

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN KHOA HỌC MÁY TÍNH

-----o0o-----



ĐỒ ÁN MÔN HỌC

[Lập trình game bằng công cụ Unity và C#]

GVHD: Phạm Hải Đăng

SVTH: Trần Đình Vũ

MSSV: 20176126

TP. HÀ NỘI, THÁNG 6 NĂM 2020

LỜI CẢM ƠN

Em xin chân thành cảm ơn Thầy Phạm Hải Đăng – Giáo viên hướng dẫn của em – đã hướng dẫn hỗ trợ cũng như cung cấp các lời khuyên cần thiết để thực hiện, triển khai, giải quyết các vấn đề và hoàn thiện đề tài.

Mặc dù em đã cố gắng hoàn thành đề tài trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, mong sự cảm thông và tận tình chỉ bảo của quý thầy cô và các bạn.

Tp. Hà Nội, ngày 22 tháng 5 năm 2020 .

Sinh viên Trần Đình Vũ

MỤC LỤC

1. GIỚI THIỆU	1
1.1 Tổng quan.....	1
1.2 Nhiệm vụ đề tài.....	1
2. LÝ THUYẾT	2
2.1 Unity là gì?	2
2.2 Các khái niệm trong Unity	3
2.3 Workflow của Unity	4
2.4 Tìm hiểu về các Game Programming Pattern được áp dụng.....	5
6. THIẾT KẾ GAME.....	6
6.1 Nội dung game.....	6
6.2 Nền tảng hướng tới	6
6.3 Yêu cầu đặt ra cho game	7
6.4 Công nghệ sử dụng	7
6.5 Concept.....	7
6.6 Người chơi.....	7
6.7 Mục tiêu.....	7
6.8 Hành động	7
6.9 Đồ hoạ	8
6.10 Gameplay.....	8
6.11 Ý tưởng giao diện	8
6.12 Usecase.....	9
.....	9

7. PHÁT TRIỂN SẢN PHẨM PHẦN MỀM	10
7.1 Hệ thống các script chính.....	10
7.2 Một số biểu đồ khác	12
7.3 Các Unity Assets được sử dụng trong dự án:.....	17
8. KẾT QUẢ THỰC HIỆN.....	17
9. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	21
9.1 Kết luận	21
9.2 Hướng phát triển	21
10. TÀI LIỆU THAM KHẢO.....	21

1. GIỚI THIỆU

1.1 Tổng quan

Xã hội ngày càng phát triển, nhu cầu giải trí của con người ngày càng tăng cao. Cùng với sự phát triển của công nghệ thông tin, điện tử, các thiết bị chơi game ngày càng phổ biến. Trong những năm gần đây, ngành công nghệ game nổi lên và đem lại doanh thu khổng lồ. Có thể nói game không chỉ là một phương tiện giải trí cơ bản, nó còn là một lĩnh vực đáng quan tâm và theo đuổi. Đặc biệt, lập trình game có thể giúp ta học hỏi những kiến thức mới về công cụ Unity và ngôn ngữ C#.

Một trong các Game Engine được sử dụng để phát triển game VR thịnh hành nhất hiện nay là Unity. Qua đó, với đề tài này, em sẽ tập trung tìm hiểu về Game Engine Unity và cách phát triển một tựa game điện thoại trên đó.

1.2 Nhiệm vụ đề tài

Nội dung 1: Tìm hiểu nguyên lý, lý thuyết về ngôn ngữ C#

Nội dung 2: Tìm hiểu nguyên lý, lý thuyết về công cụ Unity

Nội dung 3: Thiết kế kịch bản game, mockup design

Nội dung 4: Thiết kế và xây dựng các hệ thống trong game

Nội dung 5: Thiết kế và kết nối UI vào hệ thống trong game

Nội dung 6: Thu thập ý kiến người chơi để sửa đổi

2. LÝ THUYẾT

2.1 Unity là gì?

- Unity là một “cross- platform game engine” tạm hiểu là công cụ phát triển game đa nền tảng được phát triển bởi Unity Technologies. Game engine này được sử dụng để phát triển game trên PC, consoles, thiết bị di động và trên websites.
- Ưu điểm của Unity:
 - + Chức năng cốt lõi đa dạng bao gồm: cung cấp công cụ dựng hình (kết xuất đồ họa) cho các hình ảnh 2D hoặc 3D, công cụ vật lý (tính toán và phát hiện va chạm), âm thanh, mã nguồn, hình ảnh động, trí tuệ nhân tạo, phân luồng, tạo dòng dữ liệu xử lý, quản lý bộ nhớ, dựng ảnh đồ thị và kết nối mạng. Nhờ có các engine mà công việc làm game trở nên ít tốn kém và đơn giản hơn.
 - + Hỗ trợ đa nền tảng: Một trong các thế mạnh của Unity3D chính là khả năng hỗ trợ gần như toàn bộ các nền tảng hiện có bao gồm: PlayStation 3, Xbox 360, Wii U, iOS, Android, Windows, Blackberry 10, OS X, Linux, trình duyệt Web và cả Flash. Nói cách khác, chỉ với một gói engine, các studio có thể làm game cho bất kỳ hệ điều hành nào và dễ dàng convert chúng sang những hệ điều hành khác nhau. Đồng thời, đây cũng là giải pháp cho các game online đa nền tảng – có thể chơi đồng thời trên nhiều hệ điều hành, phần cứng khác nhau như Web, PC, Mobile, Tablet....
 - + Dễ sử dụng: Unity3D được build trong một môi trường phát triển tích hợp, cung cấp một hệ thống toàn diện cho các lập trình viên, từ soạn thảo mã nguồn, xây dựng công cụ tự động hóa tiến trình sửa lỗi. Do được hướng đến đồng thời cả lập trình viên không chuyên và studio chuyên nghiệp, nên Unity3D khá dễ sử dụng. Hơn nữa, đây là một trong những engine phổ biến nhất trên thế giới, người dùng có thể dễ dàng tìm kiếm kinh nghiệm sử dụng của “tiền bối” trên các forum công nghệ.
 - + Tính kinh tế cao: Unity Technologies hiện cung cấp bản miễn phí engine Unity3D cho người dùng cá nhân và các doanh nghiệp có doanh thu dưới 100.000 USD/năm. Với bản Pro, người dùng phải trả 1.500 USD/năm – một con số rất khiêm tốn so với những gì engine này mang lại.

2.2 Các khái niệm trong Unity

1. GameObject

Một đối tượng cụ thể trong game gọi là một game object, có thể là nhân vật, đồ vật nào đó. Ví dụ: cây cối, xe cộ, nhà cửa, người...

