**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

****-----🙚🙘🕮🙚🙘-----

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**NHẬP MÔN PHÁT TRIỂN GAME**

**Đề tài: Lập trình Game Aladdin**

*Giảng viên hướng dẫn:*

ThS. Nguyễn Vĩnh Kha

*Nhóm thực hiện:*

Lê Võ Gia Khang - 15520341

Trần Minh Hoàng Long - 15520448

Tạ Ngọc Hoàng Lâm - 15520410

TPHCM, Ngày 23 tháng 12 năm 2017

Mục lục

[**I – Lời cảm ơn** 4](#_Toc502268017)

[**II – Giới thiệu về game** 5](#_Toc502268018)

[1. Cốt truyện 5](#_Toc502268019)

[2. Gameplay 5](#_Toc502268020)

[3. Hệ thống đối tượng và Items 6](#_Toc502268021)

[3.1. Objects 6](#_Toc502268022)

[3.2. Enemies 7](#_Toc502268023)

[3.3. Boss 9](#_Toc502268024)

[3.4. Items 10](#_Toc502268025)

[4. Tương tác game 11](#_Toc502268026)

[5. Sơ đồ luồng 12](#_Toc502268027)

[5.1. Cấu trúc Framework 12](#_Toc502268028)

[5.2. GameObject 12](#_Toc502268029)

[5.3. State 16](#_Toc502268030)

[5.4. Scene 19](#_Toc502268031)

[6. Các màn chơi 19](#_Toc502268032)

[Agrabah Market (Lv1Scene): 19](#_Toc502268033)

[Jafar’s palace (BossScene): 19](#_Toc502268034)

[MenuSelector: 20](#_Toc502268035)

[DeathScene: 21](#_Toc502268036)

[CompleteScene: 21](#_Toc502268037)

[**III – Cấu trúc game, các kỹ thuật và áp dụng** 22](#_Toc502268038)

[1. Các kỹ thuật áp dụng trong game 22](#_Toc502268039)

[1.1 Thuật toán xử lý va chạm AABB 22](#_Toc502268040)

[1.2 Thuât toán xử lý va chạm SweptAABB 22](#_Toc502268041)

[1.3 Transformation 23](#_Toc502268042)

[1.4 Thuật toán phân hoạch không gian QuadTree 23](#_Toc502268043)

[**IV – Kết luận** 24](#_Toc502268044)

[1. Đánh giá chung 24](#_Toc502268045)

[2. Bảng phân công công việc cụ thể 25](#_Toc502268046)

[3. Áp dụng kiến thức đã học 25](#_Toc502268047)

[4. Phương hướng phát triển 25](#_Toc502268048)

[5. Các tài liệu tham khảo 26](#_Toc502268049)

# **I – Lời cảm ơn**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt quá trình học tập môn “Nhập môn phát triển Game”, chúng em đã nhận được rất nhiều sự chỉ dạy cũng như khắt khe của thầy để chúng em từng bước phát triển bản thân và hoàn thiện game hơn.

Với lòng biết ơn sâu sắc nhất, chúng em xin gửi đến thầy Nguyễn Vĩnh Kha ở Khoa Công nghệ phần mềm – Trường Đại Học Công Nghệ Thông Tin đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian môn học. Và đặc biệt, đối với các bạn khoa Công nghệ phần mềm nói chung và chúng em nói riêng, đây là môn học khó khăn và đầy thử thách. Chính vì thế, được thầy tận tâm chỉ bảo là một đặc ân vô cùng lớn.

Bài báo cáo này được viết sau khi chúng em hoàn thành môn học. Tuy nhiên trong quá trình học tập vẫn còn nhiều thiếu sót, chúng em rất mong nhận được những ý kiến đóng góp của thầy để chúng em được hoàn thiện hơn.

Lời cuối cùng, chúng em xin kính chúc thầy thật nhiều sức khỏe và duy trì ngọn lửa nhiệt huyết để tiếp tục là người lái đò dìu dắt chúng em sang sông, chắp cánh cho ước mơ bé bỏng của chúng em tiếp tục bay cao và xa hơn nữa.

Một lần nữa, chúng em xin chân thành cảm ơn thầy!

Trân trọng.

Thành phố Hồ Chí Minh, ngày 23 tháng 12 năm 2017

**Nhóm sinh viên thực hiện**

*( Đại diện )*

**Lê Võ Gia Khang**

# **II – Giới thiệu về game**

## Cốt truyện

Tìm mạo hiểm, và bạn sẽ tìm thấy Aladdin. Anh ấy luôn ở nơi hành động, leaping trên rooftops và chạy qua nhộn nhịp và nhộn nhịp của Agrabah.

Các đường phố là chock - đầy đủ của con dao jugglers, snake charmers, kẻ trộm giỏ và, ồ vâng, một đội tuyển cỡ Sultan của Đội bảo vệ Cung điện, một phần của Royal Deceiver ..., làm cho Cố vấn đó ... Quân Jafar luôn hiện diện.

Có vấn đề xảy ra trong cung điện của Sultan, với công chúa Jasmine bị mắc kẹt ở giữa. Greedy Jafar sắp sửa kéo một số thủ thuật rất tồi tệ ra khỏi turban. Và Aladdin sắp bị bắt trong một cuộc đuổi tốc độ cao đầy ma thuật, hành động và những mối nguy hiểm không kể đến. Chỉ có những con chuột chạy nhanh nhất mới có thể để thoát khỏi họ!

