



TEAM NSU CENACE

Play Bytes Restaurant

BITE THE BYTE

Final lab report
CSE 115.4L
Fall 2022



Presented by

- **Aaban Saad Preyanta**
- **Ahmad Fahmid Alvi**
- **Fahim Foysal**

NSU ID

2233389642
2232502642
2233594642

Presented to

- **Shafin Rahman (faculty)**
- **Jannatul Ferdaous (Lab instructor)**

CONTENTS

Play Bytes Restaurant

1.0	Introduction.....	01
2.0	The Restaurant	03
3.0	Games	
3.1	Computer Educable Noughts and Crosses Engine (CENACE)	05
3.2	Number Guessing Game	11
3.3	Rock Paper Scissors	15
4.0	Conclusion.....	19

Play Bytes Restaurant

< BITE THE BYTE >

Introduction

This project is a food order system for a unique restaurant named **Play Bytes Restaurant**. This is an innovative restaurant that brings the best of both worlds together, providing you with any type of food you could want at the most affordable price. Here the customer can order delicious food they love at a very cheap rate. But not only that, every time they order something they will get a special offer to play some simple games. And if they win, they can get up to 15% discount. Sounds interesting, huh? The key here is to deliver benefits to the customers in every possible way.

Discount policy:

In order to give you the best experience in our restaurant we provide a discount policy. Now its your time. It is an honor to bring your kids and friends in our restaurant and offer them some hours of fun. We have three games for our customers. Tic-tac-toe, rock paper scissors and number guess. These three games are divided into two categories. Tic-tac-toe is in Category-1, it is for 10% discount. Rest of the two are in category-2 which for 5% for each game

Functions:

We have a total of 23 functions in this code. Functions used in the code are given below:

These are the restaurant's functions:

- int main()
- int main_page()
- int menu_order()
- void Bill_show(int arID[],int arqty[],int counter);
- void Bill_Payment()
- int discount()

These are CENACE's functions:

- int CENACE()
- void choose_player()
- void inputmove()
- void newboard()
- void printboard()
- int checkwin()
- int space_in_board()
- void updateLearning_data()
- void menu()
- void CENACE_intro()
- void updateGraphing_data()
- void scoreUpdate()

These are Rock Paper Scissors Game's functions:

- void game()
- int rock_paper_scissors()

These are Number Guess Game's functions:

- int number_guess()
- void number_guess_game()
- void load()

The Restaurant

Author: Fahim Foysal

NSU ID: 2233594642

Bachelor of Computer Science and Engineering
Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh

These are the restaurant's functions:

- `int main()`
- `int main_page()`
- `int menu_order()`
- `void Bill_show(int arID[],int arqty[],int counter);`
- `void Bill_Payment()`
- `int discount()`

int main()

This is the main function of Play Bytes Restaurant . It processes input from the `main_page()` function and calls other necessary functions when needed

int main_page()

This function prints the main menu page and takes menu input from the user. Then it returns the menu number to the main function. If the user presses '1' menu arrives on the screen. For paying bill the user have to press '2'. To exit the user can press '3'.

int menu_order()

This function prints the restaurant's food menu and takes order from the customer.

void Bill_show(int arID[],int arqty[],int counter)

This function shows the bill to the user. Then prints a menu on the screen where the user will be given 3 options.

1. to give another order.
2. to go to the main menu.
3. to win some discount.

Here comes the main interest. To get some discount user can play some games by pressing '3'.

int discount()

This function calls the game functions. There are 2 categories. Category 1 is for playing Tic Tac Toe with CENACE. The user has to play at least 5 games and score more than CENACE to win a discount of 10%. Category 2 is for playing Number Guess game and Rock paper scissors game. The user have to choose any one of them and win To get 5% discount.

Void Bill_Payment():

This function is for bill payment. It calculates the bill of the customer including discount. Takes payment from customer and returns changes.

Computer Educable Noughts and Crosses Engine (CENACE)

Author: Aaban Saad Preyanta

NSU ID: 2233389642

Bachelor of Computer Science and Engineering
Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh

What is CENACE?

CENACE: Computer Educable Noughts and Crosses Engine is an artificially intelligent system that can play noughts and crosses (tic-tac-toe) against a human opponent.

