

# 1팀 일기예보 파이게임 PPT!

발표 시작  
할게용~



팀장 : 박영범  
조원:  
권승현, 김예진,  
송하명, 오병훈,  
최상준



# Pygame을 활용한 슈팅게임 제작

프로젝트기간: 2021.11.23~2021.12.10

## Work Team & Member

1팀 (일기예보)

팀장 : 박영범

팀원 : 권승현, 김예진, 오병훈, 최상준, 송하명

## Work Schedule

11.23 프로젝트 계획안 발표

11.24~12.08

이미지 파일 수집, 코드 작성, 오류 수정

12.09 발표 ppt 작성

12.10 프로젝트 발표

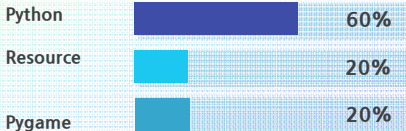
## Resource Files

freesound.org,  
youtube audio library,  
pixabay

## Python 라이브러리 Pygame을 이용하여 슈팅게임 제작

수직으로 내려오는 바이러스를 제거하여  
점수와 아이템을 얻어, 보스를 파괴하는 게임 제작.

## Skills





목차 !



개요



역할분담



설계



게임시연



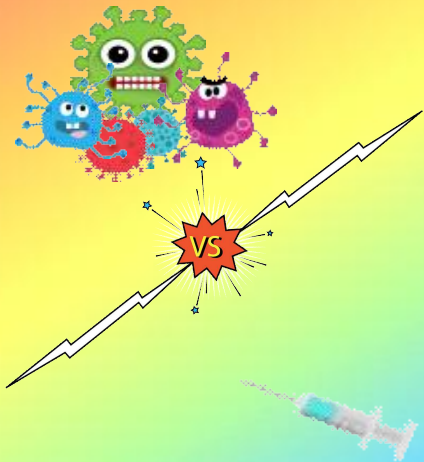
# 1. 개요

- 컨셉, 스토리라인, 게임방식, 사용프로그램 -





컨셉 및 로고



전 세계적으로 급격히 늘어나  
는 코로나 바이러스들!

언제까지 지켜만 보실겁니까?

이제는 저희가 직접 물리쳐서 세계를  
구해 봅시다!





# 게임방식

직접 백신을 사용해서 징글징글하게 늘어나는  
코로나 바이러스를 박멸하자

1. 사용키 키보드 키 상하좌우
2. 필살기 버튼 A : 내려오는 바이러스를 전체  
제거할 수 있는 기능

단) 기회는 한 번뿐이라 신중하게 사용!

3. 일정 조건에 도달하면 보스 등장!  
보스를 제압해야 게임 clear!





# 게임설명


게임에 시간초를 부여하여 시간에 맞춰 보스클리어가 목표!  
실패를 하게 된다면 세계는 아쉽지만 멸망하게 됩니다.

시간에 따라 점점 늘어나는 바이러스 !  
많이 떨어진다고 피하기만 하면 놓친 바이러스 1개당  
1초씩 시간이 줄어듭니다. (놓치지 않고 파괴하면 +1초)

일반 바이러스 사이에 변이바이러스도 나타나게 구현하여  
게임 진행시 재미를 높였습니다.

변이바이러스 파괴시 아이템이 드롭되어 난이도 있는 게임을  
조금 더 쉽게 클리어 할 수 있게 구현했습니다.

일정 스코어에 도달하게 되면 보스몬스터가 등장!  
보스를 제거해야 게임이 클리어 되니 꼭 무찔러서 세계를 구합시다!



꼭 작전에 성공해서  
마스크 없는 삶을 살게해죠..

