ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN KHOA CÔNG NGHỆ PHẦN MỀM

----- 000 -----



BÁO CÁO CUỐI KÌ Đề tài: GAME CASTLEVANIA

Nhập môn phát triển game SE102.H12

Giảng viên hướng dẫn:

ThS. Nguyễn Vĩnh Kha

KS. Huỳnh Hồ Thị Mộng Trinh

Sinh viên thực hiện: Nhóm TaHiTu

Trần Thị Minh Trang 14520990

Nguyễn Minh Hiếu 14520288

Hoàng Kim Tuấn 14521042

Tp. Hồ Chí Minh, tháng 01/2017



NHẬN XÉT CỦA GIẢNG VIÊN

LÒI CẨM ƠN

Nhập môn phát triển game được coi là một môn học tương đối khó với mỗi sinh viên học khoa Công nghệ phần mềm. Vì thế, để hoàn thành được tốt đồ án môn học, trước tiên nhóm chúng em xin gửi lời cảm ơn chân thành nhất đến thầy Nguyễn Vĩnh Kha và cô Huỳnh Hồ Thị Mộng Trinh. Chính nhờ sự giảng dạy nhiệt tình, tận tâm của Thầy, Cô cùng những bài tập tuần thú vị đã cung cấp những kiến thức cơ bản, những bài học, những kinh nghiệm quý báu để chúng em có thể hình dung được một cách khái quát các bước làm ra một game là như thế nào và tự tạo ra một game hoàn chỉnh cho đồ án cuối kì. Nhóm đã tiếp thu được nhiều kiến thức hữu ích về các kỹ thuật trong game như SweptAABB, Tilemap, Mapeditor, Quadtree,... và áp dụng thực tiễn vào đồ án một cách hiệu quả.

Bên cạnh đó, nhóm cũng xin gửi lời cảm ơn chân thành đến tất cả các anh chị, các bạn trong và ngoài lớp đã kịp thời hỗ trợ, chia sẻ tài liệu và đưa ra những nhận xét, góp ý chân thành nhằm giúp đồ án game được hoàn thiện hơn. Đặc biệt là các anh chị K8 trong khoa Công nghệ phần mềm đã nhiệt tình chia sẻ kinh nghiệm và giải đáp các thắc mắc của nhóm trong suốt quá trình thực hiện đồ án.

Tuy vậy, trong quá trình thực hiện đồ án, viết báo cáo việc không tránh khỏi những thiếu sót là điều chắc chắn. Vì thế, nhóm rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô và các bạn học cùng lớp để nhóm có thể rút kinh nghiệm và hoàn thiện đồ án một cách tốt nhất.

Nhóm xin chân thành cảm ơn!

Tp. Hồ Chí Minh, tháng 01 năm 2017 Nhóm thực hiện,

TaHiTu

MŲC LŲC

Nội dung

CHƯƠNG 1: GIỚI THIỆU DIRECTX	1
1.1. DirectX là gì?	1
1.2. Úng dụng của DirectX	1
1.3. Phiên bản hiện tại	1
CHƯƠNG 2: GIỚI THIỆU GAME CASTLE VANIA	2
2.1. Tổng quan Game Castle Vania	2
2.2. Giới thiệu Player – Simon	3
2.2.1. Simon	3
2.2.2. Hệ thống vũ khí	3
2.3. Giới thiệu hệ thống Enemy, Boss	4
2.3.1. Enemy	4
2.3.2. Boss	6
2.4. Giới thiệu hệ thống item	7
2.4.1. Các item cơ bản	7
2.4.2. Các item đặc biệt	8
CHƯƠNG 3: ĐỒ ÁN GAME CASTLE VANIA	10
3.1. Các kỹ thuật áp dụng trong game	10
3.1.1. Thuật toán xử lý va chạm AABB	10
3.1.2. Thuật toán xử lý va chạm SweptAABB	10
3.1.3. Transformation	11
3.1.4. Camera	11
3.1.5. Stage, level	11
3.1.6. Tile map	12
3.1.7. Map editor	13
3.1.8. Thuật toán phân hoạch không gian QuadTree	13
3.2. Sơ đồ các class chính trong game	15
3.2.1. Sơ đồ mô phỏng các class object có trong game Castle Vania	15
5.2.1. 50 do no phong cae class object to trong game castle valua	

3.2.2. Sơ đồ mô phỏng các class màn hình	16
3.3. Công cụ hỗ trợ	16
3.4. Kết quả thực hiện	17
3.4.1. Đánh giá mức độ hoàn thiện của Game	17
3.4.3. Bảng phân công công việc	18
3.5. Kết luận	19
TÀI LIỆU THAM KHẢO	20

CHUONG 1: GIÓI THIỆU DIRECTX

1.1. DirectX là gì?

Microsoft DirectX là 1 tập hợp các chương trình đảm nhiệm việc quản lý các tác vụ liên quan đến hình ảnh, âm thanh ví dụ như các trò chơi, trình dựng phim, thiết kế đồ họa... Microsoft DirectX có thể coi như là 1 chiếc cầu nối liền giữa phần cứng và phần mềm. Nó nhận lệnh liên quan đến hình ảnh, âm thanh từ các phần mềm (game, trình dựng phim...) rồi xử lý và truyền tải đến phần cứng (chip âm thanh, card đồ họa...). DirectX được Microsoft xây dựng thành 1 thư việc các tác vụ, mã lệnh v.v... ngày càng đồ sộ và thông minh giúp cho các game ngày càng xử lý âm thanh chân thực, hình ảnh mượt mà sắc nét. Ngày nay DirectX có thể xử lý được các hiệu ứng giả lập âm thanh nhiều chiều phức tạp và những hình ảnh chuyển động có độ phân giải cao.

