|  |
| --- |
| **Sinhgad Technical Education Society**  **RMD Sinhgad School of Engineering-Warje,Pune-58** |
| **Department of Computer Engineering** |

****

**DEPARTMENT OF COMPUTER ENGINEERING**

**210256: Data Structures & Algorithms**

**Academic Year: 2022-23**

**“Snakes & Ladders Game in Python”**

**Project Domain: Data Structures**

|  |
| --- |
| **Sinhgad Technical Education Society**  **RMD Sinhgad School of Engineering-Warje,Pune-58** |
| **Department of Computer Engineering** |

****

**Certificate**

This is to certify that, **Vedant Hendre**, a student of class **SE A**, Roll No. **44** has completed all the Term Work & Practical Work in the subject **Data Structures & Algorithms (DSAL)** satisfactorily in the Department of Computer Engineering as prescribed by Savitribai Phule Pune University, in the academic year 2022 -2023.

Mrs. Vina M. Lomte Dr. V. V. Dixit

Head of Department Principal

Department of Computer Engineering RMD Sinhgad School Of Engineering

Date: 17/05/2023

**ACKNOWLEDGEMENT**

I am glad to express my gratitude to all who rendered their valuable help for the

successful completion of this project.

I am grateful to my guide Prof. Shrutika Nalwade Ma’am for her guidance and encouragement in this work. Her expert suggestions and scholarly feedback greatly enhanced the effectiveness of this work.

I would also like to express my appreciation and thanks to Prof. Vina M. Lomte, Head of

Department, Computer Engineering.

I would also like to express our appreciation and thanks to all my colleagues and family

members who knowingly or unknowingly have assisted and encouraged me throughout

this work.

Table of Contents

[Abstract 5](#_Toc135176483)

[Problem Statement 5](#_Toc135176484)

[Introduction 5](#_Toc135176485)

[Objectives 5](#_Toc135176486)

[Software & Hardware Requirements 6](#_Toc135176487)

[Advantages & Disadvantages 6](#_Toc135176488)

[Implementation 6](#_Toc135176489)

[Results 6](#_Toc135176490)

[Code 7](#_Toc135176491)

[Challenges Faced 11](#_Toc135176492)

[Conclusion & Future Scope 11](#_Toc135176493)

[References 12](#_Toc135176494)

# Abstract

This project is designed to implement a basic version of the Snakes & Ladders Game. The language used here is Python. The game is played through the terminal. It is a lightweight game (No GUI Involved) that makes use of data structures to instantiate obstacles & boosts in the game randomly.

# Problem Statement

Design a mini project to implement Snake and Ladders Game using Python.

# Introduction

Snakes and Ladders is a popular board game that originated in ancient India and has been played for centuries. It is a game of chance that involves moving a token along a numbered grid according to the roll of a dice. The game also has snakes and ladders that can either take the player down or up the grid, respectively. The game is fun and easy to play, and can also teach some basic concepts of probability and mathematics.

The motivation for designing a Snakes and Ladders game using Python is to apply the object-oriented programming (OOP) principles and learn how to use various data structures, such as dictionaries, lists, and classes. Python is a versatile and powerful programming language that can be used for various purposes, such as web development, data analysis, machine learning, and game development. Python also has many built-in modules and libraries that can make coding easier and faster.

# Objectives

The objectives of designing a Snakes and Ladders game using Python are:

* To create a text-based version of the game that can be played in the terminal by two players.
* To create a graphical user interface (GUI) version of the game that can be played on a computer screen by two players or against the computer.
* To implement the rules and logic of the game using OOP concepts, such as classes, objects, methods, inheritance, and polymorphism.
* To use data structures, such as dictionaries and lists, to store the information about the board, the snakes, the ladders, and the players.
* To use random module to generate random numbers for the dice rolls.
* To use time module to add delays between actions for better user experience.

# Software & Hardware Requirements

The software and hardware requirements for designing a Snakes and Ladders game using Python are:

* A computer with [Python 3](https://www.python.org/downloads/) installed. Python can be downloaded from
* A text editor or an integrated development environment (IDE) to write and run Python code. Some popular options are IDLE, PyCharm, Visual Studio Code, Sublime Text, Atom, etc.
* A terminal or a command prompt to run the text-based version of the game.
* A snake.png and a ladder.png image files to display snakes and ladders on the GUI version of the game.

