

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: Captain America and the Avengers 1991

Giảng viên hướng dẫn:

- Nguyễn Vĩnh Kha

Sinh viên thực hiện:

- Vũ Tuấn Hải - 17520433
- Nguyễn Du Lịch - 17520096

TP Hồ Chí Minh, ngày 8 tháng 4 năm 2019

## Mục lục

<b>I. Giới thiệu game</b>	2
1. Nội dung	2
2. Gameplay	2
3.1. Map	3
3.2. Vật thể	4
3.2.1. Vật thể động	4
3.2.2. Vật thể tĩnh	8
3.3. Hướng dẫn chơi	9
<b>II. Kỹ thuật lập trình game</b>	9
1. Design Pattern	9
1.1. Singleton	9
1.2. State machine	10
2. Tilemap	11
3. SweptAABB	11
4. Grid	12
<b>III. Kết luận</b>	12
1. Bảng phân công	12
2. Tự đánh giá	13
3. Tổng kết	13
4. Tài liệu tham khảo	13
5. Resources	13
<b>IV. Lời cảm ơn</b>	13

## I. Giới thiệu game

### 1. Nội dung



Captain America and the Avengers là một tựa game được phát triển và phát hành bởi Data East vào năm 1991. Game bao gồm các nhân vật trong biệt đội the Avengers thuộc Marvel Comics với nội dung xoay quanh cuộc phiêu lưu của Captain America / Hawk Eye để tìm kiếm Iron Man - Vision đã bị bắt cóc và đối chọi với Red Skull đang tập hợp một đội quân tay sai với âm mưu thôn tính thế giới.

Cùng với việc phải chiến đấu với những tay lính đánh thuê và tội phạm, người chơi cũng phải đối mặt với những boss sừng sỏ như Wizard, Ultron, sát thủ Crossbones và cuối cùng là Red Skull.

### 2. Gameplay

Người chơi có thể chọn đóng vai một trong hai thành viên của Avengers: Captain America và Hawkeye. Mỗi nhân vật có thể chiến đấu tay đôi, ăn vật phẩm và sử dụng vũ khí tầm xa (khiên của Captain America hoặc cung tên của Hawkeye). Một Avengers khác là the Wasp sẽ xuất hiện để thông tin cho người chơi mỗi khi chuyển màn.

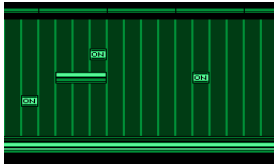
### 3. Thiết kế

### 3.1. Map

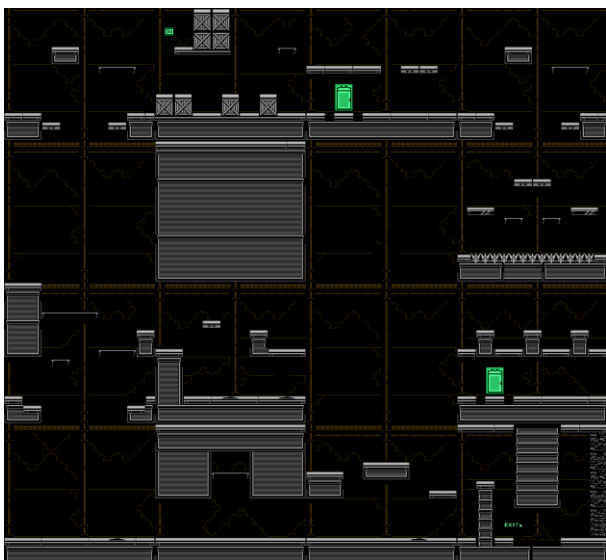
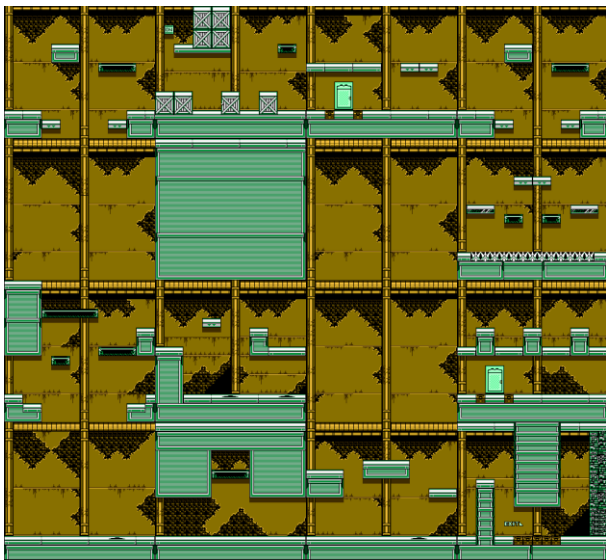
#### 3.1.1. Chaleston



