Space Game – Using Stubs and Test Methods

To use stubs, your application must be designed so that the different components are not dependent on each other, but only dependent on interface definitions. Instead of being coupled at compile time, components are connected at run time. This pattern helps to make software that is robust and easy to update, because changes tend not to propagate across component boundaries.

Our first test would be to ensure that our method that allows use to utilize the menus. It would have to ensure that the program can only continue if the user enters a valid option on all menus to include them traveling to another planet, buying and selling goods, and being able to refuel their ship.

We would next ensure that the proper amount of fuel is being discarded after the user travels to another planet. Also, we would ensure that the proper amount of money is being spent when a user buys product from other planets. Likewise, an opposite action is occurring when a user goes to sell his products. We would also have to test that the fuel is being filled to max capacity when a user refills his tank, and that he is not allowed to overfill his tank when he refills his tank.

Another feature that needs to be tested is the amount of fuel being wasted when traveling to another planet. Each planet will have a specific distance from each other, so we would have to test travel from every planet to every planet, to ensure that everything is working properly. We also need to check that user is denied the able to travel to a given planet if he does not have the capabilities to travel that far with a less amount of fuel needed to make the journey.

The last major function that needs to be tested is to ensure the entire program is functioning all together as a smooth game. The game will not be made too hard to win and should not be easy to lose (go bankrupt) for the sake of simplicity. We will be monitoring for any glitches that may occur within the program. We will also we are checking that all possible exemptions are being handled accordingly to ensure a great experience for the user. We also need to monitor that we can continue to buy and sell goods but that our money has a certain cap promoting the user to finally travel to the last planet to buy the final ship that will end the game.

All in all, the tests are going to be essential to discovering any problems that will arise in our game. There are going to be a lot of different pieces that will make the game function properly and we as developers need to ensure that we provide the product in a non-problematic experience for the user. This project provides this principle in a fashion that will teach us exactly how a giant problem can be narrowed down to small pieces and be tested for functionally. This experience will stimulate that mentality in our minds when designing big projects in the future, and ultimately brings all that we have learned into a single collaboration.