## MINI PROJECT REPORT

**on**

**SNAKES AND LADDERS**

**Submitted in partial fulfilment for the completion of**

**BE-III Semester**

**In**

**INFORMATION TECHNOLOGY**

**By**

**Shiva Raj Devarakonda (160119737171)**

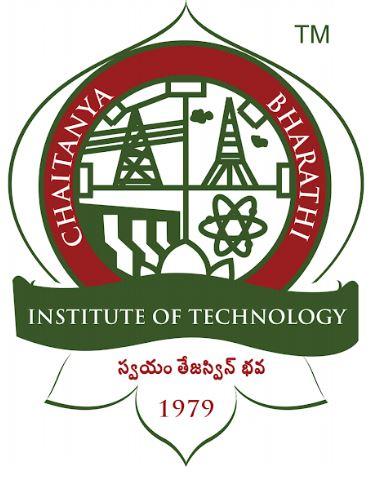
**Suhas Gajawada (160119737174)**

**Under the guidance of**

**Dr Pragati Priyadarshinee**

**Assistant Professor,**

**Dept. of IT, CBIT.**

****

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)**

**(Affiliated to Osmania University; Accredited by NBA(AICTE) and NAAC(UGC), ISO Certified 9001:2015)**

**GANDIPET, HYDERABAD – 500 075**

**Website:** [**www.cbit.ac.in**](http://www.cbit.ac.in)

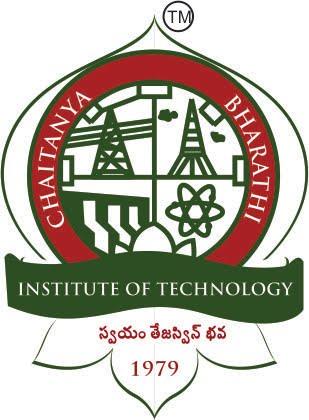
**2019-2020**

**CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)**

DEPARTMENT OF INFORMATION TECHNOLOGY

**(Affiliated to Osmania University)**

**GANDIPET, HYDERABAD – 500 075**

****

**CERTIFICATE**

This isto certify that the project work entitled “**Snakes and Ladders**” submitted to **CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY,** in partial fulfilment of the requirements for the award of the completion of III semesters of B.E in Information Technology, during the academic year 2020-2021, is a record of original work done by **Shiva Raj Devarakonda (160119737171), Suhas Gajawada (160119737174)** during the period of study in Department of IT, CBIT, HYDERABAD, under our supervision and guidance.

**Project Guide**  **Head of the Department**

**Dr Pragati Priyadarshinee Dr Radhika**

Assistant Professor, Dept. of IT, Professor, Dept. of IT,

CBIT, Hyderabad. CBIT, Hyderabad.

**ACKNOWLEDGEMENT**

It is our privilege to acknowledge with a deep sense of gratitude and devotion for keen personal interest and invaluable guidance rendered by our Project Guide

**Dr Pragati Priyadarshinee**, Assistant Professor, Department of Information Technology, Chaitanya Bharathi Institute of Technology.

Our respects and regards to **Dr Radhika**, Professor, Department of Information Technology, Chaitanya Bharathi Institute of Technology, for his invaluable suggestions that helped us in the successful completion of the project

We are grateful to our Principal **Dr G. P. Saradhi Varma**. Chaitanya Bharathi Institute of Technology, for his cooperation and encouragement.

Finally, we also thank all the faculty of Dept. of IT, CBIT, our friends, and all our family members who with their valuable suggestions and support, directly or indirectly helped us in completing this project work.

**ABSTRACT**

Snake and ladder is a very popular game in India and adopted with many variations in other places across the globe. It was considered a game of life- the snakes represented the vices and the ladders represented virtues.

The rules of the game are straightforward and are well-known. This mini-project aims to be playable by any number of players from one to four and is a text-based game. The game can be played in single-player mode or multiplayer mode. It can be played in several combinations of players, namely: one player and one or two or three computers, two players and one or two computers, three players and one computer and four players. The users can interact with the game using the keyboard. The positions of snakes and ladders are generated fixed. The purpose of the game is to get to the treasure on the 100th square box before the opponents. All players start from the same position and the dice roll which outputs a number from 1 to 6, using the random library, which determines the speed of movement. Player’s pawn moves forward by the number on the dice. If on completion of a move, a player's pawn lands on the lower-numbered end of a ladder, the player moves the pawn up to the ladder's higher-numbered square box. If the player lands on the higher-numbered square box of a snake, the pawn must be moved down to the snake's lower-numbered square box. The game outputs position, snake or ladder faced in each dice-roll.

