

University of Science

Viet Nam National University Ho Chi Minh City

Introduction to Computer Science - CS161

FINAL PROJECT: MINESWEEPER

Student: Trinh Nguyen Thao Vi

ID: 22125120

A. Features (Already done)

Basic features:

1. Drawing the board (including cells, mines and flags)
2. Changing the state of each cell when it is selected
3. Timer
4. Functions: new game, change board dimensions. Select a function from screen
5. Randomizing positions of the mines
6. Receiving new position of mouse (to select a cell) from screen
7. Saving the current state of the board and loading in the next playing session
8. Saving high scores and showing high scores

Other features:

1. Using mouse (left, right and double click) to select a cell or an option
2. Rendering images
3. Using struct

B. Files in folder “Src” and their functions

Header files:

- **graphics.h**: storing functions to render images, to work with mouse
- **myFunctions.h**: storing function declarations

C++ files:

- **myFunctions.cpp**: storing function definitions
- **minesweeper.cpp**: storing main program, which is run to have the executable file

Text files:

- **lastgame.txt**: saving state of the board to load in the next playing session
- **statistics.txt**: saving high scores

Executable file:

- **minesweeper.exe**: playing the game

Folder “resources”: storing images (jpg file)

C. Structure of the program

I. Important arrays and variables

- **char array[50][50]**: save position of bombs, number of bombs around a cell
 - o `array[x][y] = 'B'`: cell (x, y) has a bomb inside it
 - o `array[x][y] = '0', '1', '2', ..., '8'`: cell (x, y) does not have a bomb, and `(int)(array[x][y]) - 48` is the number of bombs around it
- **char state[50][50]**: save the present state while playing
 - o `state[x][y] = 'N'`: cell (x, y) has not been uncovered yet
 - o `state[x][y] = 'U'`: cell (x, y) has been uncovered already
 - o `state[x][y] = 'F'`: cell (x, y) has been marked with a flag
- **int Unlocked**: number of cells which are unconvered
- **int time**: playing time
- **int idLevel**:
 - o `idLevel = 0`: beginner level (9x9 cells, 10 bombs)
 - o `idLevel = 1`: intermediate level (16x16 cells, 40 bombs)
 - o `idLevel = 2`: expert level (25x25 cells, 99 bombs)
 - o `idLevel = 3`: user-custom level (maximum 25x25 cells, number of bombs is always less than number of cells)

II. Structs

- **struct level**: saving information of levels. Members:
 - o **int size**: size of the board
 - o **int bomb**: number of bombs
 - o **int yFirst**: y-coordinate of position on screen where we start printing out the board (different

levels have different positions, so the board is in the middle of screen)

- **int xFirst**: x-coordinate of position on screen where we start printing out the board
 - **int sizeCell**: size of a cell of the board (different levels have different sizes, the less number of cells the bigger sizes of cells)
- **struct Image**: saving information of images. Members:
 - **string name**: filename of image
 - **int x**: x-coordinate of top-left corner of image on screen
 - **int y**: y-coordinate of top-left corner of image on screen

III. Functions

- **void setupPixel(int _smolPiece)**: set up size, coordinates of buttons and levels
- **void printImage(Image img)**: print out images
- **void drawThickRect(int x, int y, int weight, int height, int COLOR, int thickness)**: print out rectangles having thick border
- **bool isInside(int X, int Y, Image img)**: check whether coordinates (X, Y) is inside the image or not
- **void clearMouse()**: clear state of mouse
- **void printTime(int time, int yFirst, bool flag)**: print out playing time and number of flags

- **void about()**: show information about the author
- **int menu(bool kind)**: print out options and allow user to choose
- **void printCus(int num, int xFirst, int yFirst, bool kind)**: print out size and number of bombs of user-custom level
- **bool customLevel()**: allow user to custom level (maximum 25x25 cells, number of bombs is always less than number of cells)

- **void printScore(int num, int id, bool time):** print out playing time and size of board
- **void statistics(vector<pair<int, int>> timePlay):** print out top 5 fastest playing time
- **void randomMine(char (&array)[50][50], int idLevel):** randomize positions of the mines
- **void init(char (&array)[50][50], char (&state)[50][50], int idLevel):** initialize value of board and call the randomMine function
- **void printFirst(char (&array)[50][50], int idLevel):** print a new board, not open any cells
- **void printFinal(char (&array)[50][50], char (&state)[50][50], int idLevel):** open all cells and print out the board
- **void printLastGame(char (&array)[50][50], char (&state)[50][50], int &flag, int &Unlocked, int &time, int idLevel):** print last state of the board (last game)
- **void bfs(int i, int j, char (&array)[50][50], char (&state)[50][50], int &Unlocked, int idLevel):** use breadth-first-search algorithm to print out area not having bombs
- **bool win(char (&array)[50][50], char (&state)[50][50], int idLevel):** check whether user win or not
- **void saveGame(char (&array)[50][50], char (&state)[50][50], int &time, int &flag, int &Unlocked, int idLevel):** save current state of the board (for loading in the next playing session)
- **void playing(char (&array)[50][50], int &flag, char (&state)[50][50], vector<pair<int, int> > &timePlay, int &Unlocked, int &time, int idLevel):** allow user to uncover a cell, mark a cell with flag, remove flag from a cell, open cells around a cell if user think they mark exactly bomb cells, notify if user win or lose