# 게임 캐릭터 및 출현 몬스터 소개



기본 바이러스 몬스터



변이 바이러스 몬스터 아이템을 드롭



바이러스를 물리치는 백신 모양의 전투기와, 드롭되는 아이템







## 2. 역할분담 및 리소스 파일





## 역할분담

박영범 : 변이바이러스 구현, 파이게임 발표

권승현 : 정보수집, 기본적인 틀 구현, ppt제작

김예진 : 필살기 구현 및 오류수정

송하명 : 초기시간 및 시간초 구현

오병훈 : 전체적인 틀 구현, 아이템개발 구현 및 오류수정

최상준 : 게임 화면 구성 및 오류수정





# 리소스 파일

pygame

파일 홈 공유 보기 사진 도구

← → ↑ > pygame > pygame

pygame 검색

즐거찾기

- 바탕 화면
- 다운로드
- 문서
- 사진
- Image
- OpenCV
- pygame
- pygame\_project (1)
- OneDrive
- 내 PC
- 3D 개체
- 다운로드
- 동영상
- 문서
- 바탕 화면
- 사진
- 음악
- 로컬 디스크 (C:)
- DRIVERS (D:)
- 네트워크

background boss1 explosion explosion01 explosion02 explosion03 explosion04

fighter gameover item1 item2 item3 item4 missile

music music1 music2 한글 NanumGothic Pew shieldfighter

v1 v2 v3 v4 v5 vv1 vv2 vv3 vv4 vv5

31개 항목

리소스 파일 출처 :  
freesound.org, youtube  
audio library, pixabay



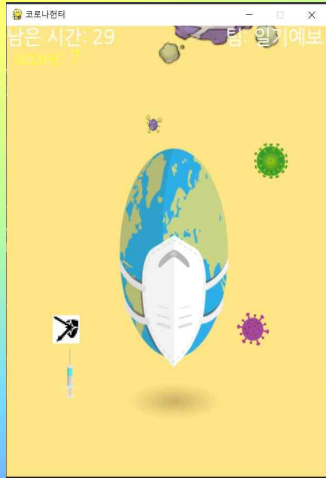
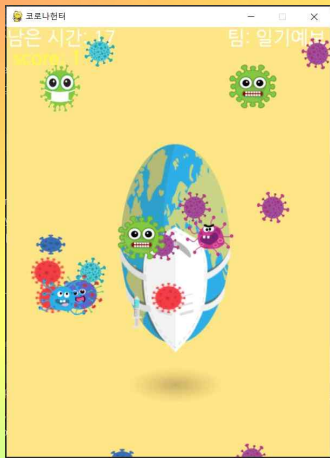


### 3. 게임 설계 및 코드





# 게임 실행 화면





# 게임 코드

# 전투기 클래스 = 전투기 생성 하는거

```
class Fighter(pygame.sprite.Sprite):
```

```
    def __init__(self):
```

```
        super(Fighter, self).__init__()
```

```
        self.fighter_images = ['fighter.png', 'shieldfighter.png']
```

```
        self.image = pygame.image.load(self.fighter_images[0]) # 이미지 불러오기
```

```
        self.rect = self.image.get_rect()
```

```
        self.rect.x = int(WINDOW_WIDTH / 2) # 전투기 x 축 위치
```

```
        self.rect.y = WINDOW_HEIGHT - self.rect.height # 전투기 y 축 위치
```

```
        self.dx = 0 # 초기 전투기 방향설정 x 축
```

```
        self.dy = 0 # 초기 전투기 방향설정 y 축
```

```
    def update(self):
```

```
        self.rect.x += self.dx
```

```
        self.rect.y += self.dy
```

```
        if self.rect.x < 0 or self.rect.x + self.rect.width > WINDOW_WIDTH:
```

```
            self.rect.x -= self.dx
```

```
            # 만약 전투기가 옆으로 화면을 벗어나려고 하면 못 벗어나게 조건을 준다.
```

```
        if self.rect.y < 0 or self.rect.y + self.rect.height > WINDOW_HEIGHT:
```

```
            if self.rect.x < 0 or self.rect.x + self.rect.width > WINDOW_WIDTH:
```

```
                self.rect.x -= self.dx
```

```
            # 만약 전투기가 옆으로 화면을 벗어나려고 하면 못 벗어나게 조건을 준다.
```

```
        if self.rect.y < 0 or self.rect.y + self.rect.height > WINDOW_HEIGHT:
```

```
            self.rect.y -= self.dy
```

```
            # 전투기가 위로 화면을 벗어나려고 하면 못 벗어나게 조건을 준다.
```

```
    def get_shield(self, shield): # 쉴드
```

```
        self.shield = shield
```

```
    def draw(self, screen, shield):
```

```
        screen.blit(pygame.image.load(self.fighter_images[shield]), self.rect)
```

```
# self = 전투기 이니까 스크린에 보여주는거 전투기를
```

```
def collide(self, sprites):
```

```
    for sprite in sprites:
```

```
        if pygame.sprite.collide_rect(self, sprite):
```

```
            return sprite
```

백신(전투기)을(를) class 에 넣어서 자주 꺼낼 수 있게 해줍니다!





