**Game Design Document(GDD)**

**Random Falling Aliens**

**1.Game Overview**

**Game Title**

Random Falling Aliens

**Genre:** Arcade / Action

**Platform:** PC (Windows)

**Engine:** Pygame (Python)

**Summary:**

"Random Falling Aliens" is a retro-style 2D arcade game. The player controls a spaceship at the bottom of the screen, shooting descending aliens to earn points. The game features animated backgrounds, sound effects, start/end screens, and a scoring system, providing quick and engaging gameplay.

**2. Objectives**

* Avoid collisions with aliens.
* Shoot down aliens to increase score.
* Stop aliens from reaching the bottom.
* Restart or quit after game over.

**3. Gameplay Mechanics**

**Controls:**

* Left Arrow: move left
* Right Arrow: move right
* Space / Up Arrow: shoot
* Enter: start game / restart
* Mouse clicks: interact with buttons

**Core Loop:**

* Player moves left/right along the bottom.
* Aliens spawn at the top at intervals and fall.
* Player shoots bullets upward to destroy aliens.
* Collisions destroy aliens, earning points.
* Aliens reaching the bottom decrease score and cause game over if they touch player.
* Background animated for visual dynamism.

**Score System**

**Table**

|  |  |
| --- | --- |
| **Action** | **Points** |
| Alien destroyed | +10 |
| Alien reaches bottom | -5 |

**4. Visual Elements**

**Table**

|  |  |
| --- | --- |
| **Element** | **Description** |
| Background | Animated GIF (bg image animated.gif), static start and game over backgrounds. |
| Player | Image (Player img.png) or green rectangle placeholder. |
| Alien | Image (aline img.png) or red rectangle. |
| Bullets | Yellow rectangle for projectiles. |
| UI | Score, start/play button, game over prompt, yes/no buttons |

**5. Audio**

* Shooting sound (shooter sound.wav)
* Alien-kill sound (enemy kill sound.wav)
* Optional: No sounds if files are missing.

**6. User Interface**

* Start Screen
* Title: "Random Falling Aliens"
* Button: "PLAY NOW"
* Prompt: "Press ENTER to Start"
* Main Screen
* Animated background
* Player sprite
* Falling aliens
* Score display
* Game Over Screen
* Background image
* Message: "Do you want to play again?"
* Buttons: "YES" and "NO"

**7. Game States**

* Start Screen: Await ENTER or click to start.
* Gameplay: Active game.
* Game Over: Show options to replay or exit.

**8. Technical Details**

**Table**

|  |  |
| --- | --- |
| **Aspect** | **Details** |
| Resolution | 800x600 pixels |
| Frame Rate | 60 FPS |
| Controls | Arrow keys, space/up for shooting, Enter for start/restart |
| Dependencies | Pygame, PIL (Pillow) |

**9. Complete Program Code**

python

import pygame

import sys

import random

from PIL import Image

# Initialize Pygame

pygame.init()

pygame.mixer.init()

# Screen setup

WIDTH, HEIGHT = 800, 600

screen = pygame.display.set\_mode((WIDTH, HEIGHT))

pygame.display.set\_caption("Random Falling Aliens")

# Colors

WHITE = (255, 255, 255)

BLACK = (0, 0, 0)

GREEN = (0, 255, 0)

# Load animated GIF frames

def load\_gif\_frames(gif\_path, size):

frames = []

try:

gif = Image.open(gif\_path)

for \_ in range(gif.n\_frames):

gif.seek(gif.tell() + 1)

frame\_image = gif.convert("RGB")

frame\_surface = pygame.image.fromstring(

frame\_image.tobytes(), frame\_image.size, frame\_image.mode

).convert()

frame\_surface = pygame.transform.scale(frame\_surface, size)

frames.append(frame\_surface)

except Exception as e:

print(f"Error loading GIF: {e}")

fallback = pygame.Surface(size)

fallback.fill(BLACK)

frames.append(fallback)

return frames

# Load background GIF frames

bg\_frames = load\_gif\_frames("bg image animated.gif", (WIDTH, HEIGHT))

bg\_frame\_index = 0

bg\_frame\_timer = 0

bg\_frame\_delay = 100 # ms between frames

# Start screen background

try:

start\_bg\_image = pygame.image.load("Bg image.jpg").convert()

start\_bg\_image = pygame.transform.scale(start\_bg\_image, (WIDTH, HEIGHT))

except:

start\_bg\_image = pygame.Surface((WIDTH, HEIGHT))

start\_bg\_image.fill((30, 30, 60))

