Date: 21.3.2022

Catch the Apple

CG PROJECT

[A game made with pygame]

NAME: JENIS WINSTA

ROLL NO.: 21

UID: 215038

Contents

1.Introduction …….03

2.Working of the game……..05

3.Code snippet……...07

4.Screenshots of the game……....16

5.Feedbacks ……..20

6.Modification after feedback…….....20

7.References……..21

INTRODUCTION

Catch the Apple is a simple game made with pygame. It is a one player game where the player is expected to catch the apples falling one by one with the basket provided. The basket can move with the help of arrow keys. The basket has only right and left movements.

We have 3 elements in the game window:

1. **Score:** The score is initially 0 then incremented by 1 if the basket and apple touch each other.

2. **Loss:** The loss is initially 0 then incremented by 1 if the apple reaches the end of the screen (If we don't catch the apple)

3. **Lives:** The lives get decremented if the loss becomes 5 i.e., if we lose 5 apple, we lose 1 life. In total we have 3 lives.

Lifeline:

We can use 2 lifelines. For lifeline we will get a general Knowledge question in the console which we need to answer. If we get the answer correct then our lives get incremented by 1.

Record :

At the end of the lives, the score is evaluated. If the score is greater than 50, "YOU HAVE CREATED A RECORD" is shown in

the screen. Then you had to go to the console and write you name which is written in the record.txt file along with the score.

Lost :

At the end of the lives, the score is evaluated. If the score is lesser than 50,”YOU HAVE LOST” is shown in the screen and the game is exited automatically.

WORKING OF THE GAME

The game 2 actual game loop. Each loop is defined inside a function. The actual game is present in the **main()** function. And the menu is present in the **main_menu()** function. Whenever we run the program the main_menu() is executed and when we press the mouse button the main() function is called and we enter into the game loop. And if we lose 5 apples we enter back to the main_menu() with lives decremented by 1.

In the main_menu(), we can also enter into **lifeline_fun()** by pressing the spacebar. After pressing the spacebar, we have to minimize the game window and go to the console. We will have a question selected randomly by random.randiant() function from range 1-10. If we answer the question correctly, we will get an increment in our life. After answering the quest3ion, we have to go back to the gaming window and continue our game. The maximum no. of lives in 3, so if you use your lifeline even if you have 3 lives then the lifeline will be wasted. So, concentration on no. of lives is also important for proper usage of lifeline.

At the end of each life, the score and the loss are printed in the console. Since score is a global variable, we

have our score working continuous(i.e., if our first life ends with a score:10, then our second life began with score:10 and continue with the game). But loss is initialized each time it enters the game loop. And at the end the total score gets printed.

Since the main() function is called inside the main_menu() function, whenever we close the main() [gaming loop] the lives get decremented and goes to main_menu() screen. And if we close the main_menu() [instruction screen] the game is quitted and prints the total score in the console. Here even if you score more than 50 your name is not recorded because you quitted the game.

As mentioned earlier, at the end when the score is more than 50, **record()** function is executed. The game screen shows the message “You have created a record”. After that you have to minimize the gaming screen and go to the console and enter your name.

IMPORTANT: At this time, if you close the game loop the game gets terminated and you won’t be able to enter your name.(i.e. even scoring more than 50 you won’t have a record of your score).

