Software Requirements Specification (SRS)

Software It Counts (SWIC)

CMSC 447

Updated on November 20, 2015

Table of Contents

Scope

This section shall be divided into the following paragraphs.

Identification

For this project, we will be working with the web application, Parable of Polygons. This software simulates segregation levels between two different shapes, allowing them to move to different locations on the board in an attempt to be happy. The website itself is programmed in html while the application is programmed in javascript

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. This will be accomplished with 3 sliders formed into a triangle. Each slider will determine the relationship between 2 out of 3 shapes, allowing all of the relations to be shown on the edges.

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.”

Document overview

This document will describe the requirements we must fulfil for the project as we discussed them with Dr. Cain. It will also briefly discuss how these requirements will be fulfilled

Referenced documents

This document will reference the following repos for the parable of polygons
https://github.com/ncase/polygons

https://github.com/dncnmcdougall/polygons

Requirements

This section shall be divided into the following paragraphs to specify the CSCI requirements, that is, those characteristics of the CSCI that are conditions for its acceptance. CSCI requirements are software requirements generated to satisfy the system requirements allocated to this CSCI. Each requirement shall be assigned a project-unique identifier to support testing and traceability and shall be stated in such a way that an objective test can be defined for it. Each requirement shall be annotated with associated qualification method(s) (see section 4) and traceability to system (or subsystem, if applicable) requirements (see section 5.a) if not provided in those sections. The degree of detail to be provided shall be guided by the following rule: Include those characteristics of the CSCI that are conditions for CSCI acceptance;

defer to design descriptions those characteristics that the acquirer is willing to leave up to the developer. If there are no requirements in a given paragraph, the paragraph shall so state. If a given requirement fits into more than one paragraph, it may be stated once and referenced from the other paragraphs.

Required states and modes

There are 2 states for this project, idle, and active. During the idle state the board state will be able to be micromanaged, whether by moving individual polygons or adjusting the ratios of said polygons. During the active state, the program will move the polygons around attempting to sort them out based on the chosen algorithm, such that every polygon is happy

