

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

MÔN HỌC: NHẬP MÔN PHÁT TRIỂN GAME

Đề tài: Game Captain America and The Avengers

SINH VIÊN THỰC HIỆN:

- Lưu Biều Nghị - 17520794
- Nguyễn Duy Minh – 17520754
- Đinh Hoàng Luân – 17520727

GVHD: Ths. Nguyễn Vĩnh Kha

Thành phố Hồ Chí Minh, ngày 9 tháng 8 năm 2019

MỤC LỤC

Mục lục

Phần I: Giới thiệu lược về game Captain America and The Avengers	3
1. Nhà phát hành:	3
2. Nguồn gốc:	3
Phần II: Thiết kế các thành phần trong Game.....	4
1. Danh sách các màn trong Game:	4
2. Danh sách các objects:	6
2.1 Nhân vật.....	6
2.2 Vật phẩm (Item):	7
2.3 Vật thể khác:	8
3. Hướng dẫn người chơi:.....	9
Phần III: Các kỹ thuật được áp dụng trong lập trình Game	9
1. Singleton Design Pattern:	9
2. Sử dụng FSM (Finite State Machine) biểu diễn trạng thái các đối tượng Game:	10
3. Quy hoạch không gian với Grid:	11
4. Xử lý va chạm sử dụng phương pháp SweptAABB:	11
5. Tilemap:.....	12
Phần III: Đánh giá và kết luận	12
1. Tổng kết và đánh giá:	12
2. Bảng phân công công việc giữa các thành viên	13
Phần IV: Tài liệu tham khảo	13

Lời nói đầu

Game đã và đang là một phần không thể thiếu trong cuộc sống của nhiều người. Ngành công nghiệp Game cũng đang trên đà phát triển mạnh mẽ, và môn học Nhập môn phát triển Game hỗ trợ chúng em xây dựng một bộ phông vững chắc để đón đầu xu hướng này.

Đồ án được hoàn thành dựa trên một sự cố gắng và công sức không nhỏ của cả nhóm đã bỏ ra, để đạt được mức độ hoàn chỉnh như hiện tại. Tuy nhiên, sẽ không đạt được thành quả như ngày hôm nay, nếu thiếu đi sự hướng dẫn và chỉ bảo tận tình từ thầy **Nguyễn Vĩnh Kha**.

Xin cảm ơn thầy đã nhiệt tình hướng dẫn để chúng em có được kết quả như ngày hôm nay.

Phần I: Giới thiệu lược về game Captain America and The Avengers

1. Nhà phát hành:

Captain America and The Avengers là một tựa game 2D được phát triển và phát hành bởi Data East vào năm 1991.

2. Nguồn gốc:

Dựa trên cốt truyện của Marvel Comics kể về cuộc phiêu lưu của các anh hùng Avengers nhằm đánh bại Đầu Lâu Đỏ (Red Skull), kẻ đứng đầu tổ chức hắc ám HYDRA. Red Skull đã âm mưu xây dựng một đội quân các siêu chiến binh với vũ khí tối tân nhằm lật đổ và thống trị thế giới. Bên cạnh hắc tặc ta còn có các cận vệ xấu xa khác như Klaw, the Living Laser, Wizard, The Juggernaut, ...



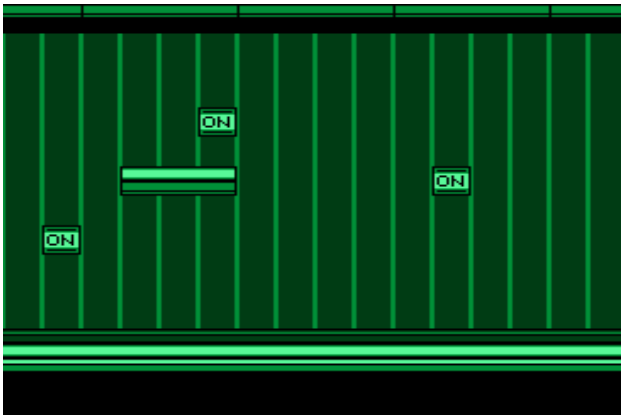
Phần II: Thiết kế các thành phần trong Game