CODE SNIPPET:

```
import pygame as py
import random
py.init()

screen_width = 1540
screen_height = 800
screen = py.display.set_mode((screen_width, screen_height))
py.display.set_caption("cg project")
clock = py.time.Clock()

apple = py.image.load('apple.png')
apple = py.transform.scale(apple, (80, 80))
apple_rect = apple.get_rect()
apple_mask = py.mask.from_surface(apple)

sprite = py.image.load('basket.png')
sprite = py.transform.scale(sprite, (160, 160))
basket_mask = py.mask.from_surface(sprite)
basket_rect = sprite.get_rect()

bg = py.image.load('BG.jpg')
bg = py.transform.scale(bg, (screen_width, screen_height))

ins_bg=py.image.load('ins.jpg')
ins_bg= py.transform.scale(ins_bg, (screen_width, screen_height))

score = 0
bom = 0
lives = 3
n = 0
def new_app():
    x = random.randint(80, screen_width - 80)
    y = random.randint(-1500, 180 )
```

```
num_app = 1
app = [ ]
for _ in range(num_app ):
    app.append(new_app() )


def new_frame():
    global app
    next_app = [ ]  # list that will contain the next state
    for x, y in app:
        y += 5
   # if at least part of the apple is in the window, we'll keep it
        if y + 80 > 0 and y - 80 < screen_height:
            next_app.append((x, y) )
    for x, y in app:
        if y + 80 > screen_height:
            pass

    while len(next_app ) < num_app:
        next_app.append(new_app() )
    app = next_app
    for x, y in app:
        ix, iy = int(x ), int(y )
        apple_rect.x = ix
        apple_rect.y = iy
        screen.blit(apple, (ix, iy) )


def main():
    run = True
    fps = 100
    clock = py.time.Clock()
    global score
    global bom
```

```
global lives
bom = 0
global app
win = 50
lost_font = py.font.SysFont("comicsans", 50 )

def in_screen(x):
    if x <= 0:
        x: int = 0
    if x > 1540 - 160:
       x = 1540 - 160
   return x


while run:
   screen.blit(bg, (0, 0))

   new_frame()  # apples start falling
   basket_rect.x = in_screen(basket_rect.x )
   basket_rect.y = 600
   if bom >= 5:
      if score >= win:
         win_lable = lost_font.render("You created a record!!", True, (0,0,0) )
         screen.blit(win_lable, (screen_width / 2 - win_lable.get_width() / 2, 350) )
      else:
         lost_lable = lost_font.render("You Lost!!", True, (0,0,0) )
         screen.blit(lost_lable, (screen_width / 2 - lost_lable.get_width() / 2, 350) )
      print("score:", score )
      print("loss:", bom )  #
      run = False

   offset = (int(apple_rect.x - basket_rect.x ), int(apple_rect.y - basket_rect.y ))
   collision = basket_mask.overlap(apple_mask, offset )
```

```
if collision:
    score += 1
    app.remove((apple_rect.x, apple_rect.y) )

j = py.draw.line(screen, (255, 255, 255), (0, 799), (1540, 799), 1 )
if j.colliderect(apple_rect):
    bom += 1
    app.remove((apple_rect.x, apple_rect.y) )

 for event in py.event.get():
    if event.type == py.QUIT:
        print("score:", score )
        print("loss:", bom )
        run = False

keys = py.key.get_pressed()
if keys[py.K_LEFT]:
    basket_rect.x -= 8
if keys[py.K_RIGHT]:
    basket_rect.x += 8

score_label = lost_font.render(f"score:{score}", True, (0,0,0) )
loss_label = lost_font.render(f"loss:{bom}", True, (0,0,0) )
lives_lable = lost_font.render(f"lives:{lives}", True, (0,0,0) )
screen.blit(score_label, (10, 10) )
screen.blit(loss_label, (screen_width - loss_label.get_width() - 10, 10) )
screen.blit(lives_lable, (screen_width - lives_lable.get_width() - 750, 10) )
screen.blit(sprite, basket_rect )
clock.tick(fps )
py.display.flip()
```

```
def main_menu():
    run = True
    global score
    global lives
    count = 0
    win = 50
    main_font = py.font.SysFont( "comicsans", 80, bold=True, italic=True)
    title_font = py.font.SysFont("comicsans", 50 )
    subtitle_font = py.font.SysFont("comicsans", 30)
    while run:
        screen.blit(ins_bg,(0,0))
        main_text="CATCH THE APPLE"
        game = "Press the mouse to enter the game..."
        game1="In total 3 LIVES"
        game2="You lose 1 life if you LOSE 5 APPLES"
        game3="The game starts immediately when mouse button is pressed"
        game4 = "When record is created minimize the game screen and use the terminal"
        lifeline = "Press the spacebar for LIFELINE"
        lifeline1 = "NOTE:"
        lifeline2="In lifeline, answe the question given and get an extra life "
        lifeline3 = "In total two lifelines"
        lifeline4 = "To access lifeline go to the command line(minimize the game window)"

        main_lable = main_font.render(main_text, True,(255,250,250))
        game_lable = title_font.render(game, True, (0, 0, 25) )
        game1_lable = subtitle_font.render( game1, True, (0, 0, 25) )
        game2_lable = subtitle_font.render( game2, True, (0, 0, 25) )
        game3_lable = subtitle_font.render( game3, True, (0, 0, 25) )
        game4_lable = subtitle_font.render( game4, True, (0, 0, 25) )


        title_lable = title_font.render(lifeline, True, (0, 0, 25) )
        title_lable1 = subtitle_font.render(lifeline1, True, (0, 0, 25) )
        title_lable2 = subtitle_font.render(lifeline2, True, (0, 0, 25) )
        title_lable3 = subtitle_font.render(lifeline3, True, (0, 0, 25) )
        title_lable4 = subtitle_font.render( lifeline4, True, (0, 0, 25) )
```

```
    screen.blit( apple, (screen_width / 2 - main_lable.get_width() / 2-100, 60) )
    screen.blit( apple, (1180, 60) )
    screen.blit(main_lable, (screen_width / 2 - main_lable.get_width()/2, 50))
    screen.blit(game_lable, (screen_width / 2 - game_lable.get_width() / 2-50, 200) )
    screen.blit( game1_lable, (screen_width / 2 - game_lable.get_width() / 2 , 270) )
    screen.blit( game2_lable, (screen_width / 2 - game_lable.get_width() / 2 , 310) )
    screen.blit( game3_lable, (screen_width / 2 - game_lable.get_width() / 2, 350) )
    screen.blit( game4_lable, (screen_width / 2 - game_lable.get_width() / 2 , 390) )
    screen.blit(title_lable, (screen_width / 2 - game_lable.get_width() / 2-50, 450) )
    screen.blit(title_lable1, (screen_width / 2 - game_lable.get_width() / 2-50, 530) )
    screen.blit(title_lable2, (screen_width / 2 - game_lable.get_width()/2 , 570) )
    screen.blit(title_lable3, (screen_width / 2 - game_lable.get_width() / 2 , 610) )
    screen.blit( title_lable4, (screen_width / 2 - game_lable.get_width() / 2 , 650) )

    for event in py.event.get():
        if event.type == py.QUIT:
            print("TOTAL:", score )
            run = False
        if event.type == py.KEYDOWN and event.key == py.K_SPACE:
            lifeline_fun()
        if event.type == py.MOUSEBUTTONDOWN:
            count += 1
            if count == 1:
                print("First round score:" )
            elif count == 2:
                print("Second round score:" )
            else:
                print("Third round score:" )
            main()
            lives -= 1
            if lives <= 0:
                print("TOTAL:", score )
                run = False
                if score >= win:
                    record()
    py.display.update()
py.quit()
```

