

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

---



**BÁO CÁO BÀI TẬP LỚN**  
**LẬP TRÌNH PYTHON**  
**CHỦ ĐỀ : PYGAME\_SHOOTER**

Tên thành viên

Mã số sinh viên

Trần Chung Khải

B22DCCN443

Phạm Văn Đức

B22DCCN244

Trần Đức Phương

B22DCCN640

Dương Văn Hiếu

B22DCCN302

Nguyễn Văn Thái

B22DCCN783

Giảng viên hướng dẫn: Nguyễn Quỳnh Chi

**Hà Nội**

**ĐÓNG GÓP CỦA CÁC THÀNH VIÊN TRONG NHÓM 05**

<b>Họ và tên</b>	<b>Mã sinh viên</b>	<b>Công việc</b>
Trần Chung Khải	B22DCCN443	Lên ý tưởng + viết báo cáo
Phạm Văn Đức	B22DCCN244	Code chính
Trần Đức Phương	B22DCCN640	Code chính
Dương Văn Hiếu	B22DCCN302	Tìm hiểu thư viện và hỗ trợ code
Nguyễn Văn Thái	B22DCCN783	Tìm ảnh, thiết kế Map

# ĐỀ TÀI: PYGAME - SHOOTER

## I. Giới thiệu đề tài

### 1. Mở đầu

Giới thiệu về Python và Pygame - shooter

Python là một ngôn ngữ lập trình nổi bật, được ưa chuộng bởi sự dễ sử dụng và tính linh hoạt. Pygame là một thư viện Python mạnh mẽ giúp người lập trình dễ dàng phát triển các trò chơi 2D. Pygame cung cấp các công cụ để xử lý đồ họa, âm thanh, sự kiện và nhiều tính năng khác, giúp việc phát triển game trở nên đơn giản và nhanh chóng. Trong bài viết này, chúng ta sẽ xây dựng một trò chơi bắn súng đơn giản bằng Python và Pygame.

## II. Kịch bản

Player sẽ có 100 máu 10 viên đạn và 5 quả bom khi bắt đầu chơi

### 1. Di chuyển nhân vật

- Nhân vật sẽ được di chuyển bằng các phím W, A, D
- Bắn đạn : chuột phải, E
- Nhảy : space hoặc W
- Ném bom : R
- Thời gian chơi ở góc phải màn hình
- Cập nhật thời gian nhanh nhất khi hoàn thành game

### 2. Mô tả bản đồ và hoạt động của game

#### 2.1. Mô tả bản đồ

- Game gồm 3 bản đồ
- Qua mỗi map độ khó sẽ tăng lên

#### 2.2. Hoạt động của game

- Cần tiêu diệt 100% kẻ thù để chuyển sang map mới
- Trong khoảng thời gian chơi có thể pause để tăng, giảm âm lượng, và thay đổi nút di chuyển tùy thích
- Ở map cuối khi mà tiêu diệt boss thì sẽ xuất hiện cánh cổng kết thúc game

### **III. Các thư viện python đã sử dụng**

#### **1. Pygame**

-Pygame là một thư viện mạnh mẽ và dễ sử dụng, cho phép bạn phát triển các trò chơi 2D nhanh chóng và hiệu quả trong Python. Nó là một công cụ tuyệt vời cho những người mới bắt đầu học lập trình game và giúp bạn tạo ra các trò chơi đơn giản mà không phải viết mã quá phức tạp

#### **2. Os**

- Thư viện os trong Python là một công cụ rất mạnh mẽ giúp bạn tương tác với hệ điều hành để thực hiện các tác vụ như quản lý tệp và thư mục, xử lý đường dẫn, điều khiển tiến trình, và truy cập các thông tin hệ thống. Đây là một thư viện rất hữu ích khi bạn cần xây dựng các ứng dụng Python có thể tương tác với môi trường hệ điều hành một cách linh hoạt và hiệu quả.

#### **3. Random**

- Thư viện random trong Python là một công cụ mạnh mẽ để tạo số ngẫu nhiên và thực hiện các thao tác liên quan đến sự ngẫu nhiên, như chọn lựa phần tử ngẫu nhiên từ danh sách, làm xáo trộn dãy số, và sinh số thực hoặc số nguyên trong một khoảng nhất định. Thư viện này thường được sử dụng trong các ứng dụng như trò chơi, mô phỏng, kiểm tra giả thuyết thống kê, hoặc bất kỳ trường hợp nào cần sự ngẫu nhiên.

