VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY-
UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

April 8th 2023

PROJECT'S REPORT
REPORT TOPIC : PROJECT THE MATCHING GAME

UNDER THE GUIDANCE OF:

1. Msi. Bui Huy Thong
2. Msi. Tran Thi Thao Nhi

MEMBERS:

1. Nguyen Hoang Trung Kien - 22127478 - 22CLC08
2. Nguyen Thien Nhan - 22127305 - 22CLC08

SUBJECT : PROGRAMMING TECHNIQUES

Contents

List of Figures

1 Acknowledgement

We would like to express our sincerity to Mr.Bui Huy Thong and Ms.Tran Thi Thao Nhi, lecturers of the Faculty of Information Technology,University of Science for providing us the opportunity to complete our project. This project will be the important development for us to gain the experience of dealing with other difficult projects in the future.

2 Introduction

The Matching Game (commonly known as Pikachu Puzzle Game) includes a board of multiple cells, each of which presents a figure. The player finds and matches a pair of cells that contain the same figure and connect each other in some particular pattern. A legal match will make the two cells disappear. The game ends when all matching pairs are found. Figure 1 shows some snapshots from the Pikachu Puzzle Game. In this project, we will develop a simplified version of this Matching Game by remaking the game with characters (instead of figures). (source : CSC10002 - PROJECT THE MATCHING GAME)

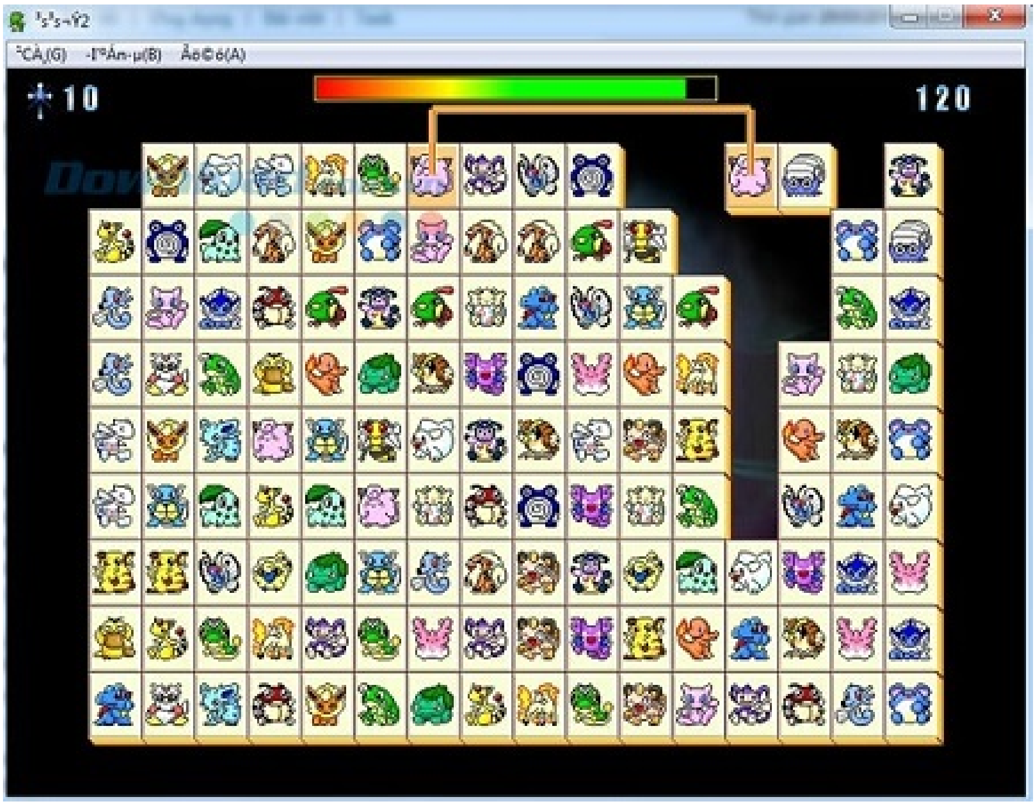


Figure 1: Pikachu game example

3 Project base format

1. IDE used: Visual Studio 2022
2. Code language: C++
3. Operating system: Window 10
4. Report's format: using Latex on overleaf.com to create pdf file.

4 Code

4.1 Library used

- Windows.h: to use functions provided by Windows.
- conio.h: to provide console input/output.
- time.h: provide standardized access to time/date manipulation and formatting.
- fstream: to read file
- string: to work with string data
- mmsystem.h: include file for Multimedia API's
- thread: is a sequence of instructions that can be executed concurrently with other such sequences in multithreading environments, while sharing a same address space. (https://www.tutorialspoint.com/cpp_standard _library/thread.htm)

4.2 Lists of file .h

Note: we also put source code reference at the end of each file .h.

- Control.h
- Gameplay.h
- Interface.h
- Matching.h
- Music.h
- Struct.h

4.3 List of file .cpp and their functions

- Control.cpp: to declare the function goToXY() used for going to one position on the program(line 3 to line 9), add mouse cursor's operation(line 11 to line 17), hide scroll bar(line 26 to line 37), resize the program's window.(line 19 to line 24)
- Gameplay.cpp: to declare functions that create matrix of characters(line 7 to line 28) and render it(line 47 to line 53)(2d array with pointer used). Make delete board function to delete all the board with characters in it then display the background theme(line 30 to line 45). Declare move function to operate the movement and selection of the game(line 55 to line 322).Finally the three funnctions playGame(), displayInstruction() and clearInstruction() to display the game's appearance and start the gameplay with other functions.(line 324 to line 396)
- Interface.cpp: to read and display Leaderboard file(line 6 to line 47) ,then add new player's informations and sorted the players by their results.(line 49 to line 103).Use getPlayerInfo() function to get the player's name(line 105 to line 121). Display status of winning and losing(line 123 to line 152),display and get background of the gameplay(line 154 to line 181). Finally display main menu and set the operations in it(line 183 to line 310).
- Matching.cpp: to check the matchings: check if two cells next to each other valid(line 3 to line 13), check if the line is empty(line 15 to line 65), I check(line 67 to line 103), L check(line 105 to line 125), Z check(line 127 to line 168), U check(line 170 to line 232). Use all the checks(line 234 to line 251). To check if there is 2 cells in the board that have the same character and true with all the cases of matching (line 253 to line 278).
- Music.cpp: to add music into the program
- Struct.cpp: create the appearance of a cell(line 6 to line 12), draw the cell in the program,add color(line 14 to line 42) and delete a cell(line 44 to line 51).
- Main.cpp: main function to operate the whole functions and files above

