EECS 448: Software Architecture
Team 7
3/28/2021

For Project 3, Team 7 has chosen the Pipes-and-Filters format for this project's design architecture. The reason the team has decided to use this architecture is that the game has multiple entities in each level, which include level hazards, complex AI enemies, the Player Character, interactable objects and structures, and special level features such as one-direction permeable floors or condition-destructible blocks. When a user performs a keystroke or mouse-click as an input, that input will need to be run through various classes in order for the game to function as intended. For example, if the keystroke to move the Player Character forward is pressed, the Player Character will have a function that moves them forward through the level space. That movement can then act as an input for a function that determines the Player Character's proximity. The output of this supposed proximity function could then go on to act as the input for an enemy AI to perform some action or as the trigger for some proximity-based level hazard. The Pipes-and-Filters design architecture is the best fit for the project, as its simplicity removes the burden from the programmers of having to put excess effort into understanding the behavior of the game system, which in turn assists the team in producing more deliverables faster. The sequential nature of the Pipes-and-Filters design architecture also allows for the architecture to be maintained with greater ease as well as transplanted into other areas of the project, which also works toward increasing the team's productivity. Due to the truncated schedule, this focus on increasing productivity and decreasing the burden on programmers is paramount.