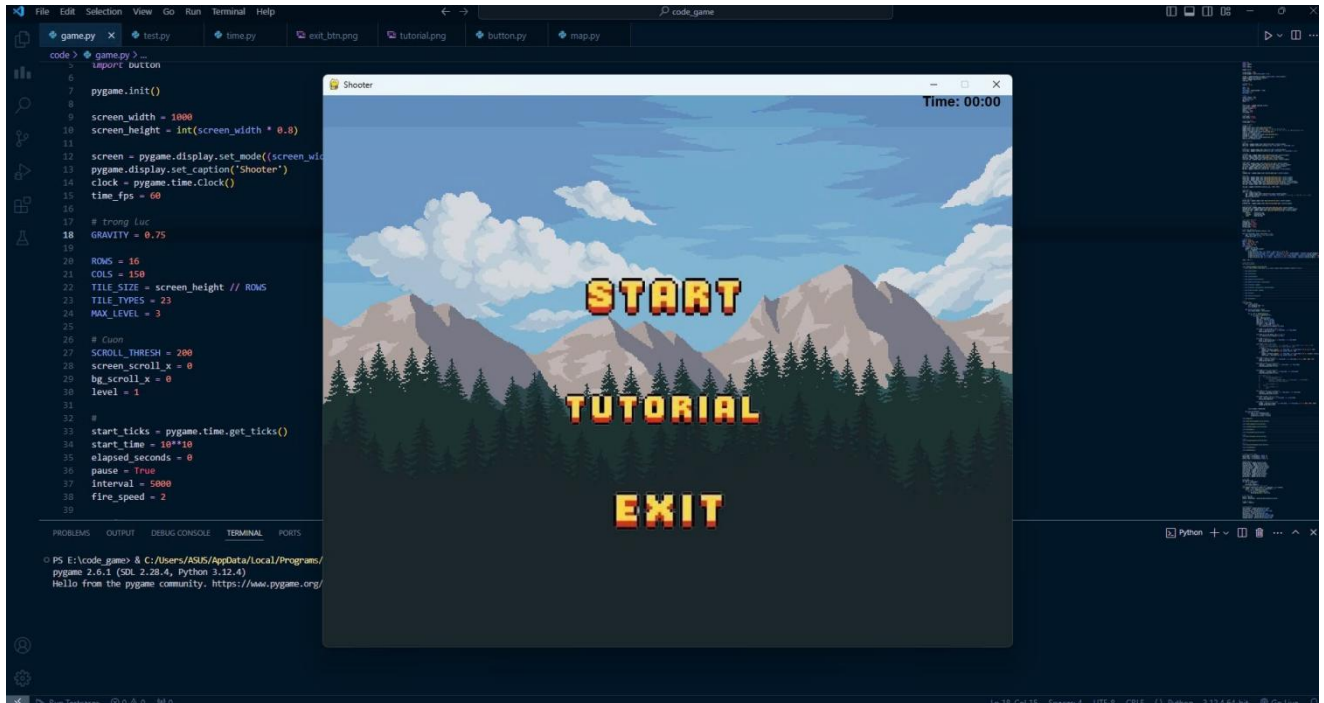
#### **4. Csv**

- Csv là một thư viện mạnh mẽ trong Python để làm việc với các tệp CSV. Thư viện này giúp bạn dễ dàng đọc, ghi, và thao tác với dữ liệu trong tệp CSV, một định dạng phổ biến trong việc lưu trữ và trao đổi dữ liệu giữa các hệ thống và ứng dụng.

- Nó hỗ trợ các tính năng như đọc và ghi các tệp CSV với hoặc không có tiêu đề, điều chỉnh dấu phân cách, và xử lý các dòng dữ liệu dưới dạng danh sách hoặc từ điển.

