ROCK PAPER SCISSORS

A Course Based Project Report Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN

CSE-CYS

Submitted by

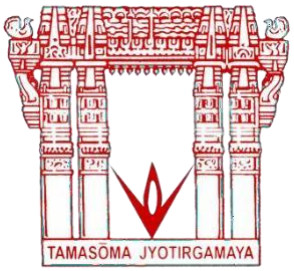
T.KRISHNA SOWMYA 21071A6259

T.ANSHU KARAN 21071A6258

Under the guidance of

E. Lalitha

(Assistant Professor, Department of CSE- CYS , VNR VJIET)



DEPARTMENT OF CSE-CYS

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA
Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi,
Affiliated to JNTUH Recognized as "College with Potential for Excellence"
by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

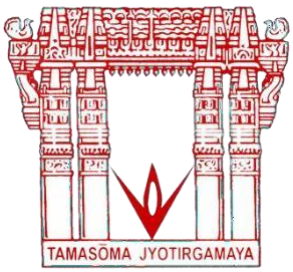
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DEPARTMENT OF CSE
(CYBER SECURITY)



CERTIFICATE

This is to certify that the project report entitled "**Rock Paper Scissors"** is a bonafide work done under our supervision and is being submitted by **T KRISHNA SOWMYA (21071A6259), T ANSHU KARAN (21071A6258),**in partial fulfillment for the award of the degree of Bachelor of Technology in CSE, of the VNRVJIET, Hyderabad during the academic year 2022-2023. Certified further that to the best of our knowledge the work presented in this thesis has not been submitted to any other University or Institute for the award of any Degree or Diploma.

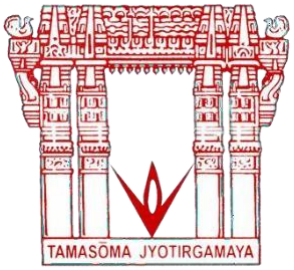
E. Lalitha
Assistant Professor
CSE-CYS

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DEPARTMENT OF CSE - CYS



DECLARATION

We declare that the major project work entitled "**Rock Paper Scissors**" submitted in the department of CSE(AIML), Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **CSE(CYS)** is a bonafide record of our own work carried out under the supervision of **E.LALITHA, Assistant Professor**. Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Place: Hyderabad

T.Krishna Sowmya (21071A6259)

T.Anshu Karan (21071A6258)

ABSTRACT

You may have played rock paper scissors before. Maybe you've used it to decide who pays for dinner or who gets first choice of players for a team.

If you're unfamiliar, rock paper scissors is hand game for two or more players. Participants say "rock, paper, scissors" and then simultaneously form their hands into the shape of a rock (a fist), a piece of paper (palm facing downward), or a pair of scissors (two fingers extended). The rules are straightforward:

- **Rock** smashes scissors.
- **Paper** covers rock.
- **Scissors** cut paper.

Now that you have the rules down, you can start thinking about how they might translate to Python code.

Table of Contents Page No

1. INTRODUCTION

Rock-paper-scissors is a classic hand game that has been played for generations. The game is simple: two players each make a fist with one hand, then simultaneously reveal their choice of rock, paper, or scissors by extending their fingers.

Rock beats scissors (by breaking them), scissors beat paper (by cutting it), and paper beats rock (by covering it). If both players make the same choice, it's a tie.

In recent years, the game has become popular as a simple programming exercise to create a computer opponent. The goal of the project is to create a program that can play rock-paper-scissors against a human opponent.

The program will take input from the user, either "rock", "paper", or "scissors", and then generate a random choice of its own. The program will then compare the choices and declare a winner or a tie.

The project can be extended in various ways, such as adding a scoring system, creating a graphical user interface, or adding more complex AI to the computer opponent. Overall, the rock-paper-scissors project is a fun and engaging way to practice programming skills and explore basic game theory concepts.

2.2 SYSTEM STUDY

Requirements gathering: Identify the project's requirements by speaking with the stakeholders and reviewing the project brief. This step involves understanding the project's scope, goals, and limitations.

Design: Develop a design for the project, including the user interface, data structures, and algorithms. In this step, you would determine how the game would work, how players would select their moves, how the computer would select its moves, and how the game's logic would be implemented.

Implementation: Develop the project according to the design, using the Python programming language. This step involves writing the code for the game, testing it to ensure it works correctly, and debugging any errors.

