

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

7/24/2017

Đồ án Flappy Plane

Môn Project Management

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Nhóm H

1. Thông tin nhóm

Website quản lý project: [Flappy Plane](#)

MSSV	Họ tên	Email
1459057	Hoàng Đức Việt	1459057@itec.hcmus.edu.vn
1459058	Trương Huỳnh Hiển Vinh	1459058@itec.hcmus.edu.vn
1459015	Hà Ngọc Gia	1459015@itec.hcmus.edu.vn
1459031	Trần Gia Linh	1459031@itec.hcmus.edu.vn
1459013	Phạm Trần Tuấn Đạt	1459013@itec.hcmus.edu.vn

Mục lục

1. Chương 1: Kế hoạch dự án “Flappy Plane”	4
1.1. Trò chơi trong lĩnh vực kỹ nghệ phần mềm	4
1.2. Về phần dự án	4
1.3. Phạm vi của trò chơi.....	4
2. Chương 2: Yêu cầu phần mềm trò chơi “Flappy Plane”.....	6
2.1. Giới thiệu.....	6
2.2. Mục đích của chương này	6
2.3. Phạm vi báo cáo	6
2.4. Sản phẩm và việc kinh doanh trò chơi.....	7
2.5. Môi trường hệ thống.....	7
2.6. Các yêu cầu.....	7
2.7. Gameplay	8
2.8. Các chương trình hỗ trợ phát triển trò chơi.....	9
2.9. Mô tả phân tích dự án Flappy plane	10
2.10. Activity Diagram	15
2.11. Lưu trữ dữ liệu	18
2.12. Biểu đồ hành vi	18
2.13. Class Diagram	20
2.14. Quản lý thay đổi yêu cầu hệ thống.....	20-21
3. Chương 3: Thiết kế và cài đặt “Flappy Plane”	22
3.1. Điều khoảng thiết kế sản phẩm	23
3.2. User Experience – UX	23
3.3. Backend Programming	23
3.4. Các tính năng trong Flappy Plane	23
3.5. Yêu cầu về phía nhân lực.....	28
3.6. Ví dụ một số scripts và class	28
4. Chương 4: TEST CASE	32
4.1. Các test case của game Flappy Plane.....	33-37
5. Chương 5: UI	38-41
6. Chương 6: Tổng kết.....	42
7. Phụ lục	44

Chương 1: Kế hoạch dự án “Flappy Plane”

Chương này bao gồm đề xuất dự án và tính khả thi của đề xuất cùng với nghiên cứu nền tảng, quan điểm sản phẩm và kinh doanh, phạm vi và một số ý tưởng ban đầu của trò chơi của chúng tôi.

1.1 Trò chơi trong lĩnh vực kỹ nghệ phần mềm

Trong 1 đồ án trò chơi điện tử thì sản phẩm sẽ là 1 trò chơi mang tính giải trí cho người dùng . Nhưng đây mới là mục đích thực sự : 1 sản phẩm về trò chơi khác hẳn với 1 phần mềm thông dụng là cung cấp cho người dùng sự giải trí khi mà người dùng họ chơi . Phần mềm chỉ là 1 phần trong dự án nó không chỉ 1 mà còn kết nối với nhiều thứ khác trong game : môi trường trong game, bối cảnh , cách chơi và nhân vật.

1.2 Về phần dự án

Nhân vật chính trong trò chơi " flappy plane" là chiếc máy bay động cơ cánh quạt, bay trong thành phố và đang cố gắng vượt qua các vật thể bay không xác định "UFO" và các tòa nhà. Ngoài ra máy bay còn phải tiếp nhiên liệu khi đang bay, cho nên phải liên tục thu thập các giọt nhiên liệu lơ lửng trên trời.

1.3. Phạm vi của trò chơi

Báo cáo này mô tả các yêu cầu của dự án Flappy Plane. Mục đích cho nghiên cứu này là để cung cấp sự liên kết giữa hình ảnh và chuyển động của các thực thể trong trò chơi này. Người chơi phải duy trì nhiên liệu và bay càng xa càng tốt, không bị các vật thể bay không xác định và tòa nhà va phải. Chúng tôi trình bày các hoạt động của trò chơi giữa các thao tác inputs, scripts và hiển thị (output). Trò chơi này không có cốt truyện cho nên chúng tôi chỉ tập trung vào cách chơi "gameplay" , đồ hoạt "graphic" , các tệp tin chứa code "scripts" và game engine.

Chương 2 : Yêu cầu phần mềm trò chơi “Flappy Plane”

Chương này mô tả về các chi tiết yêu cầu phần mềm của trò chơi Flappy Plane. Bao gồm các đặc điểm kỹ thuật, yêu cầu cụ thể và phân tích các mô hình diagram.

2.1 Giới thiệu

Trong phần này sẽ nói về các báo cáo và xác định các yêu cầu của trò chơi Flappy Plane, phạm vi và cung cấp những gợi ý cho người đọc báo cáo này.

2.2 Mục đích của chương này

Yêu cầu phần mềm "Software Requirement Specification (SRS) " sẽ cung cấp một góc nhìn tổng quan về đồ án Flappy Plane bao gồm các luồng xử lý, giao diện người dùng và trò chơi "gameplay".

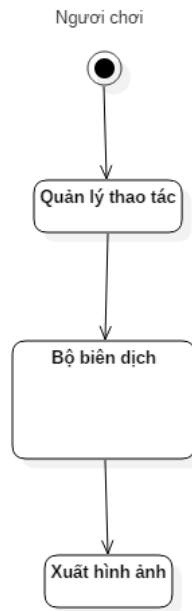
2.3 Phạm vi báo cáo

Yêu cầu phần mềm "Software Requirement Specification (SRS) " mô tả các yêu cầu chức năng và phi chức năng của dự án. Như chúng tôi đã nói lúc đầu phần mềm chỉ là 1 phần trong dự án nó không chỉ 1 mà còn kết nối với nhiều thứ khác trong game : môi trường trong game, bối cảnh , cách chơi và nhân vật. Đồ án trò chơi Flappy Plane được tạo nên bởi 5 thành viên, bao gồm 1 Quản lý dự án, 1 Developer, 2 Designer và 1 Tester. Nhóm chúng tôi sẽ tiếp tục hoàn thành dự án này và phát triển thêm cho đến khi thỏa mãn yêu cầu.

2.4 Sản phẩm và việc kinh doanh trò chơi

Đồ án này phục vụ cho mục đích học tập cho nên nhóm chúng tôi quyết định sẽ không kinh doanh trò chơi này lên AppStore hay GooglePlay.

2.5 Môi trường hệ thống



Người chơi thao tác trên điện thoại hoặc máy tính cá nhân bằng cách nhấn lên màn hình điện thoại. Nhấp chuột hoặc nhấn enter lên bàn phím máy tính.

2.6 Các yêu cầu:

2.6.1 Yêu cầu cơ bản

- Yêu cầu cơ bản bao gồm các tiêu chí được nêu lên trong cuộc họp với các bên liên quan như lập trình viên, người chơi và quản lý dự án. Sau đây là danh sách những yêu cầu thông thường của dự án:
 1. Giao diện trò chơi thân thiện với người dùng
 2. Trò chơi dễ vận hành và chạy mượt
 3. Trò chơi được xây dựng một cách chuyên nghiệp và thống nhất
 4. Đáp ứng những yêu cầu cấu hình dự kiến của máy tính và điện thoại
 5. Tiết kiệm được chi phí nhưng vẫn bảo đảm chất lượng trò chơi.
 6. Trò chơi không mang tính chất giải đố, không đòi hỏi tư duy nhiều.

2.6.2 Yêu cầu giao diện trò chơi “UX”

2.6.2.1: Giao diện người dùng

Trò chơi phải có một giao diện thân thiện và hỗ trợ cho người dùng có thể dễ dàng thực hiện những mong muốn của mình trong quá trình tương tác. Menu là một điều khá quan trọng trong tài liệu SRS, chúng tôi đã sử dụng ảnh chụp Menu trong phần hướng dẫn sử dụng Flappy Plane.

2.6.2.2: Giao diện phần cứng

Flappy Plane là một trò chơi có thể hoạt động đa nền tảng bao gồm trên PC, Android, iOS và macOS. Đối với smartphone và tablet, giao diện sẽ có phần khác so với trên PC và macOS. Dữ liệu của trò chơi đã được lưu trữ bởi Unity engine.

2.7 Gameplay

Flappy Plane là một trò chơi tương đối giống với Flappy Bird của nhà phát triển game Nguyễn Hà Đông. Nó là một trò chơi hỗ trợ nhiều hệ điều hành, có thể chạy trên Android, iOS và PC.

Khi bắt đầu trò chơi, người dùng sẽ thấy được Menu chính của game là một chiếc máy bay, hình nền chuyển động và nút thoát game.

Người chơi sẽ nhấn vào biểu tượng hình máy bay hoặc nhấn vào chữ Play game để chơi game. Sau khi đã nhấn vào 2 nút chơi game, hệ thống sẽ chuyển người dùng đến màn hình hướng dẫn chơi. Tại đây, người dùng nhấn vào nút “Tap the plane” để bắt đầu chơi.

Máy bay sẽ tự động bay về hướng từ trái sang phải và dần dần rơi xuống đất. Cho nên người chơi buộc phải bấm lên màn hình để làm cho máy bay bay lên. Nhiên liệu sẽ bị trừ dần trong khi bay là bị trừ 10 đơn vị nhiên liệu khi người dùng nhấn vào màn hình. Trong khi bay, người dùng sẽ gặp các giọt nhiên liệu. Các giọt này cung cấp 10 đơn vị nhiên liệu cho máy bay giúp người chơi trụ được lâu hơn.

