**Symbiosis Institute of Technology, Pune**

**Beyond 11**

**Fundamentals of Data Structures**

**#define MINI\_PROJECT AMAZING**

**Rishabh Kaushick, 15070121152,**

**Samartha Narang, 15070121158,**

**Tanisha Razdan, 15070121168,**

**Yash Saboo, 15070121170.**

**Date of Submission: 14 November 2016**

Table of Contents

[Header Files Used 3](#_Toc466856907)

[In Built Function Used 3](#_Toc466856908)

[Graphics Functions Used 4](#_Toc466856909)

[Functions Created 4](#_Toc466856910)

[Data Structures Concepts Used 6](#_Toc466856911)

[File Handling 6](#_Toc466856912)

[Working of The Game 7](#_Toc466856913)

[Game Algorithm 8](#_Toc466856914)

[Screen Shots: 9](#_Toc466856915)

[Color Changing Home Screen 9](#_Toc466856916)

[The Main Menu 10](#_Toc466856917)

[The Game 11](#_Toc466856918)

[End Of Game 12](#_Toc466856919)

[Source Code 14](#_Toc466856920)

# Header Files Used

|  |  |
| --- | --- |
| Header File | Reason for Use |
| #include <stdio.h> | scanf() and printf() functions |
| #include <conio.h> | getch() and clrscr() functions |
| #include <graphics.h> | For various graphics functions (described later) |
| #include <dos.h> | delay() function |
| #include <string.h> | strcpy() function |
| #include <stdlib.h> | rand() and srand() functions |
| #include <time.h> | time() function |

# In Built Function Used

|  |  |  |
| --- | --- | --- |
| Function | Header File | Reason for Use |
| kbhit() | <conio.h> | To display main menu until there is a keyboard hit |
| delay() | <dos.h> | To delay the running of the code when graphics is displayed |
| strcpy() | <string.h> | To copy the name string of player into the high score name string |
| scanf() and printf() | <stdio.h> | To read from the user and display text |
| sprintf() | <stdio.h> | In order to convert alphabets with numbers to a character string |
| clrscr() | <conio.h> | Clearing the screen without clearing the graphics |
| time(0) | <time.h> | A function that we use as a parameter for the srand function. |
| rand() | <stdlib.h> | A function that generates a pseudo random series of numbers |
| srand() | <stdlib.h> | To generate a random number between 0 and the highest number tile in the game |
| sound() | <dos.h> | To generate a sound when two of the same number tiles combine and another sound when we give a wrong input |
| nosound() | <dos.h> | To stop the sound from continuing to playing |
| fopen() | <stdio.h> | To open a file in order to read or write into it |
| fclose() | <stdio.h> | To close the file that has been opened ealier |
| fscanf() | <stdio.h> | To read from a file |
| fprintf() | <stdio.h> | To write into to a file |

# Graphics Functions Used

|  |  |  |
| --- | --- | --- |
| Function | Parameters | Reason For Use |
| initgraph() | Grapic Driver, graph mode, path driver | In order to initialize the graphics |
| setbkcolor() | Integer parameter or string enumerated color | To set the color of the screen background |
| settextstyle() | Style font, direction of text, character size | To change the font and font size of different text |
| outtextxy() | x co-ordinate, y-coordinate, string | To display the string using graphics at a specified x and y co-ordinate |
| setfillstyle() | Pattern, color | To set a colored graphics pattern |
| rectangle() | X1, Y1, X2, Y2 | It draws a rectangle on the screen |
| bar() | X1, Y1, X2, Y2 | It draws a rectangle which can be filled with the setfillstyle() function |
| setlinestyle() | Line style, user bit pattern, thickness | To change the style of the line to be displayed |
| getmaxx() | - | To obtain the maximum x co-ordinate of the display |
| getmaxy() | - | To obtain the maximum y co-ordinate of the display |
| cleardevice() | - | Clears the graphics displayed |
| closegraph() |  | Exits from the graphics screen |

