Training Project Report

Submitted in partial fulfillment of the degree of

B-tech in Electronics & Communication Engineering

By

Ishanjit Nandi
Pritam Paul
Rishikesh prasad
Aparupa Mukherjee

F-year student of

SILIGURI INSTITUTE OF TECHNOLOGY

THIS IS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

AFFILIATED TO

Maulana Abul Kalam Azad University of Technology

Under the supervision of :- Mr. Ripam Kundu

Sikharthy Infotech Pvt. Ltd.

PROJECT ON :- GAme developed By Python

By

Ishanjit Nandi
Pritam Paul
Rishikesh prasad
Aparupa Mukherjee

UNDER THE GUIDANCE OF

Mr. Ripam Kundu

Project Guide

Sikharthy Infotech Pvt. Ltd.

THIS IS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

B.Tech

IN

Electrical Engineering

SILIGURI INSTITUTE OF TECHNOLOGY

Department of Electronics & Communication Engineering

I hereby forward the documentation prepared under my supervision by **Ripam Kundu Sir** entitled **Siliguri Institute Of Technology** to be accepted as fulfillment of the requirement for the Degree of Bachelor of Technology in Electrical Engineering, **Siliguri Institute Of Technology** affiliated to **Maulana Abul Kalam Azad University of Technology** (**MAKAUT**).

Mr.Ripam Kundu
(Software Developer)
Project Guide
Sikharthy Infotech Pvt. Ltd.

HOD

Department Of Electronics &
communication Engineering, SIT

HOD
(DESH Building)
Sikharthy Infotech Pvt. Ltd.

TPO
Siliguri Institute of Technology

Certificate of Approval

The foregoing project is hereby approved as a creditable study for the B.Tech in Electronics & Communication Engineering presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorsed or approved any statement made, opinion expressed or conclusion therein but approve this project only for the purpose for which it is submitted.

Final Examination for
Evaluation of the Project

Signatures of Examiners

ACKNOWLEDGEMENT

It is a great pleasure for me to acknowledge the assistance and participation of a large number of individuals in this attempt. Our project report has been structured under the valued suggestion, support, and guidance of **Mr. Ripam Kundu**. Under his guidance, we have accomplished the challenging task in a very short time.

Finally, we express our sincere thankfulness to our family members for inspiring me all throughout and always encouraging us.

Group Mamber Signature

_____

_____

_____

_____

INTRODUCTION

We will use Python and its different libraries to developed game candy crush

WHAT LIBRARIES WE USED

Importing Libraries

The analysis will be done using the following libraries :

Pygame : Pygame is a Python module that is used for game development and multimedia applications. It provides a set of Python functions and classes that enable the creation of games and other multimedia applications, such as interactive art installations, simulations, and educational software.

Pygame is built on top of the SDL library, which is a cross-platform library for creating multimedia applications. Pygame provides an easy-to-use API for handling graphics, sound, user input, and game logic. Some of the key features provided by Pygame include:

- **Sprite and sprite group management**
- **Collision detection**
- **Sound and music playback**
- **Input handling (keyboard, mouse, joystick)**
- **Drawing and rendering graphics (shapes, images, text)**

Random : The random module is a built-in module in Python that provides a suite of functions for generating pseudo-random numbers. The import random statement at the beginning of a Python script or interactive session loads the random module into memory, making the functions it provides available for use within that script or session.

Some of the common functions provided by the random module include:

- random(): **generates a random float between 0 and 1**
- randint(a, b): **generates a random integer between** a **and** b **(inclusive)**
- choice(seq): **returns a random element from the given sequence** seq
- shuffle(seq): **shuffles the elements in a sequence** seq **in place**
- sample(seq, k): **returns a new list containing** k **unique elements randomly chosen from the sequence** seq

From pygame.locals import : **The line from pygame.locals import * is a common way to import a set of constants and enums from the pygame.locals module into your Python code.**

The pygame.locals module contains a large number of constant values that are used by Pygame to represent things like keyboard keys, mouse buttons, event types, and video modes. By importing these constants using the * syntax, you can use them in your code without having to prefix them with the pygame.locals module name.