Trong khi bay người dùng sẽ gặp 2 chướng ngại vật là UFO “vật thể bay không xác định” hay trong dân gian còn gọi là đĩa bay. Thứ 2 là các tòa nhà di chuyển lên xuống. Người dùng bắt buộc phải né các chướng ngại vật này. Khi va chạm với 2 vật thể này hoặc hết nhiên liệu, trò chơi sẽ kết thúc và hiện ra bản điểm. Điểm sẽ được tính bằng quãng đường bay xa nhất theo đơn vị đo lường là mét “m”.

Bảng điểm bao gồm khoản cách mới đây “Last distance” và khoản cách xa nhất “Longest distance”. Khi mới bắt đầu chơi, khoản cách xa nhất sẽ là 0 m và khi kết thúc trò chơi, khoản cách xa nhất sẽ bằng với khoản cách mới đây. Khoản cách xa nhất sẽ được gán bằng với khoản cách mới đây nếu nhỏ hơn.

Người chơi có thể nhấn nút Pause để tạm dừng trò chơi và nhấn resume để chơi tiếp hay chọn Reset game để chơi lại từ đầu.

2.8 Các chương trình hỗ trợ phát triển trò chơi

“Flappy Plane” được tạo nên bởi một loạt các phần mềm hỗ trợ lĩnh vực phát triển game và thiết kế đồ họa.

STT	Phần mềm	Nhà cung cấp	Phiên bản	Chức năng
1	Unity 3D	Unity Community	5.0.1f	Sử dụng Unity để phát triển nên trò chơi flappy bird và hỗ trợ đa nền tảng.
2	Adobe Photoshop	Adobe	CC 2015	Vẽ sprites cho trò chơi
3	Adobe Illustrator	Adobe	CC 2015	Thiết kế UX, UI
4	TexturePackerGUI	Code&Web	4.4.0	Tạo bảng sprites để sử dụng animation trong Unity
5	Visual Studio Community 2017	Microsoft	15.2 Release	Bộ biên dịch ngôn ngữ C# cho engine Unity
6	Android Studio	Google	2.3.3	Tạo smartphone ảo để test game
7	Android SDK	Google	V.23	Sử dụng SDK để build game cho Android
8	Adobe Audition	Adobe	CC 2015	Chỉnh sửa âm thanh cho game Flappy Plane
9	Git	Github	2.13	Dùng git command line gỡ bỏ caches và Unity của Unity. Tạo file .gitignore
10	Github Desktop	Github	0.7.0	Quản lý source code
11	Zenhub	Zenhub	2.34.64	Quản lý tiến độ project
12	Microsoft Project	Mircosoft	2016	Quản lý tiến độ project
13	Microsoft Word	Microsoft	2016	Hỗ trợ viết báo cáo
14	Xcode	Apple	V.8	Hỗ trợ xây dựng game cho iOS và Unity dự trên source code có sẵn của Unity.
15	macOS Sierra	Apple	10.12.6	Hỗ trợ chạy Xcode
16	VMWARE	VMWARE	12.5.0	Hỗ trợ tạo máy ảo cho macOS Sierra.

2.9 Mô tả phân tích dự án Flappy plane

Đây là phần mô tả yêu cầu phần mềm của dự án chúng tôi bằng mô hình kỹ thuật tích hợp.

1. Mô hình kịch bản

Mô hình kịch bản mô tả cách người dùng tương tác với hệ thống và trình tự cụ thể các hoạt động xảy ra trong quá trình vận hành games.

2. Kịch bản sử dụng:

Bảng tóm tắt các trường hợp sử dụng của hệ thống được tạo ra trên khung nhìn UX của trò chơi được thể hiện như sau:

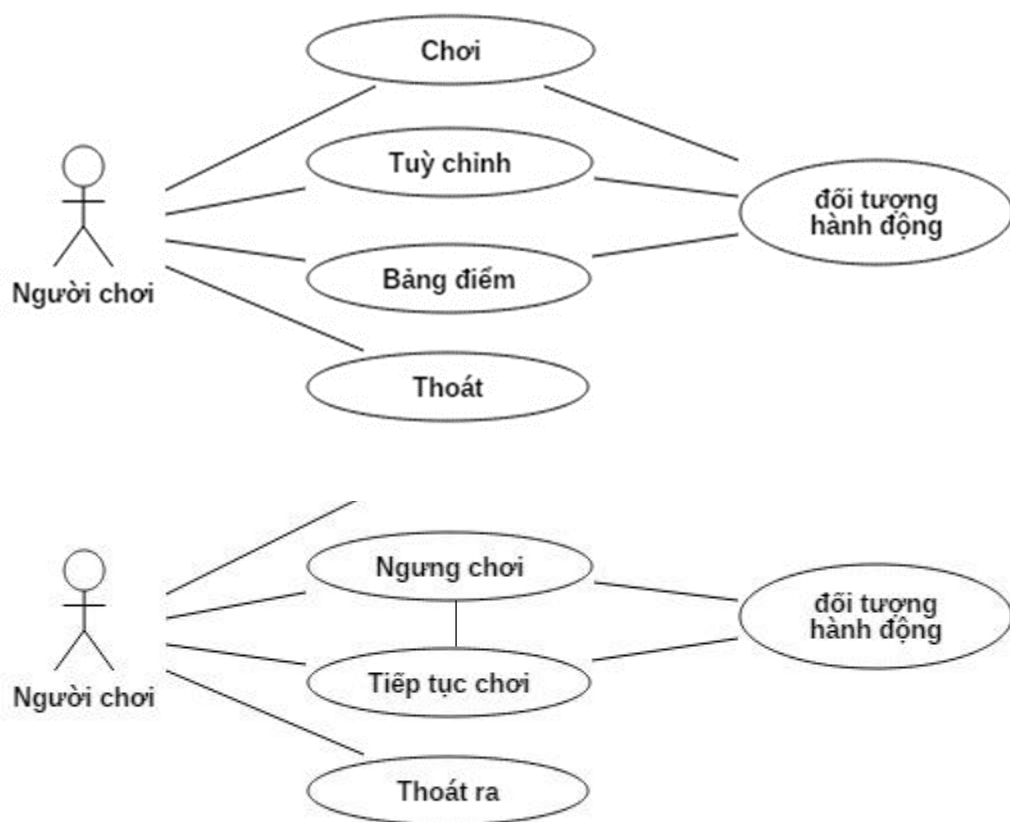
Flappy plane	Chơi	Lượt chơi mới "new game"
		Ngừng chơi "pause"
		Tiếp tục chơi "resume"
		Thoát ra "quit"
	Tuỳ chỉnh	Mở menu tùy chọn "Setting"
		Bật/ tắt nhạc và âm thanh "Play music"
	Bảng điểm	Xem bảng điểm "GameOverPanel"
	Thoát	

3. Use Case Diagram

a. Level 0



b. Level 1



c. Level 2

A. New Game

Use case ID	UC-001
Tên use case	New game
Tóm tắt	Người chơi đã vào game Flappy Plane
Tác nhân	Người chơi
Mô tả	Để bắt đầu lượt chơi mới từ đầu
Điều kiện tiên quyết	<ol style="list-style-type: none"> 1. Hệ thống hỗ trợ cấu hình trò chơi 2. Phần mềm và màn hình trò chơi hoạt động 3. Người chơi muốn bắt đầu lượt chơi mới. 4. Người chơi khởi động game
Kịch bản chính	<ol style="list-style-type: none"> 1. Khởi động game 2. Vào menu chính của trò chơi 3. Nhấn vào biểu tượng máy bay 4. Bắt đầu lượt chơi mới
Kịch bản phụ	<ol style="list-style-type: none"> 1. Game không khởi động được 2. Hiện dialog báo lỗi 3. Game không vào màn hình chơi 4. Hiện dialog báo lỗi
Ràng buộc phi chức năng	Giao diện đẹp, dễ sử dụng. Người chơi có thể sử dụng dễ dàng ở lần đầu sử dụng.

b.

Ngừng chơi

Use case ID	UC-002
Tên use case	Ngừng chơi "Pause"
Tóm tắt	Người chơi nhấn nút pause
Tác nhân	Người chơi
Mô tả	Người dùng ngừng trò chơi vì có việc
Điều kiện tiên quyết	<ol style="list-style-type: none"> 1. Trò chơi đã được bắt đầu 2. Trò chơi hỗ trợ việc ngừng thời gian hiện hành
Kịch bản chính	<ol style="list-style-type: none"> 1. Người chơi đã khởi động game 2. Người chơi đã chọn New Game 3. Người chơi đã nhấn nút pause
Kịch bản phụ	
Ràng buộc phi chức năng	Giao diện đẹp, dễ sử dụng. Người chơi có thể sử dụng dễ dàng ở lần đầu sử dụng.

c. Tiếp tục chơi

Use case ID	UC-003
Tên use case	Tiếp tục chơi "Resume"
Tóm tắt	Người chơi nhấn nút Resume
Tác nhân	Người chơi
Mô tả	Người dùng muốn chơi tiếp
Điều kiện tiên quyết	1. Trò chơi đã được dừng 2. Trò chơi hỗ trợ việc tiếp tục thời gian hiện thành
Kịch bản chính	1. Người chơi đã khởi động game 2. Người chơi đã chọn New Game 3. Người chơi đã nhấn nút pause
Kịch bản phụ	
Ràng buộc phi chức năng	Giao diện đẹp, dễ sử dụng. Người chơi có thể sử dụng dễ dàng ở lần đầu sử dụng.

