**TRƯỜNG ĐẠI HỌC BÌNH DƯƠNG**

**KHOA TIN HỌC**

**-----**🙢🕮🙠**-----**

****

**ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG**

**ỨNG DỤNG GAME BẮN VỊT TRÊN HỆ ĐIỀU HÀNH ANDROID, SỬ DỤNG THƯ VIỆN ANDENGINE**

**HUỲNH CÔNG HUY**

**ĐÀO NGỌC HẬU**

**Bình Dương, 01/2015**

**TRƯỜNG ĐẠI HỌC BÌNH DƯƠNG**

**KHOA TIN HỌC**

**-----**🙢🕮🙠**-----**

****

**ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG**

**ỨNG DỤNG GAME BẮN VỊT TRÊN HỆ ĐIỀU HÀNH ANDROID, SỬ DỤNG THƯ VIỆN ANDENGINE**

**SVTH: Huỳnh Công Huy – 13th01**

**Đào Ngọc Hậu – 13th01**

**GVHD: ThS.Phạm Thi Vương**

**Bình Dương, 01/2015**

TRƯỜNG ĐẠI HỌC BÌNH DƯƠNG CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc Lập – Tự Do – Hạnh Phúc

🙢🕮🙠 🙢🕮🙠

KHOA: TIN HỌC

BỘ MÔN: **ĐỒ ÁN TỐT NGHIỆP** **NHIỆM VỤ ĐỒ ÁN**

HỌ VÀ TÊN: **HUỲNH CÔNG HUY** MSSV: 10050023

**ĐÀO NGỌC HẬU** 10050010

NGÀNH: **CÔNG NGHỆ THÔNG TIN** LỚP: 13TH01

1. **Đầu đề đồ án:**

Tên đề tài “Xây dựng ứng dụng game bắn vịt trên hệ điều hành Android, sử dụng thư viện AndEngine”

1. **Nhiệm vụ:**

Xây dựng game với các chức năng, xử lý va chạm các đối tượng

1. **Ngày giao nhiệm vụ:** 15/9/2014
2. **Ngày hoàn thành:** 1/2015
3. **Họ và tên người hướng dẫn:** **ThS. Phạm Thi Vương**

Ngày……tháng……năm 20…

NGƯỜI HƯỚNG DẪN

(Ký và ghi rõ họ tên)

*PHẦN DÀNH CHO KHOA, BỘ MÔN*

Người duyệt(chấm sơ bộ):

Đơn vị:

Ngày báo cáo:

Điểm tổng kết:

Nơi lưu trữ luận văn:

Nhận xét của giáo viên hướng dẫn

Bình Dương, ngày…...tháng……năm……

Nhận xét của giáo viên phản biện

Bình Dương, ngày…...tháng……năm……

LỜI CẢM ƠN

🙢🕮🙠

Trong quá trình tìm hiểu và thực hiện đồ án, được sự hướng dẫn tận tình của thầy Phạm Thi Vương, thầy đã truyền thụ những kiến thức và kinh nghiệm quý báu với phương pháp làm việc và học tập đúng cách. Đồng thời, giúp chúng em nắm vững hơn về kiến thức trong việc xây dựng một ứng dụng game trên di động.

Đồ án tốt nghiệp là cơ hôi để chúng em có thể áp dụng những kiến thức đã được học, đồng thời nó cũng giúp chúng em có những kinh nghiệm quý báu cho công việc sau này được tốt hơn và không còn bỡ ngỡ.

Em xin chân thành cám ơn thầy đã giúp đỡ chúng em thực hiện đồ án. Kính chúc thầy khỏe mạnh và tiếp tục đạt nhiều thành công trong sự nghiệp nghiên cứu khoa học và sự nghiệp trồng người.

Xin cảm ơn tới những người thân, bạn bè đã quan tâm, giúp đỡ, động viên trong suốt quá trình học tập và làm đồ án.

Em xin chân thành cảm ơn!

Bình Dương, ngày … tháng … năm 20…

Sinh viên:

Huỳnh Công Huy

Đào Ngọc Hậu

LỜI NÓI ĐẦU

Với sự phát triển không ngừng của ngành công nghệ thông tin trên toàn thế giới hiện nay. Cùng với sự phát triển của ngành kinh tế, ngành công nghệ thông tin cũng đang dần chiếm lĩnh thị trường với nhiều sản phẩm của phần cứng và phần mềm ra đời phục vụ đời sống con người. Việc ứng dụng khoa học công nghệ thông tin rất mạnh mẽ vào các ngành như: giáo dục, kinh tế, xây dựng, hàng không, quân sự…góp phần thúc đẩy phát triển xã hội.

Đất nước ta đang trong thời kỳ công nghiệp hóa, hiện đại hóa, đời sống người dân   
không ngừng cải thiện, kéo theo đó là nhu cầu về vật chất và tinh thần được nâng cao. Trong  số đó phải kể đến nhu cầu về các thiết bị di động. Ngày nay, việc sử dụng thiết bị di động  (điện thoại di động, máy tính bảng ) đã trở nên rất phổ biến đối với mọi lứa tuổi, mọi tầng lớp trong xã hội. Bởi vậy, có thể dễ dàng nhận ra thị trường di động là một thị trường rất tiềm  năng mà nhiều cá nhân, tập thể, doanh nghiệp muốn nhắm tới. Cùng với sự ra đời những sản  phẩm thiết bị di động công nghệ cao của Apple, Nokia, Samsung, LG, HTC và sự phát triển  của các hệ điều hành Android, iOS, Window Phone các thiết bị di động ngày càng thu hút  một số lượng lớn người dùng trên toàn thế giới. Và một yếu tố quan trọng trong sự thu hút đó  phải kể đến game mobile – trò chơi trên thiết bị di động. Game từ lâu đã là một dịch vụ giải trí rất phổ biến đối với người dùng Việt Nam cũng như trên thế giới. Dịch  vụ game có thể được phát triển trên nhiều nền tảng như PC, mobile, browser Cùng với game  PC và game browser, ngày nay, game dành cho các thiết bị di động cũng đang thực sự bùng  nổ.

Ở Việt Nam, hệ điều hành Android của Google sử dụng trên các thiết bị di động thông  minh đang là một trong những hệ điều hành được ưa chuộng nhất. Có nhiều cách thức phát  triển dịch vụ game trên hệ điều hành Android. Trong đó, AndEngine nổi bật lên như một công  cụ đơn giản và hiệu quả bậc nhất để hỗ trợ việc phát triển game với đồ họa đẹp mắt.

MỤC LỤC

[Nhận xét của giáo viên hướng dẫn i](#_Toc408092059)

[Nhận xét của giáo viên phản biện ii](#_Toc408092060)

[LỜI CẢM ƠN iii](#_Toc408092061)

[LỜI NÓI ĐẦU iv](#_Toc408092062)

[MỤC LỤC](#_Toc408092063)

[CHƯƠNG 1](#_Toc408092064) [TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc408092065)

[1.1 TÊN ĐỀ TÀI 1](#_Toc408092066)

[1.2 LÝ DO CHỌN ĐỀ TÀI 1](#_Toc408092067)

[1.3 MỤC TIÊU ĐỀ TÀI 2](#_Toc408092068)

[1.4 PHẠM VI ĐỀ TÀI 2](#_Toc408092069)

[1.5 Ý NGHĨA THỰC TIỄN CỦA ĐỀ TÀI 2](#_Toc408092070)

[1.6 PHƯƠNG PHÁP THỰC HIỆN 2](#_Toc408092071)

[1.7 KẾ HOẠCH THỰC HIỆN ĐỀ TÀI 3](#_Toc408092072)

[CHƯƠNG 2](#_Toc408092073) [GIỚI THIỆU CHUNG VỀ GAME VÀ GAME TRÊN THIẾT BỊ DI ĐỘNG 4](#_Toc408092074)

[2.1 KHÁI QUÁT VỀ GAME 4](#_Toc408092075)

[2.1.1 Các thể loại game cơ bản 4](#_Toc408092076)

[2.1.2 Các thành phần cơ bản của một game 6](#_Toc408092077)

[2.2 GAME TRÊN THIẾT BỊ DI ĐỘNG 7](#_Toc408092078)

[2.3 SƠ LƯỢC QUY TRÌNH PHÁT TRIỂN GAME DI ĐỘNG 7](#_Toc408092079)

[2.4 MỘT SỐ FRAMEWORK HỖ TRỢ VIỆC PHÁT TRIỂN GAME TRÊN ANDROID 9](#_Toc408092080)

[2.5 CÔNG CỤ HỖ TRỢ 9](#_Toc408092081)

[CHƯƠNG 3](#_Toc408092082) [TỔNG QUAN VỀ ANDENGINE 10](#_Toc408092083)

[3.1 GIỚI THIỆU CHUNG 10](#_Toc408092084)

[3.2 CÁC TÍNH NĂNG QUAN TRỌNG CỦA ANDENGINE 10](#_Toc408092085)

[3.3 LỢI ÍCH TỪ VIỆC SỬ DỤNG ANDENGINE 10](#_Toc408092086)

[3.4 NHƯỢC ĐIỂM 11](#_Toc408092087)

[3.5 ĐỐI TƯỢNG SỬ DỤNG 11](#_Toc408092088)

[3.6 MỘT SỐ GAME ĐƯỢC LÀM TỪ ANDENGINE 11](#_Toc408092089)

[3.7 CÁC KHÁI NIỆM CHÍNH TRONG LẬP TRÌNH ANDENGINE 14](#_Toc408092090)

[3.8 VÒNG ĐỜI CỦA MỘT GAME ANDENGINE 16](#_Toc408092091)

[3.9 NỀN TẢNG HỖ TRỢ 17](#_Toc408092092)

[3.10 GIỚI THIỆU CÔNG CỤ TILED MAP 17](#_Toc408092093)

[3.11 MỘT SỐ THÀNH PHẦN ANDENGINE ĐƯỢC DÙNG TRONG GAME 22](#_Toc408092094)

[3.11.1 Cấu trúc của lớp BaseGameActivity 22](#_Toc408092095)

[3.11.2 Camera / EngineOptions 22](#_Toc408092096)

[3.11.3 BitmapTextureAtlas / Texture 23](#_Toc408092097)

[3.11.4 Sprite / AnimatedSprite 24](#_Toc408092098)

[3.11.5 Font 25](#_Toc408092099)

[3.11.6 Text 26](#_Toc408092100)

[CHƯƠNG 4](#_Toc408092101) [PHÂN TÍCH VÀ THIẾT KẾ 27](#_Toc408092102)

[4.1 PHÂN TÍCH HỆ THỐNG 27](#_Toc408092103)

[4.1.1 Giới thiệu chung 27](#_Toc408092104)

[4.1.2 Thể loại 27](#_Toc408092105)

[4.1.3 Mô tả kịch bản 27](#_Toc408092106)

[4.1.4 Phân tích kịch bản 28](#_Toc408092107)

[4.1.5 Mục đích 28](#_Toc408092108)

[4.1.6 Các lớp giao diện 28](#_Toc408092109)

[4.1.7 Thiết bị 29](#_Toc408092110)

[4.1.8 Platform 29](#_Toc408092111)

[4.1.9 Level 29](#_Toc408092112)

[3.1.10 Kỹ thuật 29](#_Toc408092113)

[**4.2 BIỂU ĐỒ USE – CASE** 30](#_Toc408092114)

[4.2.1 Danh sách actor 30](#_Toc408092115)

