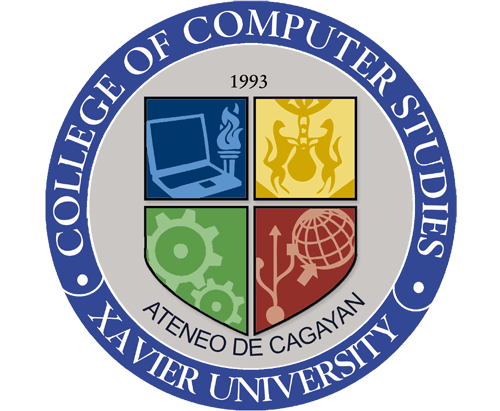
**Mazesweeper**

A project presented to the

College of Computer Studies

Xavier University





Aish, Mohamed Ahmed P.

Rosales, Joseph Edward A.

Zamora, Dave S.

March 2018

**Table of Contents**

**Abstract** **1**

**CCS Concepts** **1**

**Introduction** **1**

**Objectives** **1**

**Scope and Limitation** **1**

**Functionalities** **1**

**Pseudocode** **1**

**Data Structure and Algorithm** **2**

**Conclusion** **2**

**Appendices** **2**

**ABSTRACT**

We are required to create a program that uses data structures and algorithms in solving programming problems. In our case we created a game, named Mazeweeper, which covers 2 data structes. **CCS Concepts**

CCS → Software and its engineering → Software notations and tools → General programming languages → Language features → Classes and objects

1. **INTRODUCTION**

Mazesweeper is a text-based game. The game starts with a maze, you are in the bottom left and your goal is to reach the top right of the maze without encountering any mines that are hidden. There are 10 random mines. You are given three different choices and with each choice, there are many corresponding choices or results. You can also know how many mines are near you.

1. **OBJECTIVES**

* The objective of this program is to create an array of chars and generate 10 mines and populate it anywhere.
* Allow the user to decide which way he will take as his main goal is to reach the end without encountering any bombs.
* To be able to determine if the direction given by the player hits a mine or not. If it hits a mine then the game is over, if it’s not then continue. If the direction reaches the end. Then the user wins the game.

1. **SCOPE AND LIMITATION**

The direction inputted by the user must be only “w”, “a”, “s”, and “d”. The user can not choose the dimensions of the array or the minefield because it is standard 8 by 8 array. The user also cannot choose how many mines can be generated because it is set always to 10 mines.

1. **FUNCTIONALITIES**

**4.1 Buttons Or Direction Choices**

**4.1.1 “W” or Go upwards**

The direction will go up, from [x][y] to [x - 1][y].

**4.1.2 “A” or Go Sideward (Left)**

The direction will go left, from [x][y] to [x][y – 1] **.**

**4.1.3 “S” or Go Downward**

The direction will go down, from [x][y] to [x + 1][y].

**4.1.4 “D” or Go Sideward (Right)**

The direction will go right, from [x][y] to [x][y + 1].

1. **PSEUDOCODE**

5.1 Creating the Minefield or Array of chars

public void drawMinefield() {

initialize "y" to 0

intitialize "x" to 0

for(x = 0; x < mHeight; x ++){

System.out.print(x + " ")

}

for(y = 0; y < mWidth; y ++){

System.out.print(mMinefield[x][y] + " ");

prints the entire array that [x][y] generated through the for loop.

}

**5.2** Placing the Mines in the Minefield

public void placeMines()

initialize minesplaced to 0.

initialize minesplaced to 0.

create a new random named random.

while minesplaced is < or lesser than mMines. {

initialize x to random.nextInt(mWidth);

initialize y to random.nextInt(mHeight);

if mMinefield[x][y] is not equal to ' ',

initialize mMinefield[x][y] to ' ';

minesplaced++

}

**5.3** Checking for Winner

public void winner(int x, int y){

if the direction of user is equal to mMinefield[0][7],

call calculateHints Method

Call drawMinefield method

print out "You win!"

System.exit(0)

end game

}

**5.4** Checking for Gameover

public void gameover(int x, int y){

if the direction of user is equal to mMinefield[x][y] = ' ',

print " you stepped on a mine! game over!"