Không phải là một cuộc phiêu lưu bình thường mà Aladdin sắp phải đối mặt. Anh ta phải chiến đấu thông qua Agrabah, thoát khỏi dungeon của Sultan, sống sót trong Động Kỳ diệu, và cướp bóng đèn của Genie và chiến đấu với Jafar bằng chính bí mật của mình sanctum. Aladdin đã có một vài thủ thuật streetwise của riêng mình, cộng với một sáng scimitar để đảm bảo rằng anh ta giao dịch slash cho slash. Và có thể ... chỉ có thể ...ông sẽ có được một hoặc hai. Nhưng ngay cả đối với con chuột đường thông minh nhất thế giới, cũng có

Nguy hiểm phía trước!

## Gameplay

Aladdin là 1 thể loại game đi cảnh, người chơi sẽ phải vượt qua các vật cản như enemies, các bức tường, lửa, rắn,… bằng cách sử dụng các kỹ thuật: nhảy, ném táo, chém, leo dây, đẩy,… để đạt đến mục tiêu là di chuyển và sống sót thành công qua các màn chơi, tiêu diệt trùm cuối (Jafar) để cứu lấy công chúa.

Khởi đầu game với map Agrabah Market, nhân vật sẽ được cung cấp 3 mạng và 10 táo, di chuyển và ăn các táo có sẵn trên map để có thêm được táo. Táo dùng để ném và tiêu diệt mục tiêu như 1 loại vũ khí. Người chơi chỉ có thể mang tối đa 99 táo trên người. Khi ném hết táo thì sẽ không thể có động tác ném được nữa. Ngoài ra, táo là vũ khí duy nhất có thể tiêu diệt được Jafar.

Đao, là 1 vũ khí luôn luôn sát cánh cùng Aladdin. Được trang bị để tấn công tầm gần nhằm tiêu diệt mục tiêu trong khoảng cách cho phép. Khác với táo, đao không thể mất hoặc lấy thêm. Tuy nhiên, với một số loại enemy nhất định thì đao có sát thương thấp hơn táo. Đao không hiệu quả với Jafar.

Sau mỗi lần chết, nhân vật sẽ được tái sinh ngay tại vị trí đó trong màn chơi. Tuy nhiên sẽ không thể bắt đầu lại màn chơi đó sau khi chơi lại game. Để thực hiện được điều đó, người chơi cần gom đủ 10 rubies và trao đổi điều ước với người bán rong (peddler). Hoặc với 5 rubies nhân vật sẽ có thêm 1 mạng.

## Hệ thống đối tượng và Items

### Objects

Aladdin:

Là nhân vật chính trong game, nhiệm vụ thắng hết tất cả màn chơi mà không mất đi hết mạng. Khởi đầu với 10 quả táo và 3 mạng.

Bị tấn công và gây dame bởi các enemies, lửa và va chạm với Jafar. Có tổng cộng 10 máu mỗi mạng. Bị tấn công mỗi lần sẽ mất đi 1 máu.

Ground:

Là mặt đất, có vị trí thấp nhất trên màn chơi, và không có hình ảnh thể hiện.

Platform:

Là bục, có vị trí cố định nhất định mỗi khi load map, là nơi Aladdin có thể đứng và di chuyển trên đó.

Fire:

Là lửa, mỗi khi Aladdin đứng trên đó sẽ bị mất máu.

FloatingGround: 

Là 1 platform, tuy nhiên, khác với những platform bình thường khác, khi Aladdin chạm vào FloatingGround, sau 1 khoảng thời gian ngắn thì FloatingGround sẽ rơi xuống, khi Aladdin đi ra xa 1 khoảng cách nhất định (ngoài vùng nhìn thấy) thì FloatingGround sẽ được phục hồi.

SpringBoard: 

Là 1 vật thể chỉ có trong màn chơi Agrabah Market. Có tác dụng nâng vị trí của Aladdin lên mỗi khi Aladdin chạm vào.

Rope:

Là sợi dây, dùng để Aladdin đu vào leo lên.

Horizontal Bar:

Là thanh gỗ bắt ngang, dùng để Aladdin đu và di chuyển qua lại.

### Enemies

* **Sword Guard:**

+ Location: Agrabah Market (Lv1Scene).

+ Health: 1 Slash, 2 Táo.

+ Là enemy có thân hình to lớn, tay cầm đao, tầm đánh gần, khi phát hiện ra Aladdin thì đi lại và đánh.

* **Staff Guard:** 

+ Location: Agrabah Market (Lv1Scene).

+ Health: 1 Slash, 2 Táo.

+ Là enemy có thân hình gầy và cao, tầm đánh vừa, khi phát hiện ra Aladdin thì đi khá chậm lại và đánh.

* **Knife Guard:** 

+ Location: Agrabah Market (Lv1Scene).

+ Health: 1 Slash, 1 hoặc 2 Táo.

+ Là enemy có thân hình lùn và béo, tầm đánh xa, khi phát hiện ra Aladdin thì đi lại và ném dao.

+ *Mẹo: vì Knife Guard ném dao ở vị trí nhất định, nên có thể né bằng cách nhảy lên. Knife Guard có khoảng delay giữa các lần ném, tranh thủ lúc đó có thể lại gần và chém hoặc đứng từ xa để ném táo.*

* **Knife Juggler:**

+ Location: Agrabah Market (Lv1Scene).

+ Health: 1 Slash, 2 Táo.