[4.2.2 Danh sách use case 30](#_Toc408092116)

[4.2.3 Biểu đồ use-case 31](#_Toc408092117)

[4.2.4 Đặc tả use – case 31](#_Toc408092118)

[4.2.5 Đặc tả use case “chọn màn chơi” 32](#_Toc408092119)

[4.2.6 Đặc tả use case “trở lại” 33](#_Toc408092120)

[4.2.7 Đặc tả use case “chơi game” 34](#_Toc408092121)

[4.2.8 Đặc tả use case “thoát game” 35](#_Toc408092122)

[4.2.9 Đặc tả use case “dừng game” 36](#_Toc408092123)

[4.2.10 Đặc tả use case “tiếp tục” 37](#_Toc408092124)

[4.2.11 Đặc tả use case “tắt âm thanh” 38](#_Toc408092125)

[4.2.12 Đặc tả use case “mở âm thanh” 39](#_Toc408092126)

[4.3 MỘT SỐ XỬ LÝ TRONG GAME 40](#_Toc408092127)

[4.3.1 Xử lý điểm thưởng ngẫu nhiên. 40](#_Toc408092128)

[4.3.2 Xử lý lưu điểm tốt nhất 41](#_Toc408092129)

[4.32 Xử lý load điểm lên khi khởi động game 41](#_Toc408092130)

[4.4 THIẾT KẾ ĐỒ HỌA 42](#_Toc408092131)

[CHƯƠNG 5](#_Toc408092132) [KẾT QUẢ THỰC HIỆN VÀ HƯỚNG PHÁT TRIỂN 47](#_Toc408092133)

[5.1 KẾT QUẢ ĐẠT ĐƯỢC 47](#_Toc408092134)

[5.2 CÁC HẠN CHẾ CỦA ĐỀ TÀI 47](#_Toc408092135)

[5.3 HƯỚNG PHÁT TRIỂN 47](#_Toc408092136)

[TÀI LIỆU THAM KHẢO 48](#_Toc408092137)

CHƯƠNG 1

TỔNG QUAN VỀ ĐỀ TÀI

* 1. TÊN ĐỀ TÀI

Xây dựng ứng dụng game bắn vịt trên hệ điều hành Android, sử dụng thư viện AndEngine

* 1. LÝ DO CHỌN ĐỀ TÀI
* Đất nước ta đang trong thời kỳ công nghiệp hóa, hiện đại hóa, đời sống người dân   
  không ngừng cải thiện, kéo theo đó là nhu cầu về vật chất và tinh thần được nâng cao. Trong số đó phải kể đến nhu cầu về các thiết bị di động. Ngày nay, việc sử dụng thiết bị di động (điện thoại di động, máy tính bảng ) đã trở nên rất phổ biến đối với mọi lứa tuổi, mọi tầng lớp trong xã hội. Bởi vậy, có thể dễ dàng nhận ra thị trường di động là một thị trường rất tiềm năng mà nhiều cá nhân, tập thể, doanh nghiệp muốn nhắm tới. Cùng với sự ra đời những sản phẩm thiết bị di động công nghệ cao của Apple, Nokia, Samsung, LG, HTC và sự phát triển của các hệ điều hành Android, iOS, Window Phone các thiết bị di động ngày càng thu hút một số lượng lớn người dùng trên toàn thế giới. Và một yếu tố quan trọng trong sự thu hút đó phải kể đến game mobile – trò chơi trên thiết bị di động.  
  Game từ lâu đã là một dịch vụ giải trí rất phổ biến đối với người dùng Việt Nam. Dịch vụ game có thể được phát triển trên nhiều nền tảng như PC, mobile, browser Cùng với game PC và game browser, ngày nay, game dành cho các thiết bị di động cũng đang thực sự bùng nổ.
* Trong sự phát triển không ngừng của các công cụ phát triển game trên thế giới, các engine được phát triển luôn có những ưu điểm và nhược điểm của riêng nó.
* Khi dùng các Engine để lập trình game ta sẽ tạo ra sản phẩm nhanh hơn, không tốn quá nhiều thời gian cho các game tầm trung bình (có thể là các game lớn).
* Việc học 1 Engine thường không mất nhiều thời gian (Các Engine được tạo ra với mục đích là đơn giản, dễ hiểu, hiệu quả).
* Các Engine đều có ví dụ minh họa, từ những ví dụ này sẽ có cái nhìn tổng quan những gì mà Engine đó có thể làm được.
* Các Engine đều có điểm mạnh điểm yếu riêng của nó, khi ta có trong tay vài Engine thì việc chọn Engine phù hợp với game của mình làm sẽ giúp cho việc hoàn thành sản phẩm nhanh hơn và tốt hơn.
* Cộng đồng xây dựng rất lớn, gặp bất cứ khó khăn nào ta đều có thể hỏi ngay trên diễn đàn hoặc website.
  1. MỤC TIÊU ĐỀ TÀI
* Tìm hiểu các vấn đề :
  + Xây dựng một game AndEngine.
  + Xây dựng các bản đồ trong game bằng công cụ Tiled Map.
  + Cách load hình ảnh, nhạc nền, âm thanh va chạm giữa các đối tượng, nhân vật, ghi điểm, xử lý thua, thắng khi chơi, qua bản đồ mới,….
* Tạo ra một trò chơi với giao diện đồ họa linh động, đẹp mắt nhằm mục đích giải trí cho tất cả mọi người trong lúc thời gian rảnh rỗi.
* Game có dung lượng nhẹ, không cần thiết bị có cấu hình cao.
  1. PHẠM VI ĐỀ TÀI

Trong phạm vi đồ án của mình, em xin trình bày một số kiến thức cơ bản về AndEngine,  và cách thức xây dựng “Game bắn vịt” sử dụng AndEngine trên hệ điều hành Android. Game chủ yếu tập trung chú ý vào việc sử dụng các đối tượng mà AndEngine hỗ trợ, khả năng thể  hiện đồ họa trong AndEngine Game được phát triển theo 2 level. Level 1 sử dụng những  Sprite động và tĩnh, cùng với bộ đếm thời gian và cách tính điểm đơn giản. Level 2 áp dụng  thêm thành phần mở rộng về vật lý của AndEngine để làm game hấp dẫn hơn.

* 1. Ý NGHĨA THỰC TIỄN CỦA ĐỀ TÀI
* Ý nghĩa đối với người sử dụng :
  + Nhằm đem lại mục đích giải trí cho người sử dụng.
* Ý nghĩa đối với cá nhân
  + Tạo ra được 1 trò chơi, nâng cao trình độ hiểu biết và nhận thức trong lĩnh vực phân tích, thiết kế, hiểu sâu hơn lĩnh vực mình đã làm. Bên cạnh đó còn giúp cá nhân hoàn thành được đồ án đồng thời nâng cao trình độ, kỹ năng, biết sử dụng các tools, ngôn ngữ lập trình một cách hiệu quả.
  1. PHƯƠNG PHÁP THỰC HIỆN

Đề tài được thực hiện theo phương pháp hướng đối tượng, được tiến hành theo các bước sau:

* + 1. **Tiến hành khảo sát hiện trạng**
* Thu thập thông tin, hình thức, cách chơi từ các game đã có
* Tìm hiểu và xác định yêu cầu nghiệp vụ trong game
* Xây dựng các kịch bản có sẵn để thiết kế game
  + 1. **Tiến hành phân tích hệ thống**
* Thông qua việc xác định các yêu cầu nghiệp vụ, ta xác định các tác nhân(actor) và các trường sử dụng (use case)
* Phân tích các use case
* Tiến hành đặc tả các use case
* Xây dựng biểu đồ use case
  + 1. **Tiến hành xây dựng phần mềm**
* Thiết kế đồ họa, giao diện, cảnh trong game
* Thiết kế xử lý va chạm các đối tượng trong game
* Xây dựng các chức năng trong game như: tắt/mở âm thanh, pause/resume,..
  1. KẾ HOẠCH THỰC HIỆN ĐỀ TÀI

|  |  |  |  |
| --- | --- | --- | --- |
| Tuần | Từ ngày | Đến Ngày | Nội Dung Thực Hiện |
| 1 | 15/09/2014 | 22/09/2014 | Chương 1: Tổng quan. |
| 2 | 23/09/2014 | 30/09/2014 |
| 3 | 1/10/2014 | 8/10/2014 | Chương 2: Giới thiệu chung về game và game trên thiết bị di động |
| 4 | 9/10/2014 | 16/10/2014 |
| 5 | 17/10/2014 | 24/10/2014 | Chương 3: Phân tích và thiết kế hệ thống |
| 6 | 25/10/2014 | 2/11/2014 |
| 7 | 3/11/2014 | 10/11/2014 | Chương 4: Cài đặt game |
| 8 | 11/11/2014 | 18/11/2014 |
| 9 | 19/11/2014 | 26/11/2014 |
| 10 | 3/12/2014 | 10/12/2014 | Chương 5: Thử nghiệm – thực nghiệm. |
| 11 | 17/12/2014 | 24/12/2014 |
| 12 | 25/12/2014 | 1/1/2015 | Chương 6: Kết luận |

CHƯƠNG 2

GIỚI THIỆU CHUNG VỀ GAME VÀ GAME TRÊN THIẾT BỊ DI ĐỘNG

2.1 KHÁI QUÁT VỀ GAME

Kể từ khi máy tính xuất hiện, game đã trở thành một trong những ứng dụng phổ biến nhất trên thị trường. Trong cuốn sách Reality Is Broken, tác giả Jane McGonigal đã nêu ra 4 thuộc tính của phần lớn game trên thị trường:

+ Mục tiêu: Game chỉ ra một mục tiêu rõ ràng cho người chơi để họ đạt được. Mục tiêu là những thử thách đối với người chơi, nhưng họ có thể vượt qua. Người chơi sẽ cảm thấy hứng thú nhất nếu mục tiêu đó phù hợp với khả năng của mình, không quá khó, không quá dễ. Mục tiêu rõ ràng, thú vị cũng là yếu tố quan trọng thu hút người chơi đến với game.

+ Quy tắc: Game cũng có những quy tắc bắt buộc mà tất cả người chơi phải tuân theo. Các quy tắc này thường làm cho việc đạt được mục tiêu nêu trên khó khăn hơn. Điều đó cũng khuyến khích sự sáng tạo của mỗi người.

+ Phản hồi: Game cần phản hồi cho người chơi biết rằng họ đang làm công việc của mình tốt hay không. Một điều đáng lưu ý rằng, hệ thống phản hồi chính là chìa khóa để làm trò chơi thú vị.

+ Tham gia một cách tự nguyện: một trò chơi sẽ không còn là một trò chơi đúng nghĩa nếu bạn không thực sự thích chơi nó. Điều này ngụ ý rằng người chơi phải chấp nhận được mục tiêu, các quy tắc, và hệ thống phản hồi của trò chơi.