# Advantages & Disadvantages

The advantages of designing a Snakes and Ladders game using Python are:

* It is a fun and engaging way to learn Python programming and OOP concepts.
* It is a creative and challenging project that can showcase your coding skills and problem-solving abilities.
* It is a flexible and scalable project that can be extended or modified according to your preferences and requirements.
* It is a cross-platform project that can run on any operating system that supports Python.

The disadvantages of designing a Snakes and Ladders game using Python are:

* It may require some prior knowledge of Python syntax and OOP concepts to understand and implement the code.
* It may require some additional modules or libraries that are not part of the standard Python distribution to create GUI elements or manipulate images.
* It may encounter some errors or bugs during the development or execution of the code that need to be fixed or debugged.

# Implementation

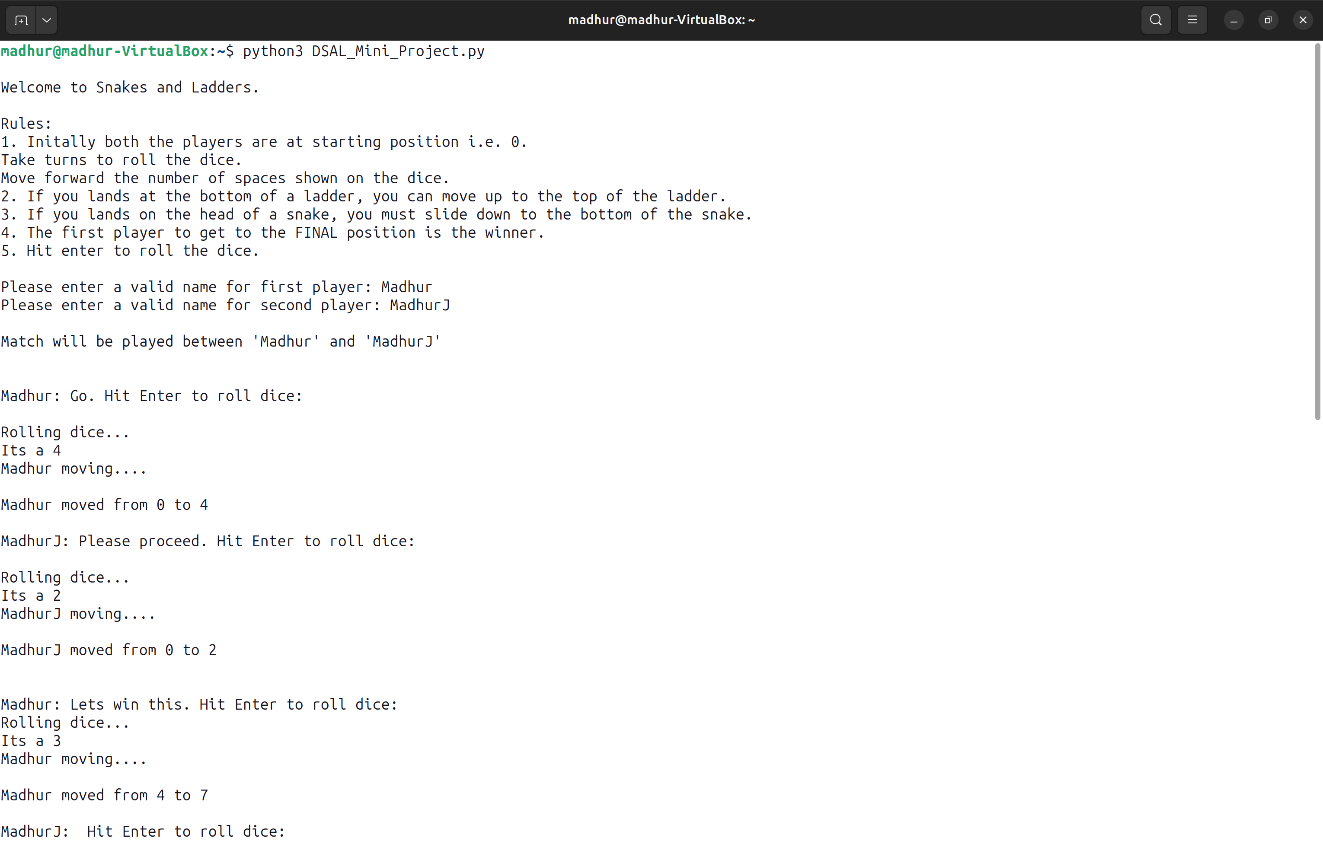
The implementation of designing a Snakes and Ladders game using Python can be summarized as follows:

