# Software Documentation (Dice Bear)

Eastern Visayas State University, PH


**Developers:**

Shawn Michael A. Sudaria
Jade Hart Lee
Renz Ivan Monteza

**Presented To:**

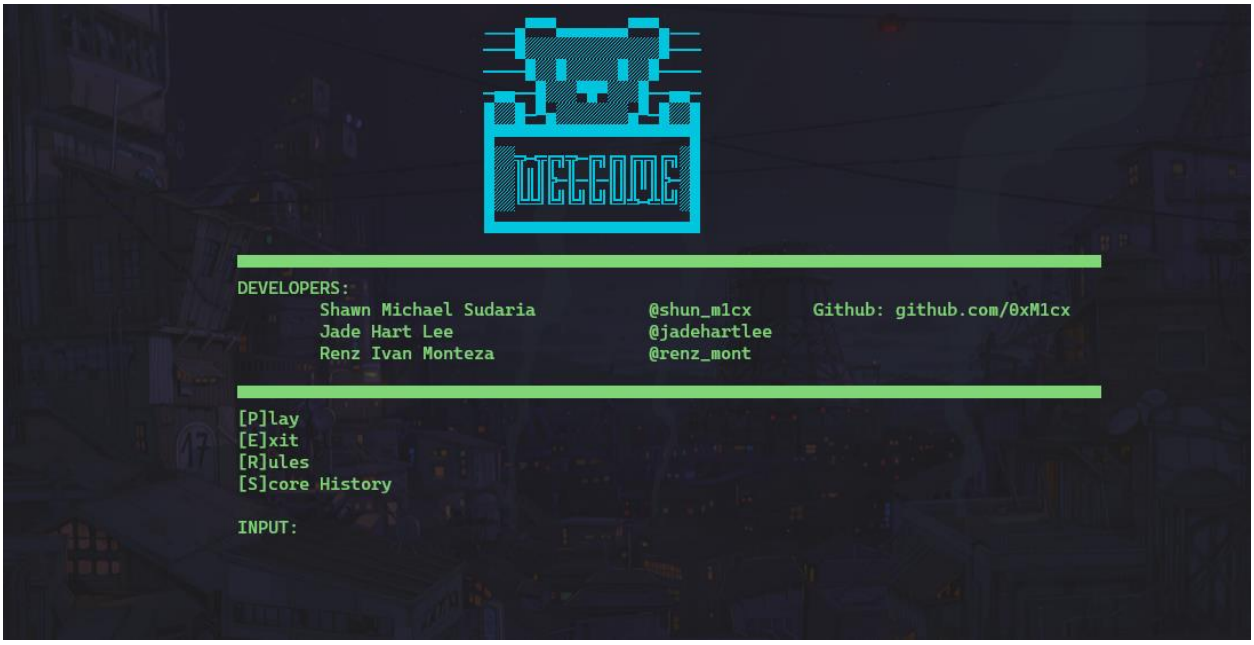
Mr. Eric Sta. Singh

## Description

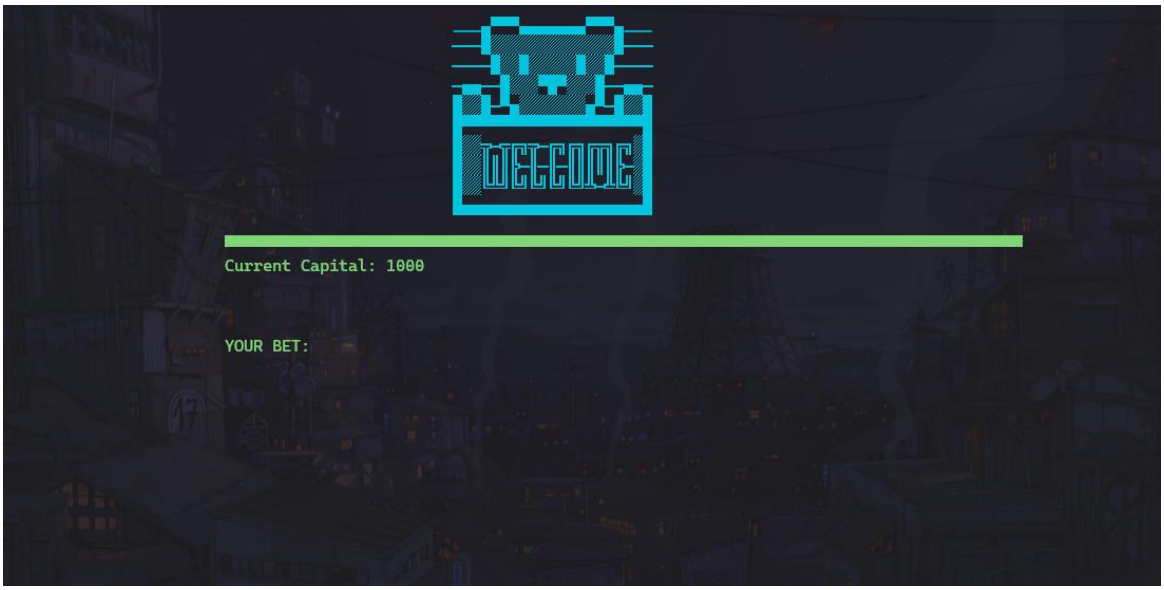
**Dice Bear** is a game created in Visual Studio Code with the C programming language and the minGW compiler. The objective of the game is to win as much money as possible based on the player's wager. When a player's capital reaches zero, the game ends. The program will generate two random numbers and compute the sum of the two numbers on the first roll. If the sum of the two numbers is 7, the player wins; otherwise, the sum is 11 or 2, and the player loses. If the sum of the preceding numbers is not 7, 11, or 2, the program will generate another set of random numbers and roll 2. On subsequent rolls, new random numbers will be generated; the player can only win if the sum of the numbers is equal to the sum of the previous roll and the player will lose if the sum is 7; otherwise, the next roll will be taken. Furthermore, the program also contains added features like the Score History section, where the user can see previous scores from previous games, Rules Section where the end user can see and read the rules of the game.

