảnh | Chức năng |
|-------------|---|--|
| Thời gian |  | Thời gian của một câu hỏi |
| Máu |  | Hiển thị số máu còn lại của người chơi |
| Câu hỏi |  | Câu hỏi mà người chơi phải trả lời |
| Câu trả lời |  | Các câu trả lời tương ứng với câu hỏi mà người chơi phải trả lời |
| Nút Pause |  | Nút dừng để tạm dừng màn chơi |



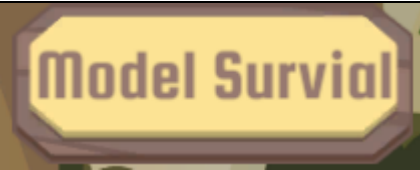
Bảng 2.6 Các đối tượng trong màn chơi

2.8.2 Giao diện các màn hình

a. Màn hình chính:

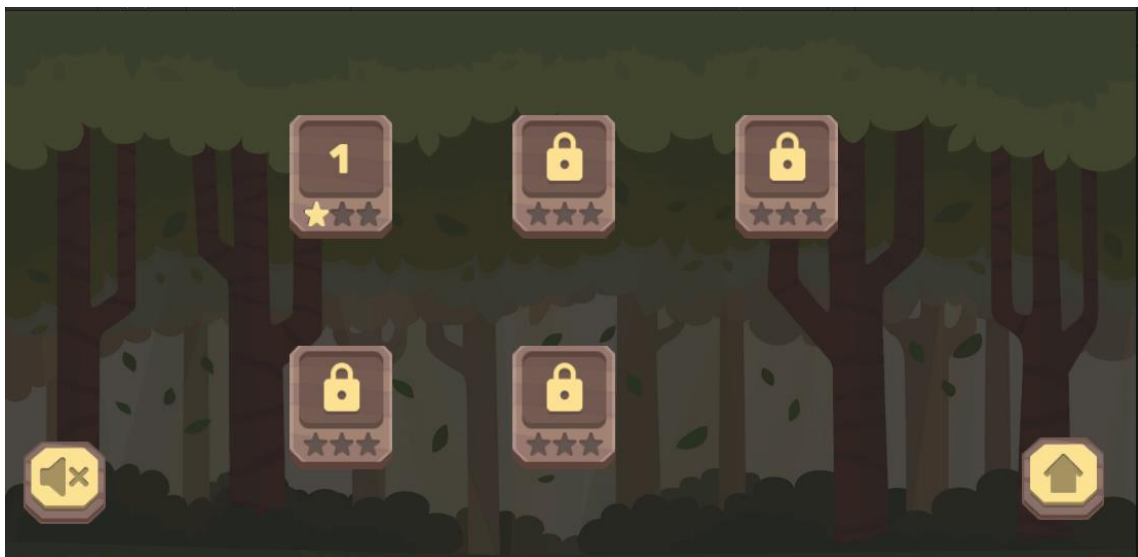


Hình 2.5 Thiết kế màn hình chính

| Đối tượng | Hình ảnh | Chức năng |
|-----------------------|---|---|
| Nút tắt/bật âm thanh |  | Khi ấn sẽ chuyển trạng thái của âm thanh thành tắt hoặc ngược lại |
| Nút chọn các màn chơi |  | Dùng để chuyển sang giao diện chọn màn chơi |
| Nút “Model Survival” |  | Tiến thẳng đến màn đang chơi gần nhất |

Bảng 2.7 Các đối tượng trong màn hình chính

b. Màn hình chọn Level



Hình 2.6 Thiết kế màn hình chọn Level



| Đối tượng | Hình ảnh | Chức năng |
|-----------------------------------|---|---|
| Nút tắt/bật âm thanh |  | Khi ấn sẽ chuyển trạng thái của âm thanh thành tắt hoặc ngược lại |
| Màn chơi đã được mở |  | Khi người chơi ấn vào sẽ tiến vào màn chơi đó |
| Màn chơi chưa được mở khoá |  | Người chơi không thể ấn vào và phải hoàn thành màn chơi trước đó |
| Số sao thu thập được của màn chơi |  | Thể hiện số sao mà người chơi đã kiếm được sau khi qua màn |
| Nút chuyển về màn hình chính |  | Khi ấn nút sẽ đưa người chơi về lại màn hình chính |