4.4 Board definition

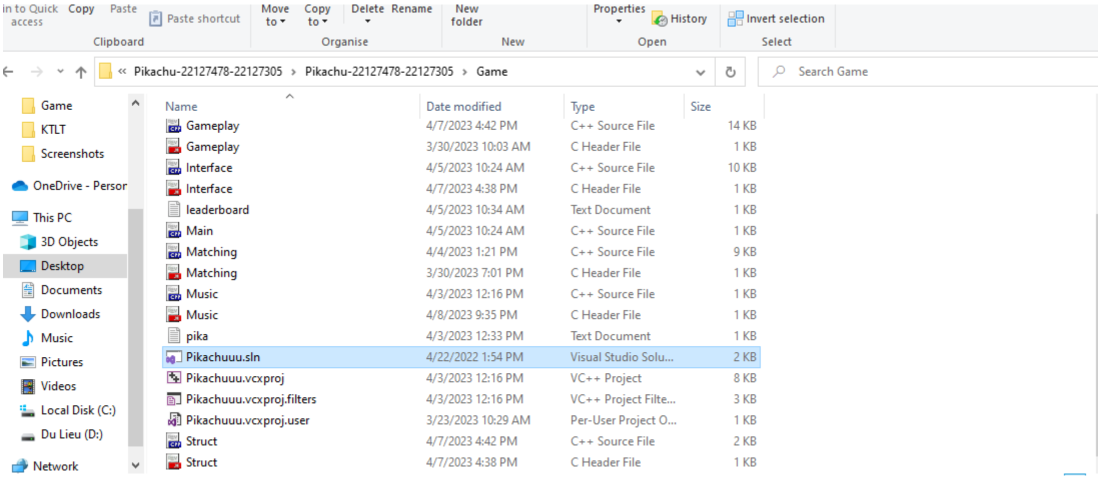
Using 2d array(matrix) with pointer.

5 Game Tutorial

5.1 Program Executing Instruction

To easily execute the program we choose Visual Studio 2022 IDE to code and compile our program. So you need to use this IDE too if you want to compile our program.

First, you need to open the game folder by the following path(Figure 2). Then open the file Pikachuuu.sln.



« Pikachu-22127478-22127305 › Pikachu-22127478-22127305 › Game

Name	Date modified	Type	Size
Gameplay	4/7/2023 4:42 PM	C++ Source File	14 KB
Gameplay	3/30/2023 10:03 AM	C Header File	1 KB
Interface	4/5/2023 10:24 AM	C++ Source File	10 KB
Interface	4/7/2023 4:38 PM	C Header File	1 KB
leaderboard	4/5/2023 10:34 AM	Text Document	1 KB
Main	4/5/2023 10:24 AM	C++ Source File	1 KB
Matching	4/4/2023 1:21 PM	C++ Source File	9 KB
Matching	3/30/2023 7:01 PM	C Header File	1 KB
Music	4/3/2023 12:16 PM	C++ Source File	1 KB
Music	4/8/2023 9:35 PM	C Header File	1 KB
pika	4/3/2023 12:33 PM	Text Document	1 KB
Pikachuuu.sln	4/22/2022 1:54 PM	Visual Studio Solu...	2 KB
Pikachuuu.vcxproj	4/3/2023 12:16 PM	VC++ Project	8 KB
Pikachuuu.vcxproj.filters	4/3/2023 12:16 PM	VC++ Project Filte...	3 KB
Pikachuuu.vcxproj.user	3/23/2023 10:29 AM	Per-User Project O...	1 KB
Struct	4/7/2023 4:42 PM	C++ Source File	2 KB
Struct	4/7/2023 4:38 PM	C Header File	1 KB

Figure 2: File game Folder

After that the Visual Studio will be opened like this. (Figure 3) Then press Ctrl+F5 to directly compile the program.

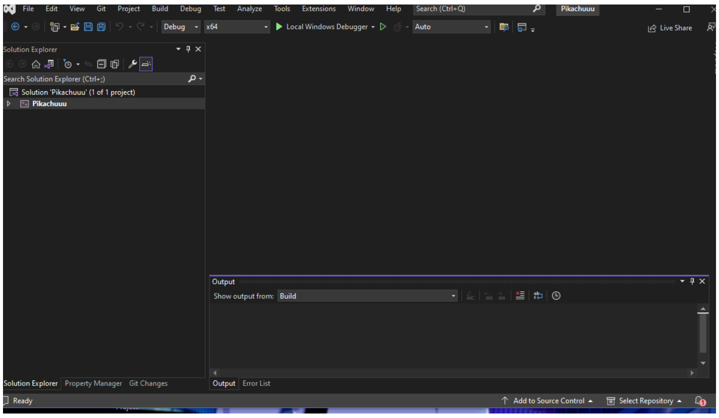

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Pikachuuu
Debug
x64
Local Windows Debugger
Auto
Live Share
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Pikachuuu' (1 of 1 project)
Pikachuuu
Output
Show output from: Build
Solution Explorer Property Manager Git Changes
Output Error List
Ready
Add to Source Control
Select Repository

Figure 3: Visual Studio's appearance

The program should be appear like this(Figure 4):

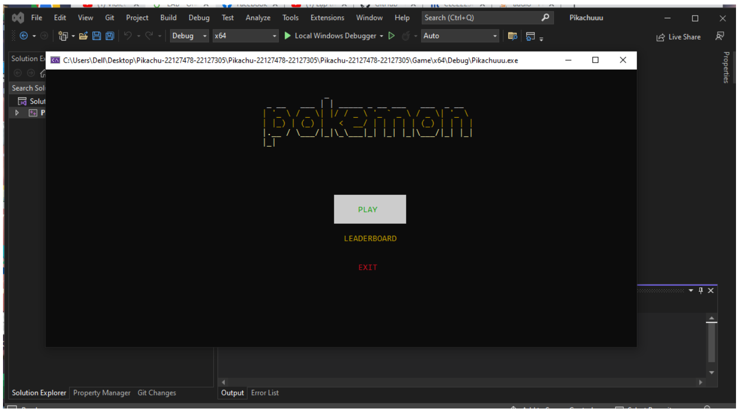

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Pikachuuu
Debug
x64
Local Windows Debugger
Auto
Live Share
C:\Users\Dell\Desktop\Pikachu-22127478-22127305\Pikachu-22127478-22127305\Pikachu-22127478-22127305\Game\x64\Debug\Pikachuuu.exe
pokemon
PLAY
LEADERBOARD
EXIT
Solution Explorer Property Manager Git Changes
Output Error List

Figure 4: Program execution

If you want a clearly view on how to execute the program,we also give you a demonstrate video on how to compile it, you can see the video in the main folder we give you.