+ Là enemy có thân hình gầy và ốm, khá giống Staff Guard, có 2 tầm đánh: trung bình và xa. Khi Aladdin vào tầm đánh xa thì sẽ quăng dao lên không trung và rơi xuống 1 khoảng cách nhất định. Khi Aladdin nằm trong tầm đánh gần thì Knife Juggler sẽ ném dao thẳng về vị trí của Aladdin. Knife Juggler không thể di chuyển, chỉ đứng yên tại vị trí nhất định.

+ *Mẹo: Đứng trong tầm đánh xa và trước khi vào tầm đánh gần, Knife Juggler sẽ ném dao ở tầm xa, do đó nhân vật sẽ không nhận damage, lợi dụng vị trí này để tiêu diệt enemy.*

* **Apple Thief:**

+ Location: Agrabah Market (Lv1Scene).

+ Health: 1 Slash.

+ Là enemy núp trong 1 chiếc thùng gỗ, và di chuyển rón rén lại vị trí của nhân vật. Khi tung ra cú đánh, nhân vật sẽ không bị mất máu mà mất 3 táo mỗi lượt. Chỉ có thể tiêu diệt bằng cách chém.

* **Pot Dropper:**

+ Location: Agrabah Market (Lv1Scene).

+ Health: 1 Slash.

+ Là enemy chỉ đứng trong nhà và thò tay ra ném các chậu xuống mỗi khi Aladdin đi vào tầm. Có thể tiêu diệt bằng cách nhảy lên và chém khi cái tay thò ra, khi tay thụt vào lại thì không thể chém được.

+ *Mẹo: đứng bên trái hoặc phải, canh lúc tay thò ra, nhảy lên và chém.*

### Boss

* **Jafar:**  

+ Location: Jafar’s palace (BossScene)

+ Health: 21 Táo.

+ Đây là boss của game, khi Jafar còn trên 13 máu, Jafar là người, với ma thuật của mình, Jafar tung ra những đòn tấn công nhưng không gây damage, nhằm kéo Aladdin về phía mình. Những đòn tấn công được tung liên tiếp theo đợt. Khi vị trí của Aladdin kế bên Jafar thì Aladdin sẽ bị mất máu theo thời gian. Khi Jafar còn dưới 13 máu, Jafar hóa thành rắn. Khi hóa rắn, Jafar hóa rắn, hắn sẽ tung từng đợt lửa theo thời gian với vận tốc trung bình và gây damage nếu trúng vào Aladdin. Ngoài ra, các bật thềm sẽ tạo ra lửa nếu Aladdin đứng lên đó.

+ *Mẹo: Khi hết táo, hãy chạy về phía còn lại và nhảy lên bục để lấy táo, hãy nhớ rằng, táo mới sẽ chỉ xuất hiện khi nhân vật không còn táo trong người. Lửa khi Jafar hóa rắn tung ra rất đều, hãy bắt nhịp và nhảy để né được lửa đồng thời ném táo về phía hắn. Cố gắng nhặt thật nhanh các quả táo khi hết táo, đừng đứng lâu quá, chết đấy!*

### Items

* **Apples:** 

+ Location: Agrabah Market (Lv1Scene) và Jafar’s palace (BossScene)

**+** Là vũ khí cơ bản trong game, tuy nhiên người chơi chỉ có thể mang theo tối đa 99 quả táo trong người. Nếu hết táo thì không thể ném được nữa.

* **Black Lamp**: 

**+** Location: Agrabah Market (Lv1Scene).

**+** Giống như 1 quả bom, khi Aladdin chạm vào Black Lamp, tất cả enemy trong vùng camera để bị tiêu diệt.

* **Blue Vase:** 
* + Location: Agrabah Market (Lv1Scene).

+ Đây là 1 checkpoint, mỗi khi Aladdin chết, người chơi sẽ bắt đầu tại vị trí Blue Vase nào được kích hoạt cuối cùng.

* **Extra Life:** 
* + Location: Agrabah Market (Lv1Scene).

+ Đây cũng là checkpoint, khi chạm vào sẽ giúp Aladdin có thêm 1 mạng nữa.

* **Rubies:** 

+ Location: Agrabah Market (Lv1Scene).

+ Mỗi khi ăn được 1 Ruby, Aladdin sẽ tích được thêm 1 Ruby. Nếu đủ 5 Rubies, người chơi có thể đổi lấy thêm mạng (Extra Life) với peddler. Hoặc nếu đủ 10 Rubies, người chơi có thể đổi lấy 1 điều ước (Wish).

* **Genie’s Heart: **
* **+** Location: Agrabah Market (Lv1Scene).

+ Dùng để hồi phục lại máu khi Aladdin mất máu, lượng máu hồi phục không bằng nhau đối với mỗi Genie’s Heart.

* **Genie’s Token:**
* + Location: Agrabah Market (Lv1Scene).

+ Khi đi qua Genie’s Token, người chơi sẽ được thưởng thêm điểm.

## Tương tác game

KEY\_LEFT\_ARROW: di chuyển qua trái.

KEY\_RIGHT\_ARROW: di chuyển qua phải.

KEY\_RETURN: lựa chọn màn chơi.

KEY\_A: ném táo (không thể ném khi hết táo).

KEY\_S: chém.

KEY\_D: nhảy.

KEY\_DOWN\_ARROW: ngồi xuống, thay đổi lựa chọn màn chơi.

KEY\_UP\_ARROW: nhìn lên, thay đổi lựa chọn màn chơi.

