

Pong GAME

Aim : To create a project for Pong game

Algorithm :

- **Balloon Movement:** Uses trigonometric functions (sine and cosine) to update balloon positions and simulate zigzag movement.
- **Collision Detection:** Uses axis-aligned bounding box checks to determine if a balloon has been clicked.
- **AI Targeting:** Chooses the balloon that is highest on the screen as the target.
- **Randomization:** Used for the balloon to come randomly.

Program :

```
import pygame
```

```
import time
```

```
import random
```

```
# Initialize Pygame
```

```
pygame.init()
```

```
# Screen dimensions
```

```
WIDTH, HEIGHT = 800, 600
```

```
CELL_SIZE = 20
```

```
# Colors
```

```
WHITE = (255, 255, 255)
```

```
BLACK = (0, 0, 0)
```

```
GREEN = (0, 255, 0)
```

```
RED = (255, 0, 0)
```

```

# Setup the display
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption('Snake Game')

# Font for scoring
font = pygame.font.SysFont(None, 35)

def draw_snake(snake_list):
    for x, y in snake_list:
        pygame.draw.rect(screen, GREEN, [x, y, CELL_SIZE, CELL_SIZE])

def draw_food(food_pos):
    pygame.draw.rect(screen, RED, [food_pos[0], food_pos[1], CELL_SIZE, CELL_SIZE])

def message(msg, color):
    mesg = font.render(msg, True, color)
    screen.blit(mesg, [WIDTH / 6, HEIGHT / 3])

def game_loop():
    game_over = False
    game_close = False

    x1 = WIDTH / 2
    y1 = HEIGHT / 2

    x1_change = 0
    y1_change = 0

    snake_List = []

```

```
Length_of_snake = 1
```

```
foodx = round(random.randrange(0, WIDTH - CELL_SIZE) / CELL_SIZE) * CELL_SIZE
```

```
foody = round(random.randrange(0, HEIGHT - CELL_SIZE) / CELL_SIZE) * CELL_SIZE
```

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

```
snake_speed = 10 # Lower the speed to slow down the snake
```

```
while not game_over:
```

```
    while game_close == True:
```

```
        screen.fill(BLACK)
```

```
        message("You Lost! Press Q-Quit or C-Play Again", RED)
```

```
        pygame.display.update()
```

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

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

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

```
                game_over = True
```

```
                game_close = False
```

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

```
                game_loop()
```

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

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

```
            game_over = True
```

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

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

```
                x1_change = -CELL_SIZE
```

```

        y1_change = 0
    elif event.key == pygame.K_RIGHT:
        x1_change = CELL_SIZE
        y1_change = 0
    elif event.key == pygame.K_UP:
        y1_change = -CELL_SIZE
        x1_change = 0
    elif event.key == pygame.K_DOWN:
        y1_change = CELL_SIZE
        x1_change = 0

if x1 >= WIDTH or x1 < 0 or y1 >= HEIGHT or y1 < 0:
    game_close = True

x1 += x1_change
y1 += y1_change
screen.fill(BLACK)
draw_food([foodx, foody])
snake_Head = []
snake_Head.append(x1)
snake_Head.append(y1)
snake_List.append(snake_Head)
if len(snake_List) > Length_of_snake:
    del snake_List[0]

for x in snake_List[:-1]:
    if x == snake_Head:
        game_close = True

draw_snake(snake_List)

```

```
pygame.display.update()
```

```
if x1 == foodx and y1 == foody:
```

```
    foodx = round(random.randrange(0, WIDTH - CELL_SIZE) / CELL_SIZE) * CELL_SIZE
```

```
    foody = round(random.randrange(0, HEIGHT - CELL_SIZE) / CELL_SIZE) * CELL_SIZE
```

```
    Length_of_snake += 1
```

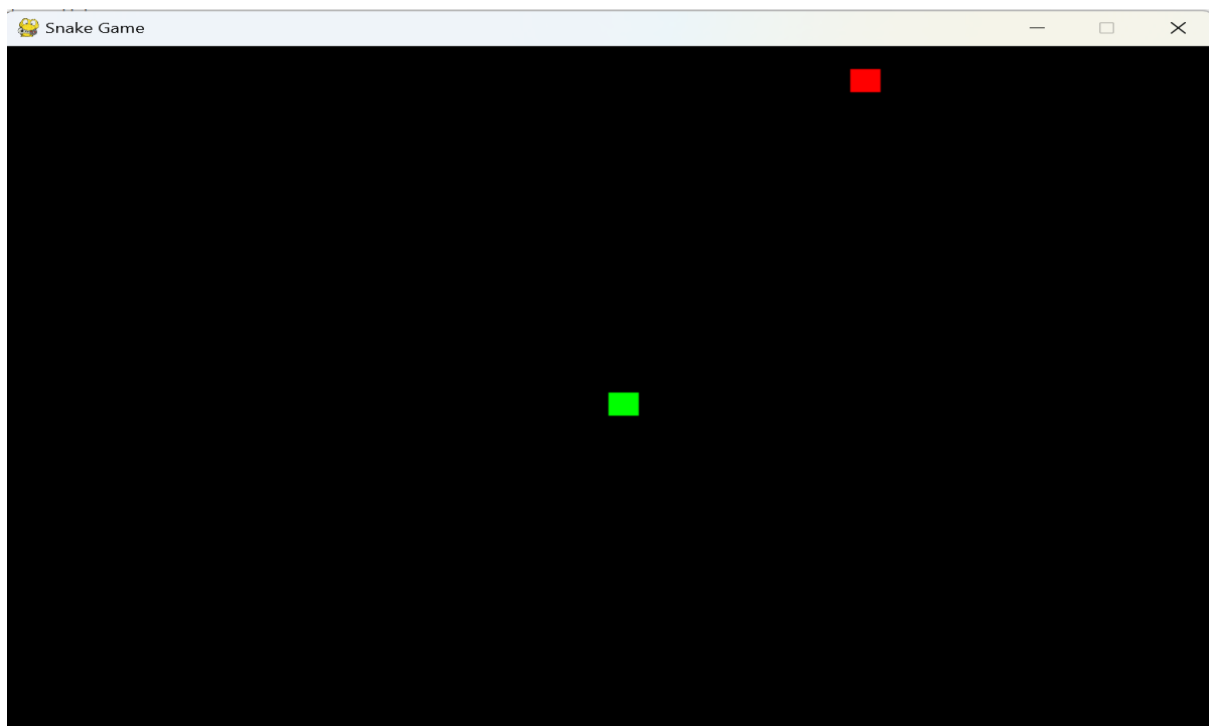
```
clock.tick(snake_speed) # Control the speed by setting FPS
```