D. Playing Instructions

Main menu

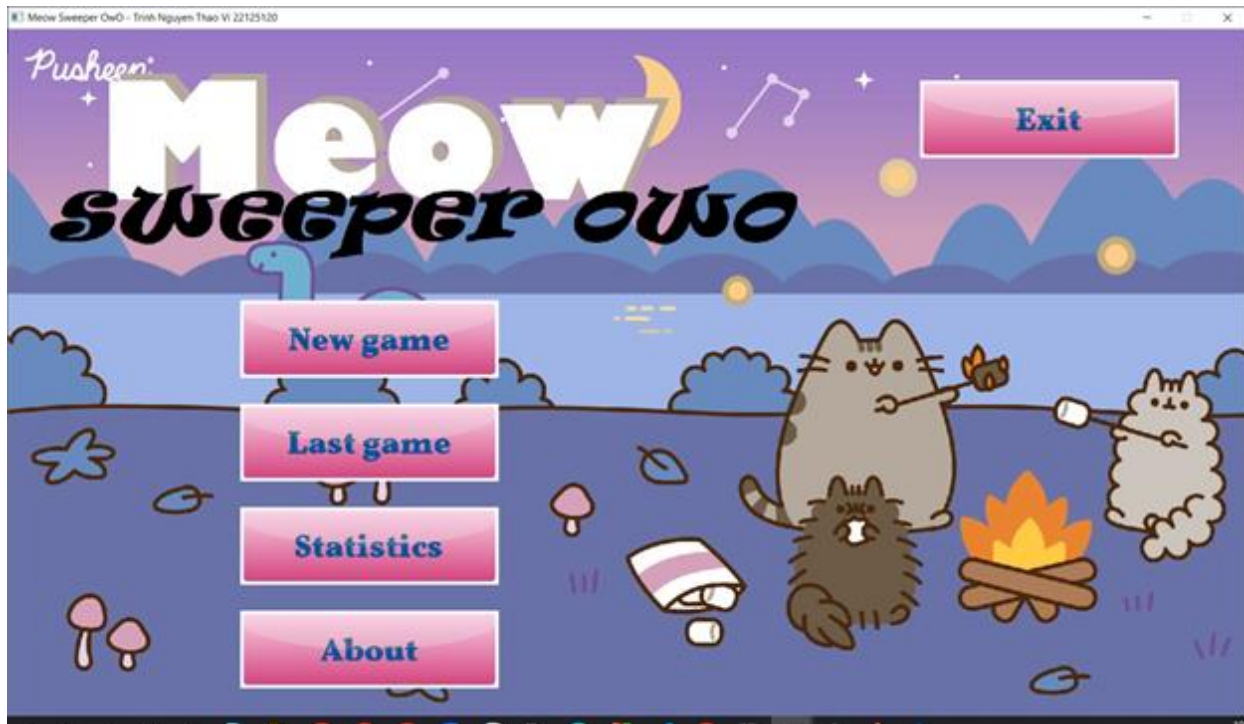


Figure 1. The "main menu"

The "main menu" includes 5 options:

- **New game:** Let user play a new game
- **Last game:** play the previous saved game
- **Statistics:** show high scores
- **About:** show information of author
- **Exit:** close the game

I. New game

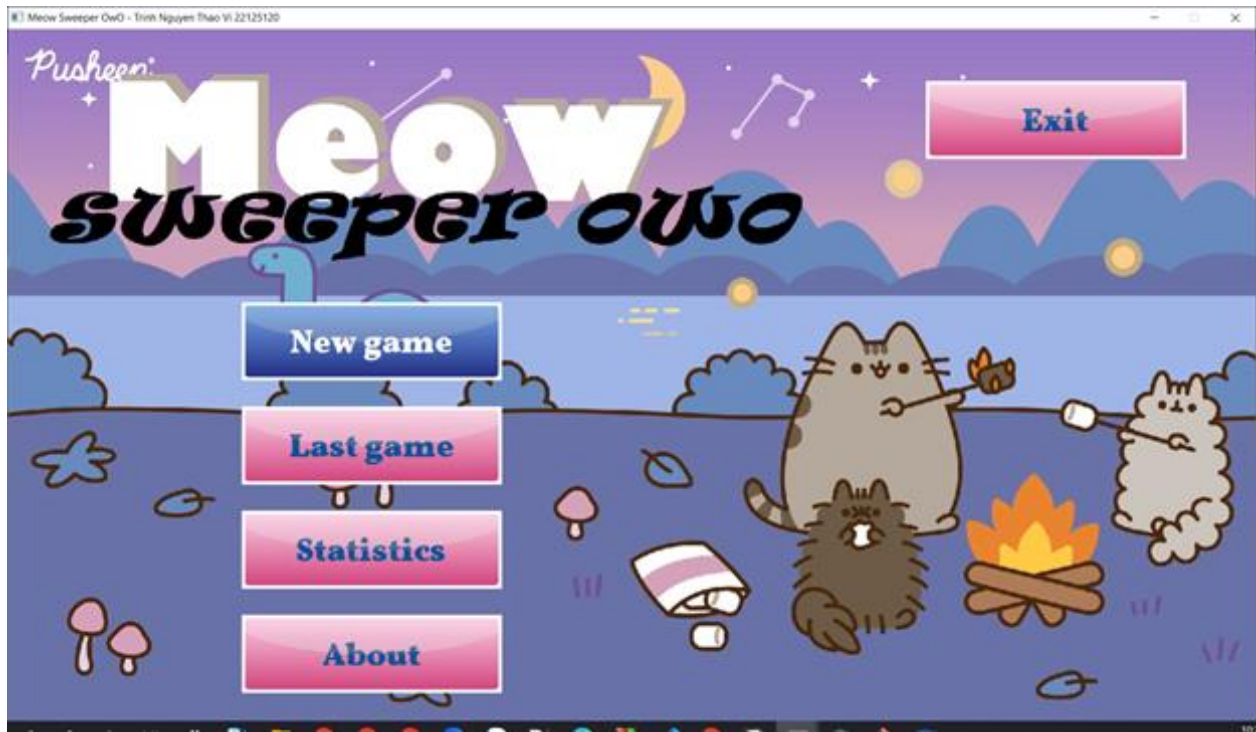


Figure 2. The "main menu" when user choose "New game"

When user clicks on **"New game"**, the program will move to the **"level menu"** to let user select game level



Figure 3. The "Level menu"

"Level menu" includes 4 options:

- Beginner
- Intermediate
- Expert
- Custom
- Home

The "Home" button lets the user return to the "main menu"

The first 3 options "Beginner", "Intermediate", "Expert" let user play a new game with fixed size of board and number of bombs. After clicking on one of these options, the program will move to "playing screen", user can play a new game immediately

- Beginner: 9x9 cells, 10 bombs
- Intermediate: 16x16 cells, 40 bombs
- Expert: 25x25 cells, 99 bombs

The **“Custom”** option lets the user choose the size of the board and the number of bombs



Figure 4. The "level menu" when user choose "Custom"

