**King Fahd University of Petroleum & Minerals**

**College of Computer Sciences and Engineering**Information and Computer Science Department
**SWE 436: Object‐Oriented Design Patterns (Term 142)**

Assignment # 1 Weight 7%

Handed : Thursday Feb 12, 2015
Due : Thursday Feb 26, 2015 10:00 pm (Solid Deadline – no extensions)

# Objectives

* Learn how to program in a new Java environment: Greenfoot
* Practice with the factory design pattern
* Learn how to create you game characters using image editing software such as Paint.net or Gimp
* Learn how to manage your time by measuring the time spent doing this assignment
* Learn how to write proper reports

# Instructions

1. Calculate the time you spend in this assignment using the table below

– any activity should be considered including

1. Reading the requirement of the homework
2. The design of your implementation, including drawing UML diagrams
3. Coding
4. Testing
5. Other activities where you spent some time on (e.g., reading a book or a website).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Start** | **End** | **Duration** | **Comments** |
| Task 1 |  |  |  |  |
| Task 2 |  |  |  |  |
|  : |  |  |  |  |
| Task n |  |  |  |  |
| Total |  |  |

1. Answers have to be typed; **handwritten solutions will not be accepted**.
2. Submission:
	1. through BlackBoard
	2. **Softcopy** (ZIP or RAR file) including:
		1. your report in PDF  format (**WORD format is NOT acceptable**).
		(CutePDF is a simple and free utility to convert documents to PDF format)
		(NOTE: it should be a self-contained report – i.e, when I read the report I should not need to open any other file)
		2. Your source code: Java files
		3. Your compiled classes (.class files)
		4. configuration files and other related resources (if any)
3. The report should include a **cover page** showing: course name, assignment number, date of submission, your name and ID
4. Include the question text and then put your answer
5. The file name should be in the following format:
 HW<#> - <YOUR ID> - <YOUR NAME>
6. **Diagrams**, **program text**, and **output** should all be included in the report
7. You should use Java as your programming language
8. Your solution should be in a SINGLE document.
9. Code should be formatted properly
	1. Copy the code in Notepad++
	2. Code should be in **“Courier New”** font
	3. To format the code choose Menu 🡪 Language 🡪Java
	4. Click Plugins 🡪 NppExport 🡪Copy RTF to clipboard
	5. Paste it in Word

Try to use some callouts (like this one) to illustrate the code

* 1. NOTE: You don’t have to copy ALL your code.
	Just copy the parts that need to be illustrated.
1. Correct solutions earn full mark. However, ***not following the previous points will reduce your mark.***
2. Include the following table in your cover page

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Grade** | **Your Grade** | **Comments** |
| Task # 1 | 5 |  |  |
| Task # 2 | 40 |  |  |
| Task # 3 | 20 |  |  |
| Task # 4 | 5 |  |  |
| Check list and penalties No Cover page with grade table -10 🞎 File name (both zip file and report) -5 🞎 Code is missing in the report -20 🞎 Code format and color -10 🞎 Time log -5 🞎 report -10 🞎 PDF format -10 🞎 Source Code/ Class files not attached -40 🞎 |
| Total | 70 |  |  |

# Introduction

In this assignment, you are required to create a simple 2D game using Java.

Download and install Greenfoot from <http://www.greenfoot.org>

A good starting point would be this youtube clip: <http://www.youtube.com/watch?v=NcGe141R2yA>

There are many examples that you can find in the Greenfoot installation folder:

C:\..................\Greenfoot\scenarios

# Your first game: Air Combat

The basic idea of this game is to create a simple game of an air jet that fights enemies (other airplanes, dragons, etc. ) on the sky. The requirements are as simple as the following:

1. When the game starts, you should have one air jet on the screen at the middle
2. The air jet keeps moving and can be controlled vertical and horizontal using the following keys:
	1. Up : Move up (vertical)
	2. Right : Go forward (horizontal)
	3. Left : Go backward (horizontal)
	4. Down : Move down (vertical)
	5. Space : Shoot only one missile (from the center of the air jet).
3. If the missiles (goes only horizontal) from the air jet hits the dragon, the dragon should disappear from the screen.
4. The dragon keeps coming and they should fire horizontally.
5. If the fire ball from the dragon hits the air jet, then air jet should be crashed and exit the game.
6. The game should be shown from the top view and the direction of the game should be from the left (location of air jet) to the right (where the dragons keeps coming).
7. Use your imagination and/or the web to get images for your characters. Use Paint.net or Gimp for any modification if necessary.

# Task # 1: Class Diagram

Draw a class diagram showing the structure of your game **(With NO Patterns).**

# Task # 2: Implement the game

Implement the game using Greenfoot. You have to implement all the requirements as specified in the game description Count and display the number of killed dragons (Bonus)

1. Calculate the number lines of code required to implement the game. (you might use some tools here)
2. Calculate the number of classes, modules, and number of links between modules in your game.

# Task# 3: The Factory Pattern

Refer to the game that you developed in Task # 2, there is one more requirement: let the user select the difficulty of the game (Easy, Medium, Difficult) and based on his choice you can control the number of the dragons and the speed of the game.

1. Modify the game to add the difficulty feature using Factory design pattern.
2. Draw the class diagram showing the structure of your game **(With Factory Pattern).**
3. Calculate the number lines of code (LOC) required to implement the game.
4. Calculate the number of classes, number of modules, and number of links between modules in your game.

# Task # 4: Comparison

1. Compare the statistics in step 3 and 4 with the same statistics in question 2 using tabular format.
2. Write your comments on the results.

<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< END >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>