System.exit(0);

end game}

**6. DATA STRUCTURE AND ALGORITHM**

**6.1 Data Structure**

One of the Data Structure used in this program is a 2 dimensional Array. The Array is created for the maze or what we call the minefield. It is where the game takes place. The Array is an Array of characters and is an 8 by 8 array standard. The array only accepts char variables.

The other Data Structure used in this program is Stacks. We created a Stacks of Strings for the use of Last direction. After the direction is inputted, the direction is pushed in the stacks, after it is printed, it is popped out, and do it again repeatedly.

**6.2 Algorithm**

6.2.3 Moving of direction

The main focus of this program is the moving of direction. None of the methods would be used if the direction isn’t moving. By moving the direction depending on what the user inputted, we used a Switch statement. We got 4 cases namely w, a, s, d. The switch statement allows the variable direction (input of user) to be tested. If the input of user is either w, a, s or d, there are corresponding movements. If the input is “W”, we created a for loop that looks for the latest input from the user ( if it is the start the latest input is [7][0]) and initialize it from [x][y] to [x – 1][y]. To move the direction up. If the input is “A”, it will do the same procedure and initialize from [x][y] to [x][y – 1]. If the input is “S”, it will do the same and initialize from [x][y] to [x + 1][y]. If the input is “D”, it will do the same and initialize from [x][y + 1]. After the latest input is initialized, it will push the variable direction to the stack and after it is printed. It will pop out the variable direction.

**7. CONCLUSION**

We people always find our way to make a new thing that makes us fun. We were inspired by the game minesweeper. We used the logic of planting mines anywhere and when hit, you lose. We decided to create the game with not too fancy and not too hard. And we realized that it is not easy at all. We also realized that we need to use more data structures or algorithm in order to make our program look more decent and clean. But with our program alone, the game is fun to play.

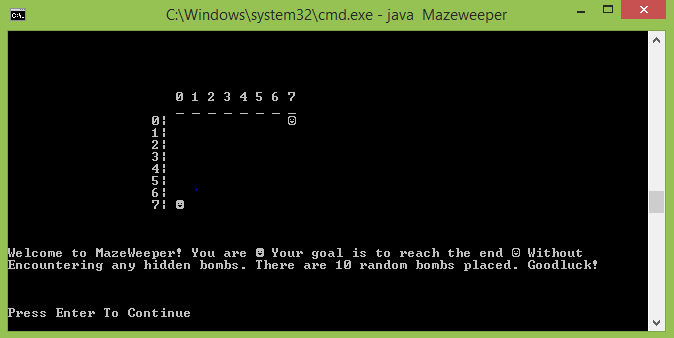
1. **Appendicies**
   1. **Project Proposal**
2. Members
3. Aish, Mohamed Ahmed P.
4. Rosales, Joseph Edward A.
5. Zamora, Dave S.
6. Title: MazeSweeper
7. Description
8. Mazesweeper is a text-based game. The game starts with a maze, you are in the bottom left and your goal is to reach the top right of the maze without encountering any mines that are hidden. You are given three different choices and with each choice, there are many corresponding choices or results. You can only input either ‘w’ ‘a’ ‘s’ or ‘d’.
9. Functionalities and Features

* User Input
  + You’re given the chance to pick which path you take.
  + WASD keys to navigate.

1. Possible Data Structure and Algorithm to be used

* Abstract Data Types
  + Stacks
  + Arrays
* Algorithm
  + Moving of Directions
  1. **Photo Documentation**

