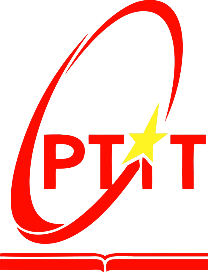
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI TP. HỒ CHÍ MINH**



**BÁO CÁO CUỐI KÌ**

**THỰC TẬP CHUYÊN SÂU**

**PHÁT TRIỂN XÂY DỰNG GAME BOMBER**

***MSSV: N19DCPT064***

Sinh viên thực hiện: Nguyễn Văn Tình Nam / ~~Nữ~~

Dân tộc: Kinh

Lớp: D19CQPU01-N – Khoa Công nghệ Thông tin II

Ngành: Công nghệ Đa phương tiện

Người hướng dẫn: Th.S Nguyễn Văn Sáu

**TP.HCM, 05/2023**

**CHƯƠNG I: DESIGN DOCUMENT**

1. **Game: Boomber**
2. **Brief description:**

Boomber là game hành động góc nhìn từ trên xuống, trong đó người chơi điều khiển một nhân vật và di chuyển qua các mê cung để tìm cách phá hủy các chướng ngại vật và tiêu diệt quái vật. Trò chơi sẽ được phát triển bằng Python và dùng thư viện Pygame.

1. **Target Audience:**

Đối tượng mục tiêu của Boomber là những game thủ thích trò chơi hành động với lối chơi thử thách và phong cách thẩm mỹ theo phong cách cổ điển. Cụ thể, trò chơi hướng đến những người:

* Đang tìm kiếm trải nghiệm chơi trò chơi đầy thử thách đòi hỏi chiến lược và kỹ năng
* Thích các trò chơi có đồ họa nghệ thuật pixel và tính thẩm mỹ lấy cảm hứng từ cổ điển
* Thích khám phá các cấp độ và khám phá bí mật
* Có hứng thú với bối cảnh giả tưởng với quái vật và ma thuật
* Đang tìm kiếm một trò chơi mang lại cảm giác hoàn thành và phần thưởng khi vượt qua các thử thách
* Nhìn chung, Boomber nhắm đến những người chơi đang tìm kiếm trải nghiệm chơi trò chơi thú vị và hấp dẫn kết hợp hành động nhịp độ nhanh với tư duy chiến lược và khám phá. Cơ chế chơi trò chơi đầy thử thách và thẩm mỹ lấy cảm hứng từ cổ điển của trò chơi được thiết kế để thu hút những game thủ đánh giá cao các trò chơi kiểu thùng cổ điển pha chút hiện đại.

1. **Gameplay:**

Boomber là trò chơi một người chơi trong đó người chơi điều khiển một nhân vật bằng các phím mũi tên trên bàn phím và dùng phím Space để đặt bom. Nhân vật có thể di chuyển theo cả bốn hướng, và có một lượng bom hạn chế để tiêu diệt quái vật và chướng ngại vật. Trò chơi sẽ có một loạt mê cung để người chơi di chuyển qua, với mỗi mê cung chứa những con quái vật phải bị tiêu diệt bằng bom. Người chơi phải tìm phá hủy các chướng ngại vật để tiếp cận quái vật và tiêu diệt chúng.

Trò chơi sẽ có một loạt mê cung để người chơi di chuyển qua, với mỗi mê cung chứa những con quái vật phải bị tiêu diệt bằng bom. Người chơi phải đặt bom một cách chiến lược để dọn đường và phá hủy các chướng ngại vật để tiếp cận lũ quái vật và tiêu diệt chúng.

Các cấp độ giống như mê cung của trò chơi được thiết kế để trở nên đầy thử thách và hấp dẫn, với nhiều con đường để khám phá và những bí mật cần khám phá. Các cấp độ sẽ có nhiều mối nguy hiểm và kẻ thù, bao gồm bẫy, gai và sinh vật của kẻ thù di chuyển theo các kiểu khác nhau và yêu cầu các chiến lược khác nhau để đánh bại.

