MONASH University
Information Technology

Advanced Programming

FIT3140

Assignment 5

Team 29 – Matthew Ready, Li Cheng

Name	Student ID
Matthew Ready	25121987
Li Cheng	24864099

Test Plans

Test 1 - Decode message test

- Brief rationale
  This test is designed to test the user story "*As an app user, I want to decode the motion sensor messages into English letters based on Morse code coding table*." We want to make sure that the morse code interpreter is working correctly to decode motion sensor messages into corresponding English letters in the morse code table.

- Setup procedure
  First, run `` `npm install` ``. Then, run `` `./node_modules/mocha/bin/mocha` `` to run all tests. This test will appear as the "MorseInterpreter" test in the test output.

- Test procedure and expected results
  To test that the motion sensor messages are being correctly interpreted, we want to make sure that the `MorseInterpreter` class is working correctly by feeding it virtual input messages which have a predetermined decoded result. We will then assert that the expected results are equal to the test results. Secondly, we want to make sure that the expected decoded result is surrounded by tildes ("~") when no matching letter in the morse code table can be found. This will allow the user to see that a transmission error has occurred.

  This is the only class we will test for this user story, we want to separate out all other functionalities and just test this unit of code for its correctness.

  We have decided upon the following test cases and expected results, assuming that a long motion is 300 time units long.

  - Success tests (testing a single-letter message)
    - Expect decoded result "E", for input message "S".
    - Expect decoded result "M", for input message "LL".
    - Expect decoded result "P", for input message "SLLS".
    - Expect decoded result "", for input message "".
    - Expect correct decoded result for morse code of the letters "MORSE CODE"
    - Expect correct decoded result for morse code of the letters "SPHINX OF BLACK QUARTZ JUDGE MY VOW" (every English letter is used)

  - Exception tests (testing a single-letter message)
    - Expect correct decoded result for morse code of the letters "MORSE CODE", even when the timing is not perfectly correct (dots >100 units, dashes >300 units, spacing > 300 units, etc.)
    - Expected tildes are used to represent unknown codes (i.e. "...===.=.=.=.===..." yields "~LSSSL~"), and that the morse coder can

correctly continue after some unknown codes. We use the following sequence:

```
=.=.=.=.=...===.===.===.===.===...===.=.=.=.===...=.=
.=.......===.===.===...=.=.=
```

and expect

```
~SSSSS~~LLLLL~~LSSSL~S OS
```

as the result.

Messages surrounded with tildes "~" are used when no matching letter in the morse code table can be found. Eg. ~SSLL~ does not have a matching letter in morse table. We use tildes since there is no morse code representation of them, so the user can be confident that an error has occurred and that they can see the data that could not be interpreted, rather than silently losing data.

Test 2 - ShortLongInterpreter test

- Brief rationale
  This test is designed to test the *"As an app user, I want the server to determine whether the incoming message is long or short."* user story. We want to make sure that the morse code interpreter is working on a correct interpretation of the underlying `motionstart` and `motionend` events into long and short signals.

- Setup procedure
  First, run `` `npm install` ``. Then, run `` `./node_modules/mocha/bin/mocha` `` to run all tests. This test will appear as the "ShortLongInterpreter" test in the test output.

- Test procedure and expected results
  To test that the motion events are being correctly interpreted, we will make sure that the `ShortLongInterpreter` class is working correctly by feeding it virtual events and asserting correct results. This is the only class we will test for this user story, we want to separate out all other functionality and just test this unit of code. We have decided upon the following test cases and expected results, assuming that a long motion is considered to be 300 units long:
  - Basic tests (testing a single motion):
    - should signal "long at 0" for events "start at 0, end at 400"
    - should signal "long at 0" for events "start at 0, end at 350"
    - should signal "short at 0" for events "start at 0, end at 250"
    - should signal "short at 0" for events "start at 0, end at 200"
    - should signal "short at 0" for events "start at 0, end at 1"
  - Error condition tests (two starts or two ends in a row)
    - should signal "short at 0" for events "start at 0, end at 100, end at 300" (2 end events error condition)
    - should signal "long at 0" for events "start at 0, start at 100, end at 300" (2 start events error condition)
  - Complex tests (testing multiple motions):
    - should signal "short at 0, short at 150" for events "start at 0, end at 100, start at 150, end at 300"
    - should signal "short at 0, long at 150, short at 601" for events "start at 0, end at 100, start at 150, end at 600, start at 601, end at 602"