**The flow of the program is as follows:**

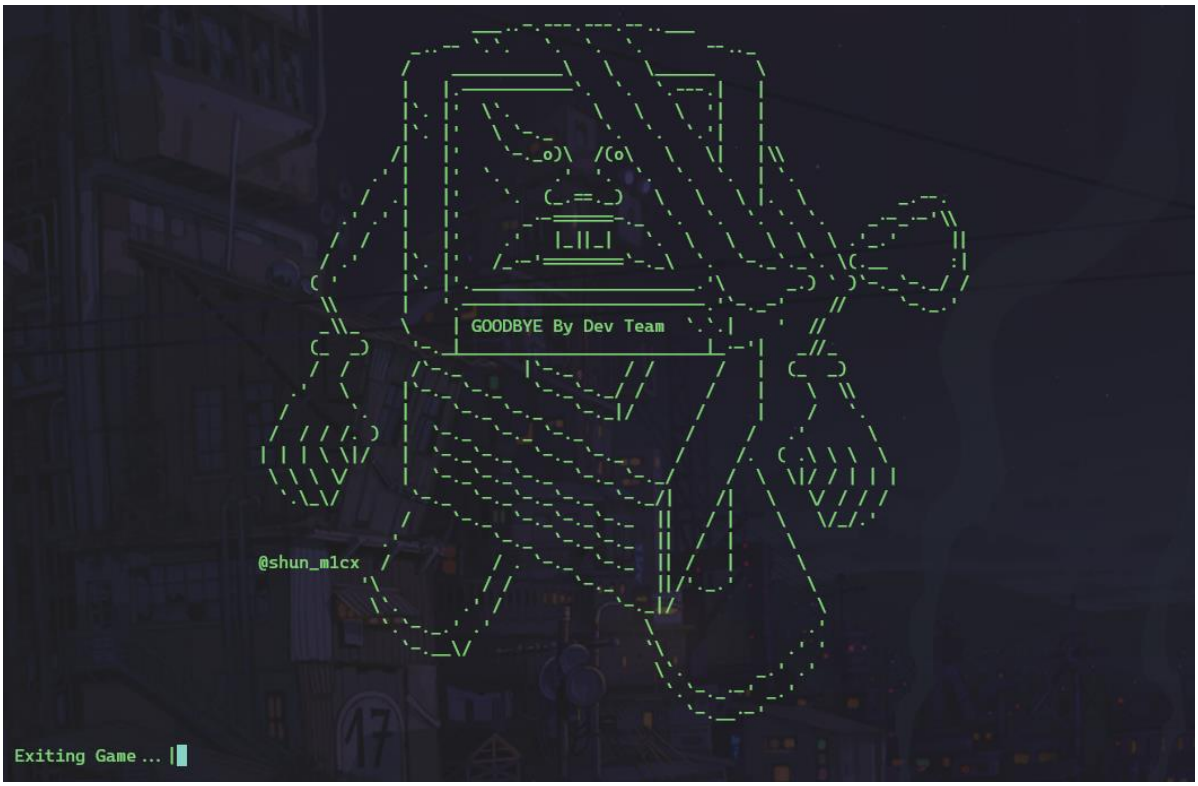
1. When the program is executed, the player will see the various options available in the start menu.



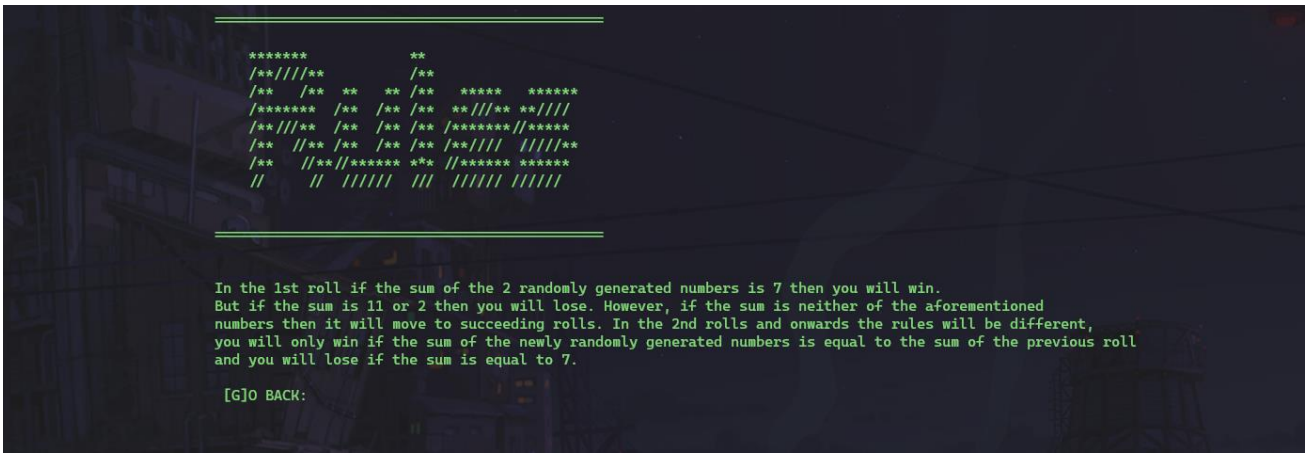
2. If the player enters P, he will be taken to the betting stage, where he will place a bet within his capital range.



