***Programming Fundamentals (CT-175)***

***Project “Super-Space”Report***

***Department: Computer Science and Information Technology***

***Batch: AI (2023-2024)***

**Group Members:**

* **Faizan Khurram (CTAI-014)**
* **Maaz Rehman (CTAI-036)**
* **Umair Hassan (CTAI-018)**

**Problem Statement and Project Title:**

This project focuses on tackling the main problem of the user in which he/she decides to spend some time playing a few small fun games (like the ones in arcades). In this case our project program allows the user to play certain amounts of games, known as minigames, as per the choice of the user. This project idea has been taken directly from real-life arcades, in which a person has to first buy some sort of subscription (or a pass) that lets him/her play the minigames available based on the coins collected by each minigame and the number of coins left to be able to play each minigame. The user is given several choices to perform various actions that are provided at ease for the user, according to the needs of the user.

This is the main idea of our project; **to allow the user to have a choice of three exclusive minigames in the form of an arcade system, where each minigame gives certain rewards based on the type of game being played.**

In this regard, overlooking general names for arcades around the world, we have decided to name our arcade minigame program as ***“SuperSpace”***

**Project Description:**

Our program first focuses on the most important thing, getting the user to subscribe to the arcade by performing some payment. Initially the user is given 2000 coins and the subscription is of 1000 coins. Obviously, this is done according to the choice of the user. After successful subscription and taking of credentials, we welcome the user into the SuperSpace. Every time the user is not playing any minigame, the main menu screen is displayed to them. The standard if-else structure has been used for implementing the complete menu-driven choice selection of the user.

Now, in regards of the three minigames implemented in this arcade program, they are the following:

* **Rock Paper Scissors**; a minigame in which the user plays with the machine and the outcomes are all completely luck-based. The outcome can be either a player win, a machine win or a tie.
* **Hangman**; a minigame in which the user tries to guess a total of 3 words that are pre-determined. Each word gets harder progressively.
* **Casino Jackpot**; a minigame in which the user has to roll 3 slots (of rows) and the outcome is determined by the number of rows matched. The outcome can be either no matches, 2 matches or the Jackpot (all matched rows).

**The Casino minigame is a bit different from the other two minigames because it is initially locked for the user after subscription. (The user must get their wallet balance to 1600 coins or more to unlock Casino).** Each minigame has an **entry fee of 200 coins** that is discarded from the total wallet balance of the user automatically as soon as the user enters a minigame. The wallet balance of the user is also displayed for them during certain segments of each minigame so as to help them on deciding whether or not to keep playing a certain minigame. Other than this, the user has also been provided with some miscellaneous options, that can only be accessed by the user on very rare and specific conditions.

After completing one minigame, the user is automatically directed back to the home screen which displays the complete menu list.

**Contributions of Group Members:**

* Faizan Khurram (CTAI-014) has worked on the minigame Casino Jackpot and has also taken care of the main structure of the entire program by basically creating the structure at first and then the other blocks of code were simply placed and properly formatted into the main program structure. This took care of the majority of the issues.
* Maaz Rehman (CTAI-036) has worked on the minigame Rock Paper Scissors. He has also created a user-defined function that is similar to the print statement, except, it does it with delay. In this regard, to some extent, Maaz has also contributed to the main program structure.
* Umair Hassan (CTAI-018) has worked on the minigame Hangman. A few parts of the main program that also required some finishing touches were also correctly formatted and coded by Umair.

**Functionalities:**

-**Main structure:**

The main structure of this program has been developed using simple if-else as-well as ladder if-else structure. Other than this, infinite while loops have been implemented in the code block of each individual minigame as-well-as in the structure too. While loops can be considered to be one of the most important parts of the main structure of this program since they help significantly with how the overall program logic flows smoothly and efficiently.

-**Separate Header Files:**

Since there are three different minigames implemented in this program, therefore, each individual minigame also has its own set of user-defined functions that promote to more simplicity of use. However, another change that has been implemented is the use of separate header files. In this program, three separate header files named “faizanfunctions.h”, “maazfunctions.h” and “umairfunctions.h” have been used, each of which contains user-defined functions of their respective minigames. This reduces the junk taken up by all the function bodies and prototypes from the main program and directly includes and calls them from separate files. Not only this but standard library (pre-defined header file functions of C) have also been called through the three custom header files that are mentioned above.