2. Component

Một GameObject sẽ có nhiều thành phần cấu tạo nên nó như là hình ảnh (sprite render), tập hợp các hành động (animator), thành phần xử lý va chạm (collision), tính toán vật lý (physical), mã điều khiển (script), các thành phần khác... mỗi thứ như vậy gọi là một component của GameObject.

3. 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.

4. Animation

Là tập một hình ảnh động dựa trên sự thay đổi liên tục của nhiều sprite khác nhau.

5. Key Frame

Key Frame hay Frame là một trạng thái của một animation. Có thể được tạo nên từ 1 sprite hay nhiều sprite khác nhau.

6. Prefabs

Là một khái niệm trong Unity, dùng để sử dụng lại các đối tượng giống nhau có trong game mà chỉ cần khởi tạo lại các giá trị vị trí, tỉ lệ biến dạng và góc quay từ một đối tượng ban đầu. Ví dụ: Các đối tượng là đồng tiền trong game Mario đều có xử lý giống nhau, nên ta chỉ việc tạo ra một đối tượng ban đầu, các đồng tiền còn lại sẽ sử dụng prefabs. Hoặc khi ta lát gạch cho một cái nền nhà, các viên gạch cũng được sử dụng là prefabs.

7. Sounds

Âm thanh trong game.

8. Script

Script là tập tin chứa các đoạn mã nguồn, dùng để khởi tạo và xử lý các đối tượng trong game. Trong Unity có thể dùng C#, Java Script, BOO để lập trình Script.

9. Scenes

Quản lý tất cả các đối tượng trong một màn chơi của game.

10. Assets

Bao gồm tất cả những gì phục vụ cho dự án game như sprite, animation, sound, script, scenes...

11. Camera

Là một game object đặc biệt trong scene, dùng để xác định tầm nhìn, quan sát các đối tượng khác trong game.Camera

12. Transform

Là 3 phép biến đổi tịnh tiến, quay theo các trục, và phóng to thu nhỏ một đối tượng

2.3 Workflow của Unity

- Unity được xây dựng và định hướng theo kiến trúc Components-based.
- Một trong những lý do là vì Unity là một có thể mạnh về giao diện sử dụng. Unity cho chúng ta thấy mọi thứ chúng ta đang làm và có thể sửa chúng trong thời gian thực (Real-time). Theo nhiều người đây là một trong những khía cạnh rất cần thiết của việc phát triển game trong hiện đại và tương lai.
 - + Tổng quan nhất trong Unity là một Project, nơi chứa tất cả các dữ liệu và tài nguyên của trò chơi của bạn. Trong đó sẽ có các Scene (cảnh game), và mỗi Scene sẽ có các Game Object. Các Scene có thể được chuyển cho nhau và chọn cho các GameObject active hay không.
 - + Ở phạm vi nhỏ hơn, ta có các GameObject và Components. Một GameObject bản thân là một hộp chứa vô hình không có gì cả. Nếu không cho thêm vào các Component thích hợp, GameObject sẽ không làm được gì. Sử dụng Component, người sử dụng có thể dễ dàng thiết kế và tạo nên các tài nguyên trong

game và sử dụng lại chúng với các GameObject khác, giúp cho việc phát triển game dễ dàng hơn.

- + Component duy nhất mà GameObject nào cũng phải có là Transform
- Hiện tại, Unity đang dần chuyển giao sang mô hình ECS (Entity Component System) dựa theo hướng dữ liệu, được cho là đem lại hiệu năng tốt hơn so với hiện tại.

2.4 Tìm hiểu về các Game Programming Pattern được áp dụng

1. Observer Pattern

- Observer pattern là một mẫu thiết kế phần mềm mà một đối tượng, gọi là subject, duy trì một danh sách các thành phần phụ thuộc nó, gọi là observer, và thông báo tới chúng một cách tự động về bất cứ thay đổi nào, thường thì bằng cách gọi 1 hàm của chúng.
- Mục tiêu:
 - + Định nghĩa mối phụ thuộc một - nhiều giữa các đối tượng để khi mà một đối tượng có sự thay đổi trạng thái, tất cả các thành phần phụ thuộc của nó sẽ được thông báo và cập nhật một cách tự động.
 - + Một đối tượng có thể thông báo đến một số lượng không giới hạn các đối tượng khác.
- Tính áp dụng:
 - + Sự thay đổi trạng thái ở 1 đối tượng có thể được thông báo đến các đối tượng khác mà không phải giữ chúng liên kết quá chặt chẽ
 - + Cần mở rộng dự án với ít sự thay đổi nhất.

2. Singleton Pattern

- Singleton Pattern là một mẫu thiết kế (design pattern) được sử dụng để bảo đảm rằng mỗi một lớp (class) chỉ có được một thể hiện (instance) duy nhất và mọi tương tác đều thông qua thể hiện này.
- Singleton Pattern cung cấp một phương thức khởi tạo private, duy trì một thuộc tính tĩnh để tham chiếu đến một thể hiện của lớp Singleton này. Nó cung cấp thêm một phương thức tĩnh trả về thuộc tính tĩnh này.

3. State Pattern

- State design pattern là một trong những mẫu thiết kế hành vi. State design pattern được sử dụng khi một Object thay đổi hành vi của nó dựa trên trạng thái bên trong của nó. Nếu chúng ta phải thay đổi hành vi của đối tượng dựa trên trạng thái của nó. Trong State pattern, chúng ta tạo các đối tượng đại diện cho các trạng thái khác nhau và một đối tượng bối cảnh mà hành vi của chúng thay đổi khi các thay đổi đối tượng trạng thái của nó.

4. Game Loop Pattern

- Game Loop pattern là một vòng lặp liên tục được kiểm soát để giữ cho game chạy.
- Bản thân Unity đã có sẵn Game Loop pattern ở trong kiến trúc của mình. Nó chính là MonoBehaviour bao gồm:
 - + OnEnable()
 - + OnDisable()
 - + Awake()
 - + Start()
 - + Update

5. Object Pool Pattern

- Chúng ta dùng Object Pool Pattern quản lý một tập hợp các objects mà sẽ được tái sử dụng trong chương trình. Chúng được gọi ra từ pool, sử dụng trong một khoảng thời gian nhất định rồi trả về pool. Trong khoảng thời gian vắng mặt đó của object, không thành phần nào có thể sử dụng tận khi nó được quay trở về pool.
- Object Pool Pattern được coi là sử dụng thích hợp khi có nhiều hơn một đối tượng và số đối tượng được khởi tạo là hạn chế.

