Software Test Plan (STP)

Software It Counts (SWIC)

CMSC 447

Updated on November 25, 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 Software test environment 3](#_Toc432634207)

[3.1 Name of test site(s) 3](#_Toc432634208)

[3.2 Software items 4](#_Toc432634209)

[3.3 Hardware and firmware items 4](#_Toc432634210)

[3.4 Other materials 4](#_Toc432634210)

[3.5 Proprietary nature, acquirer’s rights, and licensing 4](#_Toc432634210)

[3.6 Installation, testing, control 4](#_Toc432634210)

[3.7 Participating organizations 4](#_Toc432634210)

[3.8 Personnel 4](#_Toc432634210)

[3.9 Orientation plan 4](#_Toc432634210)

[3.10 Tests to be performed 4](#_Toc432634210)

[4 Test identification 5](#_Toc432634211)

[4.1 General information 5](#_Toc432634212)

[4.1.1 Test levels 5](#_Toc432634213)

[4.1.2 Test classes 5](#_Toc432634214)

[4.1.3 General test conditions 5](#_Toc432634215)

[4.1.4 Test progression 5](#_Toc432634215)

[4.1.5 Data recording, reduction, and analysis 5](#_Toc432634215)

[4.2 Planned tests 6](#_Toc432634212)

[4.2.1 Items to be tested 6](#_Toc432634213)

[4.2.1.1 Project-unique identifier of a test 6](#_Toc432634213)

[5 Test schedules 7](#_Toc432634216)

[6 Requirements traceability 8](#_Toc432634216)

[7 Notes 9](#_Toc432634217)

[A. Appendixes 9](#_Toc432634218)

# Scope

## 1.1 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

## 1.2 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 plan used to accomplish the requirements above, as given by the customer Russ Cain. The subsequent test plan for the project will thoroughly detail how the testing is to be carried out by the group (SWIC). This document will (like all documents) 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>.

## 1.3 Document overview

This document will specifically layout a plan for testing the system and its requirements that the group needs to fulfil for the project (as previously discussed by Professor Cain). The test plan will address the testing environment in detail and the items necessary to properly carry out the testing. It will also discuss the general test conditions and the overall test progression. The test procedure and handling shall be carried out via the test schedule by only group members, and the results of such shall be published and assessed by our customer, Russ Cain.

# 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

# Software Test Environment

## 3.1 Name of test site(s)

The software testing site shall be UMBC’s campus, its library and Fine Arts Hall Room 215 being the specific locations.

## 3.2 Software items

All of Team SWIC will bring his or her laptop to conduct testing on individual ends; the operating systems are either Windows 7 and onwards, or Linux. Each laptop which, regardless of operating system, has either the latest (as of November 2015) version of Fire Fox, Internet Explorer, Chrome, or any combination of the aforementioned browsers. All laptops will also have access to GitHub, where the code is stored.

## 3.3 Hardware and firmware items

No other outside devices, such as Ethernet cables, are necessary.

## 3.4 Other materials

The Parable of The Polygons web site comes with its own manual/instructions and does not require additional reading to aid in understanding. Finally, neither The Parable of The Polygons nor related materials raise a security/privacy concerns due to project nature.

## 3.5 Proprietary nature, acquirer’s rights, and licensing

The code will not need any licensing to use—it is readily available to view and use on GitHub. The code will also not gain any monetary profit.

## 3.6 Installation, testing, control

In order to set up the software testing environment, all laptops must have one or more of the latest (as of November 2015) versions of Fire Fox, Internet Explorer, and Chrome. All laptops used to code the software must also have the desktop or shell version of GitHub, as well as be a contributor to The Parable of The Polygons repository. All of these programs can be found on their official web sites. In order to “test”/prepare the software components, all that will need to be done is make sure Internet access is possible, and that laptop’s GitHub can access, push, and sync to the correct repository. For maintenance, each laptop must update its browsers and/or GitHub if updates to either software are released.

## 3.7 Participating organizations

No other facility/team outside of Team SWIC will partake in the testing or preparation thereof.