## IV. Mô tả chức năng của game (Run Game)

### 1. Start game



```
if start_game == False:
    pause = True
    draw_bg()
    # them cac nut
    if start_button.draw(screen, screen_width // 2 - 130, screen_height // 2 - 150, 0.3):
        start_game = True
        start_intro = True

    if exit_button.draw(screen, screen_width // 2 - 110, screen_height // 2 + 150, 0.3):
        run = False
    if tutorial_button.draw(screen, screen_width // 2 - 150, screen_height // 2 + 25, 0.25):
        pass
```

- Click chuột để bắt đầu game
- Exit để kết thúc

```

File Edit Selection View Go Run Terminal Help
game.py test.py time.py exit_btn.png tutorial.png button.py map.py
code > game.py > World > process_data

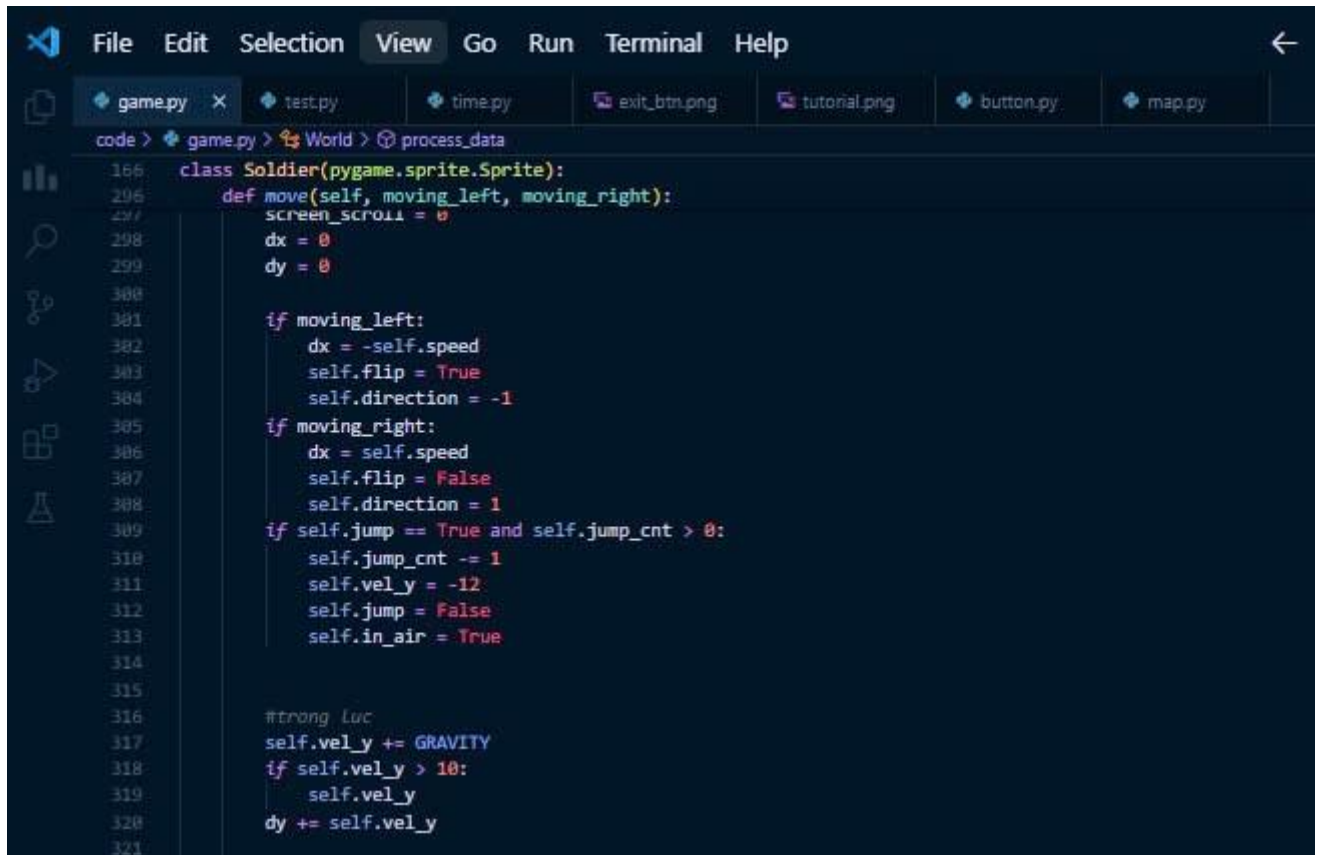
475 class World():
476     def __init__(self):
477         self.obstacle_list = []
478         self.cooldown = 0
479
480     def process_data(self, data):
481         self.level_length = len(data[0])
482
483         for y, row in enumerate(data):
484             for x, tile in enumerate(row):
485                 if tile >= 0:
486                     img = img_list[tile]
487                     img_rect = img.get_rect()
488                     img_rect.x = x * TILE_SIZE
489                     img_rect.y = y * TILE_SIZE
490                     tile_data = (img, img_rect)
491                     if tile >= 0 and tile <= 8: # đất
492                         self.obstacle_list.append(tile_data)
493
494                     elif tile >= 9 and tile <= 10: # nước
495                         water = Decoration(img, x * TILE_SIZE, y * TILE_SIZE)
496                         water_group.add(water)
497
498                     elif tile >= 11 and tile < 14: # trong tri
499                         self.obstacle_list.append(tile_data)
500
501                     elif tile == 14: # cỏ
502                         grass = Decoration(img, x * TILE_SIZE, y * TILE_SIZE)
503                         grass_group.add(grass)
504
505                     elif tile == 15: # tao player