# 게임 코드

# 날라가는 투기체 생성

```
class Missile(pygame.sprite.Sprite):
```

```
    def __init__(self, xpos, ypos, speed, missile_num): # 초기화 시켜주는 작업 / 미사일의 x포지션, y포지션, 속도를 초기화 할 예정
        super(Missile, self).__init__() # super 상속시켜 self 정보를 가져오다
        self.missile_images = ['missile.png', 'boss_missile.png'] # 미사일과 보스 미사일
        self.missile_num = missile_num
        self.image = pygame.image.load(self.missile_images[self.missile_num]) # 투기체 이미지 가져오기
        self.rect = self.image.get_rect() # self 이미지를 넣어주는?
        self.rect.x = xpos
        self.rect.y = ypos
        self.speed = speed
        self.sound = pygame.mixer.Sound('missile.wav')
        self.sound.set_volume(0.15)
```

# 우주선에서 투기체가 날라가기 때문에 우주선에 위치를 이렇게 넣어주는거란다

```
def launch(self):
```

```
    self.sound.play() # 미사일 날라갈때 소리 내주는 장치
```

```
def update(self):
```

```
    self.rect.y -= self.speed
```

```
    if self.rect.y + self.rect.height < 0:
```

```
def collide(self, sprites):
```

```
    for sprite in sprites:
```

```
        if pygame.sprite.collide_rect(self, sprite):
```

```
            return sprite # 여기도 미사일이 충돌한다면 이라는 조건을 준거
```

미사일 class 만들어서 백  
신에서 사용될 미사일과,  
나중에 보스가 뿜을 미사  
일을 class 입력합니다.





## 게임 코드

```
class Rock(pygame.sprite.Sprite):
    def __init__(self, xpos, ypos, speed):
        super(Rock, self).__init__() # 장애물 클래스 생성 후 받아올 거 또 받아옴
        rock_images = ('v1.png', 'v2.png', 'v3.png', 'v4.png', 'v5.png')

        self.image = pygame.image.load(random.choice(rock_images))
        # 장애물(이미지) 를 랜덤으로 떨어지게 만드는 작업
        self.rect = self.image.get_rect()
        self.rect.x = xpos
        self.rect.y = ypos
        self.speed = speed

    def update(self):
        self.rect.y += self.speed

    def all_kill(self):
        self.kill()

    def out_of_screen(self):
        if self.rect.y > WINDOW_HEIGHT:
```

```
class Rock2(pygame.sprite.Sprite):
```

```
    def __init__(self, xpos, ypos, speed, hp=3):
        super(Rock2, self).__init__() # 장애물 클래스 생성 후 받아올 거 또 받아옴
        rock_images2 = ('w2.png', 'w1.png', 'w3.png', 'w4.png', 'w5.png')

        self.image = pygame.image.load(random.choice(rock_images2))
        # 장애물(이미지) 를 랜덤으로 떨어지게 만드는 작업
        self.rect = self.image.get_rect()
        self.rect.x = xpos
        self.rect.y = ypos
        self.speed = speed
        self.hp = hp

    def update(self):
        self.rect.y += self.speed

    def all_kill(self):
        self.kill()

    def out_of_screen(self):
        if self.rect.y > WINDOW_HEIGHT:
            return True
```

# 운석은 위 > 아래로 떨어지니까 y값만, 스크린 밖으로 가면 True를 줘서 무슨상황을 만들예정

일반 바이러스 와 변이바이러스를 구분하여 나중에 다른 설정을 넣을 수 있게 기본 설정을 class 입력합니다.







