আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম
الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong
Department of Computer & Communication Engineering(CCE)

LAB REPORT

Experiment No: 05
Experiment Name: Animation using Python

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 Facuty,
IIUC

Experiment Date: / /

Submission Date: / /

Remark

Experiment No: 05
Experiment Name: Animation using Python

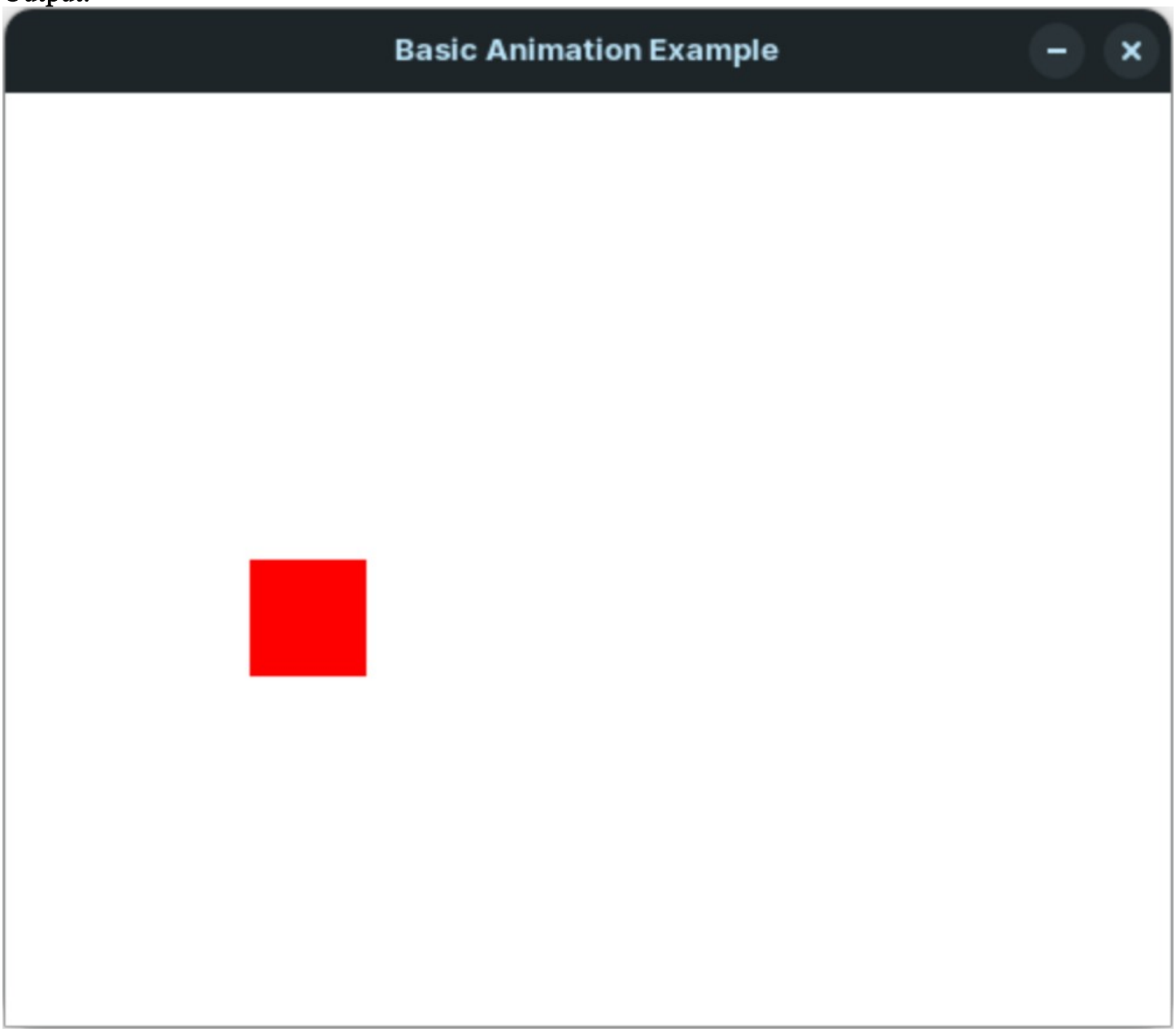
Objective: The objective of this experiment is to demonstrate how to create basic animations using Python and Pygame. By implementing a simple moving object on the screen, the experiment aims to introduce concepts like object movement, updating the screen, and using time intervals to control the animation sequence. This helps in understanding the foundational principles of game development and interactive graphical applications.

Code:

```
import pygame
import sys
# Initialize Pygame
pygame.init()
# Set up the display
screen = pygame.display.set_mode((500, 400))
pygame.display.set_caption("Basic Animation Example")
# Define colors
WHITE = (255, 255, 255)
RED = (255, 0, 0)
# Set up initial position and speed
x_pos = 0
y_pos = 200
speed = 5
# Main loop
running = True
while running:
for event in pygame.event.get():
if event.type == pygame.QUIT:
running = False
# Fill the screen with the background color
screen.fill(WHITE)
# Draw a red rectangle to represent the moving object
pygame.draw.rect(screen, RED, (x_pos, y_pos, 50, 50))
# Update the position for animation (moving to the right)
x_pos += speed
# Reset position to create a looping effect
if x_pos > 500:
x_pos = -50
# Update the display
pygame.display.flip()
# Set the frame rate to control animation speed
```

```
pygame.time.Clock().tick(60)
# Quit Pygame
pygame.quit()
```

Output:



Discussion: This experiment demonstrates basic animation using Pygame, where a red rectangle moves across the screen. The position of the rectangle is updated in each frame, and once it moves off the screen, it resets, creating a loop. The pygame.time.Clock().tick(60) function limits the frame rate, ensuring smooth animation. This simple example helps understand the principles of object movement and frame updates in animation. A limitation is the lack of variety in movement; more advanced techniques such as sprite-based animation or more complex motion could be added for a richer experience.