## Sơ đồ luồng

### Cấu trúc Framework

Framework là 1 thư mục chứa các lớp cần thiết cho việc xây dựng game như:

+ Rect: Mô tả 1 hình chữ nhật.

+ Text: Vẽ chữ.

+ Camera: tạo camera di chuyển theo nhân vật

+ Sound: tạo âm thanh cho game.

+ Graphics: chứa các hàm, thuộc tính hỗ trợ render hình ảnh.

+ GameMap: chứa tất cả các GameObject được tạo từ tilemap.

+ Input: hỗ trợ nhận tính hiệu từ thiết bị ngoại vi để điều khiển.

+ Matrix: ma trận, sử dụng trong các phép vị tự và camera,…

+ Node: đại diện cho 1 vật thể trong game.

+ PhysicsManager: hỗ trợ tương tác vật lý trong game.

+ Quadtree: hỗ trợ quadtree, giảm thiểu chi phí

+ RigidBody: tạo cho vật thể có tính chất vật lý, bao gồm collider

+ Scene: đại diện cho 1 màn chơi.

+ Size: kích cỡ của 1 vật thể, tương tự như Rect.

+ Texture: đại diện cho 1 hình ảnh, dùng để load lên game.

+ Vec2: vector dùng cho việc tính toán trong game.

### GameObject

GameObject là 1 thư mục chứa tất cả các lớp đại diện cho tất cả các vật thể có trong game. GameObject bao gồm:

#### Enemies

Enemies là các đối thủ của Aladdin trong game. Tên loại enemies đã được liệt kê ở mục 3.2. Trong thư mục enemies sẽ bao gồm:

+ BigEnemy (Sword Guard): là con của Enemy, đại diện cho đối thủ của aladdin, cầm đao.

+ FatEnemy (Knife Guard): là con của Enemy, đại hiện cho đối thủ của aladdin, ném dao.

+ HideEnemy (Apple Thief): là con của Enemy, đại hiện cho đối thủ của aladdin, núm trong thùng và trộm 3 táo mỗi lần.

+ Jafar: là con của Enemy, đại hiện cho đối thủ của aladdin, là boss ở màn chơi cuối cùng.

+ KnifeEnemy (Knife Juggler): là con của Enemy, đại hiện cho đối thủ của aladdin, ốm và cao, ném dao, có 2 tầm đánh, xa và gần.

+ ThinEnemy (Staff Guard): là con của Enemy, đại hiện cho đối thủ của aladdin, ốm và cao, cầm gậy.

+ WallEnemy (Pot Dropper): là con của Enemy, đại hiện cho đối thủ của aladdin, chỉ xuất hiện khi Aladdin nằm trong tầm tấn công, thả pot xuống mỗi lượt.

Và 2 lớp Enemy và EnemyState:

+ Enemy: là con của GameObject và là lớp cha của mọi enemies khác, gồm các thuộc tính chính là attackRange (tầm đánh), health (máu), target (mục tiêu), …

+ EnemyState: là lớp cha của mọi State của enemies, gồm các tính chất là: idle (đứng yên), walk (di chuyển lại gần mục tiêu), attack (tấn công),… có các hàm ảo là onUpdate, onRender, onExit, checkTransition,…

* Mỗi enemy sẽ có các state riêng biệt, các state đó sẽ kế thừa từ EnemyState và override lại.

#### Ground

Ground là 1 thư mục chứa tất cả các lớp đại diện cho những vật thể để Aladdin đứng và di chuyển trên đó. Ground bao gồm:

+ FloatGround: là 1 platform mà khi Aladdin chạm lên sẽ rơi xuống sau 1 khoảng thời gian, khi Aladdin di chuyển ra xa 1 khoảng cách nhất định thì FloatGround sẽ được khôi phục lại vị trí cũ.

+ SpringGround: là 1 vật thể, khi Aladdin nhảy lên sẽ nâng vị trí của Aladdin lên.

* Lớp Ground là con của GameObject và là lớp cha của 2 lớp trên, chứa các thuộc tính như RigidBody và Texture.
* Cầu thang (stair) là 1 ground. Có các đặc tính giống ground, chỉ khác về vị trí và kích thước. Vì thế nên không cần thiết tạo ra 1 lớp đại diện cho nó.
* Mỗi loại ground sẽ có những state khác nhau như rơi hay rung,…

#### Items

Items là thư mục chứa các lớp đại diện cho những items trong màn chơi. Items bao gồm:

+ Apple: là những trái táo có sẵn trong màn chơi, giúp người chơi có thêm táo để ném.

+ Camel: là con lạc đà, dùng để nhảy lên, khi đáp xuống sẽ bắn ra đạn.

* Lớp Item là con của GameObject và là cha của các lớp trên. Mỗi item riêng biệt sẽ có những đặc tính riêng, tuy nhiên chúng sẽ có điểm chung là vị trí khởi đầu hay vị trí trong map, trạng thái hiện tại, có va chạm với nhân vật hay không,…
* Lớp ItemState là lớp cha của các State của các items con.
* Mỗi item sẽ có những trạng thái khác nhau như nổ, đứng yên,…

#### Wall

Wall là con của GameObject. Đại diện cho những bức tường, mục đích để chặn các lối đi của Aladdin, hướng nhân vật theo 1 lối đi nhất định theo mong muốn của lập trình viên. Vì là con của GameObject nên wall sẽ có các đặc tính cơ bản của GameObject như độ cứng, vị trí, kích thước,…

#### Weapon

Weapon là thư mục đại diện cho vũ khí tương tác trong game. Weapon bao gồm:

+ AppleToThrow: là táo dùng để Aladdin ném. Bị vỡ khi va chạm với ground, wall, stair,

+ KnifToThrow: là dao mà enemy ném về phía Aladdin.