Bảng 2.8 Các đối tượng trong màn hình chọn Level

c. Màn hình chọn màn chơi chế độ Survival



Hình 2.7 Màn hình chế độ chơi

| Đối tượng | Hình ảnh | Chức năng |
|------------------------------|---|---|
| Các ả chơi |  | Đưa người chơi tiến vào các ả. Như ở đây là ả các phép toán "+" |
| Nút chuyển về màn hình chính |  | Khi ấn nút game sẽ đưa người chơi về lại màn hình chính |

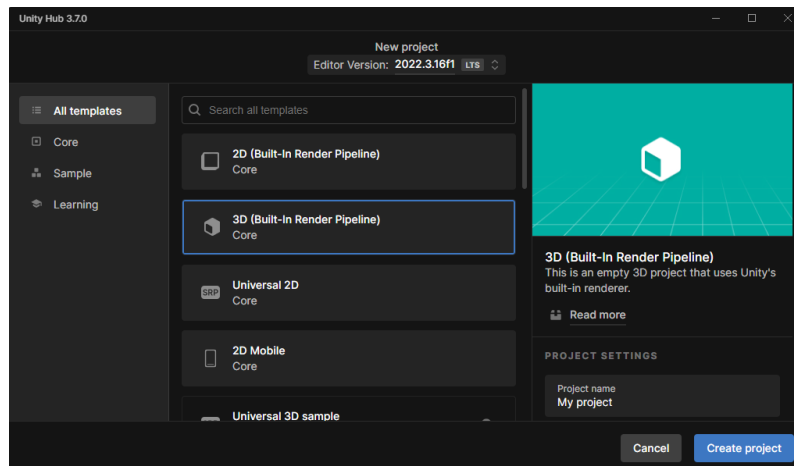
Bảng 2.9 Bảng các đối tượng trong màn chế độ chơi

CHƯƠNG 3. XÂY DỰNG GAME VÀ KẾT QUẢ ĐẠT ĐƯỢC

3.1. Xây dựng game

3.1.1 Tạo project

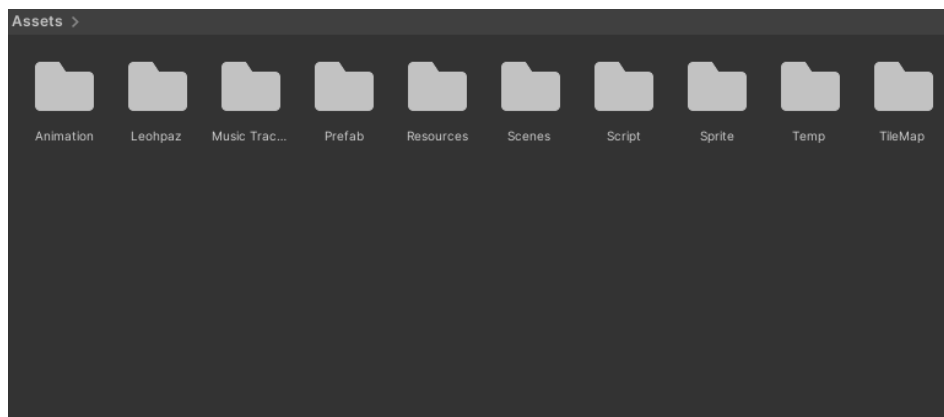
- Bước 1: Chạy phần mềm Unity
- Bước 2: Cửa sổ Unity hiển thị ra, chọn **New Project**
- Bước 3: Trong thẻ Projects, điền các thông tin
- Bước 4: Trong phần **All Templates** chọn 3d nếu muốn lập trình game 3D. Tương tự, chọn 2D nếu muốn lập trình Game 2D
- Bước 5: Nhấn **Create project** để tạo 1 dự án Game



