AISSMS
INSTITUTE OF INFORMATION TECHNOLOGY
ADDING VALUE TO ENGINEERING

Department Of Computer Engineering

Data Structure And Algorithms Lab

Mini-Project

Snake And Ladder

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AISSMS IOIT

SE COMPTER ENGINEERING

SUBMITTED BY

Kaustubh S Kabra
ERP No.- 34
Teams No.-20

सत्याला मरण नाही

2020 -2021

Mini-Project-Snake And Ladder

- ***Aim:-*** *Design a mini project to implement Snake and Ladders Game using python.*
- ***Objective:-***
    1) *To Understand the use of tkinter and pygame.*
    2) *To implement a program for Sanke and Ladder game in Python.*
- ***Theory:-***

Python GUI – tkinter

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:

1. *Importing the module – tkinter*
2. *Create the main window (container)*
3. *Add any number of widgets to the main window*
4. *Apply the event Trigger on the widgets.*

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'.

```
import tkinter
```

There are two main methods used which the user needs to remember while creating the Python application with GUI.

1. ***Tk(screenName=None, baseName=None, className='Tk', useTk=1):*** *To create a main window, tkinter offers a method 'Tk(screenName=None, baseName=None, className='Tk', useTk=1)'. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is: m=tkinter.Tk() where m is the name of the main window object*
2. ***mainloop():*** *There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.*

Snake and Ladder Problem

Given a snake and ladder board, find the minimum number of dice throws required to reach the destination or last cell from source or 1st cell. Basically, the player has total control over outcome of dice throw and wants to find out minimum number of throws required to reach last cell.

If the player reaches a cell which is base of a ladder, the player has to climb up that ladder and if reaches a cell is mouth of the snake, has to go down to the tail of snake without a dice throw.

For example, consider the board shown, the minimum number of dice throws required to reach cell 30 from cell 1 is 3.
Following are the steps:

a) First throw two on dice to reach cell number 3 and then ladder to reach 22
b) Then throw 6 to reach 28.
c) Finally through 2 to reach 30.

There can be other solutions as well like (2, 2, 6), (2, 4, 4), (2, 3, 5).. etc.

The idea is to consider the given snake and ladder board as a directed graph with number of vertices equal to the number of cells in the board. The problem reduces to finding the shortest path in a graph. Every vertex of the graph has an edge to next six vertices if next 6 vertices do not have a snake or ladder. If any of the next six vertices has a snake or ladder, then the edge from current vertex goes to the top of the ladder or tail of the snake. Since all edges are of equal weight, we can efficiently find shortest path using Breadth First Search of the graph.

- ***Program:-***

```
import random
from tkinter import *
from diceMove import dice
import time

def _create_circle(self, x, y, r, **kwargs):
    return self.create_oval(x-r, y-r, x+r, y+r, **kwargs)
Canvas.create_circle = _create_circle

class MatchPosition():
    def find_snake_or_ladder(self, block, turn, position):
        x = 35*(turn>=3)
        y = (turn%3)*35
        if(block == 3):
           return 305+x, 150+y, 22
        elif(block == 5):
            return 545+x, 390+y, 8
        elif(block == 11):
```

```
            return 185+x, 30+y, 26
        elif(block == 20):
            return 545+x, 30+y, 29
        elif(block == 17):
           return 425+x, 510+y, 4
        elif(block == 19):
           return 665+x, 390+y, 7
        elif(block == 21):
           return 425+x, 390+y, 9
        elif(block == 27):
           return 65+x, 510+y, 1
        else:
            return position[0], position[1], block


class Display(object):
    def __init__(self,master,img):

        #Create board of snake and ladder
        canvas_width = 850
        canvas_height = 600
        self.color = ["#FFF", "#F00", "#0F0", "#00F", "#FF0", "#0FF"]
        self.canvas = Canvas(master, width = canvas_width, height =
canvas_height, bg = "brown")
        self.canvas.grid(padx=0, pady=0)
        self.canvas.create_image(360,300,anchor=CENTER, image = img)

        self.x = 65
        self.y = 510
        self.m = []
        self.num_player = "Players"
        self.player = []
        self.position = []
        self.i = 0
        self.block=[]
        self.move = 1
        self.turn = 0


        #Drop Menu
        OPTIONS = ["Players", "2", "3", "4", "5", "6"]
        variable = StringVar(master)
        variable.set(OPTIONS[0]) # default value
        w = OptionMenu(self.canvas, variable, *OPTIONS,
command=self.get_choice)
        w.pack()
        w.place(x=740, y=225)
        w.config(font=('calibri',(10)),bg='white',width=5)

        #Start Game
        self.startGame = Button(self.canvas, text="Start",
background='white', command = self.startGame, font=("Helvetica"))
        self.startGame.place(x=770, y=400)


    def startGame(self):
        if(self.num_player == "Players"):
```

```
            pass
        else:
            #Dice
            #Screen
            self.canvas.create_rectangle(810, 150, 760, 100, fill='white',
outline='black')
            self.canvas.pack(fill=BOTH, expand=1)
            #Button
            self.diceRoll = Button(self.canvas,
text="Roll",background='white',
                                   command = self.gamePlay,
font=("Helvetica"))
            self.num_player = int(self.num_player)
            self.diceRoll.place(x=770, y=165)
            self.create_peice()
            self.startGame.place(x=-30, y=-30)


    def get_choice(self, value):
        self.num_player = value


    def diceMove(self, position, turn):
        move = dice()
        #move = 1
        #Print Dice Value to screen
        dice_value = Label(self.canvas, text=str(move),
                           background='white', font=("Helvetica", 25))
        dice_value.pack()
        dice_value.place(x=775, y=105)


        self.x, self.y = position[0], position[1]
        if(move+self.block[turn] > 30):
            return [self.x, self.y]

        self.move = move
        self.block[turn] += move

        self.canvas.delete(self.player[turn])
        self.peices(move, turn)

        return [self.x, self.y]

    def peices(self, move, turn):
        #Starting value of and x and y should be 120 and 120
        #In create_circle initial value of x and y should be 100 and 550
        #To reach to the last block x should be 5*x and y should be 4*y
        #X should be added to value and Y should be subtracted
        # 5x120=600 and 4*120=480
        #m is the constant that tells which side to move i.e. left to right
or right to left
        for i in range(move,0,-1):
            self.x = self.x+120*self.m[turn]

            if(self.x>665 and turn < 3):
                self.y = self.y - 120
```

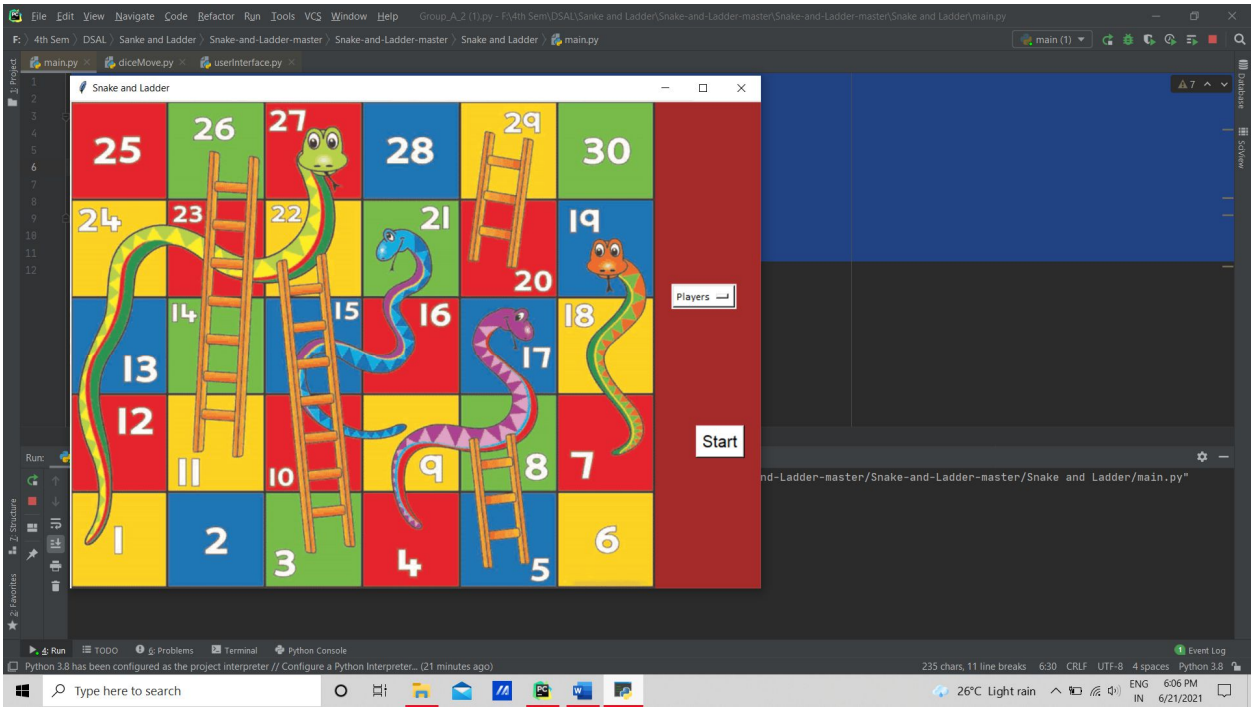
```
                self.x = 665
                self.m[turn] = -1
            elif(self.x>700 and turn >=3):
                self.y = self.y - 120
                self.x = 700
                self.m[turn] = -1
            if(self.x<65 and turn < 3):
                self.x = 65
                self.y -= 120
                self.m[turn] = 1
            elif(self.x<100 and turn >=3):
                self.x = 100
                self.y -= 120
                self.m[turn] = 1
            if(self.y<30):
                self.y=30

            # Code For the Animation of piece
            self.canvas.delete(self.player[turn])
            self.player[turn] = self.canvas.create_circle(self.x, self.y, 15,
fill=self.color[turn], outline=self.color[turn])
            self.canvas.update()
            time.sleep(0.25)


        print(self.x, self.y, self.block[turn])
        self.x, self.y, self.block[turn] =
MatchPosition().find_snake_or_ladder(self.block[turn], turn, [self.x,
self.y])

        if(any(self.y == ai for ai in [390, 425, 460, 150, 185, 220])):
            self.m[turn] = -1
        else:
            self.m[turn] = 1
        print(self.x,self.y, self.block[turn])
        self.canvas.delete(self.player[turn])
        self.player[turn] = self.canvas.create_circle(self.x, self.y, 15,
fill=self.color[turn], outline="")


    def create_peice(self):
        for i in range(int(self.num_player)):
            if(i==3):
                self.x += 35
                self.y -= 105
            self.player.append(self.canvas.create_circle(self.x, self.y, 15,
fill=self.color[i], outline=""))
            self.position.append([self.x, self.y])
            self.m.append(1)
            self.block.append(1)
            self.y += 35


    def gamePlay(self):
        if(self.move == 6):
            turn = self.turn
        else:
```

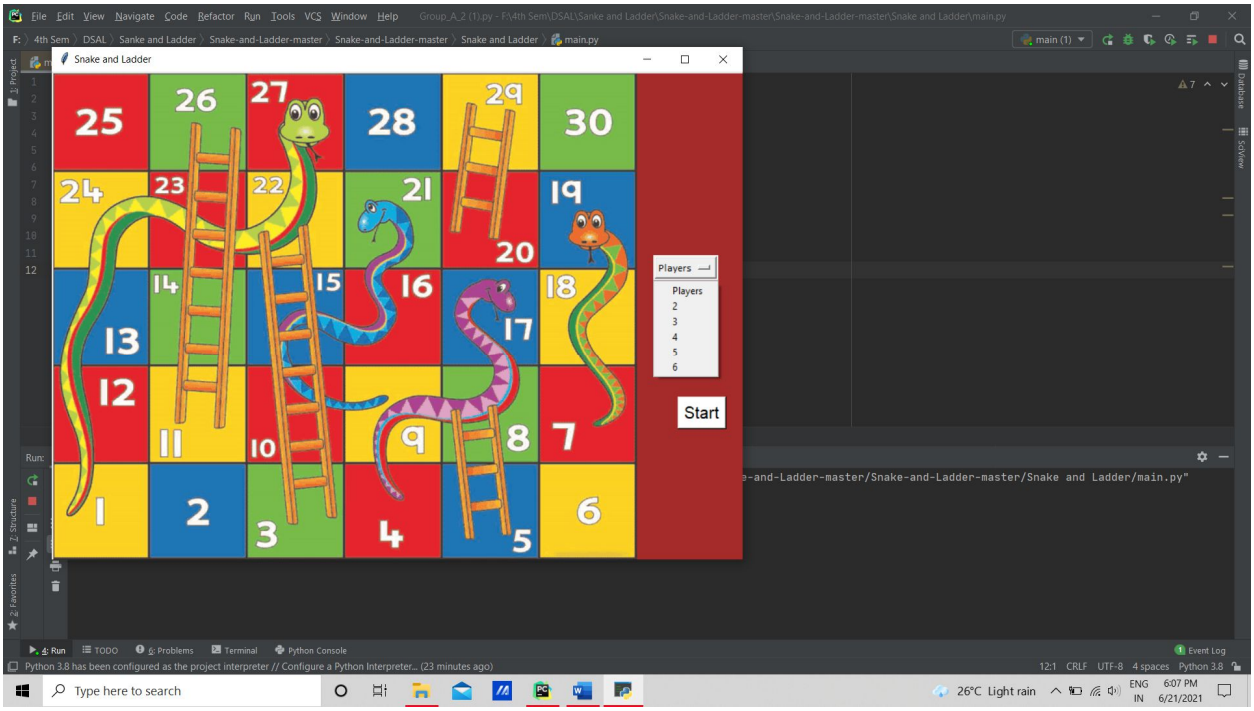
```
            turn = self.i%self.num_player
            self.i += 1
            self.turn = turn
        self.position[turn] = self.diceMove(self.position[turn], turn)
        if(self.block[self.turn] >= 30):
            self.diceRoll.place(x=-30, y=-30)
            print("Won", self.turn+1)
            top = Toplevel()
            top.title("Snake and Ladder")
            message = "Player " + str(self.turn+1) + " Won"
            msg = Message(top, text=message)
            top.geometry("%dx%d%+d%+d" % (100, 100, 250, 125))
            msg.pack()
            button = Button(top, text="Dismiss", command=top.destroy)
            button.pack()
```

