**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 Ninja Gaiden NES bằng Directx9**

*Giảng viên hướng dẫn:*

ThS. Nguyễn Vĩnh Kha

*Nhóm thực hiện:*

Đặng Thành Quyên - 17520963

Lê Quốc Thắng - 17520156

TPHCM, Ngày 14 tháng 6 năm 2019

Mục lục

[I. **GIỚI THIỆU VỀ GAME** 4](#_Toc11262477)

[1. Gameplay và các màn chơi 4](#_Toc11262478)

[2. Nhân vật 4](#_Toc11262479)

[3. Hệ thống enemies, items, weapon 5](#_Toc11262480)

[**II.** **CÁC KĨ THUẬT SỬ DỤNG TRONG VIỆC PHÁT TRIỂN GAME** 7](#_Toc11262481)

[1. Thư viện DirectX 7](#_Toc11262482)

[2. Kĩ thuật Design Pattern 7](#_Toc11262483)

[3. Kĩ thuật sử dụng Tiled map 8](#_Toc11262484)

[4. Grid – thuật toán phân hoạch không gian 8](#_Toc11262485)

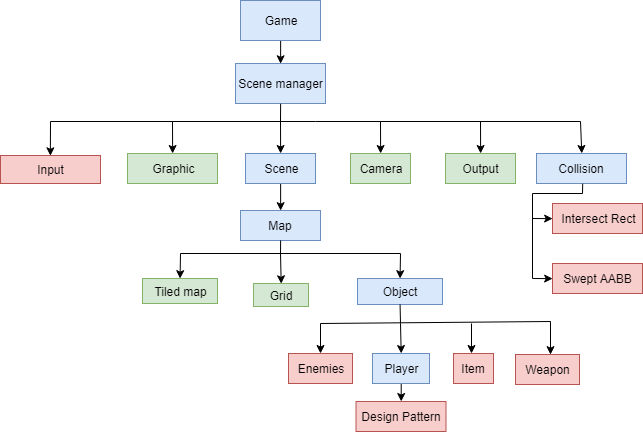
[5. Các kĩ thuật xử lý va chạm 8](#_Toc11262486)

[**III.** **KẾT LUẬN** 9](#_Toc11262487)

[1. Đánh giá mức độ hoàn thiện 9](#_Toc11262488)

[2. Hướng phát triển 9](#_Toc11262489)

[3. Tài liệu tham khảo 9](#_Toc11262490)

*Tổng quan kiến trúc game*

*Bảng phân công*

|  |  |
| --- | --- |
| Họ và tên | Nhiệm vụ chính |
| Đặng Thành Quyên | Game input, Collision detection, Enemies, Items, Player Design Pattern, Weapon |
| Lê Quốc Thắng | Game art, Graphic, Player’s base functions, Game map, Grid algorithm, Game output |

# **GIỚI THIỆU VỀ GAME**

## **Gameplay và các màn chơi**

* Ninja Gaiden là series game acarde được phát hành lần đầu vào 9/12/1988 bởi các nhà phát triển Tecmo, Hudson soft.
* Gameplay cơ bản của game là đi cảnh – cuộn màn hình. Việc cần làm chủ yếu của người chơi là di chuyển trên bản đồ, sử dụng vũ khí tấn công quái vật, nhặt item, nâng cấp vũ khí và qua màn.
* Game có 6 “Act” - tức màn chơi lớn (hay level), tổng cộng khoảng 20 màn chơi. Vì yêu cầu và giới hạn của môn học, chúng em chỉ xây dựng 3 màn chơi là Act III-1, Act III-2, Act III-3.

## **Nhân vật**

* Nhân vật chính là Ninja, tên gốc trong game là Tyu Hayabusa, với vũ khí chính là thanh kiếm Katana với khả năng tấn công không có giới hạn. Bên cạnh đó, bản thân anh có 2 lần hồi sinh, 16 điểm sinh mệnh và năng lượng để sử dụng các chiêu thức khác nhau.
* Các trạng thái chính của Ninja: đứng yên, đứng – tấn công (X), khuỵu (DOWN), khuỵu – tấn công (DOWN + X), nhảy (SPACE), bám tường (SPACE + LEFT, SPACE + RIGHT), leo tường (UP – DOWN), chạy qua phải (RIGHT), chạy qua trái (LEFT). Khi đang bám tường hoặc leo tường, Ninja không thể tấn công và chỉ có thể rời khỏi tường bằng cách nhảy với 1 hướng bất kì (SPACE + LEFT, SPACE + RIGHT).