1. Danh sách các màn trong Game:

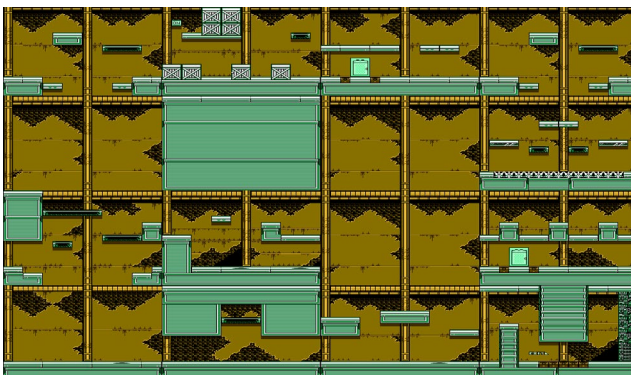
- Màn 1.1: Charleston.



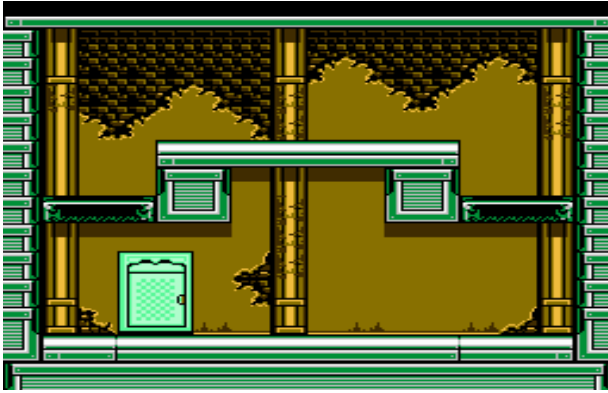
- Màn 1.2: Charleston – Boss.



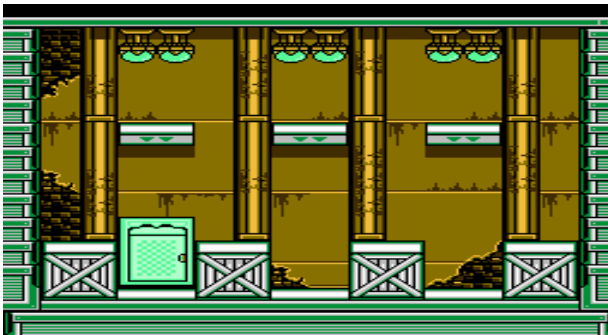
- Màn 2.1: Pittsburgh.



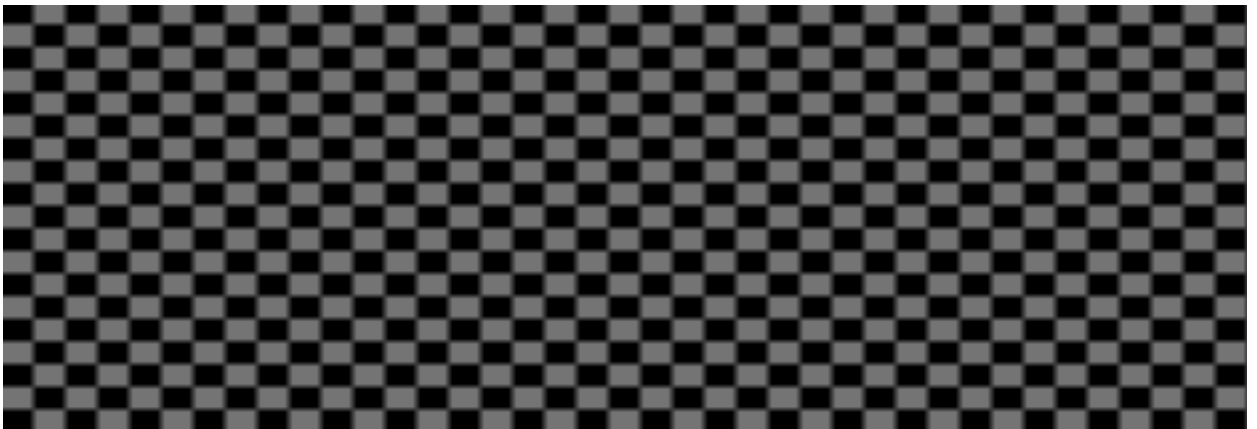
- Màn 2.1.1: Pittsburgh – Door 1 (cửa đầu tiên, nằm phía trên cùng map Pittsburgh).