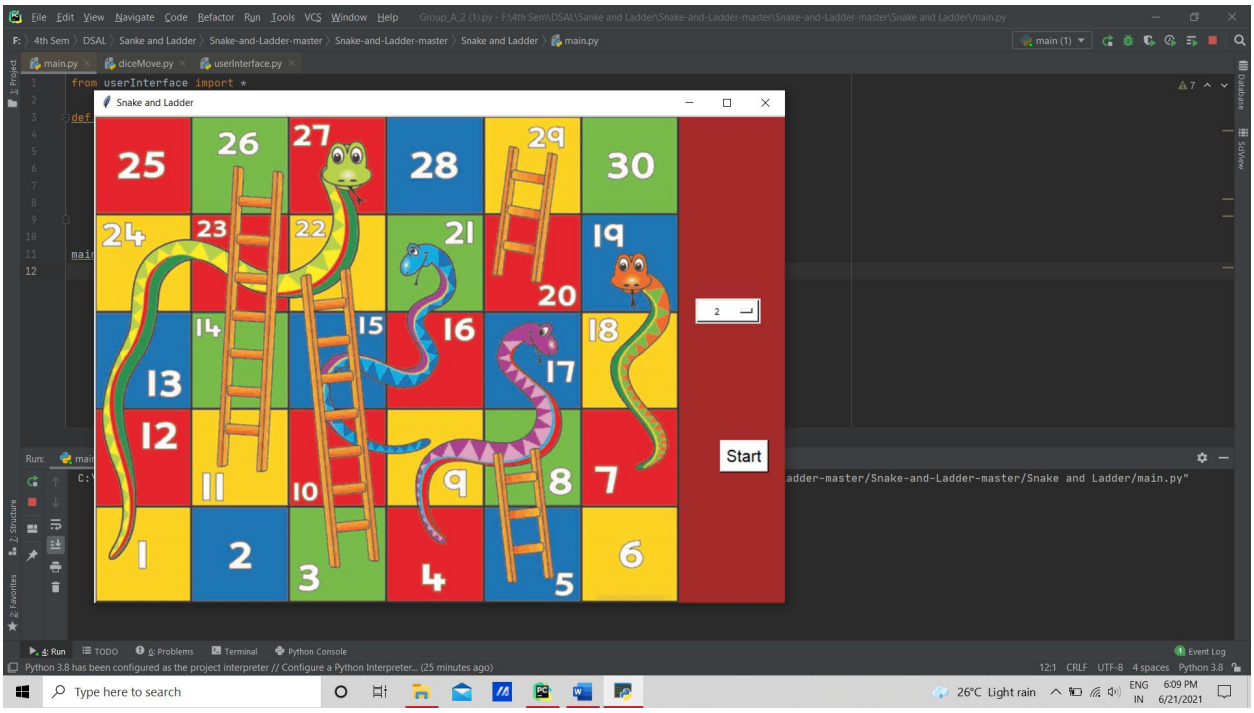
```
import random
def dice():
    move = random.randrange(1, 7, 1)
    return move
```

```
from userInterface import *

def main():
    master = Tk()
    master.title("Snake and Ladder")
    master.geometry("850x600")
    img = PhotoImage( file = "lenna.gif")
    x = Display(master,img)
    master.mainloop()

main()
```

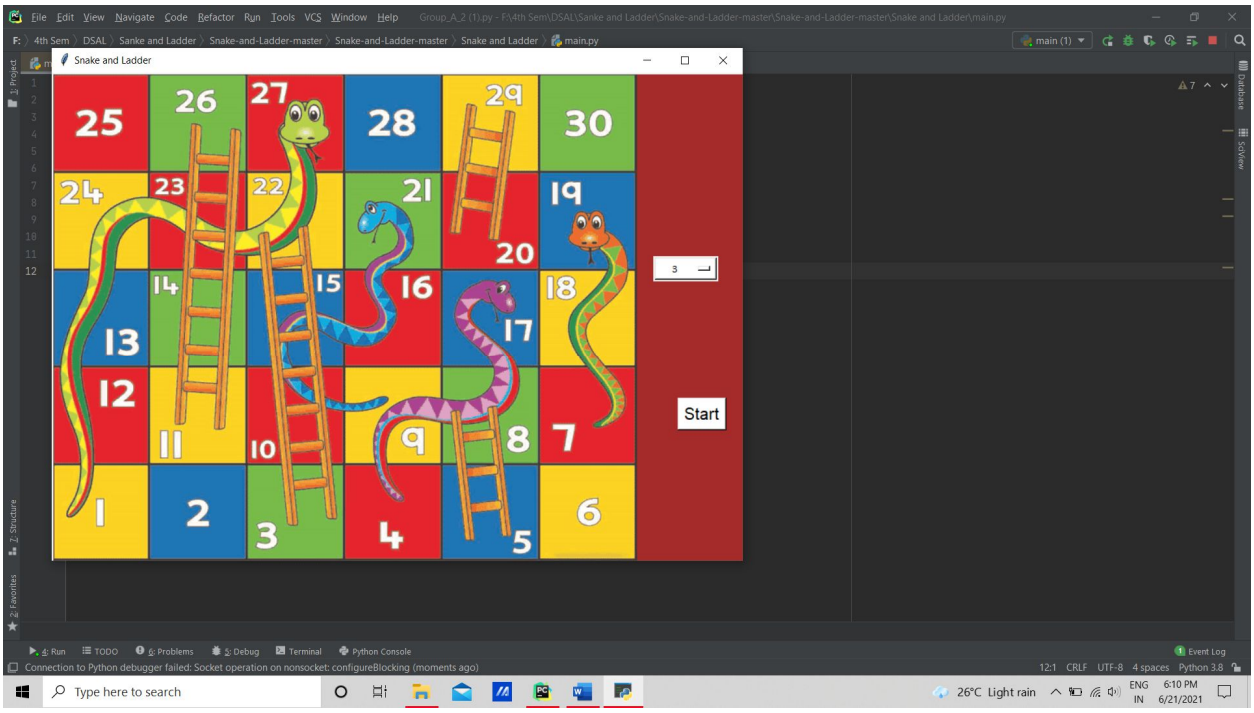
- *Output:-*


Snake and Ladder
Players
Start
Type here to search
26°C Light rain
ENG
IN
6:06 PM
6/21/2021

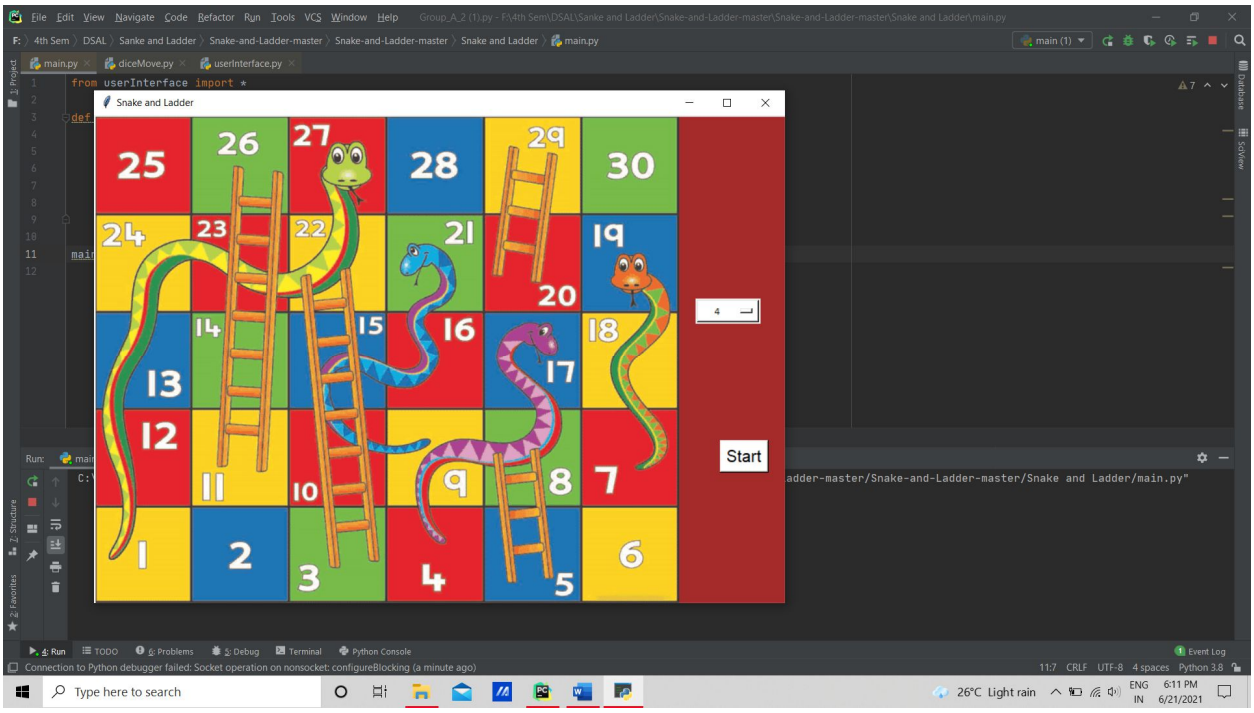

Snake and Ladder
Players
Players
2
3
4
5
6
Start
Type here to search
26°C Light rain
ENG
IN
6:07 PM
6/21/2021

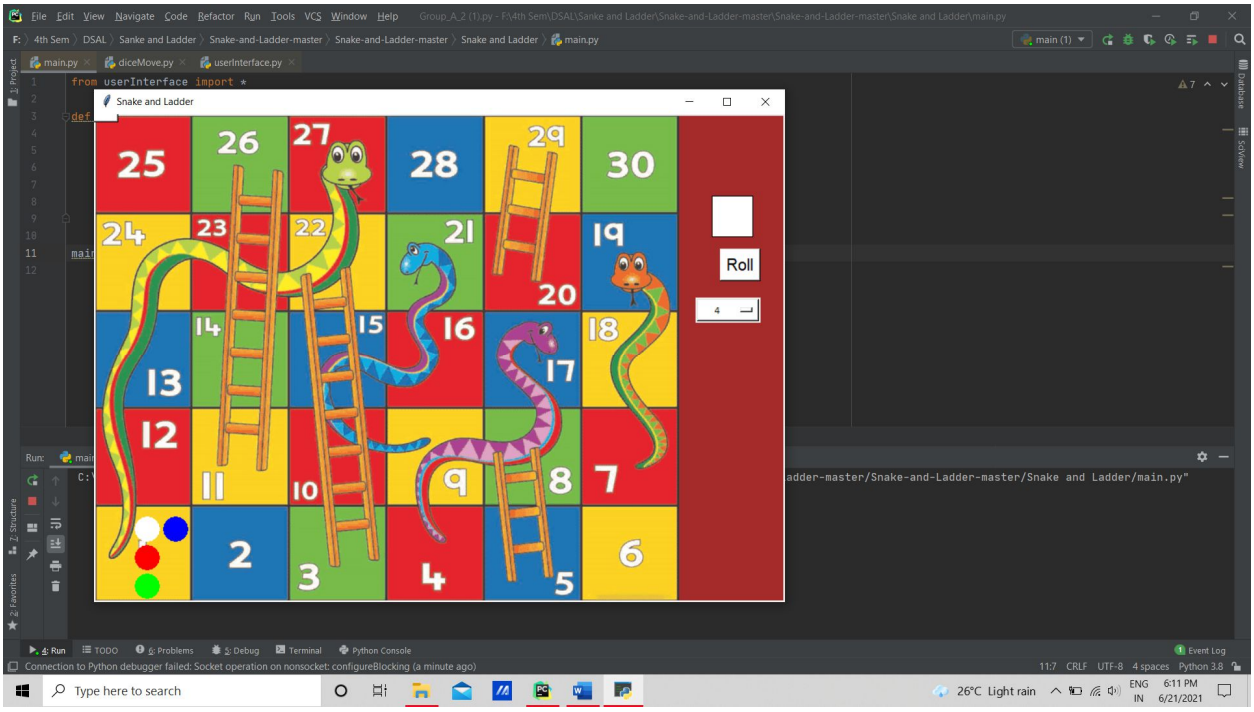

Snake and Ladder
Start
from userInterface import *
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (25 minutes ago)
Type here to search
26°C Light rain
ENG IN 6:09 PM 6/21/2021

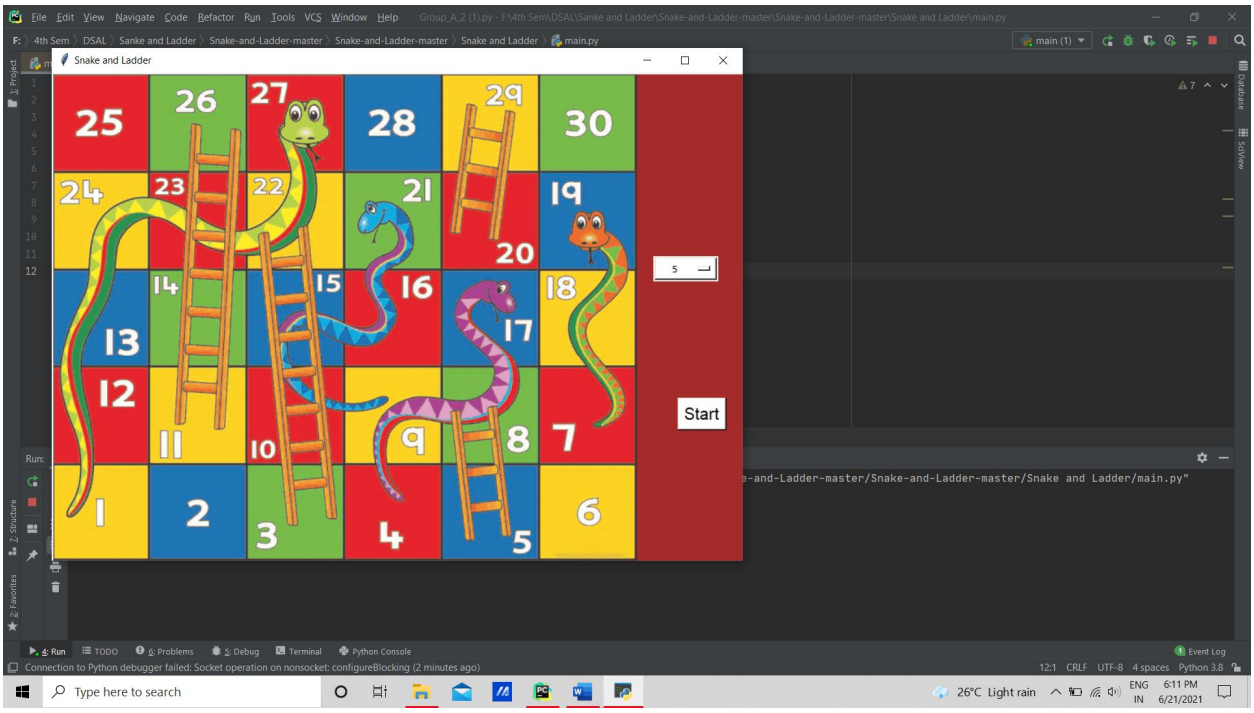

Snake and Ladder
Roll
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (23 minutes ago)
Type here to search
26°C Light rain
ENG IN 6:08 PM 6/21/2021