# Game Over screen background

try:

game\_over\_bg\_image = pygame.image.load("game over.jpg").convert()

game\_over\_bg\_image = pygame.transform.scale(game\_over\_bg\_image, (WIDTH, HEIGHT))

except:

game\_over\_bg\_image = pygame.Surface((WIDTH, HEIGHT))

game\_over\_bg\_image.fill((50, 0, 0))

# Player

try:

player\_img = pygame.image.load("Player img.png").convert\_alpha()

except:

player\_img = pygame.Surface((60, 40))

player\_img.fill(GREEN)

player\_rect = player\_img.get\_rect(midbottom=(WIDTH // 2, HEIGHT - 20))

player\_speed = 5

# Alien

try:

alien\_img = pygame.image.load("aline img.png").convert\_alpha()

except:

alien\_img = pygame.Surface((40, 30))

alien\_img.fill((255, 0, 0))

# Bullet

bullet\_img = pygame.Surface((5, 10))

bullet\_img.fill((255, 255, 0))

bullet\_speed = -7

bullets = []

# Sounds

try:

shoot\_sound = pygame.mixer.Sound("shooter sound.wav")

except:

shoot\_sound = None

try:

enemy\_killed\_sound = pygame.mixer.Sound("enemy kill sound.wav")

except:

enemy\_killed\_sound = None

# Fonts

font = pygame.font.SysFont("arial", 28)

big\_font = pygame.font.SysFont("arial", 60)

# Game state variables

aliens = []

alien\_fall\_speed = 2

alien\_spawn\_interval = 1000

last\_alien\_spawn\_time = pygame.time.get\_ticks()

score = 0

game\_over = False

game\_started = False

shoot\_cooldown = 200

last\_shot\_time = 0

clock = pygame.time.Clock()

def draw\_text(text, x, y, size=28, color=WHITE, center=False):

used\_font = big\_font if size > 30 else font

surface = used\_font.render(text, True, color)

rect = surface.get\_rect()

if center:

rect.center = (x, y)

else:

rect.topleft = (x, y)

screen.blit(surface, rect)

def draw\_button(rect, text, text\_color=BLACK, button\_color=GREEN):

pygame.draw.rect(screen, button\_color, rect)

draw\_text(text, rect.centerx, rect.centery, 30, text\_color, center=True)

def spawn\_alien():

x = random.randint(0, WIDTH - 40)

alien\_rect = pygame.Rect(x, -30, 40, 30)

aliens.append(alien\_rect)

def reset\_game():

global aliens, bullets, score, player\_rect, last\_shot\_time, last\_alien\_spawn\_time, game\_over

aliens.clear()

bullets.clear()

score = 0

player\_rect.midbottom = (WIDTH // 2, HEIGHT - 20)

last\_shot\_time = pygame.time.get\_ticks()

last\_alien\_spawn\_time = pygame.time.get\_ticks()

game\_over = False

# Main game loop

while True:

dt = clock.tick(60)

current\_time = pygame.time.get\_ticks()

screen.fill(BLACK)

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

# Start game or restart

if not game\_started:

if event.type == pygame.KEYDOWN and event.key == pygame.K\_RETURN:

game\_started = True

elif game\_over:

if event.type == pygame.MOUSEBUTTONDOWN:

if 'yes\_button' in locals() and yes\_button.collidepoint(event.pos):

reset\_game()

elif 'no\_button' in locals() and no\_button.collidepoint(event.pos):

pygame.quit()

sys.exit()

if event.type == pygame.KEYDOWN and event.key == pygame.K\_RETURN:

reset\_game()

# Start Screen

if not game\_started:

screen.blit(start\_bg\_image, (0, 0))

draw\_text("Random Falling Aliens", WIDTH // 2, 100, 60, center=True)

play\_button = pygame.Rect(WIDTH // 2 - 100, HEIGHT // 2 - 25, 200, 50)

pygame.draw.rect(screen, GREEN, play\_button)

draw\_text("PLAY NOW", WIDTH // 2, HEIGHT // 2, 30, BLACK, center=True)

draw\_text("Press ENTER to Start", WIDTH // 2, HEIGHT // 2 + 80, 24, center=True)

elif not game\_over:

# Animate background

bg\_frame\_timer += dt

if bg\_frame\_timer > bg\_frame\_delay:

bg\_frame\_timer = 0

bg\_frame\_index = (bg\_frame\_index + 1) % len(bg\_frames)

screen.blit(bg\_frames[bg\_frame\_index], (0, 0))

# Player movement

keys = pygame.key.get\_pressed()

if keys[pygame.K\_LEFT] and player\_rect.left > 0:

player\_rect.x -= player\_speed

if keys[pygame.K\_RIGHT] and player\_rect.right < WIDTH:

player\_rect.x += player\_speed

# Shooting

if (keys[pygame.K\_SPACE] or keys[pygame.K\_UP]) and (current\_time - last\_shot\_time > shoot\_cooldown):

bullet\_rect = bullet\_img.get\_rect(midbottom=player\_rect.midtop)

bullets.append(bullet\_rect)

last\_shot\_time = current\_time

if shoot\_sound:

shoot\_sound.play()