Here in the restaurant, the user has to play at least 5 matches and score more than CENACE to get a discount.

How does it work?

CENACE learns to play noughts and crosses (tic-tac-toe) by playing it repeatedly against another player, fine-tuning its technique each time, until after having played a certain number of games it becomes nearly perfect and its opponent can only draw or lose against it. Similar to how a kid learns, the process of learning is called reinforcement learning, which involves being "punished" for losing and "rewarded" for drawing or winning.

History and origin

The name CENACE originated from the 1961 MENACE* reinforcement learning model by Donald Michie. I have programmed CENACE as a computer simulation of Michie's MENACE. Instead of matchboxes, CENACE uses folders and text files to keep track of its moves.

*MENACE: The Matchbox Educable Noughts and Crosses Engine was a mechanical computer made from 304 matchboxes designed and built by artificial intelligence researcher Donald Michie in 1961. It was designed to play human opponents in games of noughts and crosses (tic-tac-toe) by returning a move for any given state of play and to refine its strategy through reinforcement learning. (Wikipedia)

Interface of CENACE

After CENACE has loaded the user will see a menu like this:

1. Auto Train CENACE
2. Play
3. Rules
4. About

5. Exit

→

User can press the corresponding number to go to that menu.

1. Auto Train CENACE

CENACE learns to play by playing hundreds and thousands of matches against the opponent and this takes A LOT OF TIME!

To fast forward this process, in this menu, CENACE can play against its opponent but, this time the opponent is also a computer. No, CENACE will not play against itself, instead, it will play against an already existing tic-tac-toe algorithm. In this case, I have hardcoded an algorithm that I created.

2. Play

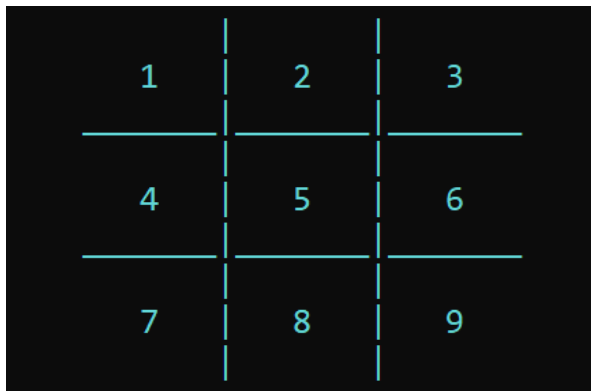
This is the main part where you can play against CENACE.
CENACE will get better and better as you play more.

3. Rules

You can see the rules here.

RULES:

1. CENACE will always go first.
2. Move positions are from 1 to 9 on the game board.



3. +3 for win, +1 for draw, -1 for lose.

4. About

Shows a short description of CENACE.

5. Exit

To exit CENACE

About the source code of CENACE

CENACE is written in C language.

Functions used in the code are:

```
int CENACE() //int main()
void CENACE_intro()
void menu()
void choose_player()
void inputmove()
void updateLearning_data()
void updateGraphing_data()
int checkwin()
void scoreUpdate()
void newboard()
void printboard()
int space_in_board()
```

int CENACE()

This is the main function. It calls other functions and coordinates everything.

It creates the "Learning_data" and the "Graphing_data" directory or folder when CENACE runs for the first time, marks the moves on the game board, and also shows messages and warnings to the user.

Learning_data: This folder will be automatically created and updated after each match and will store the data that CENACE learned.

Graphing_data: This folder will be automatically created and updated after each match and will store the data for drawing a graph of CENACE's score.

void CENACE_intro()

This function is for the loading animation.

void menu()

This function prints the main menu and takes menu input from the user.

void choose_player()

This function is for selecting player signs.

Currently 'O' is fixed for CENACE and 'X' is fixed for its opponent.

And CENACE will always play first.

Maybe I'll add this feature in a future version of CENACE.

void inputmove()

This is the MOST IMPORTANT function of the program.
It handles all the critical tasks and keeps CENACE working.

While in "Play" mode (2), it takes input from the user against CENACE

While in "Auto Train CENACE" mode (1), it takes input from the existing algorithm that I created to play against CENACE.

This is a simple algorithm that only knows how to win and how to block CENACE from winning. The rest of the time it plays randomly.