Hình 3.1 Cửa sổ Projects

3.1.2 Khởi tạo file assets

Tạo các Folder chứa tài nguyên:



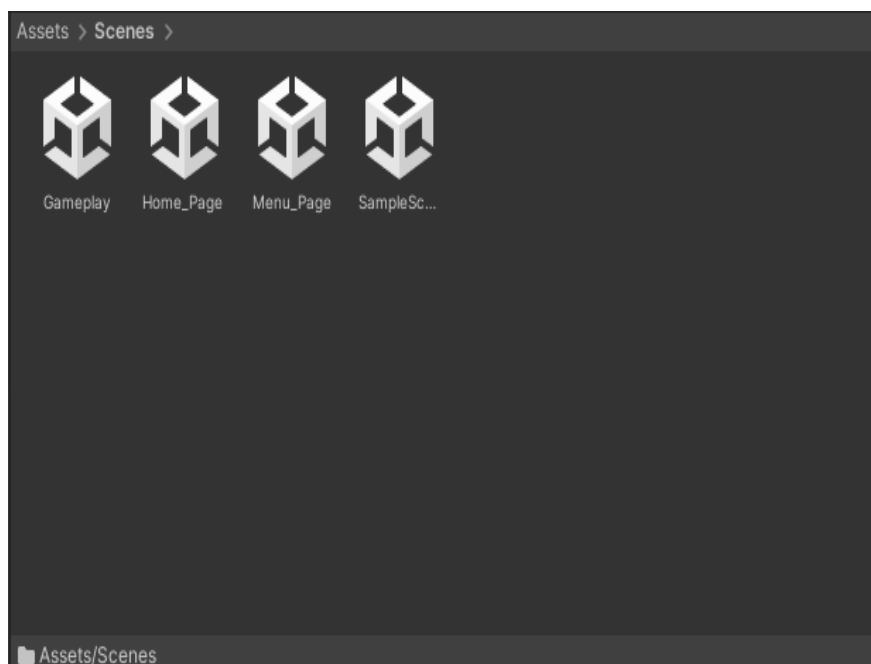
Hình 3.2 Thư mục Assets

- Animation: Thư mục chứa các Animator của các nhân vật
- Music: Chứa các âm thanh cần dùng
- Resources: Nơi chứa các dữ liệu cần thiết cho game như các level, ...
- Prefabs: Chứa gameObject cần dùng nhiều lần, có thể tái sử dụng lại trong các Scene khác
- Scenes: Chứa các cảnh game
- Script: Chứa các Script
- TileMap: Là chứa các TileMap của game để làm Background

Các Folder còn lại là các Package ở trên Assets Store, em đã Import để thêm các tài nguyên trong game

3.1.3 Nhập dữ liệu

Scenes: Tạo ra các Scene cho trò chơi



Hình 3.3 Thư mục Scenes

Resources:

Ở đây em dùng Data là những file Json và tự tạo dữ liệu bằng những file Json:

- Level_Data:

```
File Edit View
{
  "levelData": [
    {
      "levelNumber": 1,
      "masterStar": 1,
      "isCompleted": false,
      "isPlaying": true
    },
    {
      "levelNumber": 2,
      "masterStar": 0,
      "isCompleted": false,
      "isPlaying": false
    },
    {
      "levelNumber": 3,
      "masterStar": 0,
      "isCompleted": false,
      "isPlaying": false
    },
    {
      "levelNumber": 4,
      "masterStar": 0,
      "isCompleted": false,
      "isPlaying": false
    },
    {
      "levelNumber": 5,
      "masterStar": 0,
      "isCompleted": false,
      "isPlaying": false
    }
  ]
}
```