5.2 Menu

In the first step you need to run the program,then the main menu will appear with three options for player to choose.(Figure 6)

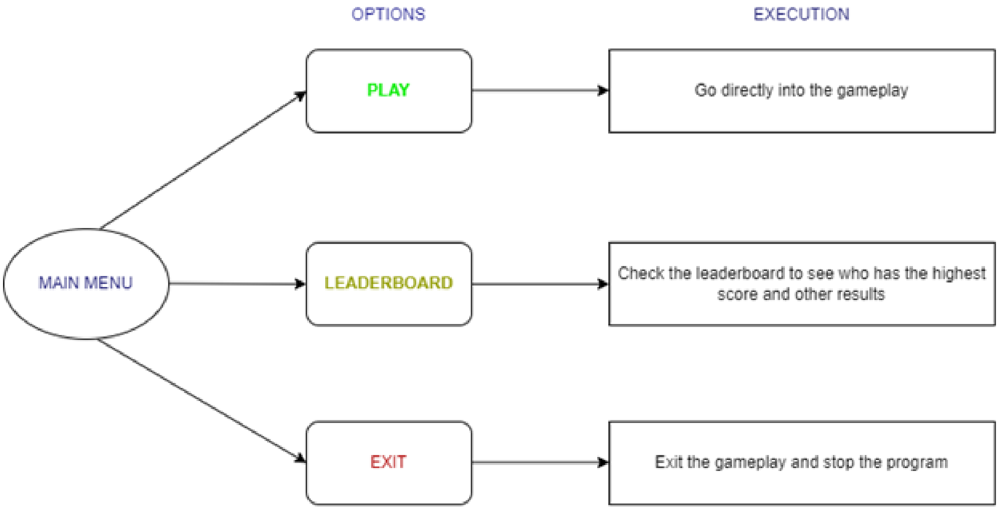

OPTIONS
EXECUTION
MAIN MENU
PLAY
Go directly into the gameplay
LEADERBOARD
Check the leaderboard to see who has the highest score and other results
EXIT
Exit the gameplay and stop the program

Figure 5: Main menu's case diagram

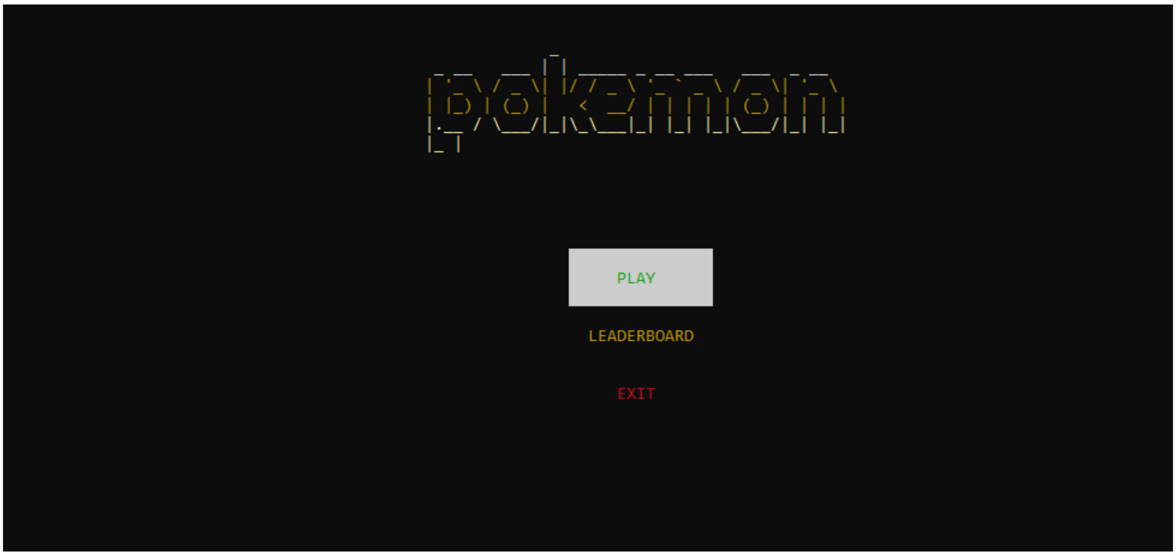
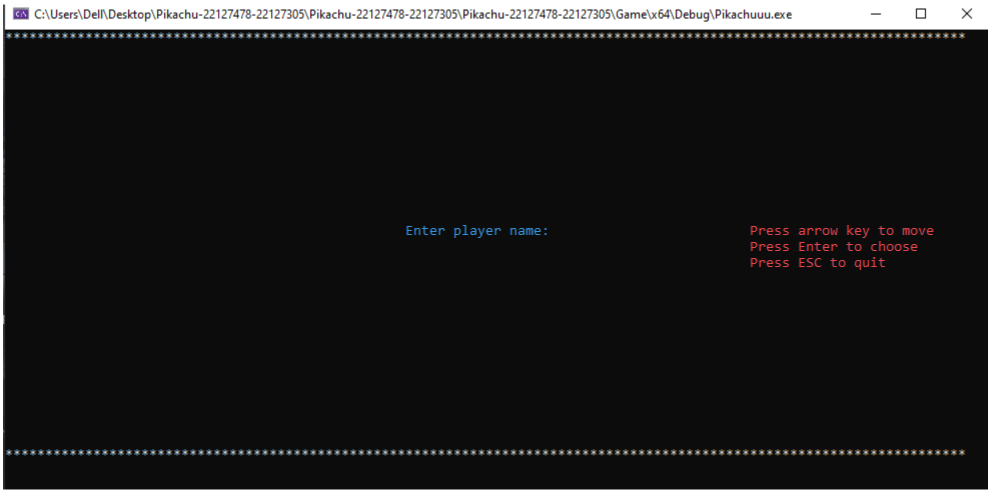


Figure 6: Main menu

5.3 Gameplay

- To go directly into the gameplay,press the "PLAY" button on the MENU display.
- After that the game will tell the player to enter the name.(Figure 7). You can also see the instructions of the game. After done typing name, press ENTER.

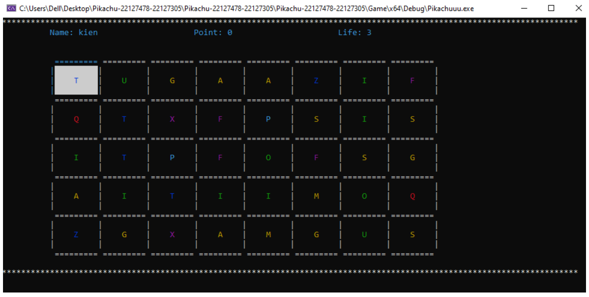


```
C:\Users\Dell\Desktop\Pikachu-22127478-22127305\Pikachu-22127478-22127305\Pikachu-22127478-22127305\Game\x64\Debug\Pikachuuu.exe

                    Enter player name:                  Press arrow key to move
                                                        Press Enter to choose
                                                        Press ESC to quit
```

Figure 7: Typing player's name and instructions display

- Then the program will generate the gameplay appearance with cells of Pikachus. (Figure 8)



```
C:\Users\Dell\Desktop\Pikachu-22127478-22127305\Pikachu-22127478-22127305\Pikachu-22127478-22127305\Game\x64\Debug\Pikachuuu.exe

    Name: kien              Point: 0              Life: 3

    ========= ========= ========= ========= ========= ========= ========= =========
   |    T    |    U    |    G    |    A    |    A    |    Z    |    I    |    F    |
    ========= ========= ========= ========= ========= ========= ========= =========
   |    Q    |    T    |    X    |    F    |    P    |    S    |    I    |    S    |
    ========= ========= ========= ========= ========= ========= ========= =========
   |    I    |    T    |    P    |    F    |    O    |    F    |    S    |    G    |
    ========= ========= ========= ========= ========= ========= ========= =========
   |    A    |    I    |    T    |    I    |    I    |    M    |    O    |    Q    |
    ========= ========= ========= ========= ========= ========= ========= =========
   |    Z    |    G    |    X    |    A    |    M    |    G    |    U    |    S    |
    ========= ========= ========= ========= ========= ========= ========= =========
```

Figure 8: Game Appearance

5.3.1 Movement

- To move between objects, press Arrow keys on keyboard.
- To choose the object, press enter.