Programming language: Python

IDE : Spyder

**CONTENTS**

**Pg. no.**

**ACKNOWLEDGEMENT iii**

**ABSTRACT iv**

**LIST OF FIGURES vi**

**LIST OF SCREEN SHOTS vii**

**1. INTRODUCTION 1**

1.1 Motivation 1

1.2 Basic Definition 1

1.3 Problem statement 1

**2. EXISTING SYSTEM 1**

**3. PROPOSED SYSTEM 2**

3.1 Methodology 2

3.2 Architecture of Proposed System 8

**4. SOFTWARE & HARDWARE REQUIREMENTS 21**

**5. IMPLEMENTATION OF PROJECT 21**

5.1 Results 21

5.2 Screen Shots 22

**6. CONCLUSION & FUTURE SCOPE 27**

**BIBILOGRAPHY**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure** | **Description** | **Page No.** |
| **Fig. 3.1.1** | **Flow chart to initiate players** | **3** |
| **Fig 3.1.2** | **Flowchart to decide the type of gameplay** | **4** |
| **Fig 3.1.3** | **Flowchart of gameplay of 2 players** | **5** |
| **Fig 3.1.4** | **Flowchart of gameplay of 3 players** | **6** |
| **Fig 3.1.5** | **Flowchart of gameplay of 4 players** | **7** |

**LIST OF SCREENSHOTS:**

|  |  |  |
| --- | --- | --- |
| **Screenshot** | **Description** | **Page No.** |
| **Screenshot 5.2.1** | **Snake and Ladders Board** | **22** |
| **Screenshot 5.2.2** | **players starting the game** | **23** |
| **Screenshot 5.2.3** | **playing with a computer** | **23** |
| **Screenshot 5.2.4** | **players starting the game** | **24** |
| **Screenshot 5.2.5** | **Getting a ladder** | **24** |
| **Screenshot 5.2.6** | **Getting a snake** | **24** |
| **Screenshot 5.2.7** | **players starting the game** | **25** |
| **Screenshot 5.2.8** | **player quitting the game** | **25** |
| **Screenshot 5.2.9** | **1 player playing with a computer** | **26** |
| **Screenshot 5.2.10** | **a player winning a game** | **26** |
| **Screenshot 5.2.11** | **more than 4 or less than 2 players want to play** | **26** |

**1. INTRODUCTION**

**1.1 MOTIVATION**

The main motivation behind us choosing to make “Snakes and Ladders” game is, it is a game of life. It teaches us that life has ups and downs. It is the most common and popular game of childhood. Its coding is challenging yet approachable. Thus, we decided to take up this as our project topic.

**1.2 BASIC DEFINITION**

a board game in which players move counters towards a goal at the top of the board, gaining an advantage by moving up pictures of ladders or a disadvantage by moving down pictures of snakes.

**1.3 PROBLEM STATEMENT**

Snakes and Ladders is a relatively basic game which everyone should at least play once in life. But it requires at least two persons, a Snake and ladders board, dice, coins, etc.

This whole process is achieved by a single code. Though the true experience of playing physically is not attained, the true essence of the game is intact.

**2. EXISTING SYSTEM**

Snakes and Ladders is an ancient Indian board game regarded today as a worldwide classic. It is played between two or more players on a game board having numbered, gridded squares. Several "ladders" and "snakes" are pictured on the board, each connecting two specific board squares. The objective of the game is to navigate one's game piece, according to dice rolls, from the start to the finish, helped or hindered by ladders and snakes, respectively.

The historic version had root in morality lessons, where a player's progression up the board represented a life journey complicated by virtues and vices. The snakes & ladders game is used to play in board manually. We are trying to develop a system which makes, look, and feel very interesting to play the game.

Snakes and Ladders originated in India as part of a family of dice board games that included Gyan chauper and pachisi (present-day Ludo and Parcheesi). It was known as moksha patam or vaikunthapaali or paramapada sopaanam (the ladder to salvation). The game made its way to England and was sold as "Snakes and Ladders", then the basic concept was introduced in the United States as Chutes and Ladders (an "improved new version of England's famous indoor sport") by game pioneer Milton Bradley in 1943.