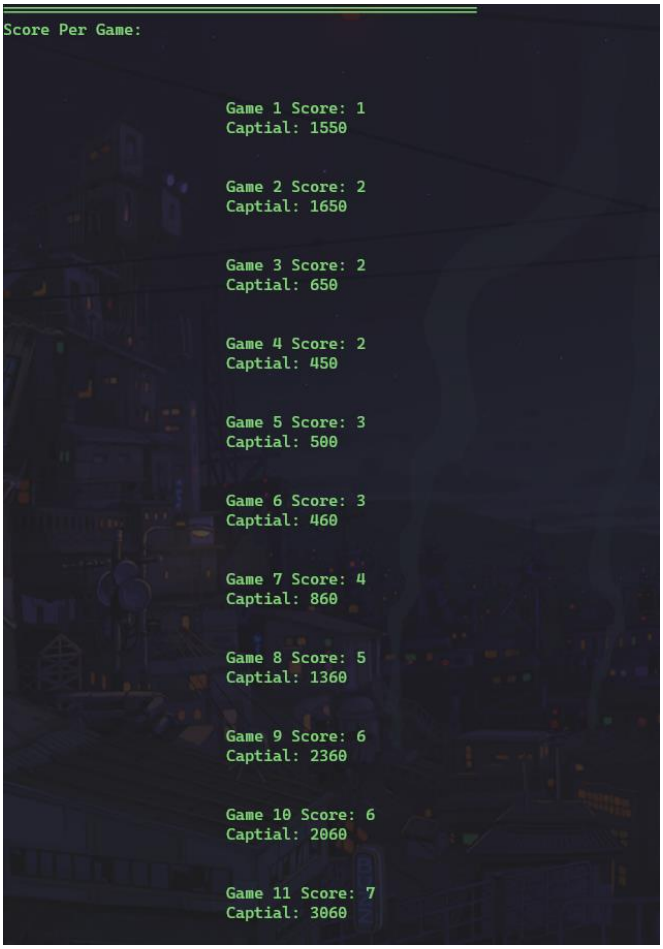

3. If the player types E, he will be directed to the Goodbye Screen.



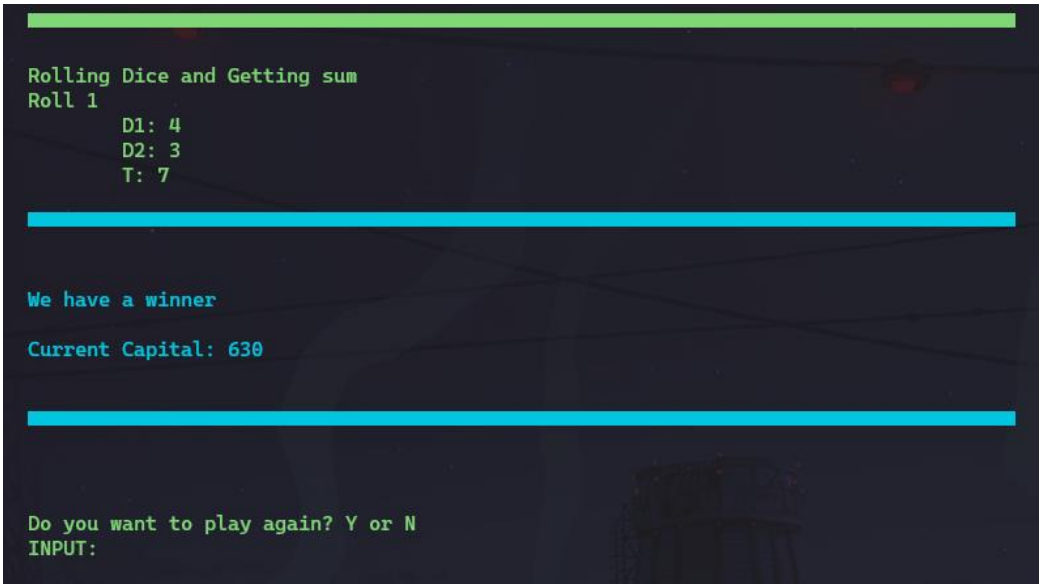
4. If the player types R, he will be redirected to the Rules Screen.