This function also takes input from CENACE itself. And this is the most critical part. Every time before playing its move, CENACE will look at the game board. the game board is actually a string of length 10 (Including the null character). for example this one "XO XO ". The blank areas on the game board are occupied with spaces. CENACE will take this string, and will make a copy of this and replace the spaces with a '-' sign. So the copied string is "XO--XO---". Now CENACE will try to create a folder named "XO--XO---" inside the "Learning_data" folder. If the folder doesn't exist then it will be created and inside "Learning_data\XO--XO---" some text files will be created with the names of all blank move positions. Move positions are from 1 to 9 (1 will be added to the string's actual element positions). So, in this case, 5 text files will be created as shown below:

3.txt
4.txt
7.txt
8.txt
9.txt

The text files represent the move positions that are available for CENACE. In these text files, there will be written a priority point. Initially, the points will be 2 for every text file. The points will update after each match. CENACE will choose its move based on the points. The higher the point is greater the chance that CENACE will choose that move.

Now let's have a look at how it works. Assume that the text files are created and there is written "2" inside all the text files. There are a total of 5 text files. CENACE will create an array of size 10 (2+2+2+2+2). there will be 2 copies of every available move (Names of the text files). The array will look like this:

{3, 3, 4, 4, 7, 7, 8, 8, 9, 9}

Now, CENACE will choose randomly from this array.

Let's say it picked up 8, so, it will place its move at the 8th position.

The file path for the given move will be "Learning_data\XO--XO---\8.txt". This path will be stored in a 2D character array. Similarly, all the other paths during the match will be saved. So that at the end of a game all the moves that CENACE gave can be tracked.

Here ends the task of "void inputmove()" function.

After a match is over all the files that CENACE used will be updated in the void updateLearning_data() function.

void updateLearning_data()

In the previous section, I explained how CENACE plays its moves.

If you notice you will see that 8 was the winning move for CENACE in that example. CENACE won the game and the match finished. We also have all the file paths of CENACE's moves from the beginning of the game. Now, Every text file in those paths will be opened one after one. and 3 points will be added to the existing point. For example, Learning_data\XO--XO---\8.txt had 2 points previously and now it will be 5. So, next time there will be a higher chance that CENACE will choose 8 if it finds the same game board again.

The reword policy is:

- +3 for winning a game
- +1 for drawing a game
- -1 for losing a game.

The maximum point for a position is 300 and the minimum point is 1.

void updateGraphing_data()

This function updates the graphing data after each match inside the Graphing_data folder.

There are two text files, Last_Score.txt and Score_List.txt inside that folder. The Score_List.txt stores the score history of CENACE. This data can be used to draw a graph which will show the results of CENACE's games against the opponent. Last_Score.txt stores the last updated score. After each match, Last_score.txt will be updated and appended to the Score_List.txt.

The scoring policy is as before:

- +3 for winning a game
- +1 for drawing a game
- -1 for losing a game.

int checkwin()

This function checks for a winning condition either for the human opponent or for CENACE and returns 1 if found. Or, returns -1 if it is a draw.

void scoreUpdate()

At the end of each match, this function updates the score that is shown on top of the game screen while playing.

void newboard()

This function clears the game board and sets it back to the default for starting a new game.