Testing: Test the game to ensure that it meets the project requirements and works as expected. This step involves playing the game multiple times, trying different moves, and testing edge cases to ensure that the game is robust.

Documentation: Write documentation for the project, including a user manual, technical documentation, and any other required documentation. This step is essential to ensure that the project can be maintained and improved upon in the future.

Deployment: Deploy the project, making it available for use by end-users. This step involves packaging the project, distributing it to the appropriate platforms, and making it available for download or online play.

Maintenance: Provide ongoing maintenance and support for the project, including bug fixes, updates, and user support. This step ensures that the project remains functional and useful over time.

Overall, a system study on rock paper scissors for a Python course project would involve a comprehensive approach that covers all aspects of the project's development, from requirements gathering to maintenance and support. By following these steps, you can develop a high-quality, functional game that meets the needs of the project's stakeholders and users.

3.DESIGN

3.1. REQUIREMENT SPECIFICATION

SOFTWARE DESCRIPTION:

IDLE Python(3.11 64-bit)

HARDWARE DESCRIPTION:

OS: Windows 11
Processor: AMD Ryzen 7

USE CASE DIAGRAM:

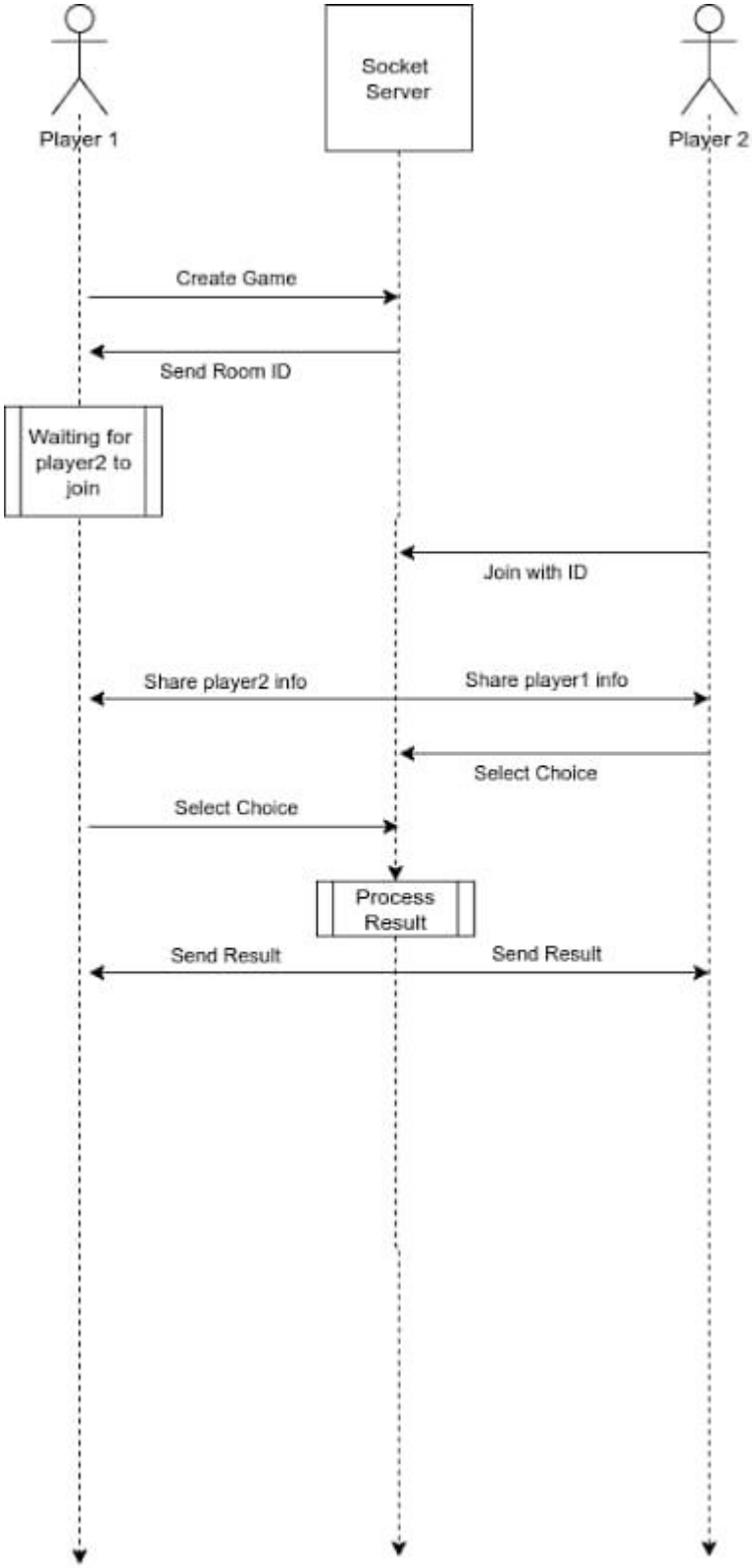


Figure 3.5 Usecase Diagram

3.2.1 SEQUENCE DIAGRAM:

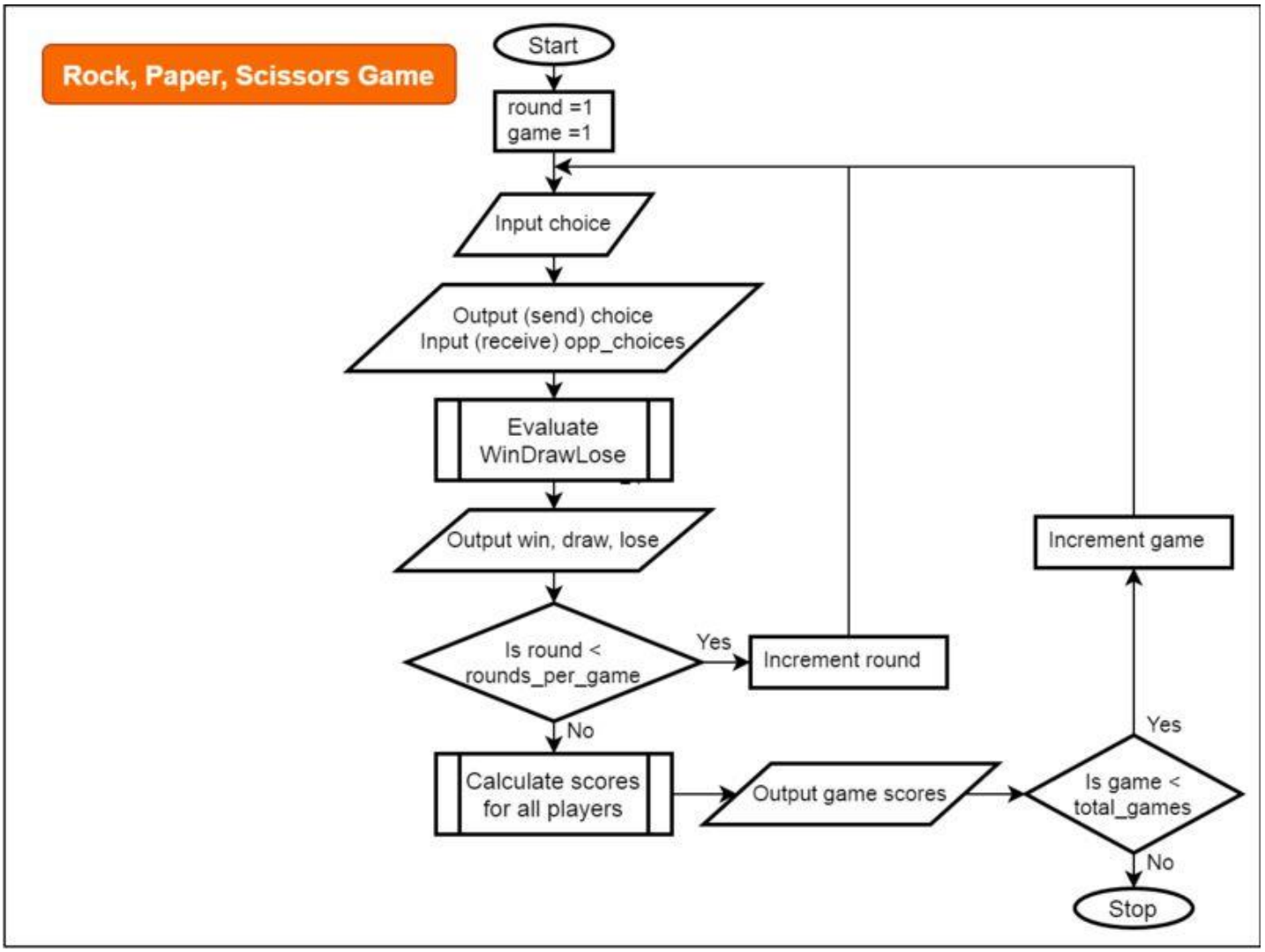


Figure 3.7 Sequence Diagram

3.2.2 ACTIVITY DIAGRAM:

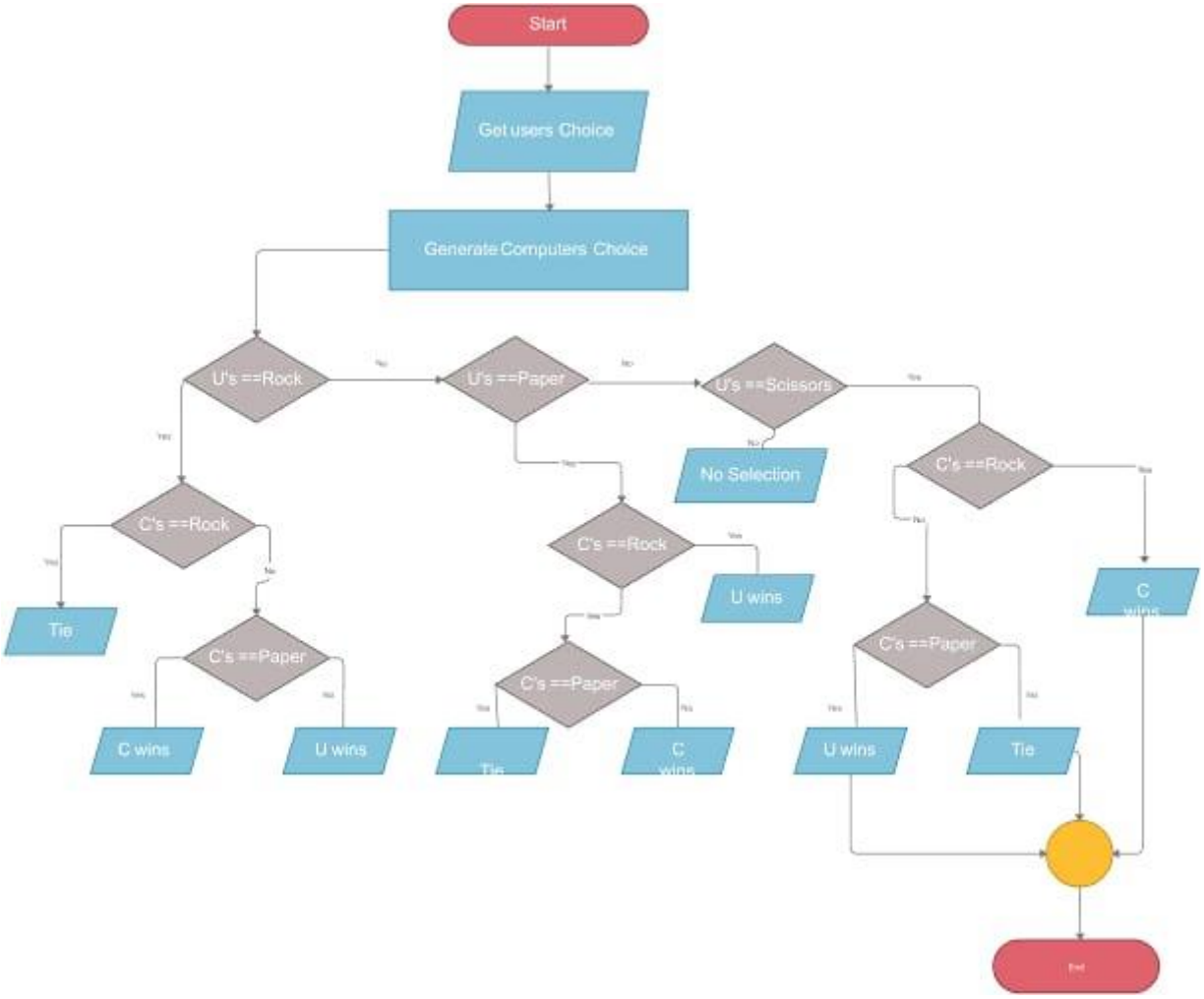


Figure 3.8 Activity Diagram

Figure 3.5 shows the Use Case Diagram use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors

Figure 3.6 shows the Class Diagram which is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Figure 3.7 shows the Sequence Diagram illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

Figure 3.8 shows the Activity Diagram that provides a view of the behavior of a system by describing the sequence of actions in a process.