d. Thoát ra

Use case ID	UC-004
Tên use case	Thoát game
Tóm tắt	Thoát trò chơi Flappy Plane
Tác nhân	Người chơi
Mô tả	Người dùng muốn thoát game
Điều kiện tiên quyết	1. Trò chơi đang ở được vận hành 2. Bảng điểm phải được hiện
Kịch bản chính	1. Người chơi nút nút pause 2. Chọn nút X để thoát
Kịch bản phụ	3. Người chơi đã thua 4. Bảng điểm hiện ra 5. Người chơi nhấn nút X để thoát
Ràng buộc phi chức năng	Giao diện đẹp, dễ sử dụng. Người chơi có thể sử dụng dễ dàng ở lần đầu sử dụng.

e. Tùy chỉnh "Settings"



f. Mở / đóng tùy chỉnh

Use case ID	UC-004
Tên use case	Mở đóng settings
Tóm tắt	Người chơi nhấn nút setting ở màn hình menu
Tác nhân	Người chơi
Mô tả	Người dùng muốn sử dụng chức năng setting để tắt mở nhạc
Điều kiện tiên quyết	1. Trò chơi đã được khởi động 2. Màn hình menu đã được nạp
Kịch bản chính	3. Người chơi click vào setting 4. Tích chọn Play music 5. Bỏ chọn Play music
Kịch bản phụ	
Ràng buộc phi chức năng	Giao diện đẹp, dễ sử dụng. Người chơi có thể sử dụng dễ dàng ở lần đầu sử dụng.

g. Bảng điểm



Use case ID	UC-005
Tên use case	Xem bảng điểm
Tóm tắt	Người chơi muốn xem điểm hiện tại
Tác nhân	Người chơi
Mô tả	Người chơi mở bảng điểm xem
Điều kiện tiên quyết	1. Người chơi đã chọn New game 2. Màn hình chơi game đã nạp
Kịch bản chính	1. Người chơi chọn pause 2. Bảng điểm hiện lên
Kịch bản phụ	1. Người chơi thua 2. Bảng điểm hiện lên
Ràng buộc phi chức năng	Giao diện đẹp, dễ sử dụng. Người chơi có thể sử dụng dễ dàng ở lần đầu sử dụng.

2.10 Activity Diagram



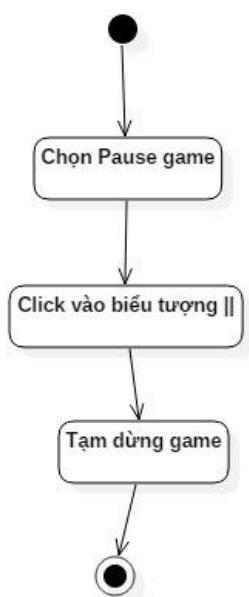
Hình 2.5.1 : Chọn new game vào game



Hình 2.5.2 : Chọn tắt mở nhạc trong setting



Hình 2.5.3: Thoát game



Hình 2.5.4: Tạm dừng game



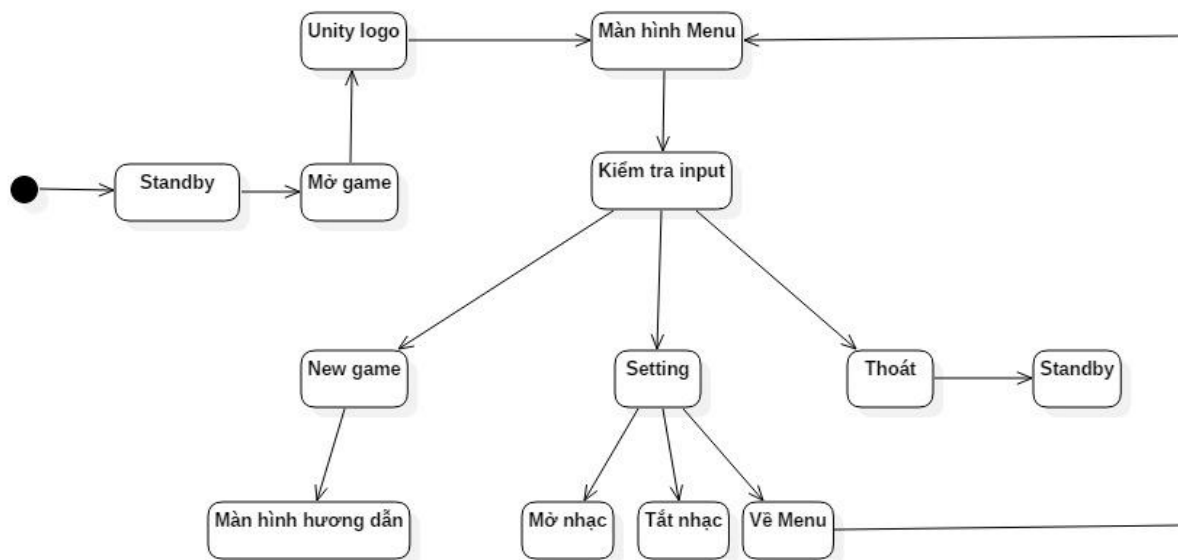
Hình 2.5.5: Bắt đầu lại trò chơi

2.11 Lưu trữ dữ liệu

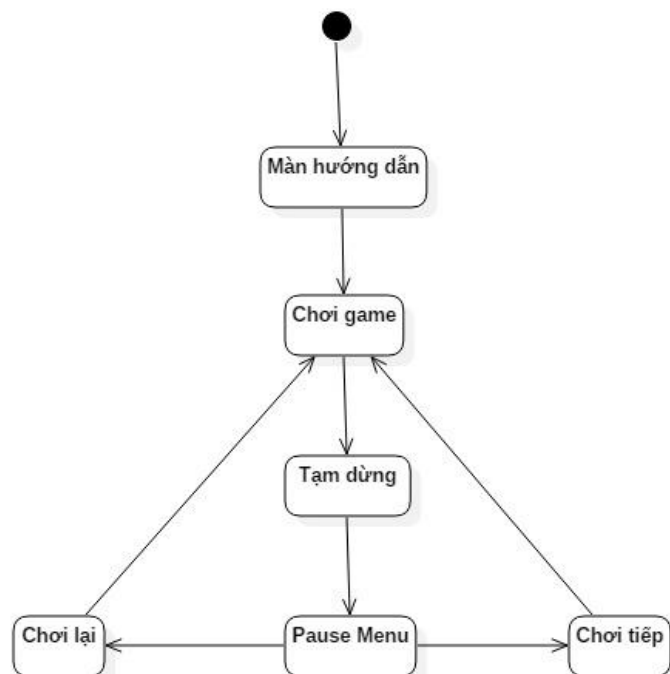
Tất cả đối tượng trong trò chơi được lưu trữ bằng PlayerPrefs "Player Preference" của Engine Unity. Mặc dù có rất nhiều đối tượng trong trò chơi như máy bay, UFO, giọt nhiên liệu, tòa nhà nhưng chúng tôi chỉ lưu: "trạng thái nhạc" có mở hay không mở và quan trọng hơn là điểm số. Để tăng tính ổn định và tốc độ cho trò chơi, chúng tôi không dùng mô hình dữ liệu phức tạp hay các database bên thứ 3 như SQL, Oracle, noSQL Cho nên mô hình dữ liệu cho dự án game Flappy Plane này là dư thừa.

2.12 Biểu đồ hành vi

Dưới đây, chúng tôi sẽ mô tả từng hành vi mà hệ thống vận hành, thông qua biểu đồ "State Chart"

