CSC207 Group 0614 Phase 2 Walkthrough (to be updated)

Welcome to our game center for CSC207 fall 2018.
We are Lizard King Software LTD aka group 0614.

To use our game center, first you must register an account (demonstrate)

- After you fill the form and click register, the app will communicate with our server database through the link https://kayyzz.com/Register.php using the StringRequest class from the android.volley package.
- This Register.php code will direct the request to the database and and store your username, name, email, and password on a database table using SQL.

Logging in does the same thing but in reverse.

- After you click the Login button, LoginRequest.java sends the inputted username and password to the database through https://kayyzz.com/Login.php and SQL and checks to see if the username and password exists. If it exists, a success message is sent back and lets you into your game centre account.

(demonstrate all 3 games)

Our most important classes are:

- GameManger.java

What design patterns did we use?

- Obersable, observer
    - Easy communication between objects
- Factory Method
    - Sliding tiles
    - Easier to test
- Serializable
    - Saving and loading functions
- VMC (view model controller)
    - Don't have to test model
    - Smaller classes and easier to make changes
- Iterator

How did you design your scoreboard? Where are high scores stored? How do they get displayed?

- he scoreboard is an abstract class with several instance variables to account for the level of complexity of a game, the duration that a game play lasts, and the number of moves and undos the current player of a game has made. Each game will have a child class that extends the scoreboard class and implements the abstract method calculateScore() in a way that is meaningful to the individual games themselves. This method also stores the newest score in two sorted arrays that are instance variables of the child class that belongs to the game intended for tracking the global high scores and the user's high scores, respectively, so that when a game ends, a text bubble displays the new score, the user's highest score, and the game highest score. Every new score, along with the corresponding game name and the current user's username, is written into a save file upon creation for the score page to display using the methods in the "[Game Name] ScoreBoard Activity" class in each game.
- How to interpret the scores: In Sliding Tiles, the score is calculated only when a game is won, and naturally the user who plays the hardest version of the game with the largest board in the shortest amount of time earns the highest possible score. In MineSweeper and 2048, the scores are updated constantly in the background during game play after every click or slide movement, and the calculateScore() method is called when the player either loses or wins the game. In MineSweeper, the more of the board is revealed at the end of the game, the higher the score will be, and the amount of time that it takes the player to achieve that score will negatively affect the score but not to a significant extent. In 2048, the scoring system precisely follows the original game's scoring system: every merge between two identical blocks creates a sum that is added to the score.

What is your unit test coverage?