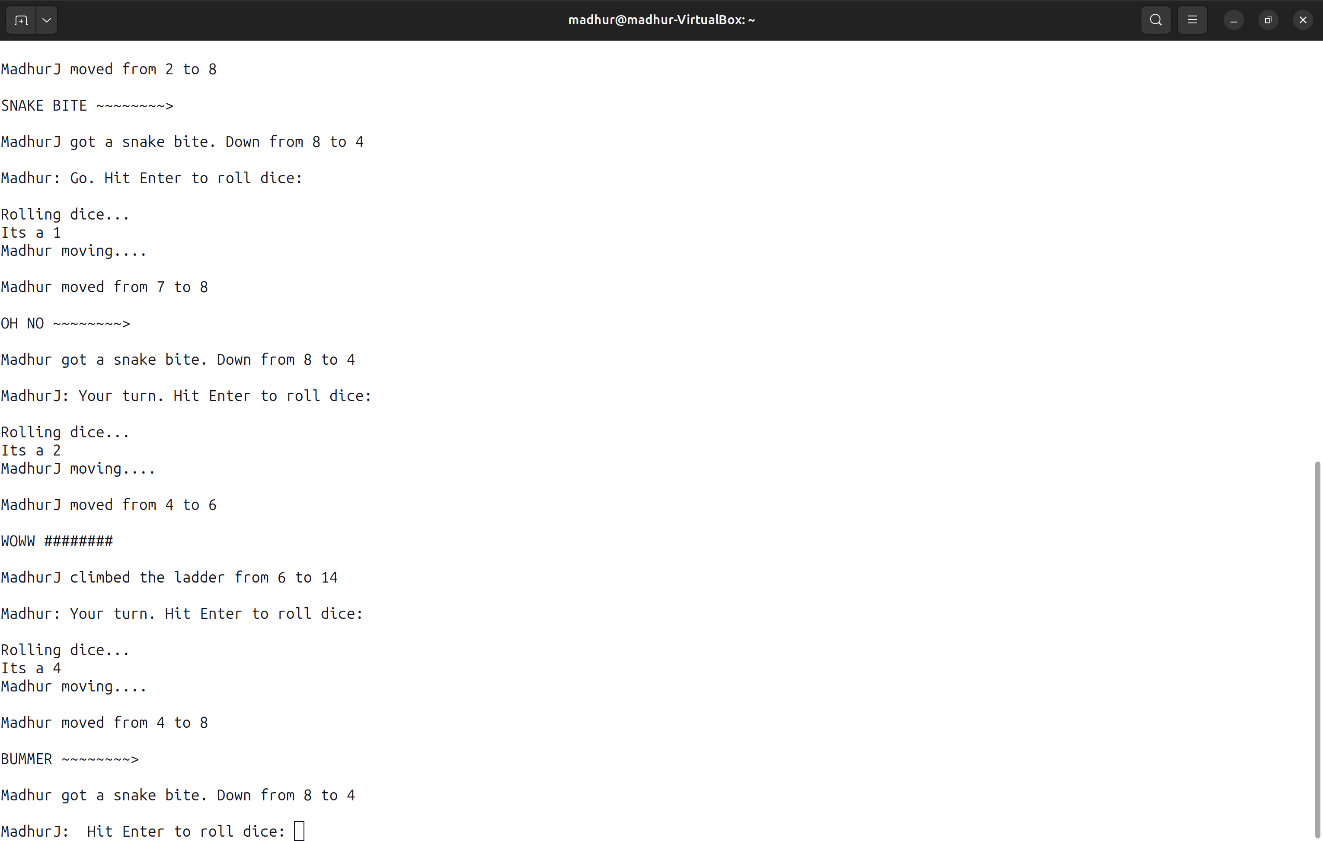
For the text-based version, create a GamePlayer class to store the player’s information, a MovingEntity class to represent snakes and ladders, and a main function to run the game logic using dictionaries, lists, random module, and time module.