**-Use of sleep () and system(“cls”):**

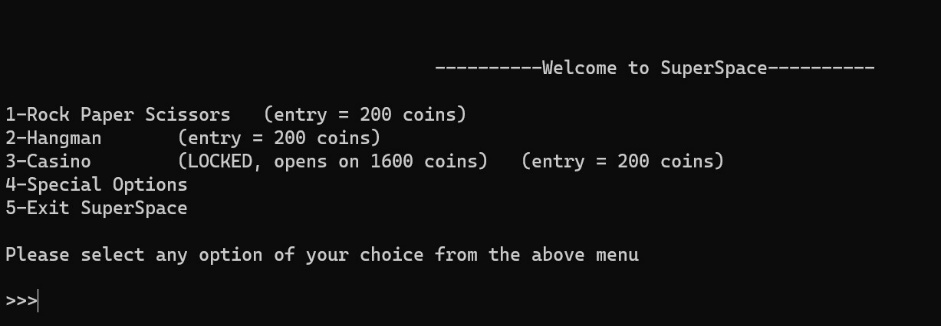
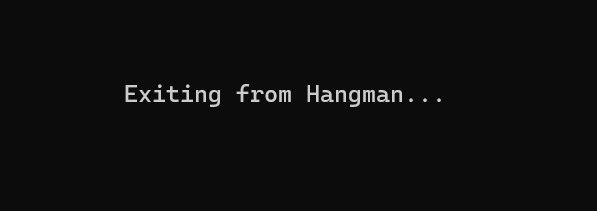
The command system(“cls”) is used to completely clear the screen after getting executed. This has helped in the overall presentation of the program. Likewise, the sleep () function determines the pause time between each individual statement being printed on the output terminal. This has also helped in the better overall presentation of the program.

-**The Random Function:**

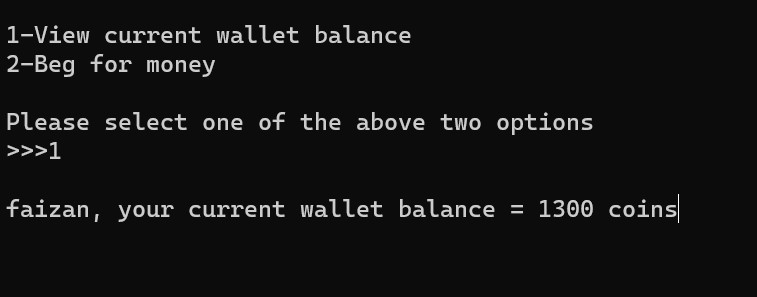
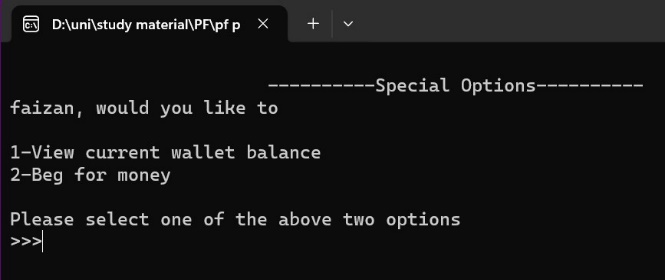
The random function is a very important function of this entire program. It is the main backbone of two minigames, Rock Paper Scissors and Casino Jackpot because both of these minigames have their outcomes based on complete luck. The random function is used to generate a number between a given range, then decide the outcome (out of several outcomes) by assigning each outcome to execute for some part of the range specified.

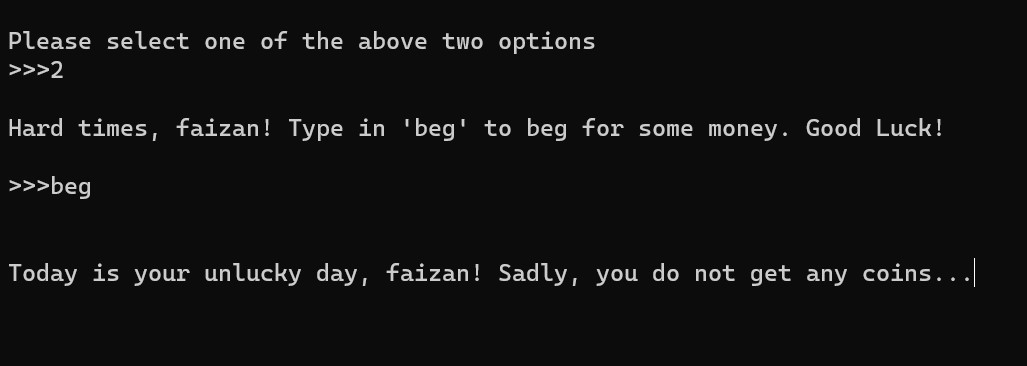
-**Customized Menu, Loading and Exiting Screens:**

 Customized loading and exiting screens have been used that use the system(“cls”) and sleep statements for proper execution. They give the program better presentation and guideline as to what is currently happening in the program, or what minigame the user is being loaded into or being exited from. Other than this, the menu display list has also been customized in such a way that it keeps a track of the current wallet balance of the user (this is specifically implemented to determine whether the user is still eligible for entering Casino or not).



