

13006107

Python Project Proposal

61090038

Thirayut Sriphochang

SE

IC

KMITL

Project Proposal

Project Title

Operation SEVEN.

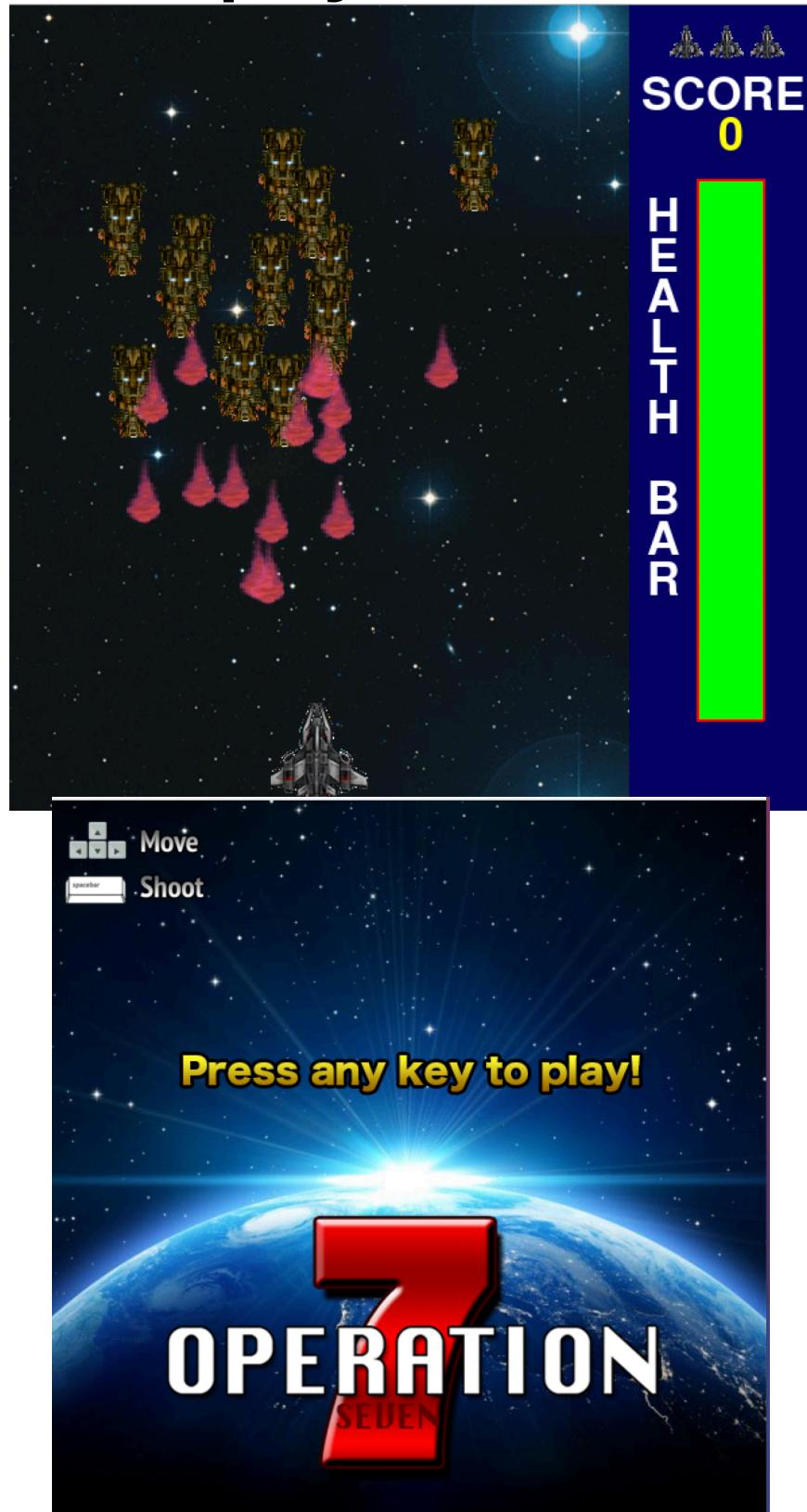
Project Description

“Operation SEVEN” is 2D space shooter arcade game . This game set player as a spaceship, and the objective is you have to make your score by shoot all of your enemies that are in your way so they don’t collide with your ship and kill you. Following that, you have 3 lives and there are 100 healths in each live. Finally, you have to do your best to make high score as many as you can.

