Software Test Report (STR)

Software It Counts (SWIC)

CMSC 447

Updated on November 20, 2015

Table of Contents

[1 Scope 3](#_Toc432634202)

[1.1 Identification 3](#_Toc432634203)

[1.2 System overview 3](#_Toc432634204)

[1.3 Document overview 3](#_Toc432634205)

[2 Referenced documents 3](#_Toc432634206)

[3 Overview of test results 4](#_Toc432634207)

[3.1 Overall assessment of the software tested 4](#_Toc432634208)

[3.2 Impact of test environment 4](#_Toc432634209)

[3.3 Recommended improvements 4](#_Toc432634210)

[4 Detailed test results 4](#_Toc432634211)

[4.1 (Project-unique identifier of a test) 4](#_Toc432634212)

[4.1.1 Summary of test results 4](#_Toc432634213)

[4.1.2 Problems encountered 5](#_Toc432634214)

[4.1.3 Deviations from test cases/procedures 5](#_Toc432634215)

[5 Test log 5](#_Toc432634216)

[6 Notes 6](#_Toc432634217)

[A. Appendixes 6](#_Toc432634218)

# Scope

This section shall be divided into the following paragraphs.

## Identification

For this project, we will be working with the web application, Parable of the Polygons. The software will simulate segregation levels between three different shapes, allowing them to move to different locations on the board in an attempt to be happy. The website will be programmed in HTML while the application will be programmed in JavaScript. The repository is located at <https://github.com/Kirkas1/polygons>, and branches off the source code at <https://github.com/dncnmcdougall/polygons>.https://github.com/Kirkas1/polygons/tree/gh-pages/documents

## System overview

This project has tasked us with inserting a 3rd polygon, a red circle, into the game, allowing for relations between 3 different shapes to be shown. In addition, two new algorithms will be created. One such algorithm will be based around movement when based on the happiness levels of single polygons, henceforth known as the “happiness algorithm.” The second algorithm will be based around the happiness of single polygons, as well as the happiness of those in the 8 squares adjacent to them, henceforth known as the “collective happiness algorithm.”

The following document will address the test results produced by the system (in fulfillment of the requirements given by the customer Russ Cain). The subsequent test report for the project will thoroughly detail the test outcomes carried out by the group (SWIC). This document will also be uploaded to the project’s repository at <https://github.com/Kirkas1/polygons>, which has the history of system development thoroughly summarized<https://github.com/Kirkas1/polygons/tree/gh-pages/documents>.

## Document overview

This document will assess the testing of the system and the results of each test (based on requirements for the project, as previously discussed by customer). The overview of the testing shall include an overall assessment of the software tested and recommended improvements. It will also discuss each test and result given by the STD, any problems encountered, and any deviations from the test case. The test results and handling shall be carried out by the group (SWIC), and the findings shall be traced back to the requirements and then published.

# Referenced documents

1. Parable of Polygons, Revised Apr 18, 2015, <https://github.com/ncase/polygons><http://ncase.me/polygons/>, Vi Hart and Nicky Case
2. Parable of Polygons Source Code, Revised Oct 25, 2015, <https://github.com/ncase/polygons>, Vi Hart and Nicky Case
3. Polygons, Revised Dec 9, 2014, [https](https://github.com/ncase/polygons)://github.com/dncnmcdougall/polygons<https://github.com/ncase/polygons>, Duncan McDougall

# Overview of test results

This section shall be divided into the following paragraphs to provide an overview of test results.

## Overall assessment of the software tested

This paragraph shall:

1. Provide an overall assessment of the software as demonstrated by the test results in this report
2. Identify any remaining deficiencies, limitations, or constraints that were detected by the testing performed. Problem/change reports may be used to provide deficiency information.
3. For each remaining deficiency, limitation, or constraint, describe:
4. Its impact on software and system performance, including identification of requirements not met
5. The impact on software and system design to correct it
6. A recommended solution/approach for correcting it

## Impact of test environment

This paragraph shall provide an assessment of the manner in which the test environment may be different from the operational environment and the effect of this difference on the test results.

## Recommended improvements

This paragraph shall provide any recommended improvements in the design, operation, or testing of the software tested. A discussion of each recommendation and its impact on the software may be provided. If no recommended improvements are provided, this paragraph shall state "None."

# Detailed test results

This section shall be divided into the following paragraphs to describe the detailed results for each test. Note: The word "test" means a related collection of test cases.

## (Project-unique identifier of a test)

The following test results will be ordered by test: Red Circle Test, Random Algorithm Test, Happiness Algorithm Test, Collective Happiness Algorithm Test, and Customized Slider Test.

### Summary of test results

**Red Circle Test** – All results as expected. A third shape, a red circle, was implemented onto the polygon board with the same functionality as the other shapes. The red circle animates and interacts with the board environment as intended.

**Random Algorithm Test** – All results as expected. The radio buttons accurately default to the “Random” algorithm upon page loading. Likewise, the “Random” algorithm mimics random polygon movement when said polygon is unhappy. The “Random” algorithm was implemented and tested as intended.