**-Special Option Menu:**

 This is a miscellaneous option that is made available as part of the main menu options. This offers the user two choices; to view their current wallet balance or to “beg” for some money in case of very low wallet balances. In this case, the begging system that provides some amount of money to the user has also been created using the random function and depends on the range criteria for determining whether to give money to the user or not.

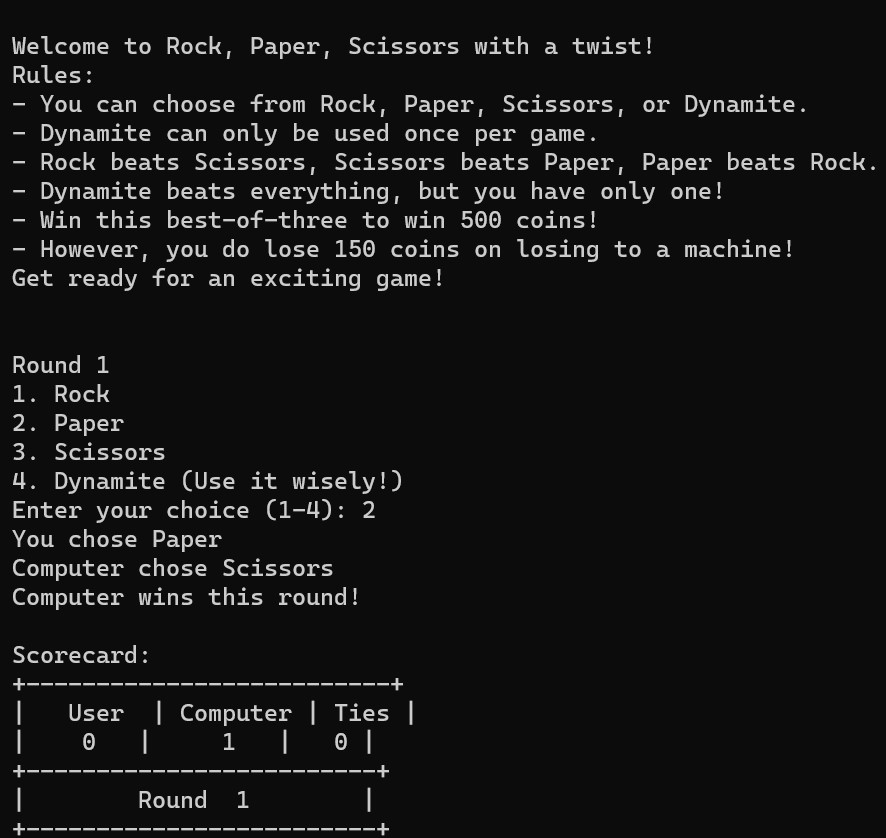


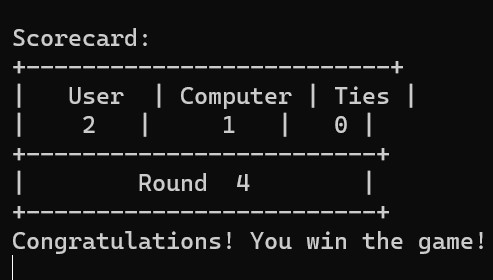
**-Printing with Delay:**

This is a custom-built user-defined function developed by Maaz that prints each individual character of a string on the terminal with some specified delay timing by the user. This function is implemented by using a pointer function. It is used to boost the overall presentation of the main program.