# Spawn aliens periodically

if current\_time - last\_alien\_spawn\_time > alien\_spawn\_interval:

spawn\_alien()

last\_alien\_spawn\_time = current\_time

# Move bullets

for bullet in bullets[:]:

bullet.y += bullet\_speed

if bullet.bottom < 0:

bullets.remove(bullet)

# Move aliens

for alien in aliens[:]:

alien.y += alien\_fall\_speed

if alien.colliderect(player\_rect):

game\_over = True

elif alien.top > HEIGHT:

aliens.remove(alien)

score -= 5 # penalty for reaching bottom

# Bullet and alien collision

for bullet in bullets[:]:

for alien in aliens[:]:

if bullet.colliderect(alien):

bullets.remove(bullet)

aliens.remove(alien)

score += 10

if enemy\_killed\_sound:

enemy\_killed\_sound.play()

break

# Draw everything

screen.blit(player\_img, player\_rect)

for alien in aliens:

screen.blit(alien\_img, alien)

for bullet in bullets:

screen.blit(bullet\_img, bullet)

draw\_text(f"Score: {score}", 10, 10)

else:

# Game Over Screen

screen.blit(game\_over\_bg\_image, (0, 0))

draw\_text(" ", WIDTH // 2, HEIGHT // 2 - 100, 60, center=True)

draw\_text("Do you want to play again?", WIDTH // 2, HEIGHT // 3 -30, 28, center=True)

yes\_button = pygame.Rect(WIDTH // 2 - 120, HEIGHT // 3 + 10, 100, 40)

no\_button = pygame.Rect(WIDTH // 2 + 20, HEIGHT // 3 + 10, 100, 40)

draw\_button(yes\_button, "YES")

draw\_button(no\_button, "NO")

# Handle button clicks

if pygame.mouse.get\_pressed()[0]:

mouse\_pos = pygame.mouse.get\_pos()

if yes\_button.collidepoint(mouse\_pos):

reset\_game()

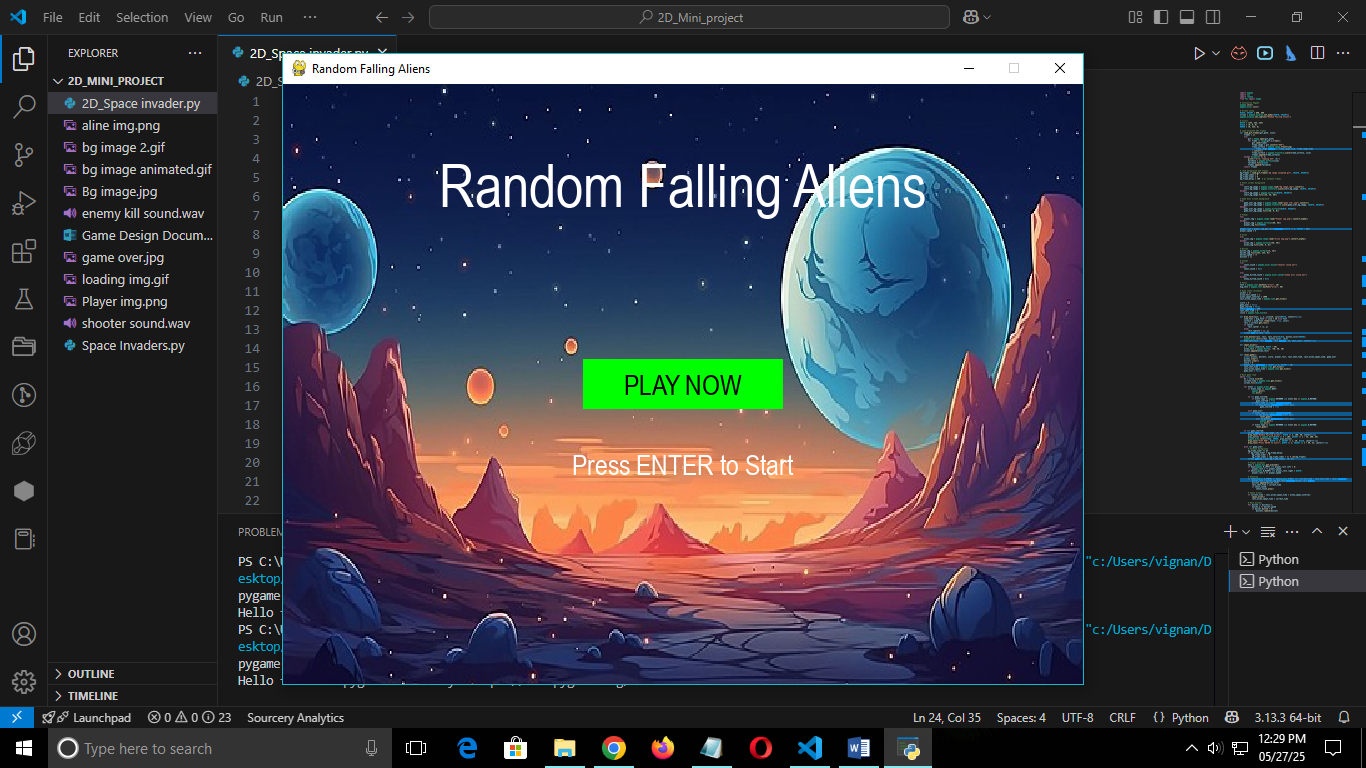
elif no\_button.collidepoint(mouse\_pos):

