My experience testing dominion was that I was surprised it seemed to actually play so well online with how easy it was to get it to fail. The programming was inconsistent (sometimes there would be loops hard coded with a value other times loops iterating through the exact same array would go on length) and hard to read at some points (take for example, the draw card function). It relied a lot on functions that had no error checking and were very easy to crash (quick sort, cardeffect and rng to name a few) and made no attempt to handle any errors. It also made very little use of the actual gameState struct. Very few functions manipulated the coins field, even when they were actually increasing the coin count, and the priority seemed to be to keep count of hand and deck cards, and somewhat ignored discard. Overall very good code for finding bugs.

It is for this reason that I find it hard to find additional bugs by examining my classmate's code. When I looked at Curtis Minks code, I found very few changes from the actual main game of dominion. It was kept as similar as possible and as such failed all the tests that my dominion code failed before I made debugging changes. I was able to find and document three bugs relating to his code alone since they were more bugs pertaining to dominion implementation itself and his code was very similar to the original implementation.

When I looked at Harrison Kaiser's code, I saw that he implemented the coin bonus in a number of cardEffect cards, and removed the temporary hand from feast as well (appears that it is unnecessary with his implementation). It also seems to have a lot of code removed throughout to streamline the code.