After user clicks the **“Custom”** option in the **“level menu”**, the program will move to the **“custom screen”**



Figure 5. The "custom screen"

In this **"custom screen"**, default size of board and number of bombs are both 10. User can choose size of board and number of bombs by clicks on buttons:

- **"-"/"+"**: decrease or increase size of board / number of bombs by 1
- **"-5"/"+5"**: decrease or increase size of board / number of bombs by 5
- **"-9"/"+9"**: decrease or size of board / number of bombs by 9

Note that the maximum size of board is 25x25 cells, number of bombs is always less than number of cells

After that, user can play the game by clicking on the **"Play"** button, the program will move to **"playing screen"**

The **"Home"** button lets the user return to the **"main menu"**

II. Last game

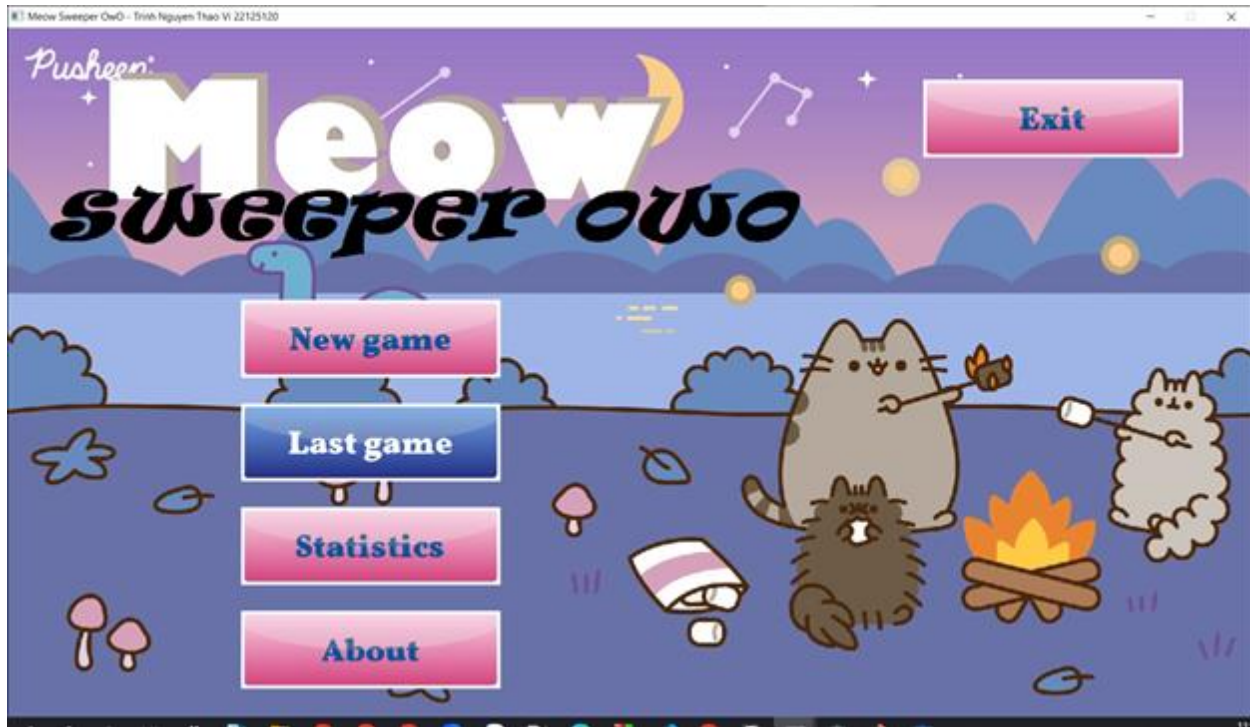


Figure 6. The "main menu" when user choose "Last game"

When user clicks on **"Last game"**, the program will move to **"playing screen"** and let user play the previous saved game. Number of flags will be the same with the last game. Also, time will continue to be counted (start with time of the last game)

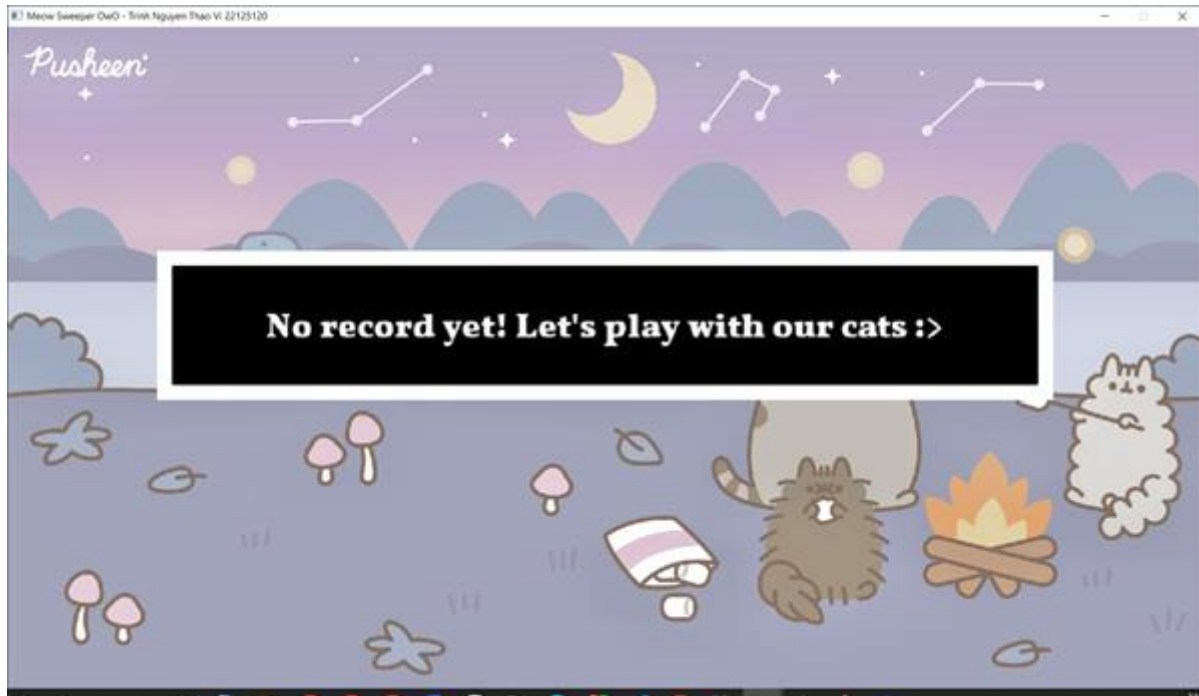


Figure 7. If no game was saved, notification appears

If no game was saved before, the notification will appear. Then the program will bring user back to "main menu"

