**CATCH BALL**

A

Mini Project Report

Submitted in partial fulfilment of the Requirements for the award of the Degree of

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

S. Bharath Reddy, 1602-20-737-007

B. Madhukar Reddy, 1602-20-737-023

V. Sujith Kumar, 1602-20-737-049



**Department of InformationTechnology**

**Vasavi College of Engineering (Autonomous)**

**ACCREDITED BY NAAC WITH 'A++' GRADE**

**(Affiliated to Osmania University and Approved by AICTE) Ibrahimbagh, Hyderabad-31**

**2022**

**Vasavi College of Engineering**

**(Autonomous)**

**(Affiliated to Osmania University) Hyderabad-500 031**

### Department of Information Technology



**DECLARATION BY THE CANDIDATE**

We**, S. Bharath Reddy , B. Madhukar Reddy** and  **V. Sujith Kumar**

bearing hall ticket numbers, **1602-20-737-007**, **, 1602-20-737-023** and **1602-20-737-049** hereby declare that the project report entitled is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**S.Bharath Reddy**

, **1602-20-737-007**

**B.Madhukar Reddy**

**1602-20-737-023**

**V.Sujith Kumar**

**1602-20-737-049**

(Faculty In-Charge) (Head,Dept of IT)

### Vasavi College of Engineering (Autonomous)

### (Affiliated to Osmania University) Hyderabad-500 031

**Department of Information Technology**



**BONAFIDE CERTIFICATE**

This is to certify that the project entitled “**CATCH BALL**” being submitted by **S. Bharath Reddy , B.Madhukar Reddy** and  **V.Sujith Kumar**

bearing hall ticket numbers, **1602-20-737-007**, **, 1602-20-737-023** and

**1602-20-737-049,** in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in **Information Technology** is a record of bonafide work carried out by him/her under my guidance.

**Dr.K.RamMohanRao,, Professor & HOD, Dept. of IT,**

**ACKNOWLEDGMENT**

We extend our sincere thanks to Dr. S. V. Ramana, Principal, Vasavi College of Engineering for his encouragement. We express our sincere gratitude to Dr. K. Ram Mohan Rao, Professor & Head, Department of Information Technology, Vasavi College of Engineering, for introducing the Mini-Project module in our curriculum, and also for his suggestions, motivation, and co-operation for the successful completion of our Mini Project.

We also want to thank and convey our gratitude towards our mini project coordinators Ms. Lingineni Divya and Sriramoju Rajyalaxmi, for guiding us in understanding the process of project development & giving us timely suggestions at every phase.

We would also like to sincerely thank the project reviewers for their valuable inputs and suggestions.

**Table of Contents**

1. ABSTRACT 6
2. INTRODUCTION 7

a.OBJECTIVE OF THE PROJECT

1. SYSTEM REQUIRNMENTS ..8
   1. HARDWARE REQUIREMENT
   2. SOFTWARE REQUIREMNET
2. PROPOSED WORK 9
   1. DESIGN………………………………………………9
   2. IMPLEMENTATION 12
   3. TESTING……………………………………………..20

5. GIT HUB LINK…………………………………………….22

6. RESULTS…………………………………………………...23

7. OUTCOMES APART FROM CURRICULUM…………….26

8. CONCLUSION ……………………………………………...26

9. REFERENCES……………………………………………...26

**1. ABSTRACT**

In this project we present a ball catcher embedded with Tkinter module. The project file only contains a python script .The game play Graphics is good enough and the controls are simple for the users. Talking about the game play, it is very easy to play

The player just has to catch balls by moving the basket left and right in order to gain score point. At each turn, the ball color changes randomly.

A simple GUI is provided for the easy game play. It’s easy to operate and understand by users. The game play design is so simple that user won’t find it difficult to use and navigate.

**2. INTRODUCTION**

The Catch Ball in Python is a simple project which is engaging and enjoyable , which reduces stress. The user can start the game by clicking on the start button. Also, you can choose the type of level, you want to play. The user should catch the balls which are randomly falling down and for every ball catched the scored gets increased by 10 and if the user misses 3 balls then the game ends.