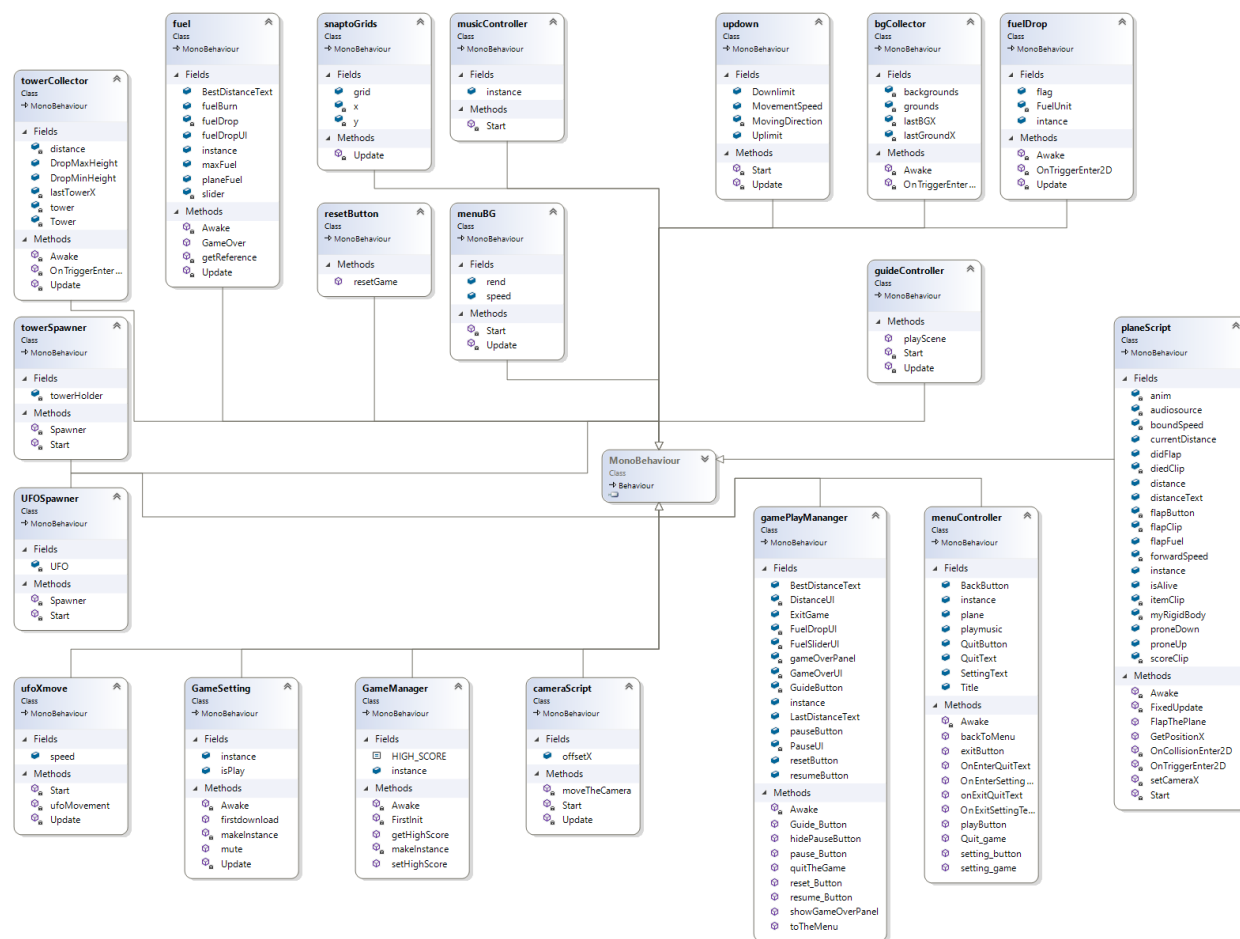


Hình 2.7.1: Menu Level state chart



Hình 2.7.2: Gameplay State chart

2.13 Class Diagram



Hình 2.8.1: Sơ đồ class diagram của game Flappy Plane

2.14 Quản lý thay đổi yêu cầu trong hệ thống

Chúng tôi có ý định hoàn thiện các chức năng trong trò chơi đã được liệt kê trong tài liệu SRS "Yêu cầu phần mềm". Vì trò chơi được thiết kế đa nền tảng nên các vấn đề xảy ra bugs, glitches là điều khó tránh khỏi khi cập nhật các tính năng mới. Nếu người chơi phát hiện bugs, errors hoặc glitches hay các trường hợp crash, xin vui lòng liên hệ tới các địa chỉ email này và mô tả các trường hợp lỗi cho nhà phát triển. Chúng tôi sẽ khắc phục sớm nhất có thể kể từ khi đã nhận được email từ phía người dùng.

STT	EMAIL	TÊN
1	1459057@itec.hcmus.edu.vn	Hoàng Đức Việt
2	1459058@itec.hcmus.edu.vn	Trương Huỳnh Hiến Vinh
3	1459031@itec.hcmus.edu.vn	Trần Gia Linh
4	1459013@itec.hcmus.edu.vn	Phạm Trần Tuấn Đạt
5	1459015@itec.hcmus.edu.vn	Hà Ngọc Gia

2.14.1: Bug & Glitches

Tại đây, người dùng có thể liên lạc với nhà phát triển để thông báo lỗi hay các trường hợp game bị crash. Người dùng cần mô tả rõ các bug & glitches để nhà phát triển có thể xử lý kịp thời. Địa chỉ email và tên để liên hệ đã được cung cấp đầy đủ trong mục 2.9. Quản lý thay đổi yêu cầu hệ thống.

2.14.2: Patches

Khi hệ điều hành PC, macOS, iOS & Android được nâng cấp, vấn đề cung cấp bản vá cho hệ điều hành mới là điều cần thiết để giúp trò chơi được chạy một cách mượt mà, tương thích với phần cứng và hệ điều hành mới.

Chương 3: Thiết kế và cài đặt “Flappy Plane”

Chương này sẽ bao quát tất cả các báo cáo liên quan đến phần thiết kế các tính năng và việc thực hiện các tính năng.

3.1 Điều khoảng thiết kế sản phẩm

Tất cả các studio, enterprise hay công ty gia công game, ứng dụng đều phải nhấn mạnh 2 mảng này bao gồm: UX (User Experience) và Backend Programming (System).

3.2 User Experience – UX

Trong công việc phát triển game, mảng thiết kế UX đóng một phần quan trọng trong một trò chơi. Nó giúp cho người dùng tương tác, điều khiển trò chơi dễ dàng. Khiến cho người dùng cảm thấy thích thú khi sử dụng ứng dụng hay một trò chơi. Một ứng dụng tốt không chỉ có một backend được optimized mà còn cần có một giao diện người dùng bắt mắt để người dùng được trải nghiệm phần mềm một cách tốt nhất.

3.3 Backend Programming

Backend là phần hỗ trợ front-end, nếu không có backend, trò chơi chỉ là một bức tranh tĩnh lặng. Backend phục trách phần database, các đối tượng và các tính năng bên trong hệ thống game. Đối với trò chơi Flappy Plane, backend sẽ được engine của Unity đảm nhận việc biên dịch code.

3.4 Các tính năng trong Flappy Plane

- 3.4.1 Giọt nhiên liệu
- 3.4.2 Thanh nhiên liệu
- 3.4.3 Máy bay
- 3.4.4 Các vật thể bay không xác định UFO
- 3.4.5 Các tòa nhà
- 3.4.6 Điểm số
- 3.4.7 Bảng điểm
- 3.4.8 Màn hình tạm dừng
- 3.4.9 Màn hình tùy chỉnh
- 3.4.10 Thoát game
- 3.4.11 Bắt đầu chơi mới
- 3.4.12 Reset game

3.4.1 Giọt nhiên liệu

- 3.4.1.1 Mô tả và độ ưu tiên
Các giọt nhiên liệu sẽ được nhân bản liên tục cho người chơi nhận để tiếp nhiên liệu cho máy bay có khả năng bay lâu hơn. Nếu không có các giọt này, máy bay sẽ mất dần nhiên liệu và game sẽ dừng lại khi nhiên liệu của máy bay bằng 0.
- 3.4.1.2 Các chuỗi phản ứng
Bước 1: Người chơi sẽ thấy được các giọt nhiên liệu xuất hiện và di chuyển về phía máy bay theo hướng từ phải sang trái và bay lên xuống.
Bước 2: Nếu người chơi nhận các giọt nhiên liệu này, thanh nhiên liệu của máy bay sẽ được cộng lên 10 đơn vị nhiên liệu.
Bước 3: Nếu không nhận được, máy bay vẫn có thể bay tiếp và thanh nhiên liệu vẫn bị giảm dần theo thời gian.
- 3.4.1.3 Yêu cầu tính năng:
REQ 1: Máy bay phải được cộng 10 đơn vị nhiên liệu cho máy bay.
REQ 2: Các giọt nhiên liệu phải hủy ngay sau khi máy bay nhận được.
REQ 3: Các giọt nhiên liệu phải tự hủy sau 4 giây khi bị máy bay bỏ qua.

3.4.2 Thanh nhiên liệu

- 3.4.2.1 Mô tả và độ ưu tiên
Thanh nhiên liệu là một phần UI HUD trên màn hình chơi game. Thanh nhiên liệu này hiển thị lượng nhiên liệu đang có trong máy bay. Nó sẽ giảm dần theo thời gian. Nếu thanh này trở về 0 tức không còn vệt màu đỏ, đồng nghĩa trò chơi sẽ dừng lại.
- 3.4.2.2 Các chuỗi phản ứng
Bước 1: Người chơi nhận lấy các giọt nhiên liệu
Bước 2: Thanh nhiên liệu hiển thị bằng các tăng chiều dài của vệt màu đỏ
Bước 3: Nếu người chơi không nhận được thì vệt màu đỏ sẽ giảm chiều dài cho tới khi hết.
- 3.4.2.3 Yêu cầu tính năng
REQ 1: Thanh nhiên liệu đỏ phải giảm dần theo thời gian.
REQ 2: Thanh nhiên liệu đỏ phải tăng chiều dài ngay sau khi máy bay nhận được giọt nhiên liệu

3.4.3 Máy bay

- 3.4.3.1 Mô tả và độ ưu tiên
Đối tượng chính trong trò chơi là máy bay. Khi người chơi tap màn hình thì máy bay sẽ bay lên với độ nghiêng là 45 độ. Máy bay sẽ di chuyển liên tục theo trục X từ trái sang phải. Máy bay sẽ dừng khi hết nhiên liệu
- 3.4.3.2 Các chuỗi phản ứng
Bước 1: Người dùng click chuột "trên PC" hoặc bấm vào màn hình điện thoại để điều khiển máy bay.
- 3.4.3.3 Yêu cầu tính năng
REQ 1: Máy bay phải bay từ trái sang phải liên tục.
REQ 2: Máy bay phải bay lên liên tục khi người dùng bấm & click chuột liên tục.
REQ 3: Máy bay và nhiên liệu phải đồng bộ với nhau khi bay và khi click chuột.

3.4.4 Các vật thể bay không xác định UFO

- 3.4.4.1 Mô tả và độ ưu tiên
Các UFO sẽ xuất hiện liên tục như các giọt nhiên liệu nhưng nó sẽ là các vật cản người chơi. Khi máy bay tiếp xúc với các vật thể này, game sẽ dừng. Các UFO sẽ di chuyển y như các giọt nhiên liệu.
- 3.4.4.2 Các chuỗi phản ứng
Bước 1: Điều khiển máy bay tránh các UFO
Bước 2: Nếu va phải các UFO, dừng game và hiện bảng điểm
- 3.4.4.3 Yêu cầu tính năng
REQ 1: Các ufo phải xuất hiện sau 3 giây và tự hủy sau khi đã vượt qua ranh giới màn hình.
REQ 2: Các ufo phải bắt được sự kiện va chạm vật lý với máy bay
REQ 3: Các ufo phải di chuyển theo trục X từ phải sang trái và lên xuống.

