We have researched unit tests in junit through gradle, but the NetBeans version we implement can only use junit4 while gradle uses junit5 tests. Additionally, these tests would need to be implemented through a line of code in the command prompt in order to be properly used, which is not to the demands of the rubric. Instead, we are defining what the unit tests could've been and are creating mock files within the program to reflect the fact that we have thought about and attempted unit tests.

https://github.com/junit-team/junit4/wiki/Use-with-Gradle

Case Scenarios by Class for TurboStar

1. AllyBullet
    a. update(): set the position and hit box location and then compare to the values anticipated. Pass if true.
    b. DetectCollisions(): put an enemy in the array list and then remove them when the variable for collision detection is set to true.
2. EnemyBullet/Bullet
    a. Set the value of BULLET_SPEED and then compare to the anticipated value. Return true if they match.
3. EnemySpawner
    a. SpawnEnemy(): Put an enemy into the array. If there is an enemy in the array, return true and the test passes.
4. MovementBase/MovementSerpentine
    a. update(String currStat): Set variables of the screen bounds. Pass in currStat outside the screen bounds and check to see if the image is within the screen bounds and pass if it falls outside these bounds.
    b. render():
    c. CheckBounds(String currStat):
5. GameOverScreen:
    a. update(String currStat): Pass in a currStat of "GameOver" and see if the output equals intended output.
    b. render(Graphics g): Pass in a graphics for (string, xpoint, ypoint) and see if the output equals intended output
6. PauseScreen:
    a. update(String currStat): If the R key is pressed and this is sensed by the program and the test is true. Otherwise if another key is detected the test fails.
    b. render(Graphics g): Pass in a graphics for (string, xpoint, ypoint) and see if the output equals intended output
7. PlaterData
    a. AddScore(String name, int score): Add a score to the player score array. Check the array for any values. If there is a value in the array, the test passes.
    b. RetriveScores(): Have a value hard coded into an array. Make the program print this score. If a value is printed, then the test passes.
8. StartSplashScreen

    a. update(String currStat): If the SPACE key is pressed and this is sensed by the program and the test is true. Otherwise if another key is detected the test fails.
    b. render(Graphics g): Pass in a graphics for (string, xpoint, ypoint) and see if the output equals intended output
9. StorageHandler
    a. See mock unit test named StoragehandlerTests within the Source Packages.