

For our implementation of project 3 our group chose to create a turn based battle game where human controlled characters would fight against AI controlled enemies. We created an activity diagram that shows the flow of our game and code. As a group we decided to code in javascript with function oriented design. In the context of function oriented design we also decided that pipes and filters would be our main software architecture. The activity diagram that shows the flow of the game and code also shows the connections of the pipes and how the data flows through the code. The main flow for the player side of the code is player selection into action selection into target selection. With the filters we intercept the pipes at each stage to accomplish things like skipping dead players on the player selection, and making sure the player has enough magic or inventory to perform the chosen action at the action selection. Finally, in the target selection we ensure that correct targets are selected. This also occurs on the back end where values must pass through filters to accomplish things like changing overheal to just max health, or truncating any non whole numbers so that the values printed to the screen remain simple. Another use of the pipes and filters is to pass around objects through the functions and insure that the proper effects are applied to them. The character being attacked will commonly be pipped through 4 or 5 functions before the action is completed so pipes and filters was a very good fit for our project.