Project Requirement

This project uses Pygame module

Gameplay screenshots



Thirayut Sripochang 61090038

Code

```
import pygame
import random
import math
from os import path

img_dir = path.join(path.dirname(__file__), 'Sprites')
snd_dir = path.join(path.dirname(__file__), 'Sounds')

WIDTH = 460
WIDTH2 = 600
HEIGHT = 600
FPS = 120
POWERUP_TIME = 5000

# Define colors
WHITE = (255, 255, 255)
BLACK = (0, 0, 0)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)
YELLOW = (255, 255, 0)
DARK_BLUE = (0, 0, 102)

# Initialization pygame
pygame.init()
pygame.mixer.init()
screen = pygame.display.set_mode((WIDTH2, HEIGHT))
pygame.display.set_caption("Operation SEVEN")
clock = pygame.time.Clock()

font_name = pygame.font.match_font('arial')
def draw_text(surf, text, size, x, y, colour):
    font = pygame.font.Font(font_name, size)
    text_surface = font.render(text, True, colour)
    text_rect = text_surface.get_rect()
    text_rect.midtop = (x, y)
    surf.blit(text_surface, text_rect)

def newmob():
    m = Mob()
    all_sprites.add(m)
    mobs.add(m)

def UI(surf, x, y):
    LENGTH = 140
    HEIGHT = 600
    rect = pygame.Rect(x, y, LENGTH, HEIGHT)
    pygame.draw.rect(surf, WHITE, rect, 2)
    pygame.draw.rect(surf, DARK_BLUE, rect)
```

```
def draw_health_bar(surf,x,y,pct):
    if pct < 0:
        pct = 0
    BAR_LENGTH = 50
    BAR_HEIGHT = -400
    fill = (pct/100) * BAR_HEIGHT
    outline_rect = pygame.Rect(x,y,BAR_LENGTH,BAR_HEIGHT)
    fill_rect = pygame.Rect(x,y,BAR_LENGTH,fill)
    pygame.draw.rect(surf,GREEN,fill_rect)
    pygame.draw.rect(surf,RED,outline_rect,2)

def draw_lives(surf,x,y,lives,img):
    for i in range(lives):
        img_rect = img.get_rect()
        img_rect.x = x+30*i
        img_rect.y = y
        surf.blit(img,img_rect)

class Player(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
##        self.image = pygame.transform.scale(player_img, (50,38))
        self.image = player_img
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.radius = 15
##        pygame.draw.circle(self.image,RED,self.rect.center,self.radius)
        self.rect.centerx = (WIDTH/2)
        self.rect.bottom = HEIGHT-10
        self.speedx = 0
        self.health = 100
        self.shoot_delay = 250
        self.last_shot = pygame.time.get_ticks()
        self.lives = 3
        self.hidden = False
        self.hide_timer = pygame.time.get_ticks()
        self.power = 1
        self.power_time = pygame.time.get_ticks()
    def update(self):
        #timeout for powerups
        if(self.power >= 2 and pygame.time.get_ticks() - self.power_time > POWERUP_TIME):
            self.power -= 1
            self.power_time = pygame.time.get_ticks()

        if self.hidden and pygame.time.get_ticks()-self.hide_timer > 1000:
            self.hidden = False
            self.rect.centerx = WIDTH/2
            self.rect.bottom = HEIGHT-10
            self.power = 1
```

```
self.speedx = 0
keystate = pygame.key.get_pressed()
