“Team48 commits to this code base!”

Moving forward with assignment 6, we have decided to use Calvin’s code base and will working in C#. Bella is comfortable with C#, has worked with it before, and is familiar with the different libraries and general functionality. Previously, her code base was written in Python, and Calvin prefers C# over Python because he is more experienced in the C#. Calvin is, however, familiar with the small number of libraries used in Bella’s code base.

Calvin’s code base abstracts various parts of the assignments quite nicely, is organized well, and passes all of the peer tests. Bella’s code base performs sub-optimally, and Python passing by reference created many bugs. We decided to use Calvin’s code base because it seemed better equipped to be modified for future assignments and was consistent in behavior and correctness. Components implemented for each assignment provide readily usable and accurate functions for other components to use. Functions also throw detailed custom error messages that provide detailed information on why the component failed.

Bella’s code base is not very modular and many functions that could be split up were all placed into the same class in the same file. It also had a lot of repetitive code that could be abstracted away to be reused instead of copied. Calvin’s code base is more modular, as there are separate classes and functions for all parts of the game. In addition, each component is split up into a private class, wrapper, and adapter to give more options for unit testing and interaction. In terms of interfacing, both code bases tried to implement protocols to mandate that inputs given to the classes are correct.

In summary, we are choosing to work with Calvin’s code base because it is more modular and flexible for use in future assignments.