For example, if you were to include the line from pygame.locals import * in your code, you could use the K_ESCAPE constant to represent the escape key on the keyboard like this:

```
import pygame
from pygame.locals import *

pygame.init()
screen = pygame.display.set_mode((640, 480))

running = True
while running:
    for event in pygame.event.get():
        if event.type == KEYDOWN:
            if event.key == K_ESCAPE:
                running = False

    # Draw game graphics here
    pygame.display.flip()

pygame.quit()
```

CREATE A CLASS

We have created a class name "Candy" within the class we made def draw function for draw image on the screen .Next we made def snap function and initialised snap function for row wise and column wise.

```
class Candy:
    def __init__(self, row_num, col_num):···
    # draw the image on the screen
    def draw(self):···
    # snap the candy to its position on the board
    def snap(self):···
    def snap_row(self):···
    def snap_col(self):
        self.rect.left = self.col_num * candy_width
```

ADD SWAP FUNCTION

Outside "Class Candy" we create another function of swap in this function for update the candies after swap on the board.

```
def swap(candy1, candy2):
    temp_row = candy1.row_num
    temp_col = candy1.col_num
    candy1.row_num = candy2.row_num
    candy1.col_num = candy2.col_num
    candy2.row_num = temp_row
    candy2.col_num = temp_col
    # update the candies on the board list
    board[candy1.row_num][candy1.col_num] = candy1
    board[candy2.row_num][candy2.col_num] = candy2
    # snap them into their board positions
    candy1.snap()
    candy2.snap()
```

MATCH THE CANDIES COLOR

We have create a find_matches function to check candies the above color,below color,left color,right color.

```
# find neighboring candies that match the candy's color
def find_matches(candy, matches):
    # add the candy to the set
    matches.add(candy)
    # check the candy above if it's the same color
    if candy.row_num > 0:···
    # check the candy below if it's the same color
    if candy.row_num < height / candy_height - 1:···
    # check the candy to the left if it's the same color
    if candy.col_num > 0:···
    # check the candy to the right if it's the same color
    if candy.col_num < width / candy_width - 1:···
    return matches
```

GAME PROCESSING

we made a while loop for matching candies we need to assign mouse and keyboard command.We use conditional statement to assign mouse and keyboard.Then we called draw function and next we updated screen.We check if there's at least three candies match

we add score and animate the candies sharking.

```
while running:
    # set of matching candies
    matches = set()
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False
        # detect mouse click
        if clicked_candy is None and event.type == MOUSEBUTTONDOWN: ···
        # detect mouse motion
        if clicked_candy is not None and event.type == MOUSEMOTION: ···
        # detect mouse release
        if clicked_candy is not None and event.type == MOUSEBUTTONUP: ···
    draw()
    pygame.display.update()
    # check if there's at least 3 matching candies
    if len(matches) >= 3:
        # add to score
        score += len(matches)
        # animate the matching candies shrinking
        while len(matches) > 0: ···
```

FUNCTIONAL REQUIREMENTS OF THE SYSTEM

SOFTWARE:

Operating System

Windows OS 11

WEB BROWSER:

Internet Explorer 7

Google Chrome

CODING LANGUAGE :

Python

CONCLUSION

Developing a Candy Crush program using Python and a game development library like Pygame can be a great way to learn and practice various programming concepts, such as object-oriented programming, algorithms and data structures, user interface design, and game development best practices.

Throughout the development process, one can learn how to apply game design principles to create a fun and engaging game experience, and how to use programming tools and techniques to implement the game mechanics and features.

Moreover, presenting the developed Candy Crush program can be a great opportunity to showcase one's programming skills and creativity, and to receive feedback from peers and experts in the field. It can also inspire others to learn and explore the world of game development and programming.

Overall, developing and presenting a Candy Crush program can be a challenging and rewarding experience, and can help one build valuable skills and knowledge that can be applied to many other programming projects in the future.

REFERENCE

candy crush game