EGG CATCHING GAME

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Under the supervision of

Mr. Ashish Tripathi

Assistant Professor, Department of Information Technology

By

AYUSH SINGH 1901640130021 3rd year

Pranveer Singh Institute of Technology

DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY

LUCKNOW

2021-2022

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to substantial extent has been accepted for the about of any degree or diploma of the university or other institute of higher learning except where due acknowledgement has been made in the text

Name: Ayush Singh

Roll no: 1901640130021

Date:

Signature:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech project undertaken during B. Tech 3rd year. We owe a special debt of gratitude to Assistant Professor Mr. Ashish Tripathi, Dept. of Information Technology, Pranveer Singh Institute of Technology, Kanpur for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been constant source of inspiration for us.

We also take the opportunity to acknowledge the contribution of Mr. Piyush Bhushan Singh, Head of Department, Information Technology, Pranveer Singh Institute of Technology, Kanpur for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and corporation during the development of our project. Last but not the least, we acknowledge our parents and friends for their contribution in the completion of our project.

Name: Ayush Singh

Signature:

CERTIFICATE

This is to certify that Project Report entitled “Egg Catching Game” which is submitted by Ayush Singh (1901640130021) of the fifth semester, in the year 2021-2022 in partial fulfilment of the requirement for the award of degree B. Tech in Department of Information Technology of Dr A.P.J. Abdul Kalam Technical University, is a record of the candidate own work carried out by him under Mr. Ashish Tripathi’s supervision. The matter embodied in this project is original and has not been submitted for award of any other degree.

Date:

PROJECT GUIDE:

Mr. Ashish Tripathi

Signature:

Ext. Examiner:

Signature:

ABSTRACT

Eggs Catcher is a classic game where the goal is to catch as many eggs as possible. In this game, every egg you catch will increase your score and if you miss 3 eggs you will lose the game. In this project, I will walk you through how to create an egg catcher game using Python.

An egg catcher game tests your concentration and the speed of your reflexes as you have to catch eggs falling all around the screen with one basket. To create an egg catcher game you have to design an animation of falling eggs at random positions and a basket will be catching all the falling eggs.

To create an eggs catcher game using Python, your first step will be to design a floor, basket, and eggs. Once your game starts, the eggs will gradually move across the floor, which will create an animation indicating that the eggs are falling. Then with the help of loops, we can constantly check that the eggs have been caught in the basket or have touched the ground. When the egg is caught or dropped, it will be an end of an event, so here the egg will be removed and the game has to adjust the score by increasing the score if the egg was caught in the basket or by decreasing your one life if the egg has touched the ground.

TABLE OF CONTENTS

Declaration ………………………………………………………………………………….. ii

Acknowledgement …………………………………………………………………….. iii

Certificate ……………………………………………………………………………... iv

Abstract ……………………………………………………………………………….. v

Table of contents ……………………………………………………………………… vi

CHAPTER 1 ………………………………………………………………………….. 1

INTRODUCTION ……………………………………………………………………… 1

1.1 Problem statement ………………………………………………………………… vii

1.2 Problem motivation ……………………………………………………………….. vii

CHAPTER 2………………………………………………………………………….. 2

PRELIMINARY STUDY……………………………………………………………….. viii

CHAPTER 3…………………………………………………………………………...... 3

MODULES……………………………………………………………………………… xi

3.1 itertools ……………..……………………………………………………….…. xi

3.2 random ……………………..…………………………………………………..... xi

3.3 Tkinter ……………………………………………………………………………… xii

CHAPTER 4

HARDWARE DESCRIPTION…………………………………………………………… 4

4.1. A Working Python Environment……………………………………………….... xiii

CHAPTER 5

SOFTWARE DESCRIPTION….……………………………………………………….. 5

5.1 Python Interepreter……………….……………………………………………… xiii

5.2 Visual Stdio Code…………………………………………………………………... xiv

CHAPTER 6

CODING……………………………………………………………………………….. 6

6 Code for Game…..……………………….………………………….………….……………….… xv

CHAPTER 7

TESTING………………………………………………………………………………………..7

7.1 System Testing For Game…………………...…………………………………………….xviii

CHAPTER 8

SNAPSHOTS……………………………………………………………………………....xx

CHAPTER 9

CONCLUSION and FUTURE SCOPE……………………………………………………….xxi

9.1 Conclusion………………………………………………………………………...xxi

9.2 Future Scope………………………………………………………………………xxi

CHAPTER 10

REFERENCES………………………………………………………………………..xxii

Problem statement

To create an egg catcher game you have to design an animation of falling eggs at random positions and a basket will be catching all the falling eggs.

