

ADT & Algorithms

Games

Dr. NOURI Nabil

October 12, 2024

1. Exercise 1: Number Guessing Game

1.1 Objective:

Implement a number guessing game.

1.2 Instructions:

- Write a program that generates a random number between 1 and 100.
- Prompt the user to guess the number.
- Provide feedback if the guess is too high, too low, or correct.
- Allow the user to continue guessing until they find the correct number.
- Keep track of the number of attempts and display it at the end.

2. Exercise 2: Rock, Paper, Scissors

2.1 Objective:

Create a rock-paper-scissors game against the computer.

2.2 Instructions:

- Write a program that asks the user to enter "rock," "paper," or "scissors."
- Generate a random choice for the computer.
- Compare the user's choice with the computer's choice and determine the winner.
- Display the results and ask if the user wants to play again.

3. Exercise 3: Tic-Tac-Toe

3.1 Objective:

Build a two-player tic-tac-toe game.

3.2 Instructions:

- Create a 3x3 grid to represent the board.
- Allow Player 1 to choose "X" and Player 2 to choose "O."
- Prompt each player to enter their move by specifying row and column.
- After each move, check for a winner or a draw.
- Display the final board state and the result of the game.

4. Exercise 4: Hangman

4.1 Objective:

Develop a hangman game.

4.2 Instructions:

- Choose a random word from a predefined list.
- Display underscores for each letter in the word.
- Allow the user to guess one letter at a time.
- If the letter is correct, reveal its position(s); if incorrect, decrease the number of attempts.
- Continue until the user guesses the word or runs out of attempts.

5. Exercise 5: Guess the Word

5.1 Objective:

Implement a word guessing game.

5.2 Instructions:

- Select a random word from a predefined list.
- Prompt the user to guess the entire word.
- If the guess is correct, congratulate the user.
- If incorrect, allow the user a limited number of attempts to guess the word.

6. Exercise 6: Number Sorting Game

6.1 Objective:

Create a game that involves sorting numbers.

6.2 Instructions:

- Generate a list of random numbers.
- Prompt the user to sort the numbers in ascending order.
- Allow the user to input their sorted list.
- Compare the user's sorted list with the correct sorted list.
- Provide feedback on how many numbers were correctly placed and how many were incorrect.