**2.1.1 Các thể loại game cơ bản**

Những người phát triển game không chủ động phân loại các trò chơi của họ. Và tất nhiên, không có danh sách chuẩn nào về việc phân loại game. Mặc dù vậy, theo thời gian, các trò chơi cũng dần được phân loại vào các lớp khác nhau theo những cách khác nhau. Việc phân loại chính xác rất khó khăn. Tuy vậy, chúng ta có thể kể đến một số loại game điển hình như sau:

* ***Trò chơi hành động (Action Game):*** Trò chơi đề cao khả năng chiến đấu của nhân vật trong từng khu vực rộng lớn; ít đề cập đến những câu đố, thông điệp phức tạp, hay những trận đánh tay đô. Người chơi “nhìn” bằng mắt của chính nhân vật. Dạng này yêu cầu bạn phải có khả năng phản xạ nhanh, phối hợp tay - mắt tốt, am tường và hiểu rõ các loại vũ khí có trong game;
* ***Trò chơi nhập vai ( Role – Playing Game):*** người chơi sẻ vào vai một nhân vật hoặc một nhóm để tham gia các sự kiện, giao tiếp với các nhân vật phụ, bước vào những cuộc chiến (theo lượt hoặc thời gian thực) với một quá trình xây dựng nhân vật thông qua việc tăng cấp độ, phát triển các kỹ năng, tính cách...
* ***Trò chơi phiêu lưu (Adventure Game):*** Nét cơ bản của của thể loại này là khám phá các vùng đất, tìm các vật dụng và giải đố; có rất ít hoặc gần như là không có các cảnh mang tính “hành động”. Thể loại trò chơi này không chú trọng vào sự phát triển nhân vật, cũng như tăng cấp độ. Trò chơi phiêu lưu có cốt truyện rất chặt chẽ và rất dài, đòi hỏi người chơi phải vận dụng trí tuệ, lòng nhẫn nại và sự quyết đoán để chọn hướng đi tốt nhất.
* ***Trò chơi chiến thuật (Strategy Game):*** Thể loại này chú trọng cách quản lý nguồn tài nguyên và quân đội trước những đối thủ do máy hoặc người chơi khác điều khiển, hầu hết đều “dính dáng” chiến tranh. Nhìn chung, thể loại này được chia làm 2 dạng phân biệt khác nhau. Thứ nhất là chiến thuật thời gian thực (RTS), thứ 2 là chiến thuật theo lượt(TBS). RTS có đặc trưng là hành động (diễn biến) xảy ra liên tục, không ngừng do danh sách các lệnh được đưa ra xuyên suốt trong trò chơi, đòi hỏi người chơi phải có cách “điều binh khiển tướng” hợp lý. TBS đòi hỏi người chơi phải biết cách tính toán “nặng nhẹ” cho mỗi nước đi của mình
* ***Trò chơi kinh dị:*** Gần giống Trò chơi hành động – Trò chơi phiêu lưu nhưng luôn tạo ra cảm giác sợ hãi và hồi hộp cho người chơi. Yếu tố kinh dị trong game thường được tạo ra từ ma quỷ (demon), linh hồn (ghost), thây ma (zombie).
* ***Trò chơi mô phỏng (Simulation Game):*** Yêu cầu duy nhất của thể loại này là phải mô phỏng làm sao cho càng giống thế giới thực càng tốt
* ***Trò chơi thể thao (Sport Game):*** Được xem là một nhánh rẽ của trò chơi mô phỏng, trò chơi thể thao mô tả một môn thể thao của thế giới thực diễn ra trong môi trường 3D. Game thường được tổ chức theo cơ cấu mùa giải, và cho phép người chơi đảm nhận các vị trí huấn luyện viên, ông bầu... bên cạnh các tính năng trao đổi, chuyển nhượng cầu thủ, quản lý ngân sách
* ***Trò chơi giải đố (Puzzle Game):*** Đây là lĩnh vực dành cho những người yêu thích các câu đố cực... khó, hơn hẳn trò chơi phiêu lưu. Game đòi hỏi người chơi phải quyết định nhanh chóng (và chính xác) các vấn đề đã được đặt ra theo hướng tư duy, suy luận logic. Đôi khi các vấn đề này lại được lồng ghép, xâu chuỗi để tăng độ khó.
* ***Trò chơi âm nhạc (Rhythm Game):*** Trò chơi tập trung vào âm nhạc và thời gian, điều khiển trong trò chơi chủ yếu bằng cách nhấn nút trên một bộ điều chỉnh thông thường
* ***Trò chơi đối kháng (Fighting Game):*** Đặc trưng của thể loại này là những trận so tài tay đôi giữa các nhân vật.
* ***Trò chơi giải trí (Entertaiment Game):*** Là dạng game ngắn, đơn giản, được thiết kế cho nhiều người chơi và... dễ chơi
* ***Trò chơi nhập vai trực tuyến ( Massively -multiplayer online Game):*** Đặc trưng của thể loại này là có sự tham gia của một số lượng lớn người vào một thế giới liên tục, thông qua giao tiếp cộng đồng trực tuyến với những người chơi khác

**2.1.2 Các thành phần cơ bản của một game**

* **Opening (Splash) Screen**: Để tối ưu hóa hiệu năng hoạt động, trước khi game sẵn sàng, đồ họa của level thường được nạp vào trước khi level đó bắt đầu. Quá trình nạp dữ liệu có thể kéo dài đến vài giây. Trong thời gian đó, bạn không muốn để người dùng phải nhìn vào một màn hình trống. Vậy nên, hãy hiển thị một màn hình chờ (splash screen). Nó cho người dùng biết game vẫn đang hoạt động bình thường.
* **Menu Screen:**Khi game đã sẵn sàng, chúng ta cần cung cấp cho người dùng nơi để họ có thể đưa ra những lựa chọn (Ví dụ như bật/tắt âm thanh, xem hướng dẫn chơi ). Điều này được thực hiện ở Menu Screen.
* **Âm nhạc (Music):** Đối với bất kì ai, âm nhạc có ảnh hưởng mạnh đến cảm xúc. Bởi vậy, nhạc nền đóng vai trò quan trọng trong việc giữ nhịp cho game, và cũng giúp cho việc chuyển đổi giữa các phần khác nhau trong game được liên tục, không bị ngắt quãng.
* **Hiệu ứng âm thanh (Sound Effects):** Những hiệu ứng âm thanh sẽ làm cho game trở nên thú vị hơn nhiều. Ví dụ như khi hai vật thể va chạm, người chơi muốn được nghe một tiếng động nào đó thể hiện sự va chạm này. Âm thanh sẽ làm cho game thân thiện hơn với người chơi.
* **Thời gian:** Hầu hết các trò chơi đều kết hợp thời gian. Yếu tố thời gian thôi thúc người chơi hoàn thiện mình qua từng màn chơi để có thể đáp ứng được mục tiêu của màn chơi.
* **Mạng (Lives):** Game sẽ trở nên hấp dẫn hơn nếu có thêm những thử thách. Bởi vậy, hãy thêm vào trong game giới hạn về số lượt chơi. Như vậy, người chơi sẽ cảm thấy được thách thức, game sẽ vui hơn nhiều.
* **Chướng ngại vật (Obstacles):** Mỗi thể loại game khác nhau sẽ sử dụng những chướng ngại vật khác nhau. Hãy lựa chọn chướng ngại vật phù hợp để tăng độ khó cũng như thử thách người chơi.
* **Cấp độ (Levels):** Thử thách trong game là một phần thúc đẩy người chơi. Nhưng hãy biết cách giới hạn nó! Hãy chia những thử thách thành nhiều cấp độ. Người chơi bắt đầu với những màn chơi dễ dàng, rồi dần dần làm quen, nâng cao kỹ năng, để chiến thắng được những cấp độ khó hơn.
* **Người chơi (Player):** Người chơi luôn là phần quan trọng nhất của bất kỳ game nào. Game sẽ chỉ thành công khi luôn giữ được sự thú vị, lôi cuốn đối với người chơi. Người chơi cần được thử thách bởi các thành phần trong game, nhưng không được quá thách thức khiến họ từ bỏ game trong thất vọng. Game cần bao gồm đủ nhiều yếu tố để duy trì sự quan tâm từ phía người chơi!
* **Màn chơi (Scenes):** Nếu coi trò chơi như một bộ phim, thì mỗi màn hình hiển thị cho người chơi cũng giống như một cảnh trong phim. Ở đó có sự thay đổi về nền đồ họa trong mỗi cảnh, nhưng không quá nhiều dù điểm nhìn của người chơi thay đổi. Sau đó, ta thêm vào những thứ chuyển động, cùng với những chướng ngại vật để tạo ra một màn hình game hoàn chỉnh.

2.2 GAME TRÊN THIẾT BỊ DI ĐỘNG

Cùng với sự đa dạng và phong phú về thể loại game có thể lựa chọn, chúng ta cần tập trung vào những yếu tố phù hợp nhất với nền tảng di động (điện thoại di động, máy tính bảng ), đặc biệt quan tâm đến những game có thể phát triển bằng một nhóm ít người.

Với tiềm năng to lớn của thị trường di động, không ngạc nhiên khi có rất nhiều nghiên cứu và tư tưởng được đưa ra nhằm mục đích tạo nên một game di động tốt. Cùng với việc tiếp tục áp dụng những nguyên tắc thông thường khi thiết kế game trên máy tính, ta cần quan tâm đến một số đặc điểm của game được thiết kế trên các thiết bị di động:

* Không lãng phí thời gian của người chơi.
* Cung cấp sự trợ giúp cần thiết cho người chơi.
* Làm cho mục tiêu của trò chơi dễ hiểu.
* Hiển thị các trạng thái trong game một cách rõ ràng.
* Người dùng di động thường chỉ chơi game trong một thời gian ngắn.
* Người chơi có thể dễ dàng tạm dừng hay tiếp tục game khi cần thiết.
* Người chơi có thể đạt được tiến bộ trong một thời gian ngắn.
* Những hạn chế của thiết bị di động ảnh hưởng đến việc xây dựng game:
* Kích thước màn hình nhỏ, và đa dạng về kích thước màn hình, độ phân giải
* Có rất nhiều cách thức nhập dữ liệu từ người dùng (bàn phím, cảm ứng, cảm ứng đa điểm, )
* Hạn chế về khả năng tính toán.
* Hạn chế về pin.

2.3 SƠ LƯỢC QUY TRÌNH PHÁT TRIỂN GAME DI ĐỘNG

Một đội phát triển game trên thiết bị di động thường có những vị trí sau:

+ Người viết kịch bản.

+ Lập trình viên.

+ Người thiết kế đồ họa.

+ Người soạn nhạc và hiệu ứng âm thanh.

+ Người kiểm tra và đóng gói sản phẩm.

Cần có một tài liệu thật tốt để mọi người trong nhóm có chung tiếng nói, hiểu được mục tiêu chung và cùng nhau phát triển tốt sản phẩm game.

Xét về mặt tổng quát, có thể chia phát triển game thành 4 giai đoạn:

**Giai đoạn 1:** Giai đoạn tiền sản phẩm/ý tưởng

Ở giai đoạn này, từng thành viên phát triển ý tưởng, sau đó thống nhất với cả đội để xác định ý tưởng chủ đạo của sản phẩm. Qua giai đoạn này, các ý tưởng cho lập trình, nội dung, thể loại game (action, puzzle, adventure, platform, sport, RPG ), phong cách đồ họa và âm nhạc dần được hình thành thể hiện qua biểu đồ, đặc tả, các thông số thử nghiệm, hình vẽ tay về nhân vật (sketch), giai điệu nhạc được lựa chọn và thống nhất xuyên suốt các giai đoạn phát triển.

**Giai đoạn 2:** Đặc tả cho lập trình

Đây là một giai đoạn rất quan trọng trong quá trình thiết kế game. Bạn càng bỏ nhiều thời gian cho giai đoạn này thì khi lập trình, gỡ lỗi càng tiết kiệm thời gian. Lập trình viên chuyên nghiệp hiểu rằng lỗi xuất hiện trong quá trình thiết kế sẽ thiệt hại hơn nhiều so với lỗi được phát hiện trong giai đoạn này.