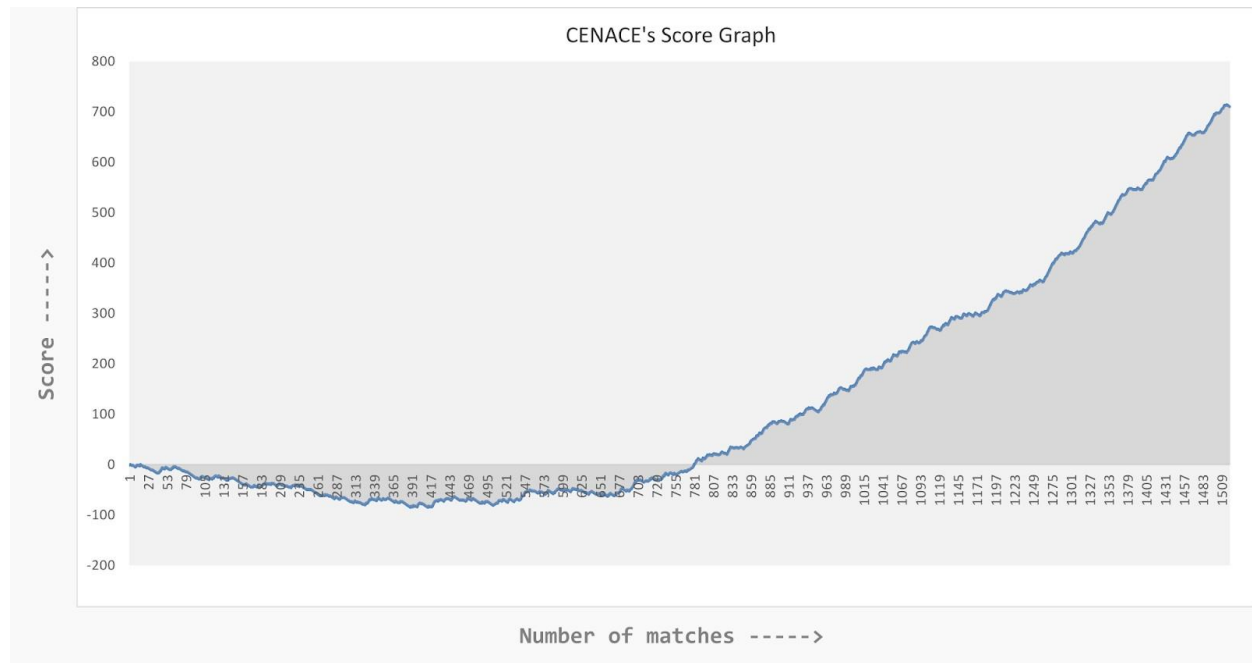
void printboard()

This function prints the current game board and shows the current score above the game board.

int space_in_board()

This function returns the number of blank spaces in the current game board. This information is needed inside the "void inputmove()" function.

Graph



The graph shows the results of CENACE's games against its opponent. +3 for winning, +1 for drawing and -1 for losing.



<https://github.com/Aaban-Saad/Project-CENACE>

Number Guessing Game

Author: Ahmad Fahmid

NSU ID: 2232502642

Bachelor of Computer Science and Engineering
Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh

What is number guessing game?

Number guessing game: an addictive game that's easy to play but hard to win. This game involves a player and a computer, where the object of the game is for the player to guess number between 0 and N in a maximum of N attempts. Before each attempt, he can make use of hints from the computer if he has none (if there are no more hints, it just gives new ones). And the N is decided by computer. If he succeeds in guessing before taking all the attempts, he wins; if not, then he loses.

About the source code of number guessing game:

Number guessing game is written in C language.

Interface of Rock Paper Scissors

After Number guessing game has opened the user will see a menu like this:

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$                                     $
$      Select any of them -         $
$                                     $
$      1. PLAY GAME                 $
$                                     $
$                                     $
$      2. EXIT                      $
$                                     $
$                                     $
$                                     $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

User can press the corresponding number to go to that menu.

1. Play

This is the main part where you can play Number guessing game.

2. Exit

To exit Number guessing game.

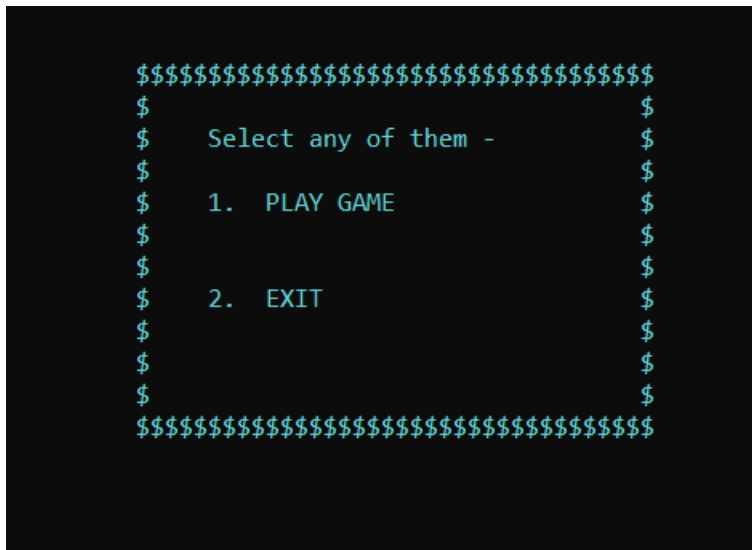
Functions used in the codes are:

- `int number_guess();`
- `void number_guess_game();`
- `void load();`

`Int number_guess();`

This function is the main function of the Number guessing game.