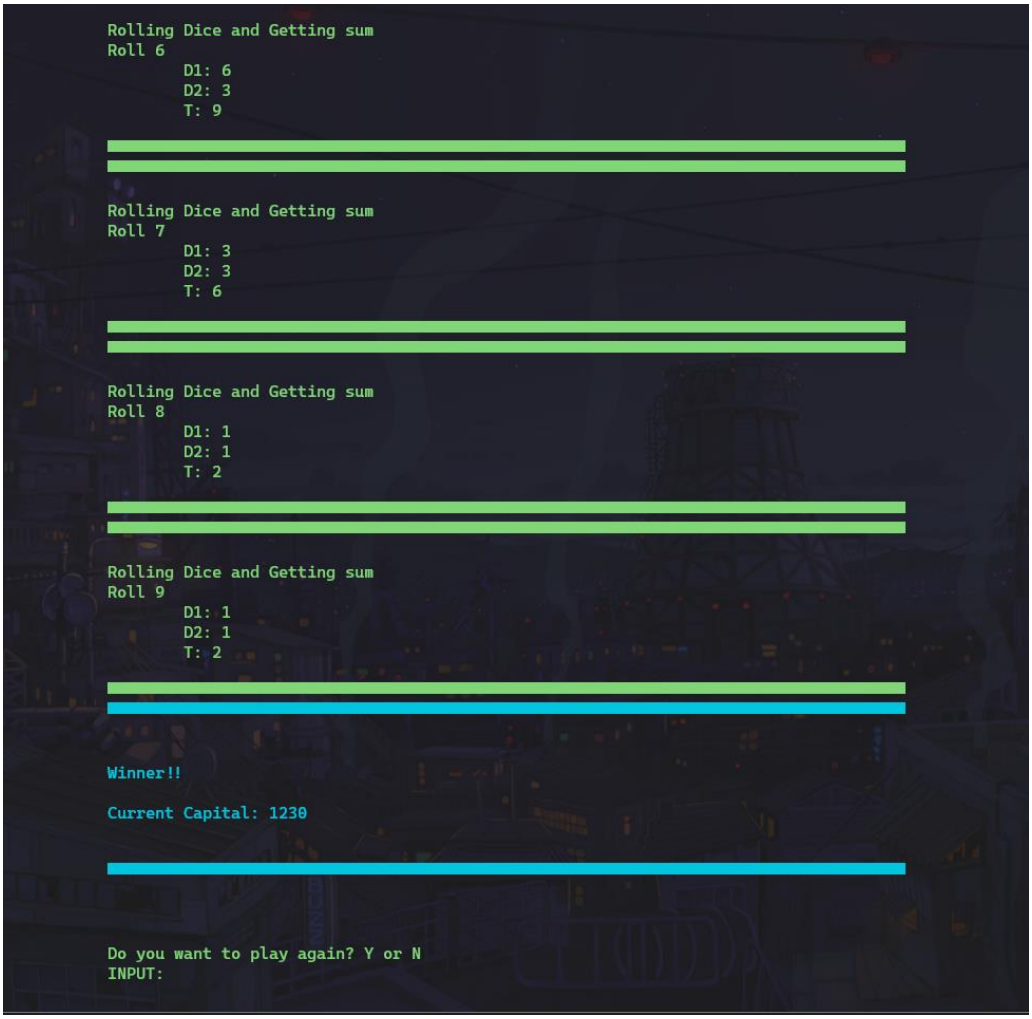



5. If the player types S, he will be redirected to the Score History Screen. The text in this score screen is taken from a file(score.txt) wherein the data was printed to the file. This acts as a flat file database for the game data to be saved. See the code from line 80 – 90 for more details.

