

Test Report: Unit Verification and Validation Plan for Sun Catcher

Sharon (Yu-Shiuan) Wu

December 21, 2019

1 Revision History

Date	Version	Notes
2019/12.21	1.0	First Version
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

[symbols, abbreviations or acronyms – you can reference the SRS tables if needed —SS]

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Functional Requirements Evaluation	1
4	Nonfunctional Requirements Evaluation	5
4.1	Usability	6
4.2	Performance	6
4.3	etc.	6
5	Comparison to Existing Implementation	6
6	Unit Testing	6
7	Changes Due to Testing	6
8	Automated Testing	6
9	Trace to Requirements	6
10	Trace to Modules	6
11	Code Coverage Metrics	6

List of Tables

1	Actual Input and Expected Output	1
2	Actual Input and Expected Output	2
3	Actual Input and Expected Output	3
4	Actual Input and Expected Output	4
5	Actual Input and Expected Output	4

List of Figures

This document is the unit testing report for the Sun Catcher

3 Functional Requirements Evaluation

1. InputBounds-id1

This test is testing the ability of identify the boundary of Φ_P .

This result will be display in the ternimal when run the testing.

id	Input	Output
id1.1	90	valid
id1.2	-90	valid
id1.3	89.9	valid
id1.4	-89.9	valid
id1.5	0	valid
id1.6	91	invalid
id1.7	-91	invalid
id1.8	90.1	invalid
id1.9	-90.1	invalid

Table 1: Actual Input and Expected Output

```
Unit tests
InputBounds tests id1
id1_1: OK
id1_2: OK
id1_3: OK
id1_4: OK
id1_5: OK
id1_6: OK
id1_7: OK
id1_8: OK
id1_9: OK
```

The result of the test shows that every test cases under Input Bounds-id1 has matched to its expected output. Therefore, the test case success.

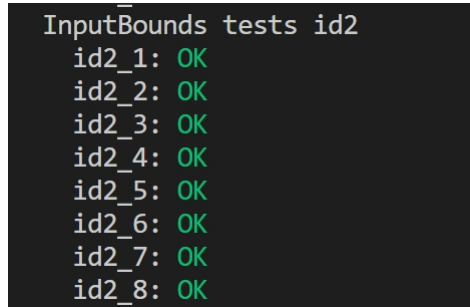
2. InputBounds-id2

This test is testing the ability of identify the input cases dayT, ($year_{Start}$, $month_{Start}$, day_{Start}), whether or not exit in the calender.

This result will be display in the ternimal when run the testing.

id	Input	Output
id2.1	(0, 0, 0)	not exist
id2.2	(-1, -1, -1)	not exist
id2.3	(2020, -1, 29)	not exist
id2.4	(2020, 02, -1)	not exist
id2.5	(-1, 02, 29)	not exist
id2.6	(2020, 13, -1)	not exist
id2.7	(2020, 02, 29)	not exist
id2.8	(2020, 02, 28)	exist

Table 2: Actual Input and Expected Output



```
InputBounds tests id2
id2_1: OK
id2_2: OK
id2_3: OK
id2_4: OK
id2_5: OK
id2_6: OK
id2_7: OK
id2_8: OK
```

The result of the test shows that every test cases under Input Bounds-id2 has matched to its expected output. Therefore, the test case success.

3. InputBounds-id3

This test is testing the ability of identify the end day whether or not smaller than the start day.

This result will be display in the ternimal when run the testing.

id	Input	Output
id3.1	(2020, 02, 28) - (2021, 02, 28)	valid
id3.2	(2020, 02, 28) - (2019, 02, 28)	invalid
id3.3	(2020, 02, 28) - (2020, 01, 28)	invalid
id3.4	(2020, 02, 28) - (2020, 02, 27)	invalid
id3.5	(2020, 02, 28) - (2020, 02, 28)	invalid

Table 3: Actual Input and Expected Output

```

InputBounds tests id3
id3_1: OK
id3_2: OK
id3_3: OK
id3_4: OK
id3_5: OK

```

The result of the test shows that every test cases under Input Bounds-id3 has matched to its expected output. Therefore, the test case success.

4. calculation-id4

This is the test case that testing the ability to calculate the Zenith angle.

It is testing by calling the exported access programs, getzenList, under the Calculation Module.

The the input can be

nd under the path :/src/tiltAngPro/test/tests” Input File Name : “id4.calculation”
Output File Nmae: “id4.calculation.golden”

```

.calculation
id4_1
Input 23.250943 Output 23.250943 Absolute Erros = 0.0
id4_2
Input -23.250943 Output -23.250943 Absolute Erros = 0.0
id4_3
Input -69.250943 Output -69.250943 Absolute Erros = 0.0

```

id	Input	Output
id4.1	43.250943	23.250943
id4.2	-43.250943	-23.250943
id4.3	-89.250943	-69.250943
id4.4	89.250943	69.250943

Table 4: Actual Input and Expected Output

id4.4

Input 69.250943 Output 69.250943 Absolute Erros = 0.0

This result shows that the absolute errors of every test case under calculation-id4 is 0. Therefore, the test case success.

5. calculation-id5

This is the test case that shows the ability to calculate the sun intensity.

It is testing by calling the exported access programs, sglSunIn, under the Calculation Module.

The the input can be

nd under the path :/src/tiltAngPro/test/tests” Input File Name : “id5.calculation”

Output File Nmae: “id5.calculation.golden”

id	Input	Output
id5.1	23.250943	0.9738212
id5.2	-23.250943	0.9738212
id5.3	-69.250943	0.5786897
id5.4	69.250943	0.5786897

Table 5: Actual Input and Expected Output

.calculation

id5.1

Input 0.9738212193883772 Output 0.9738212

Relative Erros = 1.9909586201904972e-8
 id5_2
 Input 0.9738212193883772 Output 0.9738212
 Relative Erros = 1.9909586201904972e-8
 id5_3
 Input 0.5786890162786223 Output 0.5786897
 Relative Erros = 1.1814991309755385e-6
 id5_4
 Input 0.5786890162786223 Output 0.5786897
 Relative Erros = 1.1814991309755385e-6

This result shows that the relative error in every case under calculation-id5 is close to 0. Its satisfies the error tolerance defined in Unit VnV plan. Therefore, the test case success.

4 Nonfunctional Requirements Evaluation

1. code walkthrough :

Attendant : Sharon (Yu-Shiuan) Wu, Doctor. Kal

Implement Date : 2019/12/19

Agenda :

- Check whether or not has the redundant code.
- Check whether or not the code is readable.
- Check whether or not the code is implementing with the concise pattern.

Conclusion for further improvement:

- Use lhs2tex to implement the code and the command.
- Use pattern guard to simplifies the pattern matching.
- Import external framework, AC-Angle, to convert radian to degree.
- Use writefile to write the information to the file instead of using appendFile.

4.1 Usability

4.2 Performance

4.3 etc.

5 Comparison to Existing Implementation

This section will not be appropriate for every project.

6 Unit Testing

7 Changes Due to Testing

8 Automated Testing

9 Trace to Requirements

10 Trace to Modules

11 Code Coverage Metrics