4.IMPLEMENTATION

```
import random
choices = ["Rock", "Paper", "Scissors"]
computer = random.choice(choices)
player = False
cpu_score = 0
player_score = 0
while True:
    player = input("Rock, Paper or  Scissors?").capitalize()
    ## Conditions of Rock,Paper and Scissors
    if player == computer:
        print("Tie!")
    elif player == "Rock":
        if computer == "Paper":
            print("You lose!", computer, "covers", player)
            cpu_score+=1
        else:
            print("You win!", player, "smashes", computer)
            player_score+=1
    elif player == "Paper":
        if computer == "Scissors":
            print("You lose!", computer, "cut", player)
            cpu_score+=1
        else:
            print("You win!", player, "covers", computer)
            player_score+=1
    elif player == "Scissors":
        if computer == "Rock":
            print("You lose...", computer, "smashes", player)
            cpu_score+=1
        else:
            print("You win!", player, "cut", computer)
            player_score+=1
    elif player=='End':
        print("Final Scores:")
        print(f"CPU:{cpu_score}")
        print(f"Player:{player_score}")
        break
```

5. TESTING

TEST CASES:

1. Test Case 1: Player 1 chooses Rock and Player 2 chooses Paper
   Expected Output: Player 2 wins
2. Test Case2: Player 1 chooses Rock and Player 2 chooses Scissors
   Expected Output: Player 1 wins
3. Test Case 3: Player 1 chooses Scissors and Player 2 chooses Paper
   Expected Output: Player 1 wins
4. Test Case 4: Player 1 chooses Scissors and Player 2 chooses Rock
   Expected Output: Player 2 wins
5. Test Case 5: Player 1 chooses Paper and Player 2 chooses Rock
   Expected Output: Player 1 wins
6. Test Case 6: Player 1 chooses Paper and Player 2 chooses Scissors
   Expected Output: Player 2 wins
7. Test Case 7: Player 1 and Player 2 choose the same option (Rock, Paper, or Scissors)
   Expected Output: It's a tie

6.RESULTS

```
Rock, Paper or  Scissors?paper
You lose! Scissors cut Paper
Rock, Paper or  Scissors?rock
You win! Rock smashes Scissors
Rock, Paper or  Scissors?scissors
Tie!
Rock, Paper or  Scissors?end
Final Scores:
CPU:1
Player:1
```

7.CONCLUSION

Rock-paper-scissors is a popular hand game played by two people. It is a simple game with three possible outcomes: rock beats scissors, scissors beats paper, and paper beats rock. The game is often used to settle disputes or make decisions in a fun and lighthearted way.

While the game is simple, there are some strategies that can increase your chances of winning. For example, some players may try to predict their opponent's next move based on patterns or tendencies they've observed, while others may use bluffing or mind games to try to throw off their opponent.

Overall, rock-paper-scissors is a fun and easy-to-learn game that can be enjoyed by people of all ages. It is a great way to pass the time and can even be used to improve decision-making skills and strategic thinking.

8.FUTURE SCOPE

The future scope of a Rock-paper-scissors project can be summarized as follows:

Multiplayer functionality: Adding the ability to play against other players in real-time can greatly enhance the user experience.

AI opponents: Creating AI opponents that can adapt to the user's playing style can provide a more challenging experience.

Customization options: Adding customization options can make the game more engaging and immersive for users.

Mobile optimization: Optimizing the game for mobile devices can greatly increase its reach and user base.

Gamification features: Adding gamification features can motivate users to continue playing and improve their skills.

Integration with other applications: Integrating the game with other applications can provide new opportunities for engagement and user acquisition.

REFERENCES(If any):

Bibliography:

[1] “Python for everybody”, Charles R. Severance
[2] “Think Python”, Allen B. Downey
[3] ”Python programming”, Mark lutz