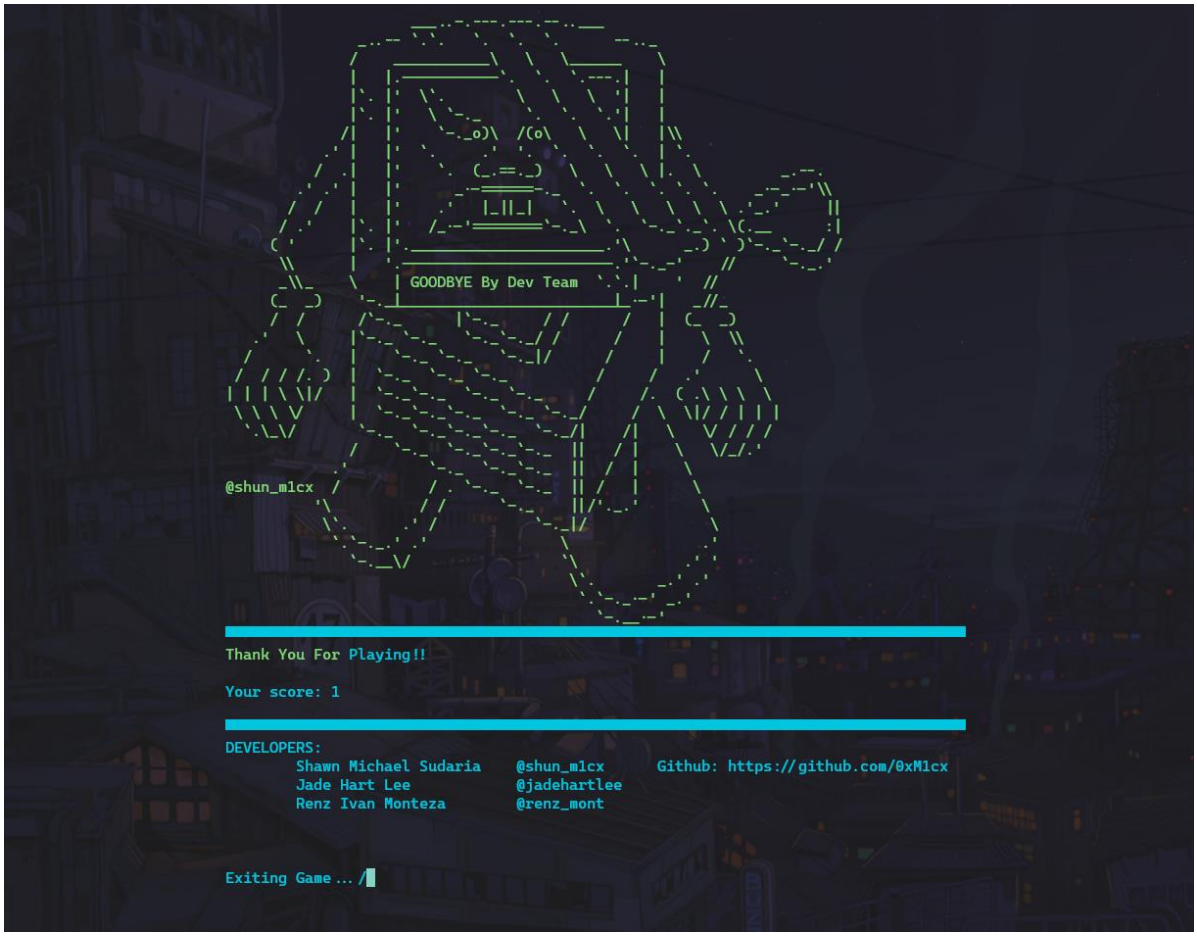


6. After the player has placed his/her bet, the program will begin rolling the dices, calculating and comparing the sum to determine whether the player has won, lost, or if the game will continue to the next roll. After the dice are rolled, the sum is calculated and compared. Regardless of the outcome, the player will see a play again text box with the option to continue playing.

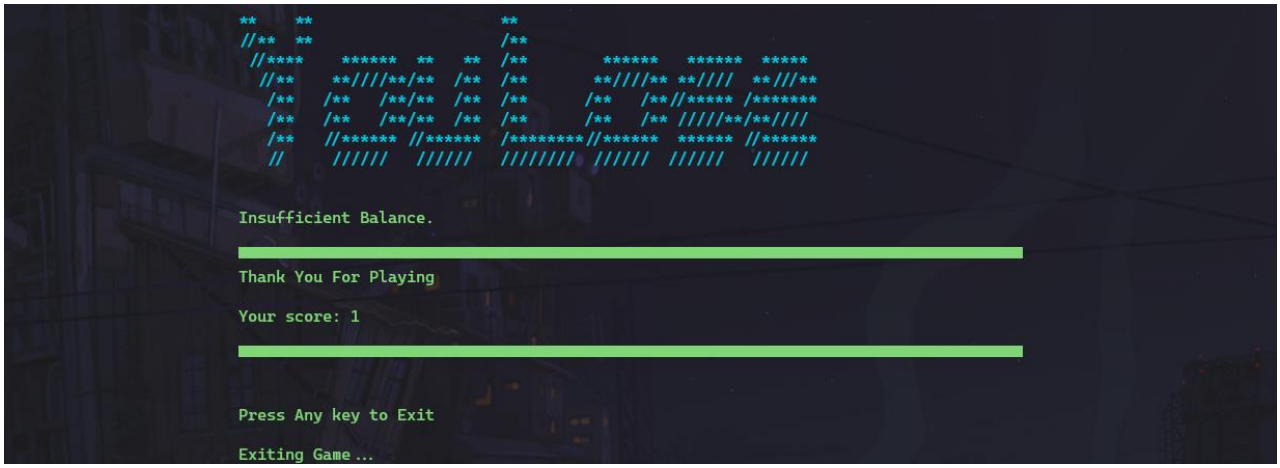

Rolling Dice and Getting sum
Roll 1
D1: 4
D2: 3
T: 7
We have a winner
Current Capital: 630
Do you want to play again? Y or N
INPUT:


Rolling Dice and Getting sum
Roll 6
D1: 6
D2: 3
T: 9
Rolling Dice and Getting sum
Roll 7
D1: 3
D2: 3
T: 6
Rolling Dice and Getting sum
Roll 8
D1: 1
D2: 1
T: 2
Rolling Dice and Getting sum
Roll 9
D1: 1
D2: 1
T: 2
Winner!!
Current Capital: 1230
Do you want to play again? Y or N
INPUT:

7. If the player selects Y, he will be taken back to the betting stage; if he selects N, he will be taken to the goodbye screen.