**3. PROPOSED SYSTEM**

**3.1 METHODOLOGY**

The proposed system is designed to give a professional look and feel UI for Snakes and Ladders game. The gameplay is very simple. The player takes a turn rolling the single die and their coin moves the number of spaces that the die shows. If the player lands on the head of a snake, the game piece will be sent back to the location of that snake's tail. If the game piece lands at the foot of a ladder, the piece will climb up to the top, thus significantly advancing the journey to the 100th square. Then the turn will be given to the next players and computers, it present. Whoever reaches the 100th square first, is declared as the winner.

A picture containing shape, arrow

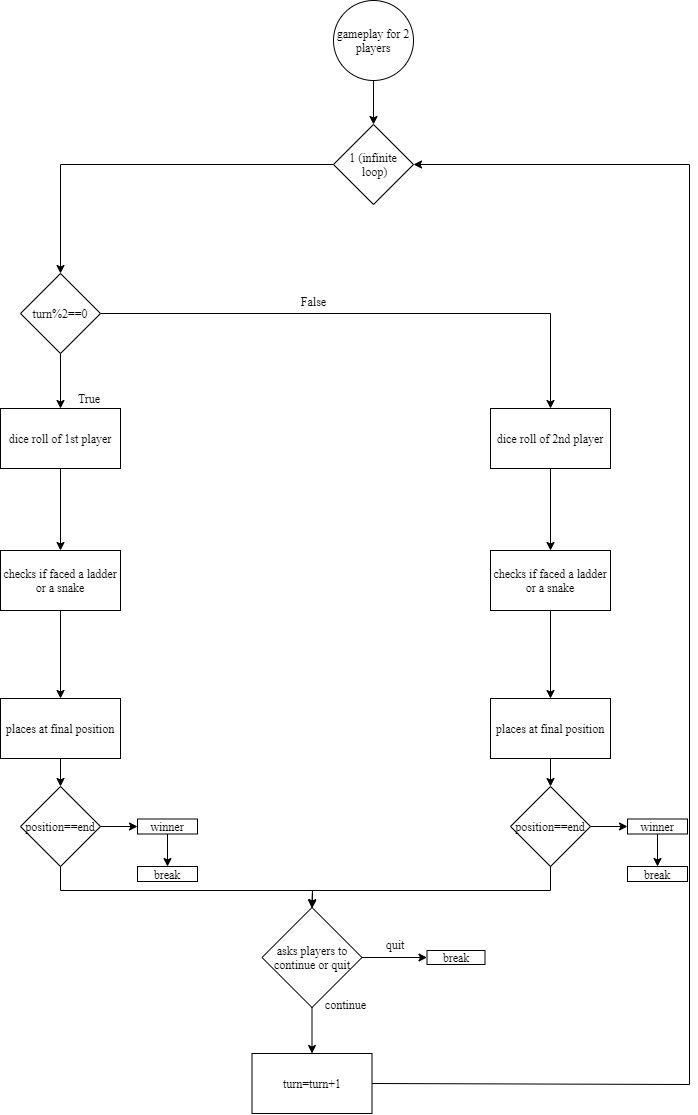
Description automatically generated

**Fig 3.1.1: Flow chart to initiate players**

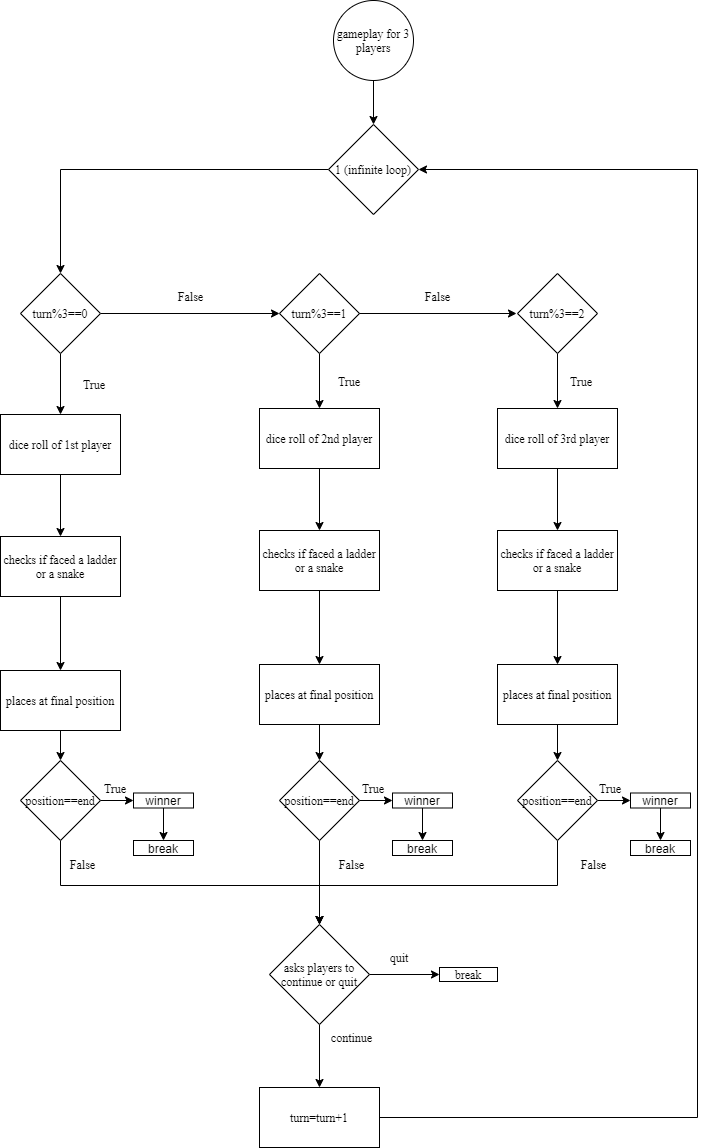
A picture containing shape

Description automatically generated

**Fig 3.1.2: Flowchart to decide the type of gameplay**

****

**Fig 3.1.3: Flowchart of gameplay of 2 players**

****

**Fig 3.1.4: Flowchart of gameplay of 3 players**

**Diagram

Description automatically generated**

**Fig 3.1.5: Flowchart of gameplay of 4 players**

**3.2 ARCHITECTURE OF PROPOSED SYSTEM**

from PIL import Image

from random import randint

end=100

def show\_board():

img=Image.open('board.jpg')

img.show()

def check\_ladder(pos):

if pos==1:

print('Ladder from 1 to 38')

return 38

elif pos==4:

print('Ladder from 4 to 14')

return 14

elif pos==8:

print('Ladder from 8 to 30')

return 30

elif pos==21:

print('Ladder from 21 to 42')

return 42

elif pos==28:

print('Ladder from 28 to 76')

return 76

elif pos==50:

print('Ladder from 50 to 67')