**OBJECTIVE OF THE PROJECT:**

The Objective of the project is to build a game using python which is easy to play and make player get rid of stress

**3. TECHNOLOGY**

**a) Hardware Requirements:**

* Minimum Ram required: 512 mb
* Minimum Disk Space required: 50 mb
* Processor: core i3 or above

## b) Software Requirements:

* Python 3.9 Latest Build
* Windows Desktop version (7/8/10/11)
* PyCharm IDE for windows devices
* Pydroid3 for Android devices

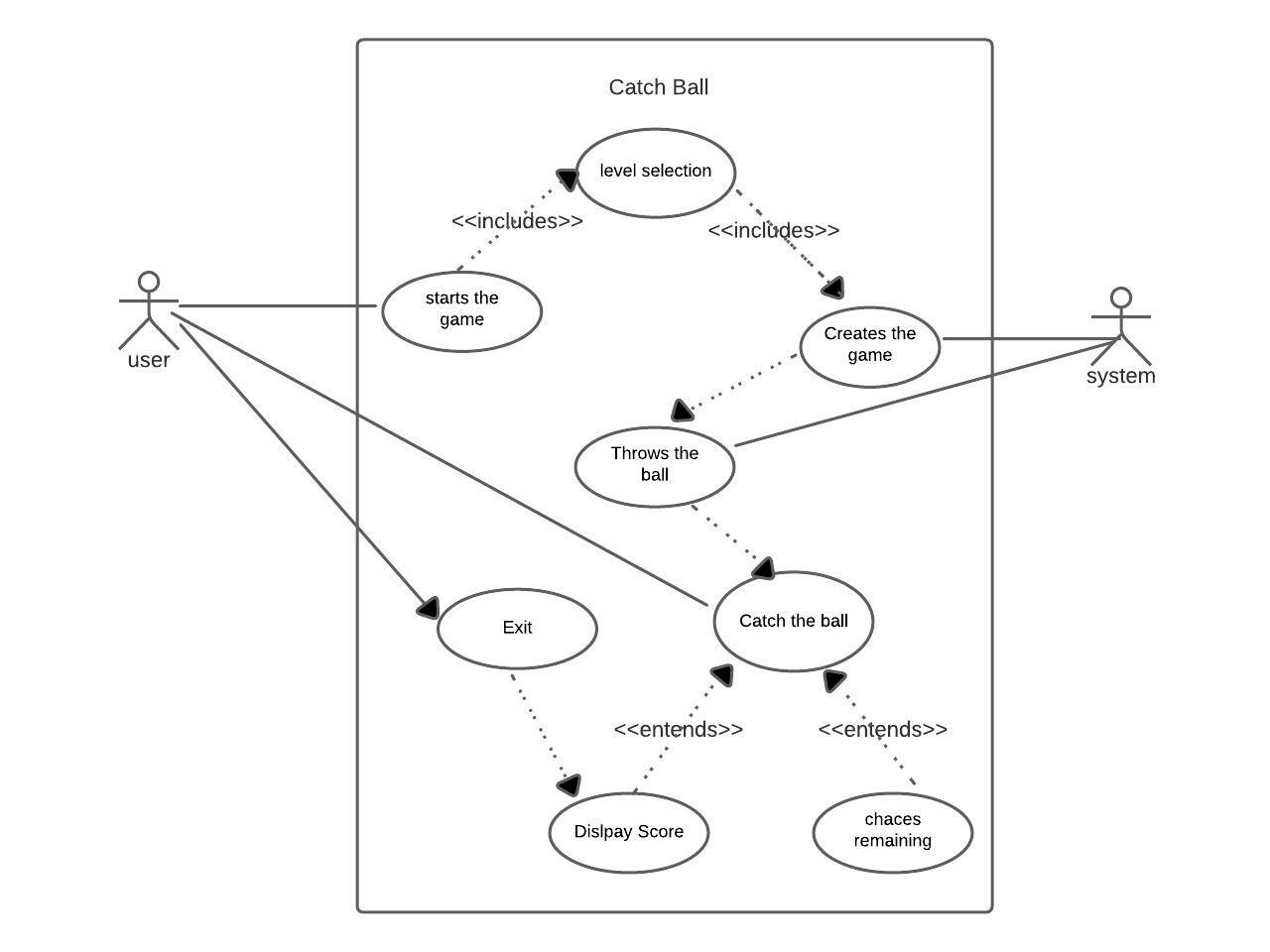
**4. PROPOSED WORK**

**a) DESIGN**

* **USE CASE DIAGRAM**

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how).

