Name of Project: 2187

Names of Team Members: Nicole Dong, Daniel Winston

Final State of System Statement: In Project 4, we stated that we wanted the following features:

1) user can move blocks around a 6x6 board using arrow keys

2) if there are 3 of the same number in a row, they combine to form powers of 3

3) after the user moves blocks, a new block pops up in an RNG place on the board

4) the user wins when they reach the number 6561. A message pops up with the options to "continue" the game or "start over."

5) squares that are 6561 or above have a sparkle effect

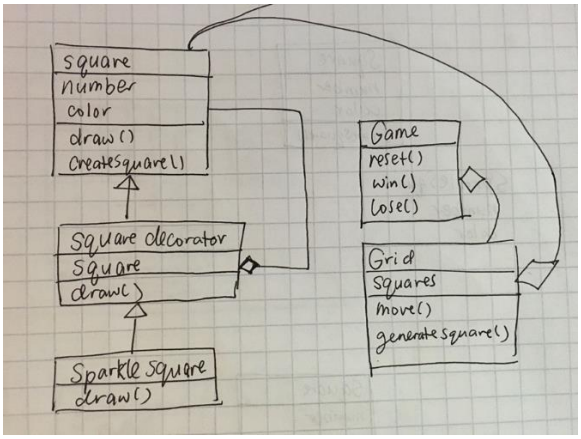
6) the user loses when every space on the grid has been filled up. The user MUST click "start over"

7) the user may choose to start over before winning/losing

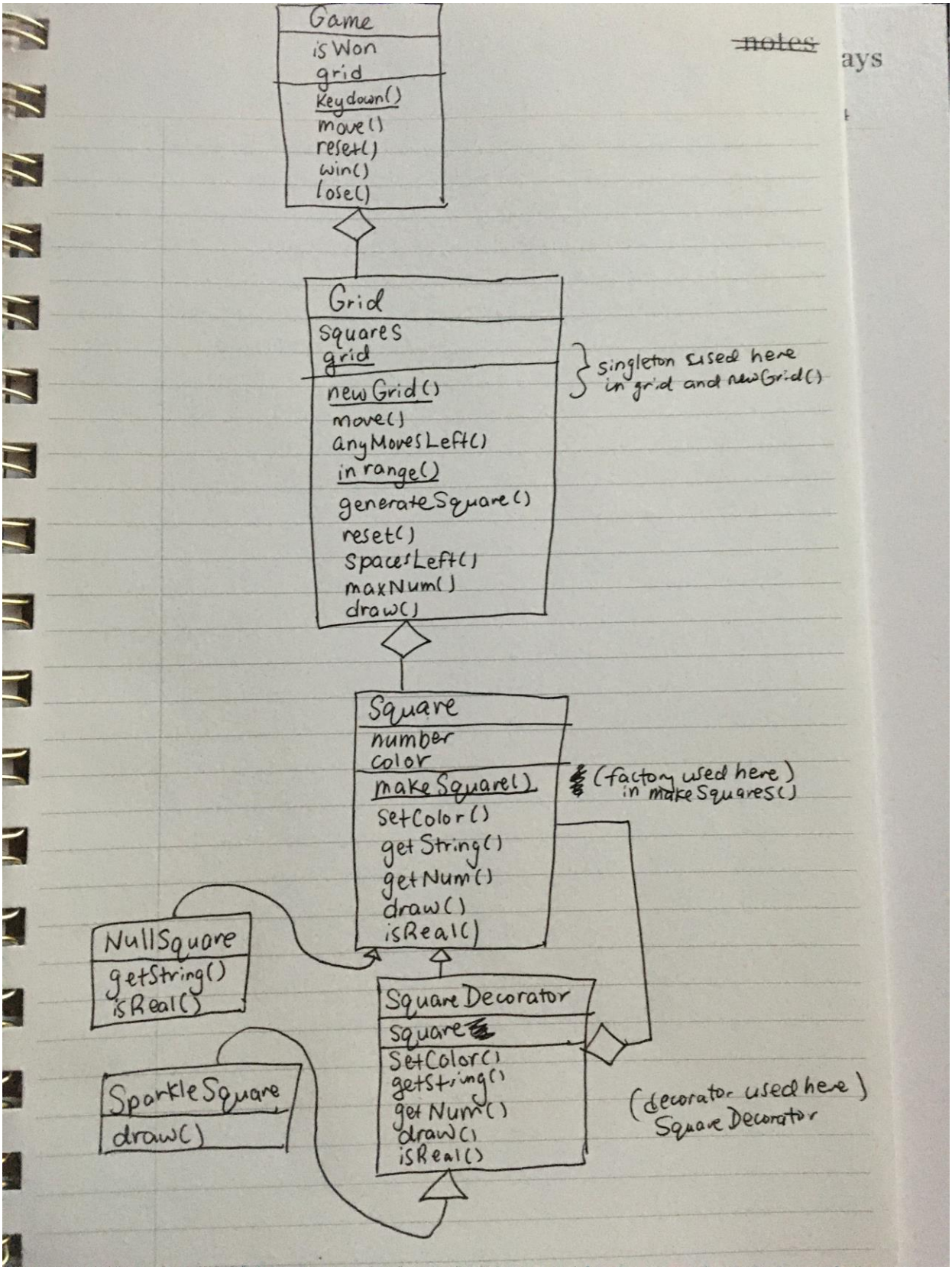
We implemented nearly every single one of the above features, with 2 important changes: First of all, originally, we wanted the winning number to be 6561. However, it was near impossible to get to that number. Thus, we decided to change it to 2187, which is still hard to achieve but it's a much more reasonable number. The second change we made, was that instead of adding sparkle effects to squares that are above the winning number, we decided to make the numbers on those squares turn gold temporarily during each new move (it's hard to explain this using words, I'd recommend seeing the demo video to see what I mean).

Final Class Diagram and Comparison Statement

This was the UML diagram from Project 4:



This is the current UML diagram:



Obviously, a lot has changed. The new UML diagram has a new class NullSquare which wasn't there before, and there are a *lot* more methods, mostly in the Game class and Grid class. The Game class now has a keydown() method, which gets called whenever the player presses a keyboard key. The Grid has many new important methods, including anyMovesLeft(), spacesLeft(), and maxNum(). What these methods do is rather obvious due to their names. The Square class also has a lot of getter functions now, which weren't there during Project 4.

Third-Party Code vs Original Code Statement

When it comes to third-party code, jQuery was imported. https://www.w3schools.com was also incredibly helpful.

Statement on the OOAD Process

1. After playing the game a couple times, we realized that it was near impossible to get to 6561, so we decided to change the winning number to 2187 instead.
2. We had to learn how to use classes in JavaScript as opposed to Java
3. We had to figure out how to use transition times in JavaScript and HTML (the win/lose alerts need to be delayed, otherwise the alerts will show up before the grid updates. Transition times were also needed for the sparkkle decorator)