**ĐẠI HỌC QUỐC GIA TP. HCM**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

---------------🙦 🕮 🙤---------------

**BÁO CÁO ĐỒ ÁN**

**NHẬP MÔN PHÁT TRIỂN GAME**

**LỚP:** **SE102.K11**

**ĐỀ Tài: ALADDIN GENESIS**

A picture containing clipart

Description automatically generated

**GIẢNG VIÊN HƯỚNG DẪN: ThS. Nguyễn Vĩnh Kha**

**NHÓM THỰC HIỆN:**

Nguyễn Lê Việt Hoàng - 17520513

Huỳnh Quốc Trung - 17520184

Nguyễn Mạnh Tùng - 17521236

TPHCM, Ngày 22 tháng 12 năm 2019

# LỜI CẢM ƠN

**Đầu tiên, em xin gởi lời cảm ơn chân thành đến tập thể quý Thầy Cô Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM và quý Thầy Cô khoa Nhập môn Công Nghệ Phần Mềm đã giúp cho nhóm chúng em có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này.**

**Đặc biệt, em xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới Thầy Nguyễn Vĩnh Kha. Đã giúp em hoàn thành tốt báo cáo môn học của mình.**

**Trong thời gian một học kỳ thực hiện đề tài, em đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, em vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, em không tránh khỏi những thiếu sót. Chính vì vậy, em rất mong nhận được những sự góp ý từ phía các Thầy Cô nhằm hoàn thiện những kiến thức mà nhóm chúng em đã học tập và là hành trang để nhóm chúng em thực hiện tiếp các đề tài khác trong tương lai.**

**Xin chân thành cảm ơn các quý Thầy Cô!**

# MỤC LỤC

[Phần 1: Giới thiệu sơ lược về game: 1](#_Toc11443854)

[**1.1 Giới thiệu game Aladdin Genesis** 1](#_Toc11443855)

[**1.2 Giới thiệu các scene trong đồ án** 2](#_Toc11443858)

[**1.2.1 Bản đồ** 2](#_Toc11443859)

[**1.2.1.1 Scene Sultan Dungeon:** 2](#_Toc11443860)

[**1.2.1.2 Scene Jafar’s Palace:** 2](#_Toc11443861)

[**1.2.2 Hệ thống Objects:** 3](#_Toc11443863)

[**1.2.2.1 Player:** 3](#_Toc11443864)

[**1.2.2.2 Enemys:** 3](#_Toc11443865)

[**1.2.2.3 Items:** 4](#_Toc11443865)

[**1.2.2.3 Obstacles:** 5](#_Toc11443865)

[**1.3 Hướng dẫn người chơi** 6](#_Toc11443866)

[Phần 2: FrameWork 7](#_Toc11443867)

[**2.1 Keyboard** 7](#_Toc11443868)

[**2.2 Graphics** 7](#_Toc11443869)

[**2.3 Sprites** 8](#_Toc11443870)

[**2.4 Textures** 8](#_Toc11443871)

[**2.5 Animation** 8](#_Toc11443871)

[**2.6 Sound** 9](#_Toc11443871)

[**2.7 Một số hàm hỗ trợ** 9](#_Toc11443871)

[Phần 3: Các kỹ thuật áp dụng trong game: 10](#_Toc11443867)

[**3.1 State Parterns**  10](#_Toc11443868)

[**3.2 Spatial Partition (Grid)** 10](#_Toc11443869)

[**3.3 Object Pooling** 11](#_Toc11443870)

[**3.4 SweptAABB** 11](#_Toc11443871)

[**3.5 Singleton** 11](#_Toc11443871)

[Phần 4: Kết luận: 12](#_Toc11443872)

[**4.1 Bảng phân công công việc trong nhóm** 9](#_Toc11443873)

[**4.2 Đánh giá** 13](#_Toc11443874)

[**4.3 Tổng kết** 13](#_Toc11443875)

[**4.4 Tài liệu tham khảo** 13](#_Toc11443876)

# 

# Phần 1: GIỚI THIỆU SƠ LƯỢC VỀ GAME

## **Giới thiệu game Aladdin Genesis**

A close up of text on a black background

Description automatically generated

Disney's Aladdin, tựa game này được thực hiện lần đầu vào năm 1993 do đội ngũ [Virgin Games USA](https://infogame.vn/Virgin-Games-USA-ptag.html) phát triển dành riêng cho các hệ máy Game Boy, NES,... Tuy có đến 4 phiên bản khác nhau bao gồm nhiều nhà phát triển cùng thực hiện, song phiên bản được nhiều game thủ Việt biết đến và đón tiếp nhận nồng hậu nhất vẫn chính là Disney's Aladdin do [Capcom](https://infogame.vn/Capcom-ptag.html) mang tới vào năm 1993.

