***Game Design Document (GDD)***

**Game Title:**

Car Racing Game

**Game Overview:**

A top-down car racing game where the player controls a car moving left and right to avoid oncoming obstacles (other cars) on a three-lane road. The goal is to survive as long as possible without crashing.

**Platform:**

PC (Windows/Linux/macOS)  
Implemented using Pygame (Python)

**Genre:**

Arcade / Endless Runner / Avoidance Game

**Target Audience:**

- Casual players, all ages  
- Fans of simple, quick reflex games

**Gameplay Mechanics:**

**Player Controls:**

- Move the car left and right using the arrow keys (Left/Right).  
- Movement is restricted within the boundaries of the road.

**Objective:**

- Avoid collisions with incoming cars (obstacles).  
- Survive as long as possible.

**Obstacles:**

- Cars spawn randomly in one of three lanes at the top of the screen.  
- They move downward toward the player at a constant speed.  
- If a car collides with the player’s car, the game ends.

**Road Lines:**

- Vertical road divider lines scroll downward to simulate movement.

**Visuals & Assets:**

**Player Car:**

- Image: car.png  
- Size: 100x100 pixels (scaled)

**Obstacles (Other Cars):**

- Image: car1.png  
- Size: 100x100 pixels (scaled)

**Road:**

- Black background with grey vertical dashed lines as road divider.

**Audio:**

- (Optional) Background music and sound effects for crashes and movement.

**User Interface**

**Game Over:**

Display “Game Over!” text in red at the center of the screen when player collides with obstacle.

**Technical Details:**

**Screen Resolution:**

400 x 600 pixels

Frame Rate

60 FPS (frames per second)

**Controls:**

| Key | Action |  
|-------------|-------------------|  
| Left Arrow | Move player left |  
| Right Arrow | Move player right |

**Game Flow:**

1. Player starts in the center lane at the bottom.  
2. Obstacles spawn from the top at regular intervals.  
3. Player dodges obstacles by moving left or right.  
4. Game ends when a collision is detected.  
5. Player can close the window to exit.

**Future Enhancements :**

- Add scoring based on time survived or obstacles dodged.  
- Increase difficulty over time (increase obstacle speed or spawn rate).  
- Add sound effects and background music.  
- Add a restart or main menu screen.  
- Add animations or particle effects on collision.  
- Add different types of obstacles and power-ups.

**Project code:**

import pygame

import random

import sys

pygame.init()

WIDTH, HEIGHT = 400, 400

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

pygame.display.set\_caption("Car Racing Game")

clock = pygame.time.Clock()

FPS = 60

WHITE = (255, 255, 255)

BLACK = (0, 0, 0)

try:

background\_img = pygame.image.load('road1.png').convert()

player\_img = pygame.image.load('car.png').convert\_alpha()

obstacle\_img = pygame.image.load('car1.png').convert\_alpha()

except:

print("Make sure 'road1.png', 'car.png', and 'car1.png' are in the same folder.")

pygame.quit()

sys.exit()

background\_img = pygame.transform.scale(background\_img, (WIDTH, HEIGHT))

player\_img = pygame.transform.scale(player\_img, (50, 50))

obstacle\_img = pygame.transform.scale(obstacle\_img, (50, 50))

lanes = [75, 175, 275]

player\_rect = player\_img.get\_rect()

player\_rect.center = (WIDTH // 2, HEIGHT - 80)

obstacles = []

SPAWN\_OBSTACLE = pygame.USEREVENT + 1

pygame.time.set\_timer(SPAWN\_OBSTACLE, 1500)

speed = 4

font = pygame.font.SysFont(None, 40)

big\_font = pygame.font.SysFont(None, 50)

game\_over = False

game\_started = False

score = 0

start\_time = 0

bg\_y1 = 0

bg\_y2 = -HEIGHT

def show\_start\_screen():

screen.blit(background\_img, (0, 0))

title = big\_font.render("Car Racing Game", True, WHITE)

prompt = font.render("Press SPACE to Start", True, WHITE)

screen.blit(title, title.get\_rect(center=(WIDTH // 2, HEIGHT // 2 - 40)))

screen.blit(prompt, prompt.get\_rect(center=(WIDTH // 2, HEIGHT // 2 + 20)))

pygame.display.flip()

def show\_game\_over():

over = big\_font.render("Game Over!", True, (255, 0, 0))

prompt = font.render("Press R to Try Again", True, WHITE)

screen.blit(over, over.get\_rect(center=(WIDTH // 2, HEIGHT // 2 - 30)))

screen.blit(prompt, prompt.get\_rect(center=(WIDTH // 2, HEIGHT // 2 + 30)))

def draw\_score(seconds):

score\_text = font.render(f"Score: {seconds:.1f}", True, WHITE)

screen.blit(score\_text, (10, 10))

running = True

while running:

if not game\_started:

show\_start\_screen()

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

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

game\_started = True

game\_over = False

obstacles.clear()

player\_rect.center = (WIDTH // 2, HEIGHT - 80)

