Term Project, CMSC 3101 Object Oriented Programming, Fall 2012

Objective: To practice object oriented design paradigm by developing a computer game using Java GUI and 2D graphics API's.

Scope of the project:

- Begin with the basic structure of the *MissileCommand* demo program that consists of three components for animation:
  1. Shared Data Structure (GameData class): All action figures in the game
  2. Renderer (GamePanel class): Draw all game action figures on the screen
  3. Animator (Animator class): Updates periodically action figures position, size, color, etc.
- Enhance the program to become a fully functioning game. For example, games similar to missile command, space invader arcade games, etc.
- Students should define clearly the scope (goal) of the project in the proposal as explained below.

Demo and Turn-in Environment: You are encouraged to continue to use the NetBeans IDE for developing programs, but you must transfer the project to your account of the **cs.uco.edu** Linux server for demo and turn-in purposes. Note that the server runs **JDK 6**. For effortless transfer of your programs to the Linux server:

- In NetBeans, create classes in "default" package in a new project. (Choose JDK6 from property)
- Use WinSCP (FillZilla or Fugu for Mac) to transfer files between your computer and the server.
- Turn on "Enable X11 forwarding (Connection → SSH → X11)" in PuTTY configuration. (For Mac, use "`ssh -X hsxxx@cs.uco.edu`" from X11 terminal)
  - For Windows, you need to install "Xming" so that PuTTY can display graphics.
  - Youtube video for Xming installation: http://www.youtube.com/watch?v=D-xV5HRBtQI&feature=related

Proposal: **DUE Nov 8 (Thursday) 7:30pm**

- Create a pdf file with the following sections/contents: **ONLY Sections I and II are required by Nov 8**. The rest of Sections are due Nov 13 Tuesday.
  - **I. Game Description** (minimum half page in single space): Describe the game that you are going to create using complete English sentences.
  - **II. The List of Game Features:** List features you want to add so that it becomes a fully functioning game.
  - **III. UML diagram**: Draw a UML diagram which clearly shows the relationships among classes you have identified. In the diagram, list methods only if they are relevant to the discussion of polymorphism in the next section.
  - **IV. Polymorphism**: Discuss in detail how polymorphism might be utilized in your design.
  - **V. Design Patterns:** Discuss in detail what design patterns you are going to use and how they can be implemented in your program. The use of minimal two design patterns is required.

Class Schedule for the rest of the semester

- Nov 8 (Thur): Initial proposal due (Sections I and II only)
    - Individual meeting with the instructor.
        - **Bring in a printed copy** of the proposal (Section I and II only)
        - 2:00-4:00pm OR 7:30-8:15pm (At the instructor's office)
    - Students will get an approval of the project scope at the meeting.

- Nov 13 (Tue): Individual meeting
    - **A completed proposal due** (ALL Sections of I through V).
        - Submit via D2L ("proposal" link on Dropbox) **before** you meet with the instructor (D2L submission due: 7:30pm)
    - Individual meeting with the instructor.
        - Create a sub-folder on your account of the cs server named "**progress1**" and put in the folder all source programs (.java files and image files if any) of your project.
        - Give the demo of your progress from the cs server.
            - 2:00-4:00pm OR 7:30-8:15pm (At the instructor's office)
            - Students may choose to show the missile command demo from their account if any running part of the program is not ready yet.

- Nov 15 (Thur): No class meeting; work on the term project.

- Nov 20 (Tue): Individual Meeting
    - Individual meeting with the instructor.
        - Create a sub-folder on your account of the cs server named "**progress2**" and put in the folder all source programs (.java files and image files if any) of your project.
        - Give the demo of your progress from the cs server.
            - 2:00-4:00pm OR 7:30-8:15pm (At the instructor's office)

- Nov 22 (Thur): No class meeting. Happy Thanksgiving Day!

- **Nov 27 (Tue): CLASS MEETING!**
    - Progress demo from the cs server.
        - Create a sub-folder on your account of the cs server named "**progress3**" and put in the folder all source programs (.java files) of your project before the class starts.
        - Each student's demo must be done **within 2 minutes**.

- Nov 29 (Thur): No class meeting; **Try to wrap up the project!**

- **Dec 4 (Tue): Individual meeting (final)**
    - Put your final programs in a folder named "**final**"
    - **Print the revised UML diagram** (based on your actual programs) and bring it to the meeting.

  - Meet with the instructor: 2:00-4:00pm OR 7:30-8:15pm
    - There will be an **oral test** about the program you have developed. You may be asked (but is not limited to):
      - to explain a certain part of your program.
      - to show the code where you used polymorphism / design patterns.
      - to explain why the code is polymorphism / design patterns.

- **Dec 8 (Sat) Midnight**: Due to submit finalized programs and the project documentation.
  - Update programs in the "**final**" folder if you revise after the final demo.
  - Submit design documents (one single PDF file) via D2L that contains:
    - The revised UML diagram which correctly reflects your final program.
    - An essay which explains in detail how you utilized polymorphism and design patterns in your program.
      - This is similar to Section III, IV, and V of the proposal, but you need to revise the UML diagram based on your final program, and **significantly enhance** the discussion of the polymorphism and design patterns based on your final program.