Implementation:

Our project is implemented using a model-view-controller architecture. Our models represent the data and the operations that affect it. The views simply display the models. And, the controllers contain the logic to react to user interaction. The most significant model in our game is the Grid. This class defines a matrix of values and is implemented as a two dimensional list.

A number of data structures are used throughout the game. In order to implement the undo function, we use a stack. The LIFO protocol allows us to add the player's next move to the stack and then pop it off when they undo their move. Additionally, a map is used to correlate numbers to colors. Within the grid, integers are stored to represent the number state of each cell. A map is used to translate those numbers to colors that are drawn.

In order to handle multiple views, including the menu, how-to, and game views, we were forced to make a design decision. Instead trusting the game make decisions based on the type of the current view, the views make their own decisions. In order to do this, the views implement a common interface. Each view must be able to draw itself as well as handle a mouse click. Part of the trade off of this decision is that the view takes on some additional logical responsibilities. This is not ideal, but doing so allows the other parts of the game to be ignorant of the specific view.

Reflection:

Looking at the big picture of the project, we did a good job. We completed the project with the features we specified in the time-frame we agreed upon. The project was finished on time with all of the expected features. We met frequently to work on the game and we were both responsible for our tasks. As well, we were able to produce a functional and fun game.

On the smaller scale, things could be improved. While we both took responsibility for our components, we never conducted a code review or took some time to look through each others contributions. Additionally, there could be more testing of each class.