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

****

**USE CASE DESCRIPTION**

**User**

The user has to run the program and start the game then after user has to select mode (easy,medium,hard) after selecting the mode the game starts

**System**

The System builds the GUI and starts the game and throws the balls

**Score**

The user gets 10 points if user catches a ball and the score gets incremented for every ball catched

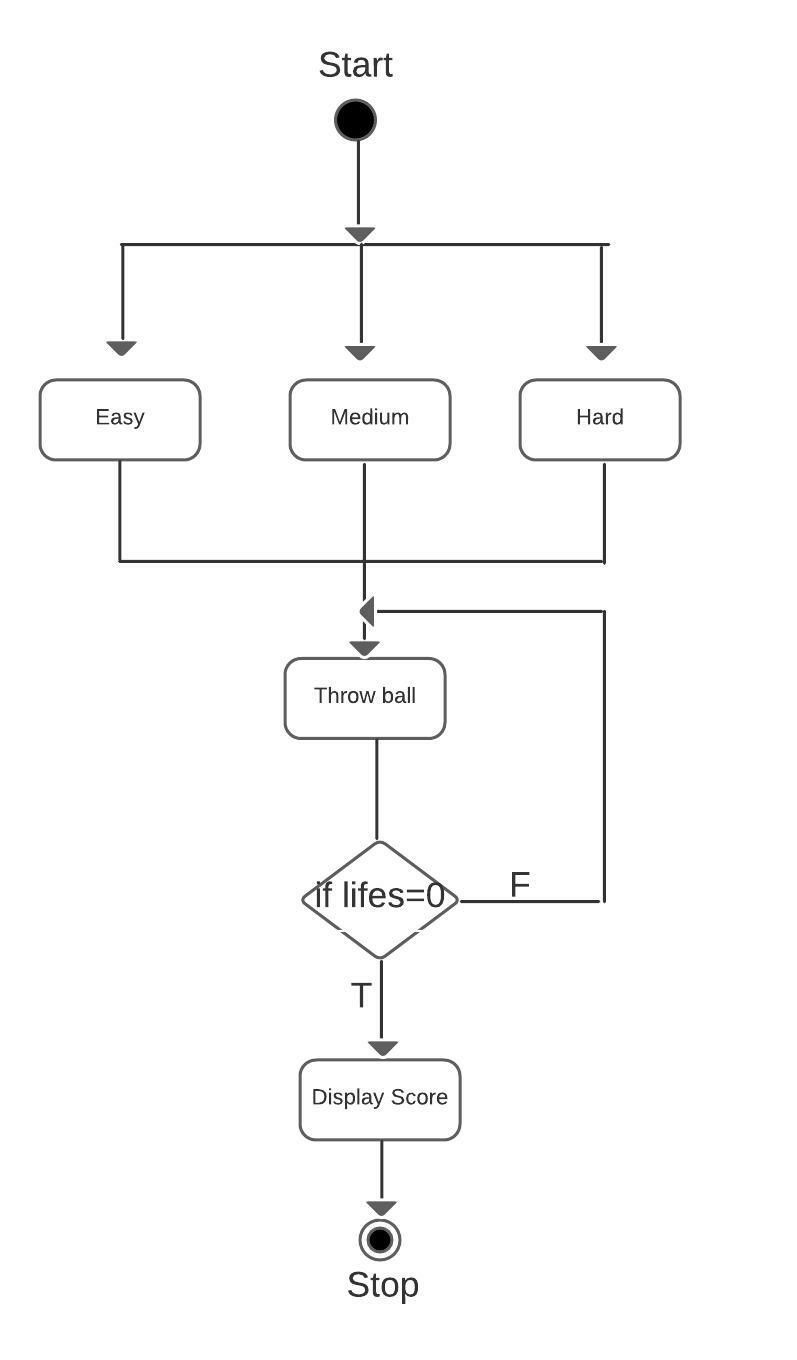
**Life**

The user gets 3 lives and the lives gets reduced if the user miss the ball

**Exit**

The game ends when the lives become 0 or if the user selects the end button, the system at last displays the score