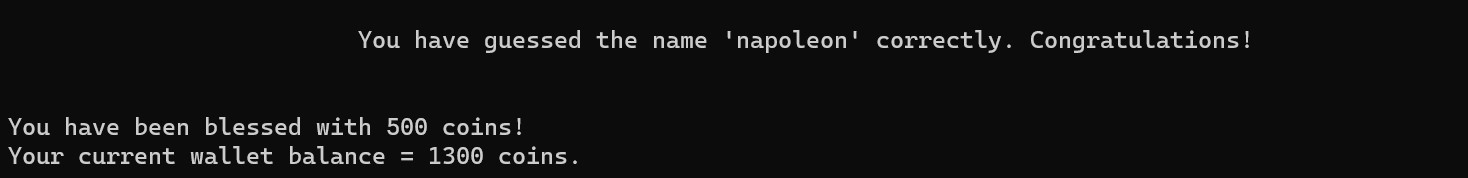
-**Rock Paper Scissors:**

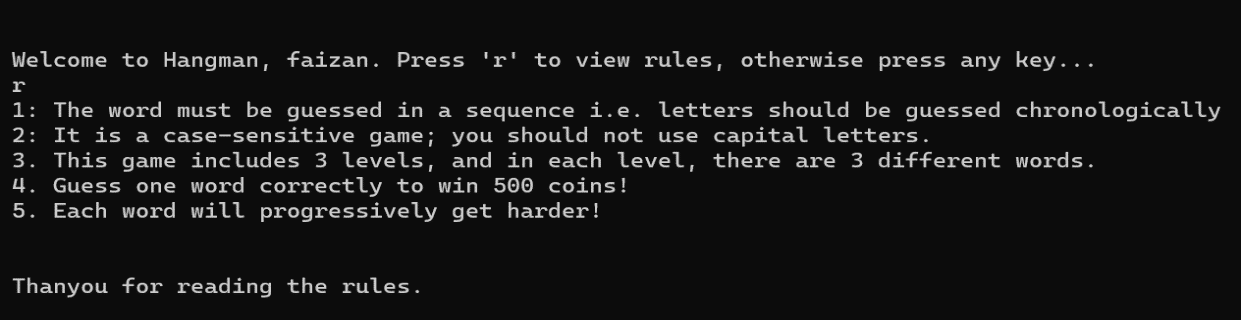
The functionality of this minigame is that Rock is the outcome of the machine if the random number generated for machine is 1, Paper if it is 2 and Scissors if it is 3. The user gets to choose what to throw. There is also an option for dynamite that automatically wins the user the round. The updated scorecard is displayed after each round with the score of user and machine as-well-as round count and number of ties. If the user wins this best-of-three game, then they are rewarded with 500 coins. Otherwise, they lose an additional 150 coins for losing to the machine.