1.2. Úng dụng của DirectX

Như đã nói, DirectX là 1 chương trình xử lý các tác vụ liên quan đến âm thanh và hình ảnh trên nền windows, nên nếu bạn còn dùng windows thì bạn vẫn còn phải dùng đến DirectX. Hãy tưởng tượng nếu không có DirectX thì chiếc máy tính siêu thông minh sẻ trở lại thành máy tính thời nguyên thủy với không có âm thanh và đồ họa kém. Nhiều người cho rằng có game mới cần đến DirectX. Thật ra phần lớn là game, nhưng còn rất nhiều chương trình đồ họa, dựng phim v.v.. cũng phải cần phiên bản DirectX thích hợp mới có thể hoạt động tron tru được.

1.3. Phiên bản hiện tại

Phiên bản DirectX hiện tại của tôi là bao nhiều và tại sao phải cập nhật DirectX lên phiên bản mới? Từ Windows XP trở đi thì Microsoft đã cài sẵn trong hệ điều hành 1 phiên bản DirectX. Với Windows 8 mới nhất bây giờ thì có bản DirectX 11. Thường thì những bản DirectX càng về sau càng tốt hơn các bản trước và bao gồm tất cả các chức năng của các bản trước.

Có rất nhiều game từ thời xa xưa, khi mà nhà sản xuất game chưa lường trước được sự phát triển của DirectX. Họ nghĩ rằng DirectX 9.0 là quá hoàn hảo rồi, sẽ không cần và không có bản nâng cấp nữa nên khi lập trình họ mặc định cài Game phải yêu cầu DirectX 9.0, điều này gây nên sự khá hài hước là nếu bạn cài các game cũ loại này vào các máy tính đời mới chạy windows 8 với DirectX 11 thì game sẽ báo là phiên bản DirectX không phù hợp, yêu cầu cài lại Direct 9 ^^. Tuy nhiên nếu gặp trường hợp này thì bạn hoàn toàn yên tâm bỏ qua bước cài DirectX đi nhé. Vì như mình đã đề cập là phiên bản DirectX càng về sau càng chuẩn hơn và bao gồm chức năng bản trước đó nên các game trước đây hoàn toàn có thể chạy tron tru với các bản Direct sau này.

CHƯƠNG 2: GIỚI THIỆU GAME CASTLE VANIA

2.1. Tổng quan Game Castle Vania

Bắt đầu với một tựa game hành động platformer từ 1986, Castlevania đã trải qua một lịch sử hơn 20 để có được hình thái và vị thế như ngày nay.

Castlevania là một trong những tựa game phiêu lưu hành động kinh điển của hệ máy NES trước đây. Ra đời vào năm 1986, cuộc phiêu lưu của chàng hiệp sĩ Simon Belmont trong tòa lâu đài của Dracula đã cuốn hút biết bao thế hệ người chơi. Sở hữu một vũ khí chính là sợi dây xích "Vampire Killer" và rất nhiều vũ khí phụ khác như dao găm, nước thánh, rìu, cùng với rất nhiều đồ vật ẩn, nhiều bí mật trong mỗi màn chơi, Castlevania gần như đã trở thành một seri game mẫu mực trong các tựa game đi cảnh side-scrolling.

Vào năm 2010, Castlevania đã được khoác lên mình một diện mạo mới đẹp hơn nhiều lần với tên gọi: Castlevania: Lords of Shadow. Phát triển bởi hãng MecurySteam cùng Kojima Productions và phát hành bởi Konami, tựa game đã đạt được thành công rất lớn về doanh thu cũng như nhận được những lời phê bình khá tích cực từ cộng đồng game thủ. Một điểm đáng tiếc của trò chơi đó là chỉ phát hành trên Xbox 360 và PS3. Và đến tháng 8/2015, game thủ PC sẽ được trải nghiệm trò chơi này thông qua hệ thống Steam.

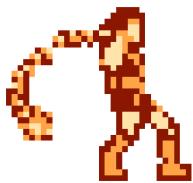
Không tiếp tục phát triển cốt truyện của series Castlevania truyền thống, Lords of Shadow đã có khá nhiều thay đổi. Game thủ sẽ vào vai Gabriel Belmont. Anh là một trong những hiệp sĩ của tổ chức Brotherhood of Light, những người gìn giữ hòa bình cho thế giới khỏi ác quỷ. Mọi chuyện bắt đầu khi vợ của Gabriel bị hại nhưng cô không chết mà linh hồn vẫn kẹt ở thế giới song song, giúp đỡ Belmont trong trận chiến chống lại ác quỷ, mang lại cân bằng cho thế giới.

Một điểm tương đồng với các bản Castlevania cũ là vũ khí của nhân vật chính, giờ đây mang tên Combat Cross. Đó sẽ là điểm nhấn của game, một vũ khí không chỉ dùng để chiến đấu mà còn sử dụng trong di chuyển, mở đường, giải đố và khai phá các bí mật trong game.

