**CHAPTER 1**

**INTRODUCTION**

**CHAPTER 1**

**1.1 Introduction to Python**

Python is an object-oriented high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development as well as for use as a scripting or glue language to connect existing components together. Python’s simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python Interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and cam be freely distributed

**1.2 Introduction to Tkinter**

The Tkinter package the standard Python interface to the TCL/TK GUI toolkit. Both Tk and Tkinter are available on most Unix platforms, including macOS, as well as on Windows systems. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

**1.3 Introduction to Handcricket**

Handcricket is a simple game where two players take turns batting and bowling against each other. Here, instead of a second player, we play against a computer. The computer randomly generates choices and compare it with the player choices. The score is tallied as the game goes on and the player with the highest score wins

**1.4 Scope and importance of work**

The scope of the project is to create an entertaining game that allows users to play Handcricket using their computer.

**CHAPTER 2**

**IMPLEMENTATION**

**CHAPTER 2**

**2.1 Modules**

1. **Tkinter:**

Tkinter module is used to create a GUI for the program, allowing user to press buttons to give input.

1. **Random:**

The random module is used to generate random numbers used in the toss and during batting and bowling

**2.2 Functions**

1. **Six():**

This function generates a number between 1 and 6 during batting and bowling for the computer

1. **Two():**

This function generates a number between 1 and 2 during the toss

1. **Bat():**

This function manages all the parts of batting for the player

1. **Bowl():**

This function manages all the parts of bowling for the player

1. **Game():**

This function controls the game, and decides what part of the game will be played next

1. **TossMenu():**

This function is used during the toss, and checks if the player has won or lost the toss

**2.3 SOURCE CODE**

from tkinter import \*

import random

#Random Number

def six():

return random.randrange(1,7)

def two():

return random.randrange(0,2)

#Creating window for the game:

window = Tk()

window.title("Handcricket")

window.geometry("640x360")

window.resizable(width=False, height=False)

window.iconbitmap('E:\Github\handcricket\icon.ico')

bgframe = Frame(window)

bgframe.pack(side="top", expand=1)

bgframe.place(x= 0, y= 0)

bgimage = PhotoImage(file = 'E:\Github\handcricket\wallpaper.png')

bgwidget = Label(bgframe, image = bgimage)

bgwidget.pack()

buttonimage = PhotoImage(file = 'E:\Github\handcricket\\button.png')

#Starting Heading

Header = "MENU"

HeaderWidget = Label(window, text = "Handcricket",height = 1, width = 30, bg = 'gray', highlightthickness=2,highlightcolor= 'black')

HeaderWidget.pack(pady=10)

#Heading Change

def Heading(Header):

HeaderWidget.config(text=Header)

#The Game

def Game():

Play.destroy()

Exit.destroy()

result = TossMenu()

#Player won toss or not

if result == 1:

Match(BatBowl())

else:

Match(two())

#Batting

def Bat():

score = 0

rand = 0

choice = IntVar()

Heading("Batting")

ScoreWidget = Label(window, text = "Your Score: "+str(score), height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

PlayerWidget = Label(window, text = "Your Choice: "+str(choice.get()), height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

OpponentWidget = Label(window, text = "Opponent's Choice: "+str(rand), height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

ScoreWidget.pack()

PlayerWidget.pack()

OpponentWidget.pack()

c1 = Button(window, text='1',command=lambda: choice.set(1), height = 20, width = 190, image = buttonimage, compound=CENTER)

c2 = Button(window, text='2',command=lambda: choice.set(2), height = 20, width = 190, image = buttonimage, compound=CENTER)

c3 = Button(window, text='3',command=lambda: choice.set(3), height = 20, width = 190, image = buttonimage, compound=CENTER)

c4 = Button(window, text='4',command=lambda: choice.set(4), height = 20, width = 190, image = buttonimage, compound=CENTER)

c5 = Button(window, text='5',command=lambda: choice.set(5), height = 20, width = 190, image = buttonimage, compound=CENTER)

c6 = Button(window, text='6',command=lambda: choice.set(6), height = 20, width = 190, image = buttonimage, compound=CENTER)

c1.pack()

c2.pack()

c3.pack()

c4.pack()

c5.pack()

c6.pack()

while True:

ScoreWidget.config(text = "Your Score: "+str(score))

window.wait\_variable(choice)

rand = six()

if choice.get() == rand:

break

else:

score += choice.get()

PlayerWidget.config(text = "Your Choice: "+str(choice.get()))

OpponentWidget.config(text = "Opponent's Choice: "+str(rand))

ResultWidget = Label(window, text = "You are out", height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

ResultWidget.pack(pady=1)

ContinueButton = Button(window, text='Continue',command=lambda: choice.set(0), height = 20, width = 190, image = buttonimage, compound=CENTER)