In this game `number_guess_game_score` counts the score of the full game. If user wins then 1 number added to score and if not it remains and if user wins the game user gets 5% discount. Then the interference of the game is printed in this function.



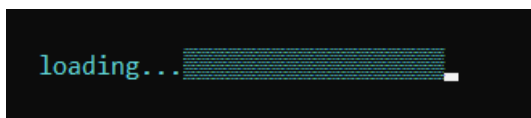
```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$                                                                    $
$      Select any of them -                                          $
$                                                                    $
$      1.  PLAY GAME                                                $
$                                                                    $
$      2.  EXIT                                                      $
$                                                                    $
$                                                                    $
$                                                                    $
$                                                                    $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  
```

And according to the instructions if user press those number , user can go to that menu.As the other to functions `void number_guess_game();` and `void load();` is called in this function.If user selects 1 then user can play the game if user selected 2 then exits from the game.

`void load();`

This function is to make a loading graphics.



```

loading...#####_
  
```

`Void number_guess_game();`

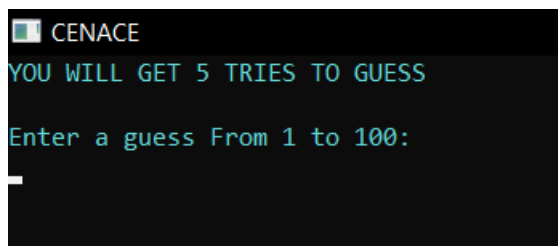
- `number_guess_game_score`; is to store the score.
- `srand(time(NULL))` is used to seed and generate random numbers with time.

The game is played 5 times in a row and If user guess the number 1 number added or remains the same score and from that score the winner is selected.

- `int number=rand() % 100 +1;`

This will give me random numbers from 0 to infinity. So to get 1 2 3 ,It will mod that integer with 3 and it will give me 0,1,2 and to get 1,2,3 there is a plus 1 at the end.

As the game is loaded this interference comes



```

CENACE
YOU WILL GET 5 TRIES TO GUESS

Enter a guess From 1 to 100:
_

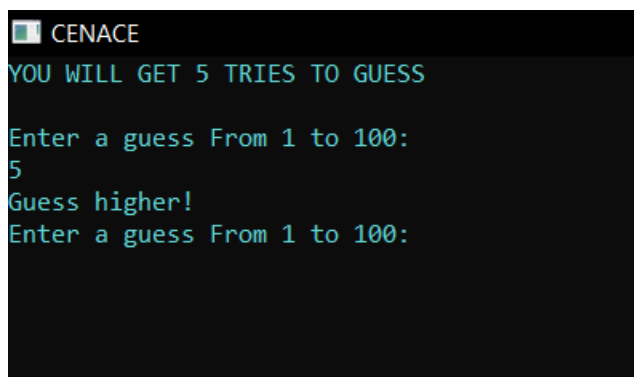
```

From this menu, the user can select his/her choice to give his/her number. Then, the number is stored into guess.

To create a successful game, there is a lot of if else conditions in it. The program will check the conditions according to the game rule and print if the user wins or loses or draws the game and prints the score.

And if the user doesn't get the correct number then the computer gives you hints to get the answer correct answer.

- If user guessed number is lower than the correct number then It will tell Guess higher!
And gives the user another chance to get the correct answer.



```

CENACE
YOU WILL GET 5 TRIES TO GUESS

Enter a guess From 1 to 100:
5
Guess higher!
Enter a guess From 1 to 100:

```

- If user guessed number is higher than the correct number then It will tell Guess lower!
And gives the user another chance to get the correct answer.

```
CENACE
YOU WILL GET 5 TRIES TO GUESS

Enter a guess From 1 to 100:
80
Guess lower!
Enter a guess From 1 to 100:
```

- If user's guessed number is correct then It will tell You got it!
And shows the answer and the score.

```
CENACE
You got it!
The number is 67
Your score is :1
```



https://github.com/alvi00/Number_Guessing_Game-in-c

Rock Paper Scissors

Author: Ahmad Fahmid

NSU ID: 2232502642

Bachelor of Computer Science and Engineering
Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh

What is Rock Paper Scissors?