**b) ACTIVITY DIAGRAM**

****

**b) IMPLEMENTATION**

* **MODULE WISE CODE**

Main\_start.py:

from tkinter import \*  
  
  
def start\_main\_page():  
 def start\_game(args):  
 main\_window.destroy()  
 if args == 1:  
 from Options1 import easy  
 easy.root.mainloop()  
 elif args == 2:  
 from Options1 import medium  
 medium.root.mainloop()  
 elif args == 3:  
 from Options1 import hard  
 hard.root.mainloop()  
  
 def option():  
  
 sel\_btn1 = Button(text="Easy",width=18,borderwidth=8,font=("", 18),fg="#000000",bg="#99ffd6",cursor="hand2",command=lambda: start\_game(1),)  
 sel\_btn2 = Button(text="Medium",width=18,borderwidth=8,font=("", 18),fg="#000000",bg="#99ffd6",cursor="hand2",command=lambda: start\_game(2),)  
 sel\_btn3 = Button(text="Hard",width=18,borderwidth=8,font=("", 18),fg="#000000",bg="#99ffd6",cursor="hand2",command=lambda: start\_game(3),)  
 ##lab\_img1.grid(row=0, column=0, padx=20)  
 sel\_btn1.grid(row=0, column=4, pady=(50, 0), padx=50, )  
 sel\_btn2.grid(row=6, column=4, pady=(100, 0), padx=70, )  
 sel\_btn3.grid(row=12, column=4, pady=(130, 0), padx=90, )  
  
 def show\_option():  
 start\_btn.destroy()  
  
 lab\_img.destroy()  
  
 option()  
  
 main\_window = Tk()  
  
 main\_window.geometry("500x500+500+150")  
 main\_window.resizable(0, 0)  
 main\_window.title("Catch Ball")  
 main\_window.configure(background="#000000")  
 img0 = PhotoImage(file="catch.png")  
 lab\_img = Label(main\_window,image=img0,bg='#e6fff5',)  
 lab\_img.pack(pady=(50, 0))  
 start\_btn = Button(main\_window,text="Start",width=18,borderwidth=8,fg="#000000",bg="#78cc0a",font=("", 13),cursor="hand2",command=show\_option,)  
 start\_btn.pack(pady=(50, 20))  
 main\_window.mainloop()  
  
  
start\_main\_page()

easy.py::

from itertools import cycle  
from random import randrange  
from tkinter import Canvas, Tk, messagebox, font,Button  
#from tkinter import \*  
from winsound import \*  
import pygame  
  
pygame.mixer.init()  
def windowdestroy():  
 messagebox.showinfo("Game Over!", "Final Score: " + str(score))  
 root.destroy()  
def play():  
  
  
 pygame.mixer.Sound("C:\\Users\\Ram Reddy\\Desktop\\Space-Invaders-Pygame-master\\background.wav")  
 pygame.mixer.music.set\_volume(0.7)  
 pygame.mixer.Sound.play(loops=-1)  
  
canvas\_width = 800  
canvas\_height = 600  
  
root = Tk()  
  
#play()  
c = Canvas(root, width=canvas\_width, height=canvas\_height, background="black")  
c.create\_rectangle(-5, canvas\_height-100, canvas\_width+5, canvas\_height+5, fill="#859923", width=0)  
c.create\_oval(-80, -80, 120, 120, fill='black', width=0)  
c.pack()  
c.create\_text(canvas\_width/2,20,fill="darkblue",font="Times 20 italic bold",text="Catch Ball")  
btn = Button(root, text='Exit', width=5,height=2, bd='5', command=windowdestroy)  
btn.place(x=0, y=0)  
  