- Màn 2.1.2: Pittsburgh – Door 2 (cửa thứ hai, nằm phía bên phải map Pittsburgh).



- Màn 2.2: Pittsburgh – Boss (màn cuối).

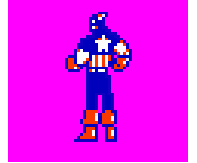


2. Danh sách các objects:

2.1 Nhân vật

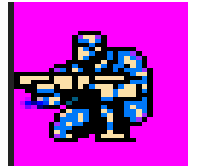
a. Captain America (Player) (4 máu / mạng, số mạng khởi đầu: 2)

- Nhân vật do người chơi điều khiển. Có một bộ rất nhiều kỹ năng mà người chơi có thể sử dụng để tấn công hoặc phòng thủ.



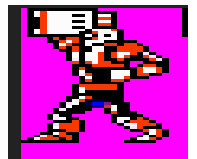
b. Blue Solder (2 máu):

- Lính canh, có thể bắn đạn.
- Có thể bắn tại chỗ hoặc chạy một lúc rồi bắn.



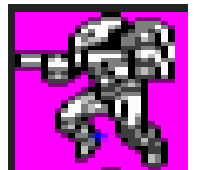
c. Red Rocket Robot (5 máu):

- Robot, có khả năng bắn hỏa tiễn khi đứng và ngồi.
- Có thể di chuyển.



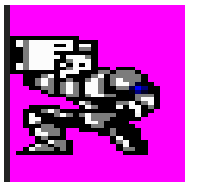
d. White Flying Robot (2 máu):

- Có thể bay theo hình Ellipse.
- Bắn ra đạn tên lửa có thể theo mục tiêu với góc nhất định.



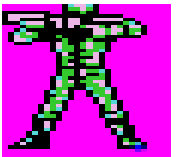
e. White Rocketeer (5 máu):

- Không bắn đạn.
- Có khả năng né khiên khi khiên bay tới.



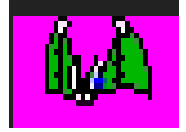
f. Green Solder (2 máu):

- Giống solder nhưng có thể vừa đi (hoặc đứng tại chỗ) vừa bắn và bắn đạn rất nhanh.



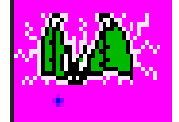
g. Normal Bat (2 máu):

- Dơi ban đầu ở trạng thái giống Item Looter, khi bị đánh thì chuyển sang trạng thái bay và bay 1 quỹ đạo cố định.
- Có khả năng gây sát thương khi va chạm.



h. Eletric Bat (2 máu):

- Dơi có khả năng phóng ra điện gây shock điện.
- Khi đang ở trạng thái phóng điện thì bất tử.



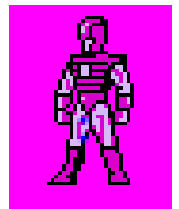
i. Canon (60 máu):

- Có thể bắn ra đạn sau mỗi 1.5s.
- Khi bị tác động thì sẽ quay 1 góc 45 độ ngược chiều kim đồng hồ.



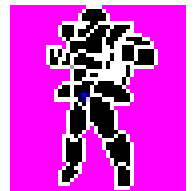
j. Boss Wizard (60 máu):

- Là con boss ở cuối màn 1.
- Có thể bay lượn.
- Có thể bắn ra 2 loại đạn khác nhau.
- Có thể tắt đèn khi đang đánh và ẩn thân trong bóng tối.



k. Mini Boss (3 máu ở trạng thái ném thùng, 16 máu khi ở trạng thái cuồng nộ):

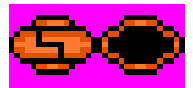
- Có thể bắn ra đạn gây sát thương lớn.
- Bất tử, chỉ có thể bị gây sát thương khi đang cầm thùng bằng cách đánh vào thùng.
- Khi ở trạng thái cuồng nộ, Boss được tăng tốc độ chạy và tốc độ bắn đạn tăng lên.