ContinueButton.pack(pady=1)

window.wait\_variable(choice)

c1.destroy()

c2.destroy()

c3.destroy()

c4.destroy()

c5.destroy()

c6.destroy()

PlayerWidget.destroy()

OpponentWidget.destroy()

ResultWidget.destroy()

ContinueButton.destroy()

return score

#Bowling

def Bowl():

score = 0

rand = 0

choice = IntVar()

Heading("Bowling")

ScoreWidget = Label(window, text = "Computer's Score: "+str(score), height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

PlayerWidget = Label(window, text = "Your Choice: "+str(choice.get()), height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

OpponentWidget = Label(window, text = "Opponent's Choice: "+str(rand), height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

ScoreWidget.pack()

PlayerWidget.pack()

OpponentWidget.pack()

c1 = Button(window, text='1',command=lambda: choice.set(1), height = 20, width = 190, image = buttonimage, compound=CENTER)

c2 = Button(window, text='2',command=lambda: choice.set(2), height = 20, width = 190, image = buttonimage, compound=CENTER)

c3 = Button(window, text='3',command=lambda: choice.set(3), height = 20, width = 190, image = buttonimage, compound=CENTER)

c4 = Button(window, text='4',command=lambda: choice.set(4), height = 20, width = 190, image = buttonimage, compound=CENTER)

c5 = Button(window, text='5',command=lambda: choice.set(5), height = 20, width = 190, image = buttonimage, compound=CENTER)

c6 = Button(window, text='6',command=lambda: choice.set(6), height = 20, width = 190, image = buttonimage, compound=CENTER)

c1.pack()

c2.pack()

c3.pack()

c4.pack()

c5.pack()

c6.pack()

while True:

ScoreWidget.config(text = "Computer's Score: "+str(score))

window.wait\_variable(choice)

rand = six()

if choice.get() == rand:

break

else:

score += rand

PlayerWidget.config(text = "Your Choice: "+str(choice.get()))

OpponentWidget.config(text = "Opponent's Choice: "+str(rand))

ResultWidget = Label(window, text = "Computer is out", height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

ResultWidget.pack(pady=1)

ContinueButton = Button(window, text='Continue',command=lambda: choice.set(0), height = 20, width = 190, image = buttonimage, compound=CENTER)

ContinueButton.pack(pady=1)

window.wait\_variable(choice)

c1.destroy()

c2.destroy()

c3.destroy()

c4.destroy()

c5.destroy()

c6.destroy()

PlayerWidget.destroy()

OpponentWidget.destroy()

ResultWidget.destroy()

ContinueButton.destroy()

return score

def Match(myplay):

playerscore = 0

opponentscore = 0

if myplay == 0:

playerscore = Bat()

opponentscore = Bowl()

else:

opponentscore = Bowl()

playerscore = Bat()

Heading("Game Over")

if playerscore == opponentscore:

ResultWidget = Label(window, text = "Draw!", height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

ResultWidget.pack(pady=15)

elif playerscore>opponentscore:

ResultWidget = Label(window, text = "You Win!", height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

ResultWidget.pack(pady=15)

else:

ResultWidget = Label(window, text = "Computer Wins!", height = 1, width = 20,bg = 'gray', highlightthickness=2,highlightcolor= 'black')

ResultWidget.pack(pady=15)

#Toss Menu

def TossMenu():

choice = IntVar()

Heading("Heads or Tails?")

Heads = Button(window, text='Heads',command=lambda: choice.set(0), height = 20, width = 190, image = buttonimage, compound=CENTER)

Tails = Button(window, text='Tails',command=lambda: choice.set(1), height = 20, width = 190, image = buttonimage, compound=CENTER)

Heads.pack(pady=2)

Tails.pack(pady=2)

window.wait\_variable(choice)

Heads.destroy()

Tails.destroy()

if choice.get() == two():

#Player wins the toss

return 1

else:

#Player loses the toss

return 0

#Player chooses whether to start with bowling or batting

def BatBowl():

choice = IntVar()

Heading("Do you choose to bat or bowl?")

Bat = Button(window, text='Bat',command=lambda: choice.set(0), height = 20, width = 190, image = buttonimage, compound=CENTER)

Bowl = Button(window, text='Bowl',command=lambda: choice.set(1), height = 20, width = 190, image = buttonimage, compound=CENTER)

Bat.pack(pady=2)

Bowl.pack(pady=2)

window.wait\_variable(choice)

Bat.destroy()

Bowl.destroy()

return choice.get()

#Main Menu

Play = Button(window, text='Play',command=lambda: Game(), height = 20, width = 190, image = buttonimage, compound=CENTER)

Play.pack(pady=2)

Exit = Button(window, text='Exit',command=exit, height = 20, width = 190, image = buttonimage, compound=CENTER)