Không chỉ kế thừa những đặc điểm đã làm nên thành công của các phiên bản console, PC còn có thêm phần mở rộng trong cốt truyện đồng thời đi kèm nền tảng đồ họa tiên tiến, vũ khí mới và các kĩ năng mới, Combat Cross sẽ trở nên đa chức năng hơn trong cuộc phiêu lưu, thêm nhiều trận chiến với những vị Titan khổng lồ và hơn hết, bổ sung nhiều điều để khám phá trong thế giới của game: những tàn tích, những lâu đài tăm tối, những cánh rừng hay những khu đất đóng băng...

2.2. Giới thiệu Player - Simon

2.2.1. Simon



Xuyên suốt game, người chơi sẽ điều khiển Simon Belmont. Castlevania là một cuộc phiêu lưu ngắn (kỉ lục về tốc độ qua màn của trò chơi là 15 phút) của Simon Belmont trong lâu đài, vượt qua các đày tớ của Dracula và cuối cùng kết liễu hắn để bảo vệ ngôi làng của mình. Simon sở hữu một vũ khí chính là sợi dây xích "Vampire Killer" và rất nhiều vũ khí phụ khác như dao găm, nước thánh, rìu, cùng với rất nhiều đồ vật ẩn, nhiều bí mật trong mỗi màn chơi.

2.2.2. Hệ thống vũ khí

Hình ảnh	Tên	Mô tả
\	Dagger	Dagger là vũ khí cơ bản nhất. Nó bay nhanh, thẳng và chỉ gây sát thương cho một mục tiêu ở phía trước Simon.
*	Throwing Axe	Throwing Axe là một vũ khí mạnh mẽ nhưng khó sử dụng. Nó bay cao lên và từ từ rót xuống theo hình vòng cung. Nó không thích hợp tấn công kẻ thù trước mặt nhưng rất tuyệt để tấn công kẻ thù trên cao.
<u></u>	Holy Water	Holy Water là một vũ khí yếu nếu không biết cách sử dụng. Khi Holy Water bị phá vỡ trên trên gạch, nó sẽ tạo ra một ngọn lửa như một quả boom nhỏ. Holy Water có thể dùng để tấn công trực tiếp nhưng nếu dùng ngọn lửa để bẫy kẻ thù di chuyển chậm như Boss, Axe Kinight thì rất hiệu quả.
+	Boomerang	Boomerang là một vũ khí mạnh như Throwing Axe. Boomerang có thể tấn công nhiều kẻ thù với một cú ném và không chỉ thế nó còn có thể quay ngược lại. Nếu Simon nhảy qua các Boomerang, nó còn có thể tấn công kẻ thù phía sau.



Stopwatch

Stopwatch là một vũ khí phụ bất thường. Nó là vũ khí duy nhất đòi hỏi 5 trái tim trong một lần sử dụng. Nó cũng không phải vũ khí trực tiếp đánh kẻ thù. Thay vào đó nó đóng băng kẻ thù, di chuyển qua bẫy an toàn trong 5 giây. Tuy nhiên Stopwatch có 4 hạn chế sau: Boss miễn dịch, không đóng băng được mục tiêu chống lạnh, Simon vẫn nhận thiệt hại khi va chạm với kẻ thù bị đóng băng, và Simon vẫn chết khi roi xuống hố.

2.3. Giới thiệu hệ thống Enemy, Boss

2.3.1. Enemy

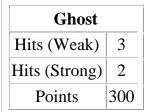
Enemy	Hình thức tấn công	Cách tiêu diệt
Bat Hits (Weak) 1 Hits (Strong) 1 Points 200	Bat bám vào trần nhà và khi Simon lại gần nó sẽ bay sà xuống Simon.	Nên cẩn thận hoặc có thể ném 1 Axe vào nó trước khi nó di chuyển.
Bone Tower Hits (Weak) 8 Hits (Strong) 6 Points 400	Bone tower bất động nhưng trong khoảng thời gian nhất định, nó sẽ bắn hai quả cầu lửa vào Simon. Bone tower sẽ chớp đỏ trước khi bắn.	



Fleaman		
Hits (Weak)	1	
Hits (Strong)	1	
Points	500	

Fleamen xuất hiện di chuyển khá ngẫu nhiên. Mô hình tấn công điển hình của Fleamen là hai bước nhảy ngắn theo sau là một bước nhảy vọt lớn.

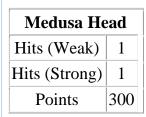
Cho Simon chạy dưới tầm cao và đánh vào Fleamen ngay khi Fleamen chạm đất. Chú ý đừng để Fleamen tới gần, bao quanh Simon.





Ghost xuất hiện từ đẳng sau Simon, không di chuyển nhanh, nhưng Ghost cố gắng di chuyển xung quanh ngay vị trí của Simon từ nhiều góc khác nhau.

Cần ít nhất 2 cú đánh whip để tiêu diệt Ghost hoặc di chuyển nhanh ra khỏi đường đi của chúng.





MedusaHead di chuyển theo đường hình cung, xuất hiện cách nhanh 1 khoảng thời gian nhất định. Các MedusaHead sẽ xuất hiện từ 2 phía màn hình theo hướng đối diện với Simon.