# Functions Created

|  |  |
| --- | --- |
| Function Name | It's Function |
| Void home() | It includes a series of graphics functions to display the home or starting page of the game. |
| Int checkHighScore() | It is called after the game is over to see if the current player has beaten the high-score  If the current player beat the high score, then the current player's name and score will be stored into the text file |
| Void displayHighScore() | It reads the high-scores from the text file and then displays them on the screen. |
| Void mainMenu() | This function is to display the main menu from where the user would enter his name and the game would start. |
| Void buildrectangle() | This function sets the background and layout of the grid of the game. |
| Int isempty() | This checks if the particular block of the grid has a number more than 0 in it. This is required so that we can make sure that a new number can be inserted in that particular block or not. |
| Int Gameover() | This function checks if any of the blocks have the ability to get the next number added into. If yes, then the game is not over.  If no, then the game is over. |
| Int rand\_int(int, int) | This function uses a function called rand() to generate random number between the lowest number in the grid and highest number in the grid (include the highest number). |
| Void checkadj(int, int, int) | This function is used to compare the adjacent block’s value with the inserted value of the user's block.  This is necessary to figure out if there can be a “fusion” of the block’s value. |
| Int algo(int) | This function figures out which all blocks are to be considered for comparison with the user’s chosen block to perform “fusion”. |
| Int high1() | It calculates the highest number in the grid so that we can find the upper bound for random number creation |
| Void printscore() | This function prints the score in the graphics |
| Void writetext() | This function prints the block’s value in its particular block.  It also showcases the block number with respect to the grid so that the user can effortlessly play the game. |
| Int perform(int, int) | This function will be the most important function among all because it’s functionality describes the “Performance” of the game. |
| Void main() | We have used the main to initialize the use of graphics and laid out the main outline of the working of the code by effectively calling the functions above mentioned. |

# Data Structures Concepts Used

In order to store the name and score of the currently playing player and the name and score of the high scorers we have implemented a structure as follows:

struct scoreCard

{

char name[10];

int score;

}highScore[5], newPlayer;

With this structure we have linked the player name with their own score.

# File Handling

The use of file handling was necessary in order to be able to permanently store the name and score of a person who has gotten the highest score. Therefore once the game is closed and reopened later, the stored high score will not get erased.

In order to achieve this we initialized a file pointer, fp, as follows:

FILE \*fp;

We open or create a file by using the fopen() function and assigning it to the file pointer. For writing into the file we will pass the parameter 'w' and for reading from the file we pass the parameter 'r'.

fp = fopen("mytextfile.txt",' r ');

After opening the file in read mode, we can read from the text file by using the fscanf() function and store the contents into the structure variable. We used the function as follows:

fscanf(fp, "%s %d" ,highScore[i].name, &highScore[i].score);

Similarly fprintf() function can be used to write some text into the file:

fprintf(fp, "%s %d", highScore[i].name, highScore[i].score);

Finally after the operations on the file have been completed, we must close the file by using fclose() funnction.

fclose(fp);

# Working of The Game

Initialize the array  
Initialize the graphics

Read from .txt file and store the data into the structure

Display Main Menu

Display the Game Screen & Input choice

Valid Choice?

No

Yes

Display score & highscore list

No

Yes

gameover?

Perform Algorithm

Print Invalid

# Game Algorithm

Yes

Yes

Yes

No

No

current block value++

flag == 1?

All adjacent blocks are checked?

set flag =1

adjacent block value =0

No

No

Adjacent block == current block number?

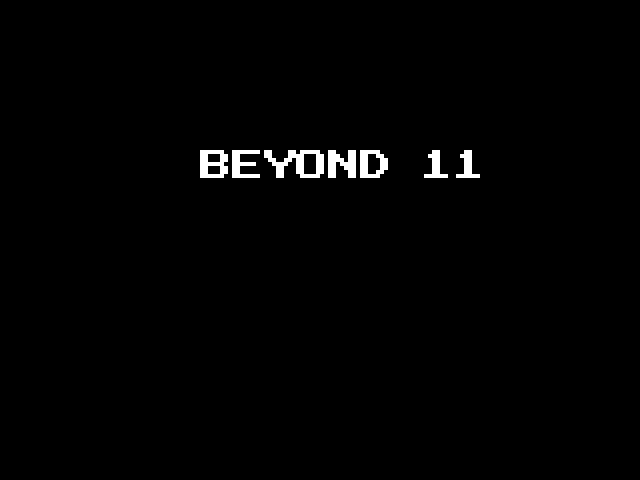
Check the adjacent block with the current block value

