Milestone 8

David Gartzke, Brodie Lockard, Francis Meng

**Progress this week**

Chance and Community Chest cards, displaying houses on board, controlled die roll, and button for buying houses.

**User Scenario for next Milestone**

Internationalization, information menu, and more than 2 player support.

**Code Coverage**

84%

**Lines per team member**

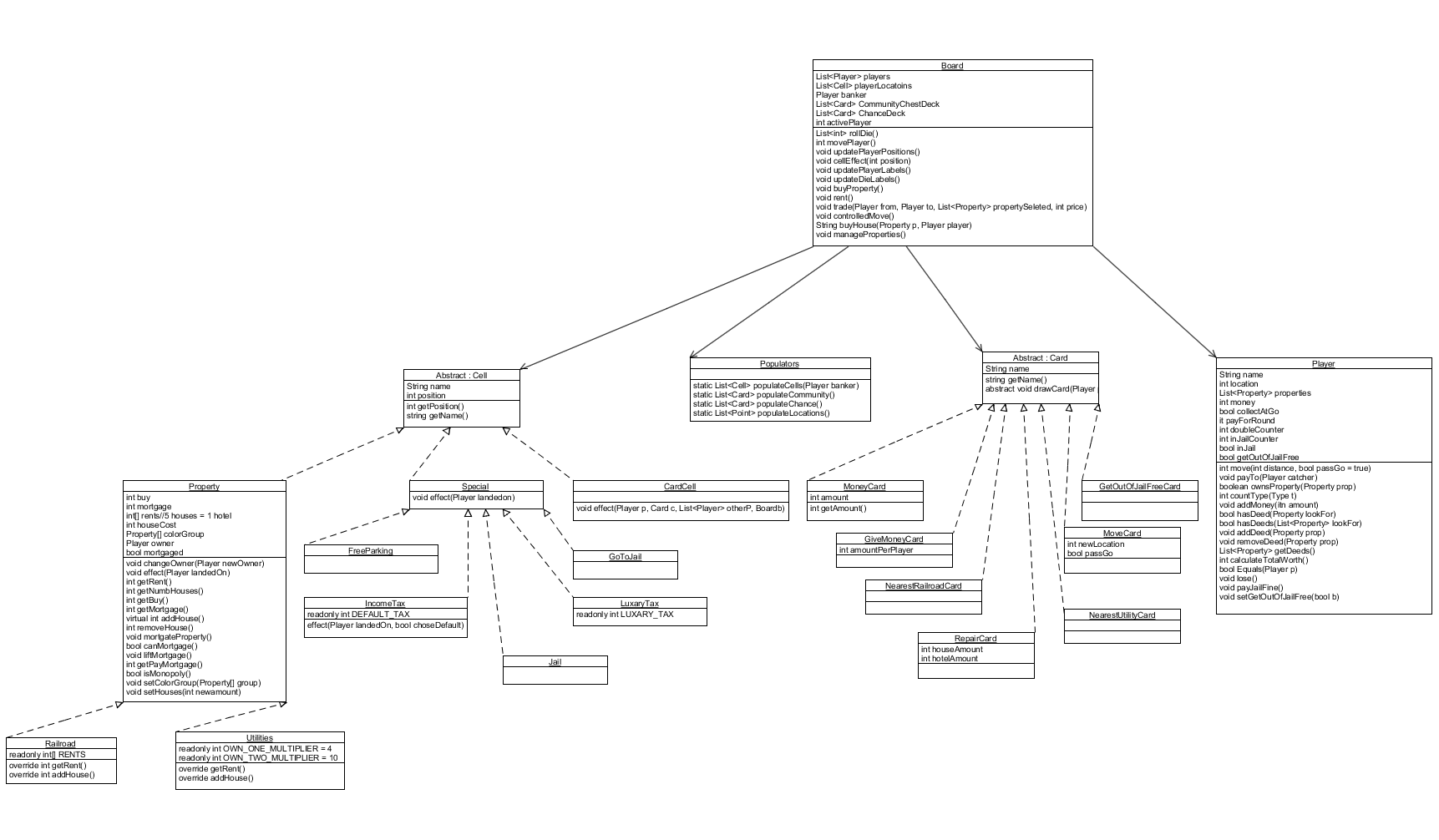
David ~ 250

Brodie ~ 400

Francis ~ 280

**UML diagram**

(next page)



**Code Quality Metrics**

The average lines per function should be no longer than 20 to promote simplicity.

Every class should depend on no more than 5 other classes in order to reduce coupling.

There should be at least 3 tests per function in order to ensure thorough tests.

All branches of if and switch statements must be tested to make sure all possible cases are tested.

Make sure every piece of functionality has been tested.

**“Fun” Metrics**

In order to evaluate how enjoyable our game is to our users, we have established four “fun” metrics: counting the number of questions someone familiar with them games asks while playing the game, timing how long it takes players to give up or get bored with the game, and counting the number of questions someone unfamiliar with the game asks while playing, and measuring how much fun we receive from testing and playing the game. Counting the number questions someone familiar with the game asks allows us to assess the intuitiveness of the GUI. We believe that a complex GUI results in player frustration and thusly decreasing the total enjoyment players’ receive from playing our game. If the players ask a lot of questions, it means our GUI is not intuitive, and we will need to consider reformatting the layout, how events are handled, and possibly color schemes. Similar to this metric is measuring the number of questions someone unfamiliar with the game asks. The point of having someone unfamiliar with the game test our program is to allow us to gather data on how well the rules to the game are explained and if the game grabs new players’ attention. Based on the number of questions acquired, we can modify the start menu and help windows. Because Monopoly can take a long time, and usually people quit early, we want our game to our players to enjoy the game for as long as possible. Therefore, we would measure how long it took players to finish or how long until they became bored with the game and quit. This measurement also gives a measurement of the average play time for our game which can be used for marketing and setting up testing. If most people end up finishing the game, we can work more on making the GUI prettier; however, if most players end up quitting the game, we need to consider adding additional features to keep the players attached and interested in the game. Some features might include shortened game modes where players try to earn a certain amount of money or number of properties. While all these metrics test how people outside of development enjoy the game, we feel it is important to assess how much fun we have fun playing. If we are completely bored or quit early, we cannot expect customers or even testers to sit through a game and have fun. Therefore, this is a metric we can use while constantly developing the code and will give us a rough estimate of how outside players will respond.