Có thể ngắm đường bay của chúng để né và canh góc đánh của Whip để tiêu diệt chúng. Chú ý hạn chế đổi chiều di chuyển Simon liên tục vì MedusaHead sẽ bay ra từ 2 phía màn hình

Spear Knight		
Hits (Weak)	3	
Hits (Strong)	2	
Points	400	



Spear Knights đi lang thang xung quanh qua lại. Các Spear Knights có thể quay lại về phía Simon bất cứ lúc nào.

Tránh tiếp cận chúng từ cự ly gần trong khi nó đi xa, đặc biệt là sau khi Simon nhảy xuống hoặc nhảy lên. Vì các SpearKnight di chuyển chậm nên có thể dùng HolyWater hoặc Boomerangs để tiêu diệt chúng.



Raven tương đối yếu, nhưng nguy hiểm do tốc độ. Khi Simon lại gần, Raven sẽ cất cánh và tạm Có thể tiêu diệt Raven từ 1 khoảng cách nhất định bằng Axe hoặc Dagger.

Raven	
Hits (Weak)	1
Hits (Strong)	1
Points	200

dừng một thời gian ngắn trước và lao thẳng vào hướng của Simon và chết hoặc chúng sẽ nhanh chóng chuyển hướng tông vào Simon từ đằng sau.



SkeletonHits (Weak)1Hits (Strong)1Points300

Skeletons di chuyển nhanh, nhảy xung quanh Simon rất nhiều, và tấn công bằng cách tung các khung xương vào Simon theo hình vòng cung. Skeleton thường ném từ 1 – 3 khung xương cùng lúc vào Simon.

Tuy nhiên cũng khá dễ để tiêu diệt chúng bằng các vũ khí mà Simon có trừ khi chúng ở phía trên Simon và không thể lại gần.

2.3.2. Boss

Boss	Hình thức tấn công	Cách tiêu diệt
Medusa Hits (Weak) 12 Hits (Strong) 8	mà Stopwatch có thể chống lại. Medusa sẽ thoát ra bên ngoài bức tượng ở trung tâm của căn phòng (không có cảnh	Medusa xuất hiện. Khi Medusa đến, bắt đầu ném vũ khí phụ của Simon. Nếu Simon có Double or Triple Shot, Medusa sẽ bị tiêu diệt một cách nhanh chóng vì Medusa sẽ không thể di chuyển nhiều. Có thể sử dụng Stop Watch đóng băng Medusa tại chỗ và sau đó



Mummy Men		
Hits (Weak)	48	
Hits (Strong)	32	
Points	3000	

Mummy Men sẽ bắt đầu di chuyển khi Simon đi vào phạm khi hoạt động của nó. Mummy Men rất nguy hiểm vì nó có hai con, tấn công Simon từ hai phía, chúng luôn giữ khoảng cách với Simon và tấn công bằng bandage của chúng. Bandage từ Mummy Men có thể bị tiêu diệt từ 1 đòn đánh duy nhất.

Simon nên tìm cách để các Munmy Men tâp trung về một phía. Khi Mummy Men đã tập trung lại, bắt đầu ném vũ khí phụ hoặc Whip của Simon để tấn công đồng thời cả Mummy hai Men. Boomerang, Holy Water là những vũ khí hữu dụng để tiêu diệt chúng. Nếu không có sẵn vũ khí, sử dụng roi và nhảy đứng trên những khối gach cao hơn để tấn công khi chúng đến gần. Hãy cố gắng để các Mummy Men di chuyển trùng khớp vào nhau và tấn công cùng lúc.

2.4. Giới thiệu hệ thống item

2.4.1. Các item cơ bản

Hình ảnh	Tên	Mô tả
4	Whip Upgrade	Roi của Simon có thể nâng cấp hai lần; lần đầu sẽ biến roi thành một ngôi sao sáng, và tăng sát thương, lần hai sẽ tăng chiều dài chiếc roi. Whip Upgrade sẽ xuất hiện trong nến tại đầu mỗi stage. Roi sẽ trở lại bình thường khi Simon chết.
•	Small Heart	Hearts như là số đạn trong một khẩu súng, nó cho phép Simon sử dụng vũ khí tầm xa của
•	Large Heart	mình. Small Heart có giá trị một heart, Large Heart có giá trị hai heart. Và khi kết thúc mỗi màn chơi điểm thưởng sẽ được tăng theo số heart còn xót lại.

4	Red Money Bag	Money Bag sẽ xuất hiện khắp lâu đài trong nế Mặc dù Simon không cần tiền nhưng thu thậ
	Purple Money Bag	mạc du Simon không can tiên nhưng thủ thập túi tiền sẽ tăng điểm số. + Red bag: 100 points + Purple bag: 400 points
	White Money Bag	+ White bag: 700 points

2.4.2. Các item đặc biệt

Hình ảnh	Tên	Mô tả		
	Cross	Nếu Simon nhặt Cross, mọi kẻ thù trong màn hình sẽ bị tiêu diệt ngay lập tức dù còn bao nhiêu máu. Cross có thể được tìm thấy trong nến. Nó không hoạt động với Boss mặc dù sẽ phá hủy tất cả các vật do Boss phóng ra. Ngoài ra sẽ không nhận được điểm khi kẻ thù chết bởi Cross.		
30	Nếu Simon uống được Invisibility Potion, Sin sẽ có thể tàn hình tuyệt đối với bật kỳ kẻ thù cặm bẫy nào mà không nhận sát thương. nhiên Simon vẫn có thể chết khi rớt xuống hố			
~	Pork Chop	Pot Roast/Pork Chop là một item cực kỳ giá trị Nó sẽ khôi phục cho Simon 6 đơn vị trong than sức sống của Simon. Nó chỉ có thể tìm thấy trong các khối gạch khác thường.		
	Double Shot Thông thường chỉ có thể có một vũ khí phụ màn hình cùng một lúc. Với Double Sho			