Problem Motivation

Python is a multipurpose language and can be used in almost every field of development. Python can also be used to develop different type of game. Let’s try to develop a simple Catching the Egg game using Python and TKinter.

PRELIMINARY STUDY

1. Use Tkinter package in python for building GUI (Graphical user interface).

2. Use Canvas for drawing objects in Python – Canvas is a rectangular area intended for drawing pictures or other complex layouts. We can place graphics, text, widgets or frames on Canvas.

3. Game is very simple. There is one bar at the bottom of game window which can be moved left or right using the buttons that are in the game window. Red ball will continuously fall from top to bottom and can start from any random x-axis distance. The task is to bring that bar to a suitable location by moving left or right so that the egg will fall on that basket (catch the egg onto the basket) not on the ground. If player catches the egg onto the basket then score will get increase and that ball will disappear and again a new egg will start falling from top to bottom starting from random x-axis distance. If player miss the egg from catching it on the basket 3 times then you will lose the game and then finally scorecard will appear on the game window.

Problem Specification

To create an egg catcher game using Python, we need to use three different loops:

1. One to create new eggs.

2. Another to check if the catcher has caught an egg.

3. And the third loop to move eggs and to check if the eggs touched the ground.

Modules

To create this game, I will be using three Python modules:

1.itertools: to change the colours of the falling eggs.

2.random: to make the eggs appear at random positions.

3.Tkinter: to animate the game on the screen.

Python's Itertool is a module that provides various functions that work on iterators to produce complex iterators. This module works as a fast, memory-efficient tool that is used either by themselves or in combination to form iterator algebra.
For example, let's suppose there are two lists and you want to multiply their elements. There can be several ways of achieving this. One can be using the naive approach i.e by iterating through the elements of both the list simultaneously and multiply them. And another approach can be using the map function i.e by passing the mul operator as a first parameter to the map function and Lists as the second and third parameter to this function.

Python Random module is an in-built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random *actions such as generating random numbers, print random a value for a list or string, etc.*

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.
To create a tkinter app:

Importing the module – tkinter

Create the main window (container)

Add any number of widgets to the main window

Apply the event Trigger on the widgets.

HARDWARE DESCRIPTION

PC or a Laptop with Minimum 4 GB RAM and Intel i3 or above processor or AMD Ryzen 3 or above processor.

SOFTWARE DESCRIPTION

A working Python Environment Preferred Version – 3.6.2 to 3.10.0

Microsoft VS Code or Visual Studio Code is used with 3.9.0 python interpreter for my project with no additional dependencies.

- Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.
- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

- Visual Studio Code is a distribution of the Code - OSS repository with Microsoft-specific customizations released under a traditional Microsoft product license.
- Visual Studio Code combines the simplicity of a code editor with what developers need for their core edit-build-debug cycle. It provides comprehensive code editing, navigation, and understanding support along with lightweight debugging, a rich extensibility model, and lightweight integration with existing tools.

