

## COMP 1451 – Assignment #1 (50 points)

Due: 11:59 p.m. the night before Session 5

Many computer games are based on the notion of a grid, i.e. a two-dimensional array. Some examples: Minesweeper, Battleship.



Minesweeper – a game from an old version of Windows

Your assignment is to implement a GuessingGame project. You are given an outline for the project consisting of four classes: InputReader, CoordinatePair, Grid, Game. Classes InputReader and CoordinatePair are complete and require no changes. Classes Grid and Game have the instance variables declared, and some suggested methods as “stubs”, i.e. each method has a comment explaining what it will do, and the method signature. Your task is to complete these methods and implement the game. Feel free to add other classes and/or methods as you see fit. Note that all classes require regular “get” and “set” methods.

Here is how the GuessingGame works:

- A two-dimensional array of String (the working grid) is constructed. Use constants for the dimensions (number of rows and columns). By default Java will set the contents of the array to the default for the data type, in this case null.
- A “target” symbol (you decide which symbol) is placed in an arbitrary number of randomly-chosen grid locations, e.g. “T” is placed in five different grid locations, randomly generated. Be sure to fill five different locations if five is your chosen number of targets. (Check for duplicates.) Use constants for the target symbol and the number of locations to fill.

- A second grid is constructed of the same dimensions as the original. This is the display grid, used to give feedback to the user. In the beginning it is filled with a symbol (your choice) that indicates all locations are hidden.
- The user is prompted a number of times to type grid coordinates. (Use a constant to specify how many tries the user gets.) For each pair of coordinates the user types, the display grid is updated (and re-displayed) with one symbol for a “hit” and another for a “miss”. A “hit” occurs when the user’s choice of coordinates matches the placement of a target in the working grid. The “hit” and “miss” symbols replace the default “hidden” symbols in the appropriate locations in the display grid. If the user types invalid coordinates an error message is displayed and the user is prompted to try again.
- Give your user clear instructions – this is important any time you write software for people to use and will be expected throughout the course even if it is not explicitly mentioned in the requirements.
- When the user’s turn has ended a message is displayed telling how many targets were hit, and the working grid is displayed.
- When displaying a grid, put a space between symbols. See the example below.
- Turn on “unlimited buffering” in the BlueJ terminal window so you can record a transcript of your game play. Copy and paste this output into a Word document that you include with your assignment.

Think before you code! What are the attributes and responsibilities of each class? How do the classes relate to each other?

Marks will be given for:

- Style – see the style guide Appendix J in your textbook.
- Correctness and completeness – code meets the requirements listed above.

Create a .zip file containing your entire BlueJ project (zip the folder, not the individual files). Name the .zip file with your name and the assignment number, e.g. “SusanWongAssign1.zip”. Upload the file to the D2L dropbox before the cutoff time.

Below is an example transcript of game play.

```
Welcome to the Guessing Game!
You will be shown a grid with 6 hidden targets.
You will be given 10 chances to 'hit' a target.
When prompted, type the coordinates to 'hit'.
The grid will display * for a successful hit and - for a miss.
```

```
$ $ $ $ $
$ $ $ $ $
$ $ $ $ $
$ $ $ $ $
$ $ $ $ $
```

Try number 1

Coordinates start from zero.

Type your choice of coordinates (row column) with a space between >0 0

```
- $ $ $ $
$ $ $ $ $
$ $ $ $ $
$ $ $ $ $
$ $ $ $ $
```

Try number 2

Coordinates start from zero.

Type your choice of coordinates (row column) with a space between >5 0

Not a valid coordinate, please re-enter

Try number 2

Coordinates start from zero.

Type your choice of coordinates (row column) with a space between >4 0

```
- $ $ $ $
$ $ $ $ $
$ $ $ $ $
$ $ $ $ $
- $ $ $ $
```

Try number 3

Coordinates start from zero.

Type your choice of coordinates (row column) with a space between >4 4

```
- $ $ $ $
$ $ $ $ $
$ $ $ $ $
$ $ $ $ $
- $ $ $ *
```

Try number 4

Coordinates start from zero.

Type your choice of coordinates (row column) with a space between >0 4

```
- $ $ $ -
$ $ $ $ $
$ $ $ $ $
```

```
$ $ $ $ $  
- $ $ $ *
```

Try number 5

Coordinates start from zero.

Type your choice of coordinates (row column) with a space  
between >1 1

```
- $ $ $ -  
$ - $ $ $  
$ $ $ $ $  
$ $ $ $ $  
- $ $ $ *
```

Try number 6

Coordinates start from zero.

Type your choice of coordinates (row column) with a space  
between >2 2

```
- $ $ $ -  
$ - $ $ $  
$ $ - $ $  
$ $ $ $ $  
- $ $ $ *
```

Try number 7

Coordinates start from zero.

Type your choice of coordinates (row column) with a space  
between >3 3

```
- $ $ $ -  
$ - $ $ $  
$ $ - $ $  
$ $ $ - $  
- $ $ $ *
```

Try number 8

Coordinates start from zero.

Type your choice of coordinates (row column) with a space  
between >1 0

```
- $ $ $ -  
* - $ $ $  
$ $ - $ $  
$ $ $ - $  
- $ $ $ *
```

Try number 9

Coordinates start from zero.

Type your choice of coordinates (row column) with a space  
between >2 0

```
- $ $ $ -
```

```

* - $ $ $
- $ - $ $
$ $ $ - $
- $ $ $ *
```

Try number 10

Coordinates start from zero.

Type your choice of coordinates (row column) with a space between >4 1

```

- $ $ $ -
* - $ $ $
- $ - $ $
$ $ $ - $
- * $ $ *
```

Sorry, you only hit 3 targets out of 6

```

$ $ $ $ $
T $ $ $ $
$ T $ $ $
$ $ T $ T
$ T $ $ T
```