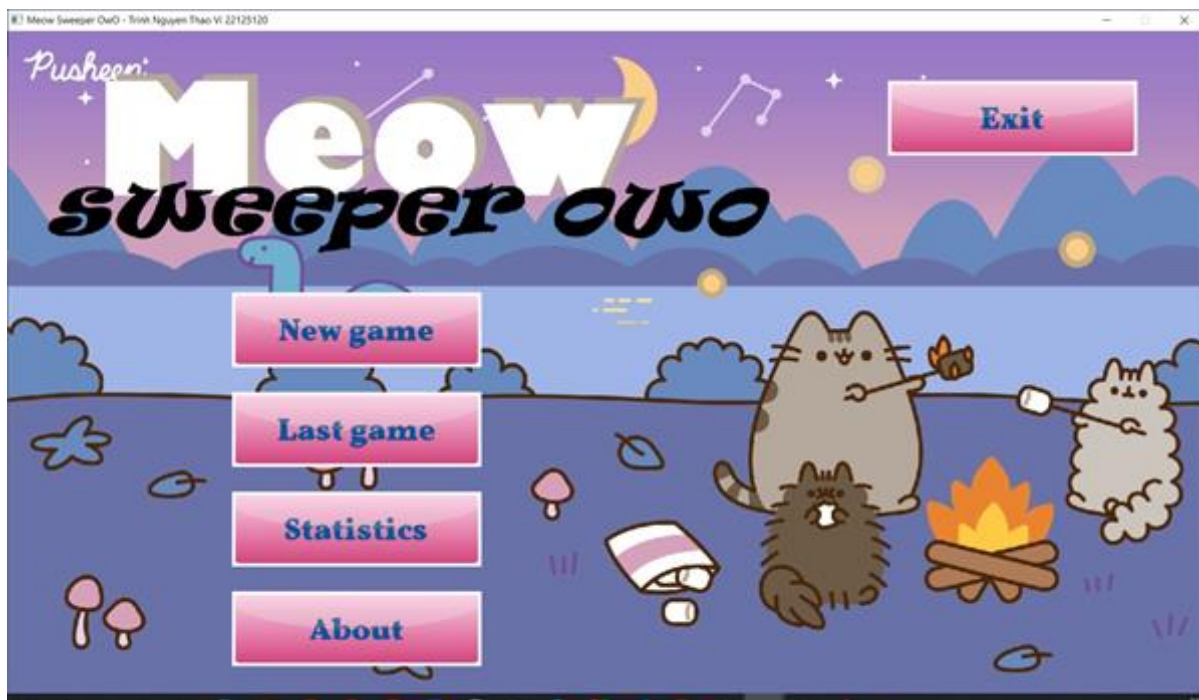


Figure 8. The "main menu"

III. Playing game

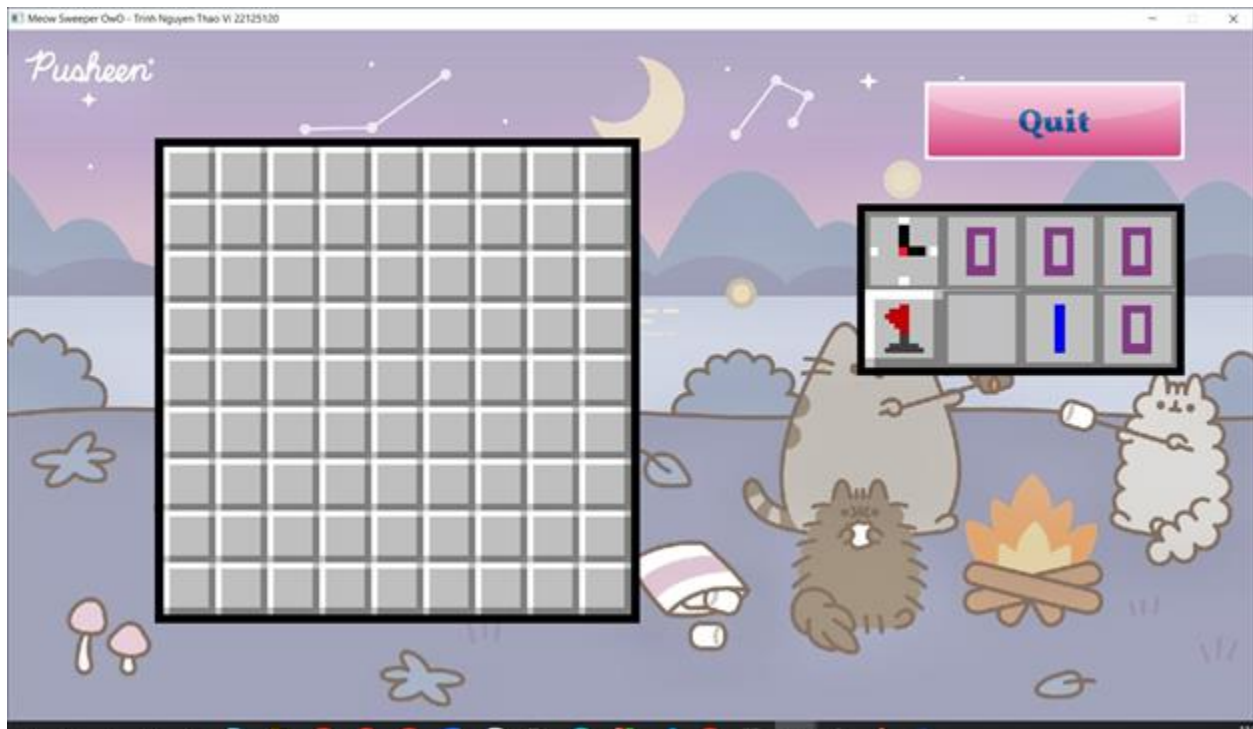


Figure 9. Playing screen

After choosing level or customizing user's own level in "level menu", or choosing "Last game" in "main menu", the program will move to "playing screen". User can play the game now! The game rules are similar to [minesweeper](#)



Figure 10. Playing

- Uncover a cell: Use left-mouse to click on that cell
- Mark a cell predicted to contain bomb with flag: Use right-mouse to click on that cell
- Unmark a cell: Use right-mouse to click on that cell
- Chording:
 - Double-click on an uncovered cell to uncover all 8 adjacent cells if its number **equaling** to number of adjacent cells marked with flags.
 - Note: if user use chording but cells marked by flags are not bomb cells, user will lose immediately. Be careful!

