**PROJECT REPORT**

**ON**

Name of Project

****

REPORT SUBMITTED

TO

VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

FOR THE PBL OF PYTHON FOR ENGINEERS

IN

**ENGINEERING AND APPLIED SCIENCE DEPARTMENT**

BY

**Aum Hari Giri – 2029/22010477**

**Atharva Bhale- 2030/ 22010493**

**Samiksha Kadam - 2035/ 22010573**

**Class: F.Y Division: B Batch:B2**

**Batch Teacher**

Prof. Deepti Pandit

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Contents** | **Page No.** |
| **1** | **ABSTRACT** |  |
| **2** | **Introduction and Theory** |  |
| **3** | **FLOW of Program** |  |
| **4** | **source code of the program** |  |
| **5** | **output** |  |
| **6** | **LIST OF THE TOPICS/ CONCEPT WHICH ARE COVERED FROM THE SYLLABUS** |  |
| **7** | **LIST OF THE TOPICS/ CONCEPT WHICH ARE OUT OF SYLLABUS** |  |
| **8** | **CONCLUSION** |  |
| **9** | **REFERENCES** |  |

**ABSTRACT:**

**Nowadays, many games are being developed using high level programming languages including python.**

**In this project we have developed a game named “The Ball Catcher Game”. We have used modules like Pygame, tkinter which have inbuilt GUI. We have also used basics of python to make game variables like loops, functions, etc.**

**It is a single player game and the gameplay design is simple so that the user won’t find it difficult to use and navigate.**