return 67

elif pos==71:

print('Ladder from 71 to 92')

return 92

elif pos==88:

print('Ladder from 88 to 99')

return 99

else:

#No ladder is encountered

return pos

def check\_snake(pos):

if pos==32:

print('Snake from 32 to 10')

return 10

elif pos==36:

print('Snake from 36 to 6')

return 6

elif pos==48:

print('Snake from 48 to 26')

return 26

elif pos==62:

print('Snake from 62 to 18')

return 18

elif pos==88:

print('Snake from 88 to 24')

return 24

elif pos==95:

print('Snake from 95 to 56')

return 56

elif pos==97:

print('Snake from 97 to 78')

return 78

else:

#No snake encountered

return pos

def play():

while(1):

no\_comp=0

no\_human=int(input('No of Players:'))

if(no\_human>4):

print("This game can't be played by more than 4 players:(")

continue

if(no\_human!=4):

no\_comp=int(input('No of computers: '))

if(no\_human==1 and no\_comp==0):

print('You need one more player or a computer to play the game :(')

elif (no\_human + no\_comp > 4):

print("This game can't be played by more than 4 players:(")

elif(no\_human==0 and no\_comp>0):

print("This game can't be played by only computers")

elif ((no\_human + no\_comp)<=4 and (no\_human + no\_comp)>1 and no\_human>=1):

break

if(no\_human==1):

p1\_name=input('Player 1, Please input your name: ')

p1\_pos=0

if(no\_comp==1):

p2\_name='computer'

p2\_pos=0

if(no\_comp==2):

p2\_name='computer 1'

p2\_pos=0

p3\_name='computer 2'

p3\_pos=0

if(no\_comp==3):

p2\_name='computer 1'

p2\_pos=0

p3\_name='computer 2'

p3\_pos=0

p4\_name='computer 3'

p4\_pos=0

elif(no\_human==2):

p1\_name=input('Player 1, Please input your name: ')

p2\_name=input('Player 2, Please input your name: ')

p1\_pos=0

p2\_pos=0

if(no\_comp==1):

