

School of Computer Science and Information Technology. RMIT
COSC2531 Programming Fundamentals
Assignment 2

Due date: 9am 10th Oct 2016
30% of course assessment

Wumpus!

Wumpus is a text-based game, where a player moves through a cave system searching for gold and trying to avoid bottomless pits and a feared Wumpus. The cave system is a 4 x 4 grid. Each grid element can hold one *GameItem*. A *GameItem* is one of: Gold, Pit, Wumpus or ClearGround.

The player can move forward, backward, left or right one grid element at a time. The cave system wraps around so, for example, moving left from position [3][0] causes the player to appear in position [3][3].

The player can sense, but not see, what is in the immediate grid elements around the player. A pit will be sensed as a breeze, gold as a faint glitter and the Wumpus as a vile smell. In this assignment, the grid will be displayed, but in a real game this feature would be disabled.

If the player moves onto a pit or the Wumpus, the player dies and the game ends. If the player moves onto gold, the game score increases by one and the gold is replaced by ClearGround.

Implementation:

You should write a *Game* class, which has a 2D array of size 4 x 4 of *GameItem* called *board* for implementing the Wumpus game described above.

Classes *Gold*, *Pit*, *Wumpus* and *ClearGround* are extensions of class *GameItem*. *GameItem* should provide a public method, *display()*, for use when displaying the board and a private instance variable for specifying the displayed character. *Gold* is displayed as 'g', a *Pit* as 'p', the *Wumpus* as 'W', clear ground as '.'. *GameItem* provides a constructor *GameItem(char c)* for specifying the displayed character.

It should not be possible to instantiate an object of *GameItem*. The player is not implemented as an object, instead just being represented by *Game* as private row & column coordinates. The position of the player is displayed by a '*'.

The items should be positioned randomly over board. There is exactly one Wumpus, at least one, and up to three pieces of gold (chosen randomly) and exactly three pits. The player is positioned over a ClearGround position.

Game should provide:

- a private method *setBoard()*, which instantiates objects on the board
- a private method *display()* which will display the *board*.
- a private method *senseNearby()*, which displays text about what the player can sense from the board elements immediately surrounding the player.
- a private method *menu()* which will provide a menu asking the user to make a choice from the following and obtain the user input:
=====Wumpus=====
 1. Move player left
 2. Move player right

3. Move player up
4. Move player down 5
- Quit

- a public method *runGame()* that will display the board, print out what the player can sense, present the menu and process the user's decision according to the game play described above.

The *World* class only instantiates an object *myGame* of class *Game*, and calls *myGame.runGame()*

You need to show the Pit, Wumpus and Gold when the game is being played and also show the *SenseNearby()* field as this is testing your ability to utilize polymorphism to call method of different classes.

GameItem *display()* method is used to display each game item.

Game *display()* method is used to loop through *GameItem* array to display the board.

You may find the class *Random*, available in *java.util.** useful.

For example the following will assign integer variable *guess* a value from 0..4 inclusive.

```
Random randomGenerator = new Random();
guess = randomGenerator.nextInt(5);
```

.

Submission:

- Due date: 9am 10th Oct 2016. Representing 30% of course assessment
- This is a group assignment of maximum 3 students (prefer not solo)
- You will need to submit 7 electronic files, (*Game.java*, *GameItem.java*, *Gold.java*, *Pit.java*, *Wumpus.java*, *ClearGround.java* and *World*. Please note these must be Java source files (e.g. *World.java*), not class files (i.e not *World.class*). We will not mark a class file.
- This assignment needs to be well designed. Please draw the Class diagram of your final coding design Please screenshot a full set of successful executions. Save the class diagram and screenshots of executions in a PDF file called **Design.pdf**.
- Weblearn is used for submission. Please zip all source files and *Design.pdf* into one for submission (7z, rar, zipx or other compression format will be penalized).
- Remember to include all team members' names, student IDs and contribution % of each person in your submission of *Design.pdf* in the first page. One submission one group and multiple submission will be penalized or at your own risk of getting wrong marks.

Standard warning:

This assignment must be your own work. Any submission found in whole or in part to be plagiarized will be given zero marks and will not count towards your assessment. If an award of zero marks means that you do not pass the assignment hurdle then you will fail the course.

Frequent Asked Questions

Question: I was just wondering if the Wumpus will be moving around the board as well? Or will it be stationary like the position of the pits?

Answer: The Wumpus position is randomly generated. In each session, it will not be move around. It will be stationary like the position of the pits.

Question: Is there only 1 pit, 1 gold and 1 Wumpus in the game?

Answer: There is exactly one Wumpus, at least one, and up to three pieces of gold (chosen randomly) and exactly three pits. The player is positioned over a ClearGround position.

Question: Hey I'm hoping to submit and wanted to know if the tutor name is important, I'm not sure who my tutor is.

Answer: The tutor's name is not important. You can just submit to the default one. Yes, all files need to be zipped into one file. The zip file name is not important as weblearn will auto rename it.

Question: Is it like after generate Wumpus, pit and gold, the position of these objects are settled, no matter where the player moves on the board, these objects are not moving? Or, each time player moves a step, the position of W,P and G changes randomly? If player is surrounded by more than one objects, senseNearby() should output all information of them

Answer: After randomly generate W, P, G in the beginning of the game, these objects are settled in position. Only replaced by "." if player walks into it, like gold. Yes, senseNearby() should output all information of up, down, left and right.

Question: At starting do I need to display 4X4 Grid along with GameItems like (1 Wumpus, 3 Pit and 3 Gold).I meant player basically see all gameItems position in the grid (not like real game) at the beginning of the game.

Answer: Yes, you do need to display 4x4 grid along with GameItems at the starting point.

Question: After each user selection from Menu do I need to display grid along with gameItems position again.

Answer: Yes that is why you call display() method in the Game class.

Marking Guide:

OO Components 12 marks

Functional correct 12 marks

Documentation, Coding Style, Submission requirement, Error checking 8 marks