506                         # player = Soldier('player', x * TILE_SIZE, y * TILE_SIZE, 0.9, 5, 10, 5, 100)
507                         # health_bar = HealthBar(10, 10, player.health, 100)
508                         if level == 1:
509                             player = Soldier('player', x * TILE_SIZE, y * TILE_SIZE, 0.9, 5, 10, 5, 100)
510                             health_bar = HealthBar(10, 10, player.health, 100)
511                         if level > 1:
512                             player = Soldier('player', x * TILE_SIZE, y * TILE_SIZE, 0.9, 5, stay[0], stay[1], stay[2])
513                             health_bar = HealthBar(10, 10, player.health, 100)
514
515                     elif tile == 16: # tao enemy
516                         enemy = Soldier('enemy', x * TILE_SIZE, y * TILE_SIZE, 1.2, 2, 1000, 1000, 200)
517                         enemy_group.add(enemy)
518
519                     elif tile == 19: # tao health_box
520                         item_box = ItemBox('Health', x * TILE_SIZE, y * TILE_SIZE)
521                         item_box_group.add(item_box)
522
523                     elif tile == 17: # tao ammo_box
524                         item_box = ItemBox('Ammo', x * TILE_SIZE, y * TILE_SIZE)
525                         item_box_group.add(item_box)
526
527                     elif tile == 18: # tao grenade_box
528                         item_box = ItemBox('Grenade', x * TILE_SIZE, y * TILE_SIZE)
529                         item_box_group.add(item_box)
530
531                     elif tile == 20: # tao loi thout
532                         exit = Decoration(img, x * TILE_SIZE, y * TILE_SIZE)
533                         exit_group.add(exit)
534
535                     elif tile == 21: # tao con roog
536                         dragon = Soldier('dragon', x * TILE_SIZE, y * TILE_SIZE, 1.3, 2, 1000, 1000, 5000)
537                         dragon_group.add(dragon)
538
539
540
541         return player, health_bar
542
543     def draw_world(self):
544         for tile in self.obstacle_list:
545             tile[1][0] += screen_scroll_x