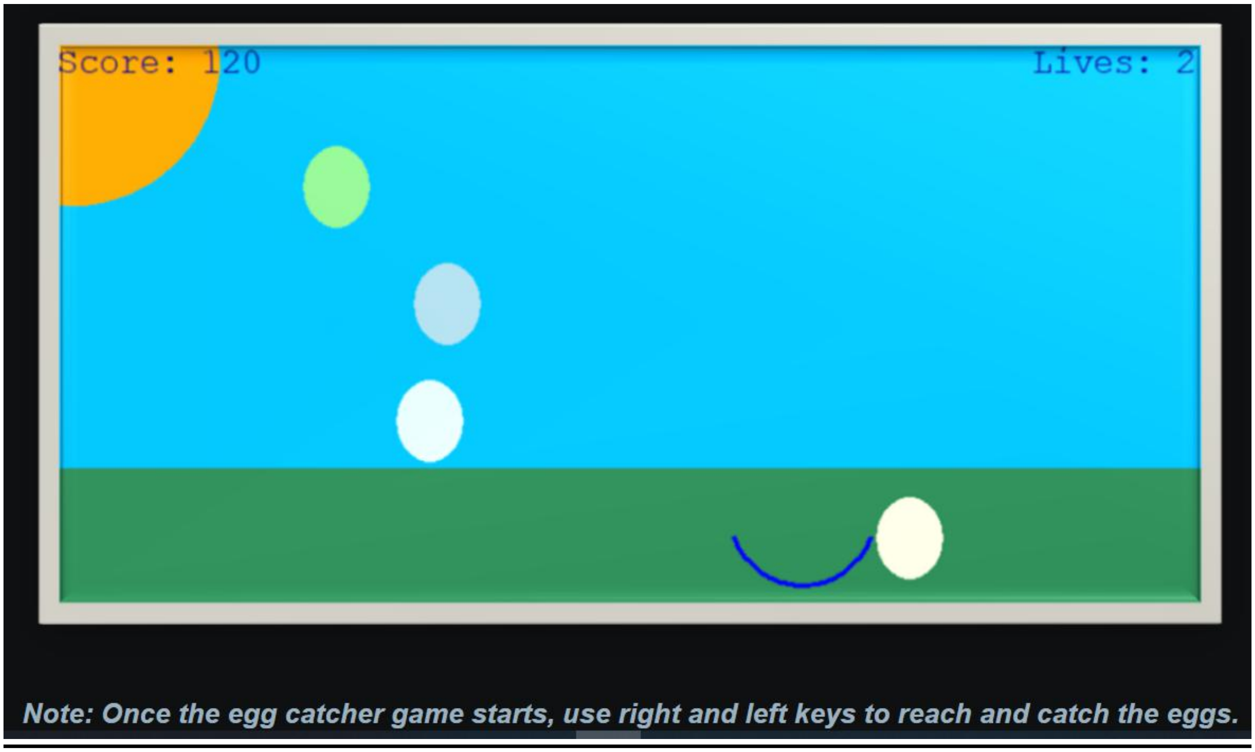
Code for Game

```
from itertools import cycle
from random import randrange
from tkinter import Canvas, Tk, messagebox, font

canvas_width = 800
canvas_height = 400

root = Tk()
c = Canvas(root, width=canvas_width, height=canvas_height, background="deep sky blue")
c.create_rectangle(-5, canvas_height-100, canvas_width+5, canvas_height+5, fill="sea green", width=0)
c.create_oval(-80, -80, 120, 120, fill='orange', width=0)
c.pack()

color_cycle = cycle(["light blue", "light green", "light pink", "light yellow", "light cyan"])
egg_width = 45
egg_height = 55
egg_score = 10
egg_speed = 500
egg_interval = 4000
difficulty = 0.95
catcher_color = "blue"
catcher_width = 100
catcher_height = 100
```

