



الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
يُونِيسْتِي إِسْلَامِيّ إِنْتَارَاغُشِيَا مَلِيسِيَا

KULLIAH OF INFORMATION & COMMUNICATION TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
SEMESTER I, 2015/2016

CSC 2104 DATA STRUCTURE AND ALGORITHM ANALYSIS

GROUP PROJECT

Apple Pen

Team members:

Amal Nadhira Binti Mohd Radzi	1514660
Nurhasyifha binti Mahazan	1517840
Muhammad Haziq Akmal Bin Ismail Shah	1516233
Muhammad Fakrudin Bin Zaidi	1516405

Table of Content

1. Feature of Games -----	2
2. Algorithm -----	3-5
3. Functions in Games -----	6

Features of Games

- **How to play?**

In this button, we provide an instruction to the user on how to play Minesweeper games.

- **Back Button**

This button will make user easy to undo or going back to homepage. The purpose of doing it because we want to provide user-friendly interface.

- **Game**

In this interface, we provide the Minesweeper game with 3 level, which are beginner, intermediate and expert.

- **Suggestion**

In this button, the user can provide their opinion regarding to our game to make an improvement to it.

Algorithm

Overall algorithm

1. Start
2. Initialize a queue
3. If current square is non-mine uncover it and add to queue, otherwise game over
4. Remove a square from queue
5. Count mines adjacent to it
6. If adjacent mine count is zero, add any adjacent covered squares to queue and uncover them
7. Go to step 3 if queue is not empty, otherwise finish
8. Stop

Welcome

1. Start
2. In start method:
 - (i). Initialize pane(padding size,text size,text colour,text font)
 - (ii). Initialize box(padding size,spacing,box alignment)
 - (iii). Initialize new button(HOW TO PLAY?,GAME,SUGGESTION)
3. If click how to play? button,instruction how to play the game will be given.
4. If click game button, the game will start.
5. If click suggestion button, the user can provide their opinion to improve our system.
6. Initialize new event handler(AboutUs,Feedback,Game)
7. Stop

Game

1. Start
2. Initialize new pane.
3. Initialize new button(Beginner,Medium,Expert,Home)
 - (i). If click beginner button, game will start in beginner level.
 - (ii). If click medium button, game will start in medium level.
 - (iii). If click exert button, game will start in expert level.
 - (iv). If click home, the game will return to home page.
4. Stop

Easy

1. Start
2. Initialize box and grid (TILE_SIZE = 400/60)

3. Start stopwatch/timer (`StopWatch st = new StopWatch();`)
4. if (`!tile.hasBomb`)= win
 - if (`tile.hasBomb`) = lose
 - If cell has `getNeighbors > 0`, show `getNeighbors` on the cell. If all cells which `hasBomb=false` are opened, end game with SUCCESS.
 - If `has bomb=true`, show message game over.
5. Get the time taken to complete (`txt.setText(String.valueOf(st.getTime()))`)
6. Stop

Medium

7. Start
8. Initialize box and grid (`TILE_SIZE = 400/40`)
9. Start stopwatch/timer (`StopWatch st = new StopWatch();`)
10. if (`!tile.hasBomb`)= win
 - if (`tile.hasBomb`) = lose
 - If cell has `getNeighbors > 0`, show `getNeighbors` on the cell. If all cells which `hasBomb=false` are opened, end game with SUCCESS.
 - If `has bomb=true`, show message game over.
11. Get the time taken to complete (`txt.setText(String.valueOf(st.getTime()))`)
12. Stop

Hard

1. Start
2. Initialize box and grid (`TILE_SIZE = 600/40`)
3. Start stopwatch/timer (`StopWatch st = new StopWatch();`)
4. if (`!tile.hasBomb`)= win
 - if (`tile.hasBomb`) = lose
 - If cell has `getNeighbors > 0`, show `getNeighbors` on the cell. If all cells which `hasBomb=false` are opened, end game with SUCCESS.
 - If `has bomb=true`, show message game over.
5. Get the time taken to complete (`txt.setText(String.valueOf(st.getTime()))`)
6. Stop

Feedback

1. Start
2. Initialize feedback text field
3. Submit suggestion (Button bsubmit = new Button("Submit Suggestion"));
4. Show message Thank You and team description
5. Stop

Major function

For this program, we decided to use object-oriented programming as this program quite a big project and thus, it is better if we can separate it function to its components rather than make it all on one file which will become hassle when we need to debug the program.

For this program, we use a few classes which each one of them have their specific function. Following is the list of the classes:

- About Us – for display information on how to play this game
- Welcome – for display the home page of the apps
- Game – for displaying the level of game
- Easy – to display game at easy level
- Medium – to display game at medium level
- Hard – to display game at Hard level
- Feedback – for player to give their suggestion about our system