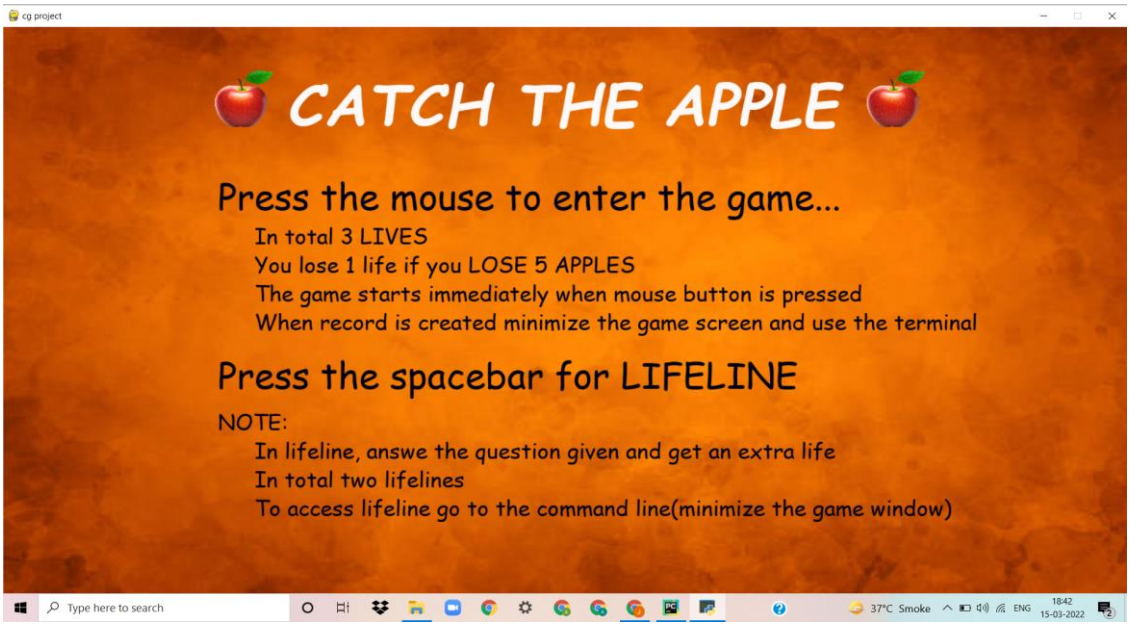
```
def lifeline_fun():
    global lives
    global n
    n += 1
    if lives < 3 and n <= 2:
        print("QUESTIONS:(write in lower case)" )
        i = random.randint(1, 10)
        print(i )
        if i == 1:
            answer = "28"  #
            ans = input("Number of states in INDIA: " )
            if ans == answer:
                print("Correct" )
                lives += 1
            else:
                print("The correct answer is:", answer )
        if i == 2:
            answer = "8"
            ans = input("Number of union territories in INDIA:  " )
            if ans == answer:
                print("Correct" )
                lives += 1
            else:
                print("The correct answer is:", answer )

        if i == 3:
            answer = "10"
            ans = input("5+15/5-2*0+2=" )
            if ans == answer:
                print("Correct" )
                lives += 1
            else:
                print("The correct answer is:", answer )
```

```
    if i == 4:
        answer = "ram nath kovind"
        ans = input("who is the president of india" )
        if ans == answer:
            print("Correct" )
            lives += 1
        else:
            print("The correct answer is:", answer )

    if i == 5:
        answer = "ganga"
        answer2 = "ganges"
        ans = input("Holy river of india is _____ river. " )
        if ans == answer:
            print("Correct" )
            lives += 1
        elif ans == answer2:
            print("Correct" )
            lives += 1
        else:
            print("The correct answer is:", answer )


    if i == 6:
        answer = "jupiter"
        ans = input("Largest planet in the solar system: " )
        if ans == answer:
            print("Correct" )
            lives += 1
        else:
            print("The correct answer is:", answer )
    if i == 7:
        answer = "no"
        ans = input("Is India one among the 5 countries having permanent veto power[yes/no]: " )
        if ans == answer:
            print("Correct" )
            lives += 1
        else:
            print("The correct answer is:", answer )
```