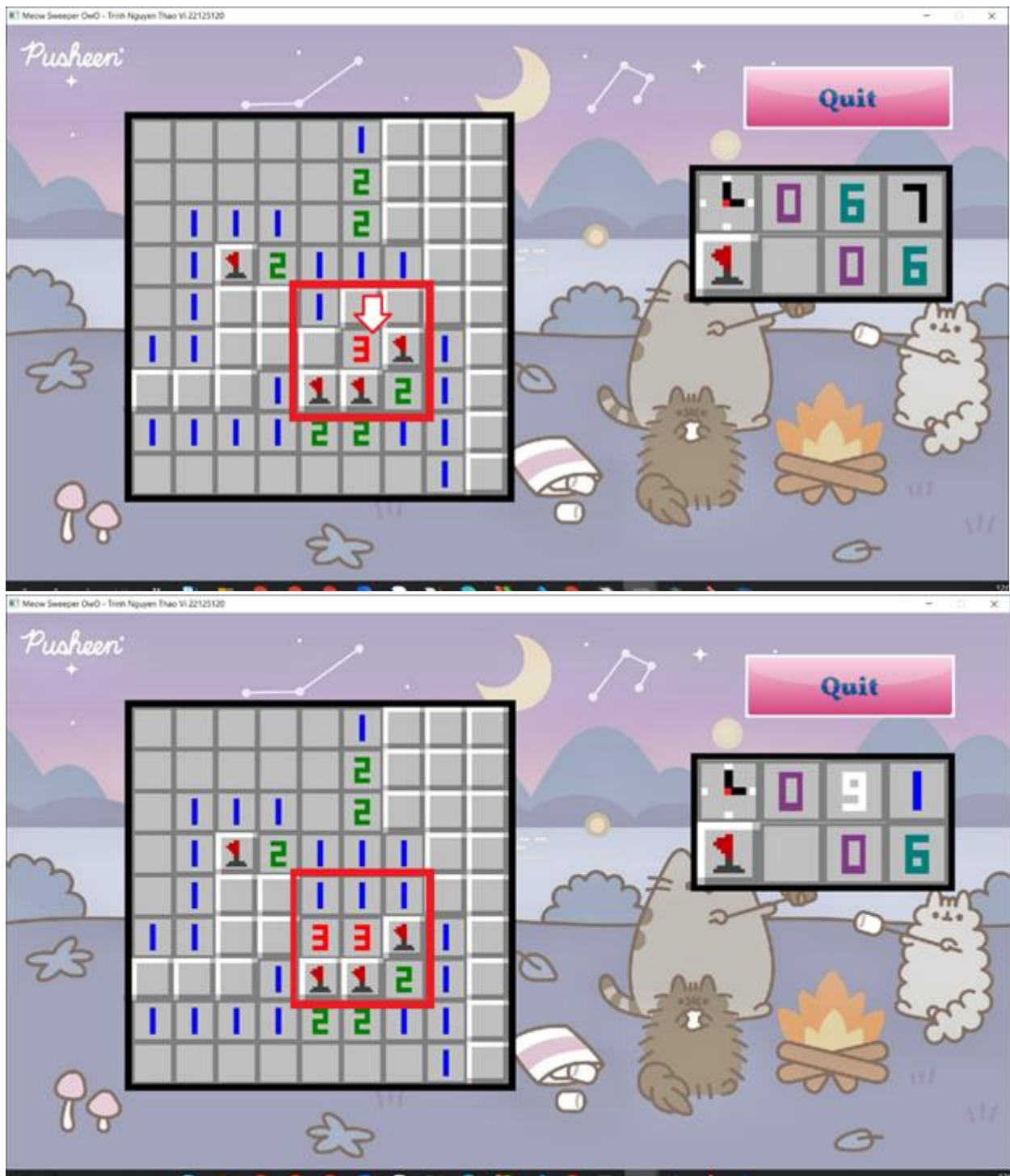


Figure 11. Chording

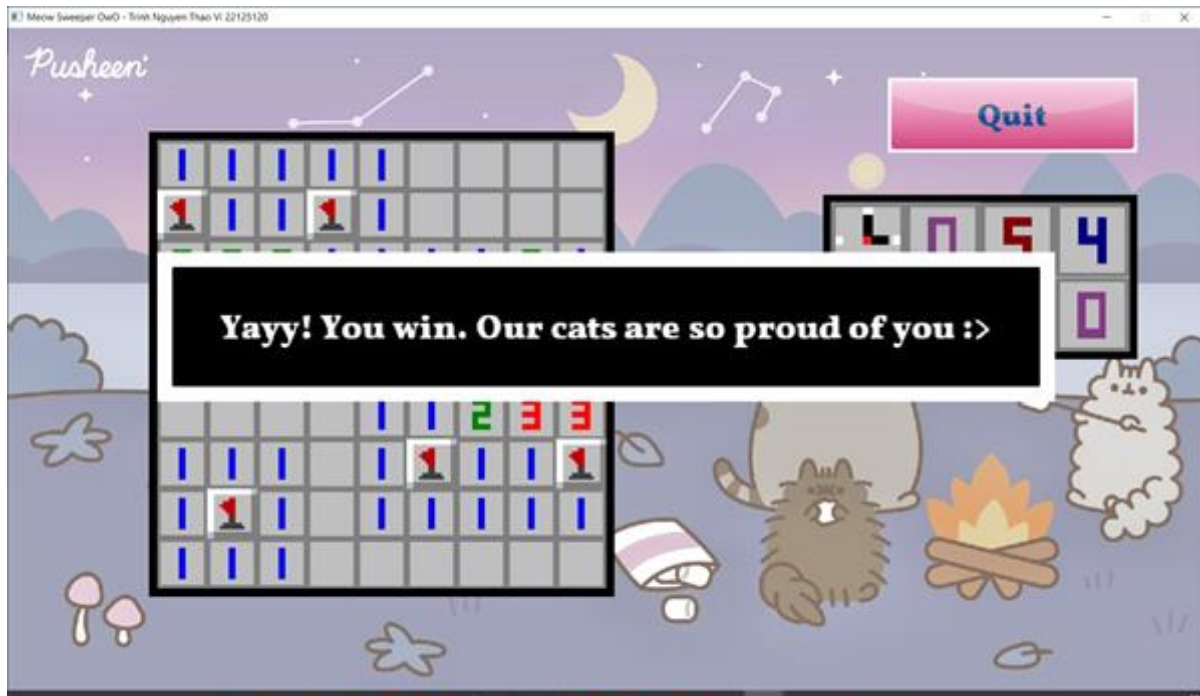


Figure 12. Notification appears when user win

If user win (uncover all cells without clicks on bomb cells) or lose, all cells will be opened and notification will appear. If user win, the time and size of board will be saved if it's one of the top 5 fastest time