6. THIẾT KẾ GAME

6.1 Nội dung game

- Người chơi sẽ quản lý nguyên liệu và phục vụ cho khách hàng là các con vật càng nhiều càng tốt

6.2 Nền tảng hướng tới

- Android

6.3 Yêu cầu đặt ra cho game

- Dung lượng thấp, cài đặt và sử dụng được trên nhiều thiết bị với các cấu hình khác nhau
- Đơn giản nhưng hấp dẫn với người chơi, có tính chơi lại cao
- Giao diện trực quan, âm thanh vui nhộn thu hút
- Có bảng điểm trực tuyến để người chơi mọi nơi có thể so sánh kết quả

6.4 Công nghệ sử dụng

- Engine: Unity 2019
- Text editor: Visual Studio Code
- Vẽ đồ họa: Aesprite

6.5 Concept

- Concept game dựa theo các game phục vụ nhà hàng, quán café

6.6 Người chơi

- Trò chơi sẽ được chơi theo cá nhân trên điện thoại và mọi người có thể so sánh điểm số với nhau

6.7 Mục tiêu

- Người chơi cần giao các đơn đồ uống nhanh và đúng trước khi thời gian hết
- Người chơi cần trụ càng lâu càng tốt để đạt điểm số cao nhất

6.8 Hành động

- Người chơi có thể:
 - + Kéo thả để phục vụ khách
 - + Bỏ sung nguyên liệu trong kho
 - + Tắt bật âm thanh
 - + Chơi lại màn chơi
 - + Xem hướng dẫn chơi
 - + Thay đổi tên của mình
 - + Tải lên điểm số của mình lên hệ thống
 - + Xem điểm số của những người chơi khác trên hệ thống

6.9 Đồ hoạ

- Lấy cảm hứng từ đồ hoạ 2D pixel 1-bit
- Các đồ vật và nhân vật sẽ có màu sắc đặc trưng dễ nhận dạng để nhấn mạnh vào cơ chế của trò chơi

6.10 Gameplay

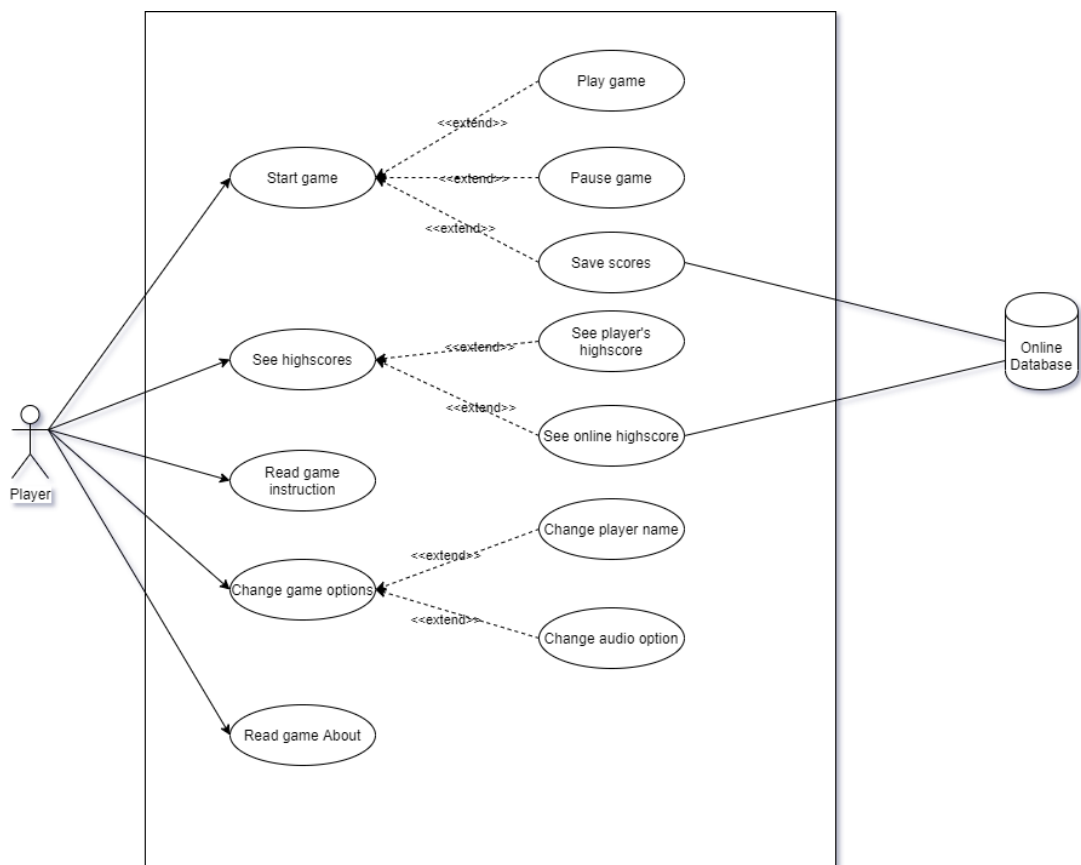
- Game sẽ bao gồm 2 scene:
 - + Menu scene: Nơi người dùng có thể bắt đầu game, mở menu nâng cấp, chỉnh settings và xem bảng điểm số
 - + Game scene: Nơi người dùng sẽ trực tiếp chơi và tương tác với game, bao gồm cả pause menu và bảng kết quả cuối game
- Các thành phần trong gameplay chính:
 - + Khách hàng: Gồm 2 loại là Khách thường và khách VIP. Khách thường sẽ gọi 1 đơn còn với khách VIP người chơi có thể đưa bao nhiêu đơn có thể phục vụ được và số điểm sẽ được x2. Các khách hàng sau một khoảng thời gian sẽ được spawn, càng chơi lâu tần suất spawn càng tăng.
 - + Kho nguyên liệu: Nguyên liệu được chia làm 3 loại: Drink, Cream, Fruit. Mỗi loại đều có 8 màu cơ bản và đều có số lượng cụ thể
 - + Cốc: Là nơi người chơi thả 3 loại nguyên liệu vào để tạo thành 1 cốc hoàn chỉnh. Sau khi thả xong người chơi có thể kéo thả cốc đến vị trí của khách hàng tương ứng
 - + Thùng rác: Khi người dùng làm sai đơn, họ có thể thả cốc vào đây để làm một cốc mới
 - + Restock Machine: Khi thiếu nguyên liệu, người dùng kéo nguyên liệu đó vào và sau một khoảng thời gian cố định số lượng nguyên liệu đó sẽ được hồi