Sở hữu một phong cách chơi hết sức đơn giản dựa theo bộ phim cùng tên, bạn chỉ cần điều khiển nhân vật Aladin đi phiêu lưu, giải đố, tiêu diệt những tên lính và cứu công chúa. Nghe thì có vẻ dễ dàng nhưng hành trình của anh vẫn luôn đầy chông gai, qua những màn chơi mới, bạn buộc phải vận dụng đầu óc nhiều hơn để vượt qua những cạm bẫy khó nhằn trên đường đi.

Game có thời lượng chơi khá dài, nhưng lại thay đổi tình tiết liên tục, các màn chơi không trùng lặp nhiều lắm tạo nên một tựa game cực xuất sắc

## **Giới thiệu các scene trong đồ án:**

### **Bản đồ**

#### **Scene Sultan Dungeon:**

A sign on the side of a building

Description automatically generated

Sultan Dungeon

#### **Scene Jafar’s Palace:**

A picture containing yellow, indoor

Description automatically generatedJafar’s Palace

### **Hệ thống Objects:**

#### **Player**



**Aladdin** (9 máu/mạng, Số mạng: 2)

Nhân vật mà người chơi điều khiển, suốt cả màn chơi, nhân vật này sẽ sử dụng vũ khí là thanh kiếm và những quả táo cùng một số kỹ năng chiến đấu đặt biệt, giúp người chơi tiêu diệt và vượt qua những thử thách trong màn chơi.

Hệ thống state trong game sử dụng state pattern để xử lý qua lại giữa các state. Chi tiết xem ở code

#### **Enemys**

|  |  |  |  |
| --- | --- | --- | --- |
| ENEMY | INFORMATION | AI | WEAPON |
| BAT | DAMAGE: 1  HP: 1  Points: 200 | Bắt đầu bay theo quỹ đạo khi Player lọt vào tầm tấn công, hướng bay phụ thuộc vào vị trí so với Player |  |
| FAT GUARD  A close up of a logo  Description automatically generated | DAMAGE: 1  HP: 2  Points: 0 | Di chuyển về hướng player khi vào tầm, khi di chuyển đủ gần chuyển sang tấn công, và khiêu khích khi player ở vị trí cao hơn |  |
| THIN GUARD | DAMAGE: 1  HP: 2  Points: 0 | Di chuyển về hướng player khi vào tầm quan sát, khi di chuyển đủ gần chuyển sang tấn công |  |
| SKELETON | DAMAGE: 1  HP: 1  Points: 0 | Khi player đủ gần, sẽ đứng lên và tự phát nổ bản thân bay xương gây sát thương cho player | DAMAGE:1 |
| JAFAR | DAMAGE: 1  HP: 10  Points: 0  BOSS | Dùng vũ khí để kéo player lại, khi va chạm sẽ gây sát thương cho player. Player chỉ có thể tấn công bằng cách ném táo. | DAMAGE: 0 |
| SNAKE | DAMAGE: 1  hp: 20  Points: 0  Hình dạng thứ  2 của BOSS | Liên tục phun lửa gây sát thương cho player.  Player chỉ có thể tấn công bằng cách ném táo. | DAMAGE:1  A close up of a logo  Description automatically generated |

#### **Items**

|  |  |
| --- | --- |
| ITEM | INFORMATION |
| APPLE | APPLES + 1 |
| BLUE HEART | HP + 1  Scores + 150 |
| GEM | GEMS + 1  Scores + 150 |
| GENIE TOKEN | Scores + 250 |

#### **Obstacles**

|  |  |
| --- | --- |
| Obstacles | INFORMATION |
| BALL | damage: 1  Có một trật tự nhất định, gây sát thương khi va chạm với player. |
| SPIKE | damage: 1  Có một trật tự nhất định, gây sát thương khi va chạm với player. |
| STONE | damage: 0  Có một trật tự nhất định, giúp Player có thể di chuyển lên địa hình cao hơn. |
| BLUE VASE | damage: 0  Lưu lại vị trí, khi player hết máu, nếu vẫn còn mạng thì sẽ được hồi sinh tại điểm này. |
| EXIT PORT | damage: 0  Vị trí kết thúc Scene Sultan Dungeon. Chuyển sang Scene tiếp theo. |
| PEDDLE | damage: 0  Người bán hàng rong, khi đủ số lượng Gems, Player có thể tới đây để mua thêm mạng. |