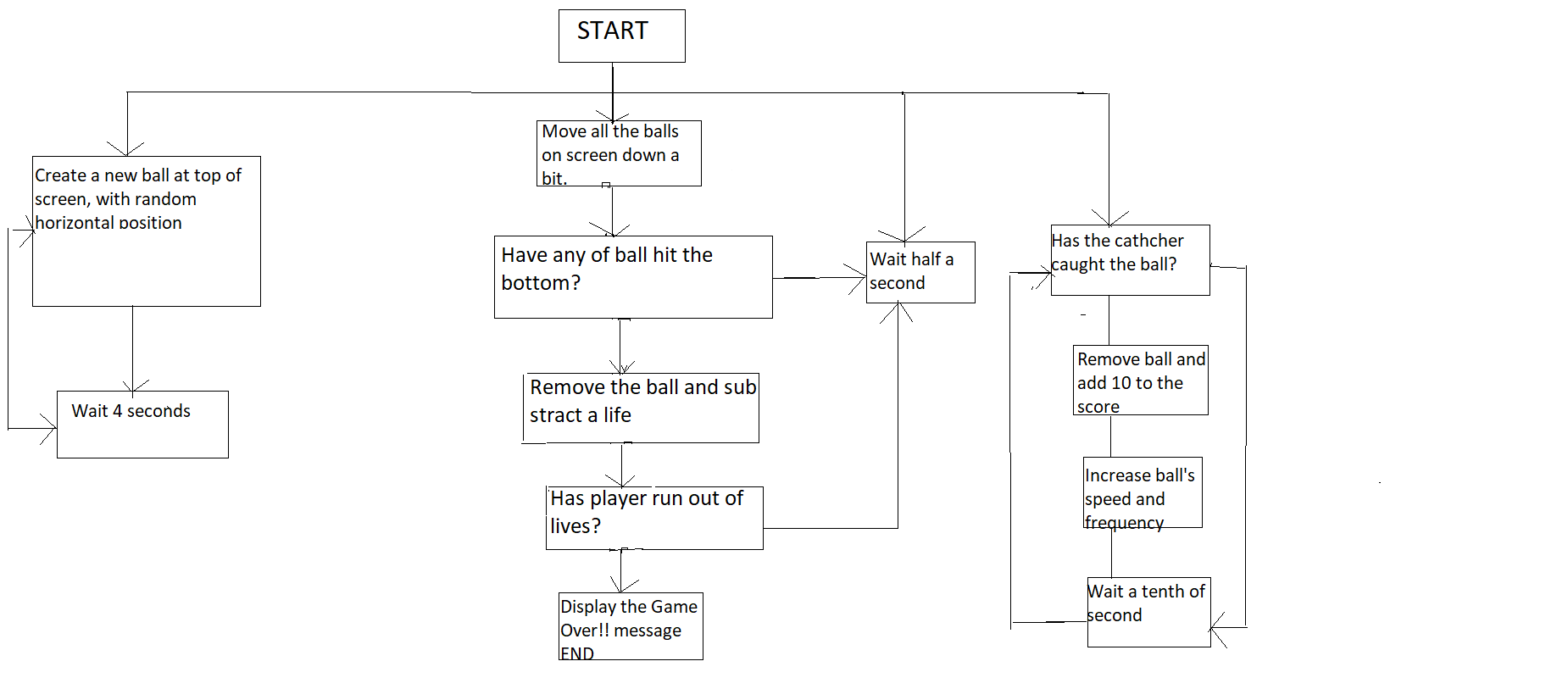
**Introduction and Theory:**

The game that we have created is a catching game in which the user has to catch the falling balls by moving the arc(basket) present at the bottom of the screen using the arrow keys. The player has 3 lives and one life is subtracted if he drops 1 ball and 10 points are awarded for every successful catch. We have created this game using many modules but most of the code has been written using tkinter. Tkinter is python’s standard GUI framework and comes bundled with python 3. The game that we have created is very simple to play and using tkinter makes it more interactive and adds colorful graphics to it. The canvas widget on Tkinter is used to draw geometrical objects on the window. It has methods for drawing various shapes. We have used the following methods of canvas in the game :-

* **Create\_oval():** To draw circle
* **Create\_text():** To print text on the window
* **delete():** Clear the drawn objects from the window

We have also used pygame module and mixer function to play audio files during the game. The itertools, datetime, time and random are some of the other modules which we have used to make the game more user friendly and make the game more fun to play.