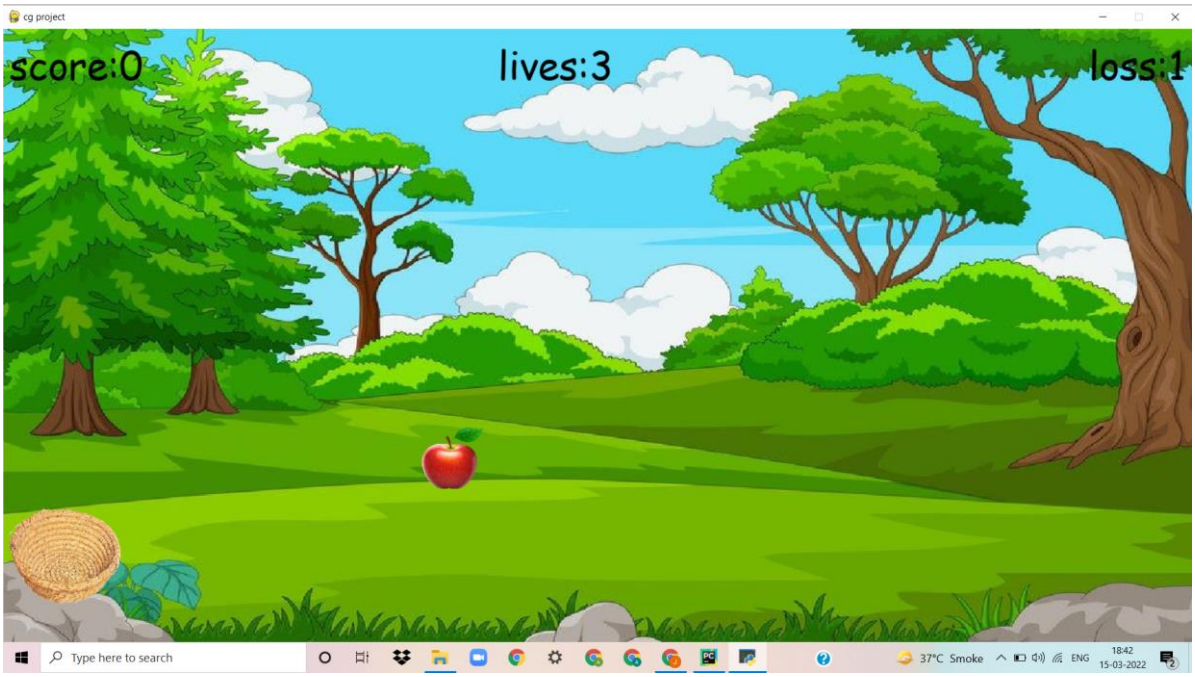
```
        if i == 8:
            answer = "yes"
            ans = input("Charles Babbage invented computer[yes/no]:  " )
            if ans == answer:
                print("Correct" )
                lives += 1
            else:
                print("The correct answer is:", answer )
        if i == 9:
            answer = "no"
            ans = input("Is HINDI national language of INDIA[yes/no]: " )
            if ans == answer:
                print("Correct" )
                lives += 1
            else:
                print("The correct answer is:", answer )
        if i == 10:
            answer = "543"
            ans = input("No. of Lok Sabha constituencies in India: " )
            if ans == answer:
                print("Correct" )
                lives += 1
            else:
                print("The correct answer is:", answer )

    elif n > 2:
        print("lifeline already used" )
    else:
        print("Can't increase lives any more" )
        print("LIFELINE WASTED" )


def record():
    name = input("Enter your name:")
    print(name)
    file = open('record.txt', 'a')
    file.write(f"\n{name}          {score}")
    print(name, "your score is recorded")


main_menu()
```