1. **Design Level:**

Các cấp độ của Boomber được thiết kế để hấp dẫn, thử thách và bổ ích cho người chơi:

* Nhiều con đường: Mỗi cấp độ có nhiều con đường mà người chơi có thể đi để đạt được mục tiêu của mình. Điều này cho phép người chơi khám phá và tìm thấy các phần thưởng và sức mạnh tiềm ẩn.
* Chướng ngại vật: Các cấp độ chứa đầy chướng ngại vật mà người chơi phải vượt qua bằng cách sử dụng bom. Những chướng ngại vật này bao gồm tường, đá và các chướng ngại vật khác cản đường người chơi
* Kẻ thù: Các cấp độ chứa đầy quái vật phải bị đánh bại bằng bom. Những con quái vật này di chuyển theo các kiểu khác nhau và yêu cầu các chiến lược khác nhau để đánh bại.
* Tăng sức mạnh: Ẩn trong các cấp độ là những vật phẩm tăng sức mạnh có thể được sử dụng để tăng sức khỏe, bom và các thuộc tính khác của nhân vật người chơi. Những lần tăng sức mạnh này rất cần thiết để người chơi tiến bộ trong trò chơi.
* Quái vật trùm: Ở cuối mỗi cấp độ, người chơi phải đối đầu với một con quái vật trùm phải bị đánh bại để lên cấp độ tiếp theo. Những con trùm này được thiết kế để thử thách và yêu cầu người chơi sử dụng tư duy chiến lược và kỹ năng giải câu đố để đánh bại.

1. **Specify the art style:**

Phong cách nghệ thuật của Boomber được thiết kế cổ điển, vui tươi và vui nhộn, tập trung vào việc tạo ra một thế giới trò chơi sống động và năng động để thu hút người chơi. Dưới đây là một số đặc điểm chính của phong cách nghệ thuật:

* Đồ họa 2D: Boomber có đồ họa 2D với phong cách nghệ thuật vẽ tay. Các nhân vật, quái vật và môi trường đều được vẽ bằng màu sắc tươi sáng, rực rỡ và thiết kế vui tươi.
* Top-Down view: Trò chơi được thiết kế với chế độ xem từ trên xuống, cho phép người chơi nhìn toàn bộ thế giới trò chơi và điều hướng qua các mê cung một cách dễ dàng.
* Thiết kế nhân vật: Nhân vật người chơi được thiết kế với phong cách dễ thương và vui tươi, với các tính năng phóng đại và tính cách vui nhộn. Các quái vật được thiết kế đa dạng và độc đáo, với mỗi quái vật có tính cách và kiểu di chuyển riêng.
* Thiết kế môi trường: Môi trường của trò chơi được thiết kế đầy tính cổ điển và vui nhộn, với nhiều bối cảnh và chủ đề khác nhau, bao gồm rừng, hang động và các địa điểm khác.
* Animation: Animation của trò chơi được thiết kế mượt mà và trôi chảy, với các nhân vật và quái vật di chuyển một cách vui tươi và sống động.
* User Interface: Giao diện người dùng của trò chơi được thiết kế đơn giản và trực quan, với các menu và điều khiển rõ ràng, dễ hiểu.

1. **Sound Effect:**

Âm thanh và hiệu ứng âm thanh của Boomber được thiết kế để trở nên đắm chìm và hấp dẫn, tập trung vào việc tạo ra bầu không khí vui vẻ và sống động bổ sung cho phong cách nghệ thuật và lối chơi của trò chơi. Dưới đây là một số tính năng chính của âm thanh và hiệu ứng âm thanh:

* Sound Effect: Trò chơi có nhiều hiệu ứng âm thanh được thiết kế vui nhộn và hấp dẫn, bao gồm âm thanh bom nổ, quái vật bị trúng đạn và thu thập sức mạnh.
* Âm nhạc: Âm nhạc của trò chơi được thiết kế theo hướng lạc quan và vui tươi, với những giai điệu hấp dẫn và nhịp điệu vui nhộn giúp tạo ra một thế giới trò chơi sống động và năng động.
* Điều khiển âm lượng: Trò chơi cho phép người chơi điều chỉnh âm lượng của hiệu ứng âm thanh và âm nhạc, cho phép họ tùy chỉnh trải nghiệm âm thanh theo ý thích.