2.2 Vật phẩm (Item):

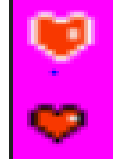
a. Item Container:

- Chứa các item trong game.
- Khi bị đánh thì rớt item ra, rớt tối đa 3 item 1 lúc.



b. Heart:

- Có tác dụng hồi máu.
- Tim nhỏ hồi 2 máu, tim lớn hồi 4 máu (1 mạng).



c. Ngọc:

- Đây là 2 vật phẩm dùng để tăng điểm cho người chơi. Ngọc nhỏ sẽ có trị giá 1 power, ngọc to sẽ có trị giá 2 power. Khi tích lũy đủ số 10 power, người chơi sẽ được nâng cấp (có nhiều máu hơn).



d. Exit Item:

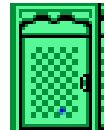
- Đây là vật phẩm dùng để qua màn tiếp theo. Người chơi cần ăn vật phẩm này để mở khóa màn chắn từ trường, từ đó có thể qua được màn chơi tiếp theo.



2.3 Vật thể khác:

a. Door:

- Chuyển sang màn con trong cùng một màn.



b. Moving Platform:

- Player có thể đứng trên platform như ground, platform di chuyển qua lại theo quỹ đạo riêng của nó.



c. Disappearing Platform:

- Công dụng như moving platform, tuy nhiên platform này không di chuyển và player chỉ đứng được khi platform appear (2 disappear platform được ghép lại).



d. Hanging Rope:

- Khi va chạm, player chuyển sang trạng thái climbing.



e. Electric Shock:

- Xuất hiện ở cuối map pittsburgh, khi va chạm player sẽ mất hết máu và chết.



f. Spike:

- Khi va chạm, player bị mất máu và chuyển sang trạng thái beaten.



g. On – Off:

- Dùng để chuyển đổi giữa map dark và map light (bật tắt đèn).



3. Hướng dẫn người chơi:

- Nhấn → để di chuyển qua phải.
- Nhấn ← để di chuyển qua trái.
- Nhấn → nhanh ← hoặc 2 lần liên tiếp thì sử dụng skill lướt.
- Nhấn ↓ để ngồi xuống.
- Nhấn ↓ và Z trên GROUND (mặt đất) thì rơi xuống bên dưới (skill độn thổ).
- Nhấn ↓ để lặn nếu đang ở dưới nước.
- Nhấn ↑ để giơ khiên lên trên.
- Nhấn Space để nhảy lên. Nếu nhấn và giữ một lúc thì sẽ xoay trên không.
- Nhấn Z:
 - Khi đang đứng và cầm khiên: Ném khiên ra và gây sát thương.
 - Khi không có khiên: Đắm về phía trước.
 - Khi đang nhảy: Đá về phía trước.

Phần III: Các kỹ thuật được áp dụng trong lập trình Game

1. Singleton Design Pattern:

- Là một pattern trong lập trình cho phép tại một thời điểm chỉ có 1 thực thể của Class được tạo ra.

- Một số đối tượng là Singleton trong game:

+ Danh sách các khởi tạo DirectX: Graphics, Sound, Keyboard, ...