```

## 2. Nhân vật

### a) Di chuyển nhân vật

- Có thể nhảy 2 lần



```
code > game.py > World > process_data
166 class Soldier(pygame.sprite.Sprite):
296     def move(self, moving_left, moving_right):
297         screen_scroll = 0
298         dx = 0
299         dy = 0
300
301         if moving_left:
302             dx = -self.speed
303             self.flip = True
304             self.direction = -1
305         if moving_right:
306             dx = self.speed
307             self.flip = False
308             self.direction = 1
309         if self.jump == True and self.jump_cnt > 0:
310             self.jump_cnt -= 1
311             self.vel_y = -12
312             self.jump = False
313             self.in_air = True
314
315
316         #trọng lực
317         self.vel_y += GRAVITY
318         if self.vel_y > 10:
319             self.vel_y = 10
320         dy += self.vel_y
321
```

```

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        run = False

    #nhấn bàn phím
    if event.type == pygame.KEYDOWN:
        if (event.key == pygame.K_a or event.key == pygame.K_LEFT) and escape_game == False: #trái
            moving_left = True

        if (event.key == pygame.K_d or event.key == pygame.K_RIGHT) and escape_game == False: #phải
            moving_right = True

        if (event.key == pygame.K_SPACE or event.key == pygame.K_w or event.key == pygame.K_UP) and player.alive and escape_game == False: #nhảy
            player.jump = True
            jump_fx.play()

        if event.key == pygame.K_ESCAPE and escape_game == False:
            death_fade
            escape_game = True
        elif event.key == pygame.K_ESCAPE and escape_game == True:
            escape_game = False
            start_intro = True

        if event.key == pygame.K_r and escape_game == False:
            grenade_key = True

    if event.type == pygame.MOUSEBUTTONDOWN and escape_game == False:
        if event.button == 3:
            shoot_key = True

    #nhả bàn phím
    if event.type == pygame.KEYUP:
        if event.key == pygame.K_a or event.key == pygame.K_LEFT:
            moving_left = False
        if event.key == pygame.K_d or event.key == pygame.K_RIGHT:
            moving_right = False
        if event.key == pygame.K_w or event.key == pygame.K_UP or event.key == pygame.K_SPACE:
            player.jump = False
        if event.key == pygame.K_r:
            grenade_key = False
            grenade_throw = False

    if event.type == pygame.MOUSEBUTTONUP:
        if event.button == 3:
            shoot_key = False

pygame.display.update()

```

- src di chuyển nhân vật

## b) Chức năng bắn súng và ném lựu đạn

```

def shoot(self):
    if self.shoot_cooldown == 0 and self.ammo > 0:
        self.shoot_cooldown = 20
        bullet = Bullet(self.rect.centerx + (0.75 * self.rect.size[0] * self.direction), self.rect.centery, self.direction)
        bullet_group.add(bullet)
        # giảm đạn số đạn
        self.ammo -= 1
        shot_fx.play()

def grenade(self):
    if self.grenade_cooldown == 0 and self.grenades > 0:
        self.grenade_cooldown = 200
        grenade = Grenade(enemy.rect.centerx + (0.5 * enemy.rect.size[0] * enemy.direction), enemy.rect.top, enemy.direction)
        grenade_group.add(grenade)
        self.grenades -= 1

```



### c) Va chạm với đất, thùng, nước

```
#trả lại va chạm
for tile in world.obstacle_list:
    if tile[1].colliderect(self.rect.x + dx, self.rect.y - 10, self.width, self.height):
        dx = 0
        # ai va vào tường sẽ quay lại
        if self.char_type == 'enemy' or self.char_type == 'dragon':
            self.direction *= -1
            self.move_counter = 0 # hình ảnh bắt đầu lại

    if tile[1].colliderect(self.rect.x, self.rect.y + dy, self.width, self.height):
        # check nv bên dưới đất
        if self.vel_y < 0:
            self.vel_y = 0
            dy = tile[1].bottom - self.rect.top
        # check nv bên trên mặt đất
        elif self.vel_y >= 0:
            self.vel_y = 0
            self.in_air = False
            self.jump_cnt = 2
            dy = tile[1].top - self.rect.bottom

# check nv chạm vào nước
if pygame.sprite.spritecollide(self, water_group, False):
    self.health = 0
```

## 3. Kẻ địch

### a) Enemy

```
def ai(self):
    if self.alive and player.alive:
        if self.idling == False and random.randint(1, 200) == 1:
            self.update_action(0)
            self.idling = True
            self.idling_counter = 50

        #check khi kẻ địch nhìn thấy nv
        if self.vision.colliderect(player.rect):
            self.update_action(0)
            self.shoot()
            if level >= 2:
                self.grenade()

        else:
            if self.idling == False:
                if self.direction == 1:
                    ai_moving_right = True
                else:
                    ai_moving_right = False
                ai_moving_left = not ai_moving_right
                self.move(ai_moving_left, ai_moving_right)
                self.update_action(1)
                self.move_counter += 1
                self.vision.center = (self.rect.centerx + 75 * self.direction, self.rect.centery)
                # pygame.draw.rect(screen, RED, self.vision)
                if self.move_counter > 40:
                    self.direction *= -1
                    self.move_counter *= -1
            else:
                self.idling_counter -= 1
                if self.idling_counter <= 0:
                    self.idling = False

    self.rect.x += screen_scroll_x
```

