What is your unit test coverage?

- 21% classes 36% lines

What are the most important classes in your program?

- BoardManagers in all three games, they contain the methods that compute and trigger updating score and swap/change tiles corresponding to the user interaction by tapping or swiping
- GestureDetectGridView that captures the action of the user
- AccountManager performs user management and account creation
- Board ensures no impossible states are reached in the SlidingTileGames, also where the tiles are updated/swapped
- GameActivity controls how to react based on the interactions with the user
- SaveAndLoad handles all the saving and loading of games and score boards

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

- Factory Pattern: Used in the TileFactory classes for all three games, output a list of tiles

  By using this pattern, we could easily switch out and change the classes for tiles when we refactor the tile class or usingg new tile classes
- Observer Pattern: Used in the Board and GameActicity classes

  It will call notifyobserver() when update is needed after swapping tile or updatetile method is called
- Iterator Pattern: Used in going over the lists of tiles in three games, and going over scoreboardkey.

  It goes over these objects in order

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

- Scoreboard activities are designed around the Strategy Design Pattern
  - We have 2 classes (User and Game Scoreboard Activities) that differ only in their behaviour (way in which they save)
  - And the Abstract ScoreBoard Activity is independent from the way in which User and Game Scoreboards save their results
- The Scoreboard itself is just a class that contains hashmaps for user and game scoreboard
  - Each of which map a gameID (specific to each game and each GameSetting must provide) to the GameSetting it self
  - Each GameSetting contains the game id as well as different settings relevant to the game, and who the max score was set by and what it was.
- High scores are stored in the ScoreBoard class which is serialized

        - This is loaded updated and re-saved anytime a user alter's their score
            - So for Alphabet its each swipe
            - For SeaInvaders its each spawn
            - SlidingTiles its upon game completion
- Scores get displayed using a GameScoreBoardActivity and UserScoreBoardActivity
    - Each of which inherit from an AbstractScoreBoardActivity
    - Which implements a table activity to represent the scores as:
        - userID, score, GameID
        - GameID specifies the game type and settings for that game
    - The the specific game and user scoreboard activities specify what's being loaded and passed to the Abstract one to show the results