# Feature #2: Class Loader

**Coders:** Kiam

**Testers:** Justin

**Feature scope:** This method is located in the GameWorld class and reads tracks from their java file, it compiles the track java files, And finally, it builds the dyn4j world.

**Summary:** This method builds the track that cars will race on by loading the track's java file and running the buildWorld method found in that class. It gets the names of the tracks by creating an array based on the images of the track with the same name found in the res folder.

**Testing:**

1. Loading a class and calling a function that returns -50
   - Status: Tested
   - Expected Result: Class will load and return -50.
   - Actual Result: Returns -50
     - How the test was ran: Automatically
     - How was it tested: A java file was created with a method that returns -50. This class was then loaded using the class loader and the return value checked.
2. Loading a class and return a string, "hello world"
   - Status: Tested
   - Expected Result: Class will load and return "hello world"
   - Actual Result: Returns "hello world"
     - How the test was ran: Automatically
     - How was it tested: A java file was created with a method that returns "hello world". This class was then loaded using the class loader and the return value checked.
3. Loading a unfinished class.
   - Status: Untested
   - Expected Result: Class will fail to load throwing null pointer error.
   - Actual Result: N/A
     - How the test will be run: Automatically
     - How was it tested:  N/A
4. Loading a class with java runtime errors.
   - Status: Untested
   - Expected Result: Class will fail to load and corresponding runtime error

- ○ Actual Result: N/A
    - ■ How the test will be  run: Automatically
    - ■ How was it tested: N/A
5. Loading a class with java syntax errors.
    - ○ Status: Untested
    - ○ Expected Result: Class will fail to load and throw null pointer error.
    - ○ Actual Result: N/A
        - ■ How the test will be  run: Automatically
        - ■ How was it tested: N/A
6. Loading a track class and visually confirming the track loads.
    - ○ Status: Tested
    - ○ Expected Result: Class will load and display the correct elements
    - ○ Actual Result: Track loaded and displayed correct elements.
        - ■ How the test was ran: Manually
        - ■ How was it tested: A track file was created and loaded, the test then visually confirmed that the elements were loaded correctly.

**Design Decisions:**
- ● TBD