1. **Features:**

* Lối chơi dựa trên mê cung: Lối chơi dựa trên mê cung của Boomber là tính năng trung tâm của trò chơi, mang đến trải nghiệm độc đáo và đầy thử thách cho người chơi. Người chơi phải điều hướng qua một loạt mê cung để tìm và tiêu diệt quái vật bằng bom, sử dụng chiến lược và kỹ năng để vượt qua các chướng ngại vật và đánh bại trùm.
* Điều khiển nhân vật: Người chơi điều khiển nhân vật bằng các phím mũi tên trên bàn phím và sử dụng phím Space để đặt bom. Nhân vật có thể di chuyển theo cả bốn hướng và có số lượng bom hạn chế để tiêu diệt quái vật và chướng ngại vật.
* Quái vật đa dạng: Trò chơi có nhiều loại quái vật có đặc điểm và hành vi độc đáo. Người chơi phải sử dụng các chiến lược khác nhau để đánh bại từng con quái vật, tạo thêm một lớp thử thách cho lối chơi.
* Tăng sức mạnh: Trò chơi bao gồm các lần tăng sức mạnh mang lại cho người chơi nhiều lợi ích khác nhau, chẳng hạn như tăng phạm vi ném bom, tốc độ và thêm mạng sống. Những vật phẩm tăng sức mạnh này được ẩn giấu trong các mê cung và người chơi phải khám phá ra.
* Chướng ngại vật đầy thách thức: Trò chơi bao gồm những chướng ngại vật đầy thách thức phải vượt qua để vượt qua các mê cung. Những chướng ngại vật này bao gồm tường, đá và các vật thể khác có thể cản đường người chơi.
* Các trận đấu trùm: Trò chơi có các trận đấu trùm mang đến thêm thử thách cho người chơi. Những trận chiến này yêu cầu người chơi sử dụng các kỹ năng và chiến lược của mình để đánh bại một con quái vật mạnh mẽ.
* Nhiều cấp độ: Trò chơi bao gồm nhiều cấp độ khó tăng dần khi người chơi tiến bộ. Mỗi cấp độ giới thiệu những con quái vật, chướng ngại vật và thử thách mới, giúp lối chơi luôn mới mẻ và hấp dẫn.

1. **Technical Requirement:**

Được phát triển bằng Python và Pygame

* Tương thích với Windows và Mac OS
* Yêu cầu hệ thống tối thiểu:

+ Bộ xử lý lõi kép 2GHz

+ RAM 2GB

+ Card đồ họa tương thích DirectX 9.0c

+ Ổ cứng trống 200 MB

1. **Story Board:**

+ Màn hình lựa chọn



+ Màn hình game



+ Màn hình kết thúc



**CHƯƠNG II: GAME DEVELOPMENT**

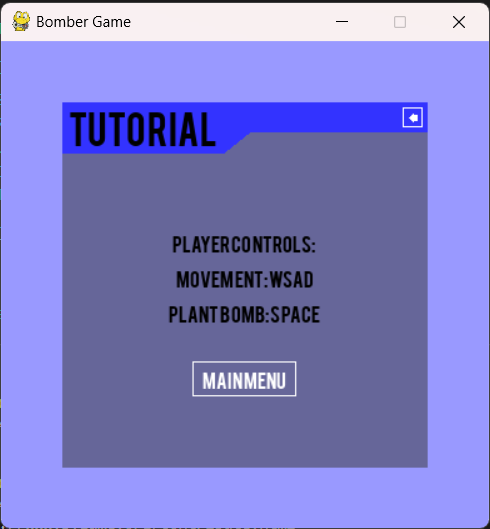
1. **Xây dựng menu:**

Dùng thư viện pygame\_menu để xây dựng bảng chọn cho người dùng, nhằm mục đích cho người dùng dễ thao tác và lựa chọn những level khác nhau của game.

