



আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম
الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong

Department of Computer & Communication Engineering(CCE)

LAB REPORT

Experiment No: 04

Experiment Name: Drawing different shapes like polygons, circles, ellipse

Course Title: Computer Animation and Game Development Sessional

Course Code: CCE-3606

Submitted By

Student Name : Ahsanul Karim Tanim

Student Id : E221013

Semester : 6th

Section : A

Submitted To

Sukanta Paul,

Adjunct Faculty,

IIUC

Experiment Date: / /

Submission Date: / /

Remark



Experiment No: 04

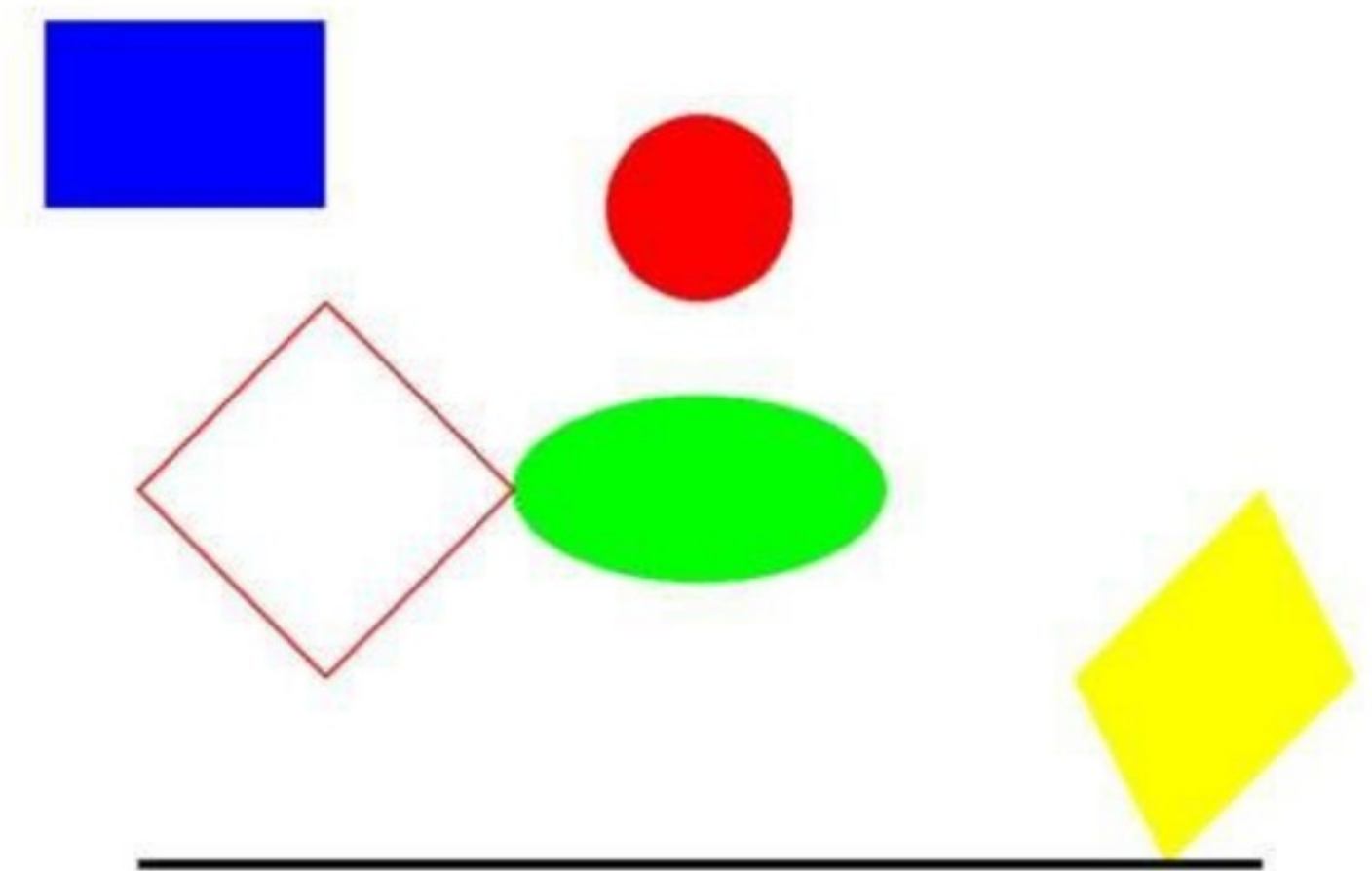
Experiment Name: Drawing different shapes like polygons, circles, ellipse, lines.

Process: This code initializes Pygame and creates an 800x600 pixel window with a white background. Inside the main loop, it listens for events like quitting the program. Various shapes are drawn on the screen using Pygame's drawing functions: a rectangle, a circle, an ellipse, a polygon, and lines. Each shape is specified by its position, size, and color. The shapes are updated on the screen in every frame, and the display refreshes continuously to show the drawn elements. When the user closes the window, the program exits gracefully. This demonstrates how to draw basic 2D shapes using Pygame's functionality.

Code:

```
import pygame
import sys
# Initialize Pygame
pygame.init()
# Set up the display
WIDTH, HEIGHT = 800, 600
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Drawing Shapes in Pygame")
# 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)
# Main loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
    # Fill the background
    screen.fill(WHITE)
    # Draw shapes
    # Draw a rectangle (x, y, width, height)
    pygame.draw.rect(screen, BLUE, (50, 50, 150, 100)) # Draw a circle (center, radius)
    pygame.draw.circle(screen, RED, (400, 150), 50)
    # Draw an ellipse (bounding rectangle)
    pygame.draw.ellipse(screen, GREEN, (300, 250, 200, 100))
    # Draw a polygon (list of points)
    pygame.draw.polygon(screen, YELLOW, [(600, 400), (700, 300), (750, 400), (650, 500)])
    # Draw a line (start point, end point, thickness)
    pygame.draw.line(screen, BLACK, (100, 500), (700, 500), 5)
    # Draw multiple lines to form a shape
    pygame.draw.lines(screen, RED, True, [(100, 300), (200, 400), (300, 300), (200, 200)], 3)
    # Update the display
    pygame.display.flip()
```

Output:



Discussion: This experiment demonstrates the use of Pygame to create and render 2D shapes, including rectangles, circles, ellipses, polygons, and lines. The code highlights key drawing functions, showcasing how shapes are positioned, sized, and colored. It emphasizes the importance of a continuous update loop for rendering graphics and handling user inputs, such as exiting the program. This lab provides a foundational understanding of graphical programming in Pygame, offering a starting point for developing more advanced visual applications like games or simulations. The program also emphasizes the importance of a continuous loop for updating the screen and handling user inputs, such as exiting the application.