5.3.2 Matching

- Choose 2 objects to check if their matching is valid. The matching is valid when there is a logical gap between two objects.
- If the matching valid, the objects will be bold in green.
- Else if it's not valid,the objects will be bold in red.
- Ways of the valid matching

  1. I matching (Figure 8)
  2. L matching (Figure 9)
  3. U matching (Figure 10)
  4. Z matching (Figure 11)

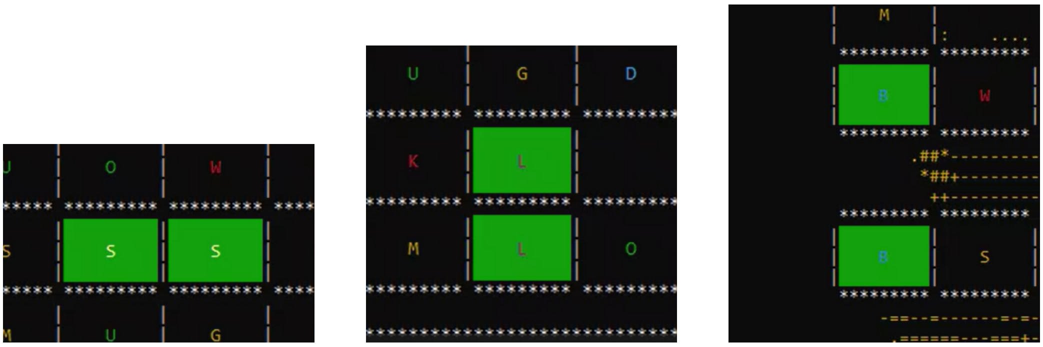


Figure 9: I matching

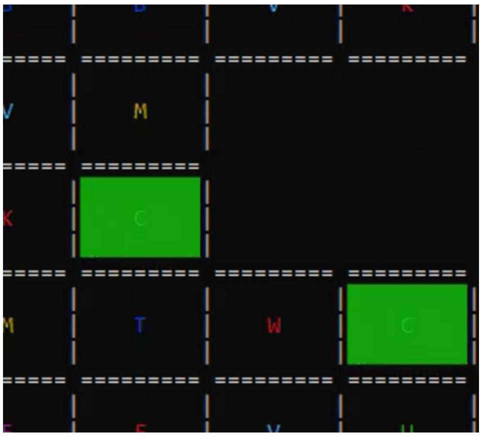


Figure 10: L matching

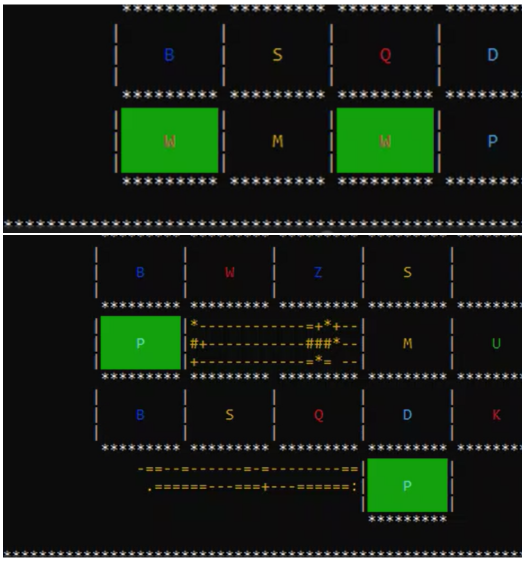


Figure 11: U matching

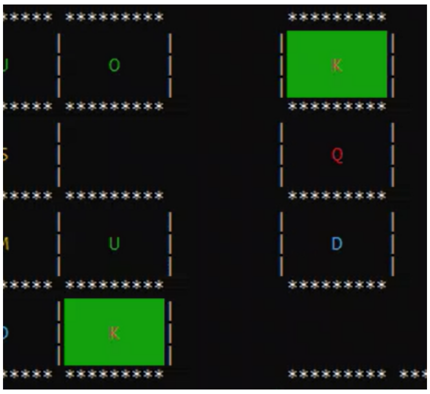
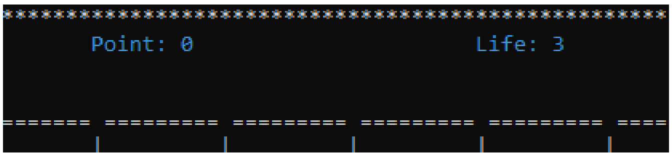


Figure 12: Z matching

5.3.3 Statistic

There will be two kinds of statistic displayed on the top of the game screen.(Figure 12)



```
****************************************************
       Point: 0                      Life: 3



======= ========= ========= ========= ========= ====
       |         |         |         |         |
```

Figure 13: Statistics

1. Point

   Point is the statistic to show the player's result in their game's performance. The default point will be 0. Each valid matchings add 20 points. Invalid matchings will not give the player any point.

2. Life

   Life is the statistics to show how many times the player is allowed to do invalid matchings left. The default life number will be 3. Each invalid matchings will subtract 1. When life point comes down to 0,you'll lose the game.

5.3.4 Game Finish Verify

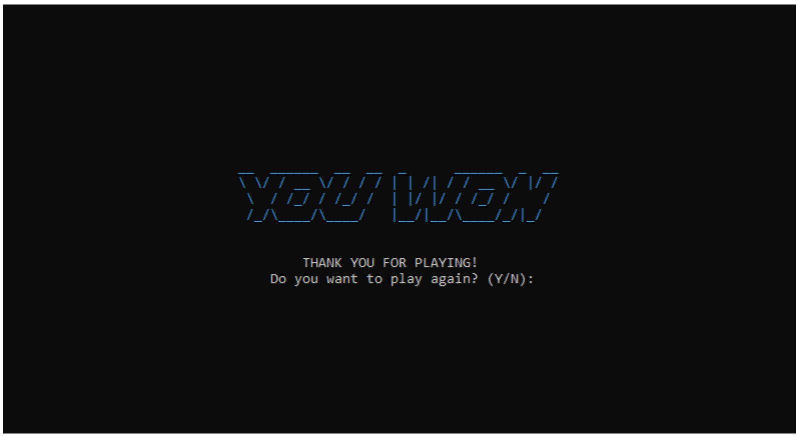
We'll devide into two cases of game finish:

1. Game Win (Figure 13)

   When player's done all the matchings and life point is not 0,the finish game winning verify will appear and ask whether the player want to play again or not. Type y or Y if yes, n or N if no.