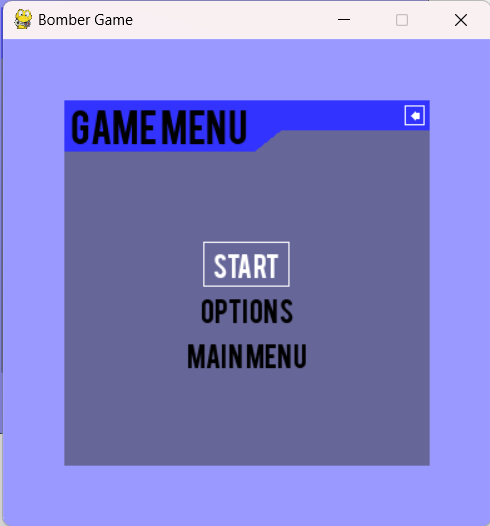
* Màn hình chính:



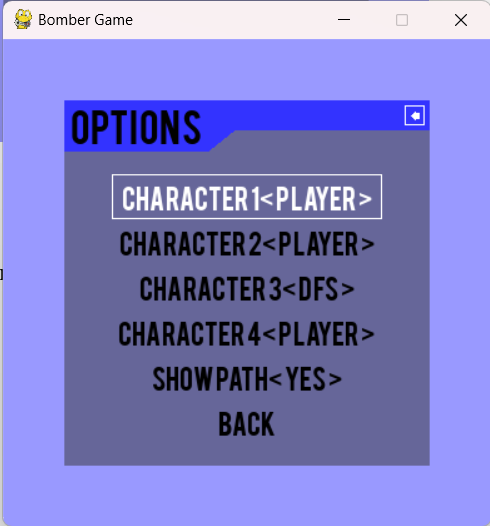
* Màn hình khi click vào tutorial:



* Màn hình khi click vào start:



* Màn hình khi click vào options: (hiện lên những thể loại game đấu mà người dùng tự lựa chọn)



1. **Xậy dựng map:**

Bản đồ sẽ bao gồm một ma trận 13 x 13, với các hàng và cột ở ngoài cùng sẽ được phủ lên đá, và thực chất người chơi và enemy chỉ có thể chơi ở map **12 x 12**.

Bên trong sẽ được mặc định bố trí vài cục đá để làm chướng ngại vật cho người chơi không thể dùng boom phá vỡ.

Tiếp theo sẽ xây dụng thùng gỗ, vật cản duy nhất làm cho người dùng không thể đi qua nhưng vẫn có thể dùng boom để phá vỡ nó. Những thồ gỗ này sẽ được random khi người chơi bấm bắt đầu game, và sẽ không theo một quy tắc nào khiến người chơi được bất ngờ với mỗi game hơn.

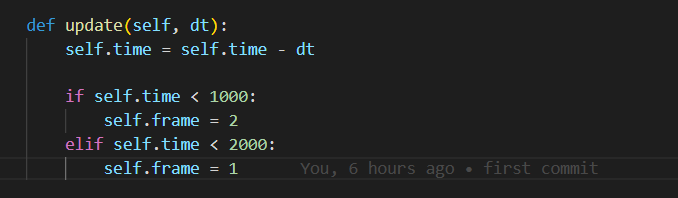
Dùng hàm random từ 0 -> 9 và số random phải bé hơn 7 (xác xuất suất hiện boom là 60%)

1. **Xây dựng đối tượng Bomb:**

Tạo class Bomb và phương thức khởi tạo bao gồm range, vị trí x, vị trí y, map, time (thời gian mà bomb nổ), sectors

Xây dựng hàm get range (lấy danh sách các khu vực nằm trong phạm vi của quả bom.). Chức năng đầu tiên thêm vị trí của quả bom vào danh sách các lĩnh vực. Sau đó, nó đi vòng qua phạm vi của quả bom, kiểm tra từng ô trong bản đồ. Bom sẽ được nổ theo quy tắc lặp từ 1 đến range được khởi tạo theo các hướng trên dưới phải trái. Nếu hướng nào gặp là đá thì sẽ thoát khỏi vòng lặp và không được tạo ra vụ nổ, hướng nào gặp thùng gỗ sẽ thêm vào danh sách vị trí nổ và thoát, ngược lại ô trống thì thêm vào và tiếp tục vòng lặp.