## b) Boss(dragon)

```
def dragon_ai(self, speed):
    if self.alive and player.alive:
        if self.idling == False and random.randint(1, 200) == 1: #self.frame_index == len(self.animation_list[self.action]) - 1:
            self.update_action(0)
            self.idling = True
            self.idling_counter = 50

        #check khi ke dich nhin thay nv
        if self.vision_dragon.colliderect(player.rect) and self.fire_cooldown < 50: # and self.fire_cooldown == 0
            self.first_fire += 1
            self.update_action(2)
            if self.frame_index == len(self.animation_list[self.action]) - 4:
                self.fire(speed)

        elif self.vision_dragon_close.colliderect(player.rect) and self.fire_cooldown != 0:
            self.update_action(2)
            if self.frame_index > len(self.animation_list[self.action]) - 2:
                if pygame.sprite.spritecollide(player, dragon_group, False):
                    if player.alive:
                        player.health -= 100
                        pass

            else:
                if self.idling == False:
                    if self.direction == 1:
                        ai_moving_right = True
                    else:
                        ai_moving_right = False
                    ai_moving_left = not ai_moving_right
                    self.move(ai_moving_left, ai_moving_right)
                    self.update_action(1)
                    self.move_counter += 1
                    self.vision_dragon.center = (self.rect.centerx + 220 * self.direction, self.rect.centery + 30)
                    self.vision_dragon_close.center = (self.rect.centerx + 100 * self.direction, self.rect.centery + 30)
                    # pygame.draw.rect(screen, RED, self.vision_dragon, 1)
                    # pygame.draw.rect(screen, RED, self.vision_dragon_close, 4)
                    # pygame.draw.rect(screen, RED, self.rect, 1)
                    if self.move_counter > 400:
                        self.direction *= -1
                        self.move_counter *= -1
                else:
                    self.idling_counter -= 1
                    if self.idling_counter <= 0:
                        self.idling = False

        elif self.alive == False and level == 3 and player.alive:
            gate = Decoration(gate_img, self.rect.centerx, self.rect.centery + 20)
            gate_group.add(gate)

    self.rect.x += screen_scroll_x
```

## Phun lửa



```
File Edit Selection View Go Run Terminal Help
game.py x test.py time.py exit_btn.png tutorial.png button.py
code > game.py > World > process_data
637 class Fire(pygame.sprite.Sprite):
638     def __init__(self, x, y, direction, speed):
648         self.health = 90
649
650
651     def update(self):
652         # duong di vien dan
653         self.rect.x += (self.direction * self.speed) + screen_scroll_x
654         self.rect.y += 0.49
655         pygame.draw.rect(screen, RED, self.rect, 1)
656         # vien dan vien mot
657         if self.rect.right < 0 or self.rect.left > screen_width:
658             self.kill()
659
660         # ktra vao cham vao dat
661         for tile in world.obstacle_list:
662             if self.rect.colliderect(tile[1]):
663                 self.kill()
664                 if self.direction == -1:
665                     explosion = Explosion(self.rect.x, self.rect.y, 3)
666                     explosion_group.add(explosion)
667                 else:
668                     explosion = Explosion(self.rect.x + 100, self.rect.y, 3)
669                     explosion_group.add(explosion)
670
671             if pygame.sprite.spritecollide(player, explosion_group, False):
672                 if player.alive:
673                     player.health -= 100
674                     self.kill()
675
676         # va cham nhan vat voi dan
677         if pygame.sprite.spritecollide(player, fire_group, False):
678             if player.alive:
679                 self.kill()
680                 if self.direction == -1:
681                     explosion = Explosion(self.rect.x, self.rect.y, 3)
682                     explosion_group.add(explosion)
683                 else:
684                     explosion = Explosion(self.rect.x + 100, self.rect.y, 3)
685                     explosion_group.add(explosion)
686
687             player.health -= 100
688
689         if self.health <= 0:
690             self.kill()
691             if self.direction == -1:
692                 explosion = Explosion(self.rect.x, self.rect.y, 3)
693                 explosion_group.add(explosion)
694             else:
695                 explosion = Explosion(self.rect.x + 100, self.rect.y, 3)
696                 explosion_group.add(explosion)
697
698             if pygame.sprite.spritecollide(player, explosion_group, False):
699                 if player.alive:
700                     player.health -= 100
701
702
```