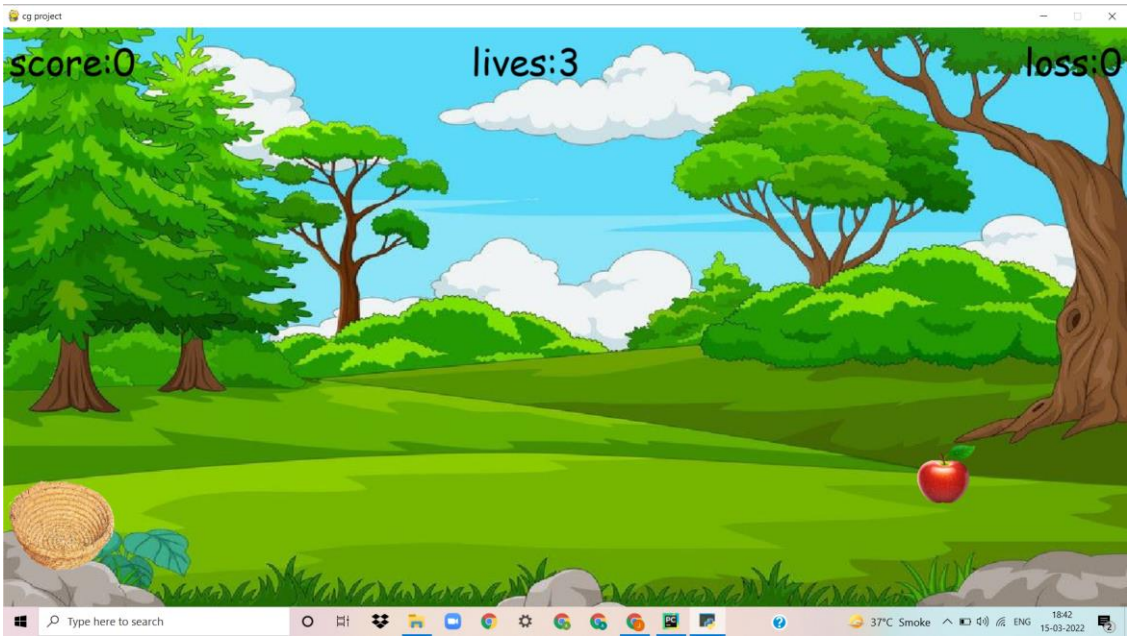
SCREENSHOTS:

