Junit Test Writeup M2

* TestFXM2
  + testGameSingletonInit
    - What it does
      * checks if the Game class singleton is non-null after starting the program.
    - why:
      * Since one game object persists throughout the session, and it must be globally visible, a singleton pattern fits it best.
      * The program (TestFX application tester) automatically calls the parameter-less constructor of Game, so the singleton instance should be assigned at the constructor.
  + testGameSwitchScreen
    - what it does:
      * checks if the switchScreen method correctly throws errors when given meaningless destinations. Such a destination ("hello world") is fed to the method, and an IllegalArgumentException is expected.
  + testInitialConfigCheckerA1, testInitialConfigCheckerA2
    - What they do:
      * A1 and A2 check if the initialConfigurationScreen handles erroneous input gracefully
      * A1 checks for empty string
      * A2 checks for white space
      * Null cannot be checked because users cannot input that to the program without editing the sourcecode
  + testInitialConfigCheckerB
    - What does it do:
      * B checks if the initialConfigurationScreen handles correct input gracefully (namely, it transfers the player to the next screen)
    - Why: Debugging should handle correct scenarios as well as incorrect ones
  + testWeapon
    - What it does:
      * Check if the radio button for selecting weapon is correctly selected.
    - Why:
      * The player can select different weapon. We want to make sure that the radio button for selecting weapon is working properly.
  + testDifficulty
    - What it does:
      * Check the slider for selecting difficulty is working properly.
    - Why:
      * The player can select different difficulty. We want to make sure that the slider for selecting difficulty is working properly.
* PlayerTest
  + TestName, TestWeapon
    - What it does:
      * check if the player name and the weapon function work correctly.
    - Why:
      * the player might enter different strings in the tab, including an empty string, the test makes sure the name and weapons function works correctly with those different strings.
  + TestDifficulties, TestMoney
    - What it does:
      * check if the game difficulty and player money function work correctly
    - Why:
      * the player might enter different integers in the tab, including 0 or nagetive numbers, the test makes sure the game difficulty and player money function works correctly with those different strings.