8. When a player's capital reaches zero, they are no longer able to play and are redirected to the lose screen.



## Code

```c
#include <stdio.h> // For the scanf and printf functions
#include <time.h> // for the time() function for generating a new random number everytime the rand() is called
#include <stdlib.h>
#include <unistd.h> // for the sleep(), compatability for linux
#include <windows.h> // for the sleep(), compatability for windows
#include <conio.h> // for the getch() function

/*
DEVELOPERS:
+ @shun_m1cx | Shawn Michael Sudaria
+ @renz_mont | Renz Ivan Monteza
+ @jadehartlee | Jade Hart Lee
*/

#define size 1000
int score = 0; // Para pag determine han score han user after han game.

// Declaring and initializing the global variables to be used
int currentCapital = 1000;
int RollsArr[size][size];
int currentRoll = 0;

// Declaring the functions to be used
int getRanNums();
void succedingRolls(int userBet, int valToWin), play(int userBet), bet(),
playAgain(), displayInfo(), rules(), loadingAnimation(), displayBanner();
void drawBlocks(), goodbye(), savingGame(), scoreHistory();
int numOfGames = 0;
int winRate;

FILE *filePointer;




void savingGame(){

    // Para pag calculate han win rate.

    filePointer = fopen("C:\\Users\\Asus\\OneDrive\\Documents\\EVSU\\Computer Programming 1\\FinalOutput\\Score.txt", "a");
```

```
    if(filePointer != NULL){
        fprintf(filePointer, "\n\n\n\t\t\tGame %d Score: %d\n\t\t\tCaptial: %d",
numOfGames, score, currentCapital);
        fclose(filePointer);
    }
}


void goodbye(){
    printf("\t\t\t                    ___..-.---.---.--.___\n");
    printf("\t\t\t               _..-- `.`.   `.  `.  `.      --.._\n");
    printf("\t\t\t             /    ___________\\   \\   \\_____    \\\n");
    printf("\t\t\t             |   |.-----------`.  `.  `.----.|   |\n");
    printf("\t\t\t             |`. |'  \\`.        \\   \\   \\  '|   |\n");
    printf("\t\t\t             |`. |'   \\ `-._     `.  `.  `.'|   |\n");
    printf("\t\t\t            /|   |'    `-._o)\\  /(o\\   \\   \\|   |\\\\\n");
    printf("\t\t\t          .' |   |' `.     .'  '.  `.  `.  `.  | `.\n");
    printf("\t\t\t         /  .|   |'    `.  (_.==._)   \\   \\   \\ |.  \\
    _.--.\n");
    printf("\t\t\t       .' .' |   |'      _.-======-._  `.  `.  `. `. `.    _.-
_.-'\\\\\n");
    printf("\t\t\t      /  /   |   |'    .'   |_||_|   `.  \\   \\   \\  \\  \\
.'_.'     ||\n");
    printf("\t\t\t     / .'    |`. |'   /_.-'========`-._\\  `.  `-._`._`.
\\(.__      :|\n");
    printf("\t\t\t    ( '      |`. |'._____________________.'\\      _.) ` )`-
._`-._/ /\n");
    printf("\t\t\t     \\\\\\      |   '.------------------------.'`-._-
'    //     `-._.'\n");
    printf("\t\t\t     _\\\\\_    \\\    | GOODBYE By Dev
Team  `.`.|    '  //\n");
    printf("\t\t\t    (_  _)    '-._|_____________________|_.-'|   _//_\n");
    printf("\t\t\t    /  /      /`-._      |`-._    / /     /   |  (_  _)\n");
    printf("\t\t\t   .'   \\     |`-._ `-._   `-._`-._/
/      /    |    \\  \\\\\n");
    printf("\t\t\t  /      `.   |    `-._ `-._   `-
._|/      /     |    /   `.\n");
    printf("\t\t\t /  / / /. )  |  `-._  `-._ `-
._          /     /   .'      \\\n");
    printf("\t\t\t| | | \\ \\|/   |  `-._`-._  `-._ `-
._     /     /.  ( .\\ \\ \\  \\\n");
    printf("\t\t\t \\ \\ \\ \\/     |  `-._`-._`-._  `-._ `-._/     /  \\  \\|/ /
| | |\n");
    printf("\t\t\t  `.\\_\\/       `-._  `-._`-._`-._  `-._/|    /|   \\   \\/ /
/ /\n");
```

```
    printf("\t\t\t              /    `-._  `-._`-._`-._  ||   /
|    \\   \\/_/.'\n");
    printf("\t\t\t            .'         `-._  `-._`-._  ||  /  |     \\\n");
    printf("\t\t\t@shun_m1cx  /          / .`-._  `-._  || /   |      \\\n");
    printf("\t\t\t          '\\          / /      `-._    ||/'._.'       \\\n");
    printf("\t\t\t           \\`.      .' /           `-._|/             \\\n");
    printf("\t\t\t            `.`-._.' .'                \\               .'\n");
    printf("\t\t\t              `-.__\\/                   `\\             .' '\n");
    printf("\t\t\t                                        \\`.       _.' .'\n");
    printf("\t\t\t                                         `.`-._.-' _.'\n");
    printf("\t\t\t                                           `-.__.-'\n");
}


// Function para mag draw horizontal block
void drawBlocks(){
    printf("\t\t\t");
    for (int i = 1; i < 85; i++)
    {
        printf("%c", 223);
    }
    printf("\n");
}







// Main function where the program will first start
void main(){
    system("cls");
    srand(time(0));
    char UserInput; // This is used to get the user input

    filePointer = fopen("C:\\Users\\Asus\\OneDrive\\Documents\\EVSU\\Computer
Programming 1\\FinalOutput\\Score.txt", "a");
    fprintf(filePointer,
"\n==================================================\nScore Per Game:\n");
    fclose(filePointer);

    displayBanner();
    printf("\033[0;32m");

    drawBlocks();
```

```
    displayInfo();
    drawBlocks();

    printf("\t\t\t[P]lay\n\t\t\t[E]xit\n\t\t\t[R]ules\n\t\t\t[S]core
History\n\n\t\t\tINPUT: ");
    scanf(" %c", &UserInput);

    if(UserInput == 'P' || UserInput == 'p'){
        bet();
    }
    else if(UserInput == 'E' || UserInput == 'e'){
        system("cls");
        goodbye();
        drawBlocks();
        savingGame();
        printf("\n\t\t\tExiting Game... ");
        loadingAnimation();
        loadingAnimation();
        exit(0);
    }
    else if(UserInput == 'R' || UserInput == 'r'){