+ PotToThrow: là chậu mà WallEnemy ném xuống Aladdin.

* Lớp Weapon là con của GameObject và là cha của các lớp trên. Có các thuộc tính như: attackRange (kích thích vùng tấn công), owner (đối tượng nào đã tạo ra loại weapon này), Texture,…

#### Other

Các lớp còn lại là những GameObject riêng lẻ, vì thế không bỏ trong 1 thư mục riêng, thay vào đó gộp chung lại gọi là other, bao gồm:

+ Aladdin: đây là nhân vật chính của chúng ta, có rất nhiều thuộc tính riêng khác với các GameObject khác như: state, eScene (ở màn chơi nào), isCollisionWith…(có đang va chạm với … hay không), health (máu),…

+ BackGroundLv1Scene: đây là con của Node, dùng để đại diện cho map, bao gồm như: hình ảnh mây, hình ảnh mặt đất, nhà phía sau, các cây cột,… ta gọi chung là các hình ảnh tượng trưng mà không thể tương tác được.

+ BackgroundMenu: là con của Node và là ảnh nền của màn hình chọn lựa.

+ BackgroundSky: là con của Node và là hình ảnh mây.

+ GameObject: là con của Node, và là cha của tất cả các vật thể có tương tác trong game. GameObject mang 1 RigidBody để tương tác vật lý.

+ KnifeMenuSelector: là con của Node và là hình ảnh con dao trong màn hình lựa chọn.

+ TitleMenuSelector: là con của Node và là hình ảnh chữ Disney Aladdin trong màn hình lựa chọn game.

### State

State là thư mục chứa tất cả các lớp đại diện cho trạng thái của Aladdin. State bao gồm:

+ State: đây là lớp cha của các lớp State khác. State có thuộc tính là 1 con trỏ node dùng để chứa địa chỉ của aladdin. Ngoài ra, State có các hàm ảo như:

+ onEnter: hàm được gọi lên mỗi khi chuyển State.

+ onUpdate: hàm được gọi lên mỗi khi update State.

+ onExit: hàm được gọi lên trước khi State bị hủy

+ checkTransition: hàm kiểm tra xem State tiếp theo là gì thông qua các giá trị của Input mà quyết định, nếu không nhận được Input thì trả về null.

Những trạng thái dưới đây đều là con của lớp ‘State’:

+ BeingAttacked: Aladdin bị thương tổn, có thể chuyển qua các trạng thái Throw, Slash, Jump, IdleToLook, IdleToSit, Run, Idle.

+ BeingAttackedAndDie: Aladdin bị thương tổn và chết

+ BeingRescued: Aladdin được cứu bởi thần đèn (sau khi chết)

+ Brake: Aladdin phanh

+ Climb: Aladdin leo dây, có thể chuyển qua các trạng thái JumpWhileClimb, SlashWhenClimb, ThrowWhenClimb,Idle.

+ Fall: Aladdin rơi, có thể chuyển qua các trạng thái JumpAndThrow, JumpAndSlash, IdleToClimb, Grounding, Flip.

+ Flip: Aladdin nhào lộn trên spring board, có thể chuyển qua các trạng thái JumpAndThrow, JumpAndSlash, Idle, Flip, Fall.

+ Grounding: Aladdin tiếp đất sau khi thực hiện 1 số State, có thể chuyển qua các trạng thái Idle.

+ HeadUp: Aladdin ngửa mặt lên trời và giữ nguyên, có thể chuyển qua các trạng thái Jump, HeadUpAndSlash, Throw, Idle.

+ HeadUpAndSlash: Aladdin thực hiện liên hoàn chém, có thể chuyển qua các trạng thái Jump, IdleToSit, Throw, Idle.

+ Idle: Aladdin trong trạng thái đứng yên 1, có thể chuyển qua các trạng thái Run, Jump, HeadUp, IdleToSit, Slash, Throw, BeingAttacked, Flip.

+ Idle2: Aladdin trong trạng thái đứng yên 2, tương tự Idle

+ Idle3: Aladdin trong trạng thái đứng yên 3, tương tự Idle

+ IdleToClimb: Aladdin trong trạng thái sẵn sàng leo dây, có thể chuyển qua các trạng thái JumpWhileClimb, SlashWhenClimb, ThrowWhenClimb, Climb, Idle.

+ IdleToLook: Chuyển trạng thái từ đứng yên sang nhìn lên trời, có thể chuyển qua các trạng thái Jump, HeadUpAndSlash, Throw, Idle, HeadUp.

+ IdleToSit: Chuyển từ trang thái từ đứng yên sang ngồi, có thể chuyển qua các trạng thái Jump, SitAndSlash, SitAndThrow, Sit, Idle.

+ IdleWhenWing: Trạng thái đứng yên trong lúc đu xà, có thể chuyển qua các trạng thái Wing, SlashWhenWing, ThrowWhenWing, Fall.

+ Jump: Nhảy, có thể chuyển qua các trạng thái JumpAndThrow, JumpAndSlash, Idle, IdleToClimb, Jump, Flip, IdleWhenWing.