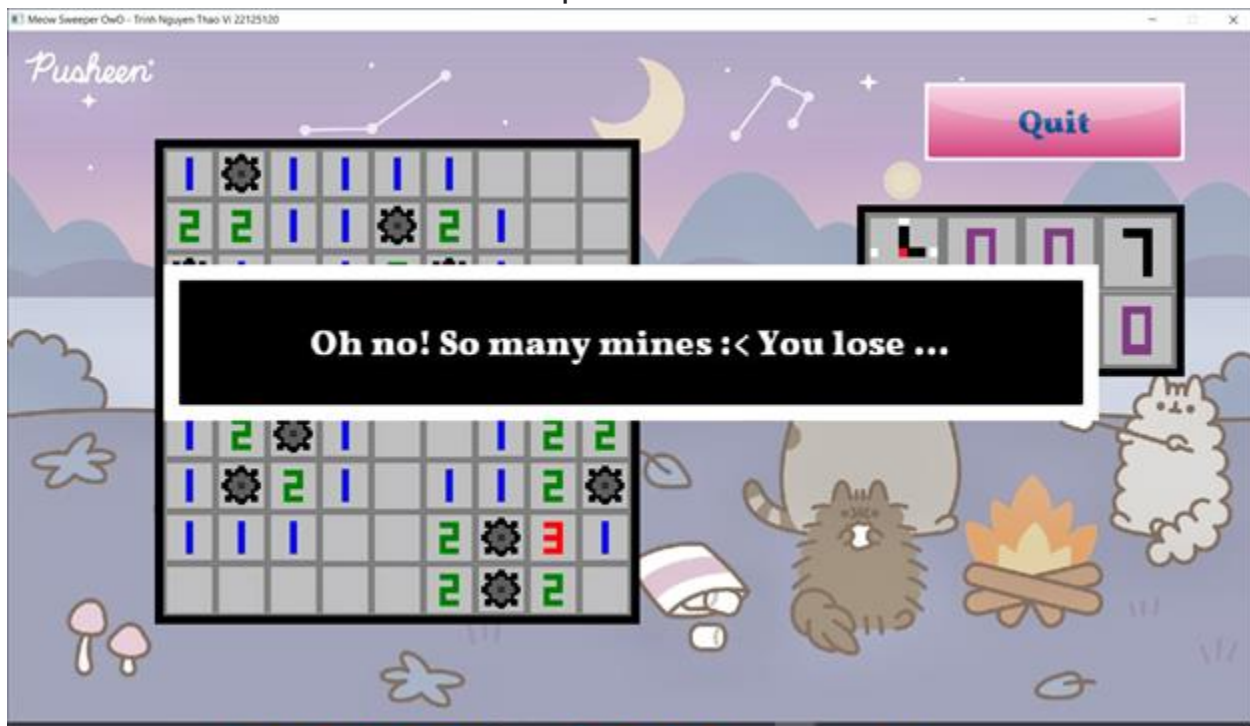


Figure 13. Notification appears when user Lose

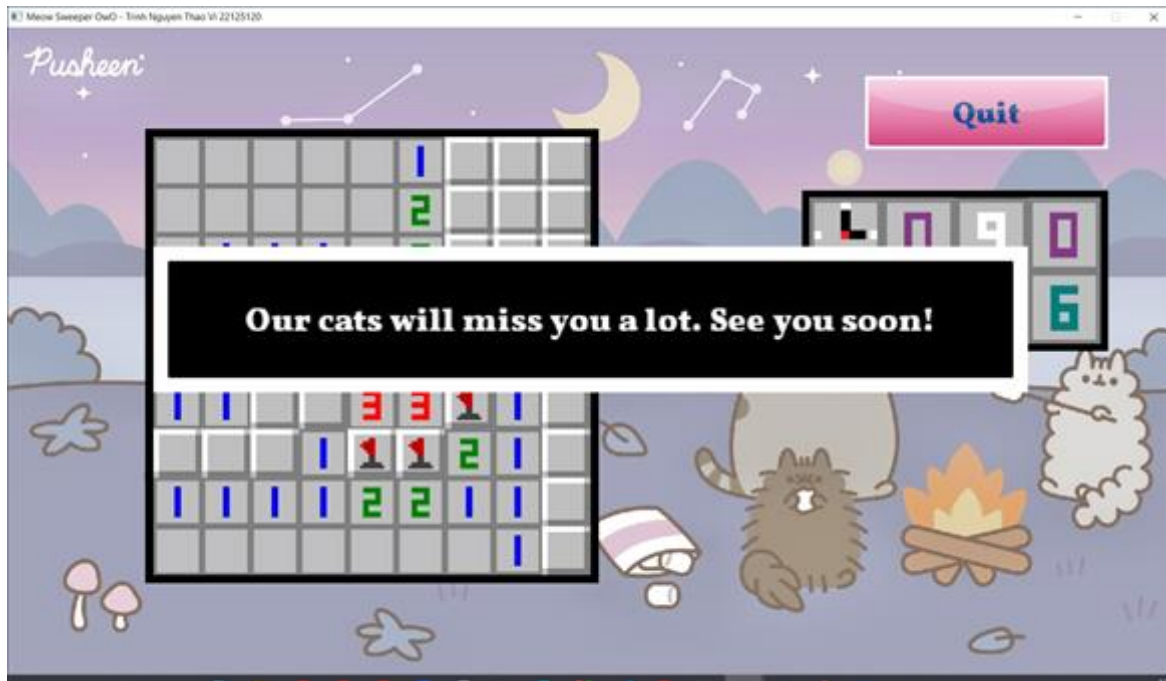


Figure 14. Notification when user want to quit

On the top-right corner, there is a **"Quit"** button. When user clicks on it, notification appears and current game will be saved (so that user can continue to play by choosing **"Last game"** in **"main menu"** next time)