3.4.5 Các tòa nhà

- 3.4.5.1 Mô tả và độ ưu tiên
Các tòa nhà sẽ xuất hiện như UFO và sẽ là vật cản người chơi.
- 3.4.5.2 Các chuỗi phản ứng
Bước 1: Điều khiển máy bay bay qua các tòa nhà
Bước 2: Nếu va phải các tòa nhà này, dừng game và hiện bảng điểm
- 3.4.5.3 Yêu cầu tính năng
REQ 1: Các tòa nhà phải xuất hiện sau 3 giây và tự hủy sau khi đã vượt qua ranh giới màn hình.

REQ 2: Các tòa nhà phải bắt được sự kiện va chạm vật lý với máy bay.
REQ 3: Các tòa nhà phải di chuyển theo trục X từ phải sang trái và lên xuống.

3.4.6 Điểm số

- 3.4.6.1 Mô tả và độ ưu tiên
Điểm số này thể hiện chiều dài quãng đường mà máy bay đi được và có đơn vị đo là mét "m". Điểm số sẽ tăng dần theo thời gian và phụ thuộc vào vị trí của máy bay.
- 3.4.6.2 Các chuỗi phản ứng
Bước 1: Điểm số tăng dần khi máy bay còn bay.
Bước 2: Điểm số lưu lại và mất khi người dùng pause hay bị game over.
- 3.4.6.3 Yêu cầu tính năng
REQ 1: Điểm số phải thay đổi liên tục.
REQ 2: Điểm số phải lưu vào data object khi game over

3.4.7 Bảng điểm

- 3.4.7.1 Mô tả và độ ưu tiên
Bảng điểm cho thấy quãng đường mới nhất và quãng đường xa nhất. Bảng điểm cũng cho phép người dùng chơi tiếp hay thoát game thông qua 2 nút điều khiển là X và reset.
- 3.4.7.2 Chuỗi phản ứng
Bước 1: Bảng điểm phải được hiển thị khi tạm dừng game
Bước 2: Bảng điểm phải được hiển thị khi game over
- 3.4.7.3 Yêu cầu tính năng
REQ 1: Bảng điểm phải lưu được điểm số
REQ 2: 2 nút reset và quit phải hoạt động

3.4.8 Màn hình tùy chỉnh

- 3.4.8.1 Mô tả và độ ưu tiên
Màn hình tùy chỉnh sẽ cho phép người dùng bật tắt nhạc trong game.
- 3.4.8.2 Chuỗi phản ứng
Bước 1: Click vào setting
Bước 2: Tích chọn play music hoặc bỏ chọn
- 3.4.8.3 Yêu cầu tính năng
REQ 1: Phải tắt mở nhạc được
REQ 2: Phải lưu được trạng thái nhạc khi thoát game và chuyển qua màn hình gameplay

3.4.9 Thoát game

- 3.4.9.1 Mô tả và độ ưu tiên
Nút thoát game sẽ đảm nhận việc thoát game, đưa người dùng về màn hình Desktop hoặc homescreen của điện thoại.
- 3.4.9.2 Các chuỗi phản ứng
Bước 1: Bấm Quit khi đang ở menu game "chỉ có trên PC"
Bước 2: Bấm Quit khi đang ở Bảng điểm để thoát game "chỉ có trên PC"
- 3.4.9.3 Yêu cầu tính năng
REQ 1: Nút quit phải thoát hoàn toàn trò chơi

3.4.10 Bắt đầu chơi mới

- 3.4.10.1 Mô tả và độ ưu tiên
Khi đã vào được menu game, để bắt đầu chơi, người dùng chọn biểu tượng máy bay để chơi game.
- 3.4.10.2 Các chuỗi phản ứng
Bước 1: Phím vào trang menu game
Bước 2: Chọn biểu tượng máy bay
Bước 3: Tap lên màn hình để tới màn hình gameplay
- 3.4.10.3 Yêu cầu tính năng
REQ 1: Phím chuyển tới trang hướng dẫn chơi game

3.4.11 Reset game

- 3.4.11.1 Mô tả và độ ưu tiên
Sau khi đã bị game over, người dùng có thể dùng chức năng này để bắt đầu game lại từ đầu.
- 3.4.11.2 Các chuỗi phản ứng
Bước 1: Chơi game
Bước 2: Va phải UFO hoặc tòa nhà
Bước 3: Bảng điểm hiện
Bước 4: Chọn nút reset
- 3.4.11.3 Yêu cầu tính năng
REQ 1: Phím load lại gameplay

3.5 Yêu cầu về phía nhân lực

Hoạt động	Kỹ năng cá nhân	Nguồn lực trong	Nguồn lực ngoài	Biến cố
Thiết kế gameplay	Mỗi thành viên trong nhóm đều đưa ra các ý kiến về gameplay	Tất cả các thành viên trong nhóm H	Dựa trên một game mobile có sẵn "Flappy Bird"	Các thành viên còn hoan mang về độ khó của trò chơi.
Unity Engine	Các cá nhân đều phải có kỹ năng về Unity 3D	Tất cả các thành viên trong nhóm H	Unity 3D	Một số thành viên có việc bận không có thời gian tương tác với Unity 3D
Thiết kế đồ họa	Các thành viên đều phải có kỹ năng dùng Adobe photoshop hoặc Illustrator	Linh và Gia sẽ đảm nhận việc vẽ sprites	Adobe Photoshop, Adobe Illustrator	Chưa thống nhất được hình ảnh của tòa nhà
Chọn lọc âm nhạc và âm thanh	Các thành viên đều phải hiểu được code bắt sự kiện âm thanh	Việt và Đạt sẽ đảm nhận việc code các scripts	Unity 3D Visual Studio Community 2017	Chưa lưu được trạng thái của nhạc khi chuyển từ gameplay sang menu
Báo cáo	Các thành viên đều phải báo cáo khi làm việc	Các thành viên trong nhóm H sẽ thông báo các công việc đã hoàn thành và chưa hoàn thành	Microsoft Word 2013 Các tài liệu mẫu từ môn Software Engineering	Một số thành viên có việc bận không có thời gian báo cáo kết quả. Dẫn tới việc Dev và PM phải tự code một số tính năng.
Đa nền tảng	Các cá nhân phải có một ít hiểu biết về ngôn ngữ C# và cách dùng Xcode	Các thành viên trong nhóm H	iPhone 6s 64gb, macOS Sierra 10.12.6 Vmware 12.5.0	Các thành viên điều không có macbook hoặc pc không đủ cấu hình chạy vmware.

3.6 Ví dụ một số scripts và class

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class fuel : MonoBehaviour
{
    public static fuel instance;

    [SerializeField]
    private GameObject fuelDrop;

    public GameObject[] fuelDropUI;

    private Slider slider;
    public float maxFuel;
    public float planeFuel;
    public float fuelBurn;

    public Text BestDistanceText;

    //public SpriteRenderer rend;

    // Use this for initialization
    void Awake()

```

```

{
    if(instance == null)
    {
        instance = this;
    }
    getReference();
}

// Update is called once per frame
void Update()
{
    if (planeScript.instance != null)
    {
        if (planeFuel > maxFuel)
        {
            planeFuel = maxFuel;
        }
        else if (planeFuel > 0)
        {
            planeFuel -= fuelBurn * Time.deltaTime;
            slider.value = planeFuel;

        }
        else
        {
            // IMPORTANT !!!
            // uncomment this when release
            // plane stop after emptying fuel
            // planeScript.instance.isAlive = false;
            Time.timeScale = 0;
            if(planeScript.instance != null)
            {
                int distance = planeScript.instance.distance;
                GameOver(distance);
            }
            //if plane crash, stop game, hide pause button & show gameOverPanel
        }
    }
}

public void GameOver(int distance)
{
    if (gamePlayMananger.instance != null)
    {
        gamePlayMananger.instance.hidePauseButton();
        gamePlayMananger.instance.showGameOverPanel(distance);
    }

    if (distance > GameManager.instance.getHighScore())
    {
        GameManager.instance.setHighScore(distance);
    }
}

```

```

    }
    BestDistanceText.text = GameManager.instance.getHighScore().ToString("F0") + "
m";
}

void getReference()
{
    slider = GameObject.Find("fuelSlider").GetComponent<Slider>();

    slider.minValue = 0f;
    slider.maxValue = planeFuel;
    slider.value = slider.maxValue;
}

}

```

Code 3.6.1 : Code của tính năng thanh nhiên liệu

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class GameSetting : MonoBehaviour {

    public static GameSetting instance;

    public Toggle isPlay;

    // Use this for initialization
    void Awake () {
        //PlayerPrefs.DeleteKey("firstdownload");
        makeInstance();
        firstdownload();
        PlayerPrefs.GetInt("myfirstkey");
        Debug.Log(PlayerPrefs.GetInt("myfirstkey").ToString());
        print("menu load");
        Time.timeScale = 1;
        if (PlayerPrefs.GetInt("myfirstkey") == 1)
        {
            isPlay.isOn = true;
        }
        else if(PlayerPrefs.GetInt("myfirstkey") == 0)
        {
            isPlay.isOn = false;
        }
    }

    public void firstdownload()
    {
        if(!PlayerPrefs.HasKey("firstdownload"))
        {
            print("First Download key not installed");
        }
    }
}

```

```

        //PlayerPrefs.SetInt("myfirstkey", 0);
        PlayerPrefs.SetInt("firstdownload", 0);

    } else
    {
        print("First Download key installed");
    }
}

// Update is called once per frame
void Update () {

}

void makeInstance()
{
    if (instance != null)
    {
        Destroy(gameObject);
    }
    else
    {
        instance = this;
        DontDestroyOnLoad(gameObject);
    }
}

public void mute()
{
    if (isPlay.isOn == false)
    {
        if (gameObject.GetComponent<AudioSource>().pitch == 1)
        {
            gameObject.GetComponent<AudioSource>().pitch = 0;

            PlayerPrefs.SetInt("myfirstkey", 0);
            PlayerPrefs.Save();
            Debug.Log(PlayerPrefs.GetInt("myfirstkey").ToString());
        }
    }
    else if (isPlay.isOn == true)
    {
        if (gameObject.GetComponent<AudioSource>().pitch == 0)
        {
            gameObject.GetComponent<AudioSource>().pitch = 1;

            PlayerPrefs.SetInt("myfirstkey", 1);
            PlayerPrefs.Save();
            Debug.Log(PlayerPrefs.GetInt("myfirstkey").ToString());
        }
    }
}
}

```

Code 3.6.1 : Code của tính năng thanh nhiên liệu

CHƯƠNG 4: TEST CASE

Chương 4 sẽ bao gồm các test case về game flappy bird để kiểm tra tính ổn định và sự hoạt động trơn tru của trò chơi.