+ JumpAndSlash: Nhảy và chém, có thể chuyển qua các trạng thái JumpAndThrow, Idle, IdleToClimb, Jump, Flip.

+ JumpAndThrow: Nhảy và ném táo, có thể chuyển qua các trạng thái Idle, IdleToClimb, Flip.

+ JumpWhileClimb: Nhảy trên dây, có thể chuyển qua các trạng thái IdleToClimb, Idle, Fall, IdleWhenWing.

+ Push: Đẩy, có thể chuyển qua các trạng thái Jump, Idle.

+ Random: Tạo ra các số ngẫu nhiên cho trạng thái Idle

+ Run: Chạy, có thể chuyển qua các trạng thái Idle, IdleToSit, HeadUp, RunAndSlash, RunAndJump, Push, IdleToClimb, Jump, Flip, Brake, IdleWhenWing.

+ RunAndJump: Chạy và nhảy, có thể chuyển qua các trạng thái JumpAndThrow, JumpAndSlash, Idle, Run, IdleToClimb, Jump, Flip.

+ RunAndSlash: Chạy và chém, có thể chuyển qua các trạng thái RunAndJump, HeadUp, Sit, Idle, Run.

+ Sit: Giữ nguyên trạng thái ngồi, có thể chuyển qua các trạng thái Idle, SitAndSlash, SitAndThrow.

+ SitAndSlash: Ngồi và chém, có thể chuyển qua các trạng thái SitAndThrow, Jump, Idle, Sit.

+ SitAndThrow: Ngồi và ném, có thể chuyển qua các trạng thái SitAndSlash, Jump, Idle, Sit.

+ Slash: Chém, có thể chuyển qua các trạng thái Throw, HeadUp, IdleToSit, Idle.

+ SlashWhenClimb: Chém trong lúc leo dây, có thể chuyển qua các trạng thái JumpWhileClimb, ThrowWhenClimb, IdleToClimb.

+ SlashWhenWing: Chém trong lúc đu xà, có thể chuyển qua các trạng thái IdleWhenWing, Fall.

+ Throw: Ném, có thể chuyển qua các trạng thái Idle.

+ ThrowWhenClimb: Ném trong lúc leo dây, có thể chuyển qua các trạng thái JumpWhileClimb, SlashWhenClimb, IdleToClimb.

+ ThrowWhenWing: Ném trong lúc đu xà, có thể chuyển qua các trạng thái Fall, IdleWhenWing.

+ Wing: Đu xà, có thể chuyển qua các trạng thái Idle.

### Scene

Có 3 màn hình (scene). Trong đó có 2 màn chơi và 1 màn hình chọn lựa:

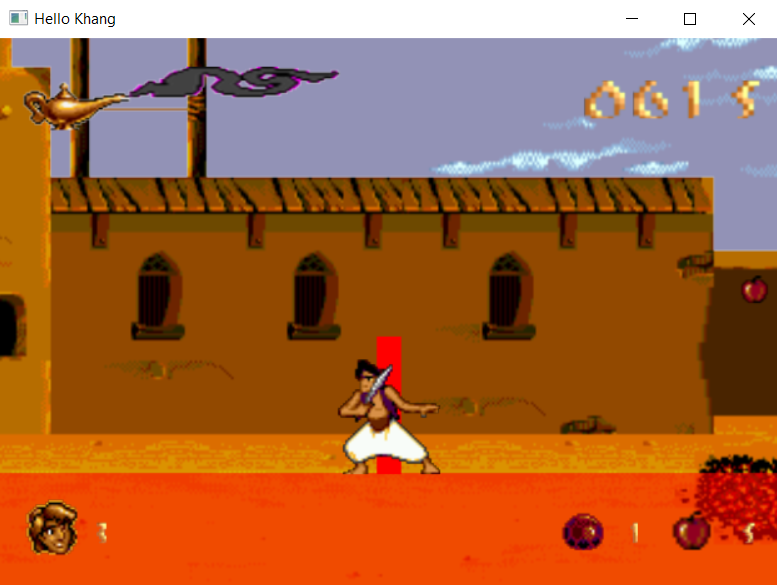
+ MenuSelectorScene: là con của Scene và là màn hình chọn lựa màn chơi cũng như bật tắt âm thanh.

+ Lv1Scene: là con của Scene và là màn chơi Agrabah Market. Trong đây bao gồm các Node như các background và các GameObject như Aladdin, táo, các Enemies, dây, tường, platform, các ground,…

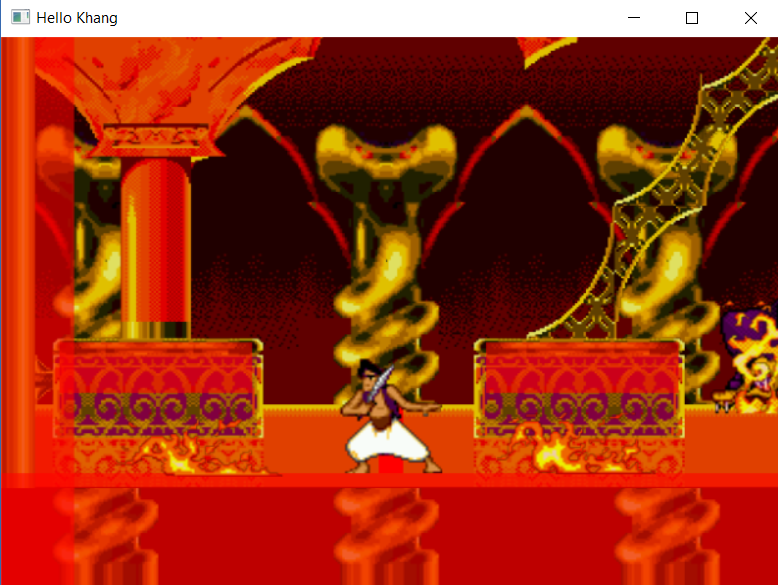
+ BossScene: là con của Scene và là mà chơi Jafar’s Palace. Scene bao gồm: Aladdin, tường, ground, Jafar, lửa, platform,…