Hình 3.4 Level_Data

- Question_Data:

```
File Edit View
{
  "questions": [
    {
      "question": "17 + 5 = ?",
      "correctAnswer": "22",
      "wrongAnswers": [ "2", "20", "23" ]
    },
    {
      "question": "10 - 2 = ?",
      "correctAnswer": "8",
      "wrongAnswers": [ "6", "12", "7" ]
    },
    {
      "question": "4 * 6 = ?",
      "correctAnswer": "24",
      "wrongAnswers": [ "20", "22", "25" ]
    },
    {
      "question": "15 / 3 = ?",
      "correctAnswer": "5",
      "wrongAnswers": [ "4", "6", "3" ]
    },
    {
      "question": "8 + 9 = ?",
      "correctAnswer": "17",
      "wrongAnswers": [ "16", "18", "15" ]
    },
    {
      "question": "20 - 7 = ?",
      "correctAnswer": "13",
      "wrongAnswers": [ "14", "12", "15" ]
    },
    {
      "question": "9 * 3 = ?",
      "correctAnswer": "27",
      "wrongAnswers": [ "24", "30", "25" ]
    },
    {
      "question": "18 / 2 = ?",
      "correctAnswer": "9",
      "wrongAnswers": [ "8", "10", "6" ]
    },
    {
      "question": "6 + 7 = ?",
      "correctAnswer": "13",
      "wrongAnswers": [ "12", "14", "11" ]
    },
    {
      "question": "30 - 15 = ?",
      "correctAnswer": "15",
      "wrongAnswers": [ "14", "16", "20" ]
    },
    {
      "question": "5 * 5 = ?"
    }
  ]
}
```

Hình 3.5 Question_Data

- Script: Lưu lại các File C# được sử dụng



Hình 3.6 Thư mục Script

3.2. Kết quả đạt được

3.2.1 Giao diện chính



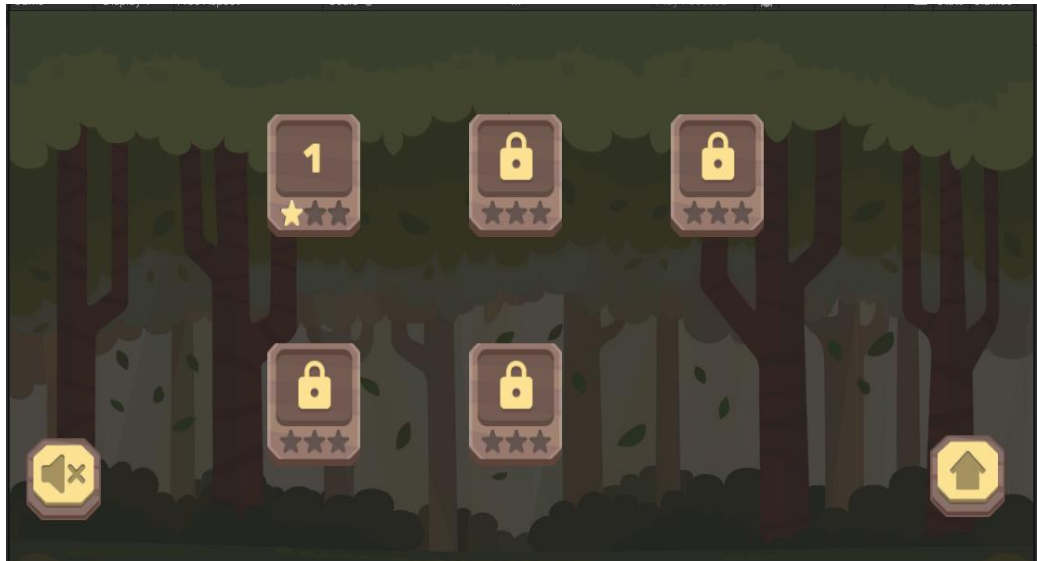
Hình 3.7 Màn hình Home

Tại màn hình bắt đầu, người chơi có 3 nút tùy chọn:

- Model Survival: Để tiến vào chế độ chơi sinh tồn
- Nút để tắt/mở âm thanh của game