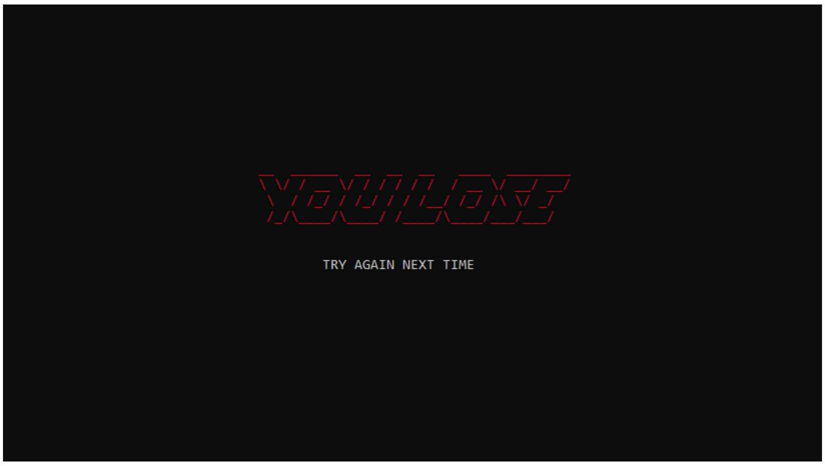
2. Game Lose (Figure 14)

   When life points come down to 0, the cells will slowly disappear and the finish game losing verify will be displayed immediately. After that the program return to Main menu theme.



```
THANK YOU FOR PLAYING!
Do you want to play again? (Y/N):
```

Figure 14: Game Win verify



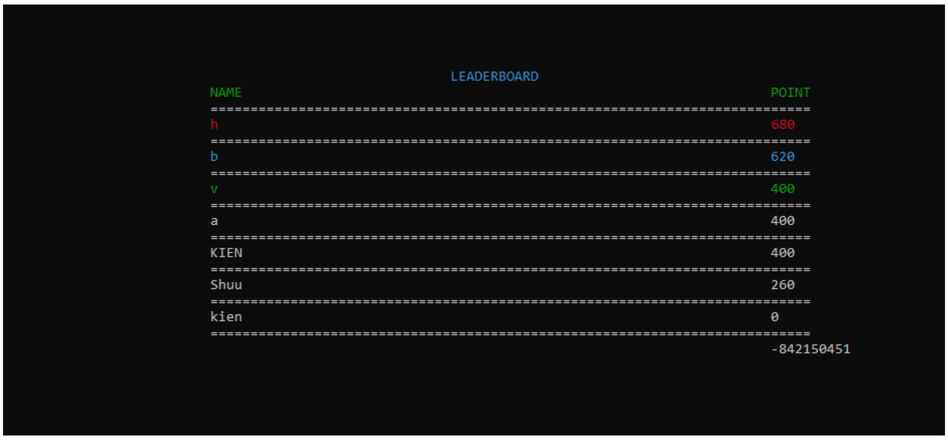
```
TRY AGAIN NEXT TIME
```

Figure 15: Game Lose verify

5.4 Leaderboard

To check Leaderboard,choose the "LEADERBOARD" option on the menu theme.

The program will display the results of the previous players' attendance in game. The results are arranged in descending order. (Figure 15)



```
                               LEADERBOARD
NAME                                                                    POINT
=============================================================================
h                                                                       680
=============================================================================
b                                                                       620
=============================================================================
v                                                                       400
=============================================================================
a                                                                       400
=============================================================================
KIEN                                                                    400
=============================================================================
Shuu                                                                    260
=============================================================================
kien                                                                    0
=============================================================================
                                                                        -842150451
```

Figure 16: Leaderboard

5.5 Exit game

To exit the program, press ESC or choose "EXIT" option on the menu display.

6 Standard Features

6.1 Game Starting

To start game we wrote function initBoard(CELL **board) to create the matrix. The way how this function works is same as how you generate 2d matrix using pointer level 2. First we created rows and columns for matrix (line 8 to line 14,Gameplay.cpp).Then we inputted a character randomly(using char data type) in each cell's data, following the rule that each 2 cells have to match each other to make sure the players are still able to match until the end of game.(line 15 to line 28,Gameplay.cpp)

6.2 I matching

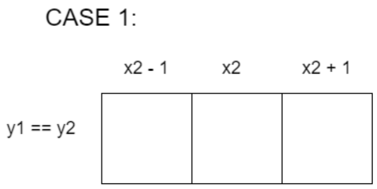
(line 3 to line 103,Matching.cpp)

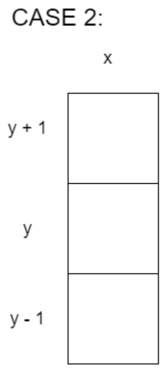
We define position 1 with coordinate(x1, y1) for cell 1 and position 2 with (x2, y2) for cell 2.

For I matching we devided into 2 ways:

1. **Check next(line 3 to line 13,Matching.cpp)**

   We used function checkNext() to check whether 2 cells match with each other or not.





Algorithm:

Case 1:

If y1 = y2 &

$$\begin{cases} x1 = x2 + 1, & \text{(y1 is to the right of y2)} \\ x1 = x2 - 1, & \text{(y1 is to the left of y2)} \end{cases}$$

And the characters in both cells are the same.

⇒ Return true

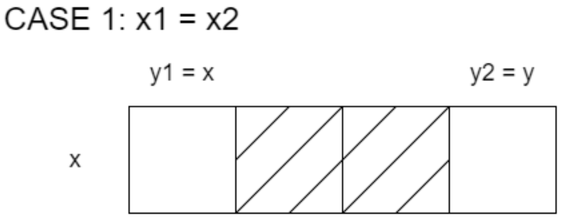
Case 2:

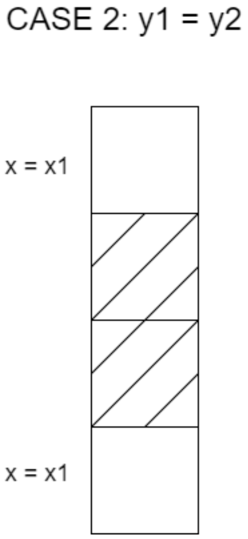
Same as case 1 but swap x to y and y to x.

2. **Check I matching(line 67 to line 103, Matching.cpp)**

   Algorithm:

   We'll go from the cell next to the first selected cell to the cell before the second selected cell , to check between them if other cells exist. If yes return to false. Else return true.