4.1 Các test case của game Flappy Plane

Test Case ID	TC-001
Test procedure	TC-001 sẽ kiểm tra xem trạng thái tắt mở nhạc có được lưu khi thoát game hay không
Expected Result	Trạng thái tắt mở sẽ được lưu
Actual Result	Trạng thái tắt đã được lưu và trước khi thoát game, toggle đã được đặt thành uncheck play music
Comment	
Conditional test	Chạy lại trò chơi

Test Case ID	TC-002
Test procedure	TC-002 sẽ kiểm tra xem nhiên liệu có được cộng dồn khi nhận giọt nhiên liệu hay không
Expected Result	Thanh nhiên liệu dài ra khi nhận giọt nhiên liệu
Actual Result	Thanh nhiên liệu dài ra khi nhận giọt nhiên liệu, máy bay tiếp tục bay tới
Comment	
Conditional test	Đang chơi game

Test Case ID	TC-003
Test procedure	TC-003 sẽ kiểm tra xem người dùng có thể về menu option khi đang chơi game để tắt mở nhạc được không.
Expected Result	Vẫn tắt mở nhạc được khi quay về menu option
Actual Result	Nhạc không tắt mở được, mặc dù nút toggle vẫn có thể tương tác
Comment	Do Gameobject "gameSetting" được load lại lần thứ 2 khi chuyển từ gameplay về menu option.
Conditional test	Đang chơi game và trở về menu

Test Case ID TC-003	
Test procedure	TC-003 sẽ kiểm tra xem game có thể lưu được điểm hay không
Expected Result	Game không lưu được điểm số "quãng đường mới" và "quãng đường xa nhất" khi chơi 1 hoặc nhiều lần.
Actual Result	Game đã lưu được điểm số "quãng đường mới" và "quãng đường xa nhất"
Comment	Đã được fix lại code logic "else if" bởi Developer Hoàng Việt
Conditional test	Đang chơi game và va phải UFO, tòa nhà, hết nhiên liệu hoặc tự vấp.

Test Case ID TC-004	
Test procedure	TC-004 sẽ kiểm tra xem game có thể tạm dừng hay không
Expected Result	Game không tạm dừng khi nhấn nút pause
Actual Result	Game tạm dừng khi nhấn nút pause
Comment	
Conditional test	Đang chơi game và nhấn nút pause

Test Case ID TC-005	
Test procedure	TC-005 sẽ kiểm tra xem tính năng reset game có thể đưa người dùng lại trạng thái ban đầu khi bắt đầu chơi mới hay không
Expected Result	Game không reset mà vẫn chơi
Actual Result	Game reset được và đưa người dùng lại vị trí đầu tiên.
Comment	
Conditional test	Đang chơi game và nhấn nút reset

Test Case ID TC-006	
Test procedure	TC-006 sẽ kiểm tra xem tính năng thoát game có tác dụng hay không
Expected Result	Game không thoát được hoặc crash lưng chừng

Actual Result	Game thoát được và đưa người chơi về Desktop hoặc home screne trên điện thoại.
Comment	
Conditional test	Đang chơi game và nhấn nút pause để hiện nút thoát "trên PC" hoặc nhấn nút home trên điện thoại.

Test Case ID	TC-006
Test procedure	TC-006 sẽ kiểm tra xem tính năng thoát game có tác dụng hay không
Expected Result	Game không thoát được hoặc crash lưng chừng
Actual Result	Game thoát được và đưa người chơi về Desktop hoặc home screne trên điện thoại.
Comment	
Conditional test	Đang chơi game và nhấn nút pause để hiện nút thoát "trên PC" hoặc nhấn nút home trên điện thoại.

Test Case ID	TC-007
Test procedure	TC-007 sẽ kiểm tra xem máy bay có bay lên khi người dùng nhấn nút enter hay click chuột, tap vào màn hình điện thoại
Expected Result	Máy bay bay lên được khi nhận input từ bàn phím và màn hình điện thoại.
Actual Result	Máy bay đã bay lên được khi nhận input từ bàn phím và màn hình điện thoại.
Comment	
Conditional test	Đang chơi game và nhấn nút enter, click chuột hoặc nhấn vào màn hình điện thoại.

Test Case ID	TC-008
Test procedure	TC-008 sẽ kiểm tra xem máy bay có va chạm với tòa nhà được hay không
Expected Result	Máy bay va chạm được với tòa nhà
Actual Result	Máy bay va chạm được với tòa nhà
Comment	
Conditional test	Đang chơi game và nhấn nút pause để hiện nút thoát "trên PC" hoặc nhấn nút home trên điện thoại. Để cho máy bay chạm vào tòa nhà.

Test Case ID TC-009	
Test procedure	TC-009 sẽ kiểm tra xem UFO có va chạm với máy bay hay không
Expected Result	Máy bay va chạm được với UFO
Actual Result	Máy bay va chạm được với UFO
Comment	
Conditional test	Đang chơi game và nhấn nút pause để hiện nút thoát "trên PC" hoặc nhấn nút home trên điện thoại. Để cho máy bay chạm vào UFO.

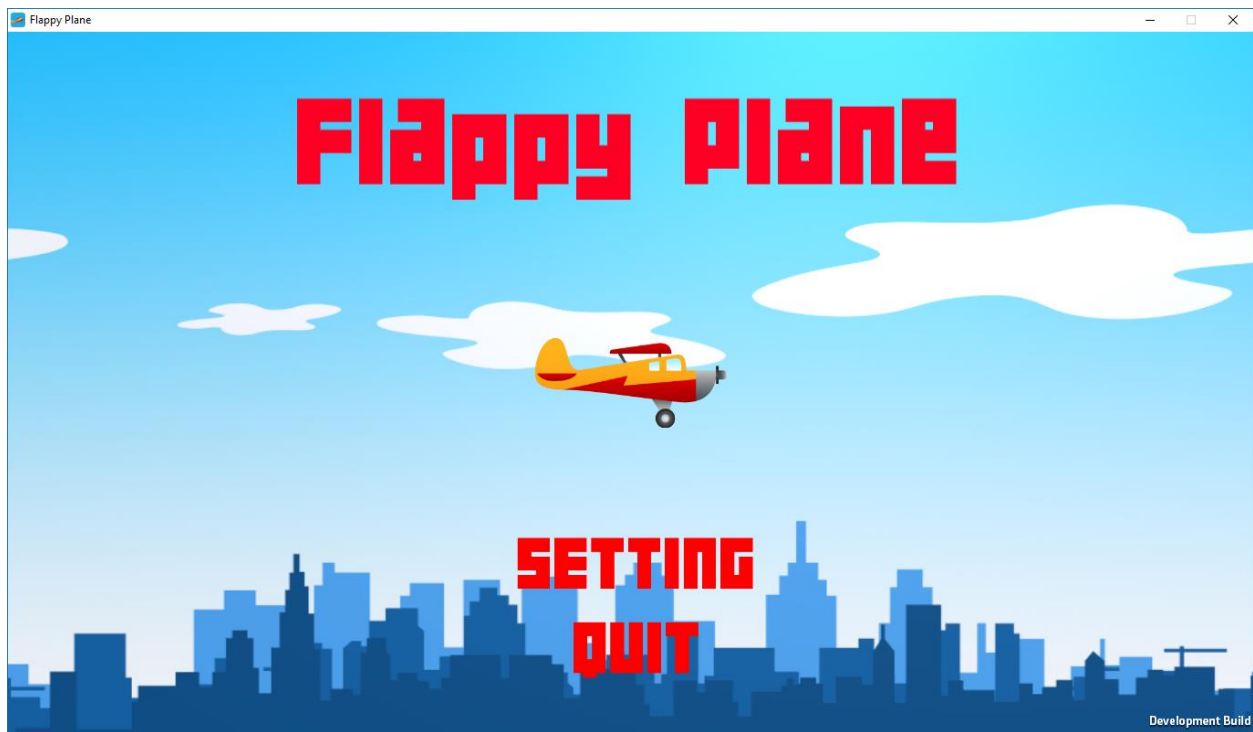
Test Case ID TC-010	
Test procedure	TC-010 sẽ kiểm tra xem UFO có tự hủy khi vượt qua ranh giới màn hình hay không.
Expected Result	UFO không tự hủy, vẫn tiếp tục di chuyển theo trục X
Actual Result	UFO tự hủy sau 3 giây vượt qua ranh giới màn hình.
Comment	
Conditional test	Đang chơi game và nhấn nút pause để hiện nút thoát "trên PC" hoặc nhấn nút home trên điện thoại. Để cho máy bay tránh UFO. Chỉ test trên UnityEditor.