Snake and Ladder
Start
Connection to Python debugger failed: Socket operation on nonsocket: configureBlocking (moments ago)

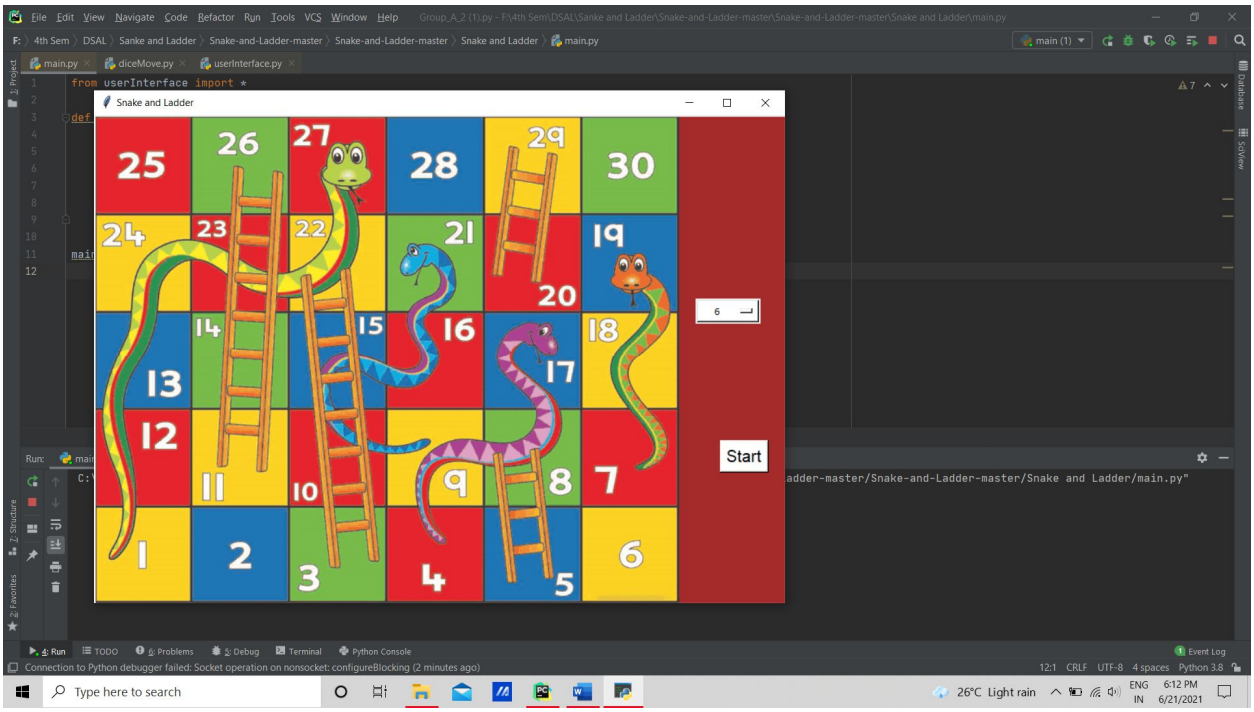

Snake and Ladder
Roll
Connection to Python debugger failed: Socket operation on nonsocket: configureBlocking (a minute ago)

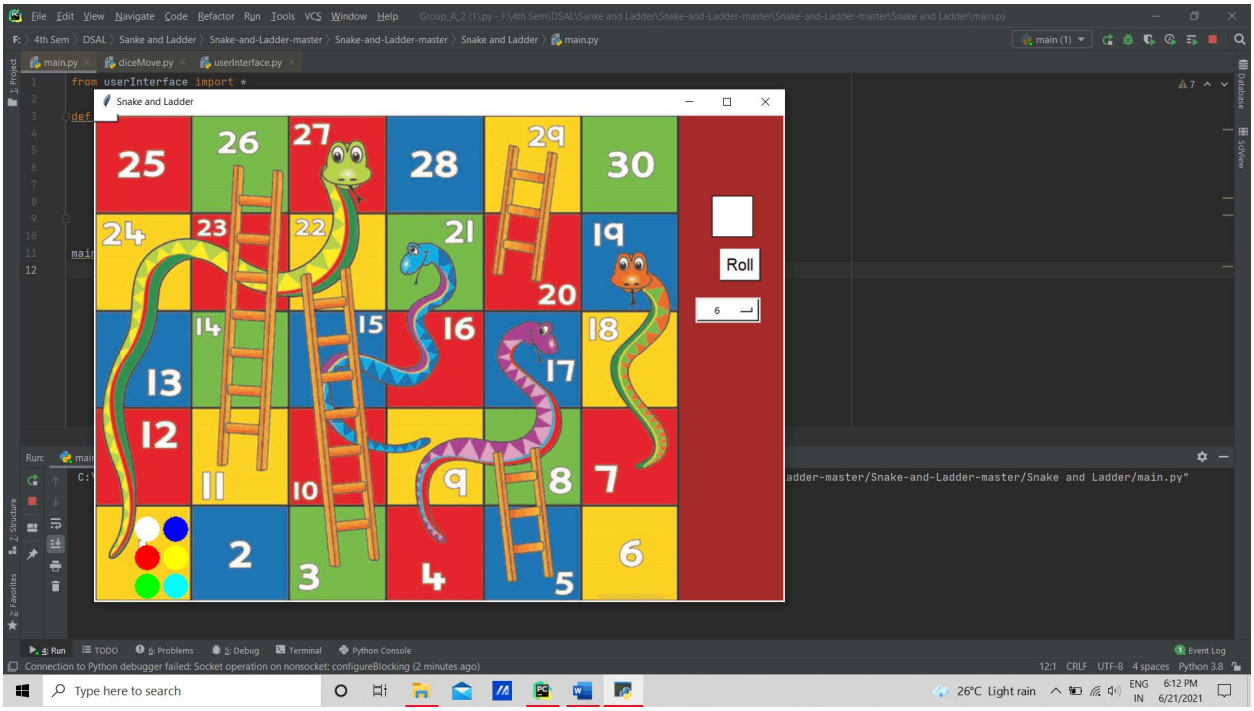

Snake and Ladder
Start
from userInterface import *
Connection to Python debugger failed: Socket operation on nonsocket: configureBlocking (a minute ago)
Type here to search
26°C Light rain
ENG IN
6:11 PM 6/21/2021

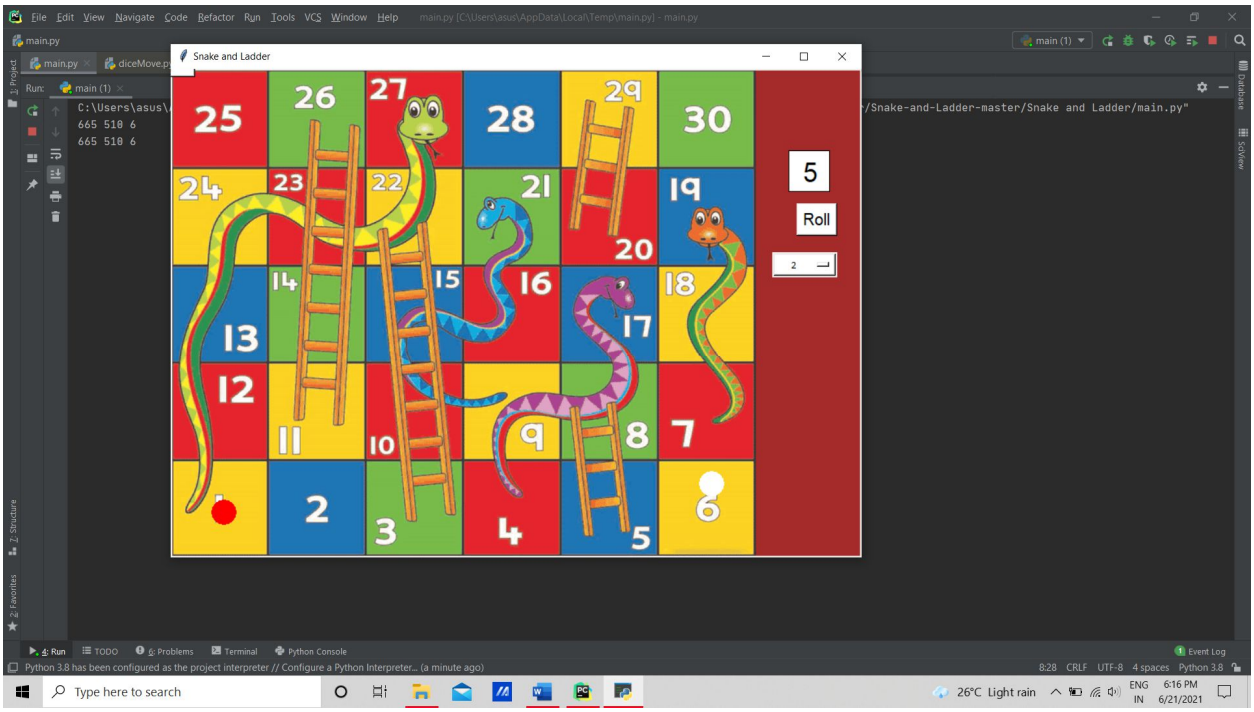

Snake and Ladder
Roll
from userInterface import *
Connection to Python debugger failed: Socket operation on nonsocket: configureBlocking (a minute ago)
Type here to search
26°C Light rain
ENG IN
6:11 PM 6/21/2021

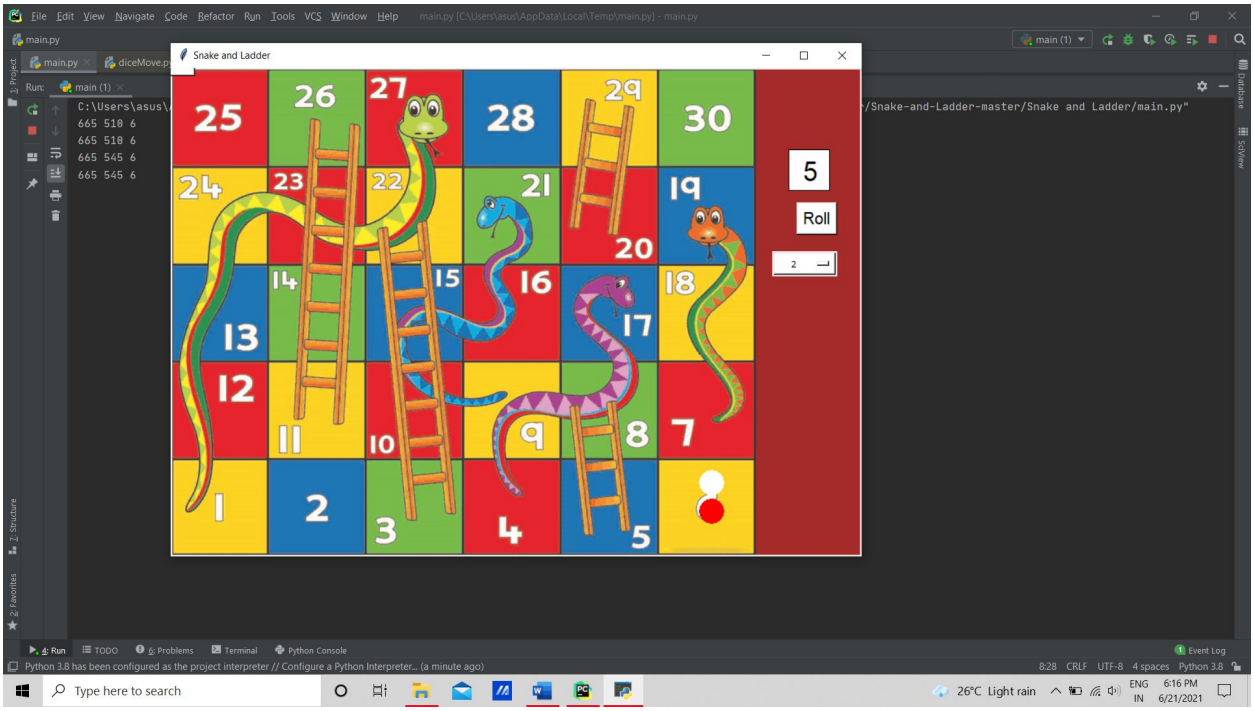

Snake and Ladder
Start
Run:
Connection to Python debugger failed: Socket operation on nonsocket: configureBlocking (2 minutes ago)
Type here to search
26°C Light rain
6:11 PM
6/21/2021