if keystate[pygame.K_LEFT]:
    self.speedx = -8
if keystate[pygame.K_RIGHT]:
    self.speedx = 8
if keystate[pygame.K_SPACE]:
    self.shoot()
self.rect.x += self.speedx
if self.rect.right > WIDTH:
    self.rect.right = WIDTH
if self.rect.left < 0:
    self.rect.left = 0
def powerup(self):
    self.power += 1
    self.power_time = pygame.time.get_ticks()

def shoot(self):
    now = pygame.time.get_ticks()
    if now - self.last_shot > self.shoot_delay:
        self.last_shot = now
        if self.power == 1:
            bullet = Bullet(self.rect.centerx, self.rect.top)
            all_sprites.add(bullet)
            bullets.add(bullet)
            shoot_sound.play()
        if self.power >= 2:
            bullet1 = Bullet(self.rect.left, self.rect.centery)
            bullet2 = Bullet(self.rect.right, self.rect.centery)
            all_sprites.add(bullet1)
            all_sprites.add(bullet2)
            bullets.add(bullet1)
            bullets.add(bullet2)
            shoot_sound.play()

def hide(self):
    self.hidden = True
    self.hide_timer = pygame.time.get_ticks()
    self.rect.center = (WIDTH/2, HEIGHT+200)

class Mob(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(enemies_img, (35,69))
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(100,300)
        self.rect.y = random.randrange(-100,-40)
```

```

        self.speedy = random.randrange(4,8)
        self.speedx = random.randrange(-2,3)
        self.last_shot = pygame.time.get_ticks()
        self.shoot_delay = random.randrange(1500,3000)
    def checkbounds(self):
        if self.rect.left < 40:
            self.rect.left = 40

        if self.rect.right > WIDTH-30:
            self.rect.right = WIDTH-30

    def update(self):
        self.checkbounds()
        self.rect.x += self.speedx
        self.rect.y += self.speedy
        self.shoot()
        if self.rect.top > HEIGHT + 10 :
            self.rect.x = random.randrange(WIDTH-self.rect.width)
            self.rect.y = random.randrange(-100,-40)
            self.speedy = random.randrange(7,15)
    def shoot(self):
        now = pygame.time.get_ticks()
        if now - self.last_shot > self.shoot_delay:
            self.last_shot = now
            bullet = Enermybullet(self.rect.centerx, self.rect.bottom)
            all_sprites.add(bullet)
            enmbullet.add(bullet)
            shoot_sound.play()

class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(bullet_img, (25,48))
        self.image.set_colorkey(WHITE)
        self.rect = self.image.get_rect()
        self.rect.bottom = y
        self.rect.centerx = x
        self.speedy = -10

    def update(self):
        self.rect.y += self.speedy
        # remove when off the top
        if self.rect.bottom < 0:
            self.kill()

class Enermybullet(pygame.sprite.Sprite):
    def __init__(self, x, y):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(enmbullet_img, (25,48))
        self.image.set_colorkey(WHITE)