Main_menu(): screen

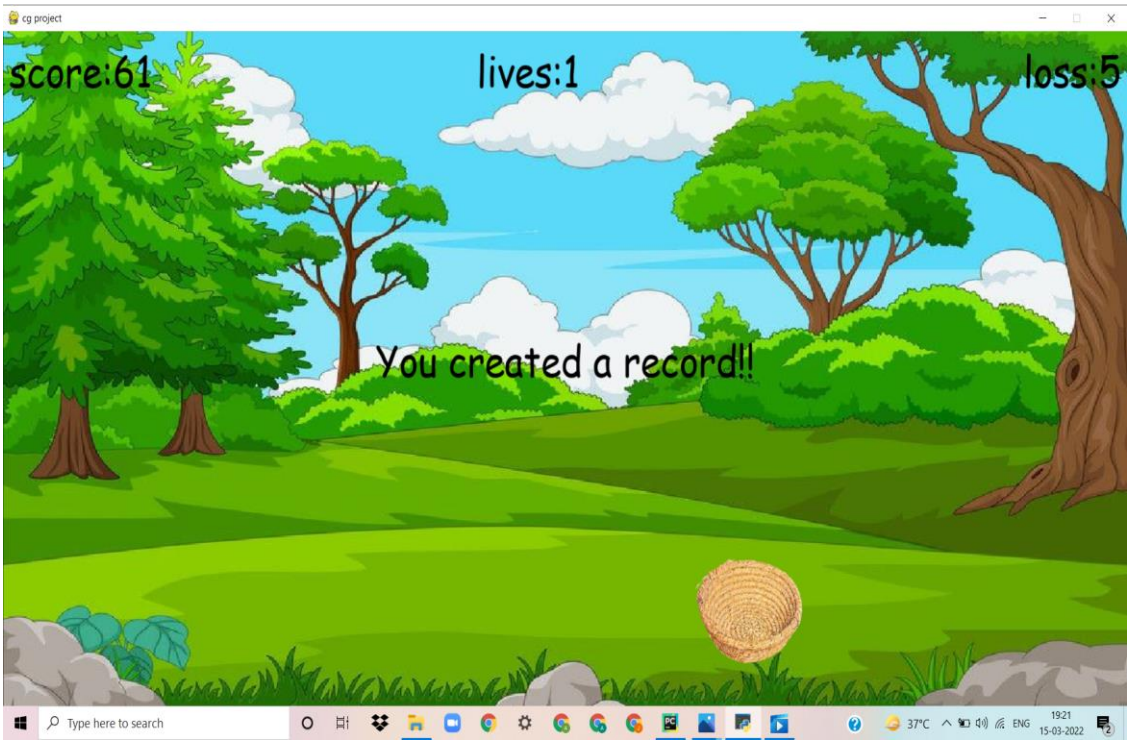


Main (): gaming loop

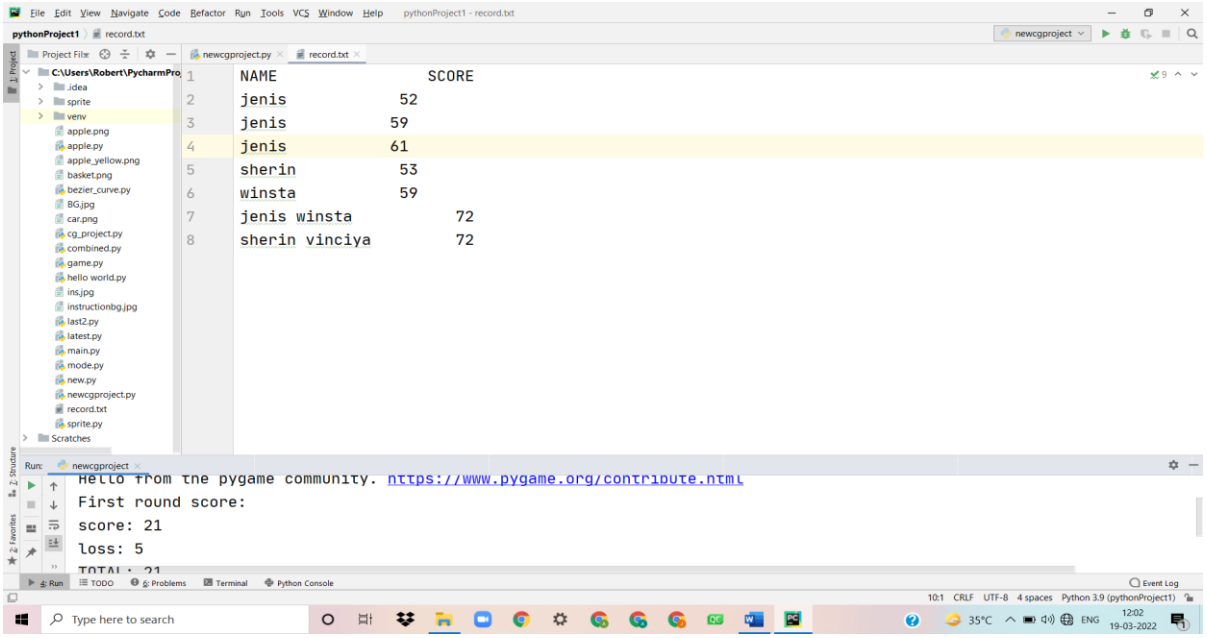



cg project
score:0
lives:3
loss:0
Type here to search
37°C Smoke
ENG
18:42
15-03-2022