6.11 Ý tưởng giao diện

- Menu scene:
 - + Play game
 - + Score
 - + How to
 - + Options
 - + About

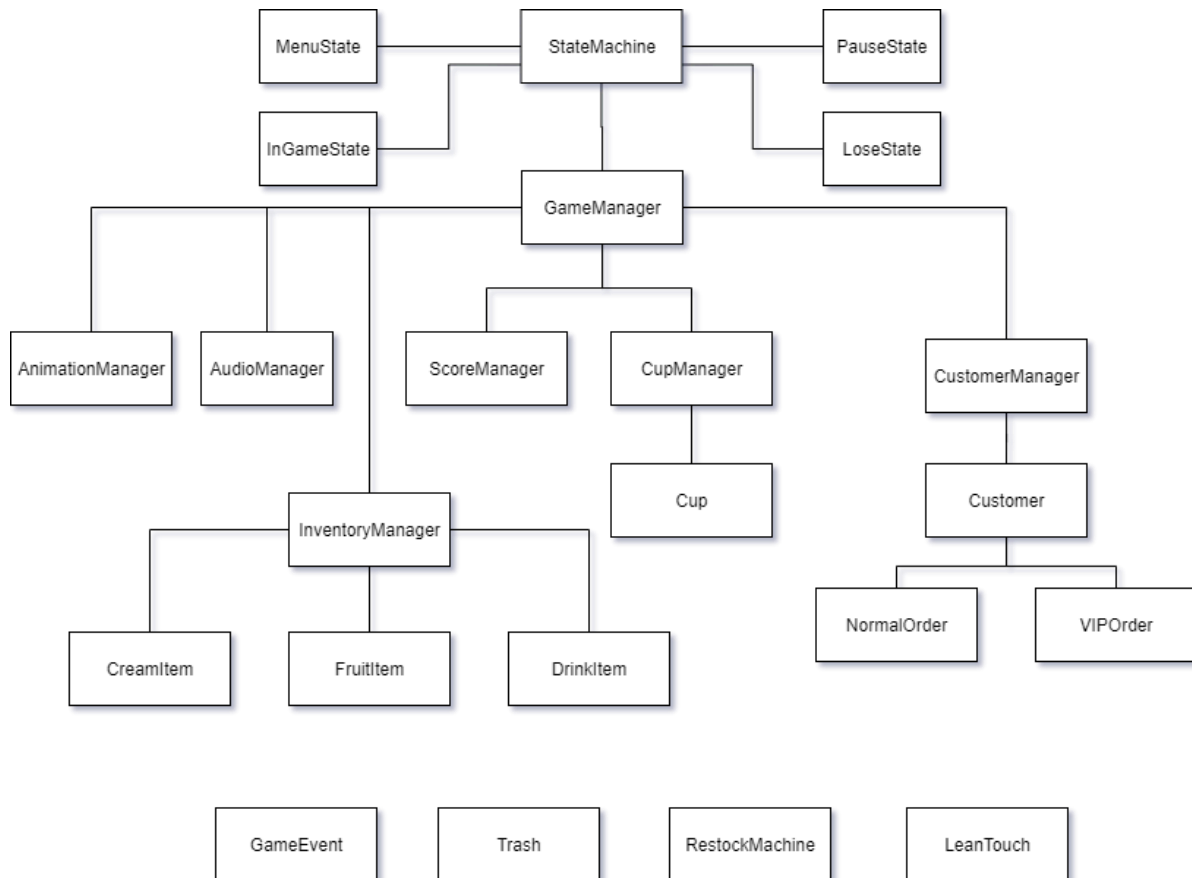
- Game scene:
 - + Background
 - + Customers
 - + Restock Machine
 - + Trash
 - + Inventory
 - + Cup
 - + Score + Multiplier
 - + Lives left
 - + Pause menu
 - + Game Over