#### **Explosive**

|  |  |
| --- | --- |
| EXPLOSIVE | INFORMATION |
|  | Hiệu ứng nổ của item Apple, Gems, BlueHeart |
|  | Hiệu ứng nổ của item GenieToken và của Jafar khi bị thương. |
|  | Hiệu ứng nổ của Enemys. |
|  | Hiệu ứng nổ của Skeleton Weapon. |
|  | Hiệu ứng nổ của Boss Jafar. |
|  | Hiệu ứng nổ của Apples Weapon. |

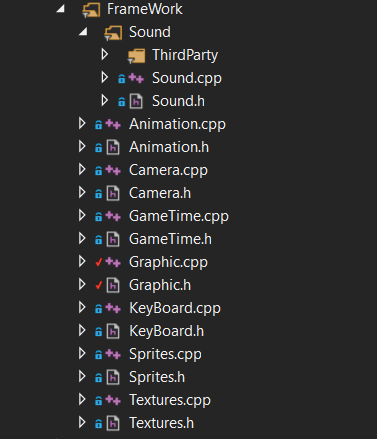
## **Hướng dẫn người chơi**

Người chơi sẽ hóa thân thành nhân vật chính là **Aladdin**, sử dụng các phím trên bàn phím để điều khiển nhân vật:

1. Nhấn ←: di chuyển qua trái;
2. Nhấn →: di chuyển qua phải;
3. Nhấn ↓: ngồi xuống;
4. Nhấn ↑: nhìn lên;
5. Nhấn D: nhảy lên;
6. Nhấn S: chém;
7. Nhấn A: Ném táo

# FRAMEWORK

* Framework thực sự directX đã hỗ trợ rất nhiều, người lập trình không phải làm gì nhiều ngoài việc khởi tạo và xây dựng hàm hỗ trợ
* Việc quan trọng là tương tác giữa các nội dung của framework



## **Keyboard (SINGLETON): Xử lý bàn phím**

* Sử dụng DirectInput8
* 3 hàm cơ bản:
  + GetKeyDown(KeyCode);
  + GetKeyUp(KeyCode);
  + GetKey(KeyCode);
* Trong đó GetKeyDown xử lý sự kiện phím nhấn xuống trong frame đó, GetKeyUp xử lý sự kiện thả phím trong frame đó. GetKey xử lý trạng thái của phím

## **Graphics (SINGLETON): Xử lý giao diện**

* Sử dụng Direct3Ddevice
* Graphic hỗ trợ vẽ chữ số để tạo bảng điểm

void DrawString(const std::string& text, D3DXVECTOR3 pos, int font);

* Graphic còn hỗ trợ vẽ texture lên tại một vị trí cố định trên màn hình (Screen coordinate)

void Draw(float x, float y, LPDIRECT3DTEXTURE9 texture);

## **Sprites: Xử lý hình ảnh**

* Sử dụng LPDIRECT3DTEXTURE9 để lưu trữ Texture kết hợp với một RECT (hoặc BoxCollider là bản float của RECT) để cắt thành sprite nhỏ
* Hỗ trợ vẽ ngay trong sprite bằng hàm

void Draw(D3DXVECTOR3 position = D3DXVECTOR3(), BoxCollider r = BoxCollider(), D3DXCOLOR colorKey = D3DCOLOR\_XRGB(255, 255, 255), bool reverse = false);

* Trong đó:
* *position: vị trí*
* *r: hình chữ nhật để crop hình trong texture (thường không dùng và được khởi tạo ngay khi tạo sprite)*
* *colorKey: pha màu*
* *reverse: đổi chiều*

## **Textures (SINGLETON): Kho hình ảnh**

* Kho lưu trữ Textures để tái sử dụng các texture, tránh trường hợp một texture được tải lên nhiều lần
* Hoạt động: gồm 2 trạng thái thêm vào và lấy ra

void Add(int id, LPCSTR path, D3DCOLOR transparentColor);

LPDIRECT3DTEXTURE9 Get(unsigned int i);

## **Animation: Mô phỏng chuyển động**

* Hoạt động mô phỏng đơn giản theo animation của Engine game Unity3D
* Có thêm các AnimationFrame (KeyFrame trong Unity3D). Vì các chuyển động theo vị trí đều dựa trên các sự kiện trong game, còn animation sẽ giữ nguyên vị trí nên AnimationFrame chỉ có mỗi sprite, nên sẽ cho AnimationFrame kế thừa sprite và có thêm một thuộc tính để thực hiện chuyển động
* Animation có thêm một số hàm hỗ trợ khác