For the GUI version, create a Board class to draw the board using tkinter module, a SnakeLadder class to inherit from MovingEntity class and display snakes and ladders using PIL module, and a main function to run the game logic using the same data structures and modules as the text-based version.

# Results

The final output looks like this:





# Code

import time

import random

import sys

SLEEP\_BETWEEN\_ACTIONS = 1

MAX\_VAL = 100

DICE\_FACE = 6

# snake takes you down from 'key' to 'value'

snakes = {

 8: 4,

 18: 1,

 26: 10,

 39: 5,

 51: 6,

 54: 36,

 56: 1,

 60: 23,

 75: 28,

 83: 45,

 85: 59,

 90: 48,

 92: 25,

 97: 87,

 99: 63

}

# ladder takes you up from 'key' to 'value'

ladders = {

 3: 20,

 6: 14,

 11: 28,

 15: 34,

 17: 74,

 22: 37,

 38: 59,

 49: 67,

 57: 76,

 61: 78,

 73: 86,

 81: 98,

 88: 91

}

player\_turn\_text = [

 "Your turn.",

 "Go.",

 "Please proceed.",

 "Lets win this.",

 "Are you ready?",

 "",

]

snake\_bite = [

 "boohoo",

 "bummer",

 "snake bite",

 "oh no",

 "dang"

]

ladder\_jump = [

 "woohoo",

 "woww",

 "nailed it",

 "woah",

 "yaayyy"

]

def welcome\_msg():

    msg ="""

Welcome to Snakes and Ladders.

Rules:

1. Initally both the players are at starting position i.e. 0.

Take turns to roll the dice.

Move forward the number of spaces shown on the dice.

2. If you lands at the bottom of a ladder, you can move up to the top of the ladder.

3. If you lands on the head of a snake, you must slide down to the bottom of the snake.

4. The first player to get to the FINAL position is the winner.

5. Hit enter to roll the dice.

"""

    print(msg)

def get\_player\_names():

    player1\_name = None

    while not player1\_name:

        player1\_name = input("Please enter a valid name for first player: ").strip()

        player2\_name = None

        while not player2\_name:

            player2\_name = input("Please enter a valid name for second player: ").strip()

    print("\nMatch will be played between '" + player1\_name + "' and '" + player2\_name + "'\n")

    return player1\_name, player2\_name

def get\_dice\_value():

    time.sleep(SLEEP\_BETWEEN\_ACTIONS)

    dice\_value = random.randint(1, DICE\_FACE)

    print("Its a " + str(dice\_value))

    return dice\_value

def got\_snake\_bite(old\_value, current\_value, player\_name):

    print("\n" + random.choice(snake\_bite).upper() + " ~~~~~~~~>")

    print("\n" + player\_name + " got a snake bite. Down from " + str(old\_value) + " to " + str(current\_value))

def got\_ladder\_jump(old\_value, current\_value, player\_name):

    print("\n" + random.choice(ladder\_jump).upper() + " ########")

    print("\n" + player\_name + " climbed the ladder from " + str(old\_value) + " to " + str(current\_value))

def snake\_ladder(player\_name, current\_value, dice\_value):

    time.sleep(SLEEP\_BETWEEN\_ACTIONS)

    old\_value = current\_value

    current\_value = current\_value + dice\_value

    if current\_value > MAX\_VAL:

        print("You need " + str(MAX\_VAL - old\_value) + " to win this game. Keep trying.")

        return old\_value

    print("\n" + player\_name + " moved from " + str(old\_value) + " to " + str(current\_value))

    if current\_value in snakes:

        final\_value = snakes.get(current\_value)

        got\_snake\_bite(current\_value, final\_value, player\_name)

    elif current\_value in ladders:

        final\_value = ladders.get(current\_value)

        got\_ladder\_jump(current\_value, final\_value, player\_name)

    else:

        final\_value = current\_value

    return final\_value

def check\_win(player\_name, position):

    time.sleep(SLEEP\_BETWEEN\_ACTIONS)

    if MAX\_VAL == position:

        print("\n\n\nThats it.\n\n" + player\_name + " has won the game.")

