**Software Development 3 Coursework** (SET09101)

**“Sky Wars”**

This coursework is out of this world!



You should write a Java program to meet the specification shown below.

This assignment constitutes 100% of the module assessment.

**Problem Statement**

Using design patterns, threads and a GUI, you are to write the Java code for a game involving space ships. You can design your own game or use the sample game given below.

If you design your own game, your game should have moves, a GUI and the ability to undo moves. Read through the sample game to get an idea about the type of game. Double check with the module leader that the game you are developing is OK.

**Sky Wars (Sample Game)**

Sky Wars consists of a Sky, a Master Space Ship and a number of Enemy Space Ships

**The Sky**

The sky can be thought of as a four by four grid.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

When the game starts the Master Space Ship is placed in a random square except for the top left corner (i.e. part of the Sky).

**Master Space Ship**

**Operational Mode**

The Master Space Ship has two operational modes: passive and aggressive. Passive is the default mode but you must be able to change the mode dynamically i.e. while the program is running.

**Moving**

When the game starts the master space ship is randomly allocated to a square in the sky, with the exception of the top left square. When the Master Space Ship moves, it can move to any of the neighbouring squares as shown below. The space ship cannot move out of the sky.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Master Ship |  |  |
|  |  |  |  |
|  |  |  |  |

**The Master Space Ship can move to any of its neighbouring squares**

The square the Master Space Ship moves to is chosen at random from all the possible squares that it can move to.

**Enemy Space Ships**

**Moving**

There are three types of enemy space ship that can enter into the sky; a BattleStar, a BattleCruiser and a BattleShooter. The top left corner of the sky is a one-directional intergalactic hole and it is through this hole that all enemy ships enter the sky (because the intergalactic hole is one directional nobody leaves the sky through the top left hand corner). Once in the sky the enemy ships move in exactly the same way as the Master Space Ship; i.e. they move randomly to any of the neighbouring squares.

Every time the Master Space Ship moves there is a one in three chance of an enemy space ship entering the sky, the type of enemy ship is completely random.

|  |  |  |  |
| --- | --- | --- | --- |
| Battle Cruiser |  |  |  |
|  |  |  |  |
|  | Master Ship |  |  |
|  |  |  |  |

**Enemy ships always enter the sky through the one-directional intergalactic hole in the top left corner of the sky**

Although there are only three types of enemy ships, new types of enemy ship may be available in the future. You should take this into account when coding your solution.

**Conflict**

If the Master Space Ship moves into the same square as an enemy ship, the Master Space Ship destroys the enemy ship and the enemy ship is removed from the sky.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Battle  Star |  |  |
|  |  |  |  |
|  |  | Master  Ship |  |
|  |  |  |  |

**The Master Space Ship destroys a lone enemy ship**

If the Master Space Ship moves into the same square as two or more enemy ships, what happens depends on the Master Ship’s operational mode. If the mode is passive, the enemy ships destroy the Master Space Ship and the game is over.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Battle  Cruiser |  |  |
|  |  |  |  |
| Battle  Star |  | Master  Ship |  |
|  |  |  |  |

**Two or more enemy ships destroy the “passive” Master Ship and the game is over**

If the Master Ship’s operational mode is aggressive it takes three or more enemy ships in the same square to destroy it. Remember that you should be able to change the operational mode of the master ship from the GUI.

**GUI**

Your game should be controlled from a GUI. From the GUI you should be able to start the game, change the operational mode of the Master Space Ship and make a move and undo moves. Your GUI should also have a visual representation of the game which should be updated as the game status changes.

**Report**

You should write a report explaining where in your code you have used threads and design patterns, which patterns you have used and why. You should highlight the problems you would have encountered if you had not used patterns.

Your report should also discuss the main features of your GUI, and include screen shots.

If you have used any programming techniques not taught on the module you should discuss what techniques have been used, where they were used and why.

**Marks**

Marks will be allocated as follows

Use of Patterns 50%

* Factory (10%)
* Observer (10%)
* Strategy (10%)
* Command (20%)

Use of Threads 10%

GUI 20%

Programming Technique not taught 10%

Report 10%

**Collaboration and Plagiarism**

This is an individual piece of assessment and the work submitted should be entirely your own. You are not allowed to collaborate with other students or to copy the work of other students.

In the event of any doubt about authorship you will be interviewed by the School of Computing Academic Conduct Officer and may be asked questions about any aspect of the work.

**Submission**

You should demonstrate your program working **on Monday 1st December 2014** (details of the demo order, times etc will be sent out by e-mail one week before the demos)**.**

After the demonstration you should submit your assignment via Moodle. The easiest way to do this is to put everything into your src directory of the project and compress the directory; this means you only need to submit one file. You should also include a JAR file of your program.

You should include your UML class diagram as a jpg file. You can do this very easily in Eclipse; Right click and select Export... UML Diagram... Select the diagram to export...Select jpg when given a selection of formats.

Your submission is entirely electronic via Moodle, you should submit your

* Java project exported to an archive file
* JAR File
* UML Diagram in jpg format
* report

by **23.59 on Monday 1st December 2014**