p3\_name='computer'

p3\_pos=0

if(no\_comp==2):

p3\_name='computer 1'

p3\_pos=0

p4\_name='computer 2'

p4\_pos=0

elif(no\_human==3):

p1\_name=input('Player 1, Please input your name: ')

p2\_name=input('Player 2, Please input your name: ')

p3\_name=input('Player 3, Please input your name: ')

p1\_pos=0

p2\_pos=0

p3\_pos=0

if(no\_comp==1):

p4\_name='computer'

p4\_pos=0

elif(no\_human==4):

p1\_name=input('Player 1, Please input your name: ')

p2\_name=input('Player 2, Please input your name: ')

p3\_name=input('Player 3, Please input your name: ')

p4\_name=input('Player 4, Please input your name: ')

p1\_pos=0

p2\_pos=0

p3\_pos=0

p4\_pos=0

turn=0

#game driver for 2 players

if((no\_human+no\_comp)==2):

while(1):

if turn%2 == 0:

print(p1\_name, ' your turn')

print('Dice rolling!!')

dice = randint(1, 6)

print('Dice showed: ',dice)

p1\_pos = p1\_pos + dice

p1\_pos=check\_ladder(p1\_pos)

p1\_pos=check\_snake(p1\_pos)

if(p1\_pos>end):

print('Roll the dice such that you land on 100, try in next turn')

p1\_pos=p1\_pos-dice

print(p1\_name, "You are at position:", p1\_pos)

if(p1\_pos==end):

print(p1\_name, "Congratulations! You Won")

break

else:

print(p2\_name, ' your turn')

print('Dice rolling!!')

dice = randint(1, 6)

print('Dice showed: ',dice)

p2\_pos = p2\_pos + dice

p2\_pos=check\_ladder(p2\_pos)

p2\_pos=check\_snake(p2\_pos)

if(p2\_pos>end):

print('Roll the dice such that you land on 100, try in next turn')

p2\_pos=p2\_pos-dice

print(p2\_name, "You are at position:", p2\_pos)

if(p2\_pos==end):

print(p2\_name, "Congratulations! You Won")

break

c=int(input("Any player, press 1 to continue or 0 to quit without a winner: "))

if c==0:

c\_f=int(input("Are you sure (Yes(0), No(1)):"))

if(c\_f==0):

break

turn=turn+1

#game driver for 3 players

elif((no\_human+no\_comp)==3):

while(1):

if turn%3 == 0:

print(p1\_name, ' your turn')

print('Dice rolling!!')

dice = randint(1, 6)

print('Dice showed: ',dice)

p1\_pos = p1\_pos + dice

p1\_pos=check\_ladder(p1\_pos)

p1\_pos=check\_snake(p1\_pos)

if(p1\_pos>end):

print('Roll the dice such that you land on 100, try in next turn')

p1\_pos=p1\_pos-dice

print(p1\_name, "You are at position:", p1\_pos)

if(p1\_pos==end):

print(p1\_name, "Congratulations! You Won")

break

elif turn%3 == 1:

print(p2\_name, ' your turn')

print('Dice rolling!!')

dice = randint(1, 6)

print('Dice showed: ',dice)

p2\_pos = p2\_pos + dice

p2\_pos=check\_ladder(p2\_pos)

p2\_pos=check\_snake(p2\_pos)

if(p2\_pos>end):

print('Roll the dice such that you land on 100, try in next turn')

p2\_pos=p2\_pos-dice

print(p2\_name, "You are at position:", p2\_pos)

if(p2\_pos==end):

print(p2\_name, "Congratulations! You Won")

break

elif turn%3 == 2:

print(p3\_name, ' your turn')

print('Dice rolling!!')

dice = randint(1, 6)

print('Dice showed: ',dice)

p3\_pos = p3\_pos + dice

p3\_pos=check\_ladder(p3\_pos)

p3\_pos=check\_snake(p3\_pos)

if(p3\_pos>end):

print('Roll the dice such that you land on 100, try in next turn')

p3\_pos=p3\_pos-dice

print(p3\_name, "You are at position:", p3\_pos)

if(p3\_pos==end):

print(p3\_name, "Congratulations! You Won")

break

c=int(input("Any player, press 1 to continue or 0 to quit without a winner: "))

if c==0:

c\_f=int(input("Are you sure (Yes(0), No(1)):"))

if(c\_f==0):

break

turn=turn+1

#game driver for 4 players

elif((no\_human+no\_comp)==4):

while(1):

if turn%4 == 0:

print(p1\_name, ' your turn')

print('Dice rolling!!')