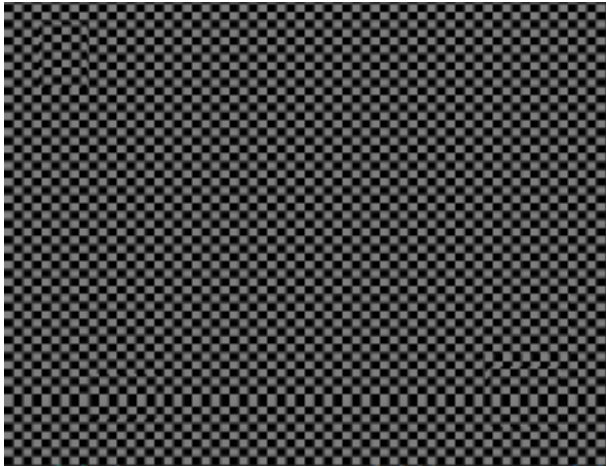
#### 3.1.2. Chaleston Boss



#### 3.1.3. Pittsburgh



### 3.1.4. Pittsburgh Boss



## 3.2. Vật thể

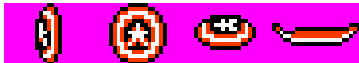
### 3.2.1. Vật thể động

#### 3.2.1.1. Người chơi



Captain America (HP: 8): nhân vật được người chơi điều khiển, vũ khí chính là khiên Vibranium, ngoài ra Captain còn có thể cận chiến nhưng kém hiệu quả hơn.

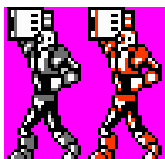
#### 3.2.1.2. Khiên



Vũ khí chính của Captain, khiên sở hữu rất nhiều chức năng hỗ trợ cho người chơi:

- Tấn công: khiên có thể bị ném đi và tự động trở về, gây sát thương (2) mỗi lần tiếp xúc cho tất cả những kẻ địch trên đường đi. Trong trạng thái dashing & shield down của Captain, khiên gây sát thương (6).
- Phòng thủ: khiên phản lại tất cả các loại đạn thường (đạn không có tính chất nổ).
- Hỗ trợ: trong trạng thái shield down, khiên giúp người chơi ngồi trên được những địa hình như mặt nước và gai.

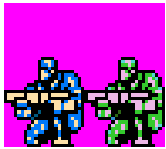
#### 3.2.1.3. Bot



Gray/Red rocket robot (HP: 5): robot có khả năng bắn tên lửa thẳng hoặc chéo gây sát thương cho người chơi. Rocket robot có thể ngồi/ đứng bắn, chạy và nhảy để né.

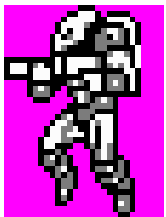
Rocket robot có 3 chế độ tấn công khác nhau:

- Tấn công thụ động: rocket robot tấn công mỗi 2s đồng thời chuyển sang trạng thái đứng hoặc ngồi.
- Tấn công áp sát: rocket robot di chuyển đến gần người chơi rồi chuyển sang chế độ tấn công thụ động.
- Tấn công chủ động: rocket robot di chuyển liên tục theo chu kì, khi kết thúc mỗi chu kì, rocket robot chuyển sang chế độ tấn công thụ động trong 2s rồi lại di chuyển.