## **Hệ thống enemies, items, weapons**

1. **Hệ thống Enemies**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Hình ảnh** | **Quỹ đạo di chuyển** | **Điểm nhận được** |
| Báo đốm |  | Di chuyển bất ngờ từ trái hoặc phải sang. | 300 |
| Dơi |  | Di chuyển quỹ đạo hình sin. | 100 |
| Đại bàng |  | Di chuyển bất ngờ tới player, sau đó bay xung quanh. | 300 |
| Chim sẻ |  | Là holder đứng yên, chứa item. | Phụ thuộc theo item bên trong. |
| Bươm bướm |  | Là holder đứng yên, chứa item. | Phụ thuộc theo item bên trong. |
| Lính bazooka |  | Đứng yên, khi xuất hiện sẽ bắn 1 viên đạn. | 200 |
| Quỷ nâu |  | Di chuyển theo quỹ đạo thẳng, từ X1 tới X2 và ngược lại. | 100 |
| Lính bắn tỉa |  | Bắn mỗi lần 3 viên đạn, quỹ đạo thẳng. | 300 |
| Xác ướp |  | Ném kiếm theo quỹ đạo parabol. | 300 |
| Boxing man |  | Di chuyển bất ngờ, tốc độ cao. | 300 |
| Boss |  | Di chuyển hình vòng cung, tấn công theo chu kì . | 3000 |

1. **Hệ thống Item**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Hình ảnh** | **Khả năng** |
| Star |  | Nhận được vũ khí Star. |
| Windmill Star |  | Nhận được vũ khí Windmill Star. |
| Time freeze |  | Ngưng đọng thời gian, đông cứng enemies trong 5s. |
| Bonus blue |  | Được cộng 500 điểm. |
| Spirit strength red |  | Được cộng 10 năng lượng. |
| Spirit strength blue |  | Được cộng 5 năng lượng. |
| Bonus red |  | Được cộng 1000 điểm. |
| Fire wheel |  | Nhận được vũ khí Fire Wheel. |

1. **Hệ thống vũ khí (Weapon)**

* Vũ khí của Ninja

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Hình ảnh** | **Khả năng** | **Năng lượng tiêu tốn** |
| Star |  | Phóng ra phi tiêu, khi va chạm với enemy sẽ nổ | 3 năng lượng |
| Windmill Star |  | Phóng ra phi tiêu, nếu va chạm với enemy sẽ bay về ninja với quỹ đạo giống boomerang. | 5 năng lượng |
| Fire wheel |  | Phóng ra một bánh xe lửa lên cao tiêu diệt enemy. | 5 năng lượng |
| Katana (Knife) |  | Vũ khí mặc định của Ninja. | 0 năng lượng |

* Vũ khí của Enemy

|  |  |  |
| --- | --- | --- |
| **Tên** | **Hình ảnh** | **Khả năng nếu va chạm Ninja** |
| Đạn của lính bắn tỉa |  | Bay theo chiều ngang và mất một máu của Ninja. |
| Đạn của lính bazooka |  | Bay theo chiều ngang và mất hai máu của Ninja. |
| Kiếm của xác ướp |  | Bay với quỹ đạo tương tự parabol và mất một máu của Ninja. |
| Đạn của Boss |  | Bay theo chiều ngang và mất một máu của Ninja. |

# **CÁC KĨ THUẬT SỬ DỤNG TRONG VIỆC PHÁT TRIỂN GAME**

## **Thư viện DirectX**

* Trong bối cảnh ngành công nghiệp game phát triển mạnh mẽ, kéo theo là cả một hệ thống các ngành khác hỗ trợ cho game phát triển. Hầu hết các game PC ngày nay đều được phát triển trên nền tảng sơ cấp nhất là thư viện đồ họa DirectX. DirectX đã giúp các người chơi game trải nghiệm được những công nghệ đồ họa mới nhất và chân thật nhất với con người.
* DirectX là một hệ thống các API phục vụ quá trình phát triển Game trên nền tảng Windows. Microsoft cung cấp DirectX như một biện pháp cho giới lập trình viên một hệ thống các API đã được chuẩn hóa, khi sử dụng DirectX lập trình viên sẽ không còn phải quan tâm vấn đề phần cứng máy tính. Mỗi đối tượng trong API cung cấp một phương thức truy cập với từng loại phần cứng khác nhau, bao gồm: Hệ thống đồ họa, Hệ thống âm thanh, Hệ thống mạng.
* Cách cài đặt DirectX nằm ở phần tư liệu tham khảo.

## **Kĩ thuật Design Pattern**

* Design Pattern trong game này được sử dụng ở việc xây dựng và thiết kế các State (trạng thái) của Ninja. Chi tiết xử lý phần mã nguồn nằm ở thư mục NinjaState trong project khi được mở bằng Visual Studio.
* Mấu chốt của Design Pattern trong game chủ yếu là sự liên kết qua lại giữa các state và chủ thể Ninja, các State thực thể đều kế thừa từ class State trong mối quan hệ Ninja – NinjaData - State. Ban đầu, nhóm sử dụng class NinjaData để tạo sự liên kết giữa hai đối tượng Ninja và State, sau đó, khi áp dụng Singleton Pattern cho class Ninja thì đối tượng Ninja có hai cách gọi, cách thứ nhất là sử dụng con trỏ Instance , cách hai là sử dụng đối tượng ninjaData để gọi.
* Nhằm tạo sự liên kết sâu hơn trong các sự kiện từ bàn phím, mỗi class kế thừa từ base class State đều có hàm Update, HandleKeyboard riêng. Sau đó chỉ việc gọi và xử lý.