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

score = 0

bg\_y1 = 0

bg\_y2 = -HEIGHT

continue

screen.blit(background\_img, (0, bg\_y1))

screen.blit(background\_img, (0, bg\_y2))

bg\_y1 += speed

bg\_y2 += speed

if bg\_y1 >= HEIGHT:

bg\_y1 = -HEIGHT

if bg\_y2 >= HEIGHT:

bg\_y2 = -HEIGHT

if not game\_over:

for obs in obstacles[:]:

obs.y += speed

screen.blit(obstacle\_img, obs)

if obs.colliderect(player\_rect):

game\_over = True

if obs.top > HEIGHT:

obstacles.remove(obs)

keys = pygame.key.get\_pressed()

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

player\_rect.x -= 5

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

player\_rect.x += 5

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

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

score = (current\_time - start\_time) / 1000

draw\_score(score)

else:

show\_game\_over()

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

if event.type == SPAWN\_OBSTACLE and not game\_over:

lane = random.choice(lanes)

new\_obs = obstacle\_img.get\_rect(midtop=(lane + 25, -100))

obstacles.append(new\_obs)

keys = pygame.key.get\_pressed()

if game\_over and keys[pygame.K\_r]:

game\_over = False

obstacles.clear()

player\_rect.center = (WIDTH // 2, HEIGHT - 80)

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

score = 0

bg\_y1 = 0

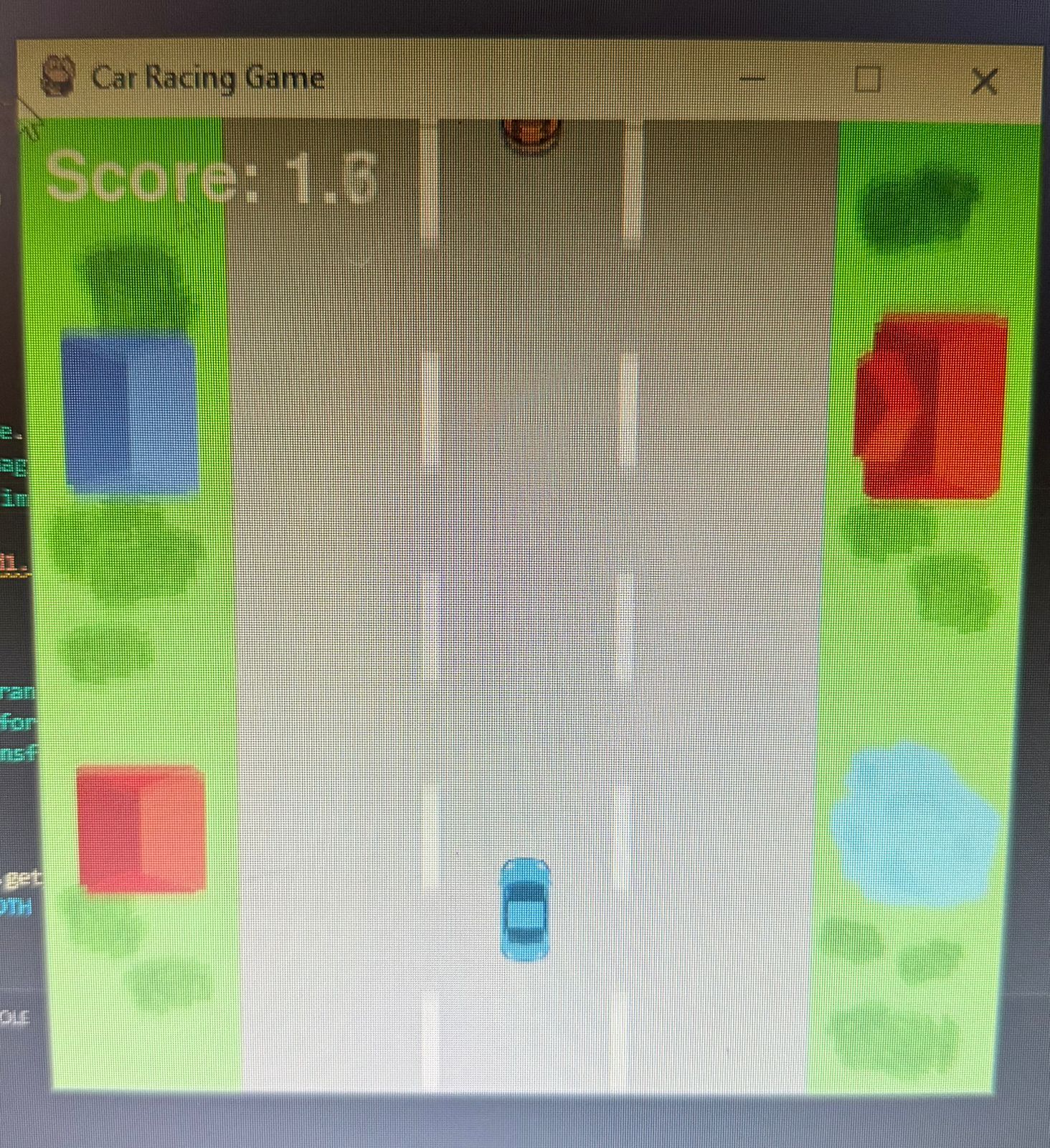
bg\_y2 = -HEIGHT

pygame.display.flip()

clock.tick(FPS)

pygame.quit()

**output:**

****