Lab 5

Submit Assignment

Due Dec 5 by 11:59pm **Points** 16 **Submitting** a file upload

During the Lab you will be presenting your running game to the instructor and you will have to succinctly explain how to use your game and how the game works. Instructor will be testing your game for a couple of minutes per team. Instructor might also review your code and ask questions about the code/implementation. Instructor might also send you messages for clarification after the submission deadline.

You have to submit zipped code source until end of the day.

Main requirements for the submission:

- Implemented using MVC pattern
- GUI programming using Java Swing (might use some predefined graphics for some elements, such as playing cards, chess pieces, etc.- if so must reference the source of the graphics)
- Incorporating event-driven programming (correctly defined ActionListeners for graphical components)
- Incorporating one additional design pattern in a game's model is a plus and will result in a higher grade for the project (see Chapter 5 and 10 for design patterns)
- Implementing correct synchronization mechanism between View and Model (only one event can update a Model at a time, and Model updates must reflect the order of the user action), application of concurrency mechanism and thread-safe data structures

Some of the elements taking into account for grading:

- Object-oriented structure, well-designed classes
- Appropriate use of data structures and algorithms
- Modular structure and correct package naming: edu.sjsu.cs.cs151....
- Correct application of the learned object-oriented concepts (when applicable): encapsulation, inheritance, interfaces and abstract classes, enumerations
- Correct application of design patterns (MVC and one additional pattern applied in the Model)
- Correct implementation of the game logic (in the Model package)
- Intuitive user interface - well-thought out design of the UI (well-design, user-friendly interface), correct application and variety of graphical components
- Responsiveness and correctness of the game application- your application must respond in a timely manner for the events generated by the user and respond correctly to the user's input (View and Model must be synchronized)
- Evidence of testing (both unit testing for the chosen classes and comprehensive testing for different scenarios of the game)
- Documentation (using javadoc)

- Evidence of team work (using SVN/Git and repository e.g. GitHub)- each team meber will be in the end graded individually based on their contribution. Your final version must be available under the link you submitted on Lab 2.
- Academic integrity - submitted projects code will be checked for plagiarism with an anti-plagiarism tool. Any violation will result in 0 for the project with no chance for regrading. If you are using any code other than standard Java library you must enter the authors name/source in the comment header. Non-reference borrowing of any code will be treated as plagiarism.