For Project 4, our team (Team 2) used Top-Down Integration. We began by starting with all of the code artifacts that needed to send a message to an artifact below. The main page that is displayed on the screen upon loading was done by Amanda Nelson. In the example given in lecture, this would be like stub “a”. The main page displays several options for the player to choose from, one of those options being which level to play. Celia Babst was in charge of completing several of the level options, and she waited for Amanda to complete the main page before she was able to link her levels to the main page. In the example this would be like stub “b”. Kevin Carlson was in charge of creating our craziest level, and he waited until Celia created the intermediate levels to create his level because that piece of code artifact relied on the ones she had created. This stub was branching off of stub b, so looking back at the example from lecture it’s stub “e”. Kayton Froeschl was in charge of displaying a timer during gameplay. She didn’t need to rely on the level creation for this, only for Amanda’s main page. So her timer artifact is like stub “c” in the picture. As you can see from this example, our operational artifacts are found in the lower levels of our integration. Our main page, which is a logic artifact, is situated close to the root because it’s part of the decision making flow. All of these different components of our integration prove it to be Top-Down Integration.