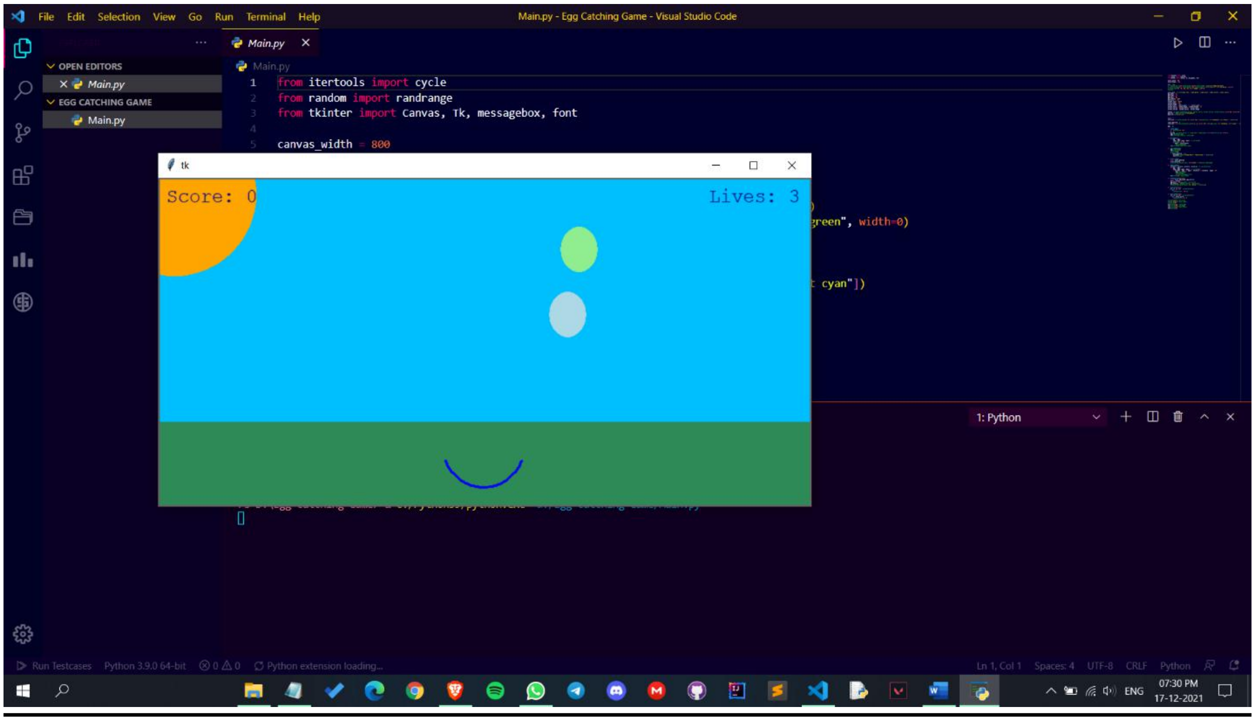
```
catcher_startx = canvas_width / 2 - catcher_width / 2
catcher_starty = canvas_height - catcher_height - 20
catcher_startx2 = catcher_startx + catcher_width
catcher_starty2 = catcher_starty + catcher_height

catcher = c.create_arc(catcher_startx, catcher_starty, catcher_startx2, catcher_starty2, start=200, extent=140, style="arc", outline=catcher_color, width=3)
game_font = font.nametofont("TkFixedFont")
game_font.config(size=18)



score = 0
score_text = c.create_text(10, 10, anchor="nw", font=game_font, fill="darkblue", text="Score: "+ str(score))

lives_remaining = 3
lives_text = c.create_text(canvas_width-10, 10, anchor="ne", font=game_font, fill="darkblue", text="Lives: "+ str(lives_remaining))

eggs = []

def create_egg():
  x = randrange(10, 740)
  y = 40
  new_egg = c.create_oval(x, y, x+egg_width, y+egg_height, fill=next(color_cycle), width=0)
  eggs.append(new_egg)
  root.after(egg_interval, create_egg)
```

```
def move_eggs():
    for egg in eggs:
        (eggx, eggy, eggx2, eggy2) = c.coords(egg)
        c.move(egg, 0, 10)
        if eggy2 > canvas_height:
            egg_dropped(egg)
    root.after(egg_speed, move_eggs)

def egg_dropped(egg):
    eggs.remove(egg)
    c.delete(egg)
    lose_a_life()
    if lives_remaining == 0:
        messagebox.showinfo("Game Over!", "Final Score: "+ str(score))
        root.destroy()

def lose_a_life():
    global lives_remaining
    lives_remaining -= 1
    c.itemconfigure(lives_text, text="Lives: "+ str(lives_remaining))

def check_catch():
    (catcherx, catchery, catcherx2, catchery2) = c.coords(catcher)
    for egg in eggs:
        (eggx, eggy, eggx2, eggy2) = c.coords(egg)
        if catcherx < eggx and eggx2 < catcherx2 and catchery2 - eggy2 < 40:
            eggs.remove(egg)
            c.delete(egg)
            increase_score(egg_score)
    root.after(100, check_catch)

def increase_score(points):
    global score, egg_speed, egg_interval
    score += points
    egg_speed = int(egg_speed * difficulty)
    egg_interval = int(egg_interval * difficulty)
    c.itemconfigure(score_text, text="Score: "+ str(score))

def move_left(event):
    (x1, y1, x2, y2) = c.coords(catcher)
    if x1 > 0:
        c.move(catcher, -20, 0)

def move_right(event):
    (x1, y1, x2, y2) = c.coords(catcher)
    if x2 < canvas_width:
        c.move(catcher, 20, 0)

c.bind("<Left>", move_left)
c.bind("<Right>", move_right)
c.focus_set()
root.after(1000, create_egg)
root.after(1000, move_eggs)
root.after(1000, check_catch)
root.mainloop()
```

System Testing and Snapshots



Note: Once the egg catcher game starts, use right and left keys to reach and catch the eggs.