## Các màn chơi

### Agrabah Market (Lv1Scene):



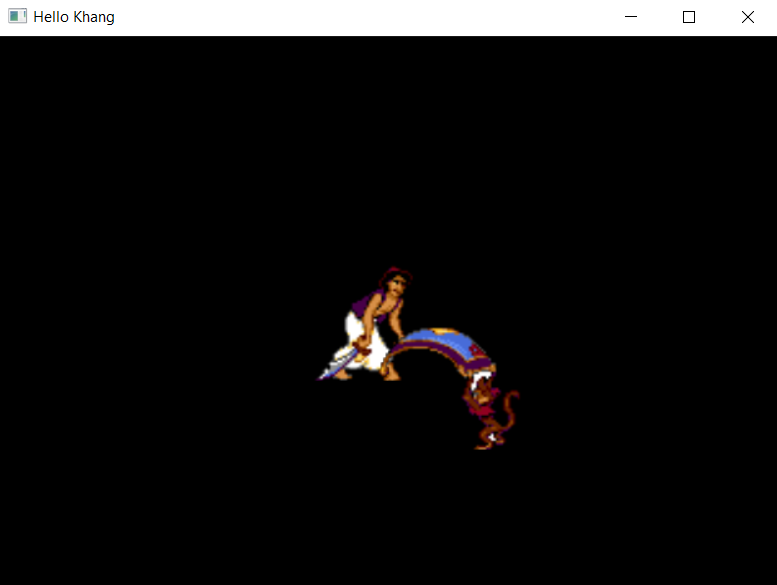
### Jafar’s palace (BossScene):



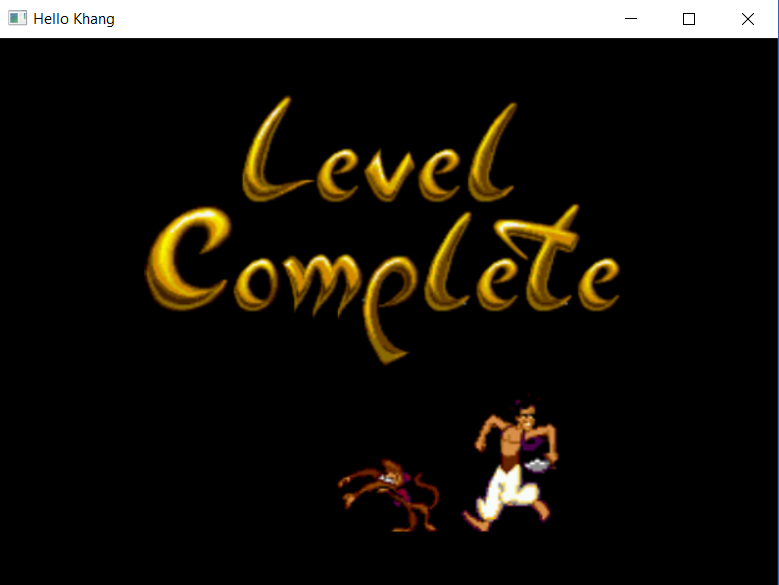
### MenuSelector:



### DeathScene:



### CompleteScene:





# **III – Cấu trúc game, các kỹ thuật và áp dụng**

## Các kỹ thuật áp dụng trong game

### Thuật toán xử lý va chạm AABB

AABB là hộp bao có dạng hình chữ nhật có các cạnh song song với các trục tương ứng và bao lấy vật thể.

* Phát hiện va chạm AABB:
* Ta giả sử mỗi hộp bao có các giá trị sau tương ứng: top, bottom, left, right.
* Phát hiện tọa độ phía trên (top), phía dưới (bottom), bên trái (left), bên phải (right) của mỗi hộp bao.
* Nếu top1 <= bottom2 và bottom1 >= top2 và left1 <= right2 và right1 => left2 thì có va chạm xảy ra.
* Nếu không xảy ra trường hợp trên thì không có va chạm xảy ra.

### Thuât toán xử lý va chạm SweptAABB

* Đa số các lập trình viên đều sử dụng phương pháp tính bao hình (AABB) là phương án cho bài toán va chạm, tuy nhiên phương pháp này khi nâng cấp lên các thuật toán nâng cao hơn như SAT và GJK lại gặp rất nhiều khó khăn. Swept AABB là một giải thuật trung gián, nó ứng dụng tư tưởng của giải thuật bao AABB thông thường nhưng là chuyển tiếp đến các giải thuật nâng cao khác, giúp lập trình viên dễ dàng tiếp cận và sử dụng.
* Phương thức kiểm tra SweptAABB được mô tả như sau:



Trong đó:

* RigidBody a: là đối tượng di chuyển.
* RigidBody b: là đối tượng tĩnh xét va chạm.
* Manifold là 1 struct chứa hướng va chạm và độ sâu của 2 vật trên.
* Giá trị trả về của phương thức sweptAABB nằm trong khoảng 0 đến 1. Nếu là 0 thì vật đang va chạm, nếu là 1 thì ở frame sau sẽ không va chạm, còn trong khoảng từ 0 đến 1 thì khoảng cách giữa 2 vật sẽ bằng giá trị trả về của hàm swept nhân với vận tốc vật đang di chuyển.

