

CS12420

Drown the Scurvy Dog

2013-2014

Individual project – CS12420

**This is due on Monday, 5th May between 9am and 5pm
and is worth 20% of your final mark for the module.**

This assignment asks you to code a variation of a hangman game.

The purpose of this assignment is to give you practice with programming something algorithmic in Java to a strict interface, JUnit testing, Swing and MVC.

Hand In

In a zipped folder (it must have the .zip extension) you should upload to Blackboard by the specified time a PDF document file (max 12-15 A4 pages, 12pt) that includes:

- A cover sheet with your name, and **very clear** instructions on how I could run your program.
- A problem analysis section that must include a use case diagram.
- A design section (including class diagrams and one sequence diagram) and rationale for the design.
- A section showing main algorithms (not code! Pseudocode is best). Have a separate algorithm for each use case.
- A section on your testing strategy and testing results (including screen shots of the GUI and JUnit tests running). Use the template system test table that is in the course materials (under the Testing section on Blackboard). JUnit need only be used to test the model. That said, each JUnit test must be related to one of the use cases. Make this connection clear.
- Evaluation section: Say what you found hard or easy, what you learned, what was omitted and why. State what mark should you be awarded and why? This will help me formulate feedback.

The zipped folder must also contain your source code. This can be an exported Eclipse project but you may use other open source IDEs or none, if you wish.

Assessment

Marking will be in line with the usual assessment criteria for project work

(<http://www.aber.ac.uk/~dcswww/Dept/Teaching/Handbook/AppendixAA.pdf>).

- | | |
|--|-----|
| • Analysis and Design as expressed in the document | 10% |
| • General documentation quality | 5% |
| • Quality of code including error checking, indentation and Javadoc style commenting | 10% |
| • Degree and correctness of your implementation with respect to the requirements | 30% |
| • Testing | 20% |
| • Evaluation | 10% |
| • Flair marks (something extra that will impress us) | 15% |

This is worth 20% of your final mark for the module.

Plagiarism. Cases of plagiarism in student assignments are not taken lightly by the department, and the consequences of plagiarism by students can be severe. Please ensure that you understand what constitutes plagiarism and read the plagiarism notice from the course handbook or on the web.

Note: this is an “individual” assignment and must be completed as a one-person effort by the student submitting the work. This assignment is **not** marked anonymously.

By submitting your work to Blackboard, you agree to the statement about the Declaration of Originality. This statement will be visible when you submit your report. It is the same as that shown on the Anonymous Marking Sheet Cover. You **do not** need to submit a separate declaration form.

If you are late then please complete a Late Assignment Submission form and hand this in to the Department office. See <http://www.aber.ac.uk/~dcswww/intranet/staff-students-internal/teaching/resources.php>.

The Problem

This project is a variation of hangman. It involves programming both a text based, and a graphics based version of the same game. The graphics based version should be pirate themed. The words to be guessed must be pirate themed and must be read from the provided file called *piratewords.txt* that has the format (you can add to the phrases: see <http://www.yarr.org.uk/talk/>):

```
3
ahoy there
me hearties
well, shiver me timbers
```

We should be able to play either the graphics version OR the text version.

Graphics Output might look something like this:



As you get letters wrong the pirate walks the plank.

Text Output might look like:

so far you have guessed letters:
you have 10 guesses left
and you can see *****

type L for letter or W for word: l
Guess a letter: e

so far you have guessed letters: E
you have 10 guesses left
and you can see **E** *****

type L for letter or W for word:

**You have been told that you must program your system to interfaces.
You need to use JUnit to check your implementation of this interface.
In addition to your testing, you should write a main program application that runs from the command line by typing: `java PirateApp`
Alternatively, if you are using a JAR file make sure you specify the command-line command we should use.**

The interface you are coding the Model of the game to is this:

```
/**
 * This interface describes the behaviour of a hangman type of game
 * To be used in CS124 individual assignment
 */
public interface GameModelInterface
{
    /**
     * this method returns what the user is allowed to see
     */
    public String getVisible();

    /**
     * this method returns the words to guess
     */
    public String getHidden() ;

    /**
     * this method tells how many guesses are left
     */
    public int guessLeft() ;

    /**
     * this method returns the letters that have already been guessed
     */
    public String getLetters() ;

    /**
     * this method sees if the letter is in the words
     * if it is it updates what the user can see (visible)
     * if not it removes a guess
     *
     * @param letter the letter to try
     * @return whether there is winner
     */
    public boolean tryThis(char letter);

    /**
     * this method sees if the word guess is correct
     * if it is it updates what the user can see (visible)
     * if not it removes 5 guesses
     *
     * @param guess the word to guess
     * @return whether there is winner
     */
    public boolean tryWord(String guess) ;
}
```