3/23/2025

OS Lab Project

Dalgona Challenge Project (Shell Script)

(Shell Script)

Created by

Bilal Ansari (23F-3033)  
 Ahad Khan (23F-3074)

**OS Lab Project**

**(Shell Script)**

# **Code:**

#!/bin/bash

# Global Variables for managing game

declare -a Shapes=("Circle" "Star" "Umbrella" "Triangle")

Player\_Score=0

Computer\_Score=0

LeaderBoardFile="leaderboard.txt"

Introduction='

||============================================================================================||

||                           Dalgona Challenge - Shell Script Game                            ||

||                 Group Members : Ahad Khan (23F-3074), Bilal Ansari (23F-3033)              ||

||       Features : Single Player, VS Computer, Multiplayer, Difficulty Levels, Leaderboard   ||

||============================================================================================||'

Timer=8  # Default for Easy mode game

# ASCII Art Definitions

declare -A ASCII\_ART

ASCII\_ART[Circle]='

        o  o

     o        o

    o          o

    o          o

     o        o

        o  o

'

ASCII\_ART[Star]='

     \_\_/\\_\_

     \    /

     /\_  \_\

       \/

'

ASCII\_ART[Umbrella]='

        \_/\\_

      \_/    \\_

    \_/        \\_

   /\_\_\_\_\_\_\_\_\_\_\_\_\

         |

         |

         |

       |\_|

'

ASCII\_ART[Triangle]='

        \*

       \* \*

      \*   \*

     \*     \*

    \*\*\*\*\*\*\*\*\*

'

Display\_Introduction()

{

    local text=$1

    local delay=$2

    for (( i=0; i<${#text}; i++ )); do

        echo -n "${text:$i:1}"

        sleep $delay

    done

    echo

}

# Function: SelectDifficulty & user able to choose the game difficulty (timer)

SelectDifficulty()

{

    cat<<MNR

                    ================================

                    #   Choose Difficulty Level    #

                    ================================

                    ||     1. Easy (8 sec.)       ||

                    ||     2. Medium (6 sec.)     ||

                    ||     3. Hard (4 sec.)       ||

                    ================================

MNR

    read -p "-=> Your Choice : " choice

    case $choice in

        1) Timer=8 ;;

        2) Timer=6 ;;

        3) Timer=4 ;;

        \*) echo "Invalid choice, defaulting to Easy !" ;;

    esac

}

# Function : PrintingASCII\_Shape &it display ASCII art for a given shape.

PrintingASCII\_Shape()

{

    echo "Here is your challenge shape : "

    echo "${ASCII\_ART[$1]}"

}

# Function : SelectRandomShape & randomly select a shape from the predefined list

SelectRandomShape()

{

    local index=$((RANDOM % 4))

    echo "${Shapes[$index]}"

}

# Function: SinglePlayerMode & user play with dynamic timer and attempts.

SinglePlayerMode()

{

    read -p "Enter Your Name : " PlayerName

    local shape=$(SelectRandomShape)

    PrintingASCII\_Shape "$shape"

    local attempts=3

    local guessed=0

    while [[ $attempts -gt 0 && $guessed -eq 0 ]]; do

        echo -e "\n$PlayerName, Enter the shape name (Attempts left : $attempts) : "

        read -t $Timer -p "> " guess

        if [[ $? -ne 0 ]]; then

            echo -e "\nTime's up !"

            break

        fi

        if [[ "${guess,,}" == "${shape,,}" ]]; then

            echo "Correct! You earned 10 points !"

            Player\_Score=$((Player\_Score + 10))

            guessed=1

        else

            echo "Wrong guess !"

            attempts=$((attempts - 1))

        fi

    done

    if [[ $guessed -eq 0 ]]; then

        echo -e "\nThe Correct Answer is : $shape"

    fi

    # Save score to leaderboard

    UpdatedBoard "Single Player" "$PlayerName" "$Player\_Score"

}

# Function: ComputerVS\_Mode & Player play game with Computer with dynamic timer

ComputerVS\_Mode()

{

    read -p "Enter Your Name : " PlayerName

    local shape=$(SelectRandomShape)

    PrintingASCII\_Shape "$shape"

    # Player's turn

    echo -e "\n$PlayerName's turn ($Timer seconds) : "

    read -t $Timer -p "> " player\_guess

    if [[ $? -ne 0 ]]; then

        echo -e "\nTime's up !"

        player\_guess=""

    fi

    if [[ "${player\_guess,,}" == "${shape,,}" ]]; then

        echo "Correct! $PlayerName earns 10 points !"

        Player\_Score=$((Player\_Score + 10))

    else

        echo "Your Guess incorrect !"

    fi

    # Computer's turn

    local ComputerGuess=${Shapes[$((RANDOM % 4))]}

    echo -e "\nComputer guess : $ComputerGuess"

    if [[ "$ComputerGuess" == "$shape" ]]; then

        echo "Computer earns 10 points !"