- Các bước cài đặt thuật toán:

+ Tính toán những khoảng cách, thời gian cần thiết trên mỗi trục x, y để xác định thời điểm va chạm.

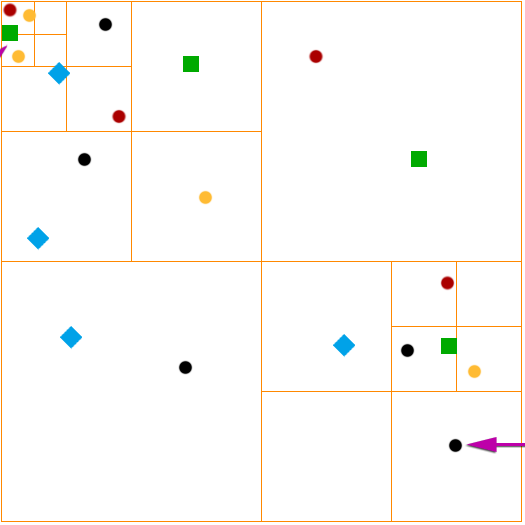
+ Căn cứ vào vận tốc của đối tượng và các chỉ số khoảng cách để tính toán khả năng va chạm của 2 đối tượng.

+ Nếu va chạm được xác nhận diễn ra, vận tốc sẽ được biến đổi để thể hiện được phản ứng của đối tượng với va chạm.

### Transformation

Transformation là phương pháp biến đổi hình học dùng để chuyển đổi các giá trị tòa độ của các đỉnh của một hình thành một tập hợp các giá trị tòa độ khác và hình gốc ban đầu có thể có kích thước khác ,ở một vị trí khác và một góc xoay khác so với tòa độ gốc ban đầu

### Thuật toán phân hoạch không gian QuadTree

* Xét va chạm (Collision Detection) là một việc quan trọng trong lập trình game. Công việc này đòi hỏi chi phí cao, đặc biệt khi số lượng thực thể (entity) trong game là rất lớn. Để đảm bảo độ chính xác của game, ta phải kiểm tra từng đôi một các entity với nhau. Làm một phép thử, nếu game của chúng ta có 100 entity riêng biệt, ta sẽ mất một lượng chi phí không nhỏ cho **C2100** lần kiểm tra. Con số này sẽ ngày càng khủng khiếp khi số lượng entity trong game ngày một tăng nhiều. Vì vậy có một giải thuật nào có thể giúp giảm thiểu chi phí cho việc xét va chạm giữa các entity trong game là giải thuật Quadtree.
* Áp dụng kĩ thuật Quadtree để tăng hiệu suất game và giảm số lần xét va chạm trong game.
* Kĩ thuật này chi màn hình thành những hình chữ nhật nhỏ. Ban đầu chia màn hình làm 4 phần bằng nhau, mỗi phần là 1 Node và cứ 1 Node như vậy lại có 4 Node con.

# **IV – Kết luận**

## Đánh giá chung

Game được hoàn thành 90% theo với mục tiêu ban đầu team đặt ra. Chỉ còn thiếu sót những chức năng nhỏ và kĩ thuật khó.

Tuy chưa được thật sự hài lòng nhưng chúng em cũng đã cố gắng hết sức. Qua môn học này, chúng em đã học được cấu trúc của 1 game, cách thiết kế, vận hành và sửa chữa các lỗi trong game.

## Bảng phân công công việc cụ thể

|  |  |  |  |
| --- | --- | --- | --- |
| Tên công việc | Khang | Long | Lâm |
| Framework (basic) | X |  |  |
| QuadTree |  | X |  |
| Physics | X |  |  |
| Aladdin’s States | X | X | X |
| Read map |  | X |  |
| Aladdin | X | X |  |
| Audio |  |  | X |
| Input | X |  |  |
| Graphics | X |  |  |
| Items |  | X |  |
| Camera | X |  |  |
| Enemies |  | X |  |
| Scenes | X | X |  |
| Viết báo cáo | X |  | X |

## Áp dụng kiến thức đã học

* QuadTree
* SweptAABB
* AABB
* Camera
* Matrix
* Graphics ( các directX )
* Texture
* Sprite
* Write – Read Map
* Read file xml
* Design Pattern (Singleton, Abstract Factory, State,…)
* Input – output

## Phương hướng phát triển

+ Hoàn thiện game.

+ Xử lý các bug còn tồn tại.

+ Tạo hiệu ứng chuyển Scene.

+ Thêm intro vào phần đầu.

+ Thêm thay đổi độ khó game.

+ Quản lý bộ nhớ.

+ Thêm nhiều màn chơi, thêm các enemies và objects.

## Các tài liệu tham khảo

<http://www.crashnbuild.com/>

<https://www.gamefaqs.com/genesis/586017-disneys-aladdin/faqs/28471>

<http://www.stdio.vn/articles/read/109/hien-thuc-quadtree-va-ung-dung-trong-lap-trinh-game>

<http://www.cplusplus.com/>

htttp://www.mapeditor.org/

Beginning Game Programming – Jonathan S. Harbour