## **Sound (SINGLETON): Xử lý âm thanh**

Sử dụng:

LPDIRECTSOUND8

LPDIRECTSOUNDBUFFER

WAVE file

*Cách sử dụng: trong giáo trình Nhập môn Phát triển Game của trường Đại học Công nghệ Thông tin – Đại học Quốc gia TP.HCM*

## **Một số hàm hỗ trợ**

Debug.h Debug.cpp

* Thực ra trong debugapi có hỗ trợ hàm giúp debug như OutputDebugString, nhưng debug viết ra theo dạng custom để dễ theo dõi các biến trong từng frame hơn

*Nguồn tham khảo:*

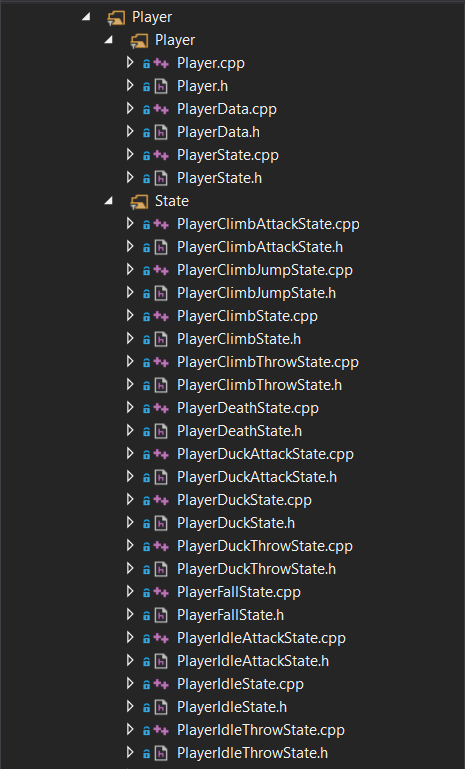
<https://github.com/dungdna2000/gamedev-intro-tutorials>

* Support: Một số hàm để hỗ trợ xử lý logic game, vẽ rect của các objects để xét va chạm

# CÁC KỸ THUẬT ÁP DỤNG TRONG GAME

## **State Parterns**

* Chia nhỏ các hoạt động của nhân vật, vật thể (trong game là Entity) ra thành các state
* Mục đích: dễ dàng quản lý, xử lý cho mỗi trạng thái.
* Ví dụ dưới cho player, tương tự enemy hoặc weapon có trạng thái phức tạp cũng sẽ sử dụng pattern này



*Nguồn tham khảo:*

<https://loctho95.blogspot.com/2017/01/chuong-ix-state-machine-player-bot.html>

## **Spatial Partition (Grid)**

* Chia nhỏ world (game scene) thành từng khối nhỏ, cũng có thể xem là matrix và đặt các object vào trong đó theo vị trí, mỗi ô gọi là **Unit**
* Mục đích cơ bản: tránh việc so sánh collide quá nhiều object trong một frames cũng giống như Quadtree, tất nhiên việc sử dụng grid sẽ đơn giản hơn Quadtree

*Nguồn tham khảo:*

<http://gameprogrammingpatterns.com/spatial-partition.html>

## **Object Pooling (Object pool partern)**

* Đặt các vật thể vào một “pool”, nếu muốn sử dụng chỉ cần lấy ra, không sử dụng thì ném vào lại “pool”.
* Mục đích: tái sử dụng

*Nguồn tham khảo:*

<https://learn.unity.com/tutorial/object-pooling>

## **Tag & Type**

* Mỗi vật thể trong thế giới game đều có một tag riêng và một layer chúng thuộc về
* Trong DirectX chúng ta sẽ đơn giản hóa việc sử dụng Tag và Layer so với Engine Game Unity3D
* Mục đích: Tag: xử lý trường hợp đặc biệt, Type: xét va chạm

*Nguồn tham khảo:*

[https://docs.unity3d.com/Manual//class-TagManager.html](https://docs.unity3d.com/Manual/class-TagManager.html)

## **SweptAABB**

* Xử lý va chạm chính xác
* Mục đích: xét trong frame đó có khả năng va chạm không, tránh việc xử lý va chạm theo truyền thống xuất hiện case lỗi

*Nguồn tham khảo: giáo trình Nhập môn Phát triển Game – Đại học Công nghệ Thông tin – Đại học Quốc gia TP.HCM*

## **Singleton**

* Singleton: mỗi class chỉ có duy nhất một đối tượng được tạo ra
* Mục đích: dễ dàng sử dụng, quản lý

## **TileMap – TileSet**

Dựa vào ma trận Tiles-map và Tileset, trong Game ta có thể linh hoạt trong việc load Map và Object. Những tài nguyên nào thực sự cần thiết thì ta mới vẽ lên, và cái nào không cần thì ta không vẽ. Như vậy ta có thể giảm được số lượng dòng lệnh khi vẽ map và làm cho game ta mượt mà hơn, ít tốn bộ nhớ hơn.