	Triple Shot	Triple Shot, Simon có thể ném hai hoặc ba vũ khí tại một thời điểm. Shot multipliers có thể được tìm thấy xung quanh lâu đài trong các khối gạch khác thường. Nó nhặt một vũ khí phụ khác số nhân sẽ biến mất. Stopwatch không hoạt động với vật phẩm nhân này nhưng nó sẽ cho 700 điểm.
	Bonuses	Bonuses là một vật phẩm bí mật, có thể tìm thấy xung quanh lâu đài bằng cách kích hoạt các địa điểm bí mật. Bonuses có giá trị 1000 điểm thưởng.
0	Crystal Ball	Một Crystal Ball sẽ xuất hiện bất cứ khi nào Simon đánh bại một con boss. Crystal Ball là dấu hiệu cho thấy một trong 6 giai đoạn của lâu đài được hoàn thành. Để chống lại Dracula, Simon phải tích lũy 5 Crystal Ball bằng cách đánh bại 5 con boss ở 5 vị trí khác nhau trong lâu đài.

CHƯƠNG 3: ĐỒ ÁN GAME CASTLE VANIA

3.1. Các kỹ thuật áp dụng trong game

3.1.1. 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.

3.1.2. Thuật toán xử lý va chạm SweptAABB

- Thuật toán SweptAABB giải quyết được vấn đề mà thuật toán AABB không kiểm soát được. Vấn đề nãy sinh ở thuật toán AABB là khi đối tượng di chuyển quá nhanh và dường như sau 1 frame di chuyển hoàn toàn không thể kiểm tra được các đối tượng có va chạm hay không. Để giải quyết vấn đề ta cần xác định khoảng di chuyển của các đối tượng qua thời gian.
- Phương thức kiểm tra SweptAABB được mô tả như sau:

float sweptAABB(Box * M, RectF * S, int & normalx, int & normaly);

- Trong đó:
 - + Box: là đối tượng di chuyển.
 - + RectF: là đối tượng tĩnh xét va chạm.
 - + normalx, normaly là giá trị để xác định kết quả phản ứng sau quá trình va chạm.
- Giá trị trả về của phương thức nằm trong khoảng 0 đến 1. Nếu giá trị là 1 nghĩa là không xảy ra va chạm. Nếu giá trị trả về là < 1 cho biết có xảy r ava chạm trong frame tiếp theo được vẽ.
- 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.

3.1.3. Transformation

- Transformation là phương pháp dùng để thực hiện tất cả biến đổi hình học
- Ví dụ ma trận dùng để xoay hình 180 theo trục x như sau:

-1	$0 \ 0 \ 0$
0	100
0	$0 \ 1 \ 0$
2 * (x + width / 2)	001

3.1.4. Camera

- Được kế thừa từ lớp BaseObject, camera game là một khung nhìn hình chữ nhật trong map xác định các đối tượng được hiển thị trong camera thật, được di chuyển sao cho luôn nằm đúng với vị trí của camera thật.
- Do tọa độ gốc của camera thật luôn nằm ở tọa độ (0, 0) nên để tọa độ của camera game trùng với tọa độ của camera thật ta cần dời tọa độ gốc của map thành giá trị âm.
- Giới hạn di chuyển của camera phụ thuộc vào kích thước của mỗi stage.
- Kích thước camera thật bằng với kích thước của backbuffer nên camera trong game cũng được thiết kế với kích thước tương ứng để khi vẽ đối tượng lên camera game sẽ không cần zoom đối tượng cho vừa với kích thước camera thật. Và khi vẽ đối tượng từ backbuffer lên framebuffer chỉ cần zoom đúng tỉ lệ màn hình với tỉ lệ hình vẽ duy nhất một lần.
- Cách hoạt động: Mỗi khi di chuyển camera cần xác định vị trí của camera so với giới hạn của stage. Nếu giới hạn bên trái của camera nhỏ hơn của stage và camera đang có hướng dịch chuyển sang trái thì xác định lại giới hạn bên trái của camera bằng của stage và cho camera dừng chuyển động, ngược lại đối với bên phải. Nếu Simon đang nằm ở nửa bên trái của camera và có chiều đi sang phải thì dừng camera đến khi Simon di chuyển đến giữa camera, ngược lại với bên phải.

3.1.5. Stage, level

- Stage và camera là các đối tượng trong map nên ta tạm định nghĩa map một một cách cơ bản như sau: Map chứa tất cả đối tượng trong game bao gồm cả camera và stage.

- Stage:

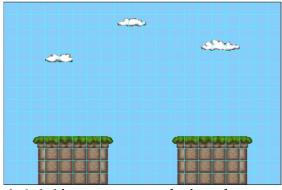
- + Stage được kế thừa từ lớp RectF là một hình chữ nhật bao quanh một vùng trong map.
- + Mỗi lần đi lên cầu thang và có dịch chuyển map lên phía trên hoặc đi qua cửa thì sẽ chuyển sang stage tiếp theo.
- + Khi chuyển stage tọa độ của map và camera sẽ được di chuyển tương ứng cho phù hợp để vào đúng tầm hoạt động của stage đó.
- + Mỗi stage có tọa độ và kích thước riêng của nó dùng để giới hạn tầm hoạt động của camera.