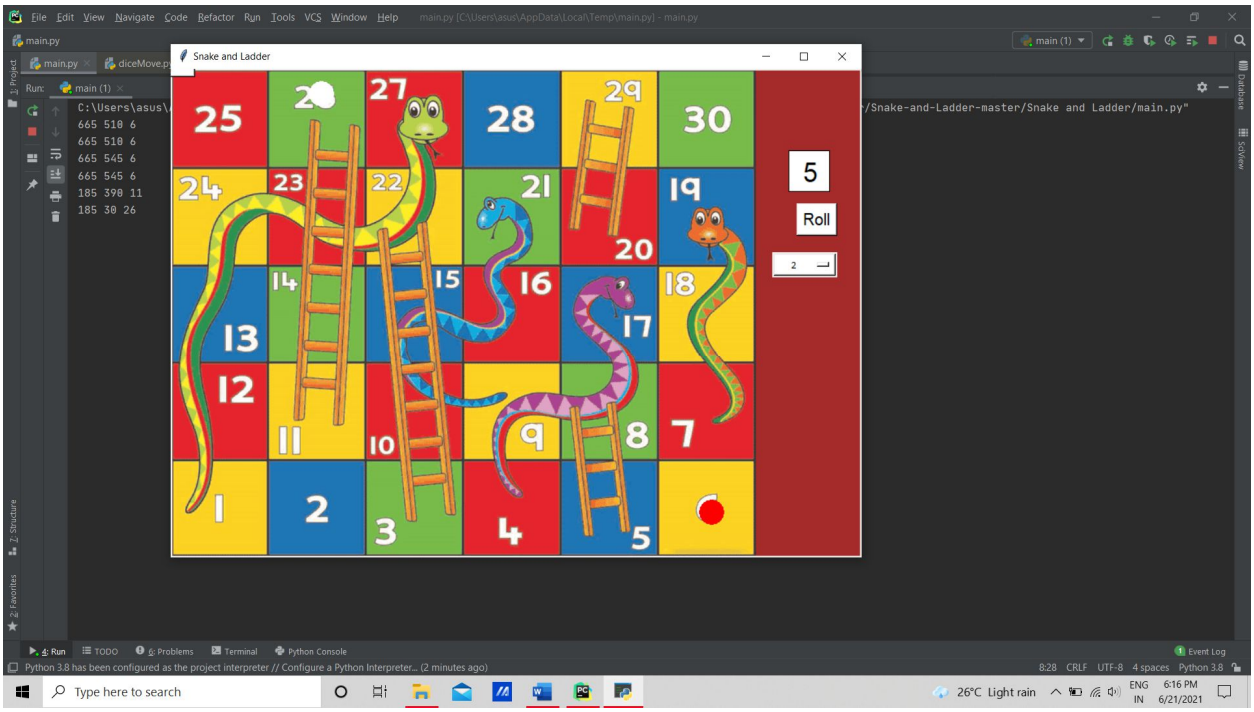

Snake and Ladder
Roll
Run:
Connection to Python debugger failed: Socket operation on nonsocket: configureBlocking (2 minutes ago)
Type here to search
26°C Light rain
6:12 PM
6/21/2021

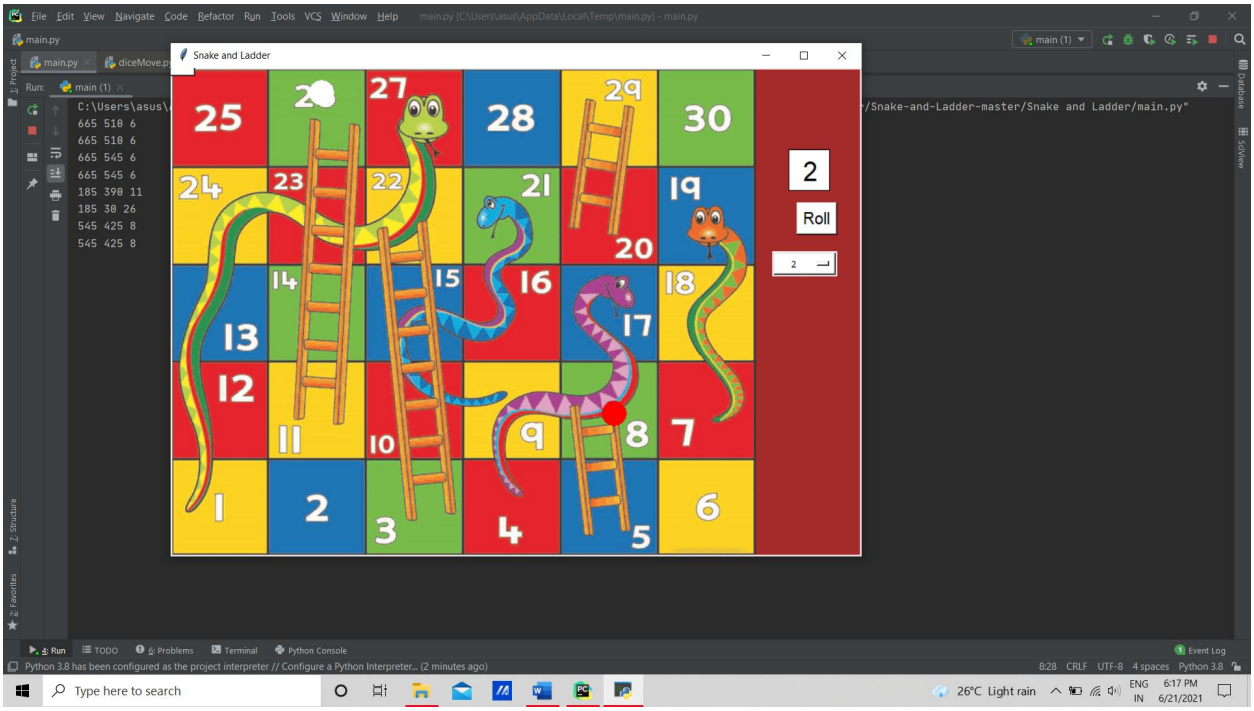

Snake and Ladder
Start
6

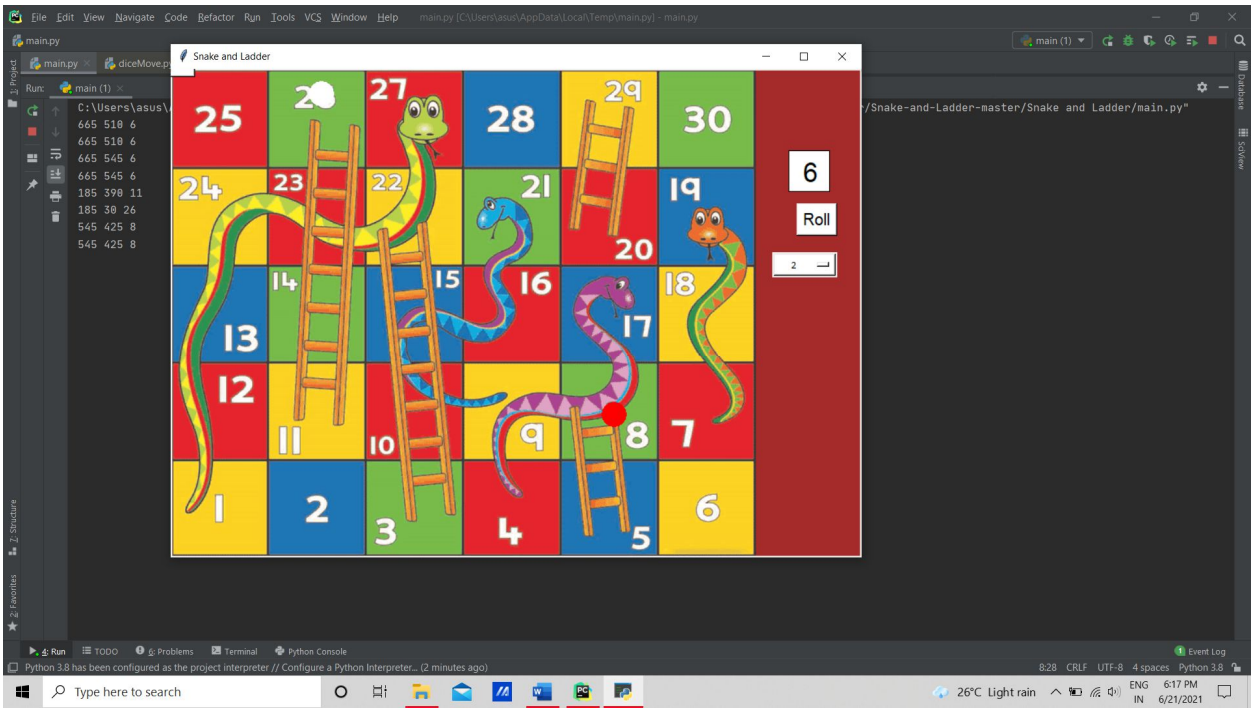

Snake and Ladder
Roll
6

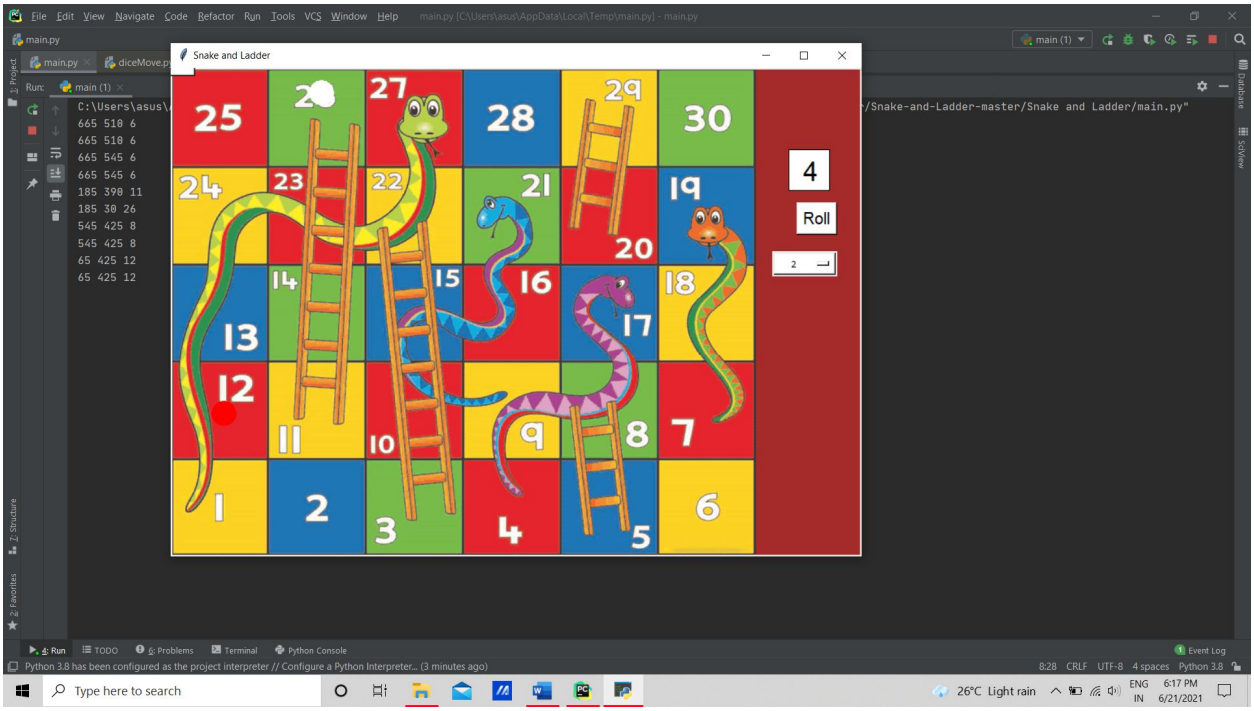

Snake and Ladder
5
Roll
2
665 518 6
665 518 6
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (a minute ago)
Type here to search
26°C Light rain
6:16 PM
6/21/2021

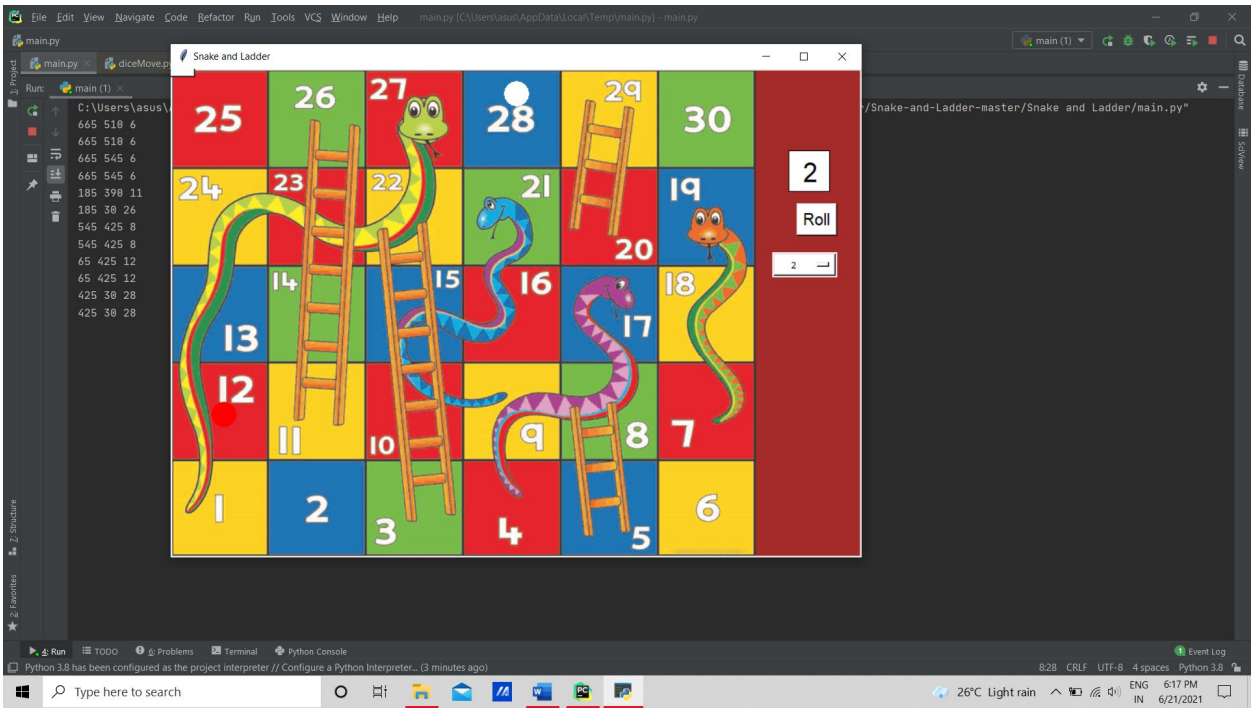

Snake and Ladder
5
Roll
2
665 518 6
665 518 6
665 545 6
665 545 6
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (a minute ago)
Type here to search
26°C Light rain
6:16 PM
6/21/2021