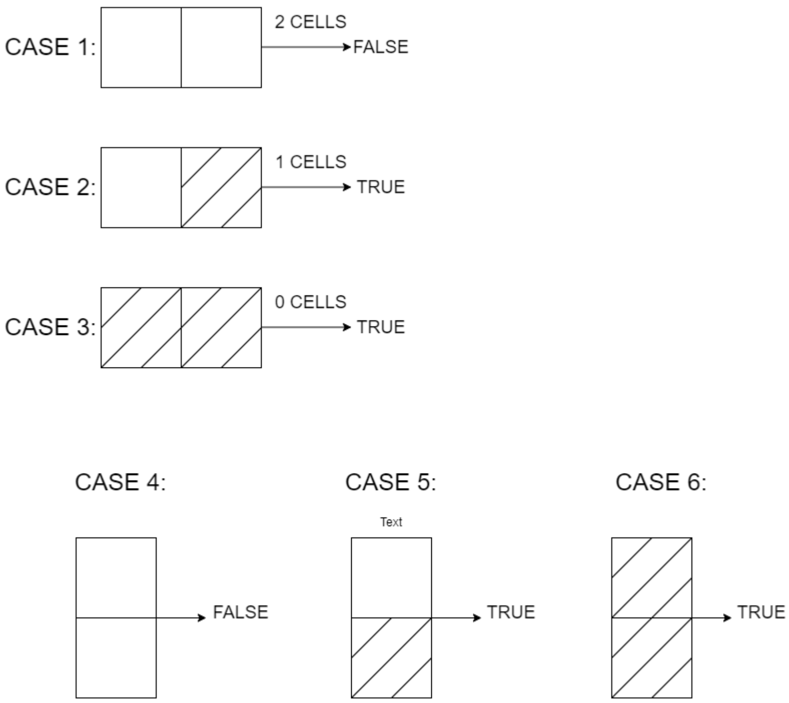




3. **Bonus: Check line(line 15 to line 65,Matching.cpp)**

   Algorithm:

   To check from cell 1 top cell 2 (selected cells) how many cells still exist.



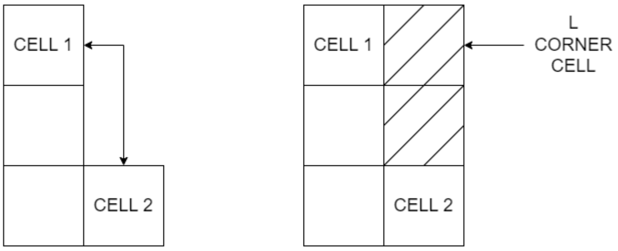
6.3 L matching

(line 105 to line 125, Matching.cpp)

Algorithm:

We check from cell 1 to the L corner cell, if any cells between them exist, return false

We continue to check from L corner cell to cell 2,if any cells between them exist, return false.



6.4 U matching

(line 170 to line 232,Matching.cpp)

Algorithm

- Case 1:

  Step 1: We check if any column is completely empty.

  Step 2: We check if there is a way out from cell 1 to the column we've checked in step 2.

  Step 3: We check if there is a way out from cell 1 to the column we've checked in step 2.

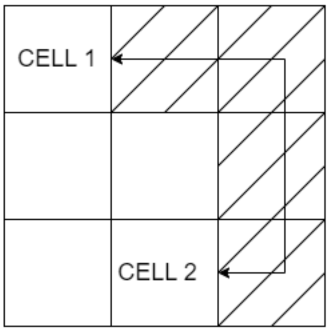
  If all steps are correct then return true.

- Case 2:

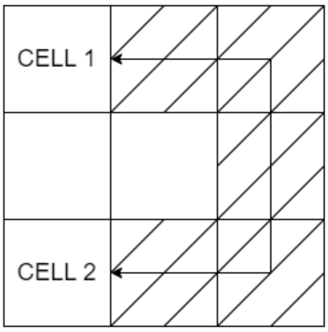
  We check first column and final column. If cell 1 and cell 2 can go to those columns then return true

- Case 3 and 4: Same as case 1 and 2 but instead of columns, we check by rows.

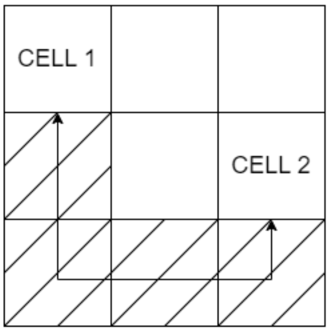
CASE 1:



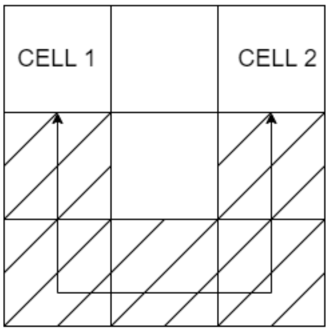
CASE 2:



CASE 3:



CASE 4:



6.5 Z matching

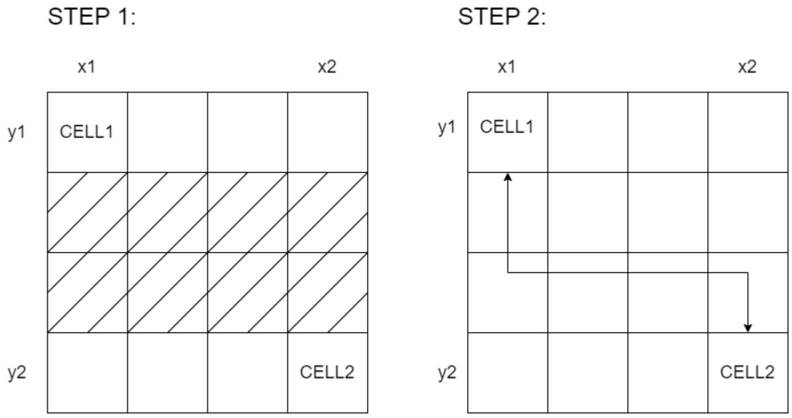
(line 127 to line 168, Matching.cpp)

Algorithm:

Step 1: We check if any rows/columns between cell 1 and cell 2 or not.

Step 2: If step 1 true,continue to check if there is a way from cell 1 and cell 2 to the row/column we detected before.

If all of the statements above true then return true.



6.6 Game Finish Verify

(line 123 to line 152, Interface.cpp)

We define “displayStatus” function to display “YOU WON” if the player won our game and “YOU LOSE” if losing. We use SetConsoleTextAttribute in Windows.h library to change the color of the text that consist of characters in ASCII table. Then, we use goToXY function to go to the position where we display the status. Besides, we also display “THANK YOU FOR PLAYING!” if you won and “TRY AGAIN NEXT TIME” if you lose.