Bạn cần giấy, bút hay chương trình văn bản và cùng với một vài người khác chơi thử game. Khi “chơi thử” bằng cách ghi ra giấy, các ưu/khuyết của thiết kế sẽ lộ rõ và bạn có thể chỉnh sửa cho đến khi cảm thấy vừa ý.

**Giai đoạn 3:** Thực hiện viết code, thiết kế đồ họa và chọn nhạc, hiệu ứng âm thanh. Với đặc tả đã có, hãy dự tính thời lượng cho việc lập trình, tính toán ngày công kỹ càng để có thể ước tính khối lượng công việc và dễ theo dõi tiến độ. Và phải đảm bảo việc code đúng với những gì đã đặc tả trong game.

Bên cạnh việc viết code, thiết kế đồ họa trong game cũng cần đặc biệt lưu tâm. Do độ phân giải của màn hình điện thoại di động khá nhỏ nên mọi vật thể đồ họa trong game đều được thực hiện bằng đồ họa điểm ảnh (pixel art), vì vậy người thiết kế đồ họa ngoài kiến thức về vẽ tay, đổ bóng còn phải có kỹ năng thao tác tỉ mỉ.

Âm thanh trong game là một phần rất quan trọng tạo nên cái hồn của game. Việc lựa chọn hay mix nhạc phù hợp sẽ tạo sự hào hứng với người chơi, tăng độ hấp dẫn cho game của bạn. Hiện tại cũng có nhiều chương trình giúp cho việc soạn nhạc dễ dàng mà không cần biết nhiều về nhạc lý, bạn chỉ cần lựa vài hợp âm, nhạc cụ, tempo, điệu nhạc , và chương trình sẽ tạo bản nhạc cho bạn.

**Giai đoạn 4:** Kiểm tra và đóng gói sản phẩm

Kiểm tra: Nhóm lập trình sẽ cùng chơi thử, cảm nhận và ghi lại tất cả các ưu và khuyết. Sửa các lỗi nếu có. Phát hành bản beta cho game thủ chơi thử và lấy ý kiến của họ.

Đóng gói: game cho di động phải được “xuất” hay đóng gói cho các dòng máy khác nhau (do các dòng máy này khác nhau về phím nhấn, bộ nhớ và thư viện hỗ trợ).

2.4 MỘT SỐ FRAMEWORK HỖ TRỢ VIỆC PHÁT TRIỂN GAME TRÊN ANDROID

Do nhu cầu không ngừng của người chơi game cũng như của những nhà phát triển, ngày nay trên thị trường có rất nhiều framework hỗ trợ việc xây dựng và phát triển game trên các thiết bị di động, đặc biệt là trên hệ điều hành Android.

Tùy theo yêu cầu và đặc điểm của từng game, mà lập trình viên lựa chọn cho mình framework thích hợp để lập trình. Một số framework cơ bản có thể kể đến như:

* Libgdx: Libgdx framework được phát triển dựa trên nghiên cứu về game development cross-platform. Nghĩa là game của bạn khi viết ra bằng Libgdx framework, nó sẽ chơi được trên Desktop, Web, Android, iOS và trong tương lai có thể thêm những platform khác nữa!

Khi bạn code game sử dụng Libgdx framework, bạn có thể code trên desktop và debug ở đó, sau khi game hoàn thiện, bạn hoàn toàn có thể mang game đó chạy trên các platform khác mà không cần sửa chữa mã nguồn nhiều. Libgdx framework tối ưu hóa game của bạn bởi các thành phần thiết kế như quản lý tài nguyên được viết bằng Native code. Libgdx framework có phần tài liệu hướng dẫn rất tuyệt vời và được hỗ trợ qua forum một cách nhanh chóng và nhiệt tình. Libgdx framework có nhiều game example, nhiều source code, và có rất nhiều game thương mại và phi thương mại đã và đang sử dụng Libgdx framework.Ligdx framework thích hợp để xây dựng những game chơi được trên nhiều platform, và những game hành động đòi hỏi di chuyển mượt mà.

* AndEngine: AndEngine là một bộ các công cụ để phát triển video game, được phát triển bởi Nicolas Gramlich. Nó bao gồm đầy đủ các thuộc tính quan trọng nhất của OpenGL, ngay cả khi bạn không có kinh nghiệm với OpenGL - bạn vẫn có thể tạo ra 1 game có đồ họa đẹp và đầy đủ tính năng hoặc 1 ứng dụng đẹp mắt sử dụng công cụ này.   
  Tất cả các phương diện quan trọng của một game đều được định nghĩa sẵn trong   
  AndEngine, và rất dễ dàng để sử dụng. AndEngine có thể làm việc trên nhiều thiết bị khác nhau, với độ phân giải khác nhau. Hỗ trợ cảm ứng đa điểm, và hình nền động. Bằng những lợi thế đó, AndEngine rất thích hợp để xây dựng những game yêu cầu đồ họa cùng những hiệu ứng đẹp mắt.

2.5 CÔNG CỤ HỖ TRỢ

* IDE Eclipse
* Android SDK
* Thư viện AndEngine
* Máy ảo Genymotion

CHƯƠNG 3

TỔNG QUAN VỀ ANDENGINE



*Hình 3.1 : Logo AndEngine*

3.1 GIỚI THIỆU CHUNG

* AndEngine là một mã nguồn mở Android OpenGL, một bộ các công cụ để phát triển video game, được phát triển bởi Nicolas Gramlich.
* Nó bao gồm đầy đủ các thuộc tính quan trọng nhất của OpenGL , ngay cả khi bạn không có kinh nghiệm với OpenGL - bạn vẫn có thể tạo ra 1 game có đồ họa đẹp và đầy đủ tính năng hoặc 1 ứng dụng đẹp mắt sử dụng công cụ này, điều quan trọng nhất là các bạn cần có kinh nghiệm lập trình Java nhất định.

3.2 CÁC TÍNH NĂNG QUAN TRỌNG CỦA ANDENGINE

* Mở rộng hỗ trợ trên các thiết bị khác nhau (độ phân giải khác nhau).
* Hỗ trợ cảm ứng đa điểm.
* Phần vật lý mở rộng - Box2D
* TiledMaps trong hỗ trợ định dạng TMX.
* Hệ thống phân tán.
* Hỗ trợ hình nền động.

3.3 LỢI ÍCH TỪ VIỆC SỬ DỤNG ANDENGINE

* Đây là 1 Engine mã nguồn mở và hoàn toàn miễn phí.
* Các phương diện quan trọng cần có của bất kỳ trò chơi nào đều được định nghĩa sẵn trong bộ công cụ này.
* Các bạn không thực sự cần nhiều kinh nghiệm về OpenGL (nhưng sẽ hữu ích hơn rất nhiều nếu bạn đã hiểu biết rõ ràng về nó).
* Nhanh, hỗ trợ cho việc thiết kế tốt và được hỗ trợ sẵn các công cụ.
* Tương đối lớn, năng động và hữu ích cho làm việc theo tổ chức hay nhóm .
* Công cụ có tính mở rộng tốt.
* Có rất nhiều ví dụ có sẵn cho thấy khả năng hỗ trợ của công cụ này là rất lớn.

3.4 NHƯỢC ĐIỂM

* Tài liệu hướng dẫn code ít, hầu như là không có
* Đôi khi xử lý chậm hơn so với các công cụ khác ( tùy thuộc vào cách sử dụng andengine để làm việc gì)
* Việc thiếu tài liệu chính là nhược điểm lớn nhất vì khi bắt đầu làm việc với bộ công cụ này sẽ gặp rất nhiều khó khăn.( khó có thể hiểu đc các tham số, ...)
* Tuy nhiên thay vì việc thiếu tài liệu hướng dẫn, andengine có sẵn các ví dụ chúng ta có thể vọc từ code , đây cũng là cách học và tìm hiểu khá tốt về cách thức hoạt động và các phương thức có trong engine này

3.5 ĐỐI TƯỢNG SỬ DỤNG

* Các nhà phát triển trò chơi, ứng dụng
* Các nhà nghiên cứu
* Sinh viên, học sinh
  + Nếu bạn có ý tưởng thực hiện các ứng dụng animation hoặc trò chơi thì AndEngine là một thư viện hỗ trỡ đắc lực, dễ sử dụng, không tốn quá nhiều thời gian cho việc thực hiện.

3.6 MỘT SỐ GAME ĐƯỢC LÀM TỪ ANDENGINE

* Break Master

<https://play.google.com/store/apps/details?id=pl.com.hubson.dev.breakout>



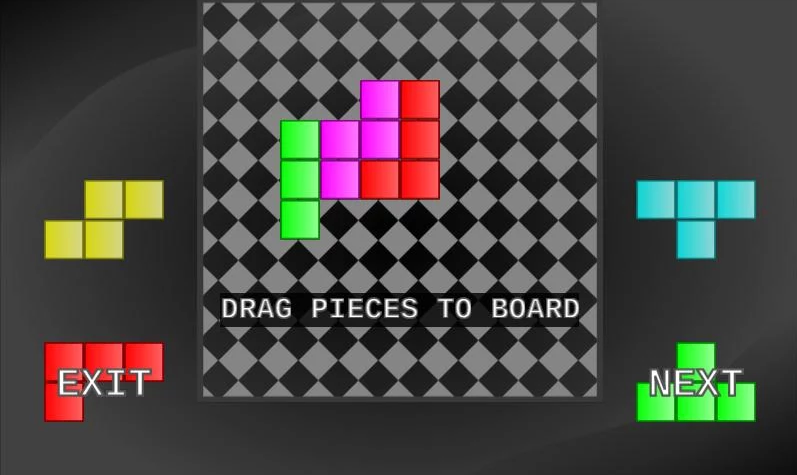
*Hình 3.2 Game Break Master*



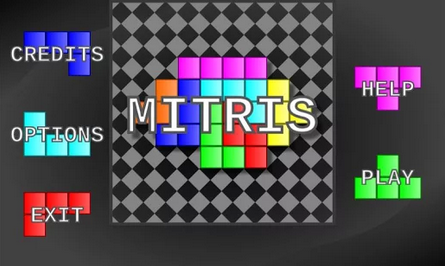
*Hình 3.3 Game Break Master*

* Mitris

<https://play.google.com/store/apps/details?id=com.damacasoft.mitris>



*Hình 3.4 Game Mitris*



*Hình 3.5 Game Mitris*

* Greedy Spiders

<https://play.google.com/store/apps/details?id=com.blyts.greedyspiders.free&feature=search_result#?t=W251bGwsMSwyLDEsImNvbS5ibHl0cy5ncmVlZHlzcGlkZXJzLmZyZWUiXQ>..

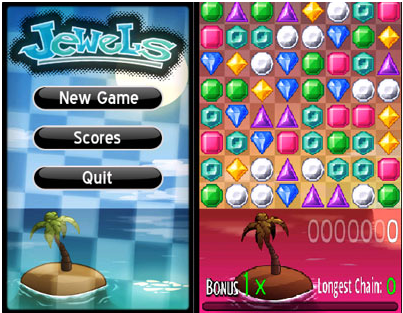


*Hình 3.6 Game Greedy Spiders*



*Hình 3.7 Game Greedy Spiders*

* Game kim cương



*Hình 3.8 Game Kim Cương*

3.7 CÁC KHÁI NIỆM CHÍNH TRONG LẬP TRÌNH ANDENGINE

Để có thể hiểu và làm việc tốt với AndEngine, trước hết chúng ta xem xét một số khái niệm chính thường được nhắc đến khi lập trình với AndEngine:

1. *Camera*

* Các "camera" của trò chơi xác định giao diện được trình bày cho người chơi. Nó giống  như một máy quay phim trong không gian hai chiều. Chiếc máy quay này có thể giãn và  phóng trên các cảnh để thay đổi khung nhìn trình bày. Sự co dãn và phóng to có thể được thực hiện dưới sự kiểm soát của người chơi hoặc được điều khiển theo chương trình.