- Level:

- + Mỗi level trong game là một map, chứa danh sách các đối tượng, level và stage hoàn toàn khác nhau.
- + Ta có thể đến level tiếp theo bằng cách nhặt Crystal Ball mỗi khi tiêu diệt một con Boss trong level đó.
- + Mỗi khi chuyển level toàn bộ số liệu trong level cũ sẽ bị xóa đi và khởi tạo lại một level mới hoàn toàn. Ngoài ra những đối tượng toàn cục trong game như Simon sẽ được giữ nguyên dữ liệu.

3.1.6. Tile map

- Tile là khái niệm chỉ đến một ảnh có kích thước nhỏ (thường là ảnh bitmap), có dạng hình vuông hoặc hình chữ nhật.
- Độ dài các cạnh của tile được tính theo pixel thường là một ước số của khung nhìn (cửa số màn hình).
- Khái niệm định nghĩa một cách tương đối, nói riêng trên lĩnh vực đồ hoạ thiết kế... Thực chất khái niệm tiles chúng ta sẽ gặp rất nhiều ngoài đời sống ví dụ như các thứ gần gũi với chúng ta như là sàn nhà, tường gạch... Không chỉ có hình chữ nhật hay hình vuông mà tiles có rất nhiều hình dạng khác nhau. Tuy nhiên trong design chúng ta thiết kế các tiles có hình chữ nhật hoặc hình vuông để tiện thao tác khi xử lí logic.





- Các hình ảnh lớn trong game thường được tạo nên từ các tile nhỏ. Nơi chứa những tile được dùng để tạo nên các map được gọi là "Tileset".
- Ưu điểm khi sử dụng Tileset:
 - + Việc chia nhỏ các hình ảnh lớn thành các tile nhỏ hơn sẽ thuận lợi cho việc lưu trữ và linh hoạt trong việc tạo ra các kiểu hình ảnh đồ họa trong game, sẽ giúp tiết kiệm bộ nhớ rất nhiều và tốc độ xử lí được cải thiện
 - + Việc thiết kế map là một công việc tốn khá nhiều thời gian nhưng nếu sử dụng các tile trong tileset có thể tiết kiệm thời gian trong việc thiết kế cũng như dễ nâng cấp, sửa chữa

- Tile map dùng để xây dựng map cho game.
- Khi sử dụng tile map trong game ta chỉ cần load những tile sheet trong vùng camera tương ứng với id trong ma trận đã có được khi sử dụng tile map giúp làm nhẹ quá trình load hình ảnh, dễ dàng tạo map mới.
- Các bước tạo và ma trận và cắt tile từ hình map có sẵn
 - + Load hình ảnh map
 - + Cắt ra các tile khác nhau với mỗi tile là một id và lưu vào một tấm hình khác.
 - + Tạo ma trận lưu id tương ứng với mỗi tile trong map.

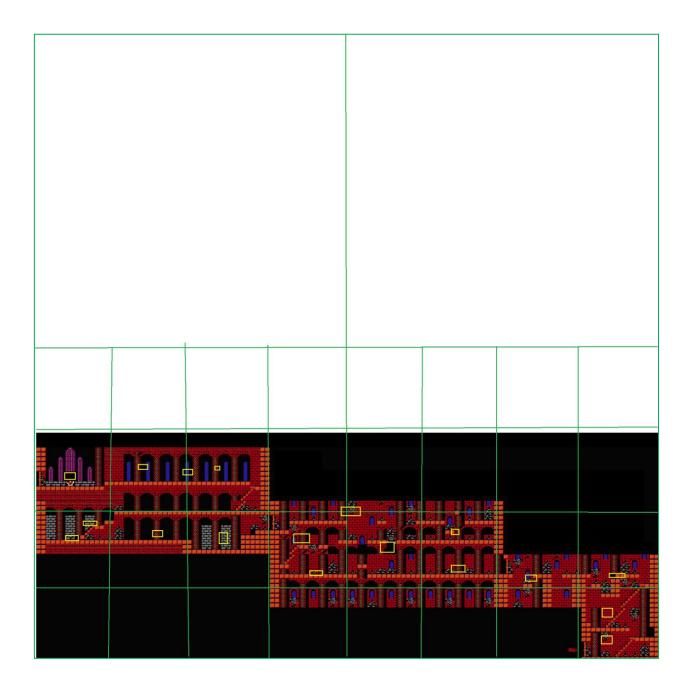
3.1.7. *Map editor*

- Để xây dụng tool map editor trước tiên ta cần sử dụng tool tile map để tạo ra ma trận id các tile trong game và tile sheet mà game sử dụng tương ứng với ma trận đó.
- Sử dụng ma trận và tile sheet đã có để load thành background của map.
- Load hình ảnh của các đối tượng vào tool, thêm id tương ứng với id được sử dụng trong source game và chọn vị trí thích hợp cần thêm đối tượng vào map để thêm đối tượng đó vào danh sách các đối tượng trong map.
- Đối với đối tượng gạch có sẵn trong background ta chỉ cần sử dụng ma trận và tile sheet đã có sẵn để xác định vị trí trong map và thêm vào danh sách đối tượng. Nhưng trước khi thêm vào danh sách đối tượng ta cần gom các viên gạch này thành nhóm để dễ dàng xử lý va chạm trong game. Trước tiên gom tất cả viên gạch cùng hàng và nằm cạnh nhau lại với nhau, tiếp theo nếu các khối gạch đã gom ở bước trước có cùng chiều dài và nằm cạnh nhau theo chiều dọc thì tiếp tục gom lại với nhau.