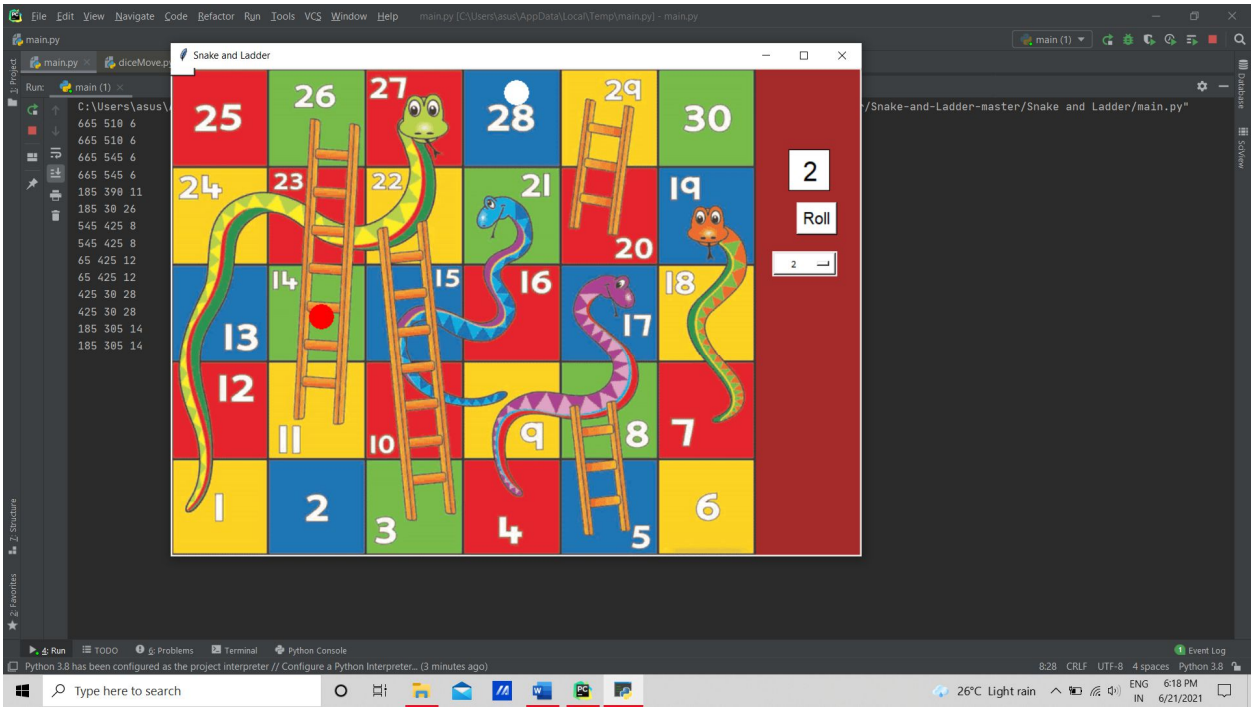

Snake and Ladder
5
Roll
2
665 518 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (2 minutes ago)
Type here to search
26°C Light rain
6:16 PM
6/21/2021

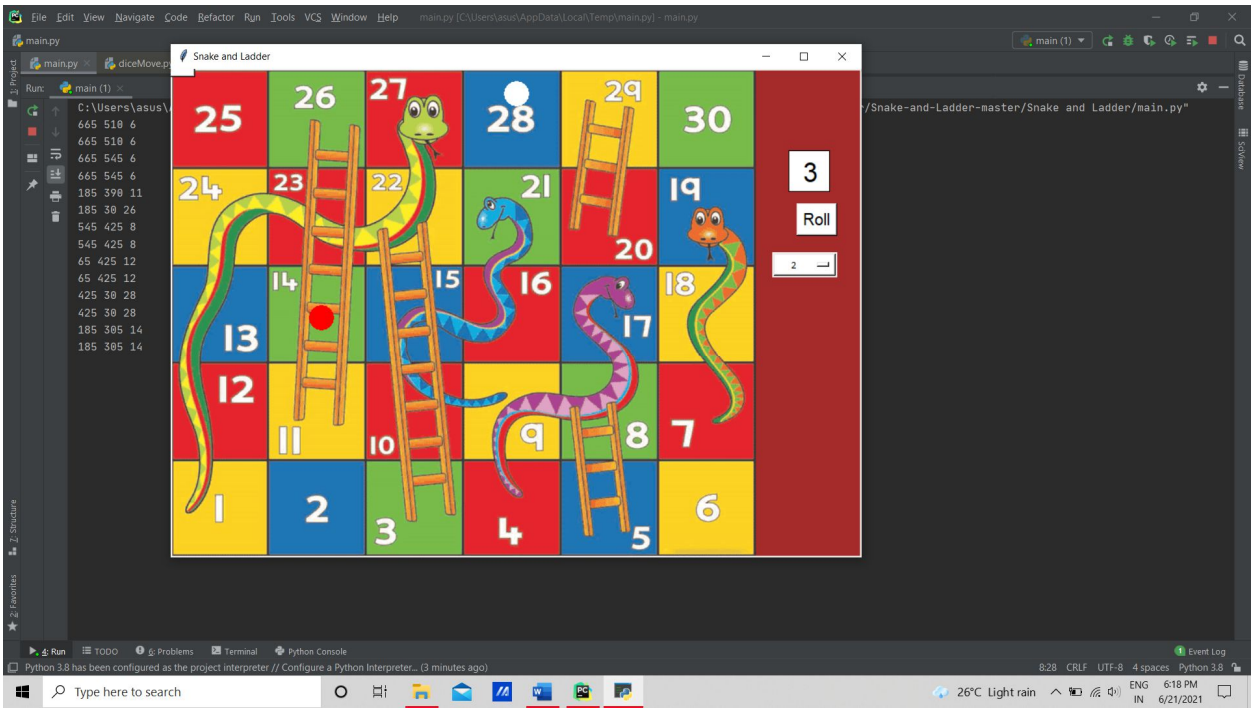

Snake and Ladder
2
Roll
2
665 518 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (2 minutes ago)
Type here to search
26°C Light rain
6:17 PM
6/21/2021

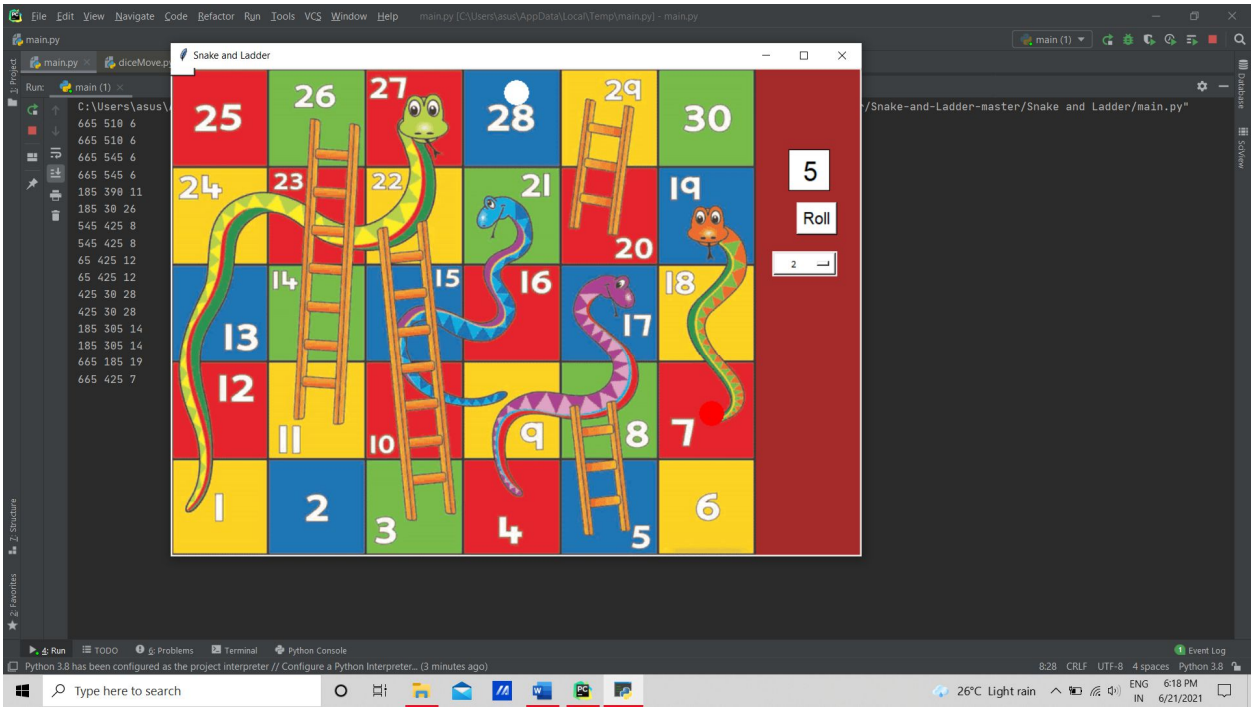

Snake and Ladder
6
Roll
2
main.py
main (1)
C:\Users\asus\
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
Run
TODO
Problems
Terminal
Python Console
Event Log
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (2 minutes ago)
Type here to search
26°C Light rain
ENG
IN
6:17 PM
6/21/2021

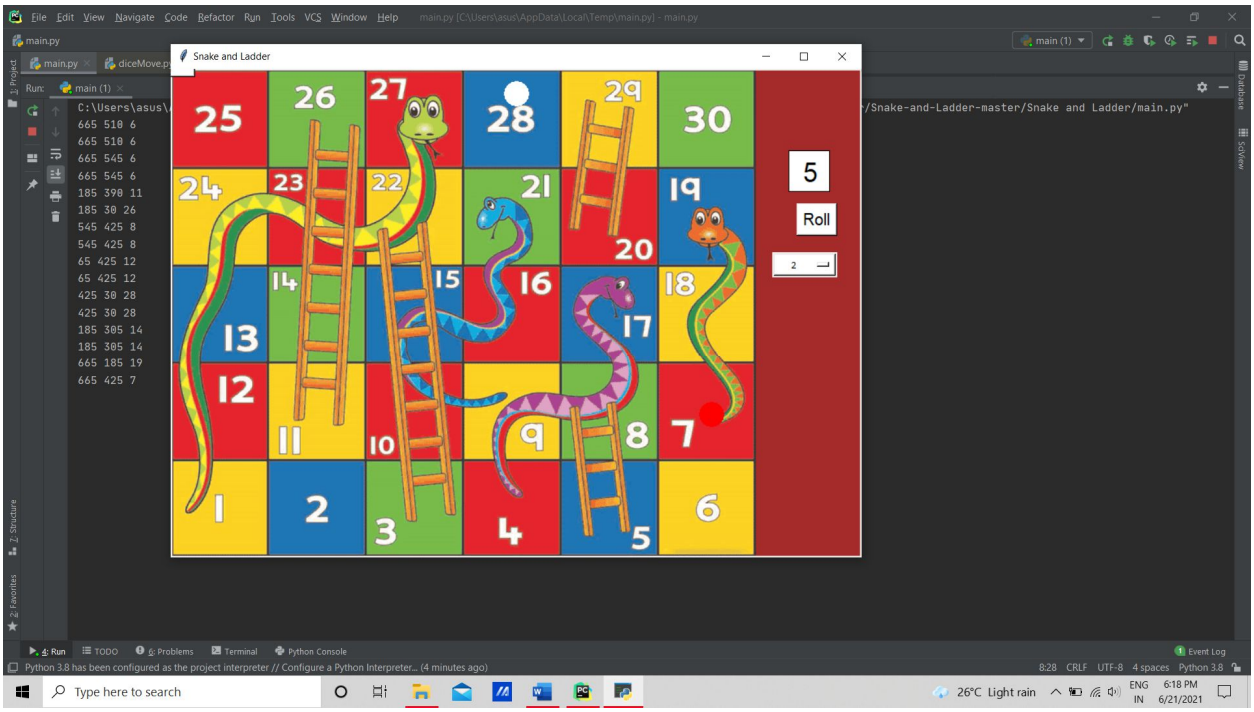

Snake and Ladder
4
Roll
2
main.py
main (1)
C:\Users\asus\
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
Run
TODO
Problems
Terminal
Python Console
Event Log
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (3 minutes ago)
Type here to search
26°C Light rain
ENG
IN
6:17 PM
6/21/2021

