Annexure-6

Copyright Submission Form (Computer Software)

Title of the Software/Application: ***Ball Breaker***

Name of authors: Dr. Bavesh kumar

Name of Student/Scholar: Sai Mouli Chukka

Registration Number: 12404236

Program Name: B. Tech. in ECE

Name/ UID of Supervisor: Dr Bhavesh kumar

School: **School of Electronoics and Electrical Engineering**

Name and UID of the all co-authors/Co-Supervisor:

Affiliation details (with Pin code) of co-authors/Co- Supervisor (if External):

Language of the Work: Python

Summary, Uniqueness and Utility of work: (900 words)

Source Code/Object Code:

```
import pygame
import sys
import random
import os

pygame.init()

WIDTH, HEIGHT = 800, 600
WHITE, RED, BLUE = (255, 255, 255), (255, 0, 0), (0, 0, 255)
PADDLE_WIDTH, PADDLE_HEIGHT, BALL_SIZE = 100, 10, 10
BRICK_WIDTH, BRICK_HEIGHT, COLS, ROWS = 75, 20, 7, 5
HIGH_SCORE_FILE = "high_score.txt"

screen = pygame.display.set_mode((WIDTH, HEIGHT))
```

```
pygame.display.set_caption("Breakout Game")


def load_high_score():
    return int(open(HIGH_SCORE_FILE).read().strip()) if os.path.exists(HIGH_SCORE_FILE) else 0


def save_high_score(score):
    with open(HIGH_SCORE_FILE, "w") as file:
        file.write(str(score))


def create_bricks():
    return [pygame.Rect(c * (BRICK_WIDTH + 10) + 35, r * (BRICK_HEIGHT + 10) + 50, BRICK_WIDTH,
BRICK_HEIGHT)
            for c in range(COLS) for r in range(ROWS)]


def display_message(text, subtext=""):
    font, small_font = pygame.font.Font(None, 74), pygame.font.Font(None, 36)
    text_surface = font.render(text, True, WHITE)
    subtext_surface = small_font.render(subtext, True, WHITE)
    text_rect, subtext_rect = text_surface.get_rect(center=(WIDTH // 2, HEIGHT // 2)),
subtext_surface.get_rect(center=(WIDTH // 2, HEIGHT // 2 + 50))
    screen.fill((20, 40, 60))
    screen.blit(text_surface, text_rect)
    if subtext: screen.blit(subtext_surface, subtext_rect)
    pygame.display.flip()
    pygame.time.delay(1500)


def main_game():
    paddle = pygame.Rect(WIDTH // 2 - PADDLE_WIDTH // 2, HEIGHT - 30, PADDLE_WIDTH,
PADDLE_HEIGHT)
    ball = pygame.Rect(WIDTH // 2 - BALL_SIZE // 2, HEIGHT // 2 - BALL_SIZE // 2, BALL_SIZE,
BALL_SIZE)
    ball_speed_x, ball_speed_y = 4 * random.choice([-1, 1]), -4
```

```
bricks, score, high_score = create_bricks(), 0, load_high_score()

clock = pygame.time.Clock()

while True:
    # Event handling
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    keys = pygame.key.get_pressed()
    paddle.x += (keys[pygame.K_RIGHT] - keys[pygame.K_LEFT]) * 6
    paddle.clamp_ip(screen.get_rect())

    ball.x += ball_speed_x
    ball.y += ball_speed_y
    if ball.left <= 0 or ball.right >= WIDTH: ball_speed_x *= -1
    if ball.top <= 0: ball_speed_y *= -1
    if ball.bottom >= HEIGHT:
        display_message("Game Over!", f"Score: {score}")
        return score

    if ball.colliderect(paddle): ball_speed_y *= -1
    for brick in bricks[:]:
        if ball.colliderect(brick):
            bricks.remove(brick)
            ball_speed_y *= -1
            score += 10

    if not bricks:
```

```
            display_message("You Win!", f"Final Score: {score}")
            return score

        screen.fill((20, 40, 60))
        pygame.draw.rect(screen, BLUE, paddle)
        pygame.draw.ellipse(screen, RED, ball)
        for brick in bricks: pygame.draw.rect(screen, BLUE, brick)

        font = pygame.font.Font(None, 36)
        screen.blit(font.render(f"Score: {score}", True, WHITE), (10, 10))
        screen.blit(font.render(f"High Score: {high_score}", True, WHITE), (WIDTH - 200, 10))
        pygame.display.flip()
        clock.tick(60)

display_message("Welcome to Breakout!", "Press any key to start")
pygame.event.clear()
while not any(event.type == pygame.KEYDOWN for event in pygame.event.get()): pass

while True:
    score = main_game()
    high_score = load_high_score()
    if score > high_score:
        save_high_score(score)
        display_message("New High Score!", f"Score: {score}")
    else:
        display_message("Try Again!", f"Score: {score}")
```