  
  
color\_cycle = cycle(["light blue", "light green", "light pink", "light yellow", "light cyan"])  
ball\_width = 25  
ball\_height = 25  
ball\_score = 10  
ball\_speed = 600  
ball\_interval = 5000  
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="chord", outline=catcher\_color, width=9)  
game\_font = font.nametofont("TkFixedFont")  
game\_font.config(size=18)  
  
  
score = 0  
score\_text = c.create\_text(670, 10, anchor="nw", font=game\_font, fill="white", text="Score: "+ str(score))  
  
lives\_remaining = 3  
lives\_text = c.create\_text(canvas\_width-10, 37, anchor="ne", font=game\_font, fill="white", text="Lives: "+ str(lives\_remaining))  
  
balls = []  
#def sound():  
# play = lambda: PlaySound('Sound.wav', SND\_FILENAME)  
i=0  
  
  
def create\_ball():  
 global i  
 if(i==0):  
  
 effects = pygame.mixer.Sound('music2.wav')  
 effects.play()  
 x = randrange(10, 740)  
 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)  
 i=i+1  
  
def move\_balls():  
 for ball in balls:  
 (ballx, bally, ballx2, bally2) = c.coords(ball)  
 c.move(ball, 0, 10)  
 if bally2 > canvas\_height:  
 ball\_dropped(ball)  
 root.after(ball\_speed, move\_balls)  
  
def ball\_dropped(ball):  
 balls.remove(ball)  
 #sound()  
 c.delete(ball)  
 lose\_a\_life()  
 if lives\_remaining == 0:  
 messagebox.showinfo("Game Over!", "Final Score: "+ str(score))  
 root.destroy()  
  
def lose\_a\_life():  
 global lives\_remaining  
 effects = pygame.mixer.Sound('laser.wav')  
 effects.play()  
 lives\_remaining -= 1  
 c.itemconfigure(lives\_text, text="Lives: "+ str(lives\_remaining))  
  
def check\_catch():  
 (catcherx, catchery, catcherx2, catchery2) = c.coords(catcher)  
 for ball in balls:  
 (ballx, bally, ballx2, bally2) = c.coords(ball)  
 if catcherx < ballx and ballx2 < catcherx2 and catchery2 - bally2 < 40:  
 balls.remove(ball)  
 c.delete(ball)  
 increase\_score(ball\_score)  
 root.after(100, check\_catch)  
  
def increase\_score(points):  
 effect = pygame.mixer.Sound('explosion.wav')  
 effect.play()  
 global score, ball\_speed, ball\_interval  
 score += points  
 ball\_speed = int(ball\_speed \* difficulty)  
 ball\_interval = int(ball\_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\_ball)  
root.after(1000, move\_balls)  
root.after(1000, check\_catch)  
root.mainloop()

medium.py::

from itertools import cycle  
from random import randrange  
from tkinter import Canvas, Tk, messagebox, font,Button  
import pygame  
  
pygame.mixer.init()  
def windowdestroy():  
 messagebox.showinfo("Game Over!", "Final Score: " + str(score))  
 root.destroy()  
  
canvas\_width = 800  
canvas\_height = 600  
  
root = Tk()  
c = Canvas(root, width=canvas\_width, height=canvas\_height, background="black")  
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='black', width=0)  
c.pack()  
c.create\_text(  
 canvas\_width/2,20,  
 fill="darkblue",  
 font="Times 20 italic bold",  
 text="Catch Ball")  
btn = Button(root, text='Exit', width=5,height=2, bd='5', command=windowdestroy)  
btn.place(x=0, y=0)  
  
color\_cycle = cycle(["light blue", "light green", "light pink", "light yellow", "light cyan"])  
ball\_width = 25  
ball\_height = 25  
ball\_score = 10  
ball\_speed = 400  
ball\_interval = 3500  
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="chord", outline=catcher\_color, width=9)  
game\_font = font.nametofont("TkFixedFont") #arc, chord, or pieslice  
game\_font.config(size=18)  
  
  
score = 0  
score\_text = c.create\_text(670, 10, anchor="nw", font=game\_font, fill="white", text="Score: "+ str(score))  
  
lives\_remaining = 3  
lives\_text = c.create\_text(canvas\_width-10, 30, anchor="ne", font=game\_font, fill="white", text="Lives: "+ str(lives\_remaining))  
  
balls = []  
i=0  
def create\_ball():  
 global i  
 if (i == 0):  
 effects = pygame.mixer.Sound('music2.wav')  
 effects.play()  
 x = randrange(10, 740)  
 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)  
 i=i+1  
  