        print("Congratulations " + player\_name)

        sys.exit(1)

def start():

    welcome\_msg()

    time.sleep(SLEEP\_BETWEEN\_ACTIONS)

    player1\_name, player2\_name = get\_player\_names()

    time.sleep(SLEEP\_BETWEEN\_ACTIONS)

    player1\_current\_position = 0

    player2\_current\_position = 0

    while True:

        time.sleep(SLEEP\_BETWEEN\_ACTIONS)

        input\_1 = input("\n" + player1\_name + ": " + random.choice(player\_turn\_text) + " Hit Enter to roll dice: ")

        print("\nRolling dice...")

        dice\_value = get\_dice\_value()

        time.sleep(SLEEP\_BETWEEN\_ACTIONS)

        print(player1\_name + " moving....")

        player1\_current\_position = snake\_ladder(player1\_name, player1\_current\_position, dice\_value)

        check\_win(player1\_name, player1\_current\_position)

        input\_2 = input("\n" + player2\_name + ": " + random.choice(player\_turn\_text) + " Hit Enter to roll dice: ")

        print("\nRolling dice...")

        dice\_value = get\_dice\_value()

        time.sleep(SLEEP\_BETWEEN\_ACTIONS)

        print(player2\_name + " moving....")

        player2\_current\_position = snake\_ladder(player2\_name, player2\_current\_position, dice\_value)

        check\_win(player2\_name, player2\_current\_position)

if \_\_name\_\_ == "\_\_main\_\_":

    start()

# Challenges Faced

Some of the challenges faced in designing a Snakes and Ladders game using Python are:

* Choosing an appropriate data structure to represent the board, the snakes, the ladders, and the players. A dictionary can be used to store the start and end positions of the snakes and ladders, while a list can be used to store the current positions of the players.
* Implementing the logic of moving the players according to the dice roll and checking for snakes and ladders. A loop can be used to iterate over the possible moves for each player and update their position accordingly. A conditional statement can be used to check if the player lands on a snake or a ladder and move them to the respective end position.
* Handling the edge cases, such as when the player reaches or exceeds the final position, when the player rolls a 6 and gets an extra turn, when the player gets three consecutive 6s and loses their turn, or when the player makes an invalid move. These cases can be handled using conditional statements and variables to keep track of the game state.
* Creating a user-friendly interface that displays the welcome message, the player names, the dice value, the player position, and the game result. The print function can be used to display text messages on the terminal, while the input function can be used to get user input.
* Adding some effects and variations to make the game more interesting and realistic. The time module can be used to add delays between actions for better user experience. The random module can be used to generate random messages for snake bites and ladder jumps. The player\_turn\_text, snake\_bite, and ladder\_jump lists can be used to store these messages.

# Conclusion & Future Scope

The conclusion of designing a Snakes and Ladders game using Python is that it is a fun and engaging way to learn Python programming and OOP concepts. It is also a creative and challenging project that can showcase your coding skills and problem-solving abilities. It is also a flexible and scalable project that can be extended or modified according to your preferences and requirements.

The future scope of designing a Snakes and Ladders game using Python is that it can be improved or enhanced in various ways, such as:

Creating a graphical user interface (GUI) version of the game that can be played on a computer screen by two players or against the computer.

Adding more features and functionalities to the game, such as different board sizes, different number of snakes and ladders, different dice faces, different themes and graphics, sound effects, etc.

Adding more complexity and difficulty to the game, such as adding more rules and constraints, adding more types of moving entities (such as wormholes), adding power-ups and penalties, etc.

Adding more interactivity and engagement to the game, such as adding multiplayer mode, online mode, chat feature, leader board feature, etc.

I would like to thank my respected faculty for providing me an opportunity to present this mini project & imbibe the value of learning & experimenting. Their guidance has been immensely helpful in accomplishing this project.

# References

1. <https://github.com/atharvabhide/SPPU-COMP-2019-PRACTICALS/blob/main/DSA/13_snake_and_ladders.py>
2. <https://www.geeksforgeeks.org/design-snake-and-ladder-game-using-python-oops/>
3. <https://dev.to/anuragrana/text-based-snake-and-ladder-game-in-python-196l>
4. <https://copyassignment.com/snake-and-ladder-game-in-python/>
5. <https://www.geeksforgeeks.org/snake-ladder-problem-2/>
6. <https://github.com/anuragrana/text-based-snake-and-ladder-game-in-python>