Blue/Green soldier (HP: 2): là loại bot phổ biến và bắn ra đạn thường. Soldier có thể đứng, bắn, ngồi, chạy và nhảy để né. Soldier có 4 chế độ tấn công khác nhau:

- Tấn công thụ động: Soldier tấn công khi vào trạng thái đứng, sau đó ngồi khoảng 2s và lặp lại chu kì. Tấn công thụ động có 2 chế độ bắn đạn đơn và đạn chùm (gồm 3 viên đạn theo 3 góc khác nhau)
- Tấn công thụ động nâng cao: chu kì tấn công giống như tấn công thụ động nhưng tốc độ tấn công tăng gấp 4 lần.
- Chạy: soldier tăng tốc độ chạy lên gấp đôi và chạy về phía người chơi.
- Tấn công chủ động: soldier chạy về phía người chơi và vào trạng thái tấn công thụ động mỗi 1s.

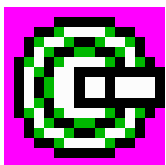


Gray robot (HP: 2): robot có khả năng bay trên không theo hình elip và bắn tên lửa tìm diệt.



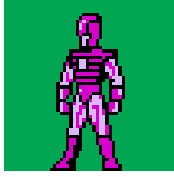
Bat (HP: 1): loại robot bay trên không theo hình số 8. Bao gồm 2 loại:

- Loại thường: không miễn nhiễm sát thương. Tuy nhiên loại này có khả năng nguy trang thành item container, gây bất ngờ cho người chơi.
- Loại phóng điện: miễn nhiễm sát thương khi phóng điện và gây sốc điện cho người chơi.



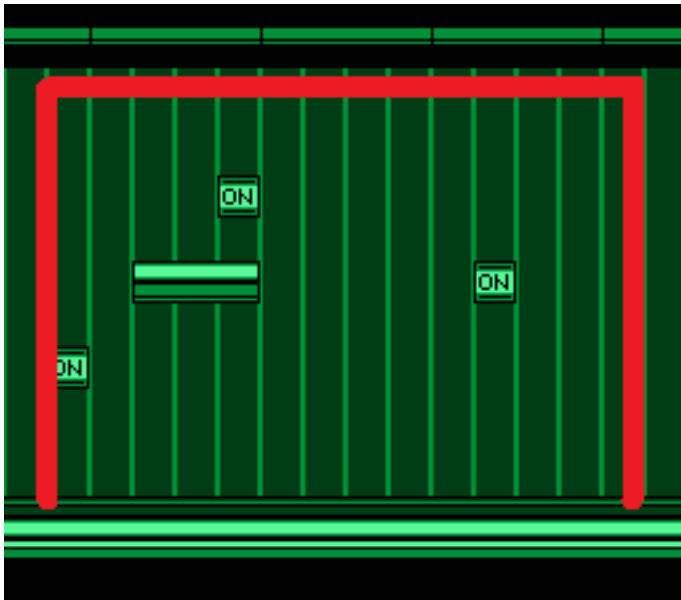
Turret (HP: 64): tháp cố định bắn đạn thường, gây sát thương cho người chơi mỗi 1.5s, tháp xoay và đổi hướng mỗi 45 độ khi nhận sát thương. Sát thương chịu tối đa mỗi lần là 1.

#### 3.2.1.4. Boss



Boss Wizard (HP: 80): boss wizard có 2 combo gây sát thương

- Combo 1: bắn 2 viên đạn laser và 1 viên đạn năng lượng, sau đó bắn 3 viên đạn laser mỗi lần cho đến khi khoảng cách với người chơi đủ gần, khi khoảng cách đủ gần, wizard sẽ tụ lực bắn 1 viên đạn năng lượng sau đó nhảy vào người chơi. Nếu hp wizard không giảm, ông ta sẽ lặp lại combo này, nếu không sẽ chuyển sang combo 2.
- Combo 2: wizard bay hình chữ u ngược 1 – 2 lần và bắn đạn năng lượng mỗi khi trùng hoành độ với người chơi. Khi kết thúc combo 2, wizard sẽ lại thực hiện combo 1.