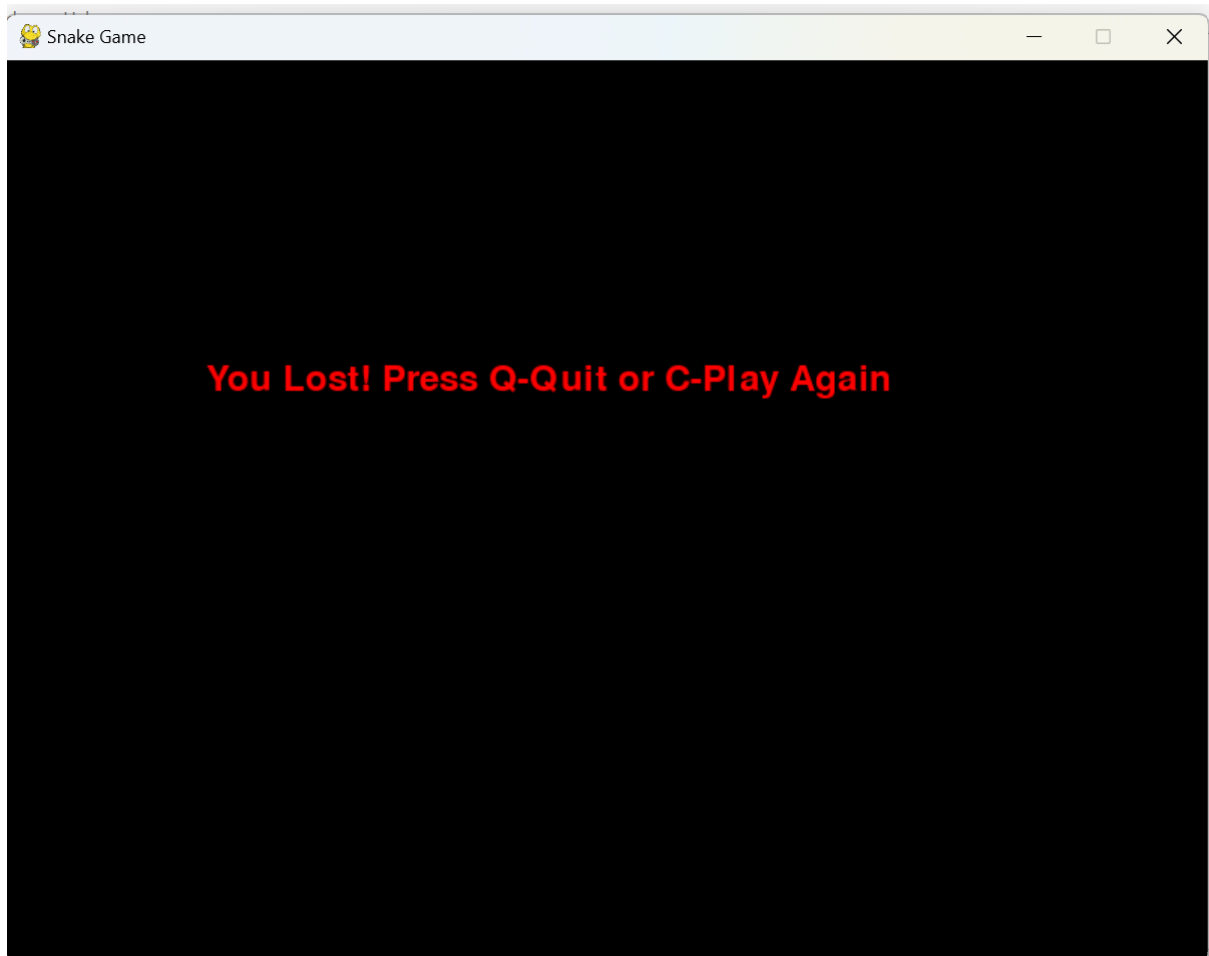
```
pygame.quit()
```

```
quit()
```

```
game_loop()
```

Output :





Conclusion : The project for pong game has been executed successfully.