Xây dưng hàm update để cập nhật frame animation của bom.



1. **Xây dựng nhân vật của người chơi:**

Bước đầu xây dựng 1 class đối tượng tên là Player. Khởi tạo vị trí đứng đầu tiên của người chơi mặc định sẽ là (1, 1). Thêm các thuộc tính sau:

+ animation (list): load animation khi khởi động game cho nhân vật, bao gồm các hành động di chuyển trên dưới phải trái

+ pos\_x, pos\_y: vị trí trục hoành và trục tung hiện tại của nhân vật

+ bomb\_limit: 1 (giới hạn được đặt boom của nhân vật)

+ direction, frame (list): nhằm biết người dùng đang đi hướng nào và frame nào cần được load animation

+ range: số ô mà boom có thể nổ.

+ life (boolean): khi người chơi bi boom nổ, life sẽ được set bằng True.

Tiếp theo cần load animation của người chơi khi khởi động game vào biến animation, bao gồm 4 action up, down, right, left. Mỗi action sẽ có 3 frame để load, khi đến frame thứ 3 sẽ tự reset về ban frame đầu tiên

f1 = pygame.image.load('images/hero/pf0.png')

        f2 = pygame.image.load('images/hero/pf1.png')

        f3 = pygame.image.load('images/hero/pf2.png')

        f1 = pygame.transform.scale(f1, (resize\_width, resize\_height))

        f2 = pygame.transform.scale(f2, (resize\_width, resize\_height))

        f3 = pygame.transform.scale(f3, (resize\_width, resize\_height))

        front.append(f1)

        front.append(f2)

        front.append(f3)

        r1 = pygame.image.load('images/hero/pr0.png')

        r2 = pygame.image.load('images/hero/pr1.png')

        r3 = pygame.image.load('images/hero/pr2.png')

        r1 = pygame.transform.scale(r1, (resize\_width, resize\_height))

        r2 = pygame.transform.scale(r2, (resize\_width, resize\_height))

        r3 = pygame.transform.scale(r3, (resize\_width, resize\_height))

        right.append(r1)

        right.append(r2)

        right.append(r3)

        b1 = pygame.image.load('images/hero/pb0.png')

        b2 = pygame.image.load('images/hero/pb1.png')

        b3 = pygame.image.load('images/hero/pb2.png')

        b1 = pygame.transform.scale(b1, (resize\_width, resize\_height))

        b2 = pygame.transform.scale(b2, (resize\_width, resize\_height))

        b3 = pygame.transform.scale(b3, (resize\_width, resize\_height))

        back.append(b1)

        back.append(b2)

        back.append(b3)

        l1 = pygame.image.load('images/hero/pl0.png')

        l2 = pygame.image.load('images/hero/pl1.png')

        l3 = pygame.image.load('images/hero/pl2.png')

        l1 = pygame.transform.scale(l1, (resize\_width, resize\_height))

        l2 = pygame.transform.scale(l2, (resize\_width, resize\_height))

        l3 = pygame.transform.scale(l3, (resize\_width, resize\_height))

        left.append(l1)

        left.append(l2)

        left.append(l3)

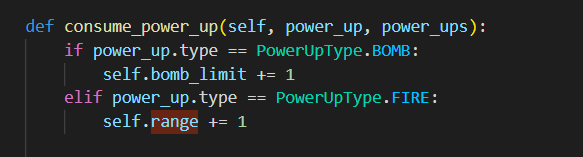
        self.animation.append(front)

        self.animation.append(right)

