**Group 4 Cases or Test Method Stubs**

In our space game “Bounty Collector” we will be implementing quality assurance to ensure that the user is interacting with a robust program. To accomplish this, we need to implement error exceptions as a fail safe to any errors the user might input. We will also implement decision statements to help guide the user across the game by narrowing down the room for error. Some examples are listed below:

**Hubb (Planet Screen)**

When the user is prompted to select from browsing the shop or to continue the mission, the user will have two clear choices to choose from:

1. **Browse the Shop.**
2. **Continue the Mission.**

If the user enters an input that is not **1** or **2**, the catch exception handler should be able to display to the user that he has to press the correct key and the program will re-prompt the user. On the other hand, if the user presses **1** or **2** the program will enter the corresponding screen. The program will decide how to react with the usage of decision statements, such as IF (input ==1) then (enter Shop screen).

**Fighting Phase**

When the user is at a planet and is fighting either a minion (weak entity) or a boss (strong entity) he/she will have to solve a math equation. If the user gets the answer right, then there will be a decision statement in the program that will handle that. If he chooses wrong, there will also be a decision statement that will trigger (in this case a loss of a life point).