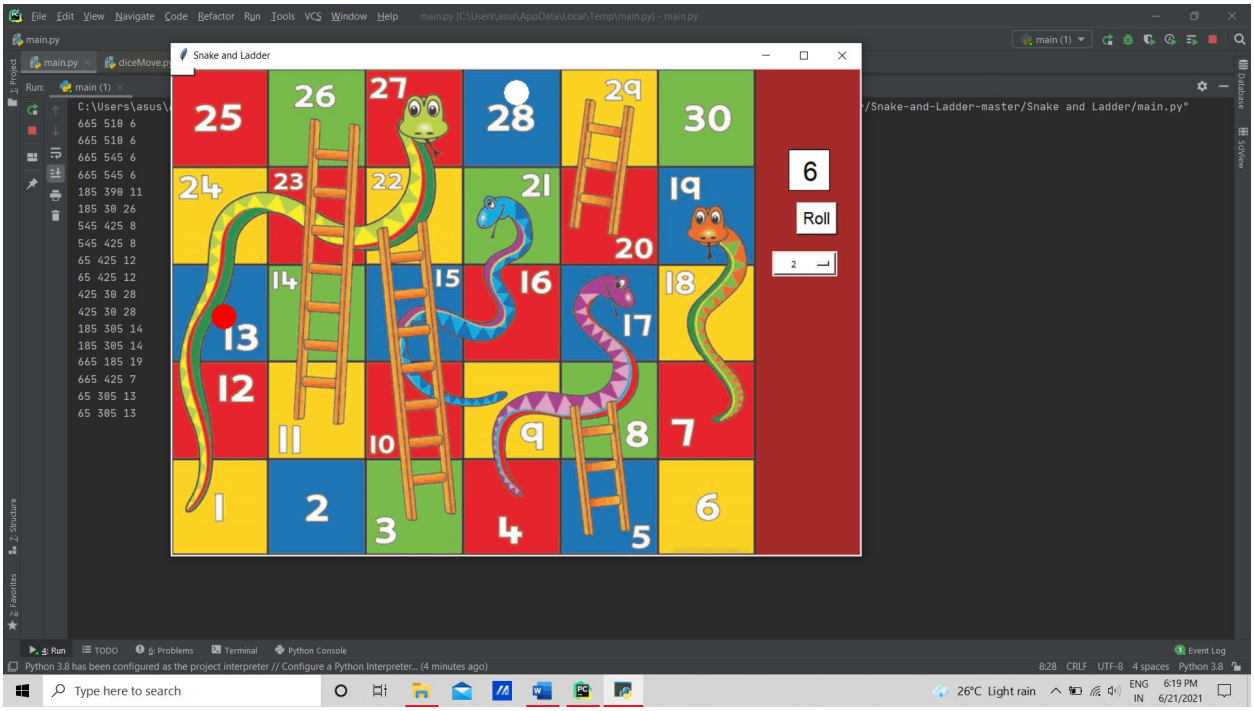

Snake and Ladder
2
Roll
2
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (3 minutes ago)
Type here to search
26°C Light rain
6:17 PM
6/21/2021

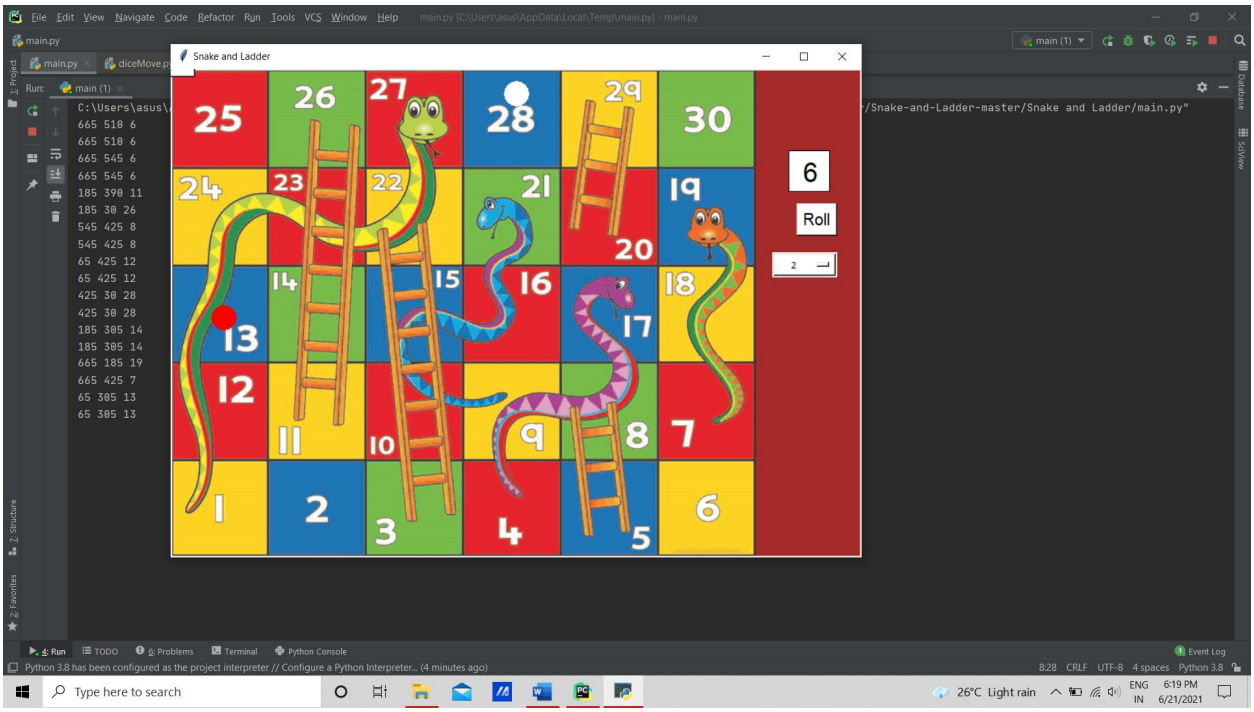

Snake and Ladder
2
Roll
2
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (3 minutes ago)
Type here to search
26°C Light rain
6:18 PM
6/21/2021

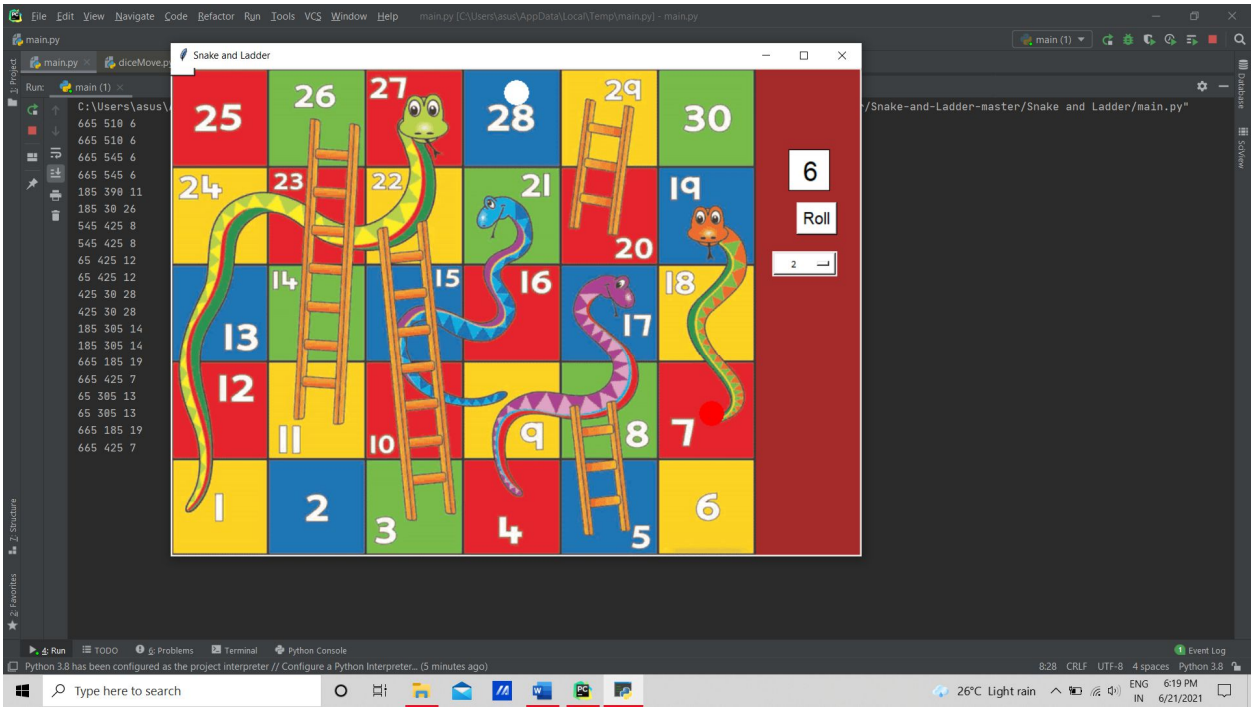

Snake and Ladder
3
Roll
2
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14