1. *Scene*

* Một trò chơi, giống như một bộ phim, bao gồm một loạt các cảnh nơi hành động diễn   
  ra. Trong một bộ phim, những cảnh được biên tập lại với nhau theo một cách cố định. Trong  trò chơi, trình tự của cảnh được dẫn dắt bởi lối chơi của trò chơi. Trò chơi giống như bộ phim được chỉnh sửa từ trên cao nhìn xuống.

1. *Layer*

* Cảnh bao gồm các lớp đồ họa. Các lớp được chồng lên nhau, giống như các tế bào hoạt hình được sử dụng để làm phim hoạt hình theo cách cổ điển. Các lớp cũng có thể được sử dụng để giới thiệu hiệu ứng 2½D, ví dụ như việc điều chỉnh máy quay, các lớp gần hơn chuyển động nhanh hơn các lớp ở xa.

1. *Sprite*

* Những Sprite là những đại diện trực quan của các diễn viên trong bộ phim của chúng ta. Dù cho các sprite là hình động hay tĩnh, chúng vẫn thường di chuyển trong những màn chơi. Kết cấu Sprite thường được nạp từ một hình ảnh lớn mà bao gồm một tập hợp các hình ảnh họa tiết, được gọi là một sprite sheet

1. *Entity*

* Trong AndEngine, các thực thể là tất cả những thứ được vẽ trên màn hình. Các thực thể có thể là các sprite, các hình dạng hình học, hay các dòng kẻ trên màn hình. Tất cả các thực thể đều có các thuộc tính, chẳng hạn như màu sắc, góc quay, độ lớn,vị trí, và có thể được thay đổi bằng các Modifier.

1. *Modifier*

* Các modifier thay đổi các thuộc tính của một thực thể, và chúng là công cụ rất mạnh mẽ được sử dụng trong AndEngine. Chúng có thể được áp dụng trên bất kỳ thực thể nào, và sự thay đổi mà chúng gây ra có thể là trực tiếp hoặc diễn ra từ từ trong một thời gian nhất định.   
  Trong trò chơi, chúng ta thường sẽ sử dụng các modifier để tạo hiệu ứng với các sprite và các thực thể khác.

1. *Texture*

* Một texture là một kết cấu 2D, thường là đồ họa dạng bitmap. Textures xác định hình   
  dạng các thực thể, và phần lớn môi trường đồ họa OpenGL được xây dựng xung quanh việc sử dụng các texture.

1. *Texture Region*

* Một texture định nghĩa một hình ảnh bitmap hoàn tất, và một texture region định nghĩa một tập hợp con các vùng texture đó.

1. *Engine*

* Mỗi scene được chạy trên một engine. Nó điều khiển các hình ảnh động và các modifier biết khi nào phải cập nhật trạng thái, điều phối các bản vẽ thực tế, xử lý các sự kiện đầu vào người dùng (cảm ứng, phím, bộ cảm biến), và nói chung là quản lý tiến trình của trò chơi.   
  Engine cũng giống như các nhà sản xuất / đạo diễn của bộ phim của chúng ta, nói với mọi người những gì họ cần phải làm.

1. *BaseGameActivity*

* Lớp này, được mở rộng của lớp Activity Android, sẽ là cơ sở của mỗi scene trong trò   
  chơi của chúng ta. BaseGameActivity làm các công việc chung cho tất cả các scene, thiết lập các công cụ trò chơi, phù hợp với các yêu cầu vòng đời hoạt động Android.

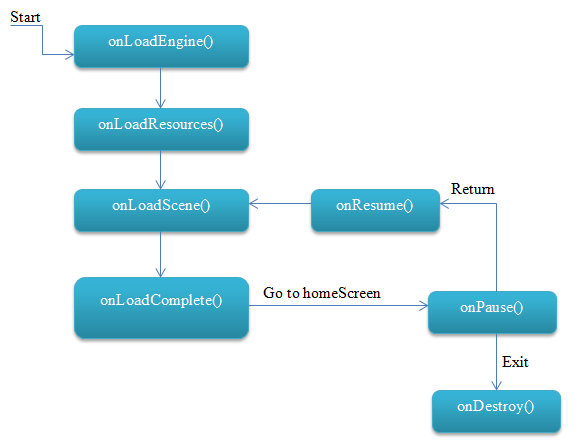
1. *Box2D Physics Engine*

* Physics Box2D là một thành phần mở rộng của AndEngine, nó hỗ trợ việc mô phỏng sự tương tác thực tế giữa các đối tượng vật lý trong các trường hợp sau đây:
  + Mô phỏng các tính chất vật lý của các vật rắn.
  + Ổn định xếp chồng.
  + Trọng lượng.
  + Các đơn vị được người dùng xác định.
  + Giải quyết hiệu quả cho sự tương tác / liên lạc.
  + Trượt ma sát.
  + Làm việc với các hình học: Hộp, hình tròn, đa giác.
  + Một số loại liên kết: khoảng cách, lăng trụ, ròng rọc,
  + Trạng thái ngủ (loại bỏ cơ thể bất động từ các mô phỏng cho đến khi chạm vào) .

3.8 VÒNG ĐỜI CỦA MỘT GAME ANDENGINE

*BaseGameActivity*

* Điều quan trọng đầu tiên, cần lưu ý đến việc khởi tạo game. Nhu cầu cơ bản của một   
  game khi khởi tạo bao gồm việc tạo ra engine, nạp tài nguyên của game, và thiết lập màn hình khởi tạo. Mỗi một game viết bằng AndEngine được thể hiện thông qua class BaseGameActivity. BaseGameActivity là một lớp con kế thừa từ class Activity của Android. Để hiểu được quá trình hoạt động của BaseGameActivity, ta xem xét vòng đời của nó.
* Một class BaseGameActivity chứa một số phương thức mà chúng ta cần phải ghi đè   
  trực tiếp. Đó là:
  + onCreateEngineOptions(): Sử dụng để khởi tạo các thành phần cơ bản của một game AndEngine như khởi tạo engineOptions, camera.
  + onCreateResources(): Sử dụng để khởi tạo tài nguyên cho game như khởi tạo texture, region, âm thanh, font chữ …
  + onCreateScene(): Khởi tạo các màn hình chơi, các nhân vật và các thành phần trong game như sprite, animatedSprite …
  + onPopulateScene(): Là nơi chúng ta gắn các entity con cho scene tương ứng.



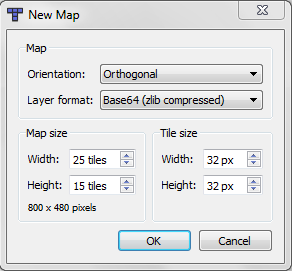
*Hình 3.9 Vòng đời của một Andengine*

3.9 NỀN TẢNG HỖ TRỢ

* Thư viện AndEngine được import, sử dụng trong các ứng dụng Android
* Các công cụ sử dụng để tạo ứng dụng Android nổi bật:
  + Eclipse tích hợp plugin ADT
  + Android Studio
  1. GIỚI THIỆU CÔNG CỤ TILED MAP
* **Tiled Map là gì?**
  + Tiled là một công cụ vẽ bản đồ được xây dựng nên bởi những hình ảnh nhỏ, còn được gọi là viên gạch . Nó hoạt động như một công cụ miễn phí cho phép dễ dàng tạo ra bố cục bản đồ. Nó là đủ linh hoạt để cho phép chỉ định nhiều hơn những thứ trừu tượng như khu vực va chạm. Nó lưu tất cả các dữ liệu này một cách thuận tiện, định dạng tmx chuẩn hóa.
  + Link tải phần mềm : <http://www.mapeditor.org/>

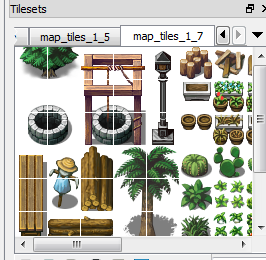
***Cách hoạt động của Tiled Map***

* + Tại cốt lõi của nó, quá trình thiết kế bằng cách sử dụng ngói để tạo bản đồ hoạt động bằng cách làm theo các bước sau:
  + Chọn kích thước bản đồ của bạn và kích thước gạch cơ sở.



*Hình 3.10 Một số hình ảnh của Tiled Map*

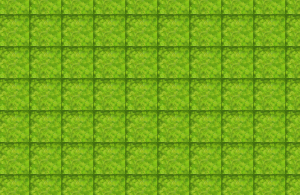
* + Thêm tilesets từ hình ảnh (s).



*Hình 3.11 Một số hình ảnh của Tiled Map*

các map\_tiles này chúng ta có thể lấy trên internet và kéo thả vào mục tilesets của Tiled Map.

* + Đặt tilesets trên bản đồ.
    - Kéo thả các map\_tiles vào vùng kích thước bản đồ đã tạo



*Hình 3.12 Một số hình ảnh của Tiled Map*

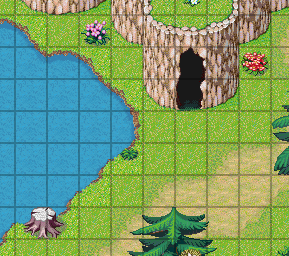
* + Thêm bất kỳ đối tượng bổ sung để đại diện cho một cái gì đó trừu tượng



*Hình 3.13 Một số hình ảnh của Tiled Map*

Chúng ta có thể thêm các hình ảnh khác vào trong bản đồ, mục đích để làm vật cản, cho nhân vật không thể đi qua đó được

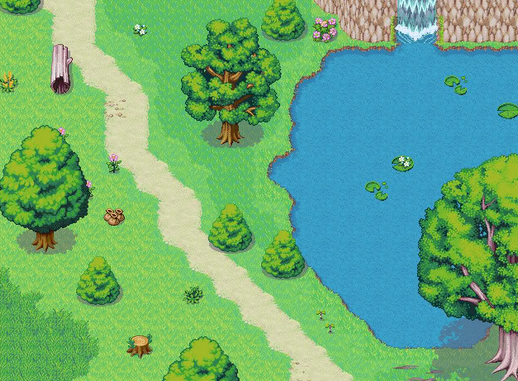
* + Lưu bản đồ như là một tập tmx.
  + Nhập khẩu các tập tin tmx và giải thích nó cho trò chơi của bạn.
* **Lý do sử dụng Tiled Map**
  + Có một hệ thống tiêu chuẩn hóa và một công cụ mạnh mẽ linh hoạt đã được đặt ra cho phép bạn tập trung vào những thứ quan trọng hơn trong trò chơi của bạn. Với Tiled bạn sẽ có thể để có một vài tilesets, tạo trình độ của bạn, và được trên con đường của bạn.
  + Bạn có thể tạo bản đồ của bạn mà không có một kích thước nghiêm ngặt về hình ảnh. Sau đó, bạn có thể sử dụng ngói cho tất cả những thực thể vô hình như khu vực va chạm, hoặc đối tượng bầy trong bản đồ của bạn
* **Một số hình ảnh được tạo từ Tiled Map của một số game**

****

*Hình 3.14 Một số hình ảnh của Tiled Map*

****

*Hình 3.15 Một số hình ảnh của Tiled Map*

****

*Hình 3.16 Một số hình ảnh của Tiled Map*

****

*Hình 3.17 Một số hình ảnh của Tiled Map*

* 1. MỘT SỐ THÀNH PHẦN ANDENGINE ĐƯỢC DÙNG TRONG GAME
     1. Cấu trúc của lớp BaseGameActivity

**public** **class** Example **extends** BaseGameActivity{

@Override

**public** Engine onLoadEngine() {

// Load kích thước màn hình

// set kiểu màn hình: ngang/dọc

// cho phép ứng dụng chèn âm thanh,...

