IIUC

আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম

الجامعة الإسلامية العالمية شيتاغونغ

International Islamic University Chittagong

Department of Computer & Communication Engineering(CCE)

LAB REPORT

Experiment No: 02
Experiment Name: A complete pygame program

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

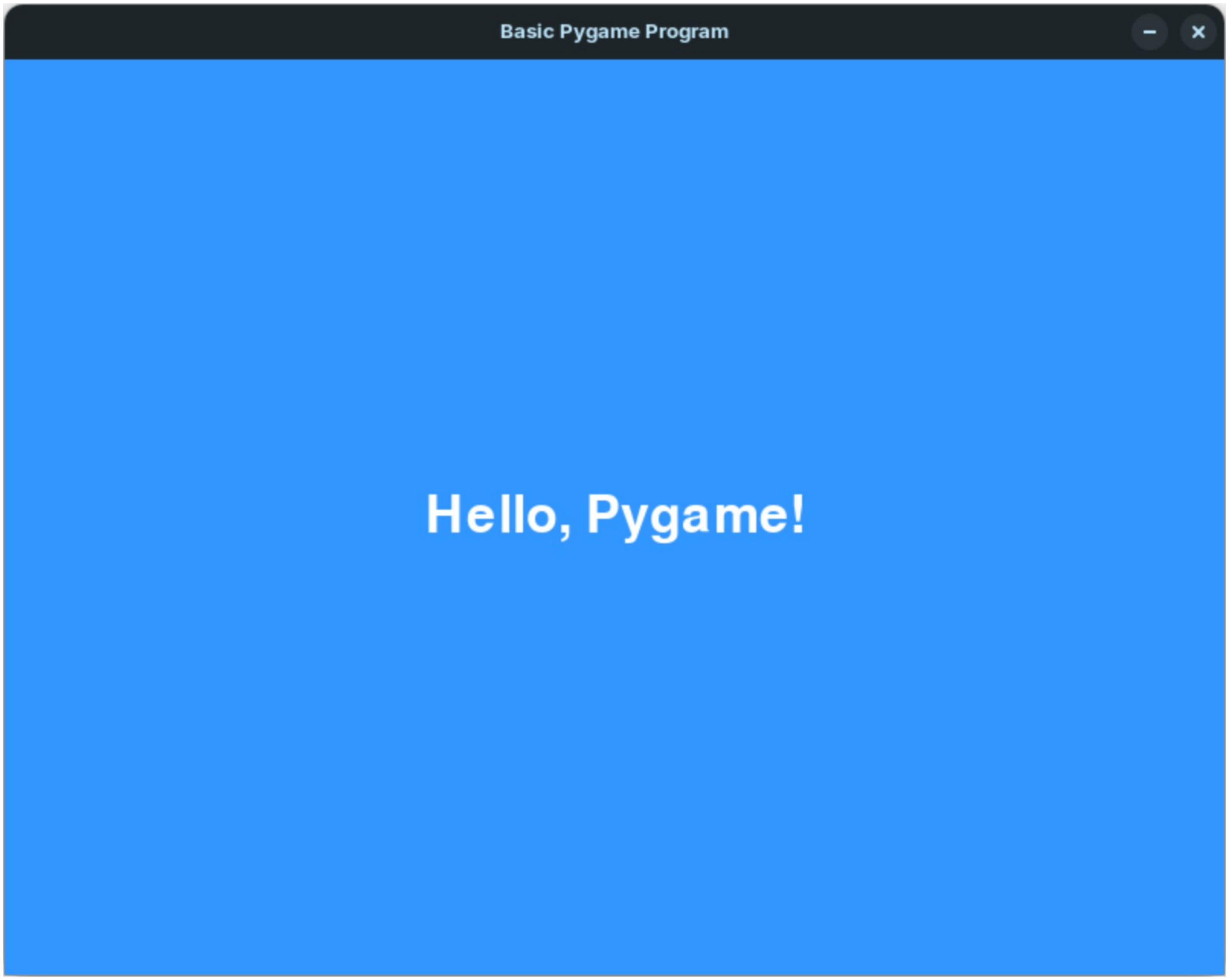
Experiment No: 02
Experiment Name: A complete pygame program

Objective: The objective of this lab is to develop a basic Pygame program that demonstrates the initialization of the Pygame library, creation of a graphical display, rendering text on the screen, and handling basic events such as exiting the program. This experiment aims to provide a foundational understanding of Pygame's functionality and structure for creating interactive applications.

Code:

```
import pygame
import sys
# Initialize Pygame
pygame.init()
# Set up the display
screen = pygame.display.set_mode((800, 600))
pygame.display.set_caption("Basic Pygame Program")
# Create a font and render the text
font = pygame.font.Font(None, 50)
text = font.render("Hello, Pygame!", True, (255, 255, 255))
text_rect = text.get_rect(center=(400, 300))
# Main loop
while True:
for event in pygame.event.get():
if event.type == pygame.QUIT:
pygame.quit()
sys.exit()
# Fill the screen with a background color
screen.fill((50, 150, 255))
# Draw the text on the screen
screen.blit(text, text_rect)
# Update the display
pygame.display.flip()
```

Output:



Hello, Pygame!

Discussion: This lab demonstrates the basic structure of a Pygame program, including initializing the library, creating a display window, rendering text, and handling events like quitting the program. The use of get_rect() for centering the text and fill() for setting a background color highlights key features of Pygame. The program provides a foundational understanding of graphical updates through a game loop. However, it lacks interactivity beyond closing the window. Future improvements could include dynamic elements like user input handling or animations to make the program more engaging and interactive while maintaining its simplicity for beginner-level understanding.