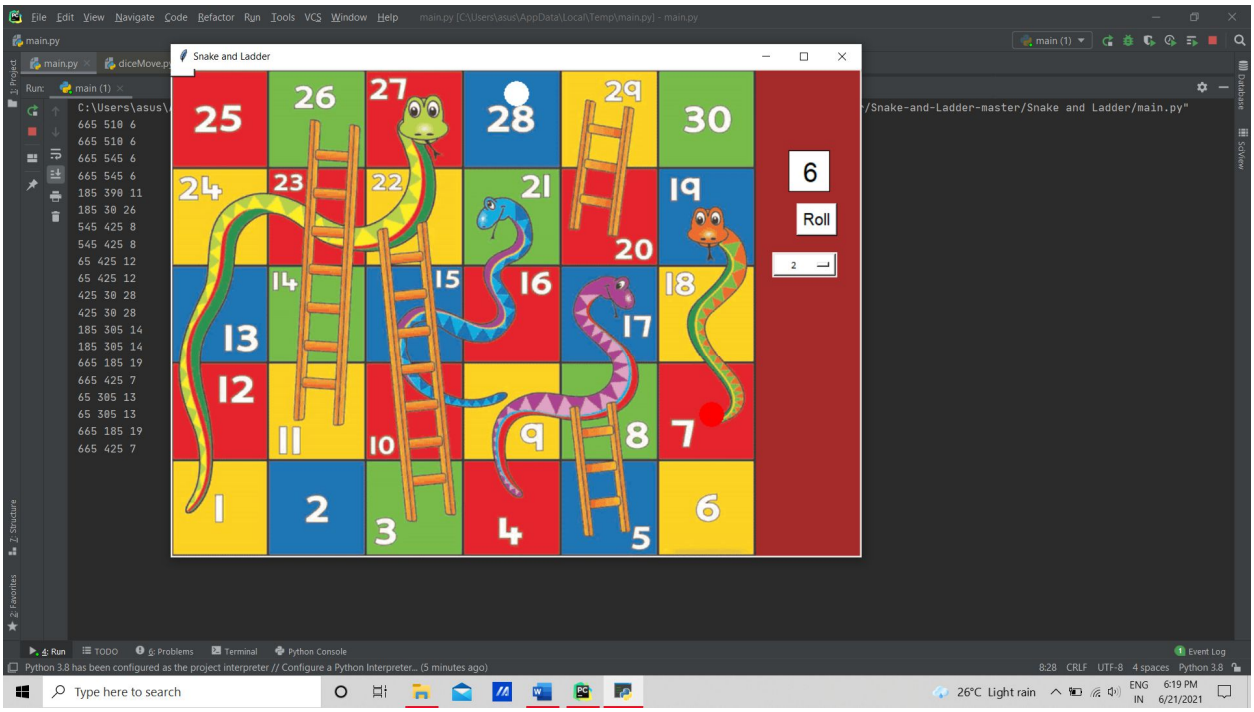

Snake and Ladder
5
Roll
2
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7

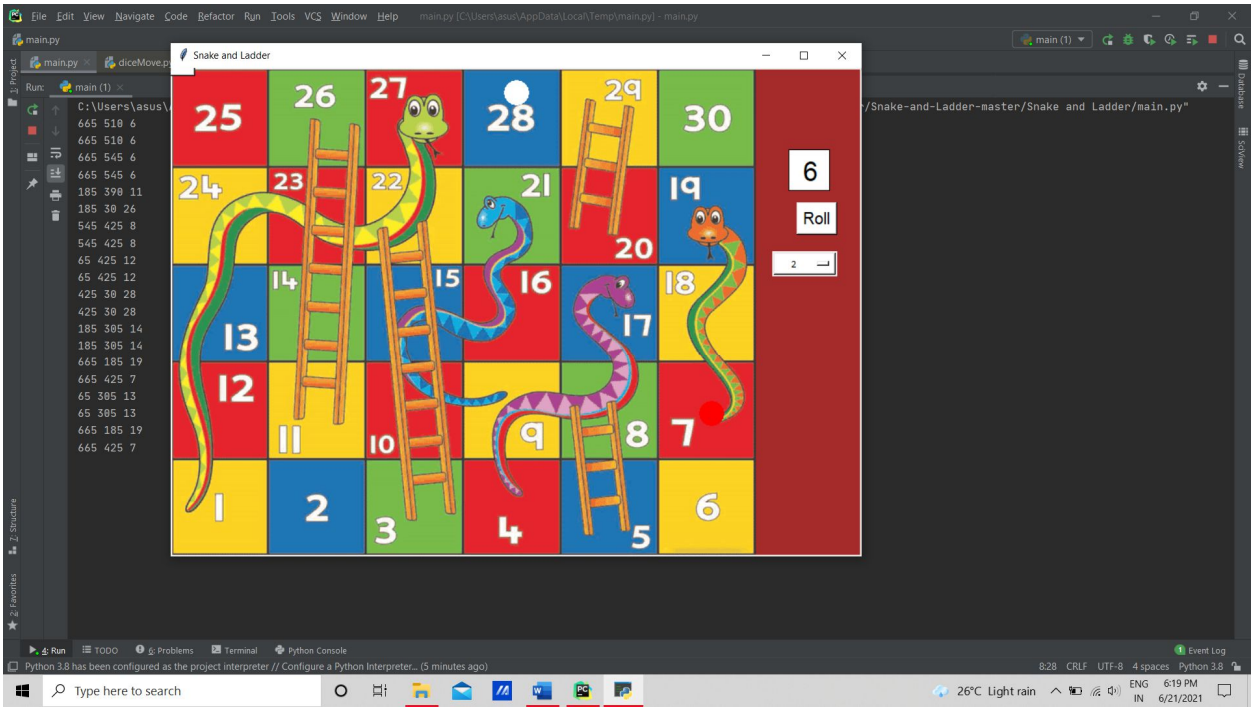

Snake and Ladder
5
Roll
2
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7

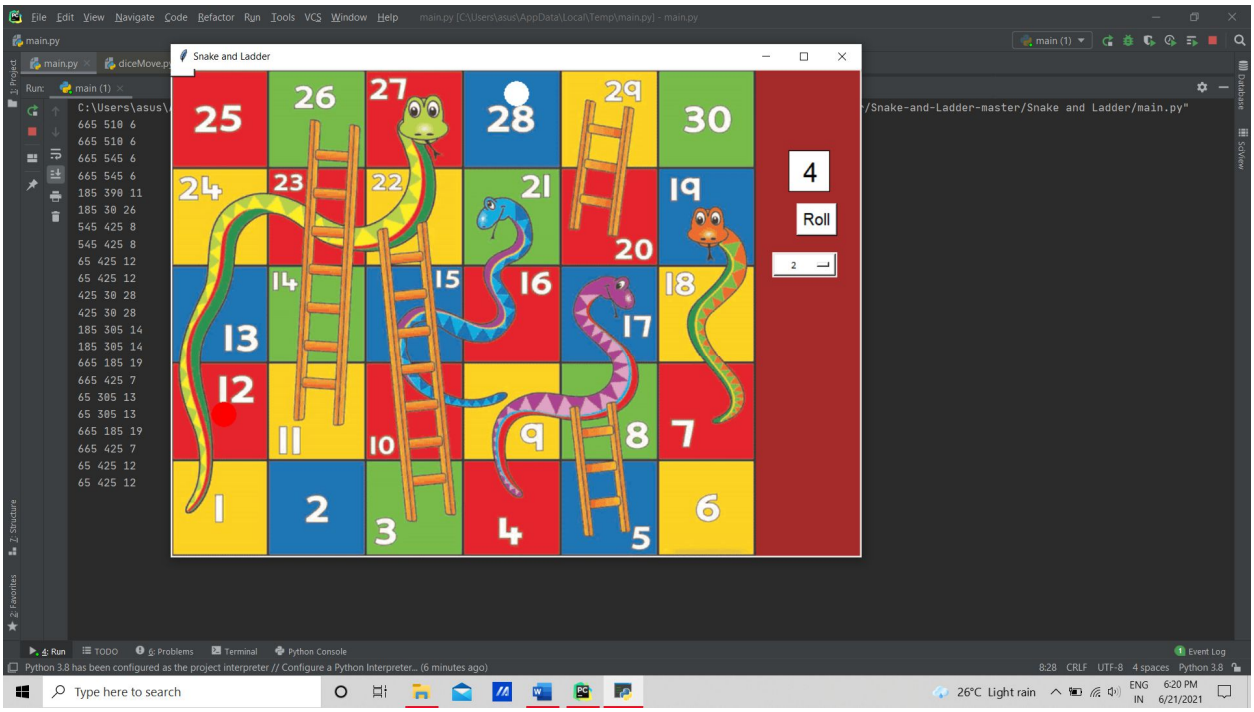

Snake and Ladder
6
Roll
2
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7
65 305 13
65 305 13

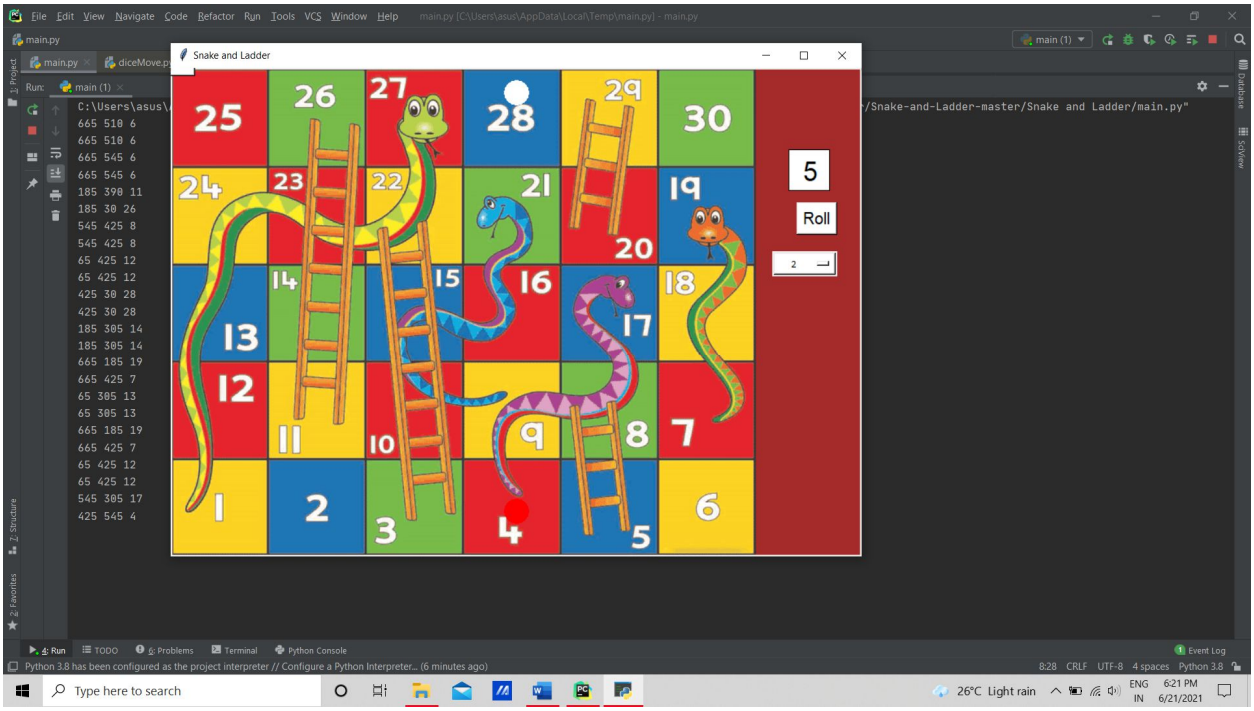

Snake and Ladder
6
Roll
2
C:\Users\asus\
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7
65 305 13
65 305 13
/Snake-and-Ladder-master/Snake and Ladder/main.py"
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (4 minutes ago)
Type here to search
26°C Light rain
ENG
IN
6:19 PM
6/21/2021

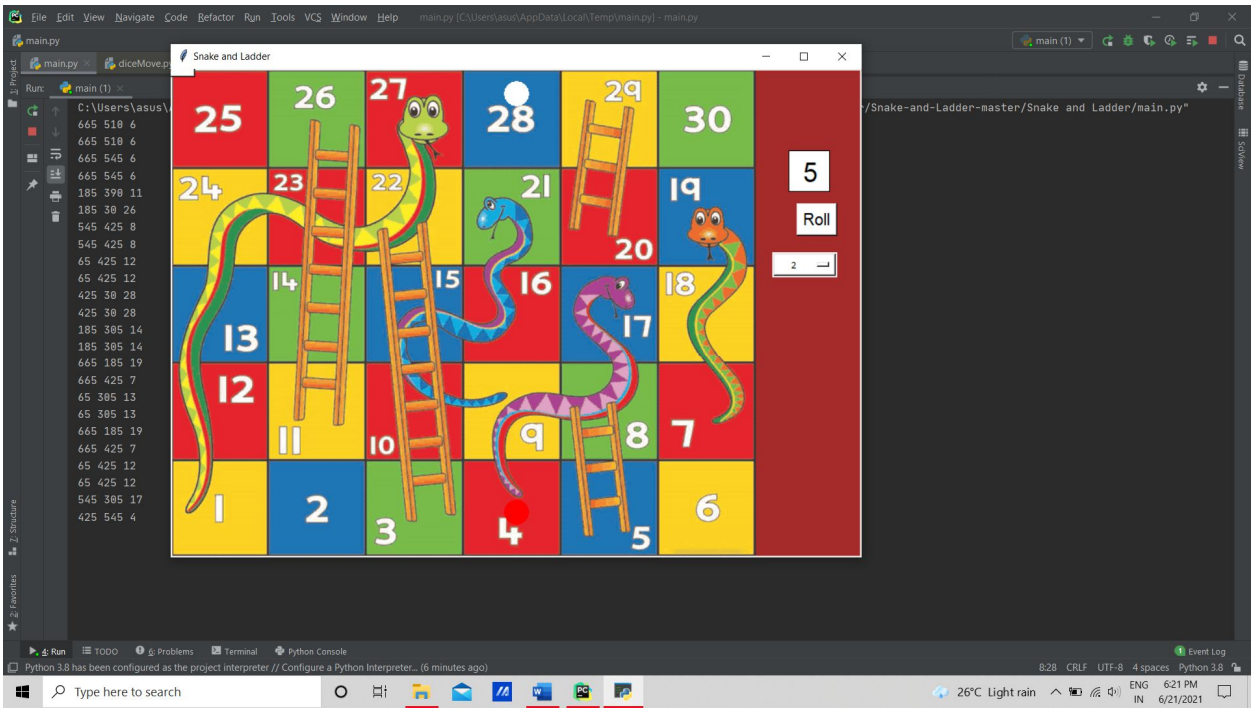

Snake and Ladder
6
Roll
2
C:\Users\asus\
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7
65 305 13
65 305 13
665 185 19
665 425 7
/Snake-and-Ladder-master/Snake and Ladder/main.py"
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (5 minutes ago)
Type here to search
26°C Light rain
ENG
IN
6:19 PM
6/21/2021

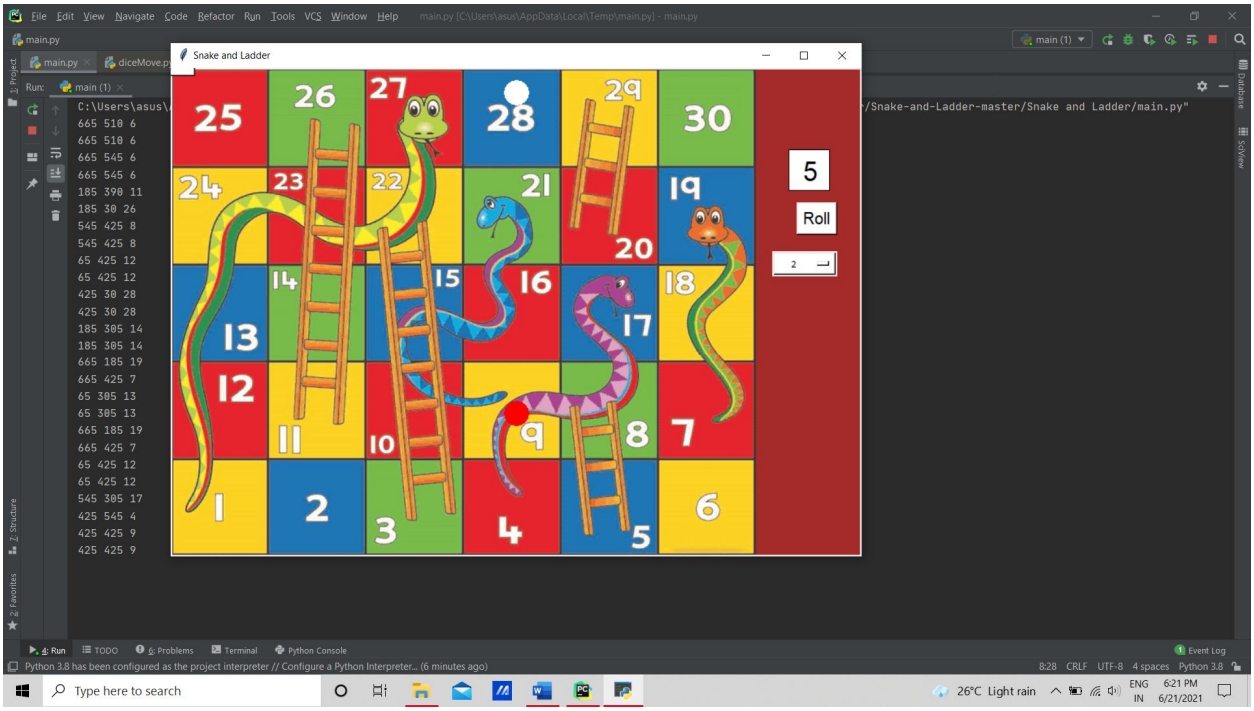

Snake and Ladder
6
Roll
2
C:\Users\asus\
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7
65 305 13
65 305 13
665 185 19
665 425 7
/Snake-and-Ladder-master/Snake and Ladder/main.py"
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (5 minutes ago)
Type here to search
26°C Light rain
ENG
IN
6:19 PM
6/21/2021

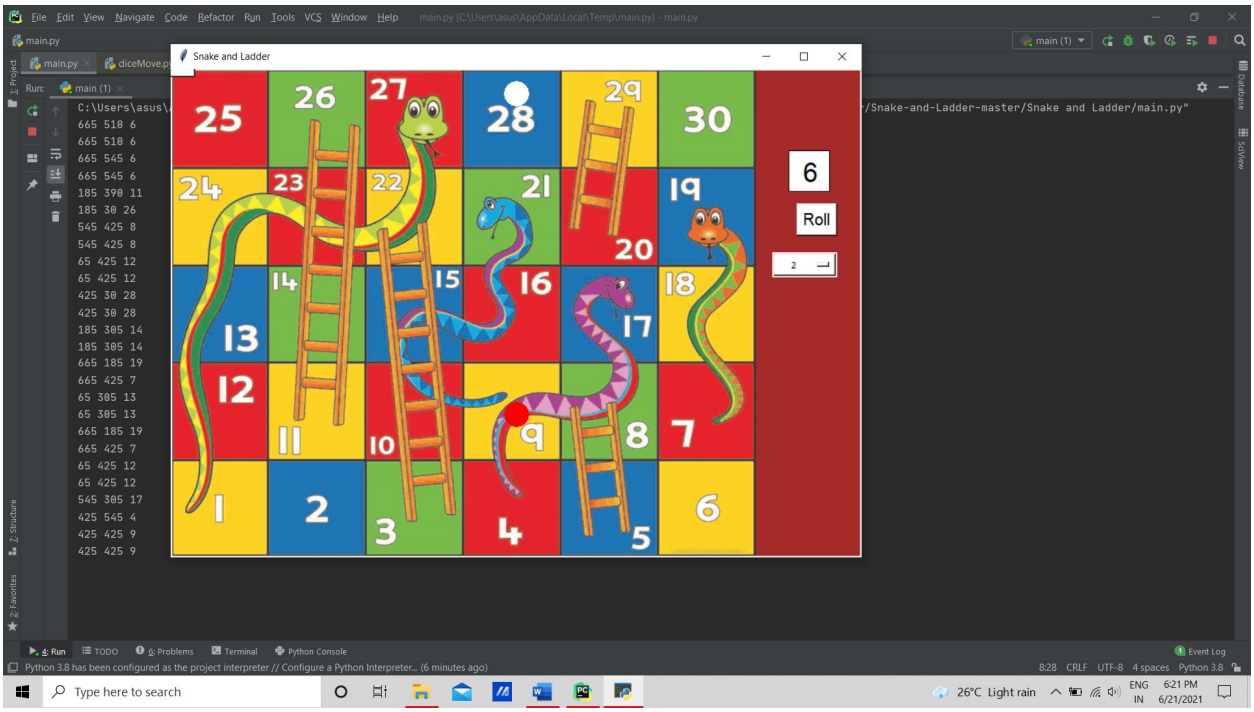

Snake and Ladder
6
Roll
2
C:\Users\asus\
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7
65 305 13
65 305 13
665 185 19
665 425 7
/Snake-and-Ladder-master/Snake and Ladder/main.py"
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (5 minutes ago)
Type here to search
26°C Light rain
ENG
IN
6:19 PM
6/21/2021