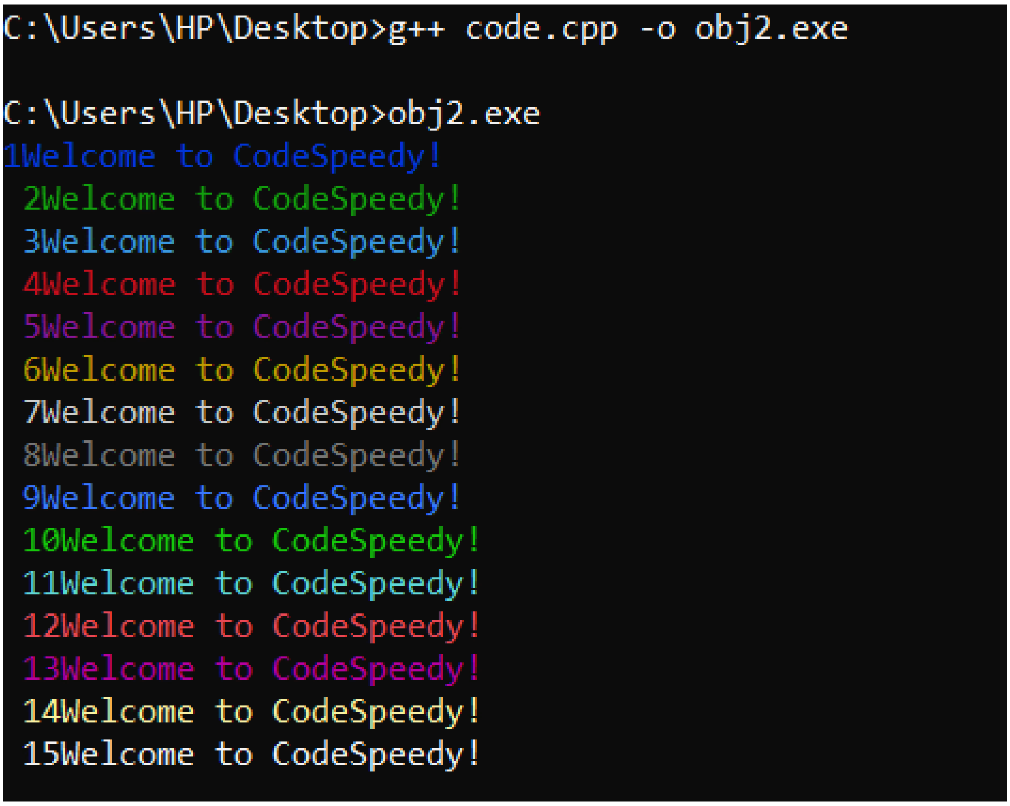
7 Advanced Features

7.1 Color effects

To change the characters' color into different colors, first we include the library Window.h. Then go to the position before the code which we use to output the characters and write:

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),number);

"Number" is where we type the number of color we want to use based on 1-15 color codes are used for text coloring(Figure 16)



```
C:\Users\HP\Desktop>g++ code.cpp -o obj2.exe

C:\Users\HP\Desktop>obj2.exe
1Welcome to CodeSpeedy!
 2Welcome to CodeSpeedy!
 3Welcome to CodeSpeedy!
 4Welcome to CodeSpeedy!
 5Welcome to CodeSpeedy!
 6Welcome to CodeSpeedy!
 7Welcome to CodeSpeedy!
 8Welcome to CodeSpeedy!
 9Welcome to CodeSpeedy!
 10Welcome to CodeSpeedy!
 11Welcome to CodeSpeedy!
 12Welcome to CodeSpeedy!
 13Welcome to CodeSpeedy!
 14Welcome to CodeSpeedy!
 15Welcome to CodeSpeedy!
```

Figure 17: 1-15 basic color codes are used for text coloring

7.2 Sound effects

To add sound effects in the program we used thread, Windows.h and mmsystem.h libraries. Then we use the loop to run PlaySound() function,the add the music's name into that function. The music file must be put in the same folder with the game file.(Music.cpp)

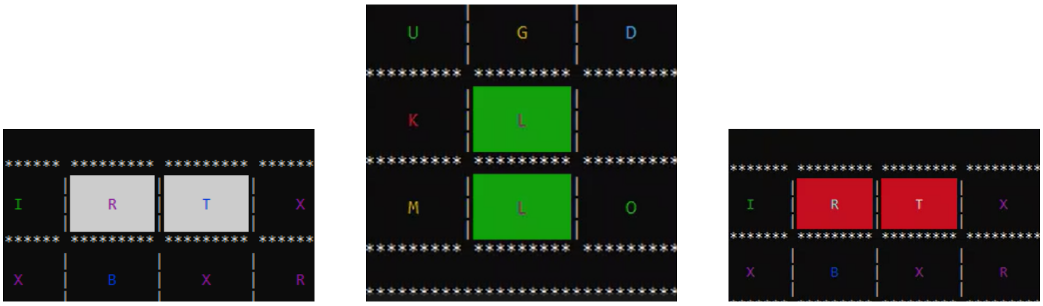
Play sound function:

PlaySound(TEXT(”filename.wav”), NULL, SND_FILENAME | SND_LOOP);

7.3 Visual effects

- **To display the selected cell, valid and invalid cells(line 55 to line 332, Gameplay.cpp):**

  We check value of “temp” by using “_getch()” function and check to see if it is different to 0 and 224 (the arrow keys). Furthermore,we use drawbox(int color) function to highlight the cell. If the value of “temp” is equal to the value of “ESC” key (27), “status” will be assigned to 2. Else if it is “ENTER” (13), we will check to see if the present cell is the same position as the cell which was chosen (is saved to selectedPos array). If true, this cell is changed to white, the “isSelected” flag return 0, the “couple” is 2 and the value of “selectedPos” return {-1, -1}. Turns to play of the player is decreased and display the remaining plays. If false, the position of the cell which is chosen is stored in selectedPos array, the cell is changed to white and the value of “couple” is fallen off. If “couple” is equal to 0, it means there are 2 cells were chosen. Then, the program will check if they have the same value. If true, their color will be highlighted in green,then will be removed from the board. Next, the background of this position is emerged. If false,their color will be highlighted in red and it will display the remaining plays. After that, return the original status and the pointer points to the next cell in the board. If “temp” is the arrow keys, when a cell is selected, it becomes highlighted.