Test Case ID TC-011	
Test procedure	TC-011 sẽ kiểm tra xem giọt nhiên liệu có tự hủy khi vượt qua ranh giới màn hình hay không
Expected Result	Giọt nhiên liệu tự hủy sau 3 giây vượt qua ranh
Actual Result	UFO tự hủy sau 3 giây vượt qua ranh giới màn hình.
Comment	
Conditional test	Đang chơi game và nhấn nút pause để hiện nút thoát "trên PC" hoặc nhấn nút home trên điện thoại. Để cho máy bay vượt qua giọt nhiên liệu "không nhận lấy". Chỉ test trên UnityEditor.

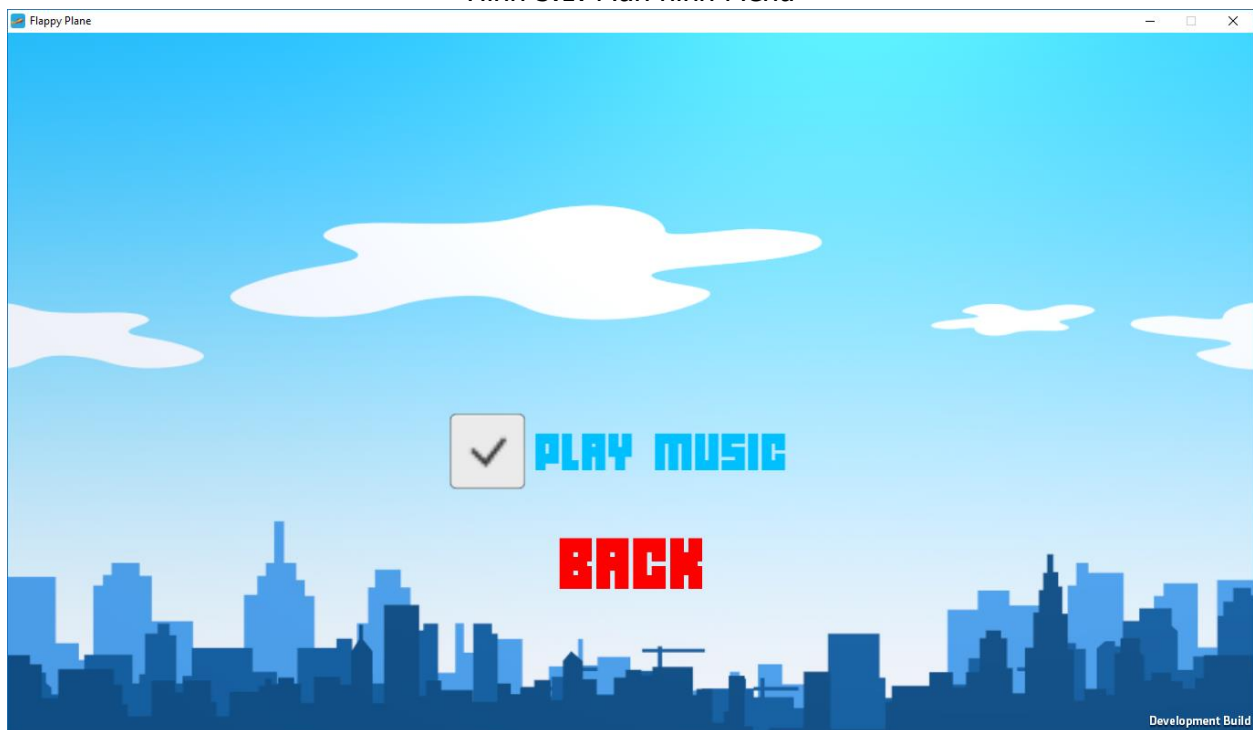
Test Case ID TC-012	
Test procedure	TC-012 sẽ kiểm tra xem game flappy plane có đạt 60 fps trên tất cả các thiết bị hay không
Expected Result	Flappy plane đạt 60 fps trên tất cả các thiết bị
Actual Result	Chỉ có PC và MAC đạt 60 fps
Comment	Application.targetFrameRate để giải quyết
Conditional test	Đã vào chơi game

Test procedure TC-013	
Expected Result	TC-013 sẽ kiểm tra xem game flappy plane có chạy trên iPhone, iPad, iPod hay không
Actual Result	Khi chạy trên IPAD, các UI bị mất do không có responsive
Comment	
Conditional test	Khởi động game trên iPhone, iPad, iPod