- Đòn đánh khi chạm vào vật cản sẽ gây sát thương nổ lan
- Chạm vào nhân vật thì sẽ game mới

## 4. Dạn được

```
class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y, direction):
        pygame.sprite.Sprite.__init__(self)
        self.speed = 10
        self.image = bullet_img
        self.rect = self.image.get_rect()
        self.rect.center = (x, y)
        self.direction = direction

    def update(self):
        # duong di vien dan
        self.rect.x += (self.direction * self.speed) + screen_scroll_x

        # vien dan vien mat
        if self.rect.right < 0 or self.rect.left > screen_width:
            self.kill()
            pass

        # ktra vao cham vao dat
        for tile in world.obstacle_list:
            if tile[1].colliderect(self.rect):
                self.kill()

        # va cham nhan vat voi dan
        if pygame.sprite.spritecollide(player, bullet_group, False):
            if player.alive:
                player.health -= 5
                self.kill()

        for enemy in enemy_group:
            if pygame.sprite.spritecollide(enemy, bullet_group, False):
                if enemy.alive:
                    enemy.health -= 50
                    self.kill()

        for dragon in dragon_group:
            if pygame.sprite.spritecollide(dragon, bullet_group, False):
                if dragon.alive:
                    dragon.health -= 50
                    self.kill()

        for fire in fire_group:
            if pygame.sprite.spritecollide(fire, bullet_group, False):
                self.kill()
                if fire.health > 0:
                    fire.health -= 30

        # if pygame.sprite.spritecollide(self, bullet_group, False):
        #     self.kill()
```

- Khi đạn chạm vào player thì trừ 5 máu
- Khi đạn chạm vào enemy, boss trừ 50 máu
- 3 viên đạn từ player có thể triệt tiêu đòn đánh từ boss



## 5. Nade

```
class Grenade(pygame.sprite.Sprite):
    def __init__(self, x, y, direction):
        pygame.sprite.Sprite.__init__(self)
        self.timer = 100
        self.vel_y = -11
        self.speed = 7
        self.image = grenade_img
        self.rect = self.image.get_rect()
        self.rect.center = (x, y)
        self.width = self.image.get_width()
        self.height = self.image.get_height()
        self.direction = direction

    def update(self):
        self.vel_y += GRAVITY
        dx = self.speed * self.direction
        dy = self.vel_y

        #ktra va chạm voi ground
        for tile in world.obstacle_list:
            if tile[1].colliderect(self.rect.x + dx, self.rect.y, self.width, self.height):
                self.direction *= -1
                dx = self.direction * self.speed

            if tile[1].colliderect(self.rect.x, self.rect.y + dy, self.width, self.height):
                self.speed = 0
                # check ben duoi dat
                if self.vel_y < 0:
                    self.vel_y = 0
                    dy = tile[1].bottom - self.rect.top
                # check ben tren mat dat
                elif self.vel_y >= 0:
                    self.vel_y = 0
                    dy = tile[1].top - self.rect.bottom

        self.rect.x += dx + screen_scroll_x
        self.rect.y += dy

        self.timer -= 1
        if self.timer <= 0:
            self.kill()
            grenade_fx.play()
            explosion = Explosion(self.rect.x, self.rect.y, 2)
            explosion_group.add(explosion)

            # khi phat na se gay sat thuong
            if pygame.sprite.spritecollide(player, explosion_group, False):
                if player.alive:
                    player.health -= 100
                    self.kill()

            for enemy in enemy_group:
                if pygame.sprite.spritecollide(enemy, explosion_group, False):
                    if enemy.alive:
                        enemy.health = 0
                        self.kill()

            for dragon in dragon_group:
                if pygame.sprite.spritecollide(dragon, explosion_group, False):
                    if dragon.alive:
                        dragon.health -= 100
                        self.kill()

            for fire in fire_group:
                if pygame.sprite.spritecollide(fire, explosion_group, False):
                    if fire.health > 0:
                        fire.health -= 100
                        self.kill()
```

- Bấm R để ném nade
- Nade sẽ nổ sau khoảng 3s và gây sát thương lên cả player lẫn enemy, boss