```

```
    self.rect = self.image.get_rect()
    self.radius = 15
##    pygame.draw.circle(self.image,RED,self.rect.center,self.radius)
    self.rect = self.image.get_rect()
    self.rect.bottom = y
    self.rect.centerx = x
    self.speedy = 15

def update(self):
    self.rect.y += self.speedy
    # remove when off the bottom
    if self.rect.bottom > HEIGHT:
        self.kill()

class Explosion(pygame.sprite.Sprite):
    def __init__(self,center, size):
        pygame.sprite.Sprite.__init__(self)
        self.size = size
        self.image = exp_ani[self.size][0]
        self.rect = self.image.get_rect()
        self.rect.center = center
        self.frame = 0
        self.last_update = pygame.time.get_ticks()
        self.frame_rate = 50
    def update(self):
        now = pygame.time.get_ticks()
        if now - self.last_update > self.frame_rate:
            self.last_update = now
            self.frame += 1
            if self.frame == len(exp_ani[self.size]):
                self.kill()
            else :
                center = self.rect.center
                self.image = exp_ani[self.size][self.frame]
                self.rect = self.image.get_rect()
                self.rect.center = center

class Pow(pygame.sprite.Sprite):
    def __init__(self, center):
        pygame.sprite.Sprite.__init__(self)
        self.type = random.choice(['shield','gun'])
        self.image = powerup_images[self.type]
        self.image.set_colorkey(BLACK)
        self.rect = self.image.get_rect()
        self.rect.center = center
        self.speedy = 5

    def update(self):
        self.rect.y += self.speedy
        # remove when off the top
```

```

    if self.rect.top > HEIGHT:
        self.kill()

def show_go_screen():
    screen.blit(menu,(0,0))
    pygame.display.flip()
    waiting = True
    while waiting:
        clock.tick(FPS)
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
            if event.type == pygame.KEYUP:
                start_sound.play()
                waiting = False

def games_over():
    screen.blit(game_over_bg,(0,0))
    pygame.display.flip()
    waiting = True
    while waiting:
        clock.tick(FPS)
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
            if event.type == pygame.KEYUP:
                show_go_screen()
                waiting = False

# Load all game sprites
beta_menu = pygame.image.load(path.join(img_dir, "menu.png")).convert()
menu= pygame.transform.scale(beta_menu,(600,600))
bg = pygame.image.load(path.join(img_dir, "in-game_bg.png")).convert()
bg_new = pygame.transform.scale(bg,(1200,1200))
bgY = -1200
game_over_bg = pygame.image.load(path.join(img_dir, "game_over.png")).convert()
bg_rect = bg_new.get_rect()
player_img = pygame.image.load(path.join(img_dir, "spaceship.png")).convert()
enermies_img = pygame.image.load(path.join(img_dir, "13.png")).convert()
bullet_img = pygame.image.load(path.join(img_dir, "bullet.png"))
enmbullet_img = pygame.image.load(path.join(img_dir, "enermy_bullet.png"))
player_lives_img = pygame.transform.scale(player_img,(25,25))
player_lives_img.set_colorkey(BLACK)
exp_ani = {}
exp_ani['all'] = []
exp_ani['player'] = []
exp_ani['small'] = []
for i in range(16):
    filename = 'ex{}.png'.format(i)