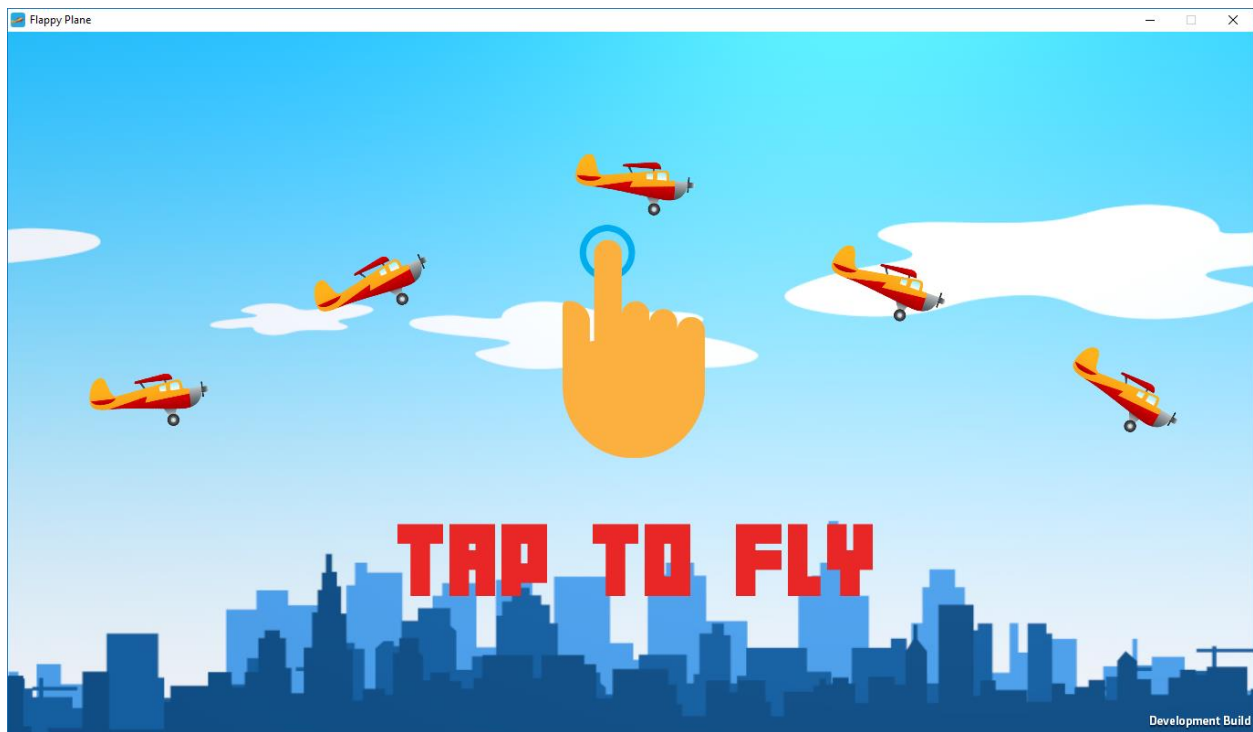
CHƯƠNG 5 – UI



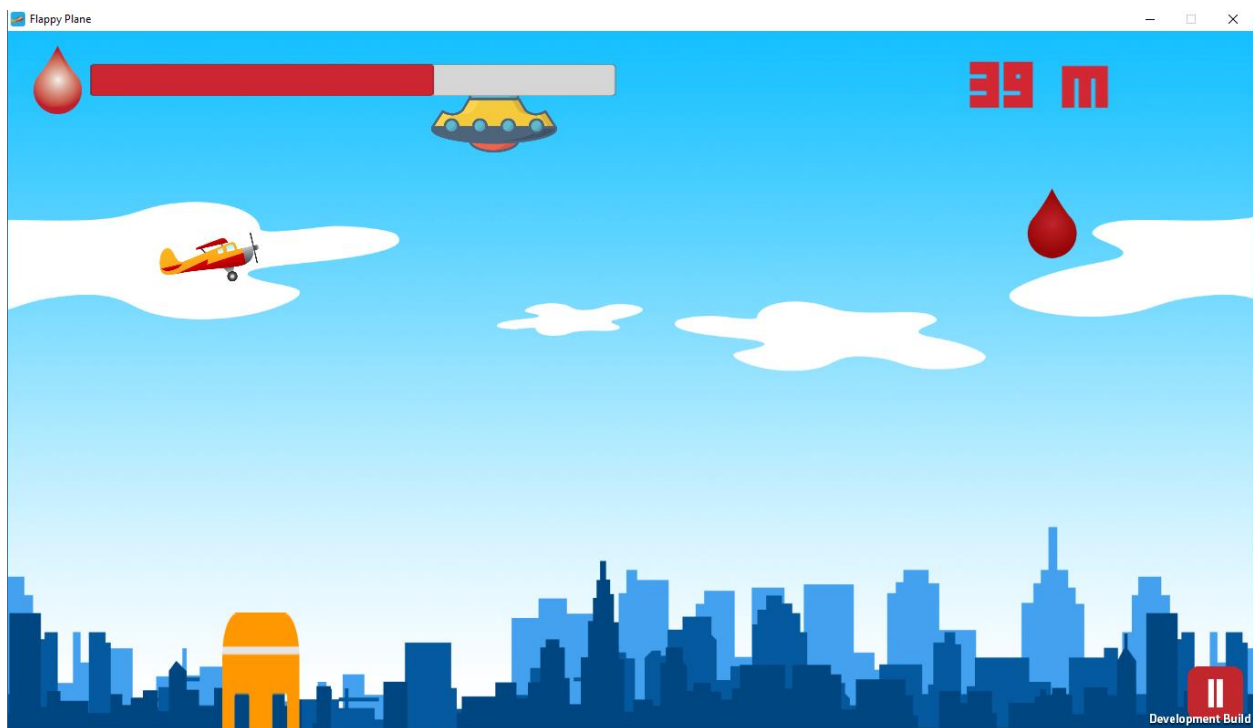
Hình 5.1: Màn hình Menu



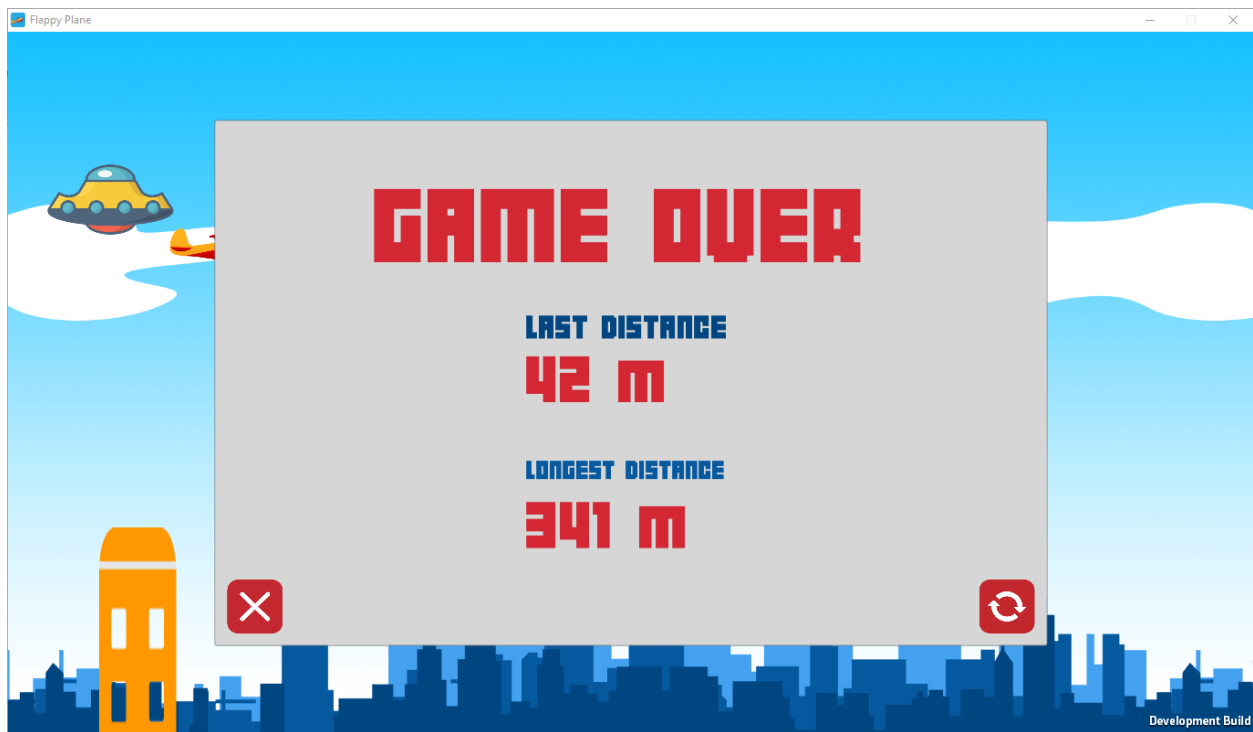
Hình 5.2: Màn hình setting



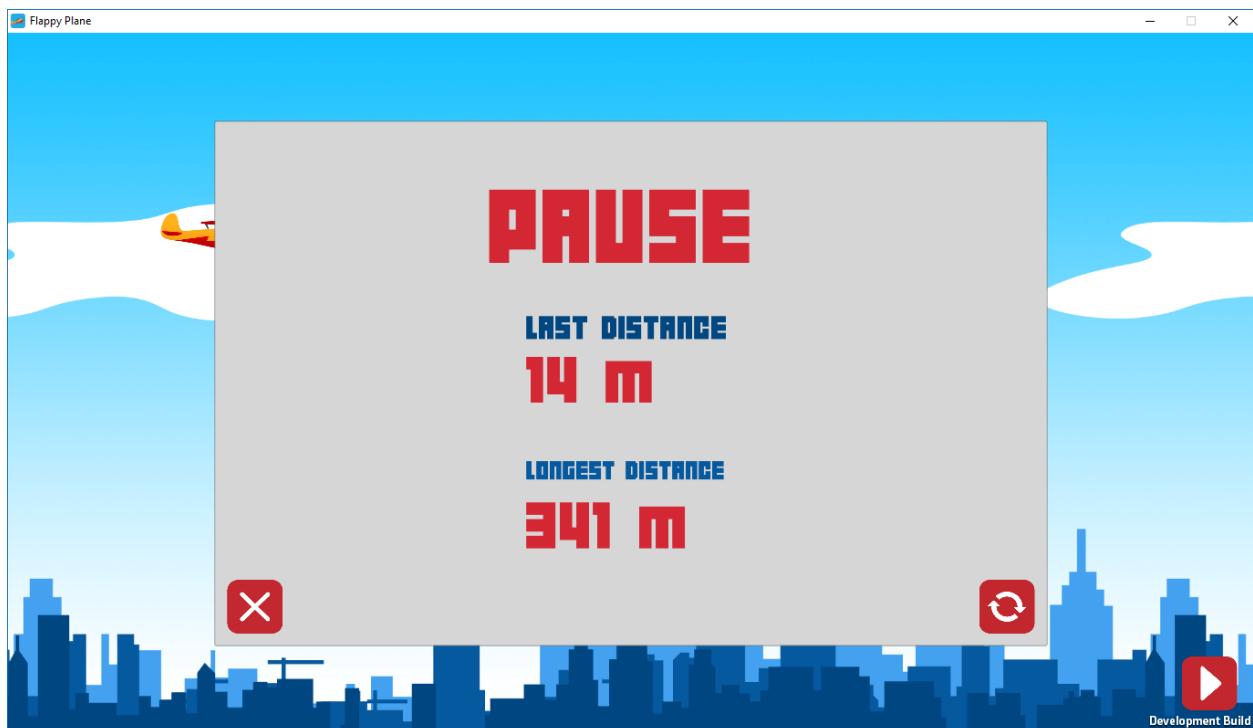
Hình 5.3 Màn hình hướng dẫn



Hình 5.4 Màn hình gameplay



Hình 5.5 Màn hình game over



Hình 5.6 Màn hình tạm dừng

Chương 6: Tổng kết

Chương 6 cũng là chương cuối cùng của tài liệu báo cáo này. Chương 6 sẽ tổng hợp lại các kỹ năng cũng như kinh nghiệm đã có được khi làm nên đồ án phát triển game “Flappy Plane” này.

6.1 Những trở ngại

- Đây là lần đầu tiên nhóm chúng em phát triển game, tương tác với Engine của Unity 3D và lập trình với ngôn ngữ C#. Thông thường chúng em làm web vì nó tương đối đơn giản vì có rất ít những thuật toán liên quan đến chuyển động. Trong khi lập trình trong Unity đòi hỏi tư duy nhiều, khả năng sáng tạo cao.
- Chúng em phát triển game “Flappy Plane” này dựa vào Google, StackOverflow, Udemy, Youtube, Cộng đồng Unity và ý tưởng sẵn có của nhà sáng lập game “Flappy Bird” Nguyễn Hà Đông.
- Đây là công việc đòi hỏi nhiều thời gian và công sức để làm quen với ngôn ngữ C# và Engine Unity.
- Tạo nên Animation Sprites cũng là một việc khó đối với chúng em.

6.2 Thành quả nhận được

- Chúng em biết rất nhiều về Engine Unity 3D cũng như cách nó tạo nên game, cách nó vận hành, các thuộc tính, vật thể, gameobjects và nhiều thứ khác.
- Chúng em biết được cách làm animation sprites cho nhân vật 2D.
- Kết hợp làm việc chung với các thành viên trong nhóm.
- Quản lý thời gian và phân chia công việc tốt hơn.
- Phát triển kỹ năng giao tiếp, phân tích tài liệu có sẵn trên Internet.
- Phát triển khả năng sáng tạo và trí tưởng tượng.
- Chúng em biết được cách chuẩn bị tài liệu SRS và các khâu chuẩn bị để phát triển một phần mềm hoặc trò chơi hoàn chỉnh.

6.3 Kết hoạch trong tương lai

- Chúng em sẽ tiếp tục phát triển game để tương thích với nhiều kích cỡ màn hình khác nhau.
- Phát triển thêm tính năng lưu trữ dữ liệu tốt hơn thông qua kỹ thuật “Loading scene”
- Tạo nên nhiều độ khó khác nhau.
- Có thể thay đổi skins cho máy bay.

PHỤ LỤC

Phụ Lục A: Tài liệu tham khảo

Tài liệu tham khảo chính

1. [Unity: From Master To Pro By Building 6 Games](#)
2. [Khoa Pham - Flappy Bird](#)
3. [Unity 5 Snap to Grid Script](#)
4. [Saving toggle state for music](#)

Đặc biệt cảm ơn

1. Youtube
2. Unity 3D
3. Unity Asset Store
4. Unity Cookie
5. Udemy
6. Lynda