Khi hp thấp hơn 40, wizard sẽ bay hình chữ u ngược và chạm vào công tắc, gây tắt điện map và mất tầm nhìn cho người chơi.

Khi hp thấp hơn 10, wizard vào trạng thái yếu và thực hiện combo 2 với tần suất cao hơn.



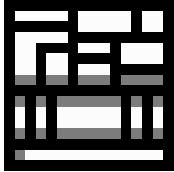
Boss Gragas (HP: 16): boss gragas có 2 trạng thái

- Trạng thái 1: gragas thực hiện combo sát thương bằng cách ném thùng, bắn hai viên super laser, chạy và đổi hướng tấn công, sau đó lặp lại combo. Ở trạng thái

này gragas miễn nhiễm tất cả sát thương. Gragas chỉ có thể chuyển sang trạng thái 2 khi người chơi phá nổ tất cả 3 thùng trước khi hắn ta ném.

- Trạng thái 2: gragas đổi hình dạng, tấn công mạnh hơn bằng cách tăng tần suất bắn super laser và liên tục chạy, sát thương khi tiếp xúc cũng được tăng gấp đôi. Tuy nhiên, trong trạng thái này gragas mất khả năng miễn nhiễm sát thương.

#### 3.2.1.5. Platform



Disappear platform: platform sẽ biến mất khi bị người chơi chạm sau khoảng 0.5, sau đó lặp lại chu kỳ xuất hiện – biến mất mỗi 1.5s.



Moving platform: platform có thể vận chuyển người chơi lên những vị trí cao hơn.

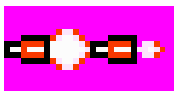


Electric shockwave barrier: xuất hiện sau mỗi 1.5s và biến mất sau mỗi 2s, gây đột tử nếu người chơi chạm phải.



On/ Off switch: platform chuyển màn chơi sang chế độ sáng/ tối nếu người chơi/ boss tấn công.

#### 3.2.1.6. Dạn



Rocket (ST: 2): được tạo ra khi roket robot vào trạng thái sẵn sàng tấn công, rocket có tính chất nổ khi va chạm nên khiên của người chơi không thể đỡ được. Rocket bao gồm 2 loại:

- Loại bắn thẳng: di chuyển theo 1 hướng cố định với tốc độ cố định.
- Loại bắn chéo: di chuyển theo đường thẳng 1 đoạn ngắn sau đó đột ngột đổi hướng với góc 45 độ.

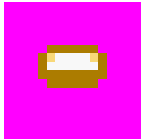




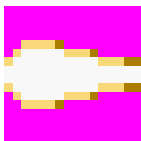
Bullet (ST: 2) được tạo ra bởi soldier hoặc turret khi ở trong trạng thái sẵn sàng tấn công, đạn thường có tính chất như rocket nhưng có thể bị chặn và đổi hướng bởi khiên của người chơi.



Gray rocket (ST: 2): được tạo ra khi gray robot vào trạng thái sẵn sàng tấn công, có khả năng thay đổi hướng bay sao cho hướng vào người chơi. Khiên không thể đỡ gray rocket.



Laser bullet (ST: 5) được tạo ra khi boss wizard thực hiện combo tấn công, laser bullet có tốc độ bay thấp, dễ né tránh và bị chặn bởi khiên.



Energy bullet (ST: 3) được tạo ra khi boss wizard thực hiện combo tấn công, energy bullet có tốc độ bay gấp 3 lần laser bullet, khó né tránh và không thể bị chặn bởi khiên.



Barrel (ST: 1): được tạo ra khi boss gragas thực hiện combo tấn công, barrel được ném đi theo hình parapol với hệ số a âm. Barrel có thể bị phá hủy bởi bất kì đòn tấn công nào, tuy nhiên khiên không thể đỡ được.



Super laser bullet (ST: 3) được tạo ra khi boss gragas thực hiện combo tấn công, laser bullet có tốc độ bay nhanh, khó né tránh và không thể bị chặn bởi khiên.