Rock paper scissors: is a hand game originating from China, usually played between two people, in which each player simultaneously forms one of three shapes with an outstretched hand. These shapes are "rock" (a closed fist), "paper" (a flat hand), and "scissors" (a fist with the index finger and middle finger extended, forming a V). "Scissors" is identical to the two-fingered V sign (also indicating "victory" or "peace") except that it is pointed horizontally instead of being held upright in the air.

How does it work?

The players may count aloud to three, or speak the name of the game (e.g. "Rock! Paper! Scissors!"), either raising one hand in a fist and swinging it down with each syllable or holding it behind their back. They then "throw" or "shoot" by extending it towards their opponent. Variations include a version where players throw immediately on the third count (thus throwing on the count of "Scissors!"), or a version where they shake their hands three times before "throwing".

Rules of Rock Paper Scissors?

The winner of the game is decided as per the below rules:

- Rock vs Paper -> Paper wins.
- Rock vs Scissor -> Rock wins.
- Paper vs Scissor -> Scissor wins.
- Same vs same -> Draw

About the source code of Rock Paper Scissors

Rock paper scissors is written in C language

Interface of Rock Paper Scissors

After Rock paper scissors has opened the user will see a menu like this:

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$                                     $
$           Select any of them -     $
$                                     $
$           1. PLAY GAME              $
$                                     $
$                                     $
$           2. EXIT                   $
$                                     $
$                                     $
$                                     $
$                                     $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

User can press the corresponding number to go to that menu.

1. Play

This is the main part where you can play Rock paper scissors.

2. Exit

To exit Rock paper scissors.

Functions used in the codes are:

- int rock_paper_scissors();
- void game();
- void load();

Int rock_paper_scissors();

This function is the main function of the Rock paper scissor game.

In this game score counts the score of the full game. If user wins then 1 number added to score and if not it remains 0 or the same score .

Then rpsMatch counts the number of games, If user wants to get discount then the user have to play the game at 5 times. If he plays more than 5 times then 1taka will be added to the main bill.if he plays the game and score 5 or more than 5 then he will get discount.

Then the interference of the game is printed in this function.

As the game is loaded this interference comes

```

CENACE
Select your throw
1) ROCK
2) PAPER
3) SCISSORS
SELECTION: 

```

From this menu, the user can select his/her choice to give his/her throw. Then, the thrown character is stored into `ch_player_throw`, as a character. Then it is type-casted to `int` type and stored into `player_throw`.

If the user does not give any of 1, 2 or 3 then a sound is produced by `Beep(450, 250)`; The function `Beep(450, 250)` executes a sound. By default, the developer creates sounds with frequencies of 450 Hz and 250 ms respectively.

- `ai_throw = (rand() % 3)`

This will give me random numbers from 0 to infinity. So to get 1 2 3, It will mod that integer with 3 and it will give me 0,1,2 and to get 1,2,3 there is a plus 1 at the end.

In this section, the program will ask the user to write down his/her throw. And then what is given by the ai will be printed on the screen.

To create a successful game, there is a lot of if else conditions in it. The program will check the conditions according to the game rule and print if the user wins or loses or draws the game and prints the score.

```

Select your throw
1) ROCK
2) PAPER
3) SCISSORS
SELECTION: 2
AI GIVES SCISSORS.

SCISSORS beats PAPER. YOU LOSE.

YOUR SCORE IS :0

```



https://github.com/alvi00/Rock_Paper_Scissors_game_in_C

Conclusion

We are glad to work on this project. We have learned a lot in this project. Especially, it helped us to learn how to work in a group for programming in a large scale. Due to shortage of time we couldn't add a lot of features. But we have plans for the future. Maybe we can create a real-life food order system for a real restaurant in future with electronic payment, SMS, real-time food tracking, adding more games like flappy birds and so on.



<https://github.com/alvi00/Play-Bytes-Resturant-enhanced-with-AI>