Figure 15. The "main menu"

III. Statistics

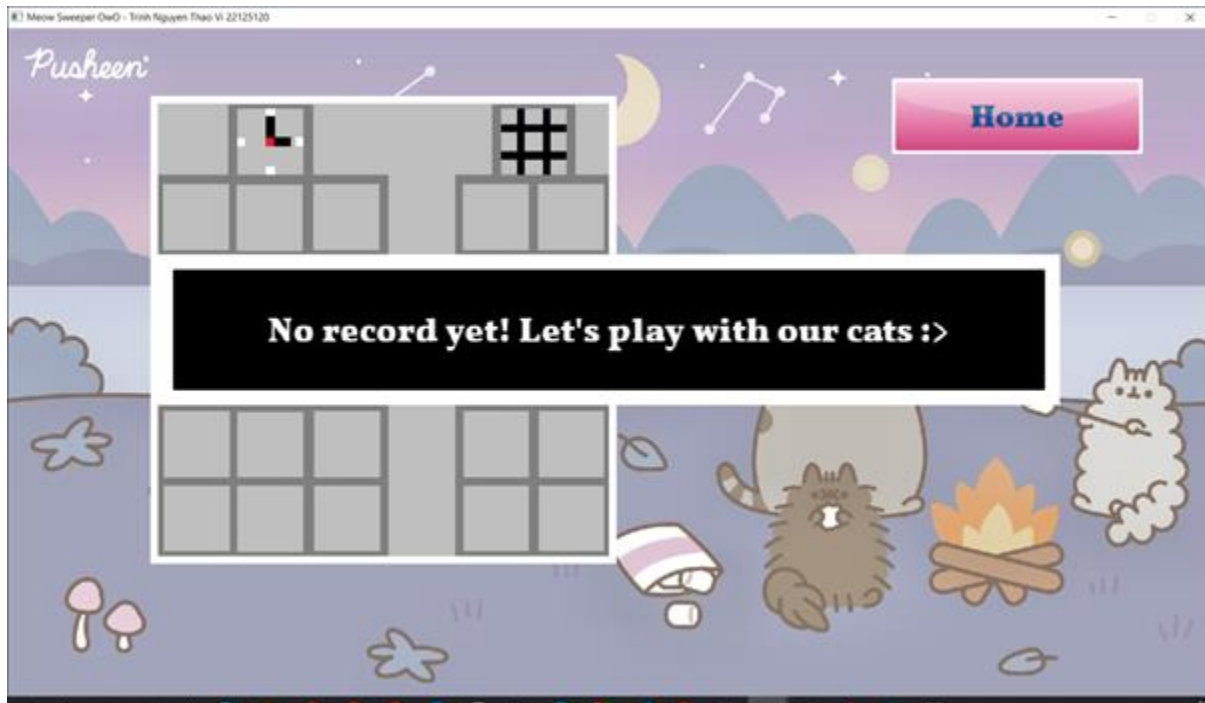


Figure 16. The "main menu" when user choose "Statistics"

After clicks on “Statistics” option, list of top 5 fastest time (also their sizes of boards) appears



Figure 17. List of top 5



If user did not win any games to have high scores, notification appears

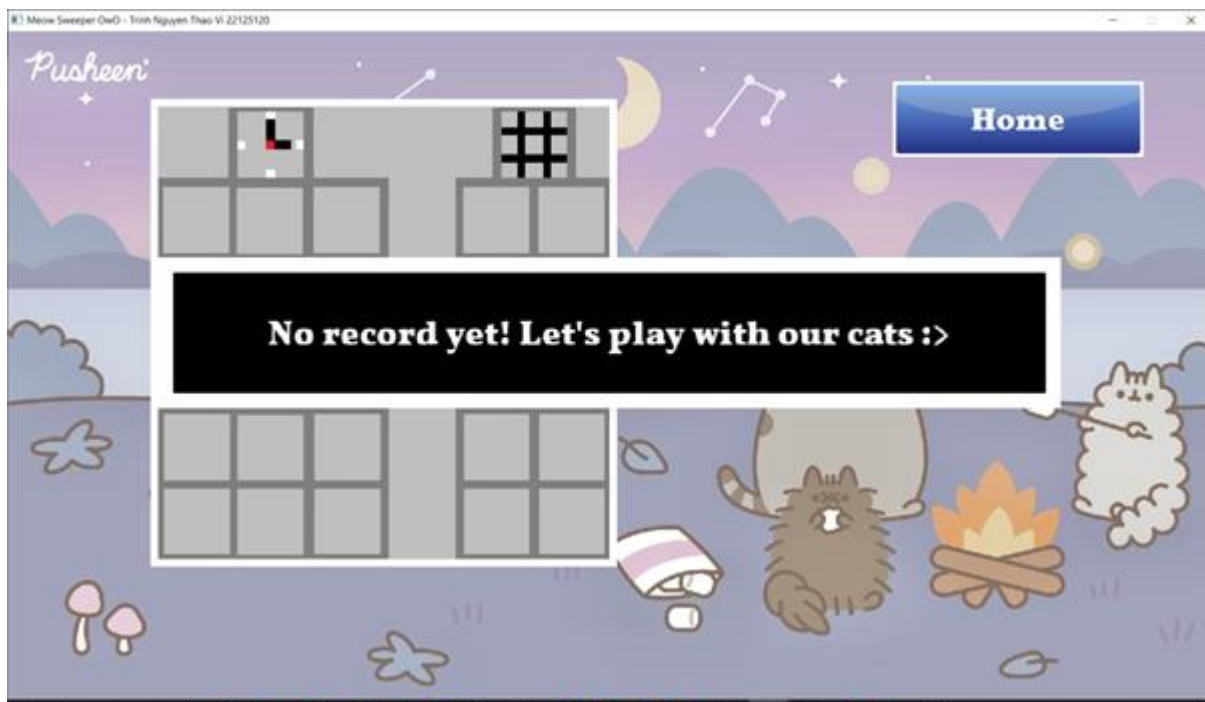


Figure 18. Screen when user choose "Home"