- Nút để chuyển đến cạnh chọn Level

3.2.2 Giao diện chọn Level



Hình 3.8 Màn hình chọn Level

Tại đây chúng ta sẽ chọn Level để chơi game, mỗi level chúng ta sẽ phải trả lời các câu hỏi được game đặt ra. Người chơi phải trả lời hết tất cả các câu trả lời để có thể qua màn. Nhiệm vụ của chúng ta là hoàn thành chúng và mở khoá các level sau. Mức độ khó sẽ tăng dần theo mỗi màn chơi tại 1 chủ đề nhất định. Vậy nên chúng ta cần nhanh tay và tăng độ tư duy của bản thân để có thể hoàn thành các màn chơi trong thời gian nhất định

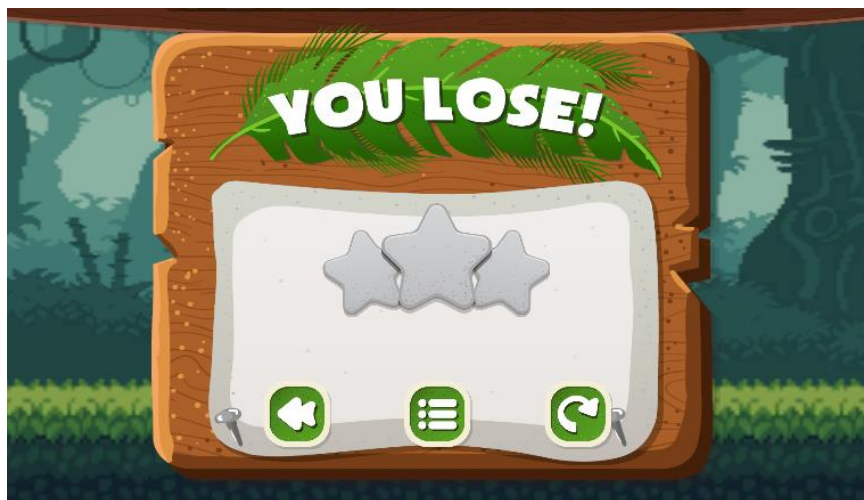
3.2.3 Giao diện chơi game



Hình 3.9 Màn hình Game Play

Nhiệm vụ của chúng ta rất đơn giản. Ở dưới giữa màn hình sẽ là 1 trong 3 câu hỏi mà màn chơi đặt ra, nhiệm vụ của chúng ta là trả lời đúng câu hỏi. Bằng các chọn câu trả lời chính xác, vị hiệp sĩ của chúng ta sẽ đánh bại quái vật và băng qua khu rừng. Nhưng hãy nhớ rằng mỗi màn chơi sẽ có thời gian nhất định, chúng ta phải hoàn thành trong thời gian chỉ định nhé và nếu chọn sai thì quái vật sẽ di chuyển nhanh hơn nên phải cẩn thận, suy nghĩ thật kỹ trước khi trả lời.

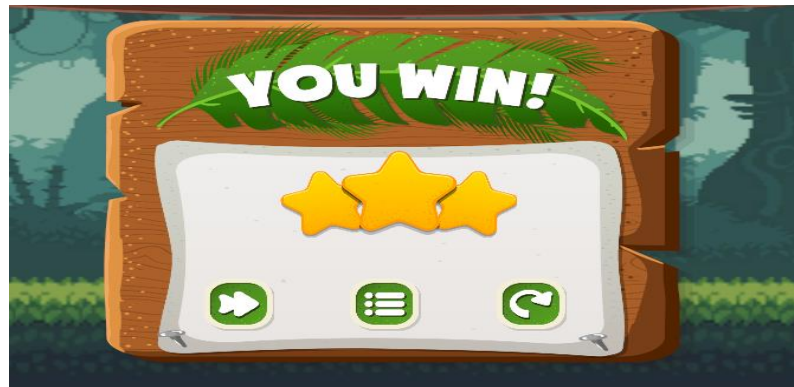
3.2.4 Giao diện khi hết thời gian



Hình 3.10 Màn hình thua

Khi người chơi không trả lời được 3 câu hỏi thì sẽ thua, lúc này màn hình sẽ hiển thị You Lose!, tại đây người chơi có thể chọn **Restart** để chơi lại màn đó, nút **Pre** để có thể thoát ra màn hình chính của game, nút còn lại để người chơi trở lại màn hình chọn Level