6.12 Usecase



7. PHÁT TRIỂN SẢN PHẨM PHẦN MỀM

7.1 Hệ thống các script chính

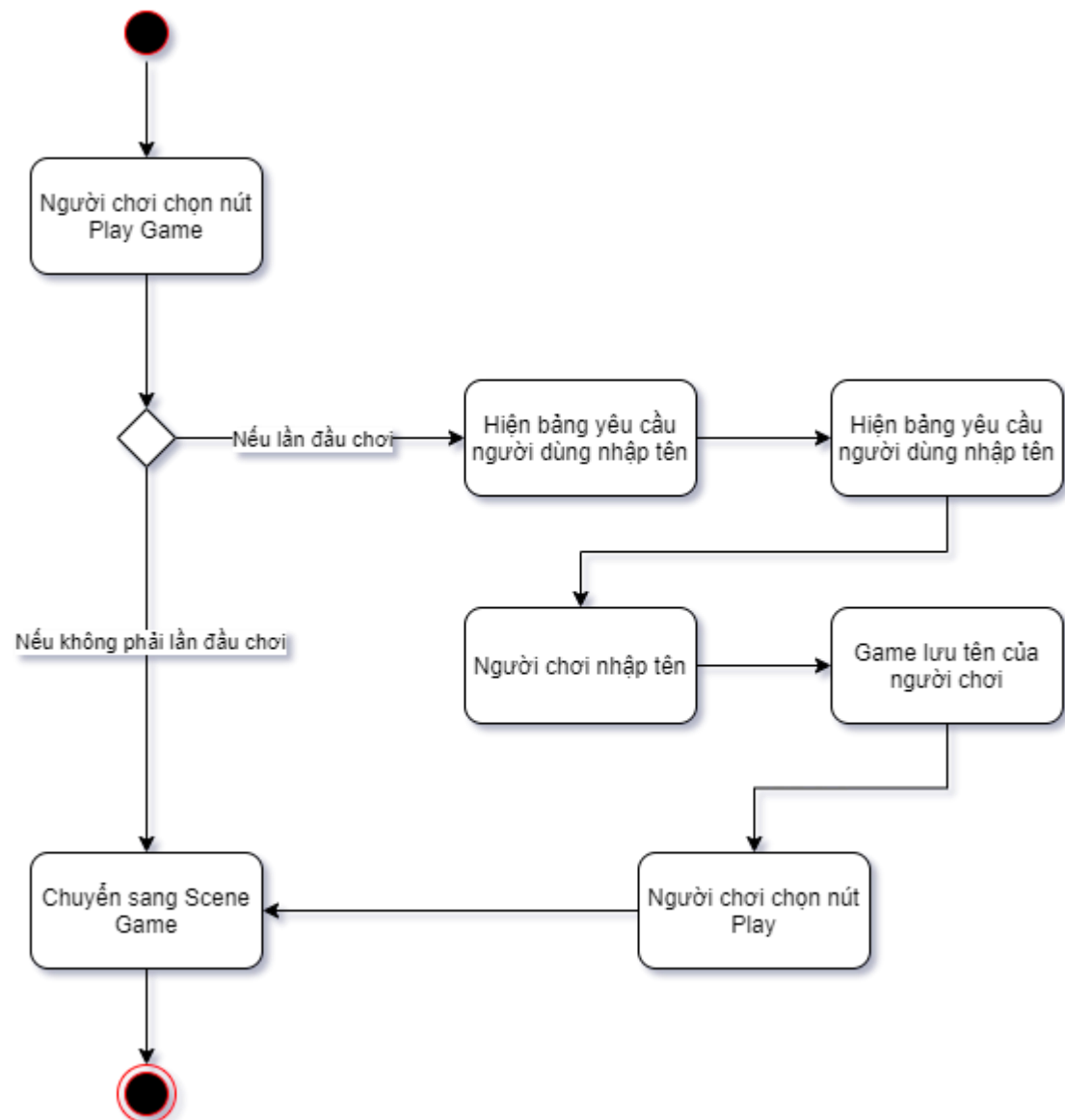


- StateMachine:
 - + Sử dụng State Machine Design Pattern để quản lý các trạng thái của trò chơi bao gồm: MenuState, InGameState, PauseState.
 - + Mỗi state script sẽ có các method để chạy code khi enter state hay exit state hoặc khi đang ở trong state.
- AnimationManager:
 - + Quản lý các hiệu ứng (tweening) và UI trong game
 - + Sử dụng Singleton pattern để các script khác có thể gọi để chạy các animation
- AudioManager:
 - + Quản lý các âm thanh như nhạc, hiệu ứng, thông báo,...
 - + Sử dụng Singleton pattern để các script khác có thể gọi để chạy các Audio

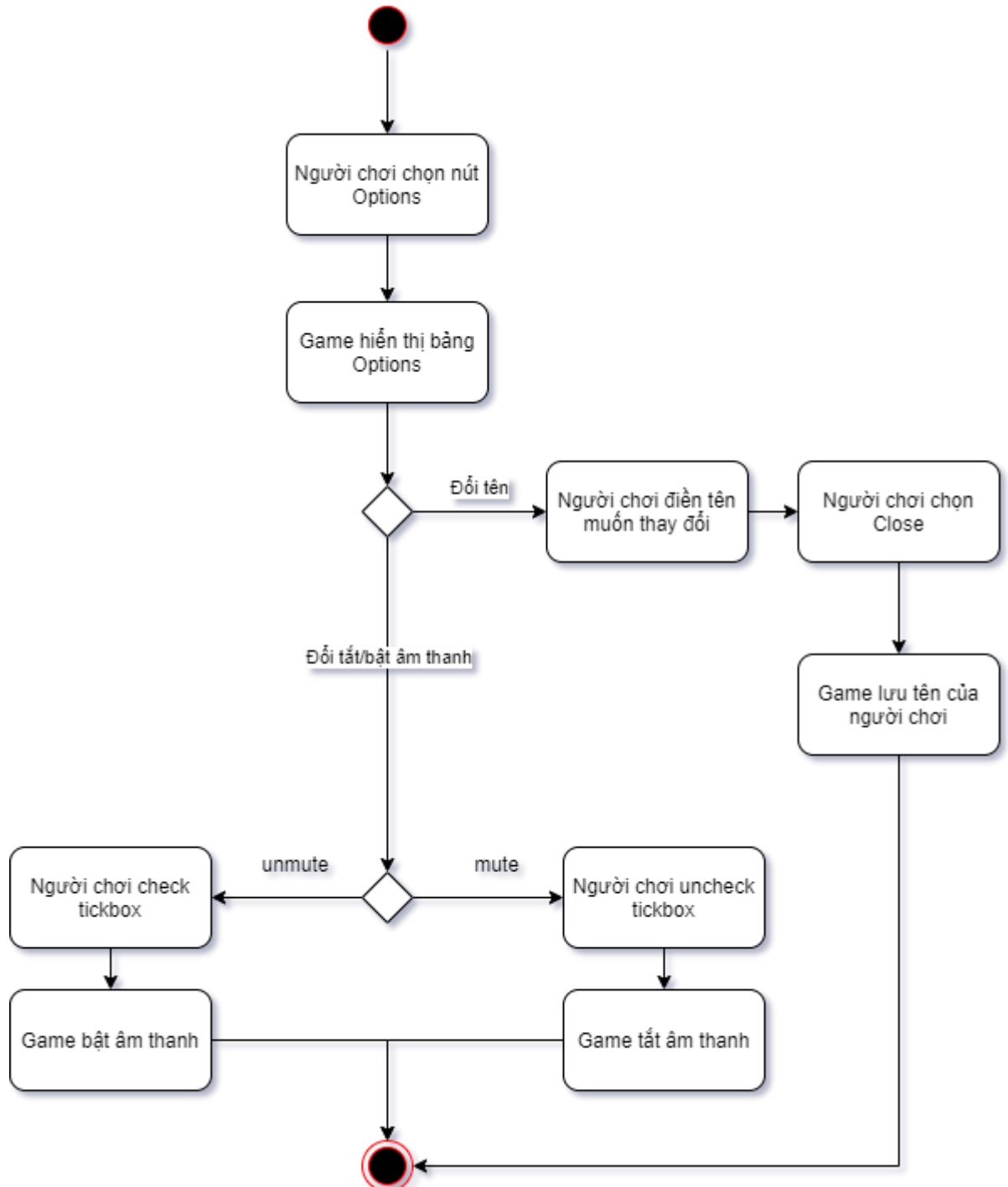
- ScoreManager:
 - + Quản lý điểm số hiện tại, số mạng sống còn lại và cập nhật highscore vào bộ nhớ và hệ thống online
 - + Sử dụng Singleton pattern để các script khác có thể gọi để truy cập, cập nhật điểm số và lưu dữ liệu
- CupManager: Phụ trách và theo dõi spawn cup khi cốc được sử dụng hoặc cho vào Trash
- CustomerManager:
 - + Quản lý tình trạng của các ghế ngồi
 - + Spawn khách hàng sử dụng Object Pool Pattern
 - + Lưu lại đáp án đúng để sau này kiểm tra
- Customer: Khởi tạo thời gian chờ của khách hàng, theo dõi order của khách
- NormalOrder: Khởi tạo ngẫu nhiên đơn đồ uống của khách
- Trash: Xóa cốc đã cho nguyên liệu của người chơi và spawn một cốc mới
- RestockMachine: Reset lại số lượng nguyên liệu được chọn về số lượng max ban đầu sau một khoảng thời gian
- GameEvent:
 - + Sử dụng Observer Pattern.
 - + Các script trong game sẽ sử dụng GameEvent như bên trung gian để gửi các lời gọi hàm, vv
- LeanTouch: Hệ thống quản lý các input của người chơi trên màn hình cảm ứng như touch, drag, swipe, hold,...
- GameManager:
 - + Theo dõi thời gian một màn game đã chạy để thay đổi độ khó của trò chơi
 - + Độ khó trò chơi theo đổi các khía cạnh như: số khách hàng gọi đồ trong cùng một lúc, thời gian khách hàng xuất hiện, thời gian chờ khách hàng