```

```
img = pygame.image.load(path.join(img_dir, filename))
img.set_colorkey(BLACK)
img_new = pygame.transform.scale (img, (100,100))
exp_ani['all'].append(img_new)
for i in range(16):
    filename = 'ex{}.png'.format(i)
    img = pygame.image.load(path.join(img_dir, filename))
    img.set_colorkey(BLACK)
    img_new = pygame.transform.scale (img, (50,50))
    exp_ani['small'].append(img_new)
for i in range(9):
    filename = 'sonicExplosion0{}.png'.format(i)
    img = pygame.image.load(path.join(img_dir, filename)).convert()
    img.set_colorkey(BLACK)
    img_new = pygame.transform.scale (img, (100,100))
    exp_ani['player'].append(img_new)
powerup_images = {}
powerup_images['shield'] = pygame.image.load(path.join(img_dir, 'shield_gold.png')).convert()
powerup_images['gun'] = pygame.image.load(path.join(img_dir, 'bolt_gold.png')).convert()
# Load sounds
shoot_sound = pygame.mixer.Sound(path.join(snd_dir, '7.wav'))
powerup_sound = pygame.mixer.Sound(path.join(snd_dir, 'spell1_0.wav'))
start_sound = pygame.mixer.Sound(path.join(snd_dir, 'chipquest.wav'))
expl_sound = pygame.mixer.Sound(path.join(snd_dir, 'explosion.wav'))
player_expl = pygame.mixer.Sound(path.join(snd_dir, 'rumble1.ogg'))
pygame.mixer.music.load(path.join(snd_dir, 'bg_music.ogg'))
pygame.mixer.music.set_volume(1)

pygame.mixer.music.play(loops =-1)

# Game loop
start_screen = True
running = True
while running:
    if start_screen:
        show_go_screen()
        start_screen = False
    all_sprites = pygame.sprite.Group()
    player = Player()
    mobs = pygame.sprite.Group()
    bullets = pygame.sprite.Group()
    enmbullet = pygame.sprite.Group()
    powerups = pygame.sprite.Group()
    all_sprites.add(player)
    for i in range(15):
        newmob()

        score = 0
    clock.tick(FPS)
    for event in pygame.event.get():
```

```
if event.type == pygame.QUIT:
    running = False

#sprite update
all_sprites.update()

#enemies bullet hit player
hits = pygame.sprite.spritecollide(player,enmbullet, True,pygame.sprite.collide_circle)
for hit in hits:
    player.health -= 20
    player_expl.play()
    death_explosion = Explosion(player.rect.center,'small')
    all_sprites.add(death_explosion)
    if player.health <= 0:
        player_expl.play()
        death_explosion = Explosion(player.rect.center,'player')
        all_sprites.add(death_explosion)
        player.hide()
        player.lives -= 1
        player.health = 100
#bullet hit mob
hits = pygame.sprite.groupcollide(mobs, bullets, True, True)

for hit in hits:
    score += 10
    expl_sound.play()
    expl = Explosion(hit.rect.center, 'all')
    all_sprites.add(expl)
    if random.random() > 0.9:
        pow = Pow(hit.rect.center)
        all_sprites.add(pow)
        powerups.add(pow)
    newmob()

#mobs hit player
hits = pygame.sprite.spritecollide(player, mobs, True, pygame.sprite.collide_circle)
for hit in hits:
    player.health -= 35
    expl_sound.play()
    expl = Explosion(hit.rect.center, 'all')
    all_sprites.add(expl)
    newmob()
    if player.health <= 0:
        player_expl.play()
        death_explosion = Explosion(player.rect.center,'player')
        all_sprites.add(death_explosion)
```

```
death_explosion = Explosion(player.rect.center,'player')
all_sprites.add(death_explosion)
player.hide()
player.lives -= 1
player.health = 100
#player died and explosion finished
if player.lives == 0 and not death_explosion.alive():
    games_over()
    start_screen = True

#player hit powerups
hits = pygame.sprite.spritecollide(player, powerups, True)
for hit in hits:
    if hit.type == 'shield':
        player.health += 5
        if player.health >= 100:
            player.health = 100
            powerup_sound.play()
    if hit.type == 'gun':
        player.powerup()
        powerup_sound.play()

# Draw
screen.fill(BLACK)
rel_bgY = bgY % bg_new.get_rect().height
screen.blit(bg_new, (0,rel_bgY - bg_new.get_rect().height))
if rel_bgY < HEIGHT:
    screen.blit(bg_new, (0,rel_bgY))
bgY +=4
all_sprites.draw(screen)
UI(screen,WIDTH,0)
draw_lives(screen,WIDTH+30,15,player.lives,player_lives_img)
draw_text(screen,'SCORE',50,WIDTH+70,50,WHITE)
draw_text(screen,str(score),50,WIDTH+75,80,YELLOW)
draw_text(screen,'H',50,WIDTH+25,140,WHITE)
draw_text(screen,'E',50,WIDTH+25,170,WHITE)
draw_text(screen,'A',50,WIDTH+25,200,WHITE)
draw_text(screen,'L',50,WIDTH+25,230,WHITE)
draw_text(screen,'T',50,WIDTH+25,260,WHITE)
draw_text(screen,'H',50,WIDTH+25,290,WHITE)
draw_text(screen,'B',50,WIDTH+25,350,WHITE)
draw_text(screen,'A',50,WIDTH+25,380,WHITE)
draw_text(screen,'R',50,WIDTH+25,410,WHITE)
draw_health_bar(screen,WIDTH+50,530,player.health)

# Flip the display,after drawing everything
pygame.display.flip()

pygame.quit()
```