Shri Govindram Seksaria Institute of technology Indore (M.P.)

Prashik Borade

29 October 2023

Programming Practices : Mini Project(Jarvis)

https://github.com/PrashikBorade/miniproject

Objectives

1. Tic Tac Toe is a 2 player game where each player has a symbol (either X or O) and plays alternately to mark their symbol on a 3×3 grid.If any player gets their(X or 0) symbol consecutively 3 times in a row, column or diagonal then that player is the winner. So, in total, we have 8 winning conditions: 3 for the rows, 3 for the columns, and 2 for the diagonals. Isn't it interesting!

Technology Used

C

Purpose

The purpose of a Tic-Tac-Toe game program is to provide a fun and engaging gaming experience. It offers entertainment to players of all ages, allowing them to enjoy a classic and simple game. Beyond entertainment, Tic-Tac-Toe encourages strategic thinking and critical decision-making, as players aim to either win or block their opponent from winning. This game also has educational value, making it a useful tool for teaching basic concepts of strategy and game theory. Furthermore, it promotes social interaction, enabling friends and family to come together and enjoy a friendly competition. Overall, a Tic-Tac-Toe game program serves as a delightful pastime that combines entertainment, education, and social interaction.

Scope

(a) The core functionality of the program should include the ability for two players to take turns, placing their respective symbols (X and O) on a 3x3 grid. The program should also have rules in place to determine when a player has won or when the game ends in a draw.

(b) The program can have a simple text-based interface or a graphical user interface (GUI) with visuals and interactive elements. The complexity of the user interface will depend on the design and user experience goals.

(c) Decide whether the game will be for two human players, or if it will include options for single-player mode against an AI opponent. Developing a basic AI for single-player mode is a common feature.

(d) If including AI opponents, you might want to provide different difficulty levels to cater to players of varying skill levels.

(e) Consider whether the game should be limited to a single device or if it should support networked multiplayer, allowing players to compete with others online.

(f) You can choose to add features that allow players to customize the appearance of the game (e.g., themes, symbols, colors) or even the grid size, creating variations of the classic game.

(g) Implement a feature that keeps track of player statistics, such as wins, losses, and draws, as well as a history of previous games.

(h) Depending on the complexity of the program, you can include sound effects and animations to enhance the gaming experience.

(i) Make the program accessible to a wide range of users, considering features such as text-to-speech, keyboard shortcuts, and adjustable font sizes and colors.

(j) Decide whether the program will run on specific platforms (e.g., Windows, macOS, or a web application) or if it should be cross-platform and accessible on various devices.

(k) Provide clear and user-friendly documentation or help features to guide players on how to use the program.

(l) Ensure that the game is thoroughly tested to eliminate bugs and issues, making it a smooth and enjoyable experience for players.

(m) Collecting data on user behavior, sales, and other key performance indicators for analysis and reporting.

(n) Consider the potential for future updates and modifications, which might include adding more features or integrating with other technologies or services.

Output

```
Do you wish to play tick-tack-toe?
yes
Here is your playing board. Player One is Os and Player Two is Xs
Entering a number 1-9 (then pushing enter) as shown below will use
the current Player's turn in that location.


 1 | 2 | 3
-----------
 4 | 5 | 6
-----------
 7 | 8 | 9
1
O | 2 | 3
-----------
 4 | 5 | 6
-----------
 7 | 8 | 9
2
O | X | 3
-----------
 4 | 5 | 6
-----------
 7 | 8 | 9
```

Debugging

```c
#include <stdio.h>
#include <stdlib.h>

typedef int bool;
#define true 1
#define false 0

{
    playGame();
    return 0;
}
int playGame()
{

    char scanned[3];
    printf("Do you wish to play tick-tack-toe?\n");
    scanf("%s", scanned);
    if(strcasecmp(scanned,"yes")==0)
        startGame();

    else
    {
        if (strcmp(scanned,"no")==0 || strcmp(scanned,"nah")==0 || strcmp
            (scanned,"naw")==0)
```

```
ERROR!
gcc /tmp/ZeylJOtfFT.c -lm
/tmp/ZeylJOtfFT.c:8:1: error: expected identifier or '(' before '{' token
    8 | {
      | ^
/tmp/ZeylJOtfFT.c: In function 'playGame':
/tmp/ZeylJOtfFT.c:18:8: warning: implicit declaration of function 'strcasecmp' [
    -Wimplicit-function-declaration]
   18 |     if(strcasecmp(scanned,"yes")==0)
      |        ^~~~~~~~~~
/tmp/ZeylJOtfFT.c:19:9: warning: implicit declaration of function 'startGame' [
    -Wimplicit-function-declaration]
   19 |         startGame();
      |         ^~~~~~~~~
/tmp/ZeylJOtfFT.c:23:13: warning: implicit declaration of function 'strcmp' [
    -Wimplicit-function-declaration]
   23 |         if (strcmp(scanned,"no")==0 || strcmp(scanned,"nah")==0 || strcmp
      |             (scanned,"naw")==0)
      |             ^~~~~~
/tmp/ZeylJOtfFT.c:3:1: note: include '<string.h>' or provide a declaration of
    'strcmp'2 | #include <stdlib.h>
+++ |+#include <string.h>
    3 |
```

```c
#include <stdio.h>
#include <stdlib.h>

typedef int bool;
#define true 1
#define false 0

int main()
{
    playGame();
    return 0;
}
int playGame()
{

    char scanned[3];
    printf("Do you wish to play tick-tack-toe?\n");
    scanf("%s", scanned);
    if(strcasecmp(scanned,"yes")==0)
        startGame();

    else
    {
        if (strcmp(scanned,"no")==0 || strcmp(scanned,"nah")==0 || strcmp
            (scanned,"naw")==0)
```

```
/tmp/ZeylJOtfFT.o
Do you wish to play tick-tack-toe?
```

Profiling