**return** **null**;

}

@Override

**public** **void** onLoadResources() {

// **load dữ liệu, hình ảnh, font chữ,…**

}

@Override

**public** Scene onLoadScene() {

// **tạo các sprite, xử lý va chạm,….**

**return** **null**;

}

@Override

**public** **void** onLoadComplete() {

// **TODO** Auto-generated method stub

}

* + 1. Camera / EngineOptions
* Camera và EngineOptions là hai đối tượng cơ bản đầu tiên, cần được thiết lập ngay từ khi khởi tạo game để tạo ra camera hiển thị, cũng như các tùy chọn cho camera đó.
  + Camera: Hàm khởi tạo được sử dụng:
    - Camera(final float pX, final float pY, final float pWidth, final float pHeight)  
      Trong đó:  
      + pX, pY: là tọa độ bắt đầu của Camera trong màn hình hiển thị của thiết bị.  
      + pWidth, pHeight: là kích thước chiều ngang và chiều dọc của Camera.
  + EngineOptions: Hàm khởi tạo được sử dụng:
    - EngineOptions(final boolean pFullscreen, final ScreenOrientation pScreenOrientation,   
      final IResolutionPolicy pResolutionPolicy, final Camera pCamera)  
      Trong đó:  
      + pFullscreen: là chế độ toàn màn hình.  
      + pScreenOrientation: Chiều của màn hình.  
      pScreenOrientation có thể nhận các giá trị:  
      ScreenOrientation.PORTRAIT\_FIXED: màn hình xoay theo chiều dọc  
      ScreenOrientation.LANDSCAPE\_FIXED: màn hình xoay theo chiều ngang  
      + pResolutionPolicy: cho biết cách thức làm việc với các màn hình khác nhau.  
      pResolutionPolicy có thể là:  
      RatioResolutionPolicy(final float pWidthRatio, final float pHeightRatio): mở rộng đồ   
      họa để điền vào nhiều nhất có thể các phần trong màn hình mà vẫn duy trì tỉ lệ ban đầu của đối tượng đồ họa.
    1. BitmapTextureAtlas / Texture
* Bất cứ khi nào chúng ta muốn nạp một đối tượng đồ họa vào trong game AndEngine, điều bắt buộc phải làm đó là khai báo cho đối tượng đồ họa đó một nơi để chứa thông tin các điểm ảnh. TrongAndEngine, BitmapTextureAtlas có thể hiểu như một khung tranh, còn Texture sẽ đóng vai trò như một bức hình để vào trong khung tranh BitmapTextureAtlas đó. Các BitmapTextureAtlas và các Texture phải được nạp vào lúc khởi tạo game, sẽ được đặt trong hàm onCreateResources();
  + BitmapTextureAtlas: Hàm khởi tạo được sử dụng
    - BitmapTextureAtlas(final TextureManager pTextureManager, final int pWidth, final int   
      pHeight, final TextureOptions pTextureOptions)   
      Trong đó:   
      + pTextureManager: là bộ quản lý các Texture, ám chỉ đến game của chúng ta. Thông thường được khai báo là Game.getTextureManager().  
      + pWidth, pHeight: là kích thước chiều ngang và chiều dọc của khung tranh.  
      + pTextureOption: là cách thức khung tranh sẽ được hiển thị.
  + TextureRegion : được sử dụng để chứa thông tin đồ họa của những Sprite tĩnh. Cách khai báo được sử dụng:
    - TextureRegion createFromAsset(final BitmapTextureAtlas pBitmapTextureAtlas, final   
      Context pContext, final String pAssetPath, final int pTextureX, final int pTextureY)  
      Trong đó:   
      + pBitmapTextureAtlas: là khung tranh BitmapTextureAtlas sẽ chứa tấm hình này.  
      + pContext: là Activity chứa đối tượng này.  
      + pAssetPath: đường dẫn đến bức hình.  
      + pTextureX, pTextureY: hai tọa độ đầu tiên trong khung tranh BitmapTextureAtlas để   
      đặt tấm hình này.
  + TiledTextureRegion: được sử dụng để nạp thông tin đồ họa của những Sprite động, tức là những TiledSprite. Trong mỗi bức hình có nhiều phần nhỏ, mỗi phần là một trạng thái của Sprite theo thời gian.
    - Cách khai báo được sử dụng:  
      TiledTextureRegion createTiledFromAsset(final BitmapTextureAtlas   
      pBitmapTextureAtlas, final Context pContext, final String pAssetPath, final int pTextureX,   
      final int pTextureY, final int pTileColumns, final int pTileRows)  
      Trong đó:   
      + pBitmapTextureAtlas: là khung tranh BitmapTextureAtlas sẽ chứa tấm hình này.  
      + pContext: là Activity chứa đối tượng này.   
      + pAssetPath: đường dẫn đến bức hình.  
      + pTextureX, pTextureY: hai tọa độ đầu tiên trong khung tranh BitmapTextureAtlas để   
      đặt tấm hình này.  
      + pTileColumns, pTileRows: là số cột, số hàng của bức hình.
    1. Sprite / AnimatedSprite
* Sprite là thành phần chính trong game. Bất kì game nào cũng cần có Sprite, thậm chí là nhiều Sprite trong mỗi màn chơi. Sprite có thể ở dạng tĩnh (Sprite) không thay đổi hình dạng, hay động (AnimatedSprite) – thay đổi hình dạng theo thời gian.Việc khai báo Sprite / AnimatedSprite sẽ được đặt trong hàm onCreateScene();
  + Sprite : Các hàm khởi tạo được sử dụng:
    - public Sprite(final float pX, final float pY, final float pWidth, final float pHeight, final  pTextureRegion)  
      Trong đó:  
      + pX, pY: là tọa độ bắt đầu của Sprite đặt trong Scene  
      + pWidth, pHeight: là kích thước chiều ngang và chiều dọc của Sprite. pTextureRegion: là tấm hình sẽ được nạp vào cho Sprite đó.
  + AnimatedSprite: Để tạo được một Sprite động thay vì dùng Sprite thông thường, ta cần sử dụng AnimatedSprite.Hàm khởi tạo được sử dụng:
    - AnimatedSprite(final float pX, final floatpY, pTiledTextureRegion)  
      Trong đó:  
      + pX, pY: là tọa độ bắt đầu của AnimatedSprite đặt trong Scene  
      + pWidth, pHeight: là kích thước chiều ngang và chiều dọc của Sprite, pTiledTextureRegion: là tấm hình sẽ được nạp vào cho Sprite đó.
    1. Font
* Trong game chắc chắn bạn sẽ phải sử dụng những đoạn text để thể hiện nội dung, ví dụ hiển thị dialog, đưa ra trợ giúp cho người chơi Thay vì sử dụng font mặc định của hệ thống, AndEngine cũng cho phép chúng ta sử dụng font tùy chọn của riêng bạn. Điều này vừa tạo ra nét riêng cho game, đồng thời tăng hiệu quả về đồ họa cho game.
* Để thay đổi một font chữ cho đối tượng trong AndEngine, chúng ta sử dụng class Font. Việc cài đặt và tải font được thực hiện trong phương thức onCreateResources(). Cách khai báo được sử dụng:
  + Font createFromAsset(final FontManager pFontManager, final ITexture pTexture, final AssetManager pAssetManager, final String pAssetPath, final float pSize, final boolean pAntiAlias, final int pColor). Trong đó
    - pFontManager: bộ phận quản lý font. Thông thường được khai báo là mGame.getFontManager();
    - pTexture: tương tự như một đối tượng đồ họa, font cũng cần được tải vào trong một texture để sau đó đưa vào sử dụng.
    - pAssetManager: bộ phận quản lý các tài nguyên trong asset. Thông thường được khai báo là mGame.getFontManager();
    - pAssetPath: đường dẫn đến font trong thư mục asset của project.
    - pSize: cỡ chữ.
    - pAntiAlias = true: bật chế độ anti-alias
    - pColor: màu sắc của chữ.
    1. Text
* Sau khi đã tải xong font, chúng ta có thể sử dụng font đó để format cho các đoạn text trong game. AndEngine cung cấp cho chúng ta đối tượng Text để thể hiện các chuỗi kí tự. Việc sử dụng Text cũng tương tự như sử dụng một đối tượng Sprite, và được đặt trong phương thức onCreateScene() của game activity. Hàm khởi tạo được sử dụng:
  + Text(final float pX, final float pY, final IFont pFont, final CharSequence pText, final int   
    pCharactersMaximum, final VertexBufferObjectManager pVertexBufferObjectManager)  
    Trong đó:
    - pX, pY: là tọa độ bắt đầu của Text đặt trong Scene.
    - pFont: là font đã được nạp trong phương thức onCreateResource().
    - pText: nội dung đoạn text.
    - pCharactersMaximum: Số kí tự tối đa của text.
    - pVertexBufferObjectManager: là đối tượng quản lý bộ đệm để tải các dữ liệu về đồ họa

**CHƯƠNG 4**

**PHÂN TÍCH VÀ THIẾT KẾ**

**4.1 PHÂN TÍCH HỆ THỐNG**

4.1.1 Giới thiệu chung



*Hình 4.1 Game logo*

* Tên game : Bắn Vịt
* Game gồm có 3 bản đồ khác nhau tương ứng với độ khó khác nhau
* Mỗi bản đồ sẽ có số lượng con vịt khác nhau, tốc độ bay khác nhau
* Nhiệm vụ người chơi ở mỗi bản đồ là phải bắn đủ số lượng con vịt trong khoảng thời gian cho phép mới có thể qua bản đồ mới.
* Game offline, có 1 nhân vật chơi
* Lưu điểm các màn chơi

4.1.2 Thể loại

* Game hành động

4.1.3 Mô tả kịch bản

* Mỗi bản đồ chơi, sẽ có 1 số lượng vịt bay theo 1 tốc độ ngẫu nhiên từ phải sang trái màn hình thiết bị, mỗi giây sẽ xuất hiện 1 con vịt
* Quy tắc chơi:
  + Người chơi sẽ được di chuyển nhân vật sang phải, trái màn hình bằng control được thiết lập và bắn các con vịt trên màn hình bằng cách chạm vào màn hình thì nhân vật sẽ bắn ra vũ khí, khi vũ khí chạm con vịt thì con vịt sẽ chết (không giới hạn số lượng vũ khí bắn ra).
  + Có các hiệu ứng, âm thanh khi va chạm giữa các đối tượng
  + Mức độ khó sẽ tăng dần qua các bản đồ tiếp theo, chẳng hạn như vịt sẽ bay tốc độ cao hơn, số lượng nhiều hơn,….
  + Xét thắng thua trong game:
    - Mỗi con vịt bị bắn thì người chơi sẽ được 1 điểm, ngoài ra mỗi con vịt chết thì ngẫu nhiên, sẽ được 2 điểm, 3 điểm tùy vào xác suất ra.
    - Bắn được đủ số điểm quy định ở mỗi màn chơi thì sẽ nhận được thông báo thắng và qua bản đồ tiếp theo
    - Game có các chướng ngại vật, nếu người chơi bắn trúng thì sẽ giảm thời gian hoặc bị trừ điểm chơi.
  + Bản đồ trong game là 1 khung cảnh gồm các hình ảnh như ảnh nền, ảnh dưới mặt đất, trên trời, được lồng vào nhau và cho nó di chuyển liên tục, chúng ta có cảm giác như khung cảnh đang di chuyển

4.1.4 Phân tích kịch bản

* Vì là một game hành động (bắn), thử thách người là phải bắn, né các mục tiêu. Bởi vậy chúng ta cần một màn hình (cảnh trong game) để hiển thị nhân vật chơi và các mục tiêu cần bắn.
* Ngoài ra, game còn có thêm các hiệu ứng hình ảnh động để làm tăng tính hấp dẫn với người chơi, bởi vậy cần bố trí thêm một số đối tượng trong game như cây cỏ, mây trên bầu trời chuyển động,…..
* Các đối tượng được bố trí nằm đè lên trên màn hình chính của game
* Game có tính điểm, xét thắng thua cho người chơi nên cần bố trí thêm các thông số trong game, các hình ảnh, hiệu ứng báo cho người chơi biết đã thắng hay thua,….

