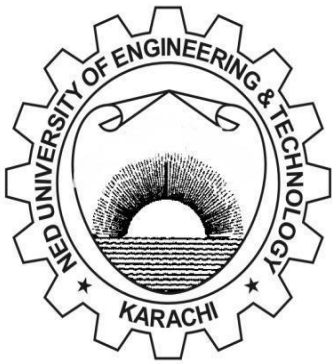
Department of Software Engineering

NED University of Engineering & Technology, Karachi – 75270, Pakistan



PROGRAMMING FUNDAMENTAL PROJECT.

TITTLE: **'CANDY CLASH'**

Name : INSHARA IQBAL

Year : 2021-22

Batch : 2021

Roll No : SE-21018

Department: SOFTWARE

SUBMITTED TO:
MAM ASMA KHAN

TITTLE:

CANDY CLASH ON PYGAME.

OBJECTIVE:

Candy clash is a simple game formed by using python programming language.
The purpose of the game is to understand the logics of the famous py - game library and to enhance my programming skills. I am presenting this in my final project of programming fundamental.
Candy clash can also be used as an entertaining purpose.

DESCRIPTION:

The performance of the game is that it will generate multiple candies at random position using random module. And the character present at the bottom will shoot it with the ball. The candies will come closer to bottom as the time passes. Hit the candies with the bottom will be GAME OVER.

FUNCTIONS AND MODULES:

1. **PYGAME**
2. **RANDOM MODULE**
3. **GLOBAL**
4. **FROM PYGAME IMPORT MIXER**
5. **MATHS FUNCTION**

And other small functions used in the game.

SYSTEM REQUIRMENTS:

Python latest version (3.10)
Pycharm community.
Processor: intel core i5
Windows 10

RESOURCES AND GUIDANCE:

I Learn the functions of pygame like movement mechanics, windows creations, import audios, import images from the source of internet, And then I work alone for the project with my simple and creative idea.

I also face so many errors while code this project although I tried my best for make this project well and good.

```python
import pygame
import random

import math
from pygame import mixer

pygame.init()

# for screen
screen = pygame.display.set_mode((800, 600))
# background
background = pygame.image.load('NEW WALLPAPER.png')
# background sound

mixer.music.load('music')
mixer.music.play(-1)

# tittle and icon
pygame.display.set_caption("candy clash")
icon = pygame.image.load('mushrooms.png')
pygame.display.set_icon(icon)

#######player
playerImg = pygame.image.load('mushrooms.png')
playerX = 350
playerY = 480
playerX_change = 0
playerY_change = 0

# candy
candyImg = []
candyX = []
candyY = []
candyX_change = []
candyY_change = []
num_of_candies = 6
for i in range(num_of_candies):
    candyImg.append(pygame.image.load('NEW CANDY.png'))
    candyX.append(random.randint(0, 736))
    candyY.append(random.randint(50, 150))
    candyX_change.append(4)
    candyY_change.append(4)
# BALL
ballImg = pygame.image.load('beach-ball.png')
ballX = 0
ballY = 480
ballX_change = 0
ballY_change = 0.9
ball_state = 'ready'
# SCORE
score_value = 0
font = pygame.font.Font('freesansbold.ttf', 32)
textX = 10
testY = 10
# Game over text
over_font = pygame.font.Font('freesansbold.ttf', 64)
```

```
def show_score(x, y):
    score = font.render("SCORE : " + str(score_value), True, (199, 9, 8))
    screen.blit(score, (x, y))


def game_over_text():
    over_text = over_font.render("GAME OVER!!!!!", True, (0, 128, 128))
    screen.blit(over_text, (200, 250))


def player(x, y):
    screen.blit(playerImg, (x, y))


def candy(x, y, i):
    screen.blit(candyImg[i], (x, y))


def fire_ball(x, y):
    global ball_state
    ball_state = 'fire'
    screen.blit(ballImg, (x + 16, y + 10))


def isCollision(candyX, candyY, ballX, ballY):
    distance = math.sqrt((math.pow(candyX - ballX, 2)) + (math.pow(candyY - ballY, 2)))
    if distance < 27:
        return True
    else:
        return False
```

```
# Game loop
running = True
while running:
    # RGB

    screen.fill((0, 0, 0))
    # background:
    screen.blit(background, (0, 0))
    # Event handling
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        # check key press*********************************
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                playerX_change = -4
            if event.key == pygame.K_RIGHT:
                playerX_change = 4
            if event.key == pygame.K_SPACE:
                if ball_state == 'ready':
                    ballX= playerX
                    fire_ball(ballX, ballY)

        if event.type == pygame.KEYUP:
            if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
                playerX_change = 0
```

```
    # Boundaries of cartoon
    playerX += playerX_change
    if playerX <= 0:
        playerX = 0
    elif playerX >= 736:
        playerX = 736
    # Candy movement
    for i in range(num_of_candies):
        # Game over
        if candyY[i] > 440:
            for j in range(num_of_candies):
                candyY[j] = 2000

            game_over_text()
            break

        candyX[i] += candyX_change[i]
        if candyX[i] <= 0:
            candyX_change[i] = 2
            candyY[i] += candyY_change[i]
        elif candyX[i] >= 736:
            candyX_change[i] = -2
            candyY[i] += candyY_change[i]
            # Collision
        collision = isCollision(candyX[i], candyY[i], ballX, ballY)
        if collision:
            ballY=480
            ball_state='ready'
            score_value +=1
            candyX[i] = random.randint(0,736)
            candyY[i]= random.randint(50,150)

        candy(candyX[i], candyY[i], i)



        candy(candyX[i], candyY[i], i)
    # Bullet movement
    if ballY <= 0:
        ballY = 480
        ball_state = 'ready'

    if ball_state == 'fire':
        fire_ball(playerX, ballY)
        ballY -= ballY_change

    player(playerX, playerY)
    show_score(textX, testY)
    pygame.display.update()
```