        rules();
    }else if(UserInput == 'S' || UserInput == 's'){
        scoreHistory();
    }else{
        printf("\t\t\tInvalid Input, it must be either P or E");
        sleep(2);
        main();
    }
}


char c; // to be used when reading the score.txt file
void scoreHistory(){
    system("cls");
    drawBlocks();
    printf("\t\t\tSCORE HISTORY\n");
    drawBlocks();
    filePointer = fopen("C:\\Users\\Asus\\OneDrive\\Documents\\EVSU\\Computer
Programming 1\\FinalOutput\\Score.txt", "r");
    while ((c = getc(filePointer)) != EOF) printf("%c", c);
    fclose(filePointer);

    printf("GO BACK?");
    getch();
```

```
    main();
}


void displayBanner(){
    printf("\033[0;37m");
    printf("\t\t\t\t\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",95,95,95,22
0, 223, 223, 223, 220,220,220,220,220,220,220,223,223,223,220,95,95,95);
    printf("\t\t\t\t\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",
196,196,196, 219, 178,178,
178,178,178,178,178,178,178,178,178,178,178,219,196,196,196);
    printf("\t\t\t\t\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",
196,196,196,
196,219,178,178,219,178,178,178,178,178,219,178,178,219,196,196,196);
    printf("\t\t\t\t\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",
196,220,220,196,196,219,178,178,178,223,219,223,178,178,178,219,196,196,220,220,1
96);
    printf("\t\t\t\t\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",
219,178,178,219,196,223,220,178,178,178,178,178,178,178,220,223,196,219,178,178,2
19);
    printf("\t\t\t\t\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",219,223,223
,223,223,223,223,223,223,223,223,223,223,223,223,223,223,223,223,223,223,219);
    printf("\t\t\t\t\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",219,178,178
, 203,196,203, 201,187,203,196,201,187,201,187,201,203,187,201,187,178,219);
    printf("\t\t\t\t\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",219,178,178
,  186,186,186,  204,196, 186,196,186,196,186,186,186,186,186,204,196,178,219);
    printf("\t\t\t\t\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",219,178,178
, 200,202,188,  200,188, 200,188, 200,188, 200,188,202,196,202, 200,188,178,219);
    printf("\t\t\t\t\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n\n",219,220,2
20,220,220,220,220,220,220,220,220,220,220,220,220,220,220,220,220,220,219);
}

// The Animated Loading Screen
void loadingAnimation(){
    for (int i = 1; i<=5; i++) {
        Sleep(90);
        printf("\b\\");

        Sleep(90);
        printf("\b|");

        Sleep(90);
        printf("\b/");

        Sleep(90);
```

```
        printf("\b-");
        Sleep(90);

    }
    printf("\b ");
}

void playAgain(){
    char userInput;

    printf("\n\n\n\t\t\tDo you want to play again? Y or N\n\t\t\tINPUT: ");
    scanf(" %c", &userInput);

    if(userInput == 'Y' || userInput == 'y'){
        savingGame();
        bet();
    }
    else if(userInput == 'N' || userInput == 'n'){
        system("cls");
        savingGame();
        winRate = (score / numOfGames) * 100;
        filePointer = fopen("C:\\Users\\Asus\\OneDrive\\Documents\\EVSU\\Computer
Programming 1\\FinalOutput\\Score.txt", "a");
        fprintf(filePointer, "\n\n\t\t\tWIN RATE: %d\n\n", winRate);
        fclose(filePointer);
        goodbye();
        printf("\033[0;37m");
        drawBlocks();
        printf("\033[0;34m");
        printf("\033[0;32m \t\t\tThank You For\033[0;37m Playing!!\n\n");
        printf("\t\t\tYour score: %d\n\n", score);
        printf("\033[0;37m");
        drawBlocks();
        displayInfo();
        printf("\n\n\t\t\tSaving Stats & Exiting Game... ");
        loadingAnimation();
        loadingAnimation();
        exit(0);
    }else{
        printf("\t\t\tWrong Input!!!");
        sleep(2);
        system("cls");
        playAgain();
    }
```

```
}

// Function para mag generate ngan compute an random numbers everytime na gin
tawag ine na function
int getRanNums(){
    int dieOne = rand() % 6 + 1; // Invokes a random number for die 1
    int dieTwo = rand() % 6 + 1;
    int sum = dieOne + dieTwo;

    RollsArr[currentRoll][0] = dieOne;
    RollsArr[currentRoll][1] = dieTwo;
    RollsArr[currentRoll][2] = sum;

    return sum;
}


void succedingRolls(int userBet, int valToWin){

    srand(time(0));
    int _valueToWin = valToWin;
    int isDone = 1;

    while(isDone){
        drawBlocks();
        int randNum = getRanNums();
        printf("\n\t\t\tRolling Dice and Getting sum  ");
        loadingAnimation();
        printf("\n");

        printf("\t\t\tRoll %d\n\t", currentRoll + 1);
        printf("\t\t\tD1: %d\n\t", RollsArr[currentRoll][0]);
        printf("\t\t\tD2: %d\n\t", RollsArr[currentRoll][1]);
        printf("\t\t\tT: %d\n\n", RollsArr[currentRoll][2]);
        drawBlocks();

        if(randNum == _valueToWin){
            printf("\033[0;37m");
            drawBlocks();
            printf("\n\n\t\t\tWinner!!\n\n");
            currentCapital += userBet;
            score++;
            printf("\t\t\tCurrent Capital: %d\n\n\n", currentCapital);
            drawBlocks();
            printf("\033[0;32m");
```

```
            loadingAnimation();
            playAgain();
            isDone = 0;
        }else if (randNum == 7)
        {
            printf("\033[0;37m");
            drawBlocks();
            printf("\n\n\t\t\tYou Lost!!\n\n");
            currentCapital -= userBet;
            printf("\t\t\tCurrent Capital: %d\n\n\n", currentCapital);
            drawBlocks();