dice = randint(1, 6)

print('Dice showed: ',dice)

p1\_pos = p1\_pos + dice

p1\_pos=check\_ladder(p1\_pos)

p1\_pos=check\_snake(p1\_pos)

if(p1\_pos>end):

print('Roll the dice such that you land on 100, try in next turn')

p1\_pos=p1\_pos-dice

print(p1\_name, "You are at position:", p1\_pos)

if(p1\_pos==end):

print(p1\_name, "Congratulations! You Won")

break

elif turn%4 == 1:

print(p2\_name, ' your turn')

print('Dice rolling!!')

dice = randint(1, 6)

print('Dice showed: ',dice)

p2\_pos = p2\_pos + dice

p2\_pos=check\_ladder(p2\_pos)

p2\_pos=check\_snake(p2\_pos)

if(p2\_pos>end):

print('Roll the dice such that you land on 100, try in next turn')

p2\_pos=p2\_pos-dice

print(p2\_name, "You are at position:", p2\_pos)

if(p2\_pos==end):

print(p2\_name, "Congratulations! You Won")

break

elif turn%4 == 2:

print(p3\_name, ' your turn')

print('Dice rolling!!')

dice = randint(1, 6)

print('Dice showed: ',dice)

p3\_pos = p3\_pos + dice

p3\_pos=check\_ladder(p3\_pos)

p3\_pos=check\_snake(p3\_pos)

if(p3\_pos>end):

print('Roll the dice such that you land on 100, try in next turn')

p3\_pos=p3\_pos-dice

print(p3\_name, "You are at position:", p3\_pos)

if(p3\_pos==end):

print(p3\_name, "Congratulations! You Won")

break

elif turn%4 == 3:

print(p4\_name, ' your turn')

print('Dice rolling!!')

dice = randint(1, 6)

print('Dice showed: ',dice)

p4\_pos = p4\_pos + dice

p4\_pos=check\_ladder(p4\_pos)

p4\_pos=check\_snake(p4\_pos)

if(p4\_pos>end):

print('Roll the dice such that you land on 100, try in next turn')

p4\_pos=p4\_pos-dice

print(p4\_name, "You are at position:", p4\_pos)

if(p4\_pos==end):

print(p4\_name, "Congratulations! You Won")

break

c=int(input("Any player, press 1 to continue or 0 to quit without a winner: "))

if c==0:

c\_f=int(input("Are you sure (Yes(0), No(1)):"))

if(c\_f==0):

break

turn=turn+1

show\_board()

play()

**4. SOFTWARE & HARDWARE REQUIREMENTS**

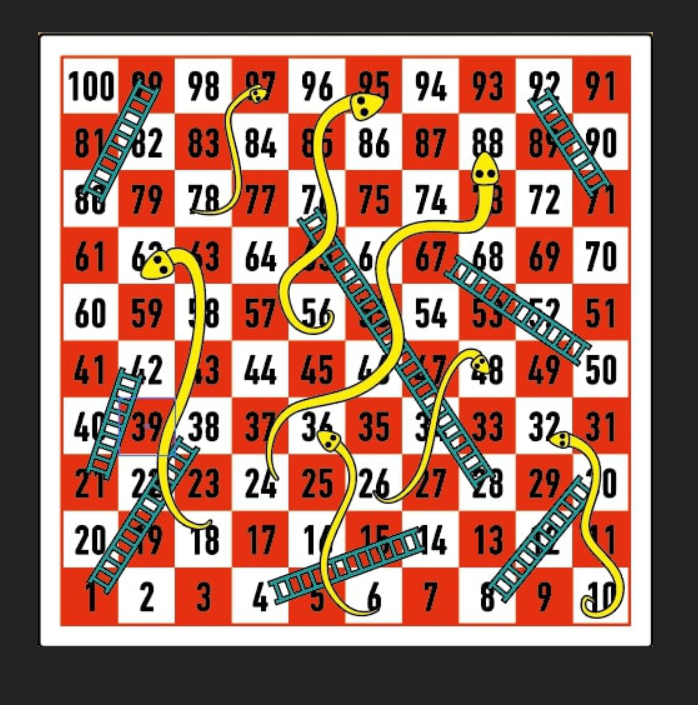
|  |  |  |
| --- | --- | --- |
| Operating System | : | Windows XP or more |
| Programming Language | : | Python 3 |
| Other Applications | : | Gallery |
| Integrated Development Environment | : | Spyder |
| Processor | : | Intel(R) Core (TM) i3 CPU M 350 @2.27GHz |
| RAM | : | 4 GB |
| Disk Space | : | 512 GB |