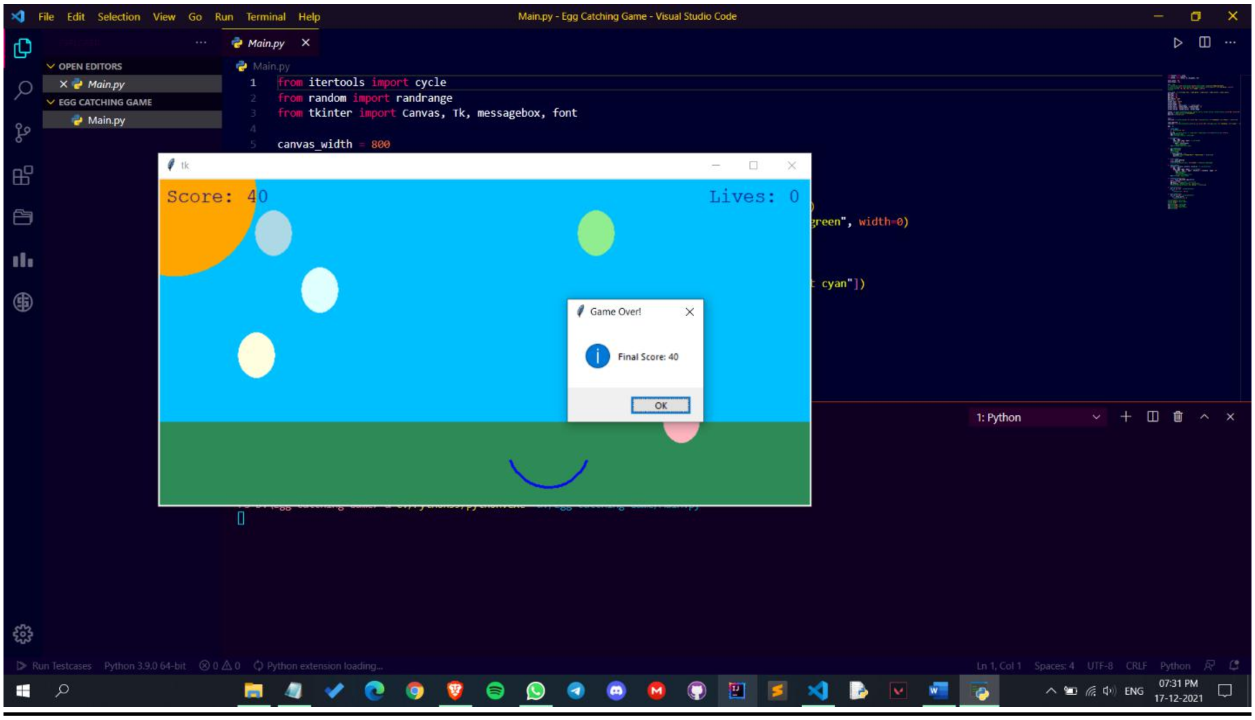


Main.py - Egg Catching Game - Visual Studio Code

```
from itertools import cycle
from random import randrange
from tkinter import Canvas, Tk, messagebox, font

canvas_width = 800
```

Score: 0

Lives: 3



Main.py - Egg Catching Game - Visual Studio Code

```
from itertools import cycle
from random import randrange
from tkinter import Canvas, Tk, messagebox, font

canvas_width = 800
```

Score: 40

Lives: 0

Game Over!

Final Score: 40

OK

Futurescope and Conclusion

Conclusion :-

The three looping functions are executed using a timer to ensure that they aren’t executed before the main loop. Then the mainloop() function is executed to run the egg catcher game.

Futurescope :-

A similar game can be developed by using Tkinter module and using the canvas to build a rectangle to define the dimensions of the playground.

An egg catcher game tests your concentration and the speed of your reflexes as you have to catch eggs falling all around the screen with one basket. To create an egg catcher game you have to design an animation of falling eggs at random positions and a basket will be catching all the falling eggs

Reference

1. https://docs.python.org/3/library/tkinter.html

   -Tkinter docs

2. https://docs.python.org/3/library/random.html
   -random module doc
3. https://docs.python.org/3/library/itertools.html

   -itertools module doc

4. https://thecleverprogrammer.com/2021/04/23/egg-catcher-game-using-python/
   -Game reference