7.2 Một số biểu đồ khác

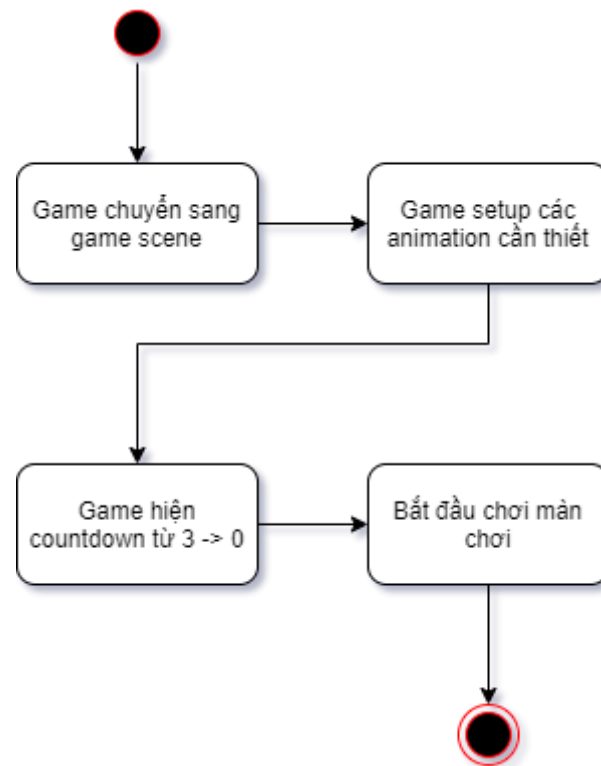
- Biểu đồ cho Start Game



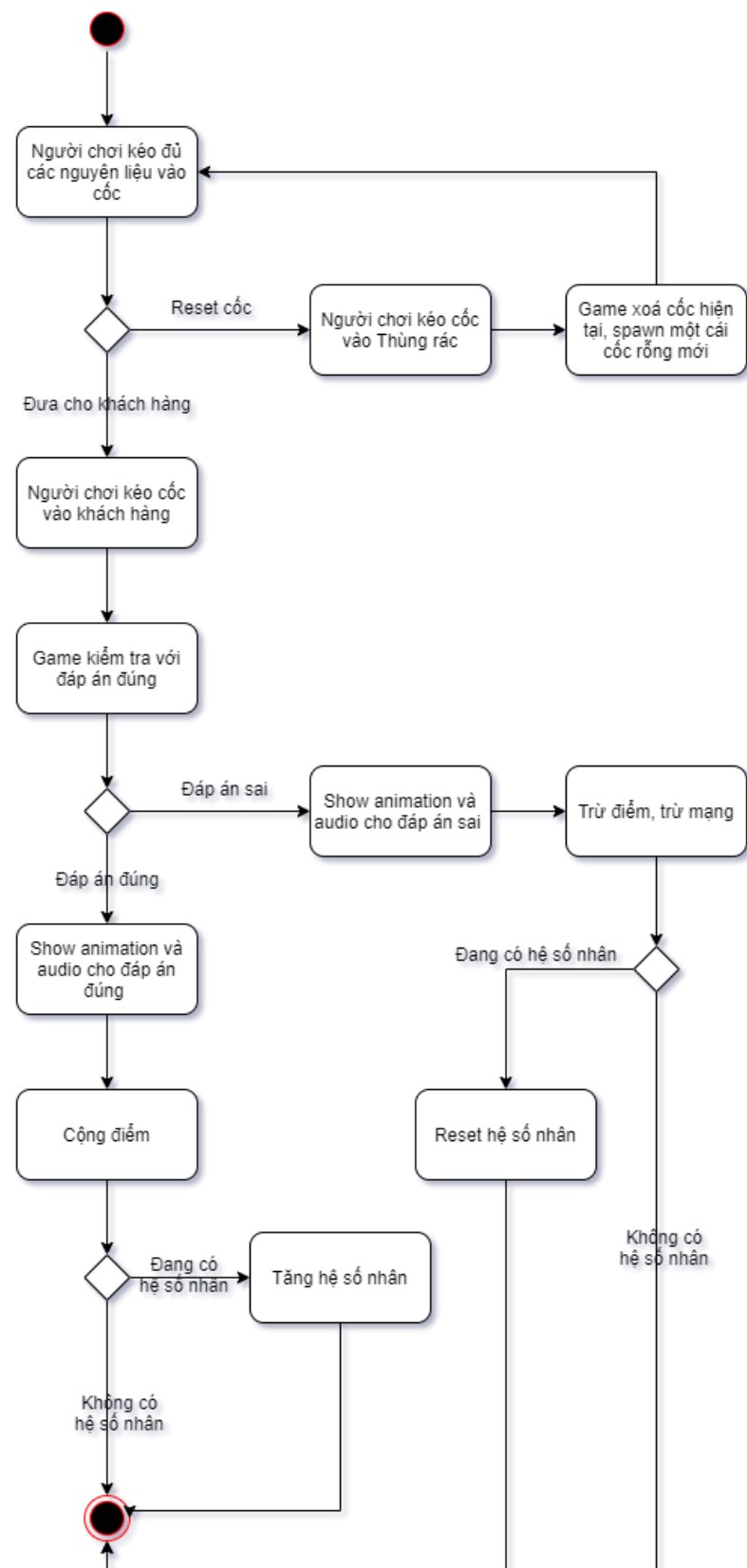
- Biểu đồ Options



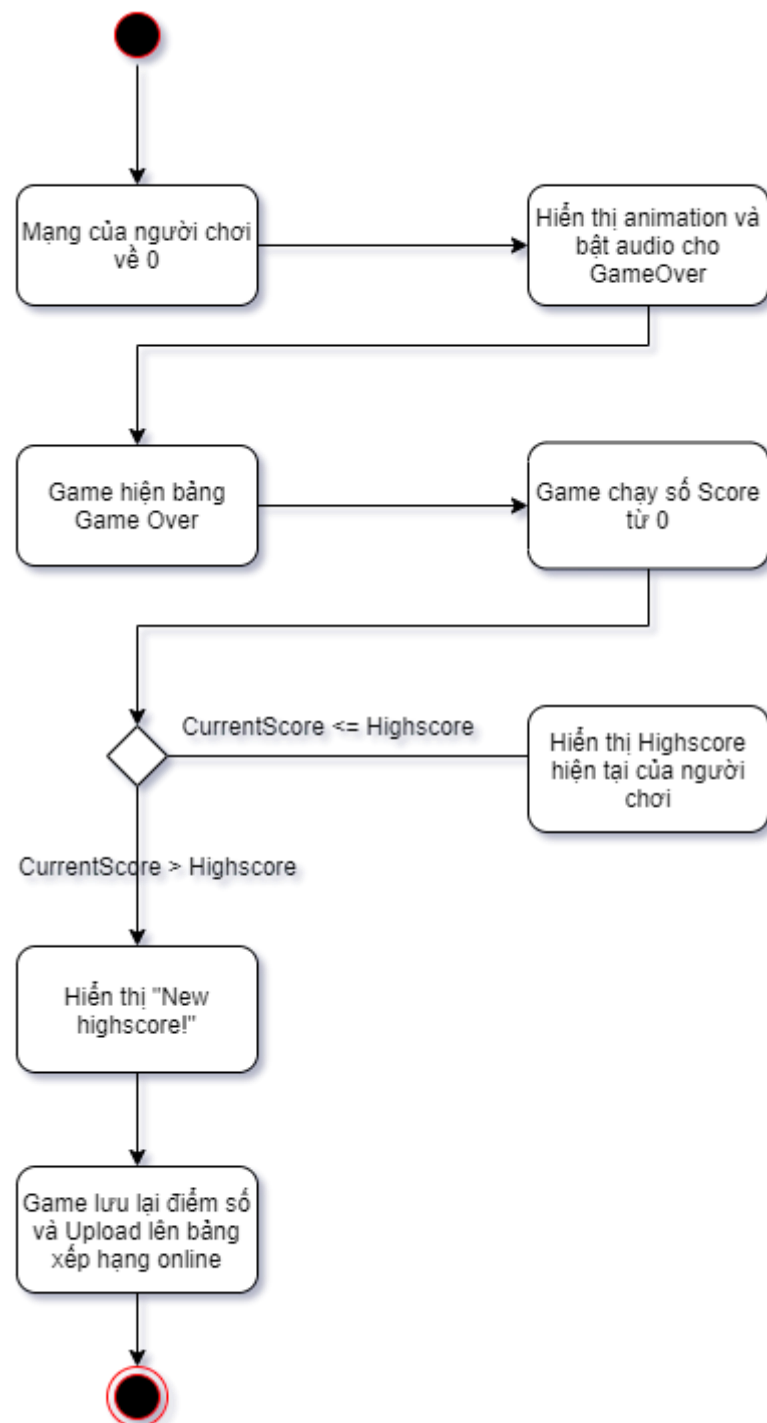
- Biểu đồ Đầu Game Scene