3.1.8. 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. Quadtree là một giải thuật đơn giản và hiệu quả giúp giảm thiểu chi phí dùng cho việc xét va chạm giữa các đối tượng trong game.
- Các bước trong thuật toán:
- + Xác định kích thước vùng phân hoạch. Từ map hình chữ nhật, kiểm tra xem chiều ngang hay chiều dọc lớn hơn để tạo thành một vùng phân hoạch hình vuông từ cạnh lớn hơn trong map. Ta sử dụng vùng phân hoạch hình vuông để các vùng phân hoạch con cũng hình vuông, thuận lợi cho việc phân chia đối tượng cho các vùng phân hoạch.
- + Từ node gốc ta chia thành 4 node con
- + Nếu các node con có kích thước > 176 và < 352 (tùy theo quy định cho phù hợp với màn hình game) hoặc không có đối tượng thì dừng phân vùng. Ngược lại ta tiếp tục chia các node thành 4 vùng con nhỏ hơn bằng nhau.

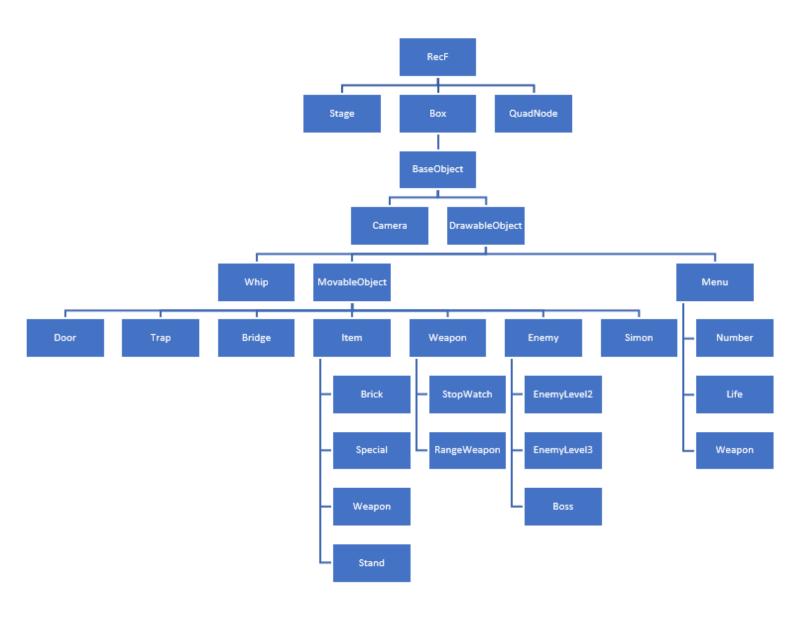
+ Sau khi chia các node xong ta xét va chạm giữa các đối tượng với các node. Nếu nó va chạm với node nào thì thêm đối tượng đó vào node đó. Nếu đối tượng nằm trong đồng thời hai node hoặc nhiều hơn thì thêm đối tượng đó vào đồng thời tất cả node đó. Xóa đối tượng trong các node không phải node lá là ta đã chia tất cả các đối tượng ra các node thành công.



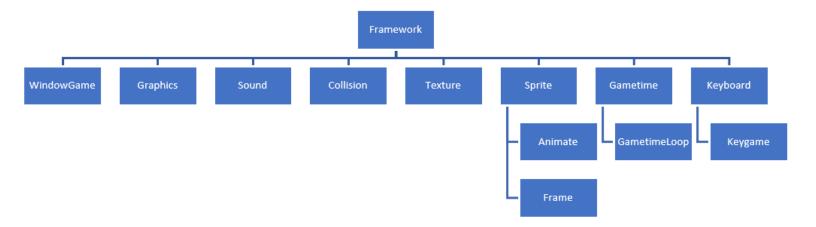
Ví dụ mô phỏng cách chia Quadtree

3.2. Sơ đồ các class chính trong game

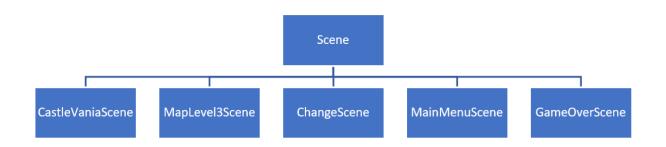
3.2.1. Sơ đồ mô phỏng các class object có trong game Castle Vania



3.2.2. Sơ đồ mô phỏng các class framework



3.2.2. Sơ đồ mô phỏng các class màn hình



3.3. Công cụ hỗ trợ

- Để phục vụ cho việc thực hiện đồ án, nhóm đã sử dụng các công cụ hỗ trợ như:
 - + Visual Studio 2015
 - + DirectX 9.0
 - + Microsoft Office 2016
 - + Microsoft Paint
 - + NotePad++
 - + Git