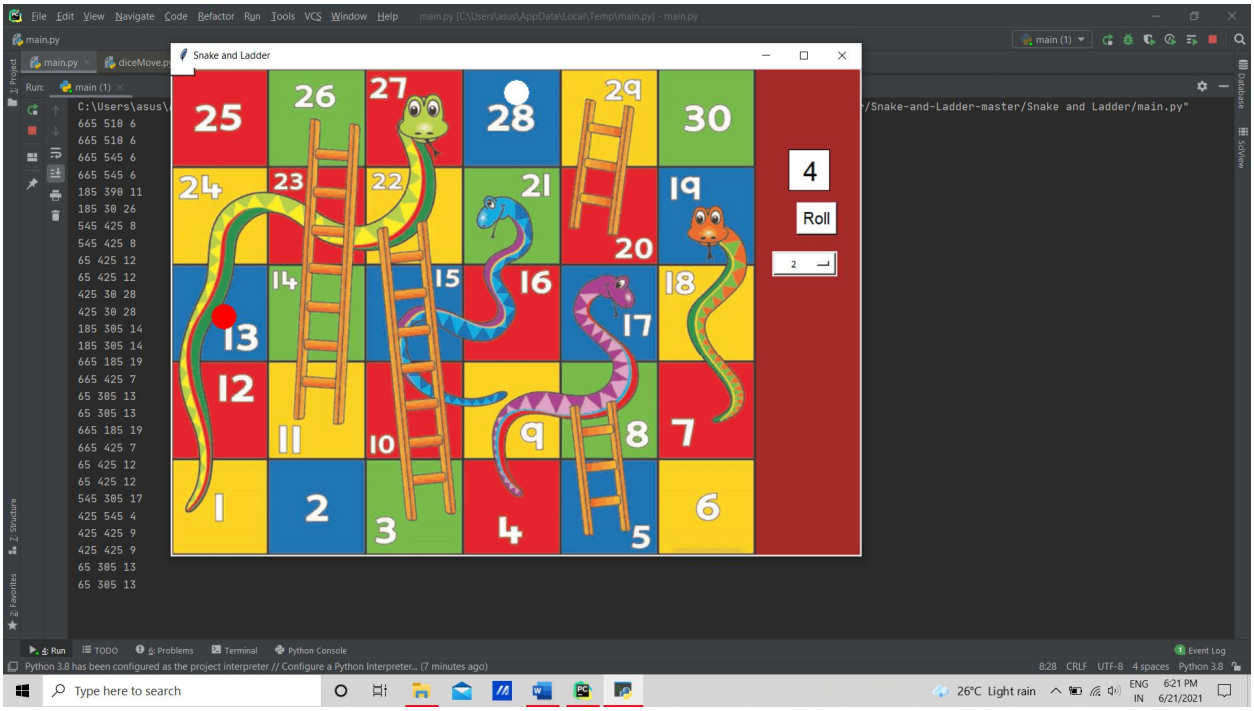

Snake and Ladder
4
Roll
2
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7
65 305 13
65 305 13
665 185 19
665 425 7
65 425 12
65 425 12

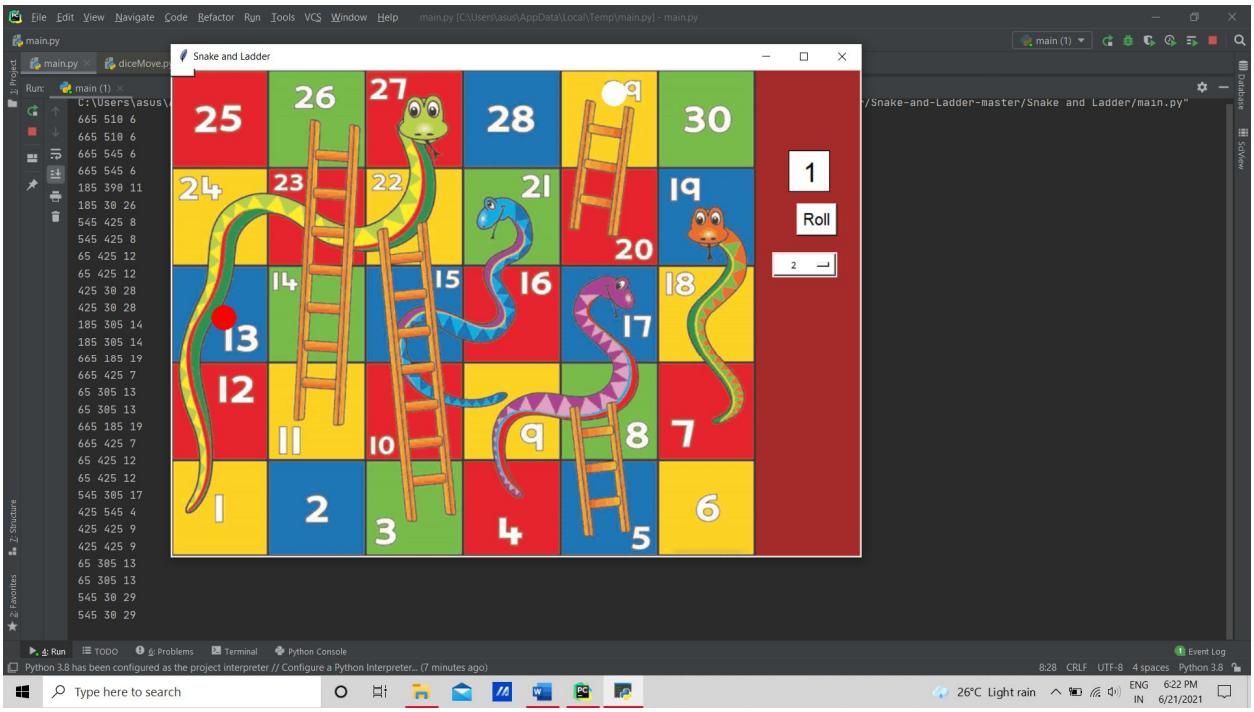

Snake and Ladder
5
Roll
2
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7
65 305 13
65 305 13
665 185 19
665 425 7
65 425 12
65 425 12
545 305 17
425 545 4

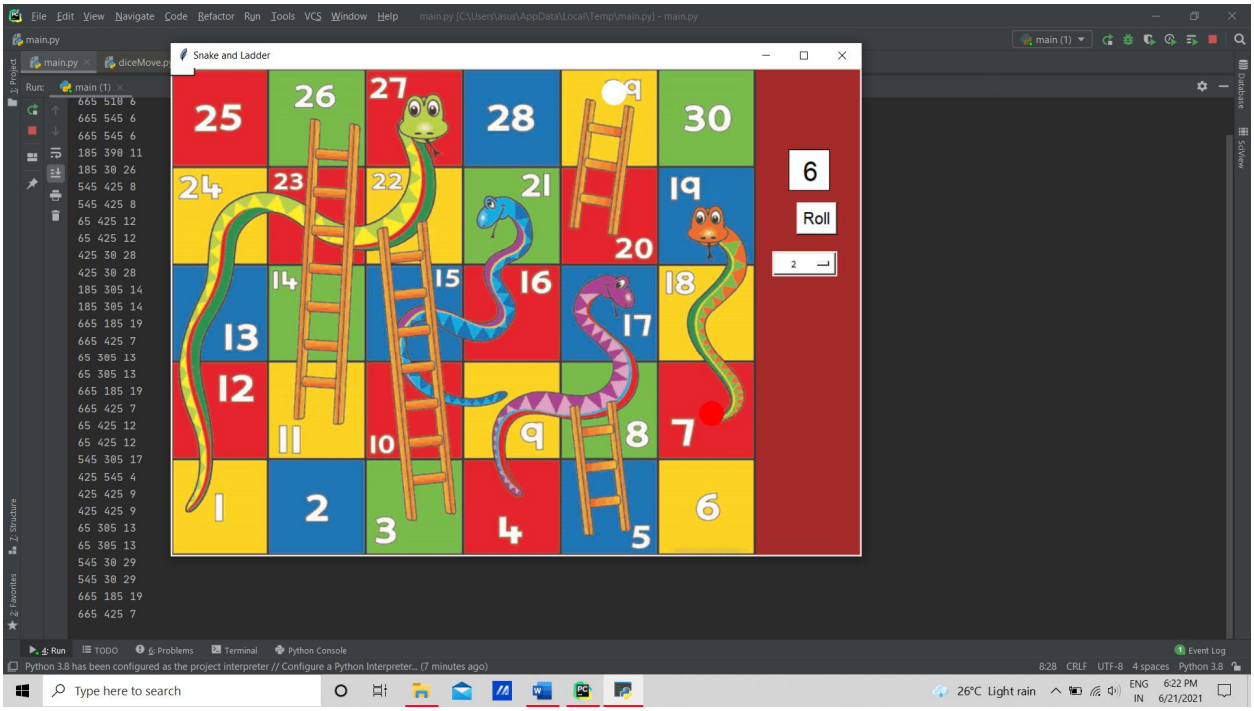

Snake and Ladder
5
Roll
2
main.py
Run:
main (1)
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (6 minutes ago)
Type here to search
26°C Light rain
ENG
IN
6:21 PM
6/21/2021


Snake and Ladder
5
Roll
2
main.py
Run:
main (1)
Python 3.8 has been configured as the project interpreter // Configure a Python Interpreter... (6 minutes ago)
Type here to search
26°C Light rain
ENG
IN
6:21 PM
6/21/2021

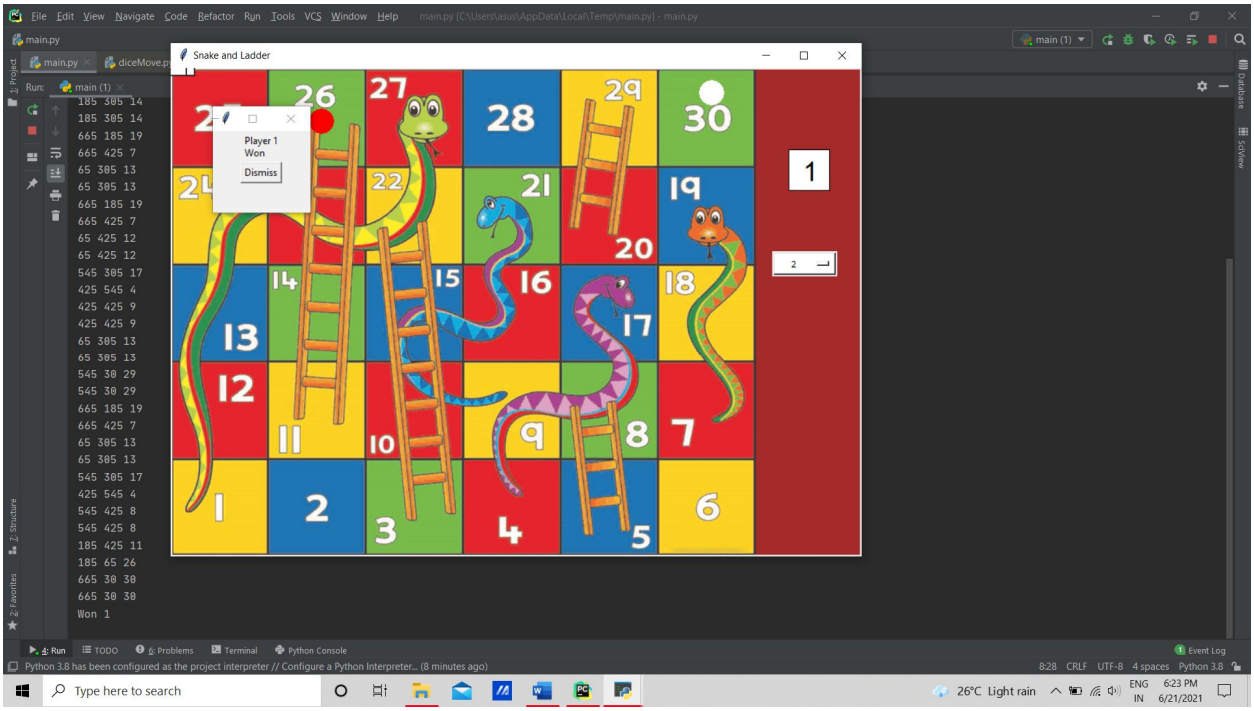

Snake and Ladder
6
Roll
2
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7
65 305 13
65 305 13
665 185 19
665 425 7
65 425 12
65 425 12
545 305 17
425 545 4
425 425 9
425 425 9


Snake and Ladder
4
Roll
2
665 510 6
665 510 6
665 545 6
665 545 6
185 390 11
185 30 26
545 425 8
545 425 8
65 425 12
65 425 12
425 30 28
425 30 28
185 305 14
185 305 14
665 185 19
665 425 7
65 305 13
65 305 13
665 185 19
665 425 7
65 425 12
65 425 12
545 305 17
425 545 4
425 425 9
425 425 9
65 305 13
65 305 13


Snake and Ladder
1
Roll
2
Type here to search
26°C Light rain
ENG
IN
6:22 PM
6/21/2021


Snake and Ladder
6
Roll
2
Type here to search
26°C Light rain
ENG
IN
6:22 PM
6/21/2021


Snake and Ladder
3
Roll
2
Type here to search
26°C Light rain
ENG
IN
6:22 PM
6/21/2021


Snake and Ladder
Player 1
Won
Dismiss
1
2
Won 1
Type here to search
26°C Light rain
ENG
IN
6:23 PM
6/21/2021