- Biểu đồ Trả đồ cho khách



- Biểu đồ Game Over

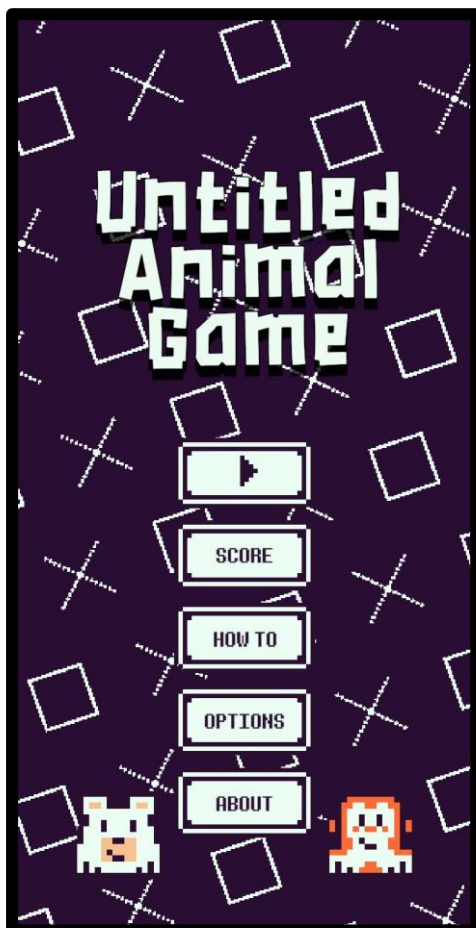


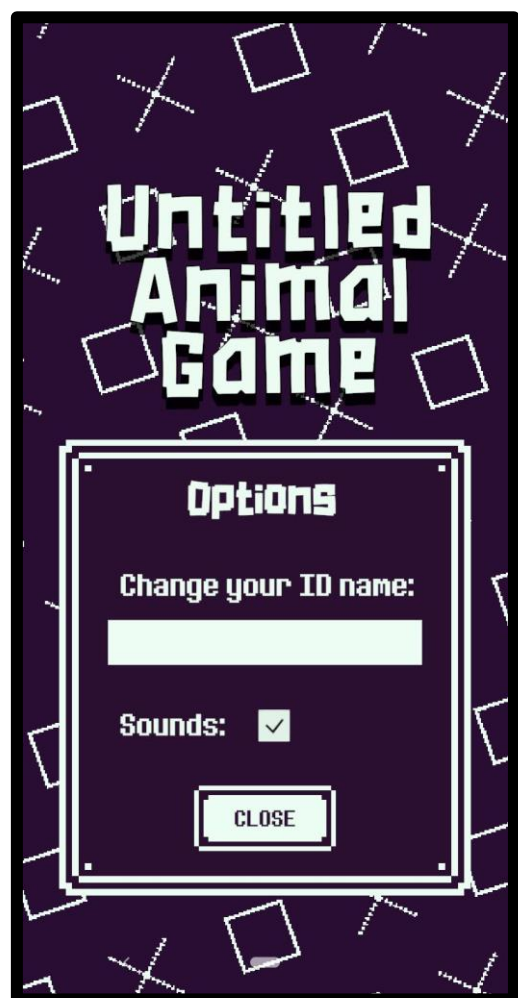
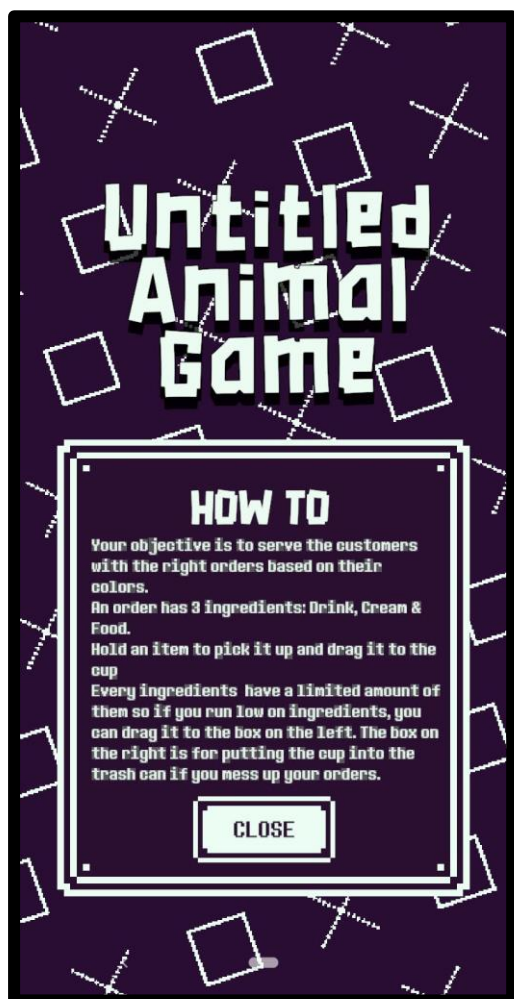
7.3 Các Unity Assets được sử dụng trong dự án:

