**PROJECT SUMMARY**

In this project, I created a game on the programming platform C++. The game is titled ‘Catchphrase’. It is a very simple grid game that is fun to play. The base idea for it is based on the popular game ‘2048’. The objective is pretty straight forward; move the tiles on the grid to acquire the letters and fill up the appropriate blanks in the given phrase.

The game is played out on a 4x4 grid. Each tile can store an alphabet. Initially the player starts with the letter ‘A’. He moves it around on the board. Four movements are possible: up, down, left or right. At the start of each turn a new ‘A’ tile appears. If the player is able to collide two pieces on the board together, he obtains the next letter of the alphabet. For instance, two ‘A’ tiles merge to form a B tile. The phrase gradually gets filled up as he acquires the letters. He plays the game until he successfully collects all the alphabets missing in from the displayed phrase.

The program itself makes full use of graphics and has an animated title screen. Instantly after that, the program directs the player to a menu screen which contains the options to start the game, choose the level, read about the game, how to play it, and the exit option. The data displayed in the ‘how to play’ and ‘about game’ segments are displayed in a text file.

While the game is in progress, the program also keeps tracks of points. At the end of each loop of the game play, the grid gets altered based on the user’s move. Appropriate messages are displayed when the player manages to win the round or if he loses.

**FUNCTION ANALYSIS**

|  |  |
| --- | --- |
| *FUNCTION* | *PURPOSE* |
| int levelselect(); | Based on user’s choice accepts a level and proceeds to game |
| int checkgame(char grid[4][4], char phrase[],char blanks[]); | Checks the status of the game and returns one of 3 values based on it |
| void menu(); | Displays the menu and accepts user’s choice |
| void help(); | Displays the ‘help’ screen |
| void aboutgame(); | Displays the ‘about game’ screen |
| void startgraphics(); | Displays the start screen graphics and animations |
| void spaceleft(char grid[4][4]); | Moves the grid to the left |
| void spaceright(char grid[4][4]); | Moves the grid to the right |
| void spaceup(char grid[4][4]); | Moves the grid up |
| void spacedown(char grid[4][4]); | Moves the grid down |
| void winscreen(int score,int &level); | Displays the ‘you win’ message and score. Helps move on to next level |
| void losescreen(int score,int &level); | Displays the ‘you lose’ message and score. Helps replay current level |
| void startgame(int level=1); | Game function. Calls the display function, moves grid according to choice, calls other game functions |
| void displaygrid(char grid[4][4]); | Displays the grid and contents of grid |
| void checkphrase(char grid[4][4], char blanks[]); | Checks if the phrase is complete |
| void up(char grid[4][4],int &score); | Moves the grid up, collides tiles and modifies score |
| void down(char grid[4][4],  int &score); | Moves the grid down, collides tiles and modifies score |
| void left(char grid[4][4],int &score); | Moves the grid left, collides tiles and modifies score |
| void right(char grid[4][4],  int &score); | Moves the grid right, collides tiles and modifies score |
| void scorecalc(char grid[4][4],int i,int j,int &score); | Calculates the score based on which tiles collide |
| void getphrase(int level,char phrase[],char blanks[]); | Generates a phrase from the stored text file based on the level |

**VARIABLE ANALYSIS**

|  |  |
| --- | --- |
| *VARIABLE* | *PURPOSE* |
| int i; | Used in display and grid moving for loops in the program |
| int j; | Used in display and grid moving for loops in the  programs |
| char grid[4][4]; | It is a 2D array which stores the tiles in the  game and acts like a grid |
| int choice; | Accepts user choice in the menu |
| int lvl; | Accepts level from the user |
| int flag; | Used to check the status of the game. Takes  value of 0 for not game over, 1 if player won, 2  if player lost |
| int x; | Stores a random value corresponding to the  grid’s position |
| int y; | Stores a random value corresponding to the  grid’s position |
| int k; | Used in for loops to move the grid around |
| int score=0; | Stores score of the user. Changes value when  the user plays the game. Initialised to 0 |
| char move; | Accepts user’s move and uses it to change the  grid |
| int tiles=0; | Checks if all tiles are occupied. For every  unoccupied tile, value gets incremented |
| Char phrase[100]; |  |
| Char blanks[100]; |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**SCOPE OF THE PROJECT**

Catchphrase is a very interesting game that is super simple to play. This game has no age limit; it is a game that anyone of any age can play. It’s made interesting because of the inclusion of incomplete phrases which leave the user guessing which letter could possible fit in the blank and what the phrase is. It is an efficient game that allows the user to select the level by choice and calculates score acquired by the user. While the graphics aren’t quite extraordinary, it is the simplicity of the game that appeals to a lot of people.

The scope of the project would be to include more phrases and more difficult levels. The game can be made more interesting by adding different grid sizes like a 3x3 grid or even a 7x7 grid. To make the game more user friendly, there can also be different types of modes such as Timer (where the user’s moves are timed) or Rewind (where the user can be allowed to undo his previous moves).

**LIMITATIONS**

* The program is only made to calculate scores and check for tiles up to the alphabet ‘J’.
* The program will stop as soon as the user gets to the letter J.
* The game does not save progress.

**BIBLIOGRAPHY**

* Computer Science with C++ by Sumita Arora
* [www.stackoverflow.com](http://www.stackoverflow.com)
* [www.bitavoncpp.com](http://www.bitavoncpp.com)
* Google search