**5. IMPLEMENTATION OF PROJECT**

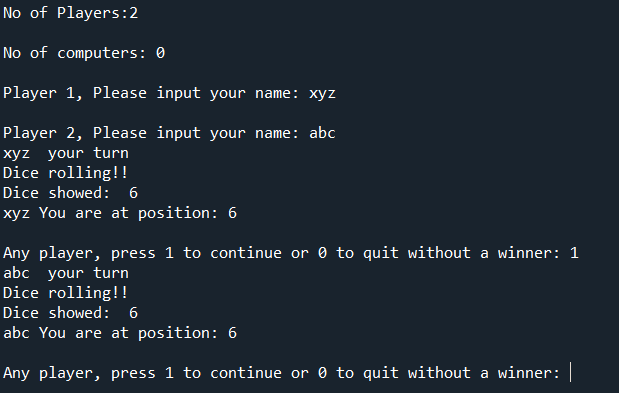
**5.1 RESULTS**

The code results in very snappy and responsive gameplay. This game can be played very efficiently with minimum hardware and software requirements. The idea of playing with computer works fine. Two players mode, three players mode and four players mode, everything works as intended.

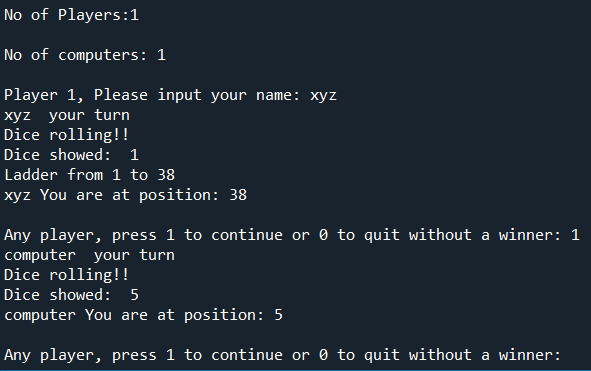
**5.2 SCREENSHOTS**

****

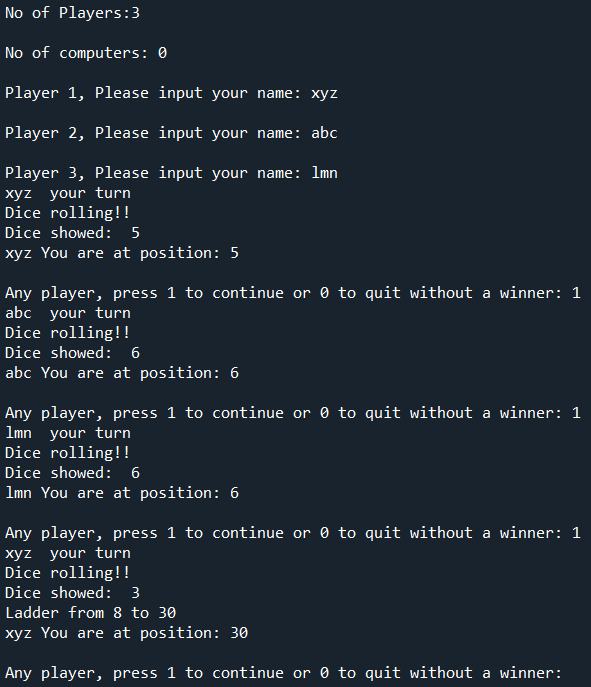
**Screenshot 5.2.1: Snake and Ladders Board**

****

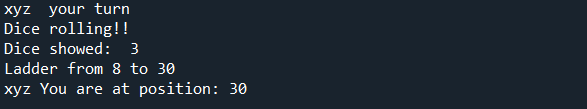
**Screenshot 5.2.2: 2 players starting the game**

****

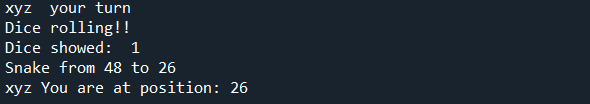
**Screenshot 5.2.3: playing with a computer**