## 6. Pause game

```
elif escape_game == True and start_intro == False:
    pause = True
    screen.fill(BLACK)
    draw_text(f'VOLUME: {round(volume * 100)} %:', font, WHITE, 10, 75)
    if up_button.draw(screen, (0 + 200), 50, 0.2) and volume <= 0.99:
        volume += 0.1

    if down_button.draw(screen, 0 + 200, 50 + 50, 0.2) and volume >= 0.01:
        volume -= 0.1

    pygame.mixer.music.set_volume(volume)
    shot_fx.set_volume(volume)
    grenade_fx.set_volume(volume)
    jump_fx.set_volume(volume)

    if continue_button.draw(screen, (0 + 100), screen_height - 100, 0.2):
        escape_game = False
        start_intro = True
        pause = False
        pass

    if exit_button.draw(screen, (screen_width - 200), screen_height - 100, 0.2):
        run = False
        pass
```

## V. Map.py

- Sử dụng File này để tạo map
- Vị trí nhân vật, đạn dược, cứu thương,... được đặt ở đâu
- Khi tạo xong map sẽ lưu vào file.csv
- Có các nút save, load, delete, create để tạo và chỉnh sửa Map trong file.csv

## VI. Button.py

- Sử dụng class button
- Dùng để vẽ các button và thực thi khi mình click chuột trái vào button

```
import pygame

#button class
class Button():
    def __init__(self, image):
        self.width = image.get_width()
        self.height = image.get_height()
        self.image = image
        self.clicked = False

    def draw(self, surface, x, y, scale):
        action = False

        self.image = pygame.transform.scale(self.image, (int(self.width * scale), int(self.height * scale)))
        self.rect = self.image.get_rect()
        self.rect.topleft = (x, y)
        # pygame.draw.rect(surface, (255, 0, 0), self.rect, 1)

        #get mouse position
        pos = pygame.mouse.get_pos()

        #check mouseover and clicked conditions
        if self.rect.collidepoint(pos):
            if pygame.mouse.get_pressed()[0] == 1 and self.clicked == False:
                action = True
                self.clicked = True

            if pygame.mouse.get_pressed()[0] == 0:
                self.clicked = False

        #draw button
        surface.blit(self.image, (self.rect.x, self.rect.y))

        return action
```

## **VII. Run game**

### **1. Class Soldier**

- Dùng để tạo và điều khiển player
- Dùng để tạo khả năng bắn, ném bom
- Dùng để di chuyển enemy, boss theo một khoảng cách nhất định, và khả năng gây đòn đánh của chúng
- Kiểm tra enemy, boss còn sống hay đã chết
- Vẽ player, enemy, boss

### **2. Class world**

- Dùng để kết hợp với Map.py, File.csv ứng với từng thông số để tạo map và nhân vật

### **3. Class repeat**

- Lặp lại những item box xuất hiện trở lại sau một thời gian nhất định ( chỗ mình mong muốn xuất hiện trở lại)

### **4. Class decoration**

- Dùng để trang trí cho map sinh động hơn

### **5. Class Water**

- Tạo nước

### **6. Class item box**

- Tạo các box với khả năng riêng của từng item box
- khi mà chạm vào health box thì tăng hp, hộp đạn tăng đạn, lựu đạn tăng

lựu đạn

### **7. Class Healthbar**

- Tạo ra thanh máu của nv mà mình muốn

### **8. Class Bullet**

- Tạo ra đạn
- Khi đạn chạm vào player thì -5 máu, chạm vào kẻ địch thì trừ 50 máu

### **9. Class grenade**

- Tạo ra bom
- Khi bom chạm vào player thì die, chạm vào kẻ địch thì cũng die, chạm vào đòn đánh của boss sẽ phát nổ lan, gây 100 sát thương lên boss

### **10. Class explosion**

- Tạo ra vụ nổ

### **11. Class screenFade**

- Dùng để Fade lên màn hình khi bắt đầu game hoặc kết thúc game

### **12. Class gameEnding**

- Màn hình khi kết thúc game
- Xuất hiện thời gian hoàn thành game và thời gian nhanh nhất hoàn thành game.

### **13. Class fire**

- Tạo ra đòn đánh của boss
- Khi chạm vào vật cản sẽ phát nổ, gây sát thương lan, chạm vào player - > die, chạm vào bullet 3 lần sẽ phát nổ lan

***VIII.***