## 게임 코드

```
class Item(pygame.sprite.Sprite):
    def __init__(self, xpos, ypos, item_num):
        super(Item, self).__init__()
        item_images = ('item1.png', 'item2.png', 'item3.png', 'item4.png')

        self.item_num = item_num
        self.image = pygame.image.load(item_images[self.item_num])
        self.rect = self.image.get_rect()
        self.rect.x = xpos
        self.rect.y = ypos

    def Item_num(self):
        return self.item_num

    def update(self):
        self.rect.y += 2

    def out_of_screen(self):
        if self.rect.y > WINDOW_HEIGHT:
            return True
```

```
for rock in rocks:
    if rock.out_of_screen():
        rock.kill()
        count_missed += 1
        total_time -= 1 # 화면을 벗어날시 -1초

for rock2 in rocks2:
    if rock2.out_of_screen():
        rock2.kill()
        count_missed += 1
        total_time -= 1

for item in items:
    if item.out_of_screen():
        item.kill()

if fighter.collide(items):
    item = fighter.collide(items)
    item.kill()
```

변이 바이러스를 퇴치시 나올 아이템  
class에 입력해줍니다  
2번 코드처럼 나중에 이런 식으로  
사용됩니다.





## 게임 코드

```
class Boss(pygame.sprite.Sprite): # 보스
    def __init__(self):
        super(Boss, self).__init__()
        self.image = pygame.image.load('boss.png') # 보스 이미지
        self.rect = self.image.get_rect()
        self.rect.x = (WINDOW_WIDTH/2) - (self.rect.width)/2
        self.rect.y = -self.rect.height

    def draw(self, screen):
        screen.blit(pygame.image.load('boss.png'), self.rect)

    def update(self, boss_speedx, boss_speed):
        self.rect.x += boss_speedx
        self.rect.y += boss_speed

    def collide(self, sprites):
        for sprite in sprites:
            if pygame.sprite.collide_rect(self, sprite):
                return sprite
```

보스까지 class로 정의가 끝났다면  
기본 설정은 이렇게 끝이 납니다





## 게임 코드

```
def draw_text(text, font, surface, x, y, main_color):
```

```
    # 화면 텍스트 설정
```

```
    text_obj = font.render(text, True, main_color)
```

```
    text_rect = text_obj.get_rect()
```

```
    text_rect.centerx = x
```

```
    text_rect.centery = y
```

```
    surface.blit(text_obj, text_rect) # blit 화면에 그리는거래
```

```
def occur_explosion(surface, x, y):
```

```
    explosion_image = pygame.image.load('explosion.png')
```

```
    explosion_rect = explosion_image.get_rect()
```

```
    explosion_rect.x = x
```

```
    explosion_rect.y = y
```

```
    surface.blit(explosion_image, explosion_rect)
```

```
    # 폭발이라는 함수를 넣어 폭발이미지 그리는거, x, y 축 설정
```

```
    explosion_sounds = ('explosion01.wav', 'explosion02.wav', 'explosion03.wav')
```

```
    explosion_sound = pygame.mixer.Sound(random.choice(explosion_sounds))
```

```
    explosion_sound.play()
```

```
    explosion_sound.set_volume(0.10)
```

```
def game_loop():
```

```
    default_font = pygame.font.Font('NanumGothic.ttf', 28)
```

```
    background_image = pygame.image.load('background.png')
```

```
    gameover_sound = pygame.mixer.Sound('gameover.wav')
```

```
    gameover_sound.set_volume(0.40)
```

```
    pygame.mixer.music.load('music.mp3')
```

```
    pygame.mixer.music.play(-1)
```

```
    pygame.mixer.music.set_volume(0.20)
```

```
    # loop
```

```
    fps_clock = pygame.time.Clock()
```

```
    # 각 소리와 폰트를 불러오기 설정, 배경음악은 따로 지정 안하고 불러옴. -1 은 무한반복
```

```
    fighter = Fighter()
```

```
    boss = Boss()
```

```
    missiles = pygame.sprite.Group()
```

```
    rocks = pygame.sprite.Group()
```

```
    rocks2 = pygame.sprite.Group()
```

```
    items = pygame.sprite.Group()
```

```
    boss_missiles = pygame.sprite.Group()
```

1번 코드는 화면에 입력될 텍스트창들을, 2번은 바이러스 파괴 및 백신이 파괴되는 사진, 소리를 입력을, 마지막 함수코드는 게임루프 함수를 이용하여 기본 배경사진 및 음악을 설정합니다. 마지막으로 class들을 pygame.group 넣어 줍니다.