+ Player

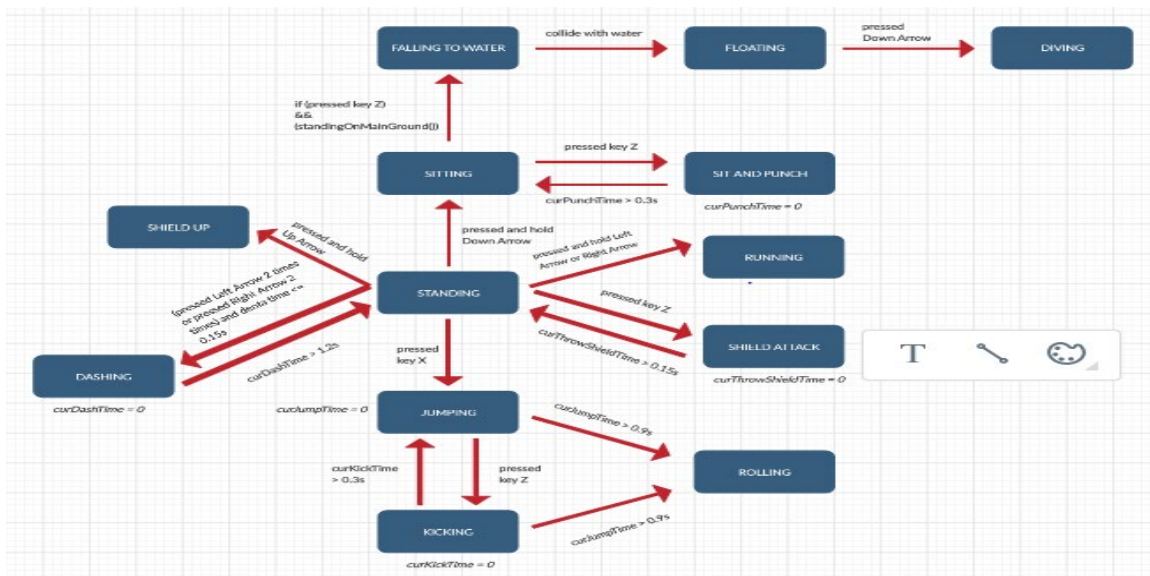
```
SpriteManager* SpriteManager::instance = NULL;

SpriteManager *SpriteManager::getInstance()
{
    if (!instance)
        instance = new SpriteManager();
    return instance;
}
```

2. Sử dụng FSM (Finite State Machine) biểu diễn trạng thái các đối tượng Game:

- Mỗi đối tượng (objects) trong Game đều có các trạng thái (State) riêng của chúng (ngoài trừ các đối tượng tĩnh như Ground, Solid, ...). Tại mỗi thời điểm, tùy vào sự kiện (event) trong quá trình GameRun mà các đối tượng sẽ chuyển trạng thái cho phù hợp với quy luật của chúng.

- Ta có thể sử dụng sơ đồ FSM (Finite State Machine) nhằm quản lý các trạng thái và sự chuyển đổi giữa chúng một cách dễ dàng hơn, giải quyết các ngoại lệ (exception) giữa các quá trình chuyển trạng thái tốt hơn.



Sơ đồ biểu diễn trạng thái của Player

3. Quy hoạch không gian với Grid:

- Grid là một cách thức dùng để tối ưu hóa việc xử lý các object. Trong Grid, ta phân chia thế giới thực ra thành các Cell. Thay vì kiểm tra toàn bộ tất cả các object trong thế giới thực, chúng ta chỉ kiểm tra những phần cần được quan tâm.

Grid có các thông số quan trọng sau:

- **Cell Width, Cell Height:** Chiều cao, chiều rộng của từng ô nhỏ mà dựa vào đó, chúng ta sẽ chia thế giới thực của chúng ta ra.
- **Current Activated Zone:** Là vùng hiện tại được quét bởi Camera. Đây sẽ là vùng diễn ra hoạt động Spawn, Update và Render.

Khởi tạo Grid:

Để khởi tạo Grid, ta phải truyền vào 2 thông số:

1. Map Width: Chiều rộng của map mà ta cần áp dụng Grid.
2. Map Height: Chiều cao của map mà ta cần áp dụng Grid.

Cơ chế hoạt động của Grid bao gồm 4 bước cơ bản như sau (xem như ta đã có Grid từ việc khởi tạo như trên):

1. Xác lập vùng kích hoạt (các Cell được kích hoạt) dựa vào thông tin của Camera ở thời điểm hiện tại.
2. Xử lý va chạm của các Cell kề nhau (do Game không có việc tấn công ở một khoảng cách > kích thước Cell, vì vậy ta chỉ cần kiểm tra việc va chạm của các Cell kề nhau).
3. Cập nhật tất cả các object trong các Cell được kích hoạt (bao gồm hồi sinh Object (Spawn), xóa Object khi nó ra khỏi vùng Active Zone, di chuyển object qua lại giữa các Cell, ...).
4. Vẽ tất cả các object trong các Cell được kích hoạt.