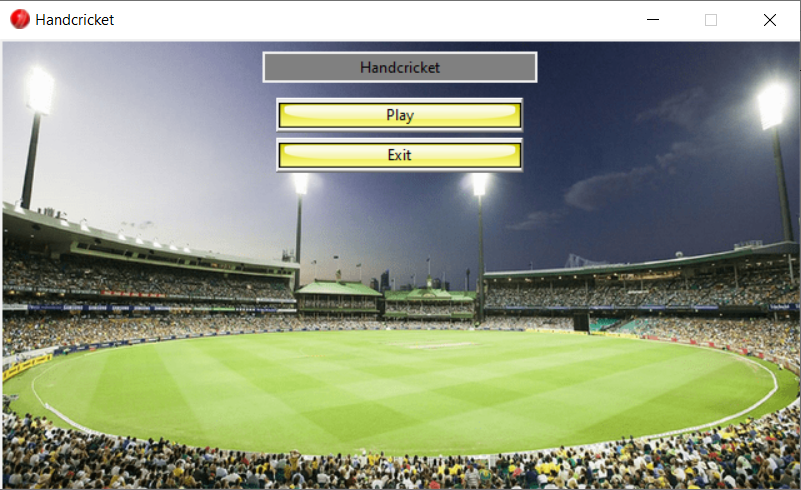
Exit.pack(pady=2)

window.mainloop()

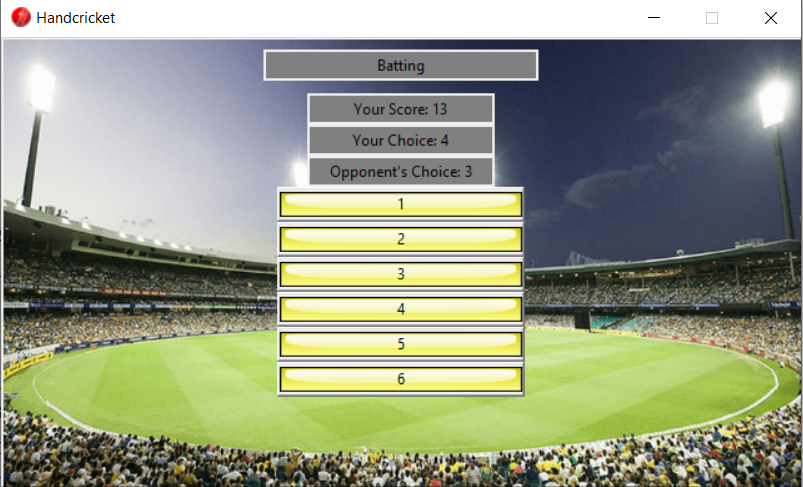
**CHAPTER 3**

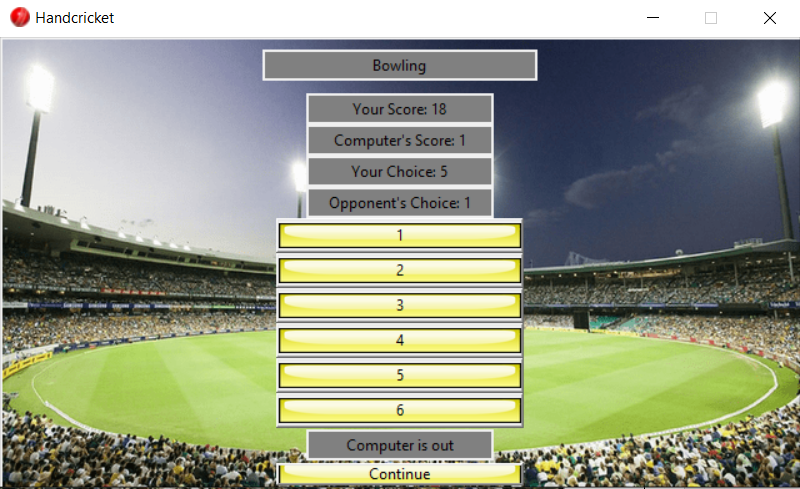
**SNAPSHOTS**

**CHAPTER 3**

****

****

****

****

**CHAPTER 4**

**CONCLUSION**

**CONCLUSION**

We have successfully implemented the Handcricket game in python and demonstrated the functionality of the Tkinter module in a neat and presentable GUI. Buttons are used to take input from the user instead of manually typing their choices. Using random function ensures that no game will be the same.

Features:

1.Interactive GUI program

2.Fun and entertaining game built with Python

3.Behind the scenes randomization and widget placement

**REFERENCES**

**[1]** <https://www.python.org/doc/>

**[2]** <https://docs.python.org/3/library/tkinter.html>

**[3]** <https://docs.python.org/3/library/random.html>

**[4]** Introduction to Python Programming by Gowrishankar S