def move\_balls():  
 for ball in balls:  
 (ballx, bally, ballx2, bally2) = c.coords(ball)  
 c.move(ball, 0, 10)  
 if bally2 > canvas\_height:  
 ball\_dropped(ball)  
 root.after(ball\_speed, move\_balls)  
  
def ball\_dropped(ball):  
 balls.remove(ball)  
 c.delete(ball)  
 lose\_a\_life()  
 if lives\_remaining == 0:  
 messagebox.showinfo("Game Over!", "Final Score: "+ str(score))  
 root.destroy()  
  
def lose\_a\_life():  
  
 global lives\_remaining  
 effects = pygame.mixer.Sound('laser.wav')  
 effects.play()  
 lives\_remaining -= 1  
 c.itemconfigure(lives\_text, text="Lives: "+ str(lives\_remaining))  
  
def check\_catch():  
 (catcherx, catchery, catcherx2, catchery2) = c.coords(catcher)  
 for ball in balls:  
 (ballx, bally, ballx2, bally2) = c.coords(ball)  
 if catcherx < ballx and ballx2 < catcherx2 and catchery2 - bally2 < 40:  
 balls.remove(ball)  
 c.delete(ball)  
 increase\_score(ball\_score)  
 root.after(100, check\_catch)  
  
def increase\_score(points):  
 effect = pygame.mixer.Sound('explosion.wav')  
 effect.play()  
 global score, ball\_speed, ball\_interval  
 score += points  
 ball\_speed = int(ball\_speed \* difficulty)  
 ball\_interval = int(ball\_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\_ball)  
root.after(1000, move\_balls)  
root.after(1000, check\_catch)  
root.mainloop()

Hard.py::

from itertools import cycle  
from random import randrange  
from tkinter import Canvas, Tk, messagebox, font,Button  
  
import pygame  
  
pygame.mixer.init()  
def windowdestroy():  
 messagebox.showinfo("Game Over!", "Final Score: " + str(score))  
 root.destroy()  
  
  
canvas\_width = 800  
canvas\_height = 600  
  
root = Tk()  
c = Canvas(root, width=canvas\_width, height=canvas\_height, background="black")  
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='black', width=0)  
c.pack()  
c.create\_text(  
 canvas\_width/2,20,  
 fill="darkblue",  
 font="Times 20 italic bold",  
 text="Catch Ball")  
btn = Button(root, text='Exit', width=5,height=2, bd='5', command=windowdestroy)  
btn.place(x=0, y=0)  
  
  
color\_cycle = cycle(["light blue", "light green", "light pink", "light yellow", "light cyan"])  
ball\_width = 25  
ball\_height = 25  
ball\_score = 10  
ball\_speed = 100  
ball\_interval = 1400  
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="chord", outline=catcher\_color, width=9)  
game\_font = font.nametofont("TkFixedFont")  
game\_font.config(size=18)  
  
  
score = 0  
score\_text = c.create\_text(670, 10, anchor="nw", font=game\_font, fill="white", text="Score: "+ str(score))  
  
lives\_remaining = 3  
lives\_text = c.create\_text(canvas\_width-10, 37, anchor="ne", font=game\_font, fill="white", text="Lives: "+ str(lives\_remaining))  
  
balls = []  
i=0  
def create\_ball():  
 global i  
 if (i == 0):  
 effects = pygame.mixer.Sound('music2.wav')  
 effects.play()  
 x = randrange(10, 740)  
 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)  
 i=i+1  
def move\_balls():  
 for ball in balls:  
 (ballx, bally, ballx2, bally2) = c.coords(ball)  
 c.move(ball, 0, 10)  
 if bally2 > canvas\_height:  
 ball\_dropped(ball)  
 root.after(ball\_speed, move\_balls)  
  
def ball\_dropped(ball):  
 balls.remove(ball)  
 c.delete(ball)  
 lose\_a\_life()  
 if lives\_remaining == 0:  
 messagebox.showinfo("Game Over!", "Final Score: "+ str(score))  
 root.destroy()  
  