4.1.5 Mục đích

* Mục đích của trò chơi là đem lại giờ phút thư giãn để chơi game cho tất cả mọi người, các đối tượng: học sinh, sinh viên, nhân viên, …..

4.1.6 Các lớp giao diện

* GameDesign: gồm các thuộc tính và các phương thức chuyên tạo ra các đối tượng đồ họa như Sprite, Image, TiledLayer,….
* GameScene: Lớp này chứa tất cả các đối tượng đồ họa lấy từ lớp GameDesign
  + Lớp này có rất nhiều phương thức như :
    - Vẽ các đối tượng lên màn hình
    - Chờ và xác nhận chạm màn hình
    - Tạo các hình ảnh chuyển động, các nhân vật
    - Xác nhận trạng thái của game : Win, Game Over , Running
  + Trong lớp GameScene có một luồng chính thực hiện tất cả các công việc trong đó Main Loop

While(running){

* + - * + Xử lý sự kiện chạm màn hình thiết bị
        + Cập nhật màn hình
        + Vẽ lại màn hình
        + Ghi điểm
        + Xử lý thời gian tính toán cho mỗi vòng lặp

}

4.1.7 Thiết bị

* Yêu cầu:
  + Thiết bị di động chạy hệ điều hành Android : mobile, tablet
  + Hệ điều hành Android 2.2 trở lên
  + Không cần cấu hình cao

4.1.8 Platform

* Java, Android SDK

4.1.9 Level

* Mỗi level sẽ có 1 bản đồ, số lượng mục tiêu, độ khó khác nhau

3.1.10 Kỹ thuật

* Phần cứng:
  + Phím home mặc định của thiết bị: thoát ứng dụng, nhưng ở trạng thái pause game, chưa thoát hoàn toàn
  + Phím back: trở lại giao diện new game, hoặc các map trước đó
* Yêu cầu mạng:
  + Game offline nên không cần thiết bị có kết nối internet
    - Âm thanh
* Game có các âm thanh trong quá trình chơi:
  + Âm thanh đạn bắn ra
  + Nhạc nền
  + Âm thanh khi bắn chết các mục tiêu
  + Âm thanh khi nhân vật va chạm các mục tiêu
    - Hiệu ứng
* Game có các hiệu ứng trong quá trình chơi:
  + Va chạm giữa nhân vật với mục tiêu
  + Va chạm giữa đạn bắn ra với mục tiêu
  + Thưởng điểm khi bắn được
    - Điểm thưởng
* Trong quá trình bắn các mục tiêu thì sẽ ngẫu nhiên được nhận được các điểm thưởng, x2, x3 so với mặc định mỗi mục tiêu là 1 điểm
* Nhằm ghi được nhiều điểm hơn, nhanh chóng qua bản đồ mới
  + - Tùy chọn trò chơi
* Tắt/ mở âm thanh
* Reset game
* Dừng game/ tiếp tục
* Thoát

**4.2 BIỂU ĐỒ USE – CASE**

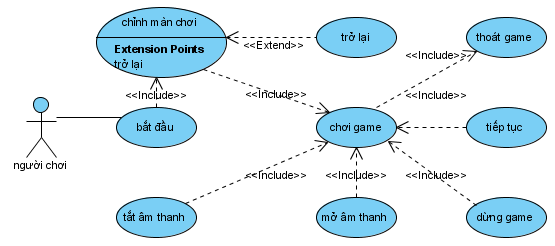
4.2.1 Danh sách actor

|  |  |  |
| --- | --- | --- |
| Tên | Thể hiện | Vai trò |
| Người chơi | Người chơi game | Là người chơi của game |

4.2.2 Danh sách use case

|  |  |  |
| --- | --- | --- |
| STT | Tên Usecase | Ý Nghĩa |
| 1 | Bắt đầu chơi | Bắt đầu trò chơi |
| 2 | Chọn màn chơi | Chọn màn chơi |
| 3 | Chơi game | Người chơi chơi game |
| 4 | Lưu game | Sẽ lưu lại màn chơi khi người chơi thoát game |
| 5 | Thoát game | Người chơi thoát không chơi nữa |
| 6 | Tắt âm thanh | Tắt âm thanh |
| 7 | Mở âm thanh | Mở âm thanh |
| 8 | Dừng game | Dừng game khi đang chơi |
| 9 | Tiếp tục | Tiếp tục chơi game |
| 10 | Trở lại | Trở lại màn hình trước đó |

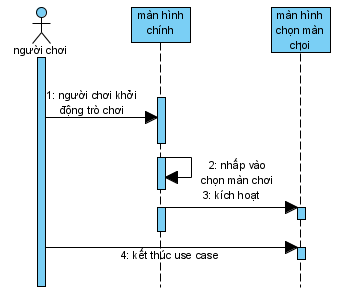
4.2.3 Biểu đồ use-case



*Hình 4.2 Use case tổng quát*

4.2.4 Đặc tả use – case

* Đặc tả use case “bắt đầu chơi”



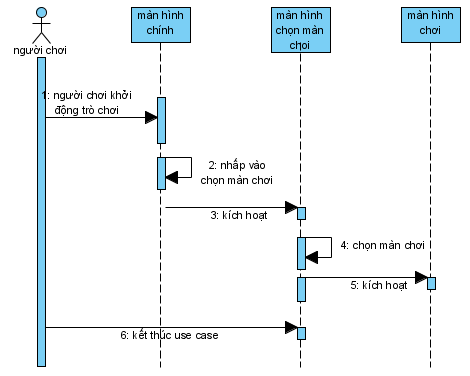
*Hình 4.3 lược đồ trình tự use case “bắt đầu chơi”*

* Tóm tắt
  + use case này là quá trình người chơi vào chọn màn chơi
* dòng sự kiện:
  + dòng sự kiện chính:
    - trò chơi được khởi động lên
    - người chơi sử dụng use case này để gọi use case “chọn màn chơi”
    - kết thúc use case
  + dòng sự kiện phụ

không có

* trạng thái hệ thống trước khi bắt đầu thực hiện use case
  + không đòi hỏi yêu cầu gì trước đó
* trạng thái hệ thống sau khi bắt đầu thực hiện use case
  + chuyển sang màn hình chọn màn chơi

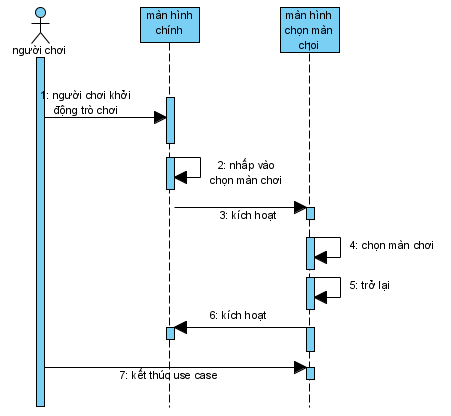
4.2.5 Đặc tả use case “chọn màn chơi”



*Hình 4.4 lược đồ trình tự use case “chọn màn chơi”*

* Tóm tắt:
  + use case là quá trình người chơi chọn màn chơi
* Dòng sự kiện:
  + Dòng sự kiện chính:
    - người chơi chọn màn chơi
    - chuyển sang màn hình chơi trò chơi
    - người chơi bắt đầu chơi game
    - kết thúc use case
* trạng thái hệ thống trước khi bắt đầu thực hiện use case
  + người chơi phải thực hiện use case “bắt đầu chơi”
* trạng thái hệ thống sau khi thực hiện use case
  + chuyển sang màn hình chơi trò chơi

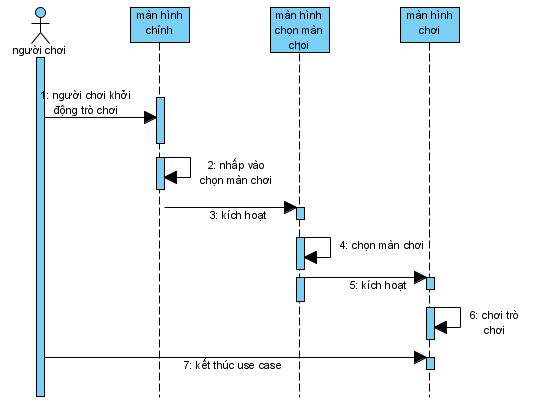
4.2.6 Đặc tả use case “trở lại”



*Hình 4.5 lực đồ trình tự use case “trở lại”*

* Tóm tắt:
  + use case này để quay lại màn hình trước đó
* Dòng sự kiện:
  + Dòng sự kiện chính:
    - use này bắt đầu khi người chơi chọn trở lại trong màn hình chọn màn chơi
    - hệ thống đưa người chơi quay lại với màn hình bắt đầu
    - kết thúc use case
* trạng thái hệ thống trước khi bắt đầu thực hiện use case
  + không có
* trạng thái hệ thống sau khi bắt đầu thực hiện use case
  + người chơi trở về giao diện bắt đầu

4.2.7 Đặc tả use case “chơi game”



*Hình 4.6 Lược đồ trình tự use case “chơi game”*

* Tóm tắt:
  + use case này là quá trình người chơi chơi game
* Dòng sự kiện:
  + Dòng sự kiện chính:
    - người chơi bắt đầu chơi
    - người chơi bắn “vịt”
    - bắn đủ 100 con vịt
    - người chơi thắng màn chơi
    - chuyển sang màn hình chọn màn chơi
    - người chơi được chọn màn chơi tiếp theo.
    - Kết thúc use case
  + Dòng sự kiện phụ
    - người chơi va chạm với “vịt”
    - kết thúc màn chơi
* trạng thái hệ thống trước khi bắt đầu thực hiện use case
  + use case này chỉ được gọi khi đã thực hiện use case “chọn màn chơi”
* trạng thái hệ thống sau khi bắt đầu thực hiện use case
  + trạng thái hệ thống không đổi sau khi thực hiện use case này

4.2.8 Đặc tả use case “thoát game”