4. Xử lý va chạm sử dụng phương pháp SweptAABB:

- SweptAABB là một kĩ thuật dùng để phát hiện và dự báo hướng va chạm giữa một vật thể và một vật thể khác trong cùng không gian. SweptAABB sẽ cho ta biết được hướng và thời gian hai vật sẽ va chạm với nhau. Tuy vậy, SweptAABB vẫn còn một số hạn chế như:

- Các vật thể thay đổi kích thước trong quá trình xét va chạm (trong frame xét va chạm và frame kế tiếp) thì kết quả của SweptAABB có thể không đúng.
- Khi xét va chạm, chỉ cho phép một vật thể chuyển động, vật còn lại phải đứng yên tương đối so với vật kia.
- SweptAABB chỉ sử dụng được cho vật thể hình chữ nhật hoặc hình vuông, chưa thể áp dụng cho vật thể có hình dạng khác.
- SweptAABB không phát hiện được va chạm nếu 2 vật đã va chạm với nhau.
- Trong frame xét va chạm và frame kế tiếp, nếu một trong hai vật thay đổi vận tốc thì SweptAABB có thể cho ra kết quả sai.
- Nếu hai vật thể đều đứng yên, SweptAABB sẽ cho ra kết quả không va chạm. Trong trường hợp 2 vật thể đã va chạm thì cần một kĩ thuật khác để xét va chạm.

- Để xét va chạm trong trường hợp 2 vật đã va chạm, nhóm sử dụng giải pháp xét phần giao nhau giữa hai vật thể để cho ra hướng đã va chạm. Giải pháp này giúp phát hiện 2 vật đã va chạm với nhau. Tuy nhiên, giải pháp này cũng còn một hạn chế nhỏ là chỉ cho ra được hướng va chạm và không thể phát hiện 2 vật đã va chạm trong thời gian bao lâu và không xác định được thời điểm hai vật sẽ kết thúc va chạm.

5. Tilemap:

- Tilemap là một kỹ thuật dùng để load map lên theo từng phần nhất định mà không cần thiết phải lưu toàn bộ map vào bộ nhớ. Ưu điểm của TileMap là với việc load từng phần sẽ có yêu cầu bộ nhớ nhỏ hơn. Tuy nhiên, nhược điểm là CPU chiếm dụng sẽ cao hơn.

Phần III: Đánh giá và kết luận

1. Tổng kết và đánh giá:

- Sau 3 tuần làm việc, nhóm đã hoàn thành được một chương trình Game, mô phỏng khá tốt so với Game gốc. Các thành phần liên quan đến player như đấm, đá và các thể loại tấn công khác đều đã được hoàn thiện.

- Game đồng thời cũng đã được tối ưu hóa tốt về mặt hiệu suất, dẫn đến Game mượt mà cho người chơi.

- Tuy nhiên Game cũng khó có thể tránh khỏi những thiếu sót nhỏ. Một vài điểm có thể kể đến là:

- Việc Collision giữa các Object với nhau đôi khi vẫn còn lỗi.
- Thời gian delay ở một vài vị trí vẫn chưa hoàn toàn chính xác so với Game gốc.

2. Bảng phân công công việc giữa các thành viên

Thành viên	Công việc
Lưu Biều Nghị	Hiện thực TileMap. Hiện thực các Scene và SceneManager để chuyển giữa các Scene. Hiện thực Camera. Hiện thực Grid. Hiện thực một vài Enemy: Red Rocket Robot, White Flying Robot.
Nguyễn Duy Minh	Implement nhân vật Player: - Implement các state của player - Xử lý các event khi có va chạm xảy ra giữa player và các object liên quan - Quản lý các thuộc tính của player trong trò chơi như: máu, damage, power-up, ... Implement khiên Implement một số Enemy: Dơi, Canon
Đinh Hoàng Luân	Hiện thực SweptAABB. Hiện thực một số Enemy: Boss Wizard, Boss Mini, Blue Soldier, Green Soldier. Hiện thực một vài vật thể khác như: Item Container, Item. Tổ chức và quản lí Resources: hình ảnh, âm thanh...

Phần IV: Tài liệu tham khảo

- Gamedev.net
- Drunkenhyena.com
- Gameprogrammingpatterns.com
- loctho95.blog
- Giáo trình Nhập môn Phát triển Game – ĐHCNTT.

[illegible]