Integration:

Our team integrated the functionalities of the project using a combination of implementing everything then integrating and the sandwich integration strategy. For project 3 and the start of project 4, we did not have any tests for the testing suite, so we focused on writing and implementing the code for a basic prototype. The structure for this project was much less defined, as more time was dedicated to writing documentation, so fewer members were writing code simultaneously. Because of this, we were not as heavily reliant on sandwich integration as with project 1 and 2. Thus, the essentials of project 3 utilized the implementing everything then integrating strategy. After that, we started with some logic artifacts for core functionalities and game structure alongside some operational artifacts such as the behavior of the pieces. For example, the first items when created were items in main.js that dictated the flow and operation of the game, which were logical artifacts. Alongside the implementation of main.js there was pieceLogic.js which were behavioral artifacts for the operation of the game and generating various pieces for the game. Towards the end of the project, we wrote the artifacts that integrated those various components and acted as intermediates between them. When we had enough functioning pieces of code, we integrated them and wrote tests for them. Currently, we do not use automated integrations. As such, the tests do not run automatically, but is run manually through index.html in the test directory. This should be done between changes. Additional changes and future improvements would be able to more closely follow the sandwich integration strategy.