- What is your unit test coverage?
- - Unit Test Coverage is approximately 65%
- What are the most important classes in your program?

  - The most important classes are the classes end with the suffix StartingActivity, and those end with the suffix LoadSaveButtonController, which performs the function of loading/saving game progress.
  - The games are displayed in classes end with the suffix GameActivity with controllers end with ‘GameController’. These classes display the game

- What design patterns did you use? What problems do each of them solve?

  - We used the MVC design pattern alongside with the Observer infrastructure for two games. The Model part is responsible for getting modified user input from the controller, updating the corresponding instances with the input and notifies the controller with the update. The View part is responsible for displaying the model and receiving and updating the view with data input. It interacts with the controller only. The controller part acts as an intermediary between the view and model as it receives user action and converts it into readable instructions for the model to execute. It receives feedback from the Model component and updates in the View.

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

  - We designed a ScoreBoardManager, which is a subclass of GameManager to facilitate the storing/loading process as we implemented a general LoadSave class for the storing/loading process. It is serialized with key parameters – filename, username –(“admin”), and gameType. For the display, the ScoreBoardActivity receives Intent from the ScoreBoardSelectionActivity so we know which game’s scoreboard should be displayed. Each score is inputted after ‘Game Over’, where the score is stored into the ScoreBoardManager with the score and the corresponding username.
  - The ScoreBoardManager and the ScoreBoardActivity which displays the Manager are designed to have flexibility. The display can be easily modified as we used a display algorithm that allows for different numbers of scores displayed(Top5, Top10, Top20). The ScoreBoardManager itself can be changed to contain different numbers of entries.