def lose\_a\_life():  
 effects = pygame.mixer.Sound('laser.wav')  
 effects.play()  
 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 ball in balls:  
 (ballx, bally, ballx2, bally2) = c.coords(ball)  
 if catcherx < ballx and ballx2 < catcherx2 and catchery2 - bally2 < 40:  
 balls.remove(ball)  
 c.delete(ball)  
 increase\_score(ball\_score)  
 root.after(100, check\_catch)  
  
def increase\_score(points):  
 effect = pygame.mixer.Sound('explosion.wav')  
 effect.play()  
 global score, ball\_speed, ball\_interval  
 score += points  
 ball\_speed = int(ball\_speed \* difficulty)  
 ball\_interval = int(ball\_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\_ball)  
root.after(1000, move\_balls)  
root.after(1000, check\_catch)  
root.mainloop()

**c.Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC01 | | **Use Case ID:** UC01 | |
| **Test Case Title:** Levels | |  | |
| **Test Case Description:**  Verify whether different levels are being accessible and playable | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1. open the game and run.  2. Select start  3. Select the level user wants to play. | The user gets access to the levels if he follows the procedure mentioned and gets an GUI interface of game. | | The user sees the levels and has to select the level |

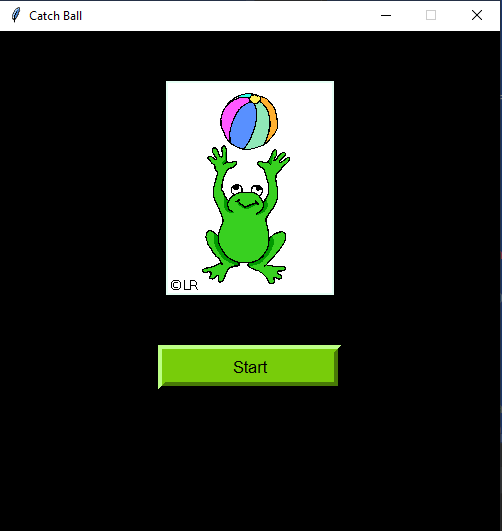
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC02 | | **Use Case ID:** UC02 | |
| **Test Case Title:** score | |  | |
| **Test Case Description:**  Verify whether the score gets increased when user catches the ball | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1. Open the game using any editor and run. | The user should see the increase in score when user catches the ball. | | The user successfully observe the increase in score |

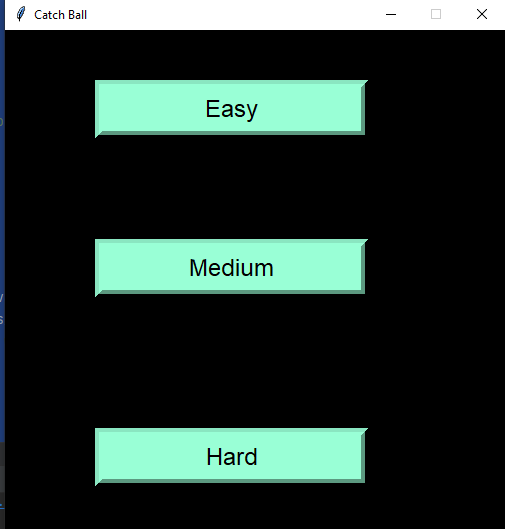
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC03 | | **Use Case ID:** UC03 | |
| **Test Case Title:** lives remaining | |  | |
| **Test Case Description:**  Verify whether lives are getting reduced when user miss the ball | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1. Open the game using any editor and run.  2.Start playing the game | The system displays the lives and lives should reduce when user miss the ball | | The lives will decrease when ball is not catched. |