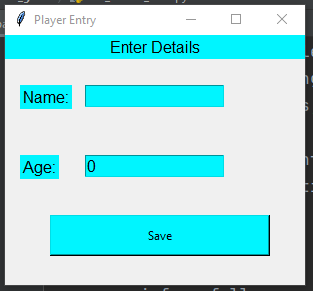
**FLOW of Program:**

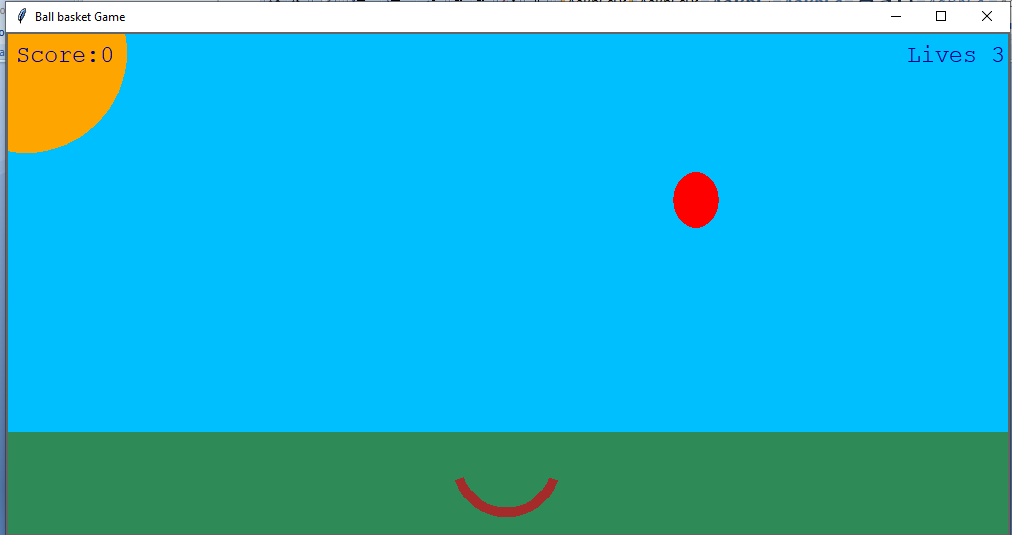
****

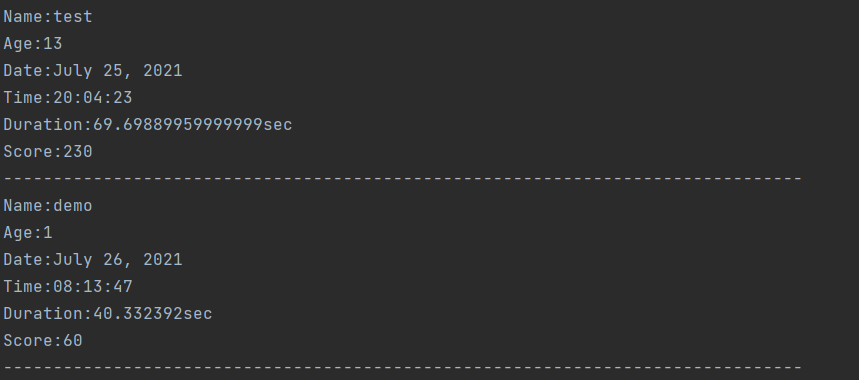
**source code of the program:**

from itertools import cycle  
from random import randrange  
from tkinter import Canvas, Tk, messagebox, font, Label, StringVar, IntVar, Entry, Button  
import pygame  
from time import perf\_counter as my\_timer  
from datetime import datetime  
  
  
def save\_info():  
 name\_info = fullname.get()  
 age\_info = age.get()  
  
 file = open("Player data.txt", "a")  
  
 file.write("Name:" + name\_info)  
  
 file.write("\n")  
  
 file.write("Age:" + str(age\_info))  
  
 file.write("\n")  
  
 file.write("Date:" + date)  
  
 file.write("\n")  
  
 file.write("Time:" + start\_time1)  
  
 file.write("\n")  
 file.close()  
 data\_screen.destroy()  
  
  
data\_screen = Tk()  
  
now = datetime.now()  
date = now.strftime("%B %d, %Y")  
start\_time1 = now.strftime("%H:%M:%S")  
  
data\_screen.geometry("300x250")  
data\_screen.title("Player Entry")  
heading = Label(text='Enter Details', bg='turquoise1', fg='Black', width='300', font='10')  
heading.pack()  
  
fullname = Label(text="Name:", bg='turquoise1', fg='black', font='10')  
age = Label(text='Age:', bg='turquoise1', fg='black', font='10')  
fullname.place(x=15, y=50)  
age.place(x=15, y=120)  
  
fullname = StringVar()  
age = IntVar()  
  
fullname\_entry = Entry(textvariable=fullname, bg='turquoise1', fg='black', width="15", font='10')  
age\_entry = Entry(textvariable=age, bg='turquoise1', fg='black', width="15", font='10')  
  
fullname\_entry.place(x=80, y=50)  
age\_entry.place(x=80, y=120)  
  
save\_button = Button(data\_screen, text="Save", command=save\_info, bg="turquoise1", width="30", height="2")  
save\_button.place(x=45, y=180)  
data\_screen.mainloop()  
  
start\_time = my\_timer()  
canvas\_width = 1000  
canvas\_height = 500  
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()  
  
pygame.mixer.init()  
pygame.mixer.music.load('music.mp3')  
pygame.mixer.music.set\_volume(0.5)  
pygame.mixer.music.play(loops=-1)  
ball\_drop\_sound = pygame.mixer.Sound('drop.wav')  
splash\_sound = pygame.mixer.Sound('splash.wav')  
game\_over\_sound = pygame.mixer.Sound('game\_over.wav')  
  
# variable initialization  
color\_cycle = cycle(['red', 'light green', 'light pink', 'light yellow', 'orange'])  
ball\_width = 45  
ball\_height = 55  
ball\_score = 10  
ball\_speed = 300  
ball\_interval = 4000  
difficulty\_factor = 0.95  
  
basket\_color = 'brown'  
basket\_width = 100  
basket\_height = 100  
basket\_start\_x = canvas\_width/2 - basket\_width/2  
basket\_start\_y = canvas\_height - basket\_height-20  
basket\_start\_x2 = basket\_start\_x + basket\_width  
basket\_start\_y2 = basket\_start\_y + basket\_height  
  
basket = c.create\_arc(basket\_start\_x, basket\_start\_y, basket\_start\_x2, basket\_start\_y2, start=200, extent=140,  
 style='arc', outline=basket\_color, width=10)  
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='dark blue', text='Score:'+str(score))  
  
lives\_remaining = 3  
lives\_text = c.create\_text(canvas\_width-1, 10, anchor='ne', font=game\_font,  
 fill='dark blue', text='Lives '+str(lives\_remaining))  
  