## 게임 코드

```
occur_prob1 = 100 # 확률적으로 얼마나 나오는지
occur_prob2 = 200
shot_count = 0

count_missed = 0
total_time = 30
start_ticks = pygame.time.get_ticks()
count_skill = 0

fighter_speed = 3.5 # 초기 전투기 속도
missile_speed = 5 # 초기 미사일 속도
missile_num = 1 # 초기 미사일 개수
shield = 0 # 초기 쉴드 개수
boss_speed = 1 # 초기 보스 y 속도
boss_speedx = 1 # 초기 보스 x 속도
boss_hp = 60 # 보스 체력

m1 = pygame.image.load('missile.png') # 미사일 이미지 가져오기
missileSize = m1.get_rect().size # 미사일 크기 불러오기
missileWidth = missileSize[0] # 미사일 가로 길이
```

기본적으로 게임에서  
숫자를 저장할 변수를  
설정하여 게임에 사용될  
변수들을 각각 초기화 해  
줍니다.





# 게임 코드

```
done = False  
while not done:
```

```
    for event in pygame.event.get():
```

```
        if event.type == pygame.KEYDOWN:
```

```
            if event.key == pygame.K_LEFT:
```

```
                fighter.dx -= fighter_speed
```

```
            elif event.key == pygame.K_RIGHT:
```

```
                fighter.dx += fighter_speed
```

```
            elif event.key == pygame.K_UP:
```

```
                fighter.dy -= fighter_speed
```

```
            elif event.key == pygame.K_DOWN:
```

```
                fighter.dy += fighter_speed
```

```
            elif event.key == pygame.K_SPACE:
```

```
                if missile_num == 1:
```

```
                    missile = Missile(fighter.rect.centerx - missileWidth/2, fighter.rect.y, missile_speed, 0) # x의 정 가운데에서 미사일이 나간다
```

```
                    missile.launch()
```

```
                    missiles.add(missile) # 미사일스 에 미사일 그룹을 추가
```

done = False 게임이 구동되는게 False  
라면 구동시 입력될 키 들을 설정해주는  
코드입니다.

상하좌우, 미사일 키도 설정해 줍니다.





## 게임 코드

```
if missile_num == 1:
    missile = Missile(fighter.rect.centerx - missileWidth/2, fighter.rect.y, missile_speed, 0) # x의 정 가운데에서 미사일이 나간다
    missile.launch()
    missiles.add(missile) # 미사일스 에 미사일 그룹을 추가

if missile_num == 2:
    missile1 = Missile(fighter.rect.x, fighter.rect.y, missile_speed, 0)
    missile2 = Missile(fighter.rect.x + fighter.rect.width - missileWidth, fighter.rect.y, missile_speed, 0)
    missile1.launch()
    missile2.launch()
    missiles.add(missile1)
    missiles.add(missile2)

if missile_num == 3:
    missile1 = Missile(fighter.rect.x, fighter.rect.y, missile_speed, 0)
    missile2 = Missile(fighter.rect.centerx - missileWidth/2, fighter.rect.y, missile_speed, 0)
    missile3 = Missile(fighter.rect.x + fighter.rect.width - missileWidth, fighter.rect.y, missile_speed, 0)
    missile1.launch()
    missile2.launch()
    missile3.launch()
    missiles.add(missile1)
    missiles.add(missile2)
    missiles.add(missile3)
```

done = False 입력칸에 아이템 중 미사일을 추가해주는 능력이 있는 아이템을 현실화 시켜주는 코드입니다. 늘어난 미사일 갯수 마다 그의 따른 효과를 입력하여 구현해논 코드입니다.