3.4. Kết quả thực hiện

3.4.1. Đánh giá mức độ hoàn thiện của Game

- Đánh giá mức độ hoàn thiện của Game: 90%
- Về cơ bản, nhóm chúng em cũng đã xây dựng nên một game tương đối hoàn chỉnh đáp ứng đủ các yêu cầu mà Thầy, Cô đã đặt ra:
 - + Xây dựng game Castlevania, có đầy đủ các thành phần âm thanh, hình ảnh, điều khiển bàn phím...
 - + Hoàn thành 2 level trong game là level 2 và level 3. 1 level bao gồm 3 stage liên tục nhau: level 2 gồm các stage từ 4 đến 6, level 3 gồm các stage từ 7 đến 9.
 - + Áp dụng các kĩ thuật trong game: AABB, Swept AABB, transformation, tilemap, quadtree.
 - + Xây dựng Map Editor.
- Game đã ra đời từ trước và nhóm chỉ thực hiện lại không mang tính thương mại hoá nên phần cốt truyện và đồ hoạ nhóm sử dụng resource sẵn có.
- Game đã hoàn thành đúng tiến độ mà nhóm đã đặt ra ban đầu. Nhóm cũng đã dành thời gian để tìm và fix lỗi, cho một số các bạn khác chơi thử và góp ý, đánh giá và nhận được phản hồi khá tích cực.

3.4.2. Hướng phát triển (nếu có)

- Dành thêm thời gian để test và fix nếu có xảy ra lỗi tìm ẩn trong game
- Hoàn thiện những thiếu sót trong game
- Hoàn thiện tất các màn chơi
- Phát triển thành game 3D
- Xây dựng thêm các tuỳ chọn cho người chơi có thể tùy chỉnh chế độ chơi

3.4.3. Bảng phân công công việc

Họ tên	MSSV	Mức độ hoàn thành	Công việc được giao	Ký tên
	14520990	100%	Nhóm trưởng, phân công công	
			việc cho các thành viên	
			Xây dựng framework	
			Xây dựng hệ thống Enemy, boss	
Trần Thị			Xây dựng hệ thống class quản lý	
Minh Trang			đối tượng	
			Xây dựng class QuadNode,	
			QuadTree	
			Quản lý class Map, TileMap,	
			ObjectsCategory	
			Xây dựng framework	
	14520288	100%	Xây dựng menu game	
Nguyễn			Xây dựng hệ thống item	
			Xây dựng class	
Minh Hiếu			DoorChangeStage, Bridge	
			Quản lý class Map, TileMap,	
			ObjectsCategory	
			Báo cáo file word	
	14521042	100%	Xây dựng framework	
			Quản lý Simon	
Hoàng Kim Tuấn			Xây dựng class Stair, Trap	
			Xây dựng hệ thống weapon	
			Quản lý class Map, TileMap,	
			ObjectsCategory	
			Xây dựng hệ thống Scene	
			Xây dựng tool Tilemap-	
			Mapeditor	

3.5. Kết luận

Nhập môn phát triển game được coi là một môn học tương đối khó với mỗi sinh viên học khoa Công nghệ phần mềm. Tuy nhiên, nhờ sự giảng dạy nhiệt tình từ Thầy Nguyễn Vĩnh Kha và Cô Huỳnh Hồ Thị Mộng Trinh với những bài học hữu ích trên lớp đã giúp nhóm biết cách làm ra một game là như thế nào và tự làm thành một game hoàn chỉnh. Các kỹ thuật trong game như: xây dựng một tile -map bằng các tilemap engine, kỹ thuật xử lý va chạm AABB, kỹ thuật phân hoạch không gian quadtree,...

Các thành viên trong nhóm khá gắn kết và có trách nhiệm. Ý thức được đây là một môn học khó nên các thành viên đều đi học đầy đủ nghe các bài giảng trên lớp, ngoài ra còn chủ động tìm hiểu, học hỏi các kiến thức liên quan, tham khảo các game Mario, Contra từ các anh chỉ khóa trên,... nhờ đó có cơ hội được tiếp cận nhiều nguồn tài liệu phong phú, đa dạng khá hữu ích. Mỗi thành viên luôn cố gắng hoàn thành đầy đủ các nhiệm vụ được nhóm trưởng phân công và nhiệt tình giúp đỡ, chia sẻ với nhau. Vì thế, nhóm đã có thể hoàn thành đồ án đúng hạn và đạt được mục tiêu đề ra.

Ngoài những kiến thức môn học cung cấp, quá trình làm việc nhóm đã giúp các thành viên thêm gắn kết, nâng cao khả năng quản lý thời gian và kỹ năng làm việc nhóm hiệu quả. Dù môn học có phần áp lực nhưng rất thú vị và là nền tảng để sinh viên định hướng theo chuyên ngành Môi trường ảo và phát triển game sau này.

TÀI LIỆU THAM KHẢO

- [1]. Giáo trình Nhập môn phát triển game PSG.TS. Vũ Thanh Nguyên và ThS. Đinh Nguyễn Anh Dũng biên soạn
- [2]. Programming 2D Games Charles Kelly
- [3]. http://strategywiki.org/wiki/Castlevania
- [4]. https://www.stdio.vn/articles/read/109/hien-thuc-quadtree-va-ung-dung-trong-lap-trinh-game
- [5]. https://msdn.microsoft.com/en-us/library/windows/desktop/bb206269(v=vs.85).aspx