Clicks on "Home" to return the "main menu"

IV. About



Figure 19. The “main menu” when user choose “About”

When user clicks on **“About”**, the screen will show information of author (me)

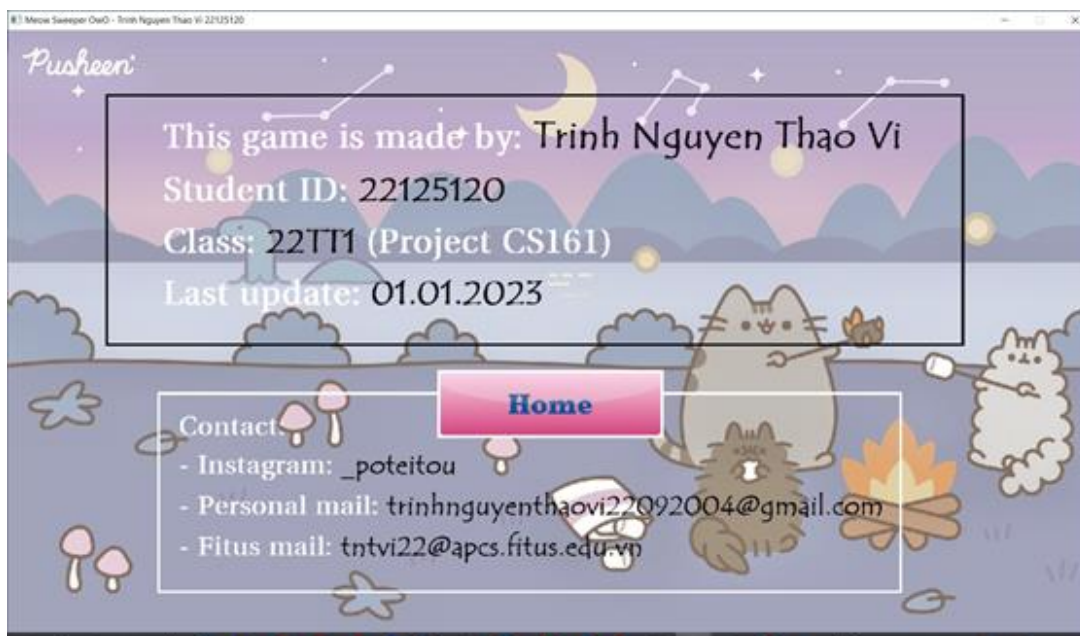


Figure 20. Information of author

Clicks **“Home”** to return to “main menu”

V. Exit



Figure 21. The "main menu" when user choose "Exit"

When user clicks on "Exit", notification appears and the program will be closed

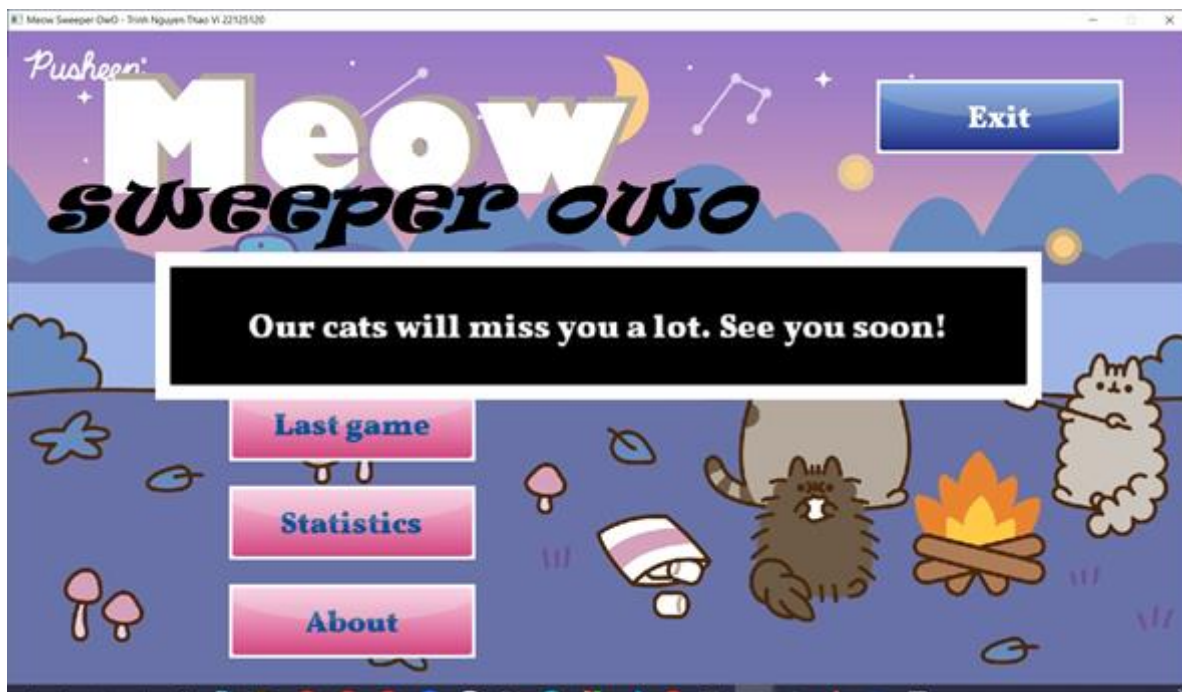


Figure 22. Notification

E. Demo video

Link demo video of the game on **Youtube**:

https://youtu.be/b2_BT_6AANU