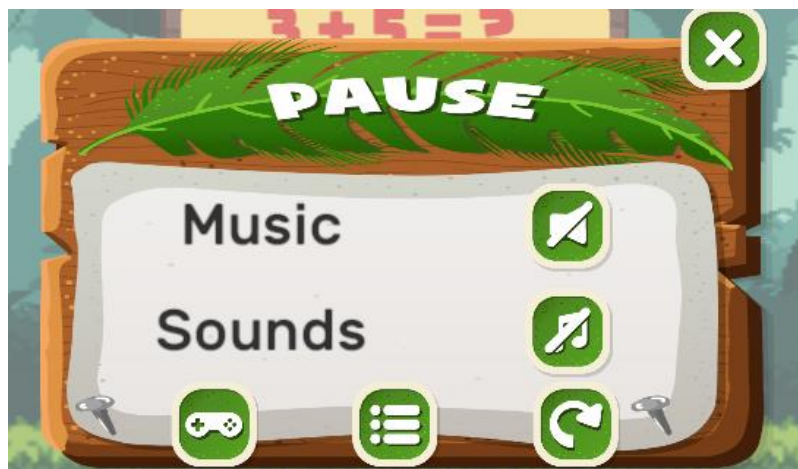
3.2.5 Giao diện khi thắng màn



Hình 3.11 Màn hình qua màn

Đây là những gì chúng ta sẽ thấy khi hoàn thành màn chơi. Số sao sẽ dựa vào số máu mà người chơi còn lại sau khi hoàn thành màn chơi. Cũng như giao diện khi thua, người chơi có thể chọn chơi lại, chơi màn tiếp theo hoặc trở về chọn level

3.2.6 Giao diện Setting:



Hình 3.12 Màn hình Setting của game

Tại đây có 5 nút hoạt động:

- Music: Để tắt/bật âm thanh của game
- Sounds: Để tắt/bật Sound Effect của game
- Nút Home: Để quay về màn hình chính của game
- Nút Level: Để quay về màn hình chọn Level
- Nút Restart: Để chơi lại Level

CHƯƠNG 4. KIỂM THỬ HỆ THỐNG

4.1. Kế hoạch kiểm thử

4.1.1 Mục tiêu kiểm thử

- Đảm bảo chức năng của trò chơi hoạt động đúng như kế hoạch
- Xác định và sửa lỗi phát sinh trong quá trình phát triển
- Kiểm tra hiệu suất và khả năng đáp ứng của trò chơi trên các thiết bị khác nhau

4.1.2 Phạm vi kiểm thử

- Kiểm thử chức năng
- Kiểm thử giao diện

4.1.3 Các ca kiểm thử

a. Kiểm thử chức năng

| STT | Nội dung | Đầu vào | Đầu ra mong muốn | Kết quả |
|-----|---|--|--|---------|
| 1 | Kiểm tra chức năng nút tắt bật âm thanh | Nút “Tắt/Bật” ở trên màn hình | Chuyển đổi giao diện hình ảnh của nút bật tương ứng với trạng thái bật/tắt | Đạt |
| 2 | Kiểm tra nút “Menu” | Nút chọn Level ở màn hình bắt đầu game | Giao diện của game sẽ chuyển sang màn hình chọn Level | Đạt |
| 3 | Kiểm tra việc chọn một màn | Chọn một màn chơi trong danh | Giao diện của game sẽ chuyển sang màn hình của màn chơi đã chọn | Đạt |