            printf("\033[0;32m");
            loadingAnimation();
            playAgain();
            isDone = 0;
        }else{
            _valueToWin = randNum;
            currentRoll++;
        }
    }
}

void play(int userBet){
    int randNum = 0; // reset the number to know have the player win because of
the last sum of the previous game
    currentRoll = 0; // to the currect roll and change the values from the
previous game
    randNum = getRanNums(); // Generating random numbers para han first Roll

    numOfGames++;

    system("cls");
    drawBlocks();
    printf("\n\t\t\tRolling Dice and Getting sum  ");
    loadingAnimation();



    printf("\n\t\t\tRoll 1\n\t");
    printf("\t\t\tD1: %d\n\t", RollsArr[currentRoll][0]);
    printf("\t\t\tD2: %d\n\t", RollsArr[currentRoll][1]);
    printf("\t\t\tT: %d\n\n", RollsArr[currentRoll][2]);

    currentRoll++;
```

```
    if (randNum == 7)
    {
        printf("\033[0;37m");
        drawBlocks();
        printf("\n\n\t\t\tWe have a winner \n\n");
        currentCapital += userBet;
        score++;
        printf("\t\t\tCurrent Capital: %d\n\n\n", currentCapital);
        drawBlocks();
        printf("\033[0;32m");
        loadingAnimation();
        playAgain();
    }
    else if(randNum == 11 || randNum == 2){
        printf("\033[0;37m");
        drawBlocks();
        printf("\n\n\t\t\tYou have lost\n\n");
        currentCapital -= userBet;
        printf("\t\t\tCurrent Capital: %d\n\n\n", currentCapital);
        drawBlocks();
        printf("\033[0;32m");
        loadingAnimation();
        playAgain();
    }else{
        succedingRolls(userBet, randNum);
    }
}

void bet(){
    int userBet;

    system("cls");

    displayBanner();
    printf("\033[0;32m");

    if (currentCapital == 0)
    {
        winRate = (score / numOfGames) * 100;
        filePointer = fopen("C:\\Users\\Asus\\OneDrive\\Documents\\EVSU\\Computer
Programming 1\\FinalOutput\\Score.txt", "a");
        fprintf(filePointer,"\n\nWIN RATE: %d\n\n", winRate);
        fclose(filePointer);
        savingGame();
        system("cls");
```

```
        printf("\033[0;37m");
        printf("\t\t\t**    **                    **
  \n");
        printf("\t\t\t//**  **                    /**
   \n");
        printf("\t\t\t
//****    ******  **   **  /**        ******   ******  *****\n");
        printf("\t\t\t  //**    **////**/**  /**  /**       **////**
**////  **///**\n");
        printf("\t\t\t   /**   /**   /**/**  /**  /**      /**   /**//*****
/*******\n");
        printf("\t\t\t   /**   /**   /**/**  /**  /**      /**   /**
/////**/**////\n");
        printf("\t\t\t   /**   //****** //******  /********//******  ******
//******\n");
        printf("\t\t\t   //     //////   //////   ////////  //////  //////   ////
//\n");
        printf("\033[0;32m");
        printf("\n\n\t\t\tInsufficient Balance.\n\n");
        drawBlocks();
        printf("\t\t\tThank You For Playing");
        printf("\n\n\t\t\tYour score: %d\n\n", score);
        drawBlocks();
        printf("\n\n\t\t\tPress Any key to Exit ");
        getch();
        printf("\n\n\t\t\tSaving & Exiting Game... ");
        loadingAnimation();
        loadingAnimation();
        exit(0);
    }
    drawBlocks();
    printf("\t\t\tCurrent Capital: %d\n\n\n", currentCapital); // Displaying the
current capital to the user
    printf("\n\t\t\tYOUR BET: ");
    scanf("%d", &userBet);

    if(userBet <= 0 || userBet > currentCapital){
        printf("\t\t\tInvalid Bet");
        printf("\n\t\t\tGoing back... ");
        loadingAnimation();
        bet();
    }
    play(userBet);
}
```

```
void displayInfo(){
    char strings[30][30] = {
        "\t\t\tDEVELOPERS:\n\t", "\t\t\tShawn Michael Sudaria\t\t",
"@shun_m1cx\t", "Github: github.com/0xM1cx\n\t", "\t\t\tJade Hart Lee\t\t\t",
"@jadehartlee\n\t", "\t\t\tRenz Ivan Monteza\t\t", "@renz_mont\n\n"
    };

    for(int i = 0; i <= 7; i++){
        for (int b = 0; b < 30; b++)
        {
            Sleep(10);
            printf("%c", strings[i][b]);
        }
    }
}


void rules(){
    char userInput;
    system("cls");
    printf("\t\t\t===========================================\n\n");

    printf("\t\t\t    *******            **                \n");
    printf("\t\t\t    /**////**          /**                \n");
    printf("\t\t\t    /**   /**  **    ** /**   *****   ******\n");
    printf("\t\t\t    /*******  /**  /** /**  **///** **////\n");
    printf("\t\t\t    /**///**  /**  /** /** /*******//***** \n");
    printf("\t\t\t    /**  //** /**  /** /** /**////  /////**\n");
    printf("\t\t\t    /**   //**//****** *** //****** ****** \n");
    printf("\t\t\t    //     //  //////  ///  ////// //////\n\n\n");
    printf("\t\t\t===========================================\n\n\n");
    printf("\t\t\tIn the 1st roll if the sum of the 2 randomly generated numbers
is 7 then you will win.\n\t\t\tBut if the sum is 11 or 2 then you will lose.
However, if the sum is neither of the aforementioned\n\t\t\tnumbers then it will
move to succeeding rolls. In the 2nd rolls and onwards the rules will be
different,\n\t\t\tyou will only win if the sum of the newly randomly generated
numbers is equal to the sum of the previous roll\n\t\t\tand you will lose if the
sum is equal to 7.\n\n");

    printf("\t\t\t[G]O BACK: ");
    scanf(" %c", &userInput);

    if(userInput == 'G' || userInput == 'g'){
        main();
    }else{
```

```
        printf("\t\t\tWRONG INPUT!!");
        loadingAnimation();
        rules();
    }
}
```