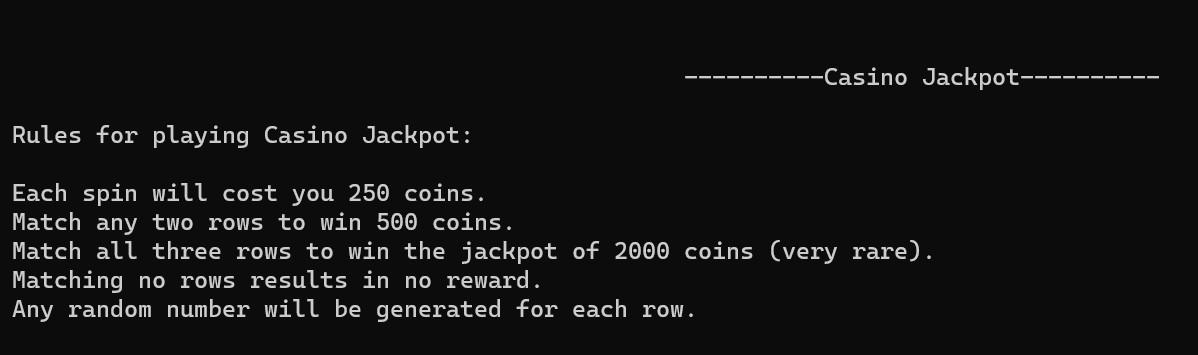


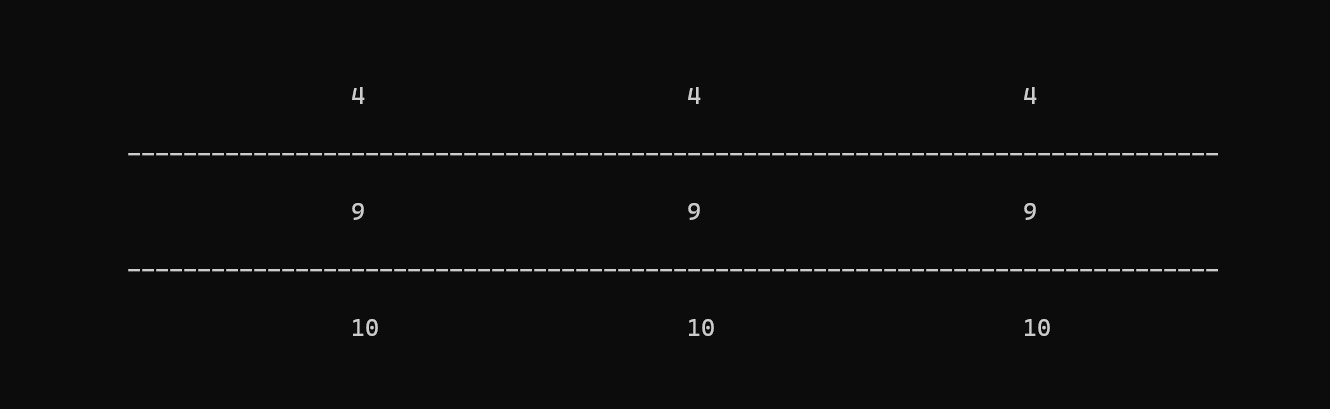
**-Hangman:**

 The functionality of this game is that the user has to guess a total of 3 words that are already determined in the structure of the program. Each word gets progressively harder. The letters of a word have to be guessed in chronological manner. If the user guesses one word correctly, then they are rewarded with 500 coins. Otherwise, they are not rewarded with anything. The program also keeps track on the number of times a letter has been guessed by the user.



**-Casino Jackpot:**

 A simple jackpot game that requires 250 coins from the user for each individual spin. If the user is able to match any two rows, then they are rewarded with 500 coins. If the user is able to match all the three rows, then they hit the jackpot of 2000 coins. Otherwise, the user only loses 250 coins for each spin, which is the spinning fee. The random function serves as the pinnacle of this minigame and its entire logic. The user can also wish to stop and cash out of the casino. In this case, it also keeps a track of the total number of coins that the user has won upon since entering casino.



**Concepts Covered:**

-**Decision Control:**

Decision control that consists of if and else-if conditions (even nested) is considered to be the most fundamental backbone of this program since the entire flow of logic and control of the main structure is implemented through the use of conditional statements.

-**Switch Statement:**

The switch statement is similar to conditional statements. For this program, it has been used in the minigame Rock Paper Scissors for identifying which number the user has pressed for the specified outcome, although if-else could have also been used.

-**Loops:**

Loops have been used in all the parts of this program, including all the three minigames. Specifically, the for and while-loops have been used, with the while loop being mostly implemented for taking care of the overall structure of the program and the for loop has been implemented in majority for each individual minigame.

-**Arrays:**

Arrays have been used in various places throughout this program. For example, storing the name of the user and many more similar array variables that help in completing the structure logic of this program. Two- Dimensional arrays have also been implemented in the hangman minigame. In case of Casino, arrays have been implemented to keep track of each individual row (one row has three numbers i.e. three columns that are all the same random integer). Therefore, a two-dimensional array has also been implemented in Casino Jackpot.

**-Strings:**

Strings provide the finishing touches in the presentation of this program. They are used for storing an important information i.e. the username of the player. Other than this, many string commands included from the header file <string.h> have also been implemented that are very useful for taking care of certain conditions that arise during the logic of the main program structure. For example, a string function has been used in this program that decides whether a particular response entered by the user is valid or invalid.

**-Functions:**

Other than the pre-built library functions of C, a great amount of user-defined functions have been used in this program. This provides better understandability of the program and also provides ease of code reading for the user, preventing coding headaches. User-defined functions have been used for all the three minigames, as-well-as for also handling some features of the main flow of logic in the program. Some examples of functions in this program include a function for displaying exit, loading and menu, a function for generating random numbers for casino and rock paper scissors, a function for displaying the rules of each minigame etc.

-**Pointers:**

Pointers have been implemented in the Casino Minigame. They are used for referencing the address and value of the random jackpot numbers that are to be assigned to each column of the same row. These three pointers are then used in a for loop with the indirection operator for storing the value taken from the addresses of where those generated random numbers were stored (variables).

**Limitations / Future Enhancements:**

* All of the minigames in this program are either single player, or between the player and machine. In this case, if there was any possibility of developing a multiplayer game, or maybe even a minigame involving an even greater number of players, then we could have implemented the use of structures, since different variables of the same related category and attributes would be linked under the same name.
* One limitation was encountered in Casino, that initially it was going to be set on a column system, i.e. deciding the outcome on the basis of the number of columns matched and not the rows. However, during implementation process, it was discovered that this simply not allowed by the C compiler, so the idea was shifted to Casino being a row-based Jackpot system
* Another limitation that was encountered is that the function that prints the letters of a string with some delay, does not work for those control strings that have some form of format specifier in them.