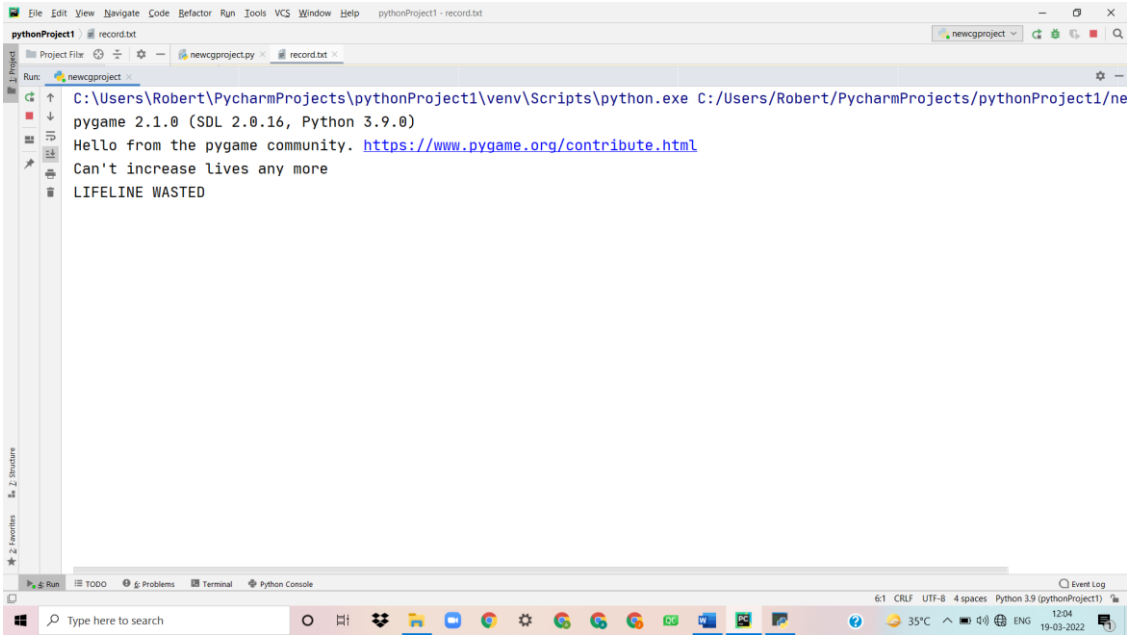
Record():


cg project
score:61
lives:1
loss:5
You created a record!!
Type here to search
37°C
ENG
19:21
15-03-2022

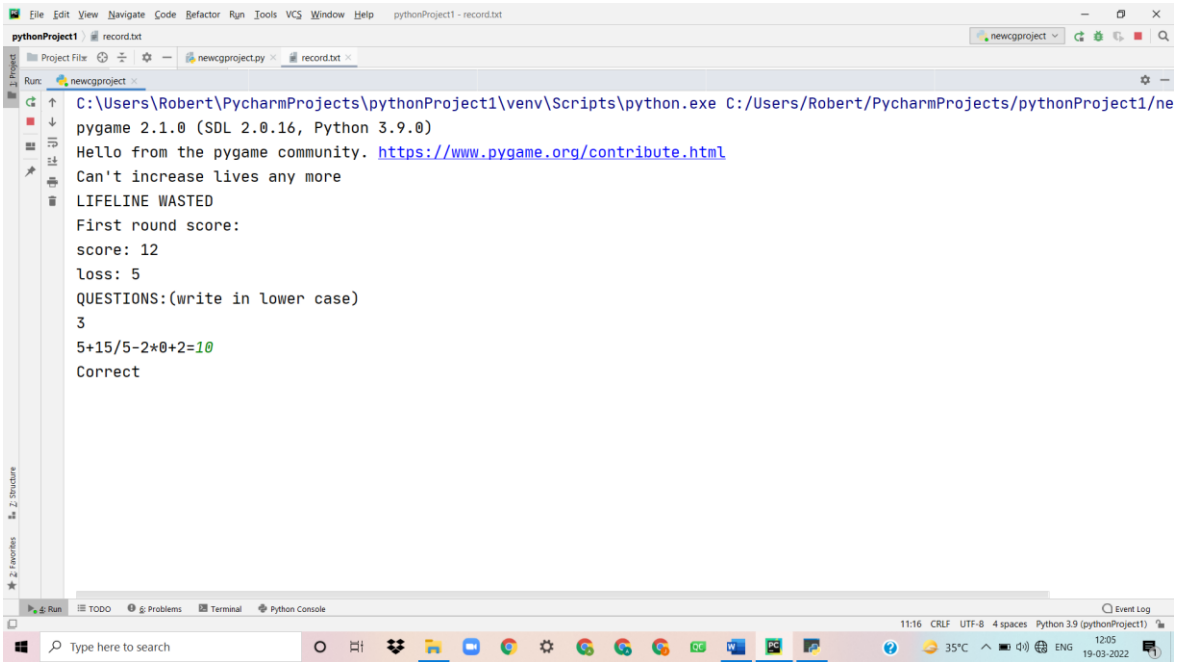
Record.txt:



Lifeline_fun(): lifeline wasted



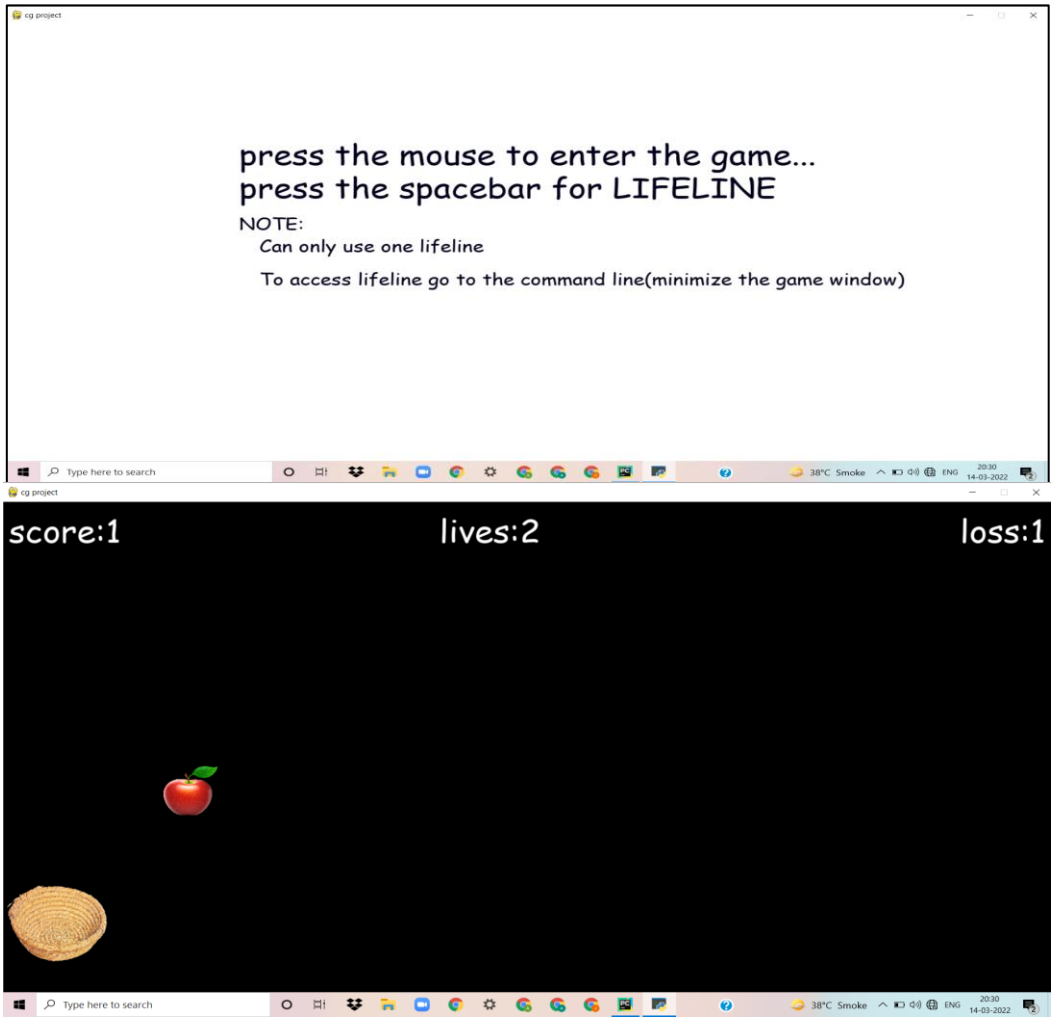
Lifeline_fun(): lifeline used



FEEDBACKS:

Sherin: Provide few more instruction for more clarity of the game. Give background because it would be more attractive.
Agnel: Minimizing the game and doing things in the console is confusing. The instruction about this seems technical like someone who know least about programming will not know what is console and all.

MODIFICATON AFTER FEEDBACK:



Earlier the game was as shown in the pictures above. After that according to certain feedback, I change the background and added few instructions.

But still I am not able to provide solution for the question of lifeline in the console. I had an idea of creating another loop but the game will further slow down as in the apples gets paused in between because when we had only one loop the game was very smooth as compared to that after 2 loops. So, still had to find some solution for that feedback.

REFERENCES:

https://www.greeksforgreeks.org

https://youtu.be/aequTxAvQq4

https://youtu.be/pm3Iwn6ZnDc

https://youtu.be/jdhFq7dork8

https://youtu.be/Q-__8Xw9KTM