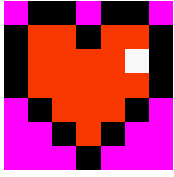
### 3.2.2. Vật thể tĩnh



Item holder: xuất hiện rải rác ở khắp nơi trong màn chơi, là vật thể chứa toàn bộ các item. Item holder sẽ nhả item mỗi khi bị người chơi đánh trúng, khi item không được người chơi thu thập, nó sẽ biến mất.



Exit item: vật phẩm được sinh ra từ 1 trong những item container của mỗi màn chơi.  
Exit item là item tiên quyết để người chơi qua màn.



Half heal item: item giúp người chơi hồi 4 HP.



Full heal item: item giúp người chơi hồi 8 HP.



Gem & point item: các loại item tăng score cho người chơi, item càng lớn thì score cộng càng nhiều.

### 3.3. Hướng dẫn chơi

Người chơi sử dụng 6 phím để điều khiển:

- ↑: người chơi vào trạng thái shield up.
- ↓: người chơi ngồi xuống, nếu trạng thái trước đó là rolling, người chơi sẽ chuyển sang trạng thái shield down. Nếu ở trong trạng thái flowing, người chơi sẽ chuyển trạng thái diving.
- →: người chơi di chuyển sang trái.
- ←: người chơi di chuyển sang phải.
- →→ hoặc ←←: người chơi vào trạng thái dashing.
- Z: người chơi tấn công bằng cách ném khiên, nếu không có khiên, người chơi sẽ đâm về phía trước. Khi ở trong trạng thái nhảy, người chơi sẽ đá.
- X: người chơi nhảy, nếu ấn giữ trong thời gian đủ lâu, người chơi sẽ ở trong trạng thái rolling.
- ↓ + Z: người chơi sẽ đâm thấp.
- ↓ + X: người chơi sẽ rơi xuống platform thấp hơn nếu platform hiện tại là ground.