        Computer\_Score=$((Computer\_Score + 10))

    else

        echo "Computer failed !"

    fi

    echo -e "\nCorrect Answer : $shape"

    # Save scores to leaderboard

    UpdatedBoard "VS Computer" "$PlayerName" "$Player\_Score" "" "" "$Computer\_Score"

}

# Function : MultiplayerMode & two players play

MultiplayerMode()

{

    read -p "Enter Player 1's Name : " Player1Name

    read -p "Enter Player 2's Name : " Player2Name

    local Player1Score=0

    local Player2Score=0

    local Round=1

    local PlayAgain="y"

    while [[ "$PlayAgain" == "y" ]]; do

        local shape=$(SelectRandomShape)

        echo -e "\n=== Round $Round ==="

        PrintingASCII\_Shape "$shape"

        # Player 1's turn

        echo -e "\n$Player1Name's turn ($Timer seconds) : "

        read -t $Timer -p "> " guess1

        if [[ $? -ne 0 ]]; then

            echo -e "\nTime's up !"

            guess1=""

        fi

        # Player 2's turn

        echo -e "\n$Player2Name's turn ($Timer seconds) : "

        read -t $Timer -p "> " guess2

        if [[ $? -ne 0 ]]; then

            echo -e "\nTime up !"

            guess2=""

        fi

        # Check guesses

        if [[ "${guess1,,}" == "${shape,,}" ]]; then

            echo "$Player1Name guessed correctly & get 10 points !"

            Player1Score=$((Player1Score + 10))

        else

            echo "$Player1Name's guess was incorrect !"

        fi

        if [[ "${guess2,,}" == "${shape,,}" ]]; then

            echo "$Player2Name guessed correctly & get 10 points !"

            Player2Score=$((Player2Score + 10))

        else

            echo "$Player2Name's guess was incorrect !"

        fi

        echo -e "\nCorrect answer : $shape"

        echo "Scores : $Player1Name -=> $Player1Score & $Player2Name -=> $Player2Score !"

        read -p "Play another Round ? (y/n) : " PlayAgain

        Round=$((Round + 1))

    done

    echo -e "\n  \*\*\* Final Scores \*\*\*   "

    echo "$Player1Name: $Player1Score"

    echo "$Player2Name: $Player2Score"

    # Save scores to leaderboard

    UpdatedBoard "Multiplayer" "$Player1Name" "$Player1Score" "$Player2Name" "$Player2Score"

}

# Function: UpdatedBoard & append the player's score to the leaderboard file.

UpdatedBoard()

{

    local ModeType=$1

    local PlayerName=$2

    local PlayerScore=$3

    local Player2Name=$4

    local Player2Score=$5

    local ComputerScore=$6

    echo "=================================================" >> "$LeaderBoardFile"

    echo "Mode : $ModeType" >> "$LeaderBoardFile"

    if [[ "$ModeType" == "Multiplayer" ]]; then

        echo "Player Name : $PlayerName & Score : $PlayerScore" >> "$LeaderBoardFile"

        echo "Player Name : $Player2Name & Score : $Player2Score" >> "$LeaderBoardFile"

    elif [[ "$ModeType" == "VS Computer" ]]; then

        echo "Player Name : $PlayerName & Score : $PlayerScore" >> "$LeaderBoardFile"

        echo "Computer Score : $ComputerScore" >> "$LeaderBoardFile"

    else

        echo "Player Name : $PlayerName & Score : $PlayerScore" >> "$LeaderBoardFile"

    fi

    echo "Date : $(date '+%Y-%m-%d %H:%M:%S')" >> "$LeaderBoardFile"

    echo "=================================================" >> "$LeaderBoardFile"

    echo "Score saved to leaderboard !"

}

# Function: MainMenu , & Display the main menu and handle user input.

MainMenu()

{

    while true; do

        cat<<Printing

        =================================================

        || \*\*\*\* Welcome to the Dalgona Challenge \*\*\*\*  ||

        ||        1. Change Difficulty Level           ||

        ||        2. Single Player Mode                ||

        ||        3. VS Computer Mode                  ||

        ||        4. Multiplayer Mode                  ||

        ||        5. View Leaderboard                  ||

        ||        6. Exit                              ||

        =================================================

Printing

        read -p "   -=> Your Choice : " choice

        case $choice in

            1) SelectDifficulty ;;

            2) SinglePlayerMode ;;

            3) ComputerVS\_Mode ;;

            4) MultiplayerMode ;;

            5) [[ -f "$LeaderBoardFile" ]] && cat "$LeaderBoardFile" || echo "No scores yet !" ;;

            6) echo "Good Bye !"

               echo "See You next Time !"

               exit 0 ;;

            \*) echo "Invalid option !" ;;

        esac

    done

}

# Start the Game

Display\_Introduction "$Introduction" 0.001

SelectDifficulty

MainMenu

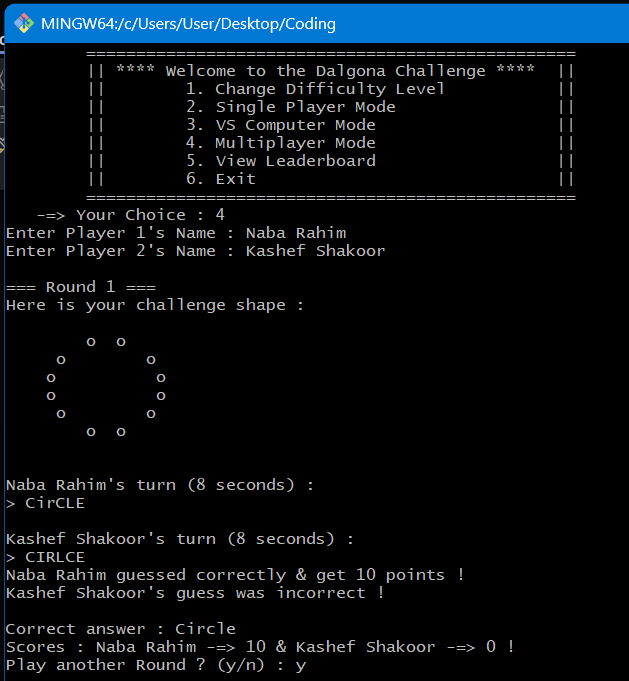
# **Screenshots:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