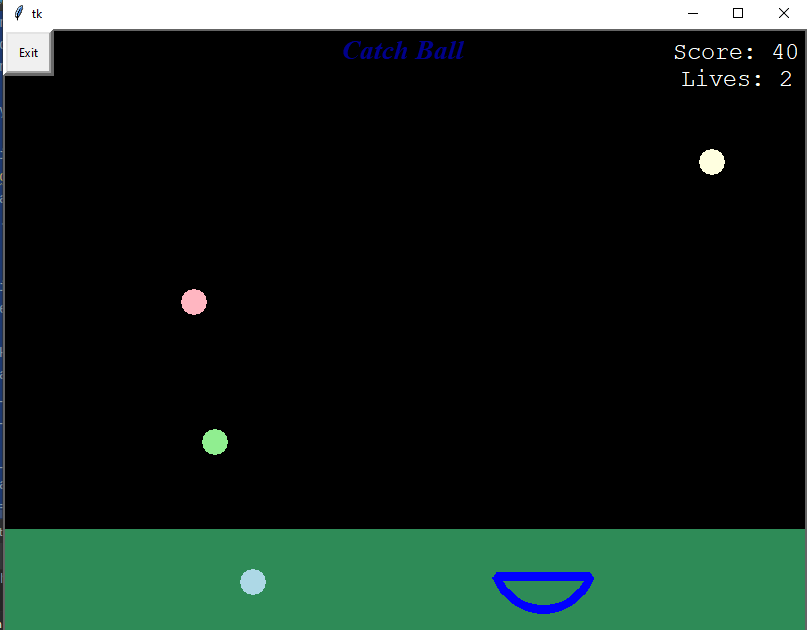
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC04 | | **Use Case ID:** UC04 | |
| **Test Case Title:** Exit | |  | |
| **Test Case Description:**  Verify whether game is being terminated or not when user opt for exit | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1.open the game using any editor and run.  2.Select exit which is at the top left end | The game terminates. | | The system successfully terminated the game |

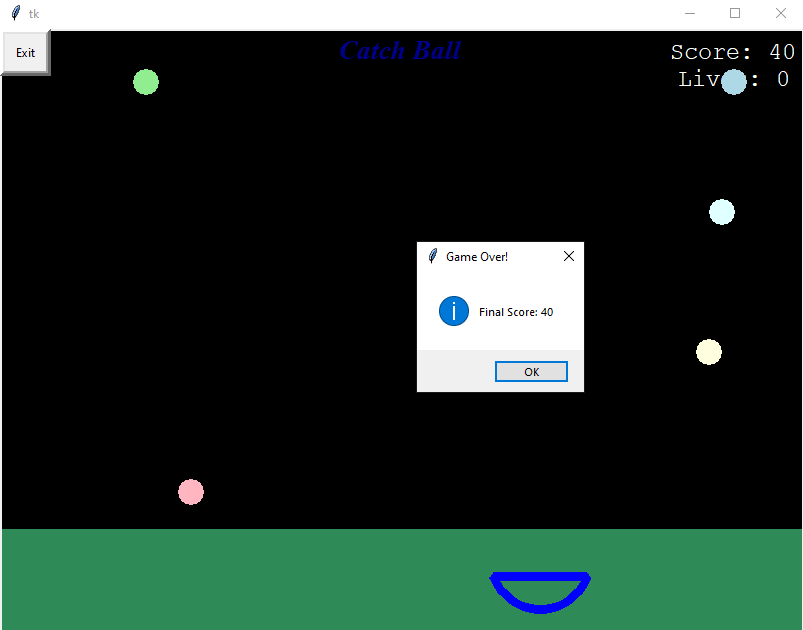
5)Github link: https://github.com/Bharathreddy11/Catch-Ball

**6)RESULTS**









**7. OUTCOMES FROM THE PROJECT APART FROM CURRICULUM**

● We have got to know how to use different kinds of modules.

● We have discovered different kinds of libraries and modules and got to know their implementation and working.

● We have got to know about Tkinter, pygame, random, itertools.

● We have learnt how to build GUI using Tkinter framework.

● We have got to know how to handle events in GUI and about the components in them.

**8 )CONCLUSION**

To conclude, this game lets children have fun. We wish to carry this spirit and work on many more projects in future.

**9. REFERENCES:**

[**https://docs.python.org/3/library/itertools.html**](https://docs.python.org/3/library/itertools.html)

[**https://docs.python.org/3/library/tk.html**](https://docs.python.org/3/library/tk.html)

[**https://www.pygame.org/docs/ref/music.html**](https://www.pygame.org/docs/ref/music.html)

[**https://www.youtube.com/watch?v=djDcVWbEYoE&t=191s**](https://www.youtube.com/watch?v=djDcVWbEYoE&t=191s)