## 3.8 Personnel

At least one person should test code, but potentially as many people as available can help to test the code. (However, the sandbox does not allow for multiple computers to see what one computer has for input.) The person must have the capability to read at the level of at least a fourth grader, perform simple fractions (e.g. 1/8, 2/5), and understand concepts such as bias, discrimination, and segregation. No other special skills will be needed. Between Team SWIC, all team members must be present from September 14th, 2015 to December 7th, 2015 to test code. Otherwise personnel from the team or otherwise will not be necessary.

## 3.9 Orientation plan

The Parable of The Polygons already provides a brief overview/training of how to use it. No outside sources will be needed to aid further understanding in the code, nor will a specialized team be needed to assist in understanding the code.

## 3.10 Tests to be performed

The tests to be performed are as follows: a red circle behaving like the yellow triangle and blue square (shakes, moves based on neighbor preference); three radio buttons which allow the user to select a desired algorithm; a triangularly-shaped slider which permits the user to change the relationships between the red circle, yellow triangle, and blue square; a second algorithm calculating how happy a polygon feels around its neighbors and moving to a spot where it would be happier; a third algorithm which is similar to algorithm 2, but also considers how its neighbors feel about it; and that the code works on the latest versions of Fire Fox, Internet Explorer, and Chrome.

# Test identification

## 4.1 General information

### Test levels

### Test classes

### General test conditions

### Test progression

### Data recording, reduction, and analysis

## Planned tests

### Items to be tested

#### Project-unique identifier of a test

# Test schedules

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Date | Location | Description | Hardware/software configurations | Actors |
| Red circle test (Observat-ional) | November 20, 2015  8 PM-10 PM | UMBC Library, on each team member’s laptop | Test the red circle to see if it actslike 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 |
| Selected algorithm radio buttons  (Observat-ional) | November 21, 2015  By 11:59 PM | Ian’s laptop, dorm | Test the three, added 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 (Ian) |
| Random Algorithm Test | November 21, 2015  By 11:59 PM | All of Team SWIC’s laptops | Test the Random Algorithm via the according radio button and make sure it functions as it did previously. | Latest version (as of November 2015) of Fire Fox, Internet Explorer, and Chrome on laptops with Windows 7 or Linux | Team SWIC |
| Happiness Test | November 22, 2015  By 11:59 PM | Chris’s laptop, dorm | Test the Happiness Algorithm and see if the polygons move to spot where they’re happy. | Latest version (as of November 2015) of Fire Fox, Internet Explorer, and Chrome on laptops with Windows 7 or Linux | Team SWIC |
| Collective Happiness Test | November 22, 2015  By 11:59 PM | All of Team SWIC’s laptops | Test the Collective Happiness Algorithm and assess whether it works as intended. | Latest version (as of November 2015) of Fire Fox, Internet Explorer, and Chrome on laptops with Windows 7 or Linux | Team SWIC |
| Internet Explorer, Fire Fox, and Chrome  Test | 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 |
| Work on Software Test Results (STR) | November 25,  2015  5 PM - 7 PM | All of Team SWIC’s laptops | Assign parts of STR and finish any necessary testing | Latest version (as of November 2015) of Fire Fox, Internet Explorer, and Chrome on laptops with Windows 7 or Linux | Team SWIC |
| 3-sided Slider Test | November 2015 | ?? | Test the slider on the interface and determine if polygon biases are accurately portrayed by ratio. | Latest version (as of November 2015) of Fire Fox, Internet Explorer, and Chrome on laptops with Windows 7 or Linux | Team SWIC |

# Requirements traceability

*Red Circle Software Test Description*:

The test cases herein acknowledge the requirements which state, “the system shall include an additional shape, to a total of three shapes,” “the third shape added to the program shall be a red circle,” and, “the bias of the new shape shall be represented with a triangular ui component, as illustrated in the SDP.” The resulting software will be tested by unit, and eventually integrated in the automatic segregator component of Parable of Polygons.

*Algorithm Software Test Description*:

The test cases herein acknowledge the requirements which state, “the system shall include a sorting algorithm which is more efficient than the original,” and “the system will provide radio buttons to assign the applied sorting algorithm.” The resulting software will be unit tested, and integrated with the other system requirements, in the automatic segregator component of Parable of Polygons.

# 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.).