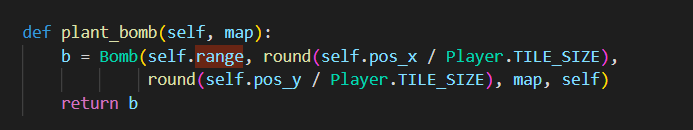
        self.animation.append(back)

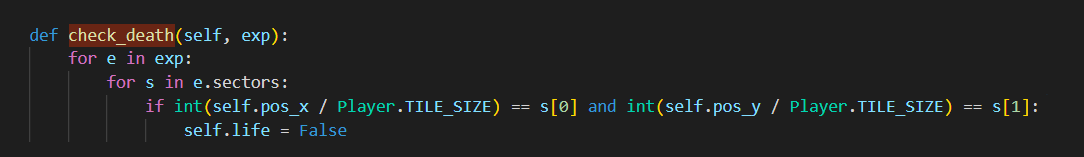
        self.animation.append(left)

Xây dựng hàm xử lý item nhặt được, có 2 loại là BOOM (tăng limit boom được đặt thêm 1) và FIRE (tăng range boom thêm 1)



Xây dựng hàm khi người dùng nhấn nút Space(đặt boom)



Hàm kiểm tra đã trúng bom:  


1. **Xây dựng kẻ thù:**

Tối đa số lượng kẻ thủ trong 1 game đấu là 3, người chơi có quyền giảm hoặc tăng số lượng kẻ thù để thay đổi mức đồ khó của game.

Xây dựng lớp Enemy tương tự với Player, khác là cần phải xây dựng thêm tự động di chuyển của lớp này.

Xây dựng 2 thuật toán khác nhau để enemy không bị di chuyển 1 kiểu, làm người chơi dễ chán và nhận ra.

+ Thuật toán DFS(Deepth First Search): Tạo một hàm đệ quy được sử dụng để tìm đường đi ngắn nhất từ ​​điểm đầu đến điểm cuối trong lưới. Hàm lấy lưới, điểm cuối, đường dẫn hiện tại và độ sâu hiện tại làm đầu vào.

Trước tiên, hàm kiểm tra xem độ sâu hiện tại có lớn hơn 200 hay không. Nếu đúng như vậy, hàm return. Điều này là để ngăn chức năng chạy quá lâu.

Sau đó, hàm sẽ kiểm tra xem điểm hiện tại có phải là điểm kết thúc hay không. Nếu có, hàm return đường dẫn.

Sau đó, chức năng này sẽ kiểm tra xem điểm hiện tại có phải là một bức tường hay không. Nếu có, hàm return.

Sau đó, chức năng này sẽ kiểm tra xem điểm hiện tại có phải là một quả bom hay không. Nếu có, hàm return.

Sau đó, chức năng đánh dấu điểm hiện tại là đã truy cập.

Sau đó, hàm sẽ xáo trộn ngẫu nhiên các hướng.

Hàm sau đó kiểm tra từng hướng. Nếu hướng an toàn, hàm sẽ thêm điểm tiếp theo, theo hướng vào đường dẫn và gọi hàm theo cách đệ quy với đường dẫn mới và độ sâu mới.

+ Thuật toán 2 (Dijkstra): Tìm đường đi ngắn nhất

1. **Xây dụng vụ nổ:**

Vụ nổ sẽ được khởi tạo khi có boom và thời gian xuất hiện của nó bằng 3s, khi khởi tạo hàm sẽ lấy thông tin từ danh sách sectors (danh sách vùng mà boom được phép kích hoạt vụ nổ) từ lớp Bomb. Nếu trong danh sách sectors có chứa vị trí của item power up đã được thêm, sẽ bị phá hủy.

Sau khi nổ sẽ kiểm tra trong danh sách sectors, nếu có chứa thùng gỗ sẽ random ra các power up.

Tạo 1 hàm update frame của vụ nổ, vụ nổ sẽ được kích hoạt trong khoảng 0,2s.