| | | | | |
|---|--|--|--|-----|
| | chơi bất kỳ | sách các màn | | |
| 4 | Kiểm tra mở khoá màn chơi khi đạt đủ điều kiện | Hoàn thành màn chơi ở trước đó và quay lại màn hình chọn level | Một màn chơi mới được mở khoá và có thể chọn để chơi | Đạt |
| 5 | Kiểm tra xem các nhân vật trong game có hoạt động đúng chức năng không | Bắt đầu một màn chơi và quan sát các hành động của nhân vật | Các nhân vật hoạt động đúng như đã thiết kế ở trên | Đạt |
| 6 | Kiểm tra câu hỏi toán học | Bắt đầu level, quan sát các câu hỏi và câu trả lời | Câu hỏi hiện thị ở đúng chỗ. Đưa ra kết quả đúng/sai tương ứng với câu hỏi | Đạt |
| 7 | Kiểm tra màn hình khi thắng và thua | <div>Kiểm tra việc hiển thị thông báo thắng</div> <div>Kiểm tra việc hiển thị thông báo thua</div> | Khi hoàn thành 1 màn chơi | Đạt |

Bảng 4.1 Kiểm thử chức năng

b. Kiểm thử giao diện

| STT | Nội dung | Đầu vào | Đầu ra mong muốn | Kết quả |
|-----|--|--|---|---------|
| 1 | Kiểm tra giao diện trên độ phân giải 1280x720 | Mở ứng dụng trên màn hình có độ phân giải 1280x720 | Tất cả các yếu tố trong giao diện hiển thị đúng, không bị lệch | Đạt |
| 2 | Kiểm tra tương tác của người chơi với câu trả lời | Trả lời một câu hỏi toán học trong game | Câu hỏi được trả lời và game phản hồi theo câu trả lời của người dùng | Đạt |
| 3 | Kiểm tra tương tác của người chơi với các tùy chọn trong màn hình tạm dừng | Thay đổi các tùy chọn trong màn hình tạm dừng | Các tùy chọn được thay đổi và áp dụng theo mong muốn của người dùng | Đạt |
| 4 | Kiểm tra tương tác của người chơi với các tùy chọn trong màn hình thắng/thua | Nhấn các nút trên màn hình thắng/thua | Trò chơi phản hồi theo hành động của người chơi | Đạt |

Bảng 4.2 Kiểm thử giao diện

KẾT LUẬN

Do thời gian thực hiện hạn chế em đã xây dựng được tựa game 2D. Qua đó em đã học được thêm nhiều kiến thức và kỹ năng lập trình game như củng cố kiến thức lập trình C#, Unity để tạo ra 1 tựa game hoàn chỉnh. Qua quá trình thực hiện game, em thu hoạch được những kinh nghiệm sau:

- ✓ Hiểu hơn về Unity Framework để tiến hành tạo game và chạy game trên Unity
- ✓ Hiểu hơn về ngôn ngữ lập trình
- ✓ Có kiến thức về thiết kế kịch bản, bối cảnh của 1 trò chơi
- ✓ Xây dựng được nhân vật và dữ liệu của trò chơi
- ✓ Hoàn thành việc thiết kế hệ thống giao diện
- Định hướng phát triển:
 - ✓ Thiết kế thêm các màn chơi.
 - ✓ Thiết kế thêm chức năng phần quà cộng thêm điểm số.
 - ✓ Tăng độ khó của trò chơi qua các màn chơi.
 - ✓ Các chức năng điều chỉnh Options (âm thanh..)
 - ✓ Thiết kế về màu sắc, bố cục game tốt hơn
 - ✓ Thêm chức năng xem quảng cáo
 - ✓ Thêm chức năng chia sẻ với bạn bè trên facebook,
 - ✓ Xây dựng thêm nhiều game hơn nữa

TÀI LIỆU THAM KHẢO

- [1] Sue Blackman (2014). Unity for absolute beginners
- [2] Alan Thorn (2013). Learn Unity for 2d game development
- [3] <https://www.raywenderlich.com/2573-unity-4-3-2d-tutorial-getting-started>
- [4] https://vietjack.com/csharp/hoc_c_sharp_co_ban_nang_cao.jsp
- [5] <https://www.w3schools.com/cs/index.php>
- [6] <https://freetuts.net/c-sharp-la-gi-tong-quan-ve-c-sharp-1045.html>
- [7] <https://itnavi.com.vn/blog/lap-trinh-unity>