Word Search game

A detailed documentation

Abstract

This document is designed to give a detailed information about the project. It will describe how the project is done, related paper work and algorithms designed to solve very particular problem. Moreover, it will include the programming concepts used in making of the project.

WAQAR ASGHAR
L1S21BSSE0119

1. Reading of the project

- Thorough reading to understand what to make (A rough idea) ?
- Detailed reading to understand the working of the project.
- Again reading to understand the complexity of the project.
- Again reading in comparison with the related projects.
- Getting the main ideas like what to make (A clear vision)?

2. Details of the problem statement of the project

After reading the whole project file I come with the following details of the project:

- First of all the **user (player) record** must have to be saved either in a file or in an array every time the user (player) logs in or a new user gets registered to play the game.
- There has to be an array or something that must be used to **display the board** from the file to the console on which there are different characters fir the user or player.
- The **player enters a word** which has to be stored in such a way that **each character** of the word can be accessed separately to search in the given board.
- Once the word is taken from the user , it has to be **searched in dictionary** .
- If word is found in dictionary then it will be **searched on the board** to ensure the location
- Otherwise, the user has to **enter another word**.
- Every time the word is found , **score must be updated**
- **Score has to be displayed** on the screen under board with player name.
- User can **pause or resume** the game , data will not lost in this case , so **dynamically** created array will not be deleted until user enters exit or quit.
- To get **highest score** , all scores of players who have played the game must be stored in an **array** as well to get highest by sorting technique or other suitable logic.
- **Searching** is done very precisely as its very **sensitive stage**.

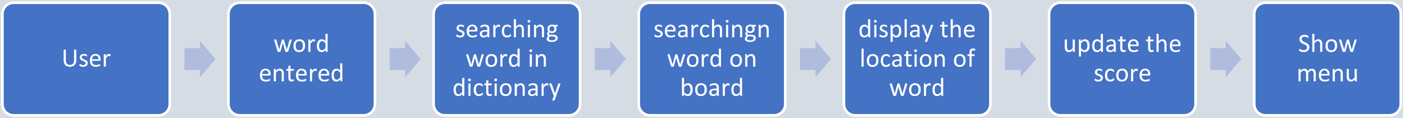


Figure 2.1 Working of the game

3. Real time application related to the project

Here are some real time applications related to this project:

I. Word Crush

https://keygames.com/word-crush-game/

II. Pictoword
https://www.wordgamesfun.com/games/pictoword or https://play.google.com/store/apps/details?id=com.kooapps.pictowordandroid&hl=en&gl=US

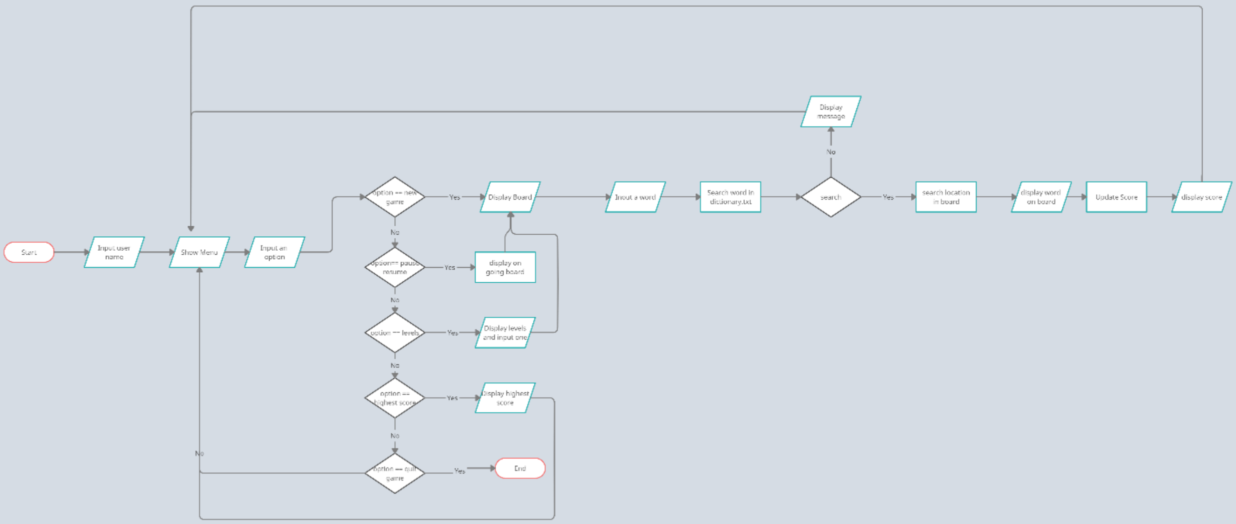
III. Wordwhizzle Search
https://play.google.com/store/apps/details?id=com.apprope.wordsearch&hl=en&gl=US

4. Programming elements used in the project

Here is the list of programming concepts used in the making of the project:

- ✓ Control Structures (If/else)
- ✓ Loops (For loops , while loops)
- ✓ Functions
- ✓ Arrays
- ✓ File Handling
- ✓ Pointers
- ✓ 2D Arrays
- ✓ Dynamic Memory Allocation

Flow Chart Diagram



A basic low-fidelity flowsheet diagram to show the control flow of the software.

5. User-defined functions:

Here is the list of functions used in this project:

- void helloUser(ofstream & , string *);
- int mainMenu(string);
- void Run(int , int [] , int , int);
- void newgame(ifstream & , int & , int & , int);
- void resumegame();
- void levels();
- //void highRecord();
- int calculateRowsInFile(ifstream &);
- int calculateColumnsInFile(ifstream & ,int &);
- void saveGridToArray(ifstream &file1 , char **grid , int &R , int &C);
- bool foundInDictionary(ifstream &);
- void displayGrid(char ** ,int , int);
- int highRecord(ifstream &, int * , int , int);

- void copyData(int * , int * , int);

6. **Arrays used:**

   1-D Dynamic array

   2-D dynamic array

7. **Loops used:**

   While loop

   For loop