Group: CHAR-izard
Members: Ethan Nguyen, Dawson Pham, Zury Perez, Riley Baltes
Class: CMPR 154

Summary of project

This project is a number-guessing game. The project will give the user multiple options on what they wish to do and then move on from there. The game will start off by asking the user to enter their name and then direct them to a menu, with the options being:

1: DISPLAY MY AVAILABLE CREDIT
2: ADD CREDIT TO MY ACCOUNT
3: PLAY THE GUESSING GAME
4: DISPLAY MY STATISTICS
5: TO EXIT

The user is only allowed to enter up to $20 at a time; however, if they win enough games, they are able to earn more than the deposit limit.

The actual gameplay works by asking the user to enter a number from 1 to 10 and then generate a random number. If the numbers match, then the player earns $2 dollars if not, they lose money.

The "DISPLAY MY STATISTICS" option will show the user how many games they won, how much money they lost, etc.

Summary of Code

For the code to work, it must be run in a compiler that is capable of running MASMx86, and that has the Irvine library available.

The code uses functions, compare, and jump features in order to work. The code also saves multiple variables such as the games played, games won, games lost, etc.

The main PROC and PlayGuessingGame PROC are the main engines behind the game, keeping track of the other code and which loops are being used and when.

mainLoop- Holds the code for displaying the main menu.
displayBalance- Shows how much available balance the player currently has.
addCredits- Allows the player to add credits to their account and has safeguards to ensure the user enters an amount from 1 to 20 dollars.
amountExceeded & invalidAmount- Redirects the player if they have entered an invalid amount.
playGame- Generates a random number and compares it to the player-entered number.
InsufficientFunds- Prompts the player to enter more credits in order to play again.
displayStatistics- Displays the player's statistics from saved variables.
exitProgram- Ends and exits the program.
CorrectGuess & WrongGuess- Keep track of how many guesses the player has gotten right or wrong and adjust the variable in '.data' to match.
PlayAgain- Restarts the game if the player enters 'y'. If the player enters 'n,' the player is returned to the main menu.

Output:

```
Please enter your username: Ethan_
```

```
*** CHAR-izard ***

*** MAIN MENU  ***

PLEASE SELECT ONE OF THE FOLLOWING:

   1: DISPLAY MY AVAILABLE CREDIT
   2: ADD CREDIT TO MY ACCOUNT
   3: PLAY THE GUESSING GAME
   4: DISPLAY MY STATISTICS
   5: TO EXIT
```

```
Available Balance: $0
Press Enter to continue...
```

```
*** CHAR-izard ***

*** MAIN MENU  ***

PLEASE SELECT ONE OF THE FOLLOWING:

   1: DISPLAY MY AVAILABLE CREDIT
   2: ADD CREDIT TO MY ACCOUNT
   3: PLAY THE GUESSING GAME
   4: DISPLAY MY STATISTICS
   5: TO EXIT

2

   2: ADD CREDIT TO MY ACCOUNT
Enter the amount to add: 20
```
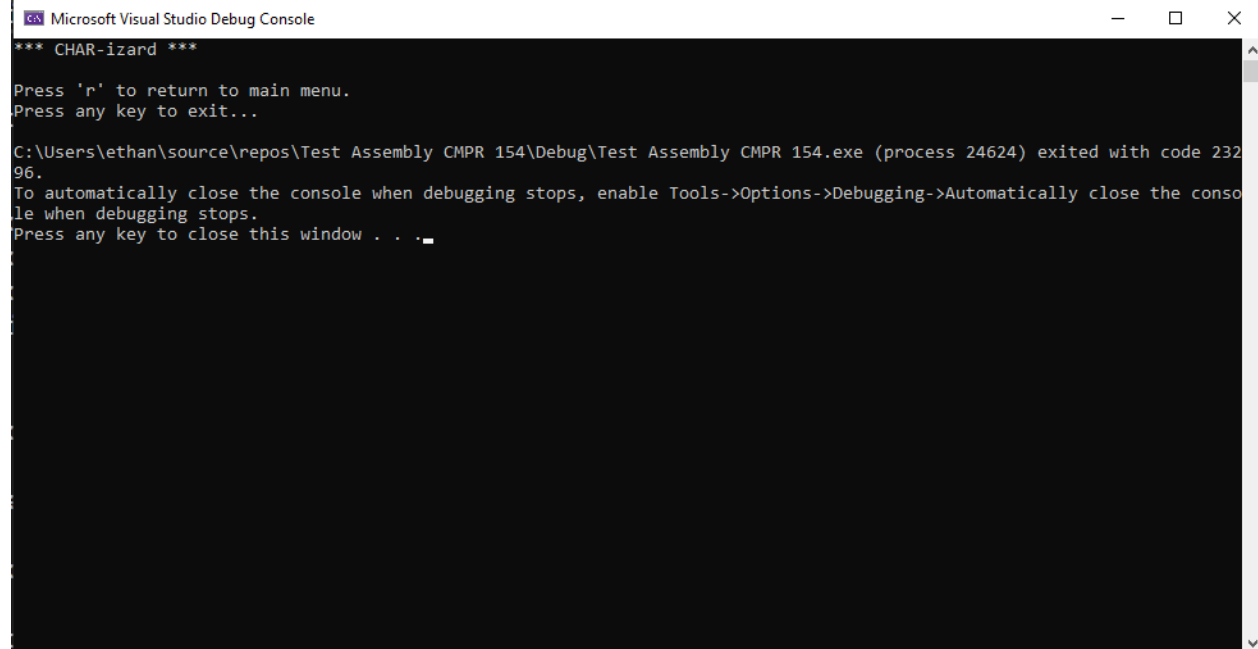
```
Guess a number between 1 and 10: 3
Sorry, wrong guess. The correct number was 6.
Do you want to play again? (y/n):
Invalid input. Please enter either 'y' or 'n'.

Do you want to play again? (y/n):
```

```
Statistics
***************************
Player Name: Ethan
Balance: 18
Games Played: 2
Correct Guesses: 0
Incorrect Guesses: 2
Money Won: $0
Money Lost: -$2

Press Enter to continue...
```

```
*** CHAR-izard ***

Press 'r' to return to main menu.
Press any key to exit...

C:\Users\ethan\source\repos\Test Assembly CMPR 154\Debug\Test Assembly CMPR 154.exe (process 24624) exited with code 232
96.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```