1. LeanTouch: Giúp giả lập các input từ chuột thành cảm ứng điện thoại để dễ dàng test ở trong Unity Editor
2. LeanTween: Giúp tạo các animation trong Unity
3. LeanPool: Nhanh gọn khởi tạo Object Pool để sử dụng Object Pool Pattern
4. Dreamlo: Hỗ trợ truy cập bảng xếp hạng online và tải lên điểm số của người chơi

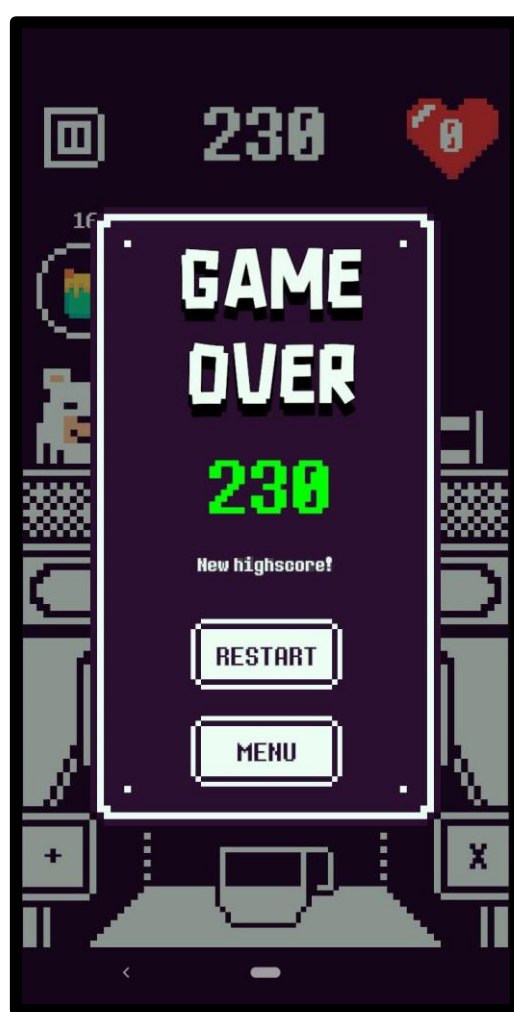
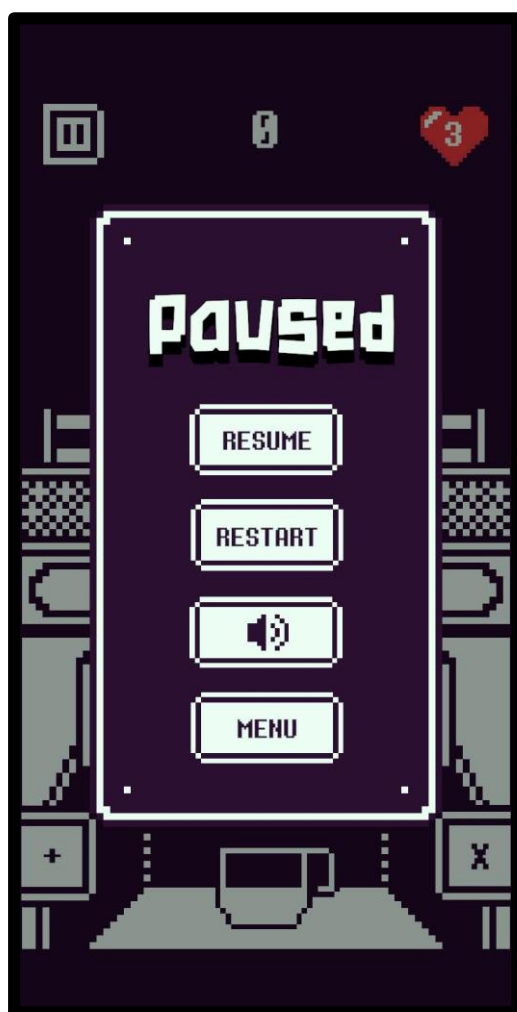
8. KẾT QUẢ THỰC HIỆN

- Sau 5 tháng tự học và thực hiện project, em đã hoàn thành được game android Untitled Animal Game bằng Unity với file APK nặng 23,016KB
- Các hình ảnh trong game:









9. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

9.1 Kết luận

- Sau khi hoàn thành đề tài em đã có thêm nhiều kiến thức về lập trình game trên Unity nói chung và lập trình game Android nói riêng. Qua việc tìm hiểu, em cũng đã tìm ra được nhiều thủ thuật và kĩ năng lập trình mới và em rất mong có thể được áp dụng chung trong các dự án tương lai.
- Tuy nhiên, tựa game vẫn còn có nhiều khuyết điểm, chẳng hạn:
 - o Còn một số bug nhỏ ở một số trường hợp hi hữu khi các event trong game xảy ra cùng một lúc
 - o Âm thanh hiệu ứng chưa đặc sắc, chưa cân bằng và chuyển đổi mượt mà
 - o Giao diện UI còn thô sơ, chưa nhạy và còn gây khó dễ cho người dùng
 - o Core game còn sơ sài, tính chơi lại chưa cao và dễ gây nhàm chán
- Tựa game này hoàn toàn có thể phát triển thành một trò chơi có nhiều khía cạnh phong phú, đa dạng khách hàng và loại đồ uống. Ngoài ra game còn có thể chau chuốt hơn về Animation và hiệu ứng cũng như chú trọng vào feedback tới người dùng. Dù sao, em mong đây có thể là một trò chơi giải trí rất hợp lí cho mọi người.

9.2 Hướng phát triển

- Triển khai và chỉnh sửa cho các nền tảng và hệ điều hành khác như IOS, WebGL
- Cải thiện thuật toán tăng độ khó phù hợp với khả năng của người chơi và các giai đoạn trong game
- Thêm các tính năng như nâng cấp, mua bán đồ để tăng tính replayable của trò chơi
- Tích hợp hệ thống leaderboard của Google Play thay vì dreamlo

10. TÀI LIỆU THAM KHẢO

- Unity Learn
- Unity Documentation
- Brackeys
- Jason Weimann
- Sebastian Lague
- Blackthornprod