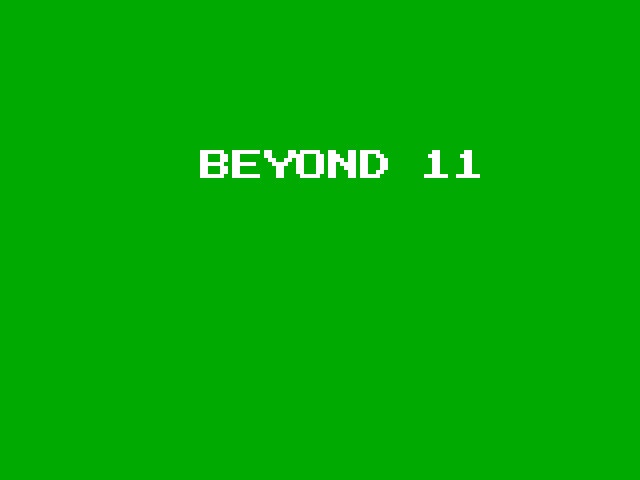
Valid Input?

Enter the box number to put the next number

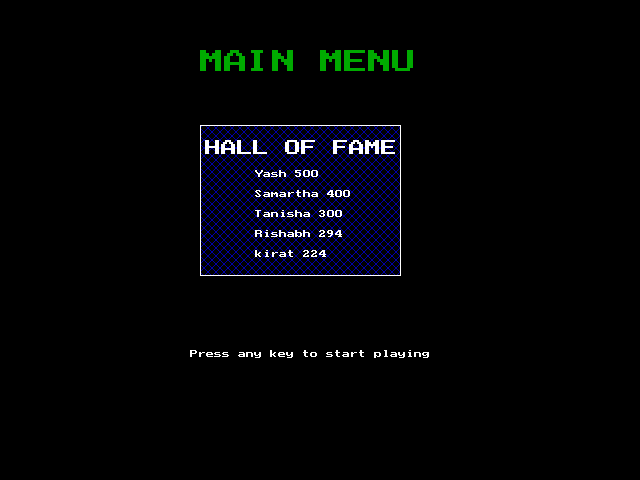
# Screen Shots:

## Color Changing Home Screen

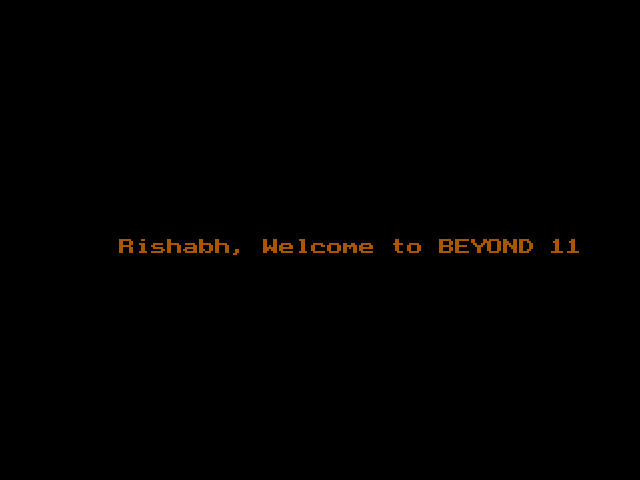




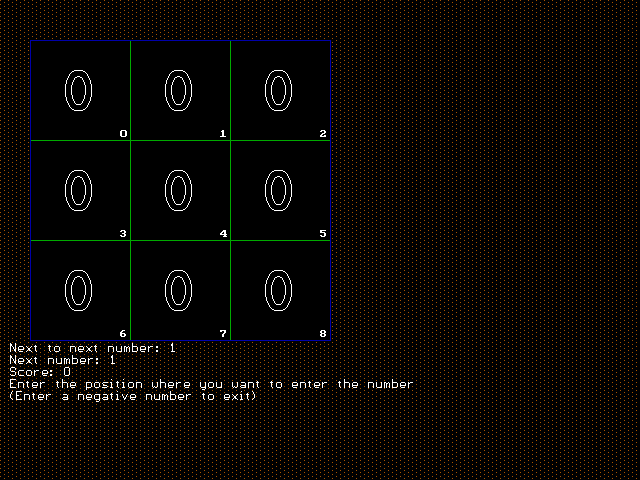
## The Main Menu







## The Game





## End Of Game







# Source Code