(a) Selected cells (b) Valid cells (c) Invalid cells

Figure 18: Color of cells

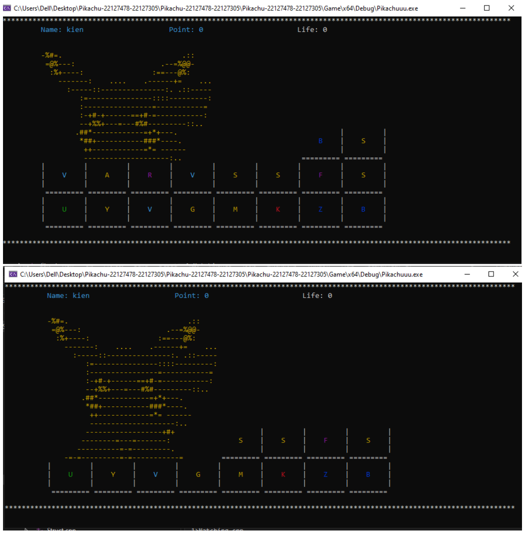


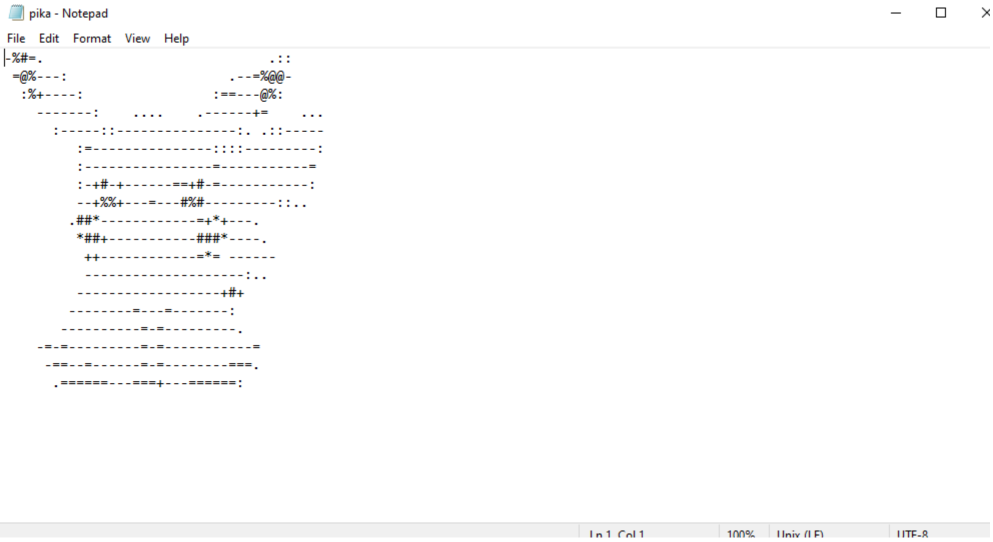
Figure 19: Deleting cells

- **To delete all cells when losing(line 30 to line 45, Gameplay.cpp):**

We used deleteBoard function to delete all cells. The cells will be deleted from left to right and from top to bottom.The background will also be displayed. Only delete cells when the cell is exist(using .isvalid to check), then delete cell using deleteBox() function. After that we used displayBackground() function to display the background of game.(Figure 19)

7.4 Background

In the program we inserted the background behind the gameplay. Only after you do the matching,you can see it.



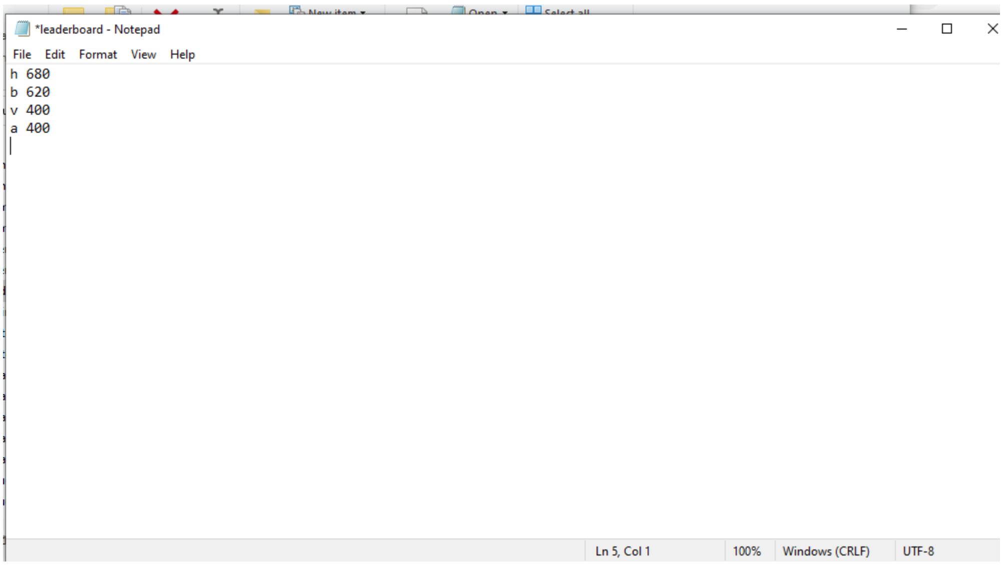
```
-%#=.                            .::
 =@%---:                    .--=%@@-
  :%+----:                :==---@%:
    -------:    ....    .------+=    ...
      :-----::---------------:. .::-----
         :=---------------::::---------:
         :----------------=-----------=
         :-+#-+------==+#-=-----------:
         --+%%+---=---#%#---------::..
        .##*------------=+*+---.
         *##+-----------###*----.
          ++------------=*= ------
          --------------------:..
         ------------------+#+
        --------=---=-------:
       ----------=-=---------.
    -=-=---------=-=-----------=
     -==--=------=-=--------===.
      .======---===+---======:
```

Figure 20: pika.txt

To read and display the file pika.txt(Figure 20) with color in the program we used function readBackground() and displayBackground(line 154 to line 181,Interface.cpp)

7.5 Leaderboard

The idea on how our leaderboard works based on reading file operation. Here we used .txt file to write the results of players. Then to display the leaderboard we just need to read the file leaderboard.txt .(Figure 21)



```
h 680
b 620
v 400
a 400
```

Figure 21: leaderboard.txt file

To read and display the file leaderboard.txt we used function readLeaderBoard() (line 6 to line 46, Interface.cpp), furthermore to add new results we used writeLeaderBoard(player p)(line 48 to line 102,Interfere.cpp). In readLeaderBoard() function,we wrote Leaderboard title and drew a table first. Then the function reads the information of point and name. Finally we'll just display the information visually.

8 References

C++ CODE:

1. Pham Van Minh(2022), Code game Pikachu on console by C++, Youtube, Github, https://www.youtube.com/watch?v=I2FbZyldJuw

   https://github.com/PhVanMin/Pikachuuu

2. C++ Library thread, tutorialspoint,

   https://www.tutorialspoint.com/cpp_standard_library/thread.htm

3. YANNAM SATYA AMRUTHA, Color text output in Console in C++, CodeSpeedy, https://www.codespeedy.com/color-text-output-in-console-in-cpp/

4. (2012) PlaySound in C++ Console Application, stackoverflow,

   https://stackoverflow.com/questions/9961949/playsound-in-c-console-application

5. 22127343 Huynh Minh Quan 22CLC08 (2023), Music.cpp, Music.h.

LATEX CODE:

1. koleygr(2017), Page Border for cover page in LaTeX, StackExchange,

   https://tex.stackexchange.com/questions/407812/page-border-for-cover-page-in-latex?

2. Josh Cassidy (August 2013), How to Write a Thesis in LaTeX (Part 3): Figures, Subfigures and Tables

   https://www.overleaf.com/learn/latex/How_to_Write_a_Thesis_in_LaTeX_(Part_3)%3A_Figures%2C_Subfigures_and_Tables

OTHER REFERENCES USED:

1. CSC10002 - PROJECT THE MATCHING GAME (2023)

2. Pikachu games online, wd-96.top,

   https://www.wd-96.top/products.aspx?cname=pikachu+games+online