PRESENTATION WALKTHROUGH

Group 0669

Before presenting (project setup)

- Use ‘Run’ -> ‘Edit Configurations’ to set up which classes are included in test coverage calculations.
- Complete some games to have information in the scoreboards for Sudoku and Memory Game only.
- For the two games, have saved games ready that are nearly complete so the TAs can see interaction with the high scores.

Presentation

1. **Introductions:** Use TEAM.md as a reference for what each team member contributed to the project, and each team member will in turn describe their role to the TAs.

2. **Design Patterns**: Preceding the code walkthrough elements of the presentation, one team member will describe the use of the MVC and Observer design patterns that provide the structure for all game implementations. This can be followed by briefly showing examples of related Model, View and Controller classes in the code itself. The most important classes in our program are the interface and abstract classes we’ve designed such as GameManager, GameController, Scoreboard, etc, as they have allowed us to generalize the design and keep a consistent MVC structure across games. The MVC pattern solved the problem of code reuse across games, and allowed us to show the user information (in intuitive ways for very visually driven games) in a completely different way than how we had to represent it internally. The Observer pattern fits into MVC by allowing each part of the MVC triad to react to changes in each other.

3. **Unit Tests:** One team member will take 1 minute to run the unit tests with coverage to demonstrate our coverage of approximately 90% or greater. Be sure to explain how the MVC model described in step 2 precludes the need to do unit testing on Model and View classes, and testing is only required on the classes we’ve chosen to include in our coverage.

4. **Scoreboard Implementation:** One team member will explain the scoreboard implementation, showing the code as it is generalized across games, making sure to explain how user scores are stored and managed. We have a Scoreboard class (using Score objects) which registers a ScoreboardFileSaver, which saves the global scores for each upon notification of observers. The scores are sorted on creation so that when they are displayed in each game's respective ScoreboardActivity screen, both for the logged in player and for the global top scores, the top scores are already in the desired order (with only the top 5 for user and global scores shown).

5. **Best unit test:** Another team member will briefly show the best unit test class in the whole project.

6. **Additional Explanation:** In case any points are missed, any team member can interject here and address those points by explanation and/or demonstrating the relevant code.

7. **Answering Questions:** At this point the TAs will ask the team questions related to what they've just seen. Any team member can answer as their expertise in the project relates to the question.