Test framework implementation review

- The code is readable and logically separated.
- The implementation of the test framework adheres to the high-level architecture figure.
- While implementing the integration tests, some minor bugs were detected. They were mostly related to refreshing and maintaining the different states of the World object:
  - refreshing turnout directions,
  - and the segments' SectionControls

  maintained by World class.
  These bugs were fixed in several commits right after the detection.

- The Assert classes responsible for test evaluation happened to be perfectly adjusted to the message queuing & event based logic. It was easy to create thorough test assertions for all major events.
- The JSON descriptors do their job well: they contain and provide all necessary information for decent integration testing.
- **The execution of the integration tests revealed that the movement of the trains is quite problematic, as the steps for reallocating a train is not clearly defined. When executing this event, one section will turn from FREE to OCCUPIED, and another one just the other way round. But none of the above works as expected:**
  - **When setting the section behind to FREE first, and then setting the section ahead to OCCUPIED afterwards, the TSM responsible of the two sections will send a false ALLOWED HeartBeatSignal as the section is considered to be free. It will send DENIED again as soon as the train occupies the other section.**
  - **When setting the section ahead to OCCUPIED first, and then setting the section behind to FREE afterwards, both sections will be OCCUPIED at the same time which results in DISABLED state for both sections.**

  The solution could be that these state changes happen together atomically, which seems to be impossible using the current messaging system and logic.

  An issue was created reporting this bug.