* Tóm tắt:
  + use này là quá trình người thoát khỏi trò chơi
* Dòng sự kiện:
  + Dòng sự kiện chính:
    - use này bắt đầu khi người chơi chọn thoát game ở màn hình bắt đầu
    - người chơi thoát khỏi trò chơi
    - kết thúc use case
* trạng thái hệ thống trước khi thực hiện use case
  + Không có
* trạng thái hệ thống sau khi thực hiện use case
  + người chơi thoát khỏi trò chơi

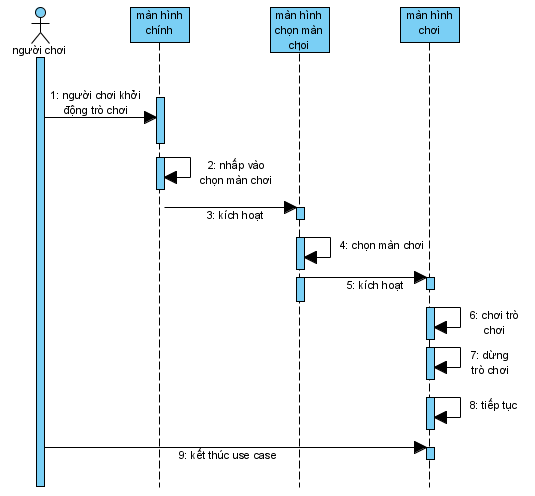
4.2.9 Đặc tả use case “dừng game”



*Hình 4.7 lược đồ trình tự use case “dừng game”*

* Tóm tắt:
  + người chơi sử dụng use case này để dừng trò chơi lại
* Dòng sự kiện:
  + Dòng sự kiện chính:
    - use này được sử dụng khi người chơi nhấp vào icon dùng trò chơi
    - game dừng lại
    - kết thúc use case
  + Dòng sự kiện phụ:
    - Không có
* trạng thái hệ thống trước khi thực hiện use case
  + người chơi đang chơi game
* trạng thái hệ thống sau khi thực hiện use case
  + Game dừng lại

4.2.10 Đặc tả use case “tiếp tục”



*Hình 4.8 lược đồ trình tự use case “tiếp tục”*

* Tóm tắt:
  + người chơi sử dụng use case này để tiếp tục chơi sau khi đã sử dụng use dừng game
* Dòng sự kiện:
  + Dòng sự kiện chính:
    - use này được sử dụng khi người chơi nhấp vào icon tiếp tục
    - người chơi tiếp tục chơi game
    - kết thúc use case
* trạng thái hệ thống trước khi thực hiện use case:
  + Game đang dừng
* trạng thái hệ thống sau khi thực hiện use case:
  + Người chơi tiếp tục chơi game

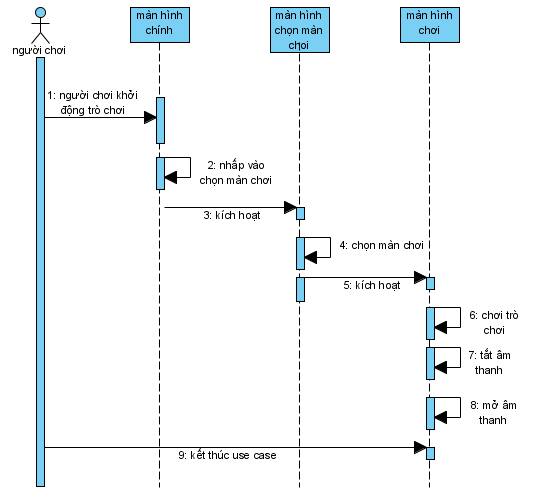
4.2.11 Đặc tả use case “tắt âm thanh”



*Hình 4.9 lược đồ trình tự use case “tắt âm thanh”*

* Tóm tắt:
  + người chơi sử dụng use case này để tắt âm thanh của trò chơi
* Dòng sự kiện:
  + Dòng sự kiện chính:
    - người chơi nhấp vào icon tắt âm thanh
    - hệ thống tắt âm thanh của trò chơi
    - kết thúc use case
* trạng thái hệ thống trước khi thực hiện use case
  + Không có
* trạng thái hệ thống sau khi thực hiện use
  + Hệ thống tắt âm thanh của trò chơi

4.2.12 Đặc tả use case “mở âm thanh”



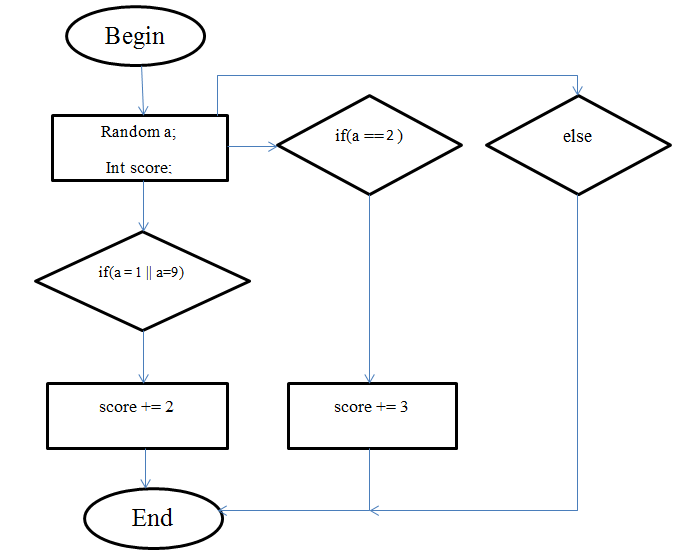
*Hình 4.10 lược đồ trình tự use case “mở âm thanh”*

* Tóm tắt:
  + người dùng sử dụng use case này để mở lại âm thanh của game
* Dòng sự kiện:
  + Dòng sự kiện chính:
    - người chơi nhấp vào icon mở âm thanh
    - hệ thống mở lại âm thanh của trò chơi
    - kết thúc use case
  + Dòng sự kiện phụ:
    - Không có
* trạng thái hệ thống trước khi thực hiện use case
  + Âm thanh của trò chơi tắt
* trạng thái hệ thống sau khi thực hiện use case
  + Hệ thống mở lại âm thanh của trò chơi

4.3 MỘT SỐ XỬ LÝ TRONG GAME

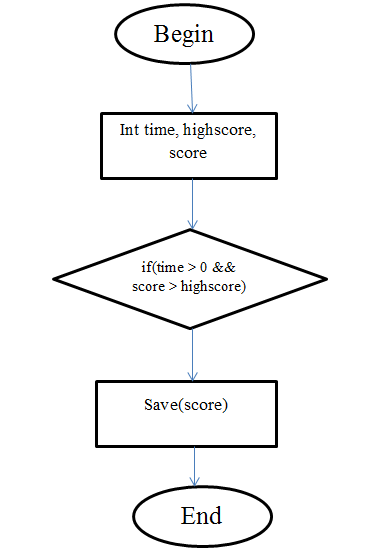
4.3.1 Xử lý điểm thưởng ngẫu nhiên.

Với mỗi con vịt bị bắn thì người chơi sẽ được cộng 1 điểm, tuy nhiên để kết thúc bản đồ nhanh hơn thì trong game còn có cơ hội bắn ngẫu nhiên sẽ được 2 , 3 điểm



Hình 4.11 thuật toán điểm thưởng ngấu nhiên

4.3.2 Xử lý lưu điểm tốt nhất



Hình 4.12 thuật toán lưu điểm

4.32 Xử lý load điểm lên khi khởi động game



Hình 4.13 thuật toán xử lý load điểm

4.4 THIẾT KẾ ĐỒ HỌA

****

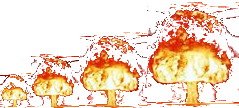
*Hình 4.14 Nhân vật*

*D:\Android\project\BanVit\assets\gfx\VitBay.png*

*Hình 4.15 Các mục tiêu để bắn*

*D:\Android\project\BanVit\assets\gfx\playpause.PNGD:\Android\project\BanVit\assets\gfx\volume.PNG*

*Hình 4.16 Các tủy chỉnh*

**

*Hình 4.17 hiệu ứng va chạm các đối tượng*

**

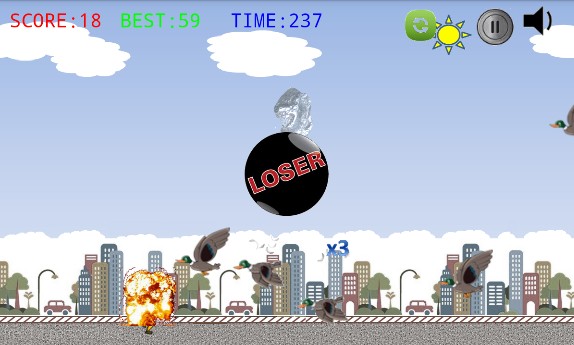
*Hình 4.18 Control điều khiển nhân vật*

**

*Hình 4.19 Thông báo thắng, thua*



*Hình 4.20 giao diện chính*



*Hình 4.21 Thông báo nhân vật chơi khi thua*



*Hình 4.22 Thông báo nhân vật chơi khi thắng*



*Hình 4.23 Hiệu ứng khi các con vịt bị bắn*



Hình 4.24 Hỗ trợ cảm ứng đa điểm

CHƯƠNG 5

KẾT QUẢ THỰC HIỆN VÀ HƯỚNG PHÁT TRIỂN

5.1 KẾT QUẢ ĐẠT ĐƯỢC

Sau hơn 3 tháng thực hiện đồ án với các yêu cầu đặt ra, đến nay căn bản đồ án đã đạt được một số kết quả như sau:

* Game căn bản đã hoàn thành.
* Xử lý va chạm giữa các đối tượng trong game.
* Có các map level khác nhau và độ khó tăng theo map level.
* Có lưu lại thời gian cao nhất ở mỗi map để tăng sự cạnh tranh cho game.
* Có thể chạy trên các thiết bị Android có cấu hình thấp.

**5.2 CÁC HẠN CHẾ CỦA ĐỀ TÀI**

* Hình ảnh game chưa được đẹp.
* Âm thanh chưa thật tốt.
* Game còn đơn giản.

**5.3 HƯỚNG PHÁT TRIỂN**

* Trong thời gian tới em sẽ cố gắng phát triển cho chương trình có thể hoàn thiện hơn về mặt các chức năng và phát triển cho chương trình có thể triển khai trên môi trường mạng, bluetooh.
* Tìm hiểu thêm một số ngôn ngữ, các phần mềm ứng dụng để nâng cao giao diện đồ họa đẹp mắt, thân thiện hơn.

TÀI LIỆU THAM KHẢO

[1] <http://www.matim-dev.com/tutorials.html>

* Thời gian tham khảo 9/2014.
* Giới thiệu AndEngine.
* Cách sử dụng các hàm trong AndEngine.

[2] <http://www.andengine.org/forums/>

* Thời gian tham khảo 9/2014.
* Diễn đàn hỏi đáp các vấn đề về AndEngine.

[2] <https://code.google.com/p/andengineexamples/>

* Thời gian tham khảo 9/2014.
* Một số code các hàm xử lý, tính tọa độ, đường bay của các mục tiêu.

[4] Tài liệu Andengine for Android Game Development Cookbook

* Năm xuất bản: 2013.
* Tài liệu từ trang: www.PacktPub.com.

[5] <http://stackoverflow.com/questions>

* Thời gian tham khảo 10/2014.
* Các câu hỏi về code.