## II. Kỹ thuật lập trình game

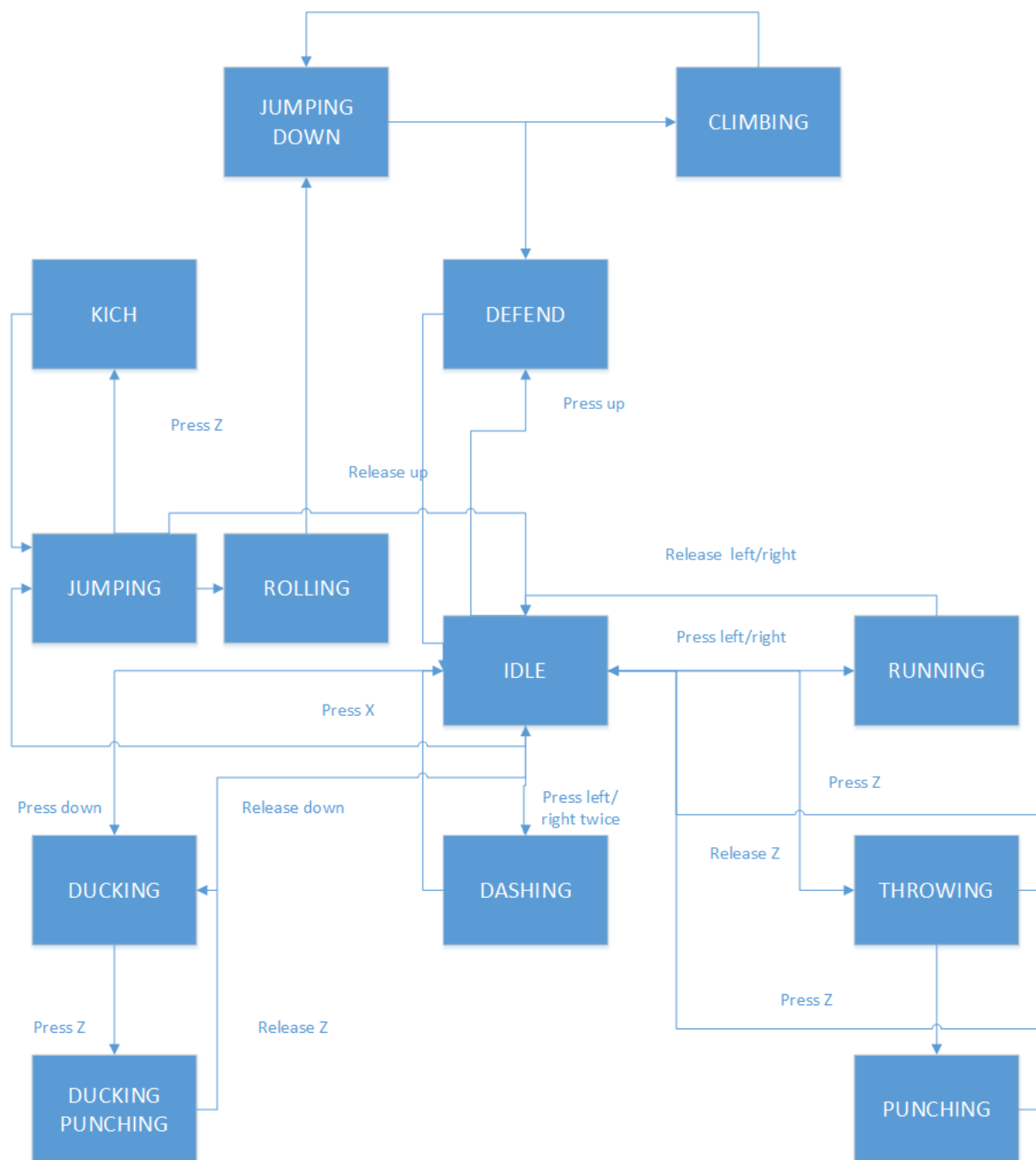
### 1. Design Pattern

#### 1.1. Singleton

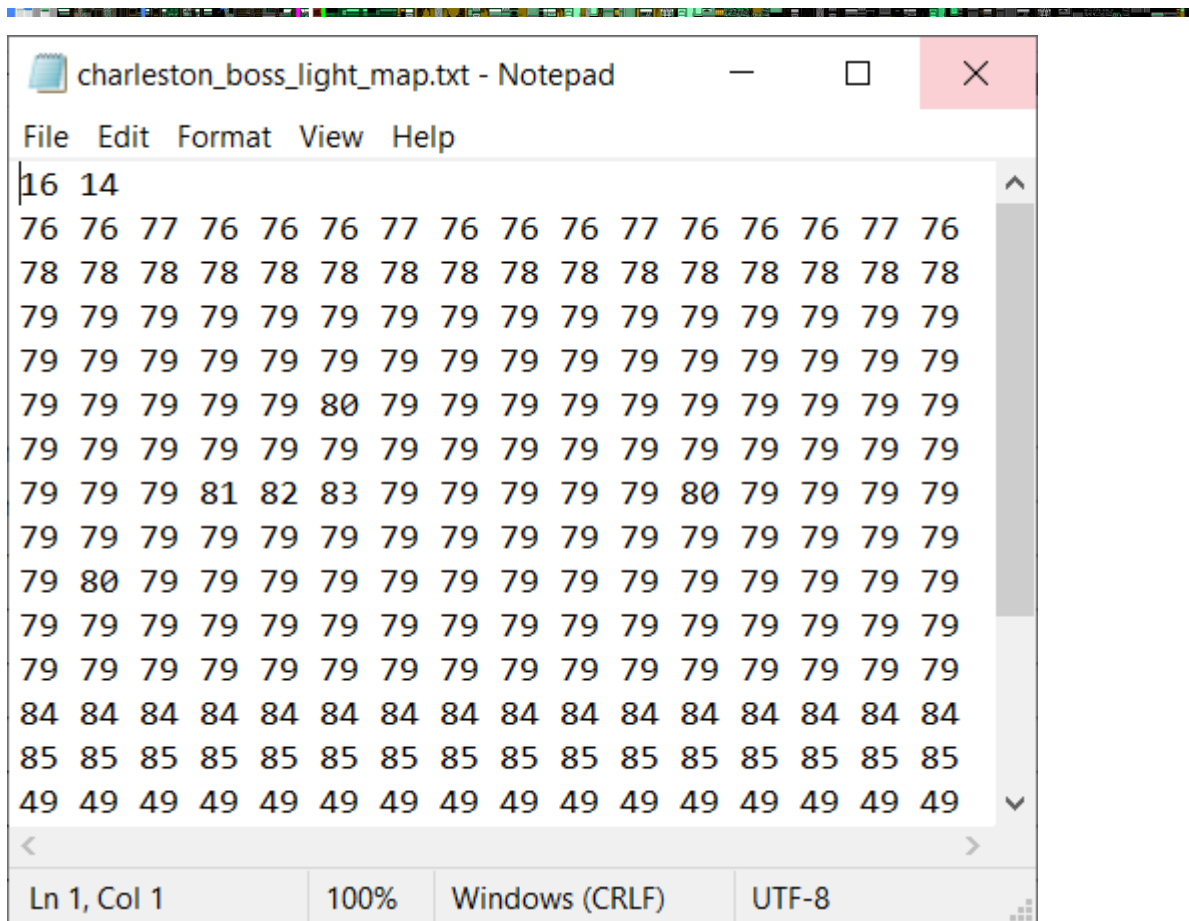
Mỗi class chỉ có duy nhất một instance, khi class được gọi sẽ trả về instance duy nhất này. Singleton được áp dụng vào các object không được tạo nhiều lần như player, camera, keyboard, ...