pygame.quit()

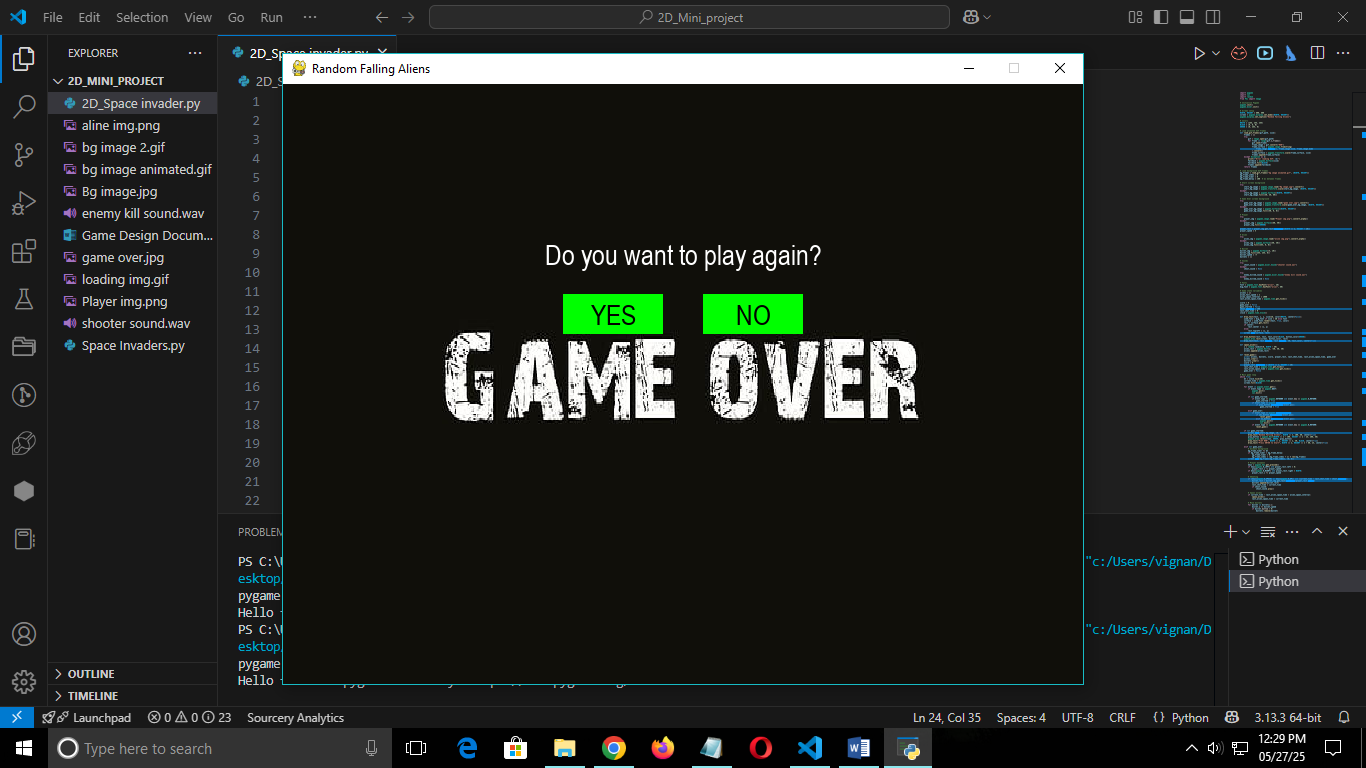
sys.exit()

pygame.display.flip()

**10.Output**

****

****

****

**11. Future Improvements**

* Add sound effects and background music.
* Introduce levels or increasing difficulty.
* Add power-ups or different alien types.
* Improve visuals and animations.