- Our Unit Tests coverage is 100% for all our abstract superclasses and account and Scoreboard-related methods. However, for unique methods in our games our test coverage is 0% since they are used in gameActivity only and thus cannot be tested.

- The most important classes in our program are AccountManager as it encapsulates most of our other code and serves as a basis for all non-game activity functions, board and board manager abstract superclasses as they lay a foundation for all of our games and the LoadAndSave class which creates, writes and reads all the files necessary for saving and loading.

- The main design pattern we used is the strategy pattern. We created a BoardManager superclass and three subclasses: SlidingTilesBoardManager, MatchingCardsBoardManager, and SudokuBoardManager. The Activity classes only need to contain BoardManager. They do not need to contain specific subclasses of BoardManager. Then they call the methods from BoardManager to talk to the subclasses. When the user plays a specific game, the specific subclass for that game is plugged into the BoardManager. So when Activity calls the method in BoardManager, actually the specific game subclass was used. This is the strategy pattern between any of the Activity files (GameActivity, StartingActivity, etc…) and BoardManager.

There's another strategy pattern between BoardManager and Board. We have a Board superclass and three Board subclasses (one for each of three games). But BoardManager only needs to contain Board and calls the methods in Board. It does not need to know the name of the subclass. When the user plays a specific game, the specific game's subclass will be instantiated.

There is also the Observer/Observable pattern. The GameActivity acts as the Observer and Board is the Observable. GameActivity observes any changes in Board. Then, when Board makes a change, GameActivity will be notified. This code was given by the teacher.

- Our Scoreboard is designed to be integrated into accountManager, since it has access to all the accounts everywhere in the code. Hence, we keep track of the score in individual accounts. Each account has a list of all its scores. The scores themselves are stored as a special class ScoreInfo, which keeps track of three things : score obtained, name of user, and which game the score belongs to. We display the scores through a Listview object by linking it to a String Array formatted by methods in accountManager. Keeping track of all the user’s score in account allows us to display a personal scoreboard as well as a global scoreboard. Additionally, keeping track of the name and the game of the score allows to selectively search through both the personal and global scoreboard.