## 1.2. State machine

Đối tượng có nhiều trạng thái khả dĩ chỉ có thể ở duy nhất một trạng thái ở một thời điểm xác định. Ví dụ như người chơi có thể idle, running, jumping hay punching, nhưng bất kì khi nào cũng ở một trạng thái xác định, hoặc idle, hoặc running, hoặc jumping, ... Mỗi trạng thái chỉ có một con đường duy nhất để chuyển sang trạng thái khác, một trạng thái có thể chuyển sang nhiều trạng thái, tuy nhiên nếu không có khả năng chuyển sang một trạng thái nào đó, nó bắt buộc phải chuyển sang trạng thái trung gian.



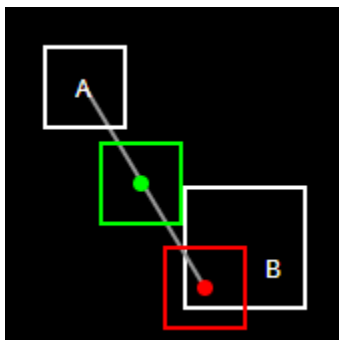
## 2. Tilemap



Việc load cả một texture map lớn lên màn hình nhỏ cộng với việc map có nhiều tile giống nhau gây tình trạng lãng phí tài nguyên (vì chương trình vẽ ra ngoài phạm vi người chơi nhìn thấy). Do đó nên chia nhỏ map ra nhiều tile (16x16) tùy vào vị trí của camera và quyết định tile nào cùng id nào sẽ được vẽ.

## 3. SweptAABB

Là thuật toán dùng để dự đoán trước va chạm và chiều va chạm trong tương lai. Tuy tốn nhiều dòng lệnh hơn xử lý va chạm thông thường nhưng SweptAABB lại giúp nhận biết và phản hồi sau va chạm tốt hơn.



Trong trường hợp hai vật đã va chạm sẵn (collided), việc áp dụng kỹ thuật này có thể dẫn đến phản hồi va chạm sai sót. Do đó thuật toán SweptAABB được sử dụng trong đồ án đã được tinh chỉnh để giải quyết ngoại lệ này.