## **Kĩ thuật sử dụng Tiled map**

* Map, hay background là phẩn không thể thiếu trong game. Và chúng ta phải tạo map làm sao cho hiệu suất tốt và tiết kiệm tài nguyên nhân. Tiled map sử dụng việc vẽ map bằng cách dùng các tile (Các ô hình ảnh) trên 1 tấm ảnh lớn (tile set) để vẽ lên. Những phần hình ảnh giống nhau sẽ chỉ sử dụng 1 hình ảnh để vẽ nên sẽ tiết kiệm được rất nhiều tài nguyên máy.
* Cấu tạo map gồm 2 phần, TileSet.png và Matrix.txt (hoặc các định dạng giống như vậy). Chúng ta sẽ lập trình game sao cho có thể đọc và xử lý được hai file này, và trả về đúng tấm ảnh ban đầu.
* Vấn đề này được giải quyết trong class **TileMap** của project. Phần xử lý hình ảnh ban đầu để tạo ra TileSet.png và Matrix.txt được lập trình ở phần mềm thứ ba, xem tư liệu tham khảo. Bản chất của việc so sánh các tile có giống nhau hay không nằm ở việc duyệt từng pixel trên ảnh, nếu có pixel có chỉ số RGB khác nhau thì chứng tỏ hai tile đó không cùng loại, phần mềm sẽ cấp hai ID phân biệt đó. Ngược lại, Sẽ cấp một ID và lưu hai tile đó dưới cùng một hình ảnh.

## **Grid – thuật toán phân hoạch không gian**

* Tại sao phải cần thuật toán phân hoạch ? Vì tài nguyên máy tính là có hạn, nên việc xử lý những gì cần xử lý là điều tốt nhất, không nên xử lý những phần không cần thiết. Nên thuật toán này được sinh ra.
* Grid có cùng bản chất với Tiled map, tức là nó sử dụng mảng hai chiều các ô vuông (Cells), trong các Cell đó chứa các Objects.
* Các vấn đề cần xử lý trong grid là

1. Các object được xử lý thế nào khi nó nằm trên nhiều cell khác nhau
2. Việc các object di chuyển qua lại giữa các cell thì làm thế nào để lấy được thông tin
3. Việc lấy thông tin các object cần xử lý, không cần xử lý như thế nào
4. Nhập, xuất như thế nào là tiện dụng và dễ thay đổi nhất
5. Việc thêm – xóa có gây bộ nhớ thừa hay không

* Chi tiết về phần xử lý thuật toán này nằm ở các class Grid, Cell trong project.

## **Các kĩ thuật xử lý va chạm**

* Xử lý va chạm trong game, nhóm sử dụng hai cách để xử lý. Cách thứ nhất là sử dụng sự va chạm (đè lên nhau) cơ bản của hai RECT. Tuy nhiên vấn đề sử dụng này lại bộc bộ nhiều khuyết điểm khi các vật thể di chuyển quá nhanh và trong không xác định được phương (hướng) va chạm. Vậy nên, áp dụng thuật toán va chạm Swept AABB vào để xử lý các va chạm đó.
* Trong project này, nhóm sử dụng RECT cho các dạng va chạm như Ground Detection, va chạm giữa Katana (vũ khí cơ bản của ninja) và các Item (cũng như Enemies). Bên cạnh đó, nhóm sử dụng SweptAABB cho các trường hợp Ninja di chuyển và cần xác định phương (hướng) va chạm với enemies, va chạm với tường, …
* Chi tiết phần này được trình bày chủ yếu trong class Ninja và class Collision. Va chạm của ninja được chia ra làm nhiều loại, với Item, với Ground, với Enemies, … đều có cách sử lý khác nhau. Nhóm sử dụng switch – case cùng với các hàm get-set của class Grid để lấy ra đối tượng phù hợp cần xử lý.

# **KẾT LUẬN**

## **Đánh giá mức độ hoàn thiện**

* Theo đánh giá cá nhân của nhóm, game chạy tốt. Vẫn còn bug nhưng không đáng kể.
* Việc tối ưu bộ nhớ chưa thực sự tốt. Còn nhiều con trỏ thừa chưa được xử lý

## **Hướng phát triển**

* Vì thời gian có hạn, nếu có thể nhóm sẽ tối ưu game, thêm mục chuyển màn game sao cho sinh động hơn và phần logic trong game hoàn thiện tốt hơn.

## **Tài liệu tham khảo**

* + Project github của nhóm: <https://github.com/lqt1912/Ninja_Gaiden_NES>
  + Tài liệu về thư viện DirectX tại MSDN: <http://bit.ly/2RhsQKO>
  + Giáo trình NMPTG : <https://bit.ly/2sJN3iy>
  + Loctho95 blogspot: <http://bit.ly/2BcmBTDs>
  + Excel sửa lỗi của lớp: <http://bit.ly/2WjC3WU>
  + Series về DirectX tại stdio.vn