balls = [] # empty list for adding balls  
  
  
def create\_ball():  
 x = randrange(10, 940)  
 y = 40  
 new\_ball = c.create\_oval(x, y, x+ball\_width, y+ball\_height, fill=next(color\_cycle), width=0)  
 balls.append(new\_ball)  
 root.after(ball\_interval, create\_ball)  
  
  
def move\_balls():  
 for ball in balls:  
 (ball\_x, ball\_y, ball\_x2, ball\_y2) = c.coords(ball)  
 c.move(ball, 0, 10)  
 if ball\_y2 > canvas\_height:  
 ball\_dropped(ball)  
 root.after(ball\_speed, move\_balls)  
  
  
def ball\_dropped(ball):  
 balls.remove(ball)  
 c.delete(ball)  
 splash\_sound.play()  
  
 lose\_a\_life()  
 if lives\_remaining == 0:  
 game\_over\_sound.play()  
 end\_time = my\_timer()  
 duration = end\_time - start\_time  
 c.destroy()  
 after\_loss\_life(duration)  
  
  
def lose\_a\_life():  
 global lives\_remaining  
 lives\_remaining -= 1  
 c.itemconfigure(lives\_text, text='Lives: '+str(lives\_remaining))  
  
  
def check\_catch():  
 (basket\_x, basket\_y, basket\_x2, basket\_y2) = c.coords(basket)  
 for ball in balls:  
 (ball\_x, ball\_y, ball\_x2, ball\_y2) = c.coords(ball)  
 if basket\_x < ball\_x and ball\_x2 < basket\_x2 and basket\_y2 - ball\_y2 < 40:  
 ball\_drop\_sound.play()  
  
 balls.remove(ball)  
 c.delete(ball)  
 increase\_score(ball\_score)  
 root.after(100, check\_catch)  
  
  
def after\_loss\_life(duration):  
  
 file = open("Player data.txt", 'a')  
 file.write("Duration:" + str(duration) + "sec")  
 file.write("\n")  
 file.write("Score:" + str(score))  
 file.write("\n")  
 file.write("-"\*80)  
 file.write("\n")  
 file.close()  
 messagebox.showinfo('Game Over!', 'Final Score: '+str(score)+"\nDuration:"+str(round(duration, 2))+"sec")  
 root.destroy()  
  
  
def increase\_score(points):  
 global score, ball\_speed, ball\_interval  
 score += points  
 ball\_speed = int(ball\_speed \* difficulty\_factor)  
 ball\_interval = int(ball\_interval \* difficulty\_factor)  
 c.itemconfigure(score\_text, text='Score: '+str(score))  
  
  
def move\_left(events):  
 (x1, y1, x2, y2) = c.coords(basket)  
 if x1 > 0:  
 c.move(basket, -20, 0)  
  
  
def move\_right(events):  
 (x1, y1, x2, y2) = c.coords(basket)  
 if x2 < canvas\_width:  
 c.move(basket, 20, 0)  
  
  
c.bind('<Left>', move\_left)  
c.bind('<Right>', move\_right)  
c.focus\_set()  
  
root.after(1000, create\_ball)  
root.after(1000, move\_balls)  
root.after(1000, check\_catch)  
  
root.title("Ball basket Game")  
root.mainloop()

**output:**

****

****

****

**LIST OF THE TOPICS/ CONCEPT WHICH ARE COVERED FROM THE SYLLABUS:**

* **Importing modules**
* **Declaring and initializing variables**
* **Creating functions**
* **List**
* **Creating text file**
* **Data type conversion.**

**LIST OF THE TOPICS/ CONCEPT WHICH ARE OUT OF SYLLABUS:**

* **Pygame : It is a set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with Python programming language.**
* **Tkinter : Most commonly used method from all GUI methods. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with Tkinter is fastest and easiest way to create GUI applications.**
* **Itertools : It is a module that provides various functions that works on iterators to produce complex iterators. It is a fast and memory-efficient tool.**
* **Date and time module : It supplies classes to work with date and time. This classes provide a number of functions to deal with dates,times and time interval.**

**CONCLUSION:**

**We have used different types of functions, modules that manage graphics, animations, sounds of the game making the game more exciting.**

**We learnt adding backgrounds, images using tkinter also adding music using Pygame and many more which helped us to focus on higher-level logics og game dynamics.**

**Creating canvas of specific size, ball size, color, iteration, increase of score and difficulty factors, adding different sounds at different intervals were some innovative tasks to learn for us.**

**REFERENCES:**