****

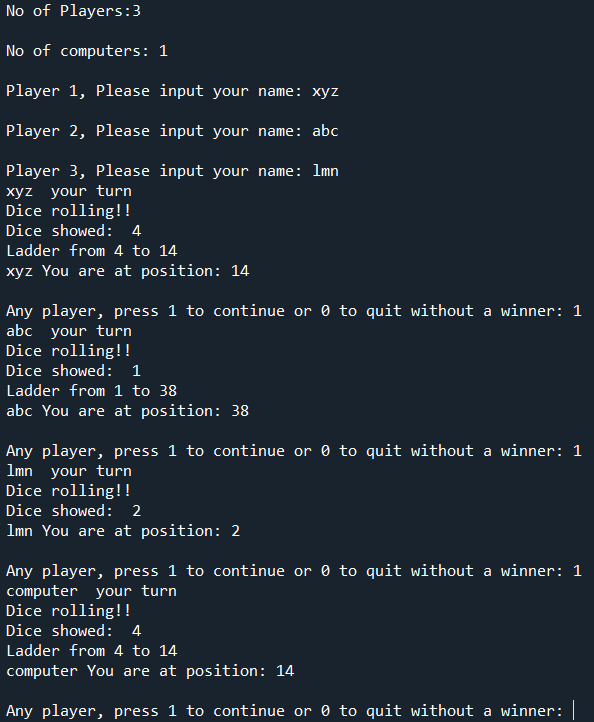
**Screenshot 5.2.4: 3 players starting the game**

****

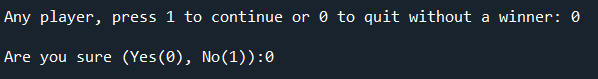
**Screenshot 5.2.5: Getting a ladder**

****

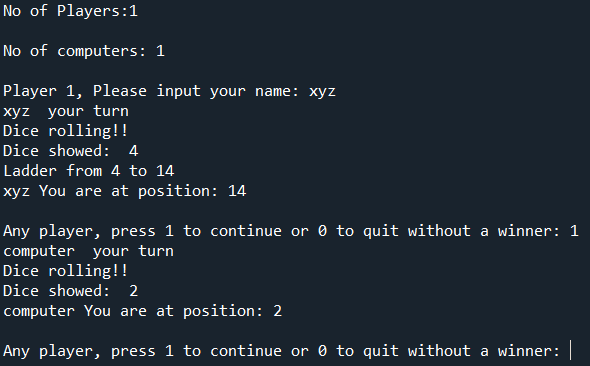
**Screenshot 5.2.6: Getting a snake**

****

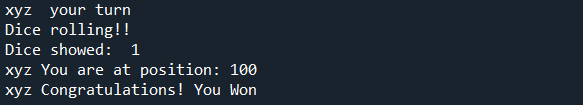
**Screenshot 5.2.7: 4 players starting the game**

****

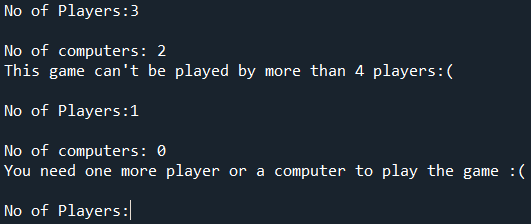
**Screenshot 5.2.8: player quitting the game**

****

**Screenshot 5.2.9: 1 player playing with a computer**

****

**Screenshot 5.2.10: player winning a game**

****

**Screenshot 5.2.11: more than 4 or less than 2 players want to play**

**6. CONCLUSION & FUTURE SCOPE**

We were successful in creating a digital version of the traditional Snakes and Ladders game with the possibility to play with bots as-well This game is very entertaining and engaging. This game is power efficient, quick, and snappy. We have learnt the process of documentation, managing, planning, coordinating and developed our programming skills in the journey of making this game for the Mini project. Our journey has been the same as this game, with ups and downs.

This game is still at its premature stage and has a very wide scope of future enhancements. We can add a graphical interface, more interactive elements, more levels, online multiplayer mode, creating a port for mobile devices.

**BIBLIOGRAPHY**

1. https://www.ymimports.com/pages/how-to-play-snakes-and-ladders
2. https://www.lexico.com/definition/snakes\_and\_ladders
3. https://a4games.company/snakes-and-ladders/#:~:text=Snakes%20and%20Ladders%20originated%20in,by%20the%20name%20Moksha%20Patam.
4. https://en.wikipedia.org/wiki/Snakes\_and\_Ladders