#### 4. Grid

Thông thường trong một màn sẽ có nhiều vật thể được tạo ra và đặt vào map. Do đó việc xét va chạm nhiều lần là điều không thể tránh khỏi. Grid tương tự như Quadtree giúp dự đoán hai vật thể nào là có thể va chạm với nhau để hạn chế việc kiểm tra nhiều lần. Nếu không có grid, một vật thể ở đầu map vẫn có thể phải kiểm tra va chạm với một vật thể ở cuối map, điều bất khả thi trong thực tế.



Grid dự đoán bằng cách chia nhỏ world ra nhiều khối nhỏ đều nhau gọi là cell và mỗi cell sẽ chứa những vật thể dùng để xét va chạm. Lúc này chỉ những object ở chung cell hay nằm ở các cell gần nhau mới có khả năng va chạm và gọi đến SweptAABB.

### III. Kết luận

#### 1. Bảng phân công

Thành viên	Công việc
Vũ Tuấn Hải	<ul style="list-style-type: none"> <li>• Tạo và quản lý người chơi.</li> <li>• Tạo và quản lý bot.</li> <li>• Tạo và quản lý boss.</li> <li>• Xử lý va chạm giữa người chơi với map, người chơi với bot, và người chơi với boss.</li> <li>• Tìm và xử lý ngoại lệ.</li> <li>• Viết báo cáo.</li> </ul>
Nguyễn Du Lịch	<ul style="list-style-type: none"> <li>• Tạo framework.</li> <li>• Tạo resources.</li> <li>• Tạo và quản lý vật thể tĩnh.</li> <li>• Tạo và load tile map.</li> <li>• Tạo và quản lý âm thanh.</li> <li>• Tìm và xử lý ngoại lệ.</li> </ul>

## 2. Tự đánh giá

- Game đã sử dụng kỹ thuật tile map và phân hoạch không gian grid để tối ưu việc quản lý map, bot và các vật thể khác.
- Game đầy đủ các sprite giống như game gốc.
- Game xử lý tất cả ngoại lệ giống như game gốc.
- Game không bị tình trạng leak memory & leak CPU.
- Game có âm thanh.

Đồ án đã hoàn thành được 99% game gốc (chỉ tính 4 map từ Chaleston đến Pittsburgh boss).

## 3. Tổng kết

- Đồ án đã hoàn thành như mục tiêu của nhóm cũng như yêu cầu môn học đề ra.
- Trong quá trình thực hiện đồ án, kỹ năng giải quyết vấn đề và làm việc nhóm được tiến bộ.
- Kiến thức nền tảng bao gồm C++, OOP và CTDL&GT được vận dụng và phát huy tối đa.
- Nhiều kiến thức mới được tiếp thu như design pattern, grid, tile map, cách xử lý âm thanh và hình ảnh, ...

## 4. Tài liệu tham khảo

- Giáo trình Nhập môn Phát triển Game, PGS.TS Vũ Thanh Nguyên, ThS Đinh Nguyễn Anh Dũng, 2013.
- Singleton Pattern. <https://gameprogrammingpatterns.com/singleton.html>
- State Machine Pattern. <https://gameprogrammingpatterns.com/state.html>
- Code samples. <https://github.com/dungdna2000/gamedev-intro-tutorials>

## 5. Resources

- Map. <https://pighthouse.info/avengers-nes-maps/?fbclid=IwAR2tY8BzvH9K0VnLaNwbG5cbCj6RlbJn4JI6Qz4qNvLtZDva5f2lZoy7sQI>
- Vật thể. [https://www.sprisers-resource.com/nes/captamericaavengers/?fbclid=IwAR2nkPA9nCCI3FkpXc8ViA4QGYyVkOWWZQ5lfuZ-P\\_HxPSEXglwtDlhKi84](https://www.sprisers-resource.com/nes/captamericaavengers/?fbclid=IwAR2nkPA9nCCI3FkpXc8ViA4QGYyVkOWWZQ5lfuZ-P_HxPSEXglwtDlhKi84)

## IV. Lời cảm ơn

Nhóm chúng em xin chân thành cảm ơn những kiến thức được truyền đạt từ thầy – ThS Nguyễn Vĩnh Kha để chúng em có thể hoàn thành đồ án và đạt được kết quả ngày hôm nay.