**Happiness Algorithm Test** – Problems encountered, incomplete. The radio buttons accurately list a “Happiness” algorithm that is selectable. However, the “Happiness” algorithm follows the exact results from the “Random” algorithm, and does not accurately portray polygons moving to places where they are happy systematically. The “Happiness” algorithm results were not as intended and therefore not a success.

**Collective Happiness Algorithm Test** – Problems encountered, incomplete. The radio buttons accurately list a “Collective Happiness” algorithm that is selectable. However, the “Collective Happiness” algorithm follows the exact results from the “Random” algorithm, and does not accurately portray polygons moving to places by group systematically, in order to make the group (the collective neighborhood) happy. The “Collective Happiness” algorithm results were not as intended and therefore not a success.

**Customized Slider Test** – Not applicable, incomplete. The customized slider used to assess polygon bias between shapes (for happiness levels) is currently not implemented on the board interface. There is currently nothing to test which is unexpected, and the slider is still stuck in design. Thus, the results of this test are not available and therefore incomplete.

### Problems encountered

This paragraph shall be divided into subparagraphs that identify each test case in which one or more problems occurred.

#### (Project-unique identifier of a test case)

This paragraph shall identify by project- unique identifier a test case in which one or more problems occurred, and shall provide:

* + - * 1. A brief description of the problem(s) that occurred
        2. Identification of the test procedure step(s) in which they occurred
        3. Reference(s) to the associated problem/change report(s) and backup data, as applicable
        4. The number of times the procedure or step was repeated in attempting to correct the problem(s) and the outcome of each attempt
        5. Back-up points or test steps where tests were resumed for retesting

### Deviations from test cases/procedures

This paragraph shall be divided into subpara- graphs that identify each test case in which deviations from test case/test procedures occurred.

#### (Project-unique identifier of a test case)

This paragraph shall identify by project- unique identifier a test case in which one or more deviations occurred, and shall provide:

* + - * 1. A description of the deviation(s) (for example, test case run in which the deviation occurred and nature of the deviation, such as substitution of required equipment, procedural steps not followed, schedule deviations). (Red-lined test procedures may be used to show the deviations)
        2. The rationale for the deviation(s)
        3. An assessment of the deviations’ impact on the validity of the test case

# Test log

This section shall present, possibly in a figure or appendix, a chronological record of the test events covered by this report.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Date | Location | Description/Results | Hardware/software configurations | Performers |
| Integrating red circle | November 20, 2015  8 PM-10 PM | UMBC Library, on each team member’s laptop | Integrated a third shape, a red circle, to the sandbox. The red circle acted like the other shapes (e.g. moved if a certain amount of shapes were around it, and shook like the other shapes). | Latest version (as of November 2015) of Fire Fox, Internet Explorer, and Chrome on laptops with Windows 7 or Linux | Team SWIC |
| Adding the selected algorithm radio buttons | November 21, 2015  1 PM | Ian’s laptop | Added three radio buttons for each algorithm (random, happiness, collective happiness) | Latest version (as of November 2015) of Fire Fox, Internet Explorer, and Chrome on laptops with Windows 7 or Linux | Team SWIC |
| Working on Internet Explorer, Fire Fox, and Chrome | November 23, 2015  7 PM | All of Team SWIC’s laptops | Tested to see if the code ran on the browsers listed. | Latest version (as of November 2015) of Fire Fox, Internet Explorer, and Chrome on laptops with Windows 7 or Linux | Team SWIC |

# Notes

This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

1. Appendixes

Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

DESCRIPTION/PURPOSE

The Software Test Report (STR) is a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software system or subsystem, or other software-related item.

The STR enables the acquirer to assess the testing and its results.

APPLICATION/INTERRELATIONSHIP

Portions of this plan may be bound separately if this approach enhances their usability. Examples include plans for software configuration management and software quality assurance.

The Contract Data Requirements List (CDRL) should specify whether deliverable data are to be delivered on paper or electronic media; are to be in a given electronic form (such as ASCII, CALS, or compatible with a specified word processor or other support software); may be delivered in developer format rather than in the format specified herein; and may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document.

PREPARATION INSTRUCTIONS

General instructions.

a. Automated techniques. Use of automated techniques is encouraged. The term "document" in this means a collection of data regardless of its medium.

b. Alternate presentation styles. Diagrams, tables, matrices, and other presentation styles are acceptable substitutes for text when data required can be made more readable using these styles.

c. Title page or identifier. The document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem, or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of the preparing organization; and distribution statement. For data in a database or other alternative form, this information shall be included on external and internal labels or by equivalent identification methods.

d. Table of contents. The document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data in a database or other alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.

e. Page numbering/labeling. Each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data in a database or other alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.

f. Response to tailoring instructions. If a paragraph is tailored out of this document, the resulting document shall contain the corresponding paragraph number and title, followed by "This paragraph has been tailored out." For data in a database or other alternative form, this representation need occur only in the table of contents or equivalent.

g. Multiple paragraphs and subparagraphs. Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability.

h. Standard data descriptions. If a data description required by this document has been published in a standard data element dictionary specified in the contract, reference to an entry in that dictionary is preferred over including the description itself.

i. Substitution of existing documents. Commercial or other existing documents, including other project plans, may be substituted for all or part of the document if they contain the required data.