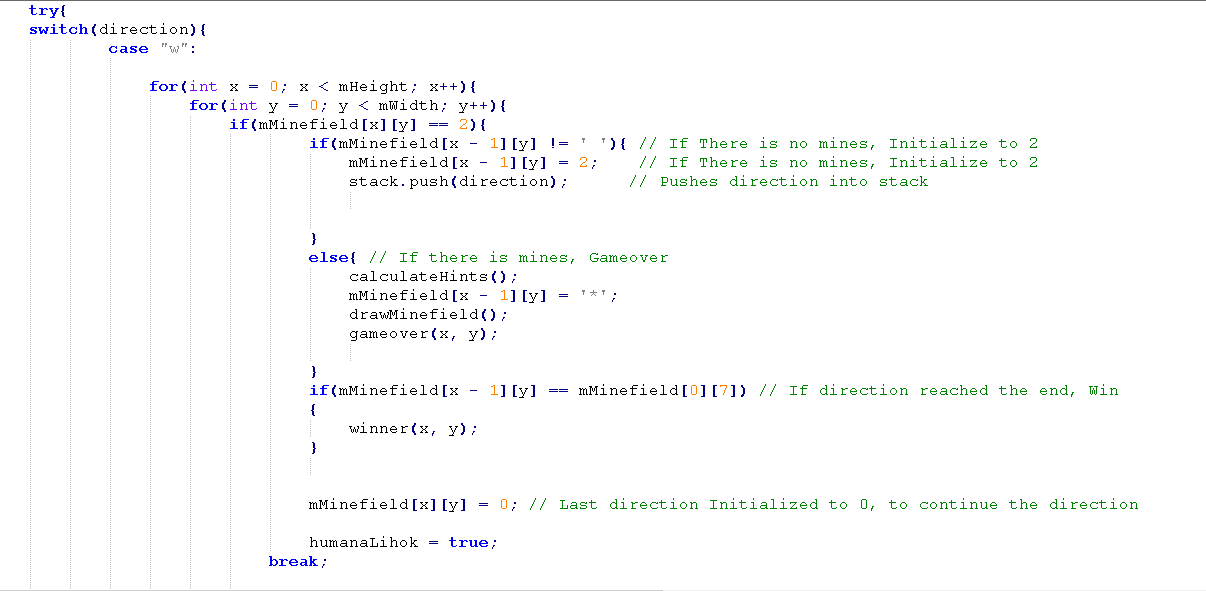
**Introduction**

****

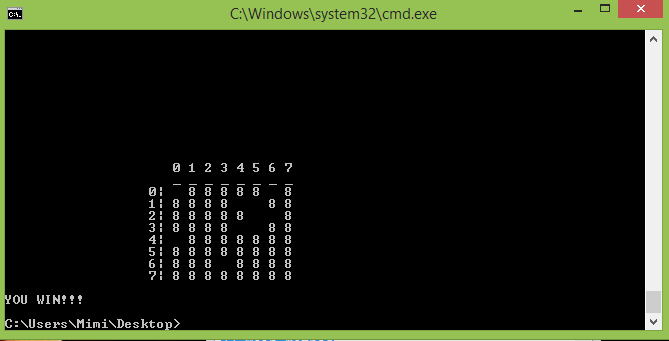
**Moving**

****

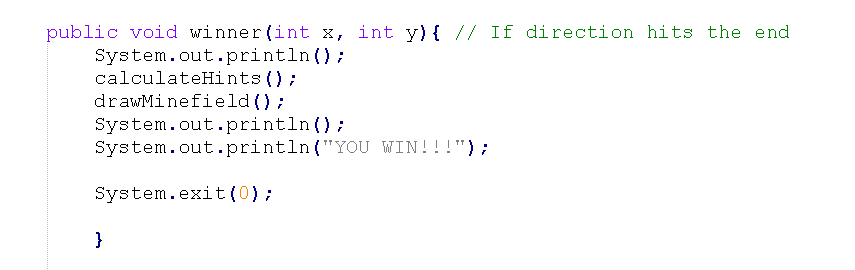
**Moving Codes**

****

**Winner**

****

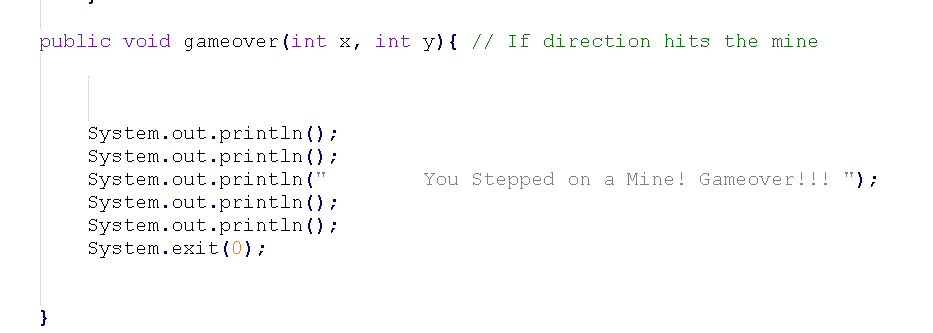
**Winner Codes**

****

**GameOver**

****

**GameOver**

****