# KẾT LUẬN

## **Bảng phân công công việc trong nhóm**

Trong quá trình làm việc nhóm, có 1 thành viên tách nhóm, việc này ảnh hưởng trực tiếp tới sự cân bằng công việc, cũng như tinh thần của các thành viên còn lại rất nhiều. Dẫu vậy chúng em cũng đã cố gắng hoàn thành phần công việc còn lại, để cho ra một sản phẩm đồ án khá hoàn chỉnh như ngày hôm nay. Sau đây là bảng phân công công việc của các thành viên trong nhóm:

|  |  |
| --- | --- |
| **Thành viên** | **Nhiệm vụ** |
| **1. Nguyễn Lê Việt Hoàng** | - Quản lý về âm thanh trong game.  - Xác định và xử lý va chạm giữa các vật thể với nhau.  - Tạo và quản lý cách di chuyển của quái vật trong game, các animation của enemys trong game.  - Tạo và quản lý các state của Enemys, Weapons (thêm và huỷ).  - Quản lý quá trình render, update các vật thể trong game.  - Giải quyết các ngoại lệ, bug liên quan, …  - Tổng hợp và viết báo cáo. |
| **2. Nguyễn Mạnh Tùng** | - Quản lý về Đồ hoạ.  - Tạo và quản lý các state của Player.  - Tạo và quản lý cách di chuyển của Player.  - Quản lý màn chơi, chuyển màn.  - Xác định và xử lý va chạm giữa các vật thể với nhau.  - Sưu tầm và thiết kế lại các resource hình ảnh, quản lý lưu trữ hình ảnh.  - Giải quyết các ngoại lệ, bug liên quan, … |
| **3. Huỳnh Quốc Trung** | - Tạo và quản lý các tilemap, tạo dữ liệu để load map lên bản đồ cùng với item và quái.  - Tạo và quản lý hiệu ứng, hình ảnh các items, obstacles trong game.  - Nghiên cứu trải nghiệm game, đo các thông số thích hợp.  - Quản lý các thông tin như máu, điểm, thông tin về các Objects.  - Tổng hợp và viết báo cáo. |

## **Đánh giá**

Đánh giá mức độ hoàn thiện Game:

* Hoàn thành khoảng 95% game gốc (tính riêng 2 scene Sultan Dungeon, Jafar’s Palace)

Một số kiến thức quan trọng được học và áp dụng:

* Grid
* Boundary collision detection
* Viewport
* Sprite
* World map
* Tile map…

## **Tổng kết**

* Bằng sự giúp đở của thầy cùng anh, chị và các bạn. Qua đó nhóm đã biết cách làm ra một game là như thế nào và tự làm thành một game hoàn chỉnh.
* Cũng như những nỗ lực không ngừng của các thành viên trong nhóm đã có thể ứng dụng các kỹ thuật trong game như: TileMap, Viewport, Camera, Sprite, SweptAABB cũng như phân hoạch không gian dưới dạng Grid.
* Các kỹ thuật này đều được nhóm chúng em áp dụng vào game như đã được trình bày ở phần trên. Từ đó tiếp thu được các kiến thức nền tảng, là bước đệm cho những sinh viên theo học chuyên ngành Môi trường ảo và phát triển game sau này.
* Nhóm chúng em sẽ cố gắng hoàn thiện Game hơn.

## **Tài liệu tham khảo**

* + - <https://loctho95.blogspot.com>;
    - <https://github.com/dungdna2000/gamedev-intro-tutorials>
    - https://gameprogrammingpatterns.com/spatial-partition;
    - <https://www.spriters-resource.com/genesis_32x_scd/aladdin/>
    - Giáo trình Nhập Môn Phát Triển Game do PSG.TS. Vũ Thanh Nguyên và ThS. Đinh Nguyễn Anh Dũng biên soạn
    - Tài liệu từ các bạn và anh chị về DirectX, Lập trình Game chia sẻ lên nhóm của lớp trong quá trình học.