# 게임 코드

```
if event.type == pygame.KEYUP:  
    if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT or event.key == pygame.K_a:  
        fighter.dx = 0  
    if event.key == pygame.K_UP or event.key == pygame.K_DOWN:  
        fighter.dy = 0
```

```
elif event.key == pygame.K_a: # 필살기 처리  
    explosion_sounds = ('Pew.mp3')  
    explosion_sound = pygame.mixer.Sound((explosion_sounds))  
    explosion_sound.play()  
    explosion_sound.set_volume(0,10)
```

```
if count_skill == 0:  
    for rock in rocks:  
        rock.all_kill()  
    total_time = total_time + 10  
    count_skill = count_skill + 1
```

```
if count_skill == 1:  
    for rock2 in rocks2:  
        rock2.all_kill()  
    total_time = total_time + 10  
    count_skill = count_skill + 1
```

1번 코드는 필살기 키를 = a 키  
로 구현한 코드이며,  
2번코드는 그 필살기 사용제한  
과 효과를 입력한 코드입니다





## 게임 코드

```
occur_of_rocks = 2 + int(shot_count / 100)
min_rock_speed = 1 + int(shot_count / 80)
max_rock_speed = 1 + int(shot_count / 40)
occur_of_rocks2 = 2 + int(shot_count / 120)
min_rock2_speed = 1 + int(shot_count / 80)
max_rock2_speed = 1 + int(shot_count / 40)
# 장애물 파괴 카운트에 따라 난이도 줘서 떨어지는 장애물 양
# 떨어지는 속도를 더 해 난이도를 줌
```

```
elapsed_time = (pygame.time.get_ticks() - start_ticks) / 1000
timer = (str(int(total_time - elapsed_time)))
```

1번코드는 바이러스가 파괴될때 마다  
(초기화 해둔 변수 + 파괴 바이러스 / 정수)

등장하는 바이러스양, 속도를 조절하는  
난이도 조절 코드입니다.

2번째 코드는 저희 게임에 중요한 시간을  
설정 해 둔 시간초 코드 입니다.







## 게임 코드

```
if int(shot_count) > 5:  
    for rock2 in rocks2:  
        rock2.all_kill()  
  
# 여기 배경노래가 종료되고 다른 노래가 나왔으면 좋겠다 안되면 포기
```

```
if boss.rect.y > 100:  
    boss_speed = -1  
if boss.rect.y < -50:  
    boss_speed = 1  
  
if boss.rect.x < 0:  
    boss_speedx = 1  
if boss.rect.x > WINDOW_WIDTH - boss.rect.width:  
    boss_speedx = -1
```

보스가 가만히 있는게 아닌  
지그재그로 움직일 수 있게  
구현한 코드

if 문을 사용하여 보스가 처음부터  
등장인 아닌 특정 임무 혹은  
특정 점수에 도달해야지 나올  
수 있게 구현한 코드입니다.





## 게임 코드

```

if boss.collide(missiles):
    missile = boss.collide(missiles)
    if missile:
        missile.kill()
        boss_hp -= 1
        print(boss_hp)
    if boss_hp < 1:
        pygame.mixer_music.stop()
        occur_explosion(screen, boss.rect.x, boss.rect.y)
        pygame.display.update()
        gameover_sound.play()
        sleep(1)
        done = True

if random.randint(1, 30) == 1:
    speed = -3
    missile = Missile(random.randint(boss.rect.x, boss.rect.x + boss.rect.width), boss.rect.y + boss.rect.height, speed, 1)
    boss_missiles.add(missile)

```

```

if random.randint(1, 30) == 1:
    speed = -3
    missile = Missile(random.randint(boss.rect.x, boss.rect.x + boss.rect.width), boss.rect.y + boss.rect.height, speed, 1)
    boss_missiles.add(missile)

```

```

boss.draw(screen)
boss.update(boss_speedx, boss_speed)
boss_missiles.draw(screen)
boss_missiles.update()

```

```

if fighter.collide(boss_missiles):
    boss_missile = fighter.collide(boss_missiles)
    if shield == 0:
        pygame.mixer_music.stop()
        occur_explosion(screen, fighter.rect.x, fighter.rect.y)
        pygame.display.update()
        gameover_sound.play()
        sleep(1)
        done = True
    else:
        shield = 0
        boss_missile.kill()

```

1번 코드는 if 문을 사용하여 보스가 백신의 미사일을 맞았을 때,  
2번 코드는 반대로 백신이 보스가 쏘는 보스의 미사일을 맞았을 경우, 보스를 클리어 할 때 효과를 구현한 코드입니다





# 게임 코드

```
for missile in missiles:
```

```
    rock = missile.collide(rock)
```

```
    if rock:
```

```
        missile.kill()
```

```
        rock.kill()
```

```
        occur_explosion(screen, rock.rect.x, rock.rect.y)
```

```
        shot_count += 1
```

```
        total_time += 1
```

```
for missile in missiles:
```

```
    rock2 = missile.collide(rock2)
```

```
    if rock2:
```

```
        missile.kill()
```

```
        rock2.hp -= 1
```

```
        if rock2.hp == 0:
```

```
            rock2.kill()
```

```
            occur_explosion(screen, rock2.rect.x, rock2.rect.y)
```

```
    item_type = random.randrange(20)
```

```
    if item_type == 0: # 1/20 확률로 0번 아이템
```

```
        item_num = 0
```

```
    elif item_type == 1 or item_type == 2 or item_type == 3: # 3/20 확률로 1번 아이템
```

```
        item_num = 1
```

```
    elif 3 < item_type < 12: # 2/5 확률로 2번 아이템
```

```
        item_num = 2
```

```
    else: # 2/5 확률로 3번 아이템
```

```
        item_num = 3
```

```
    item = Item(rock2.rect.x, rock2.rect.y, item_num)
```

```
    items.add(item)
```

```
    shot_count += 1
```

```
    total_time += 1
```

```
# 미사일이 장애물 맞으면 장애물, 미사일 사라짐, 카운터 올라감 설정
```

```
# 장애물 폭발 이미지, 장애물 x,y 값 설정
```

1번코드는 백신 미사일이 바이러스를 맞으면 일어나는 현상과 늘어나는 시간을 구현 코드,

2번 코드는 각 아이템의 드롭 확률을 설정해 둔 코드입니다.





## 게임 코드

```
rocks.update()
rocks.draw(screen)
rocks2.update()
rocks2.draw(screen)
missiles.update()
missiles.draw(screen)
fighter.update()
fighter.draw(screen, shield)
items.update()
items.draw(screen)
pygame.display.flip()
```

# 각 항목들을 위에 내용으로 업데이트 한다는 뜻 그 걸 flip() 전체반영 해주는거래

기본적인 효과를 모두 넣었다면  
update 와 draw를 사용하여  
화면에 표현되게 코드를 입력  
전체 반영을 해 줍니다.





# 게임 코드

```
if int(timer) <= 0: # 타임 오버
    pygame.mixer_music.stop()
    occur_explosion(screen, fighter.rect.x, fighter.rect.y)
    pygame.display.update()
    gameover_sound.play()
    sleep(1)
    done = True
```

1번코드는 타임 오버가 되면 게임을 패배  
한다는 조건을 넣어준 코드이며,

2번코드는 마지막으로 메인화면 구성 및  
게임 이름 을 저장하여 play or quit 로 실행  
아님 게임종료를 넣은 함수 입니다.

이것으로 코로나 헌터 코드 설명을 종료  
하겠습니다. 즐겁하세요 ^^7

```
def main():
    global screen

    pygame.init()
    screen = pygame.display.set_mode((WINDOW_WIDTH, WINDOW_HEIGHT))
    pygame.display.set_caption('코로나헌터')

    action = 'game_menu'
    while action != 'quit':
        if action == 'game_menu':
            action = game_menu()
        elif action == 'play':
            action = game_loop()

    pygame.quit()

    # 메인메뉴 함수추가로 마지막 screen 역할과 게임메뉴창, 나가기 버튼 등을 활성화함

if __name__ == "__main__":
    main()
# 메인일 경우 게임 실행하는 거
```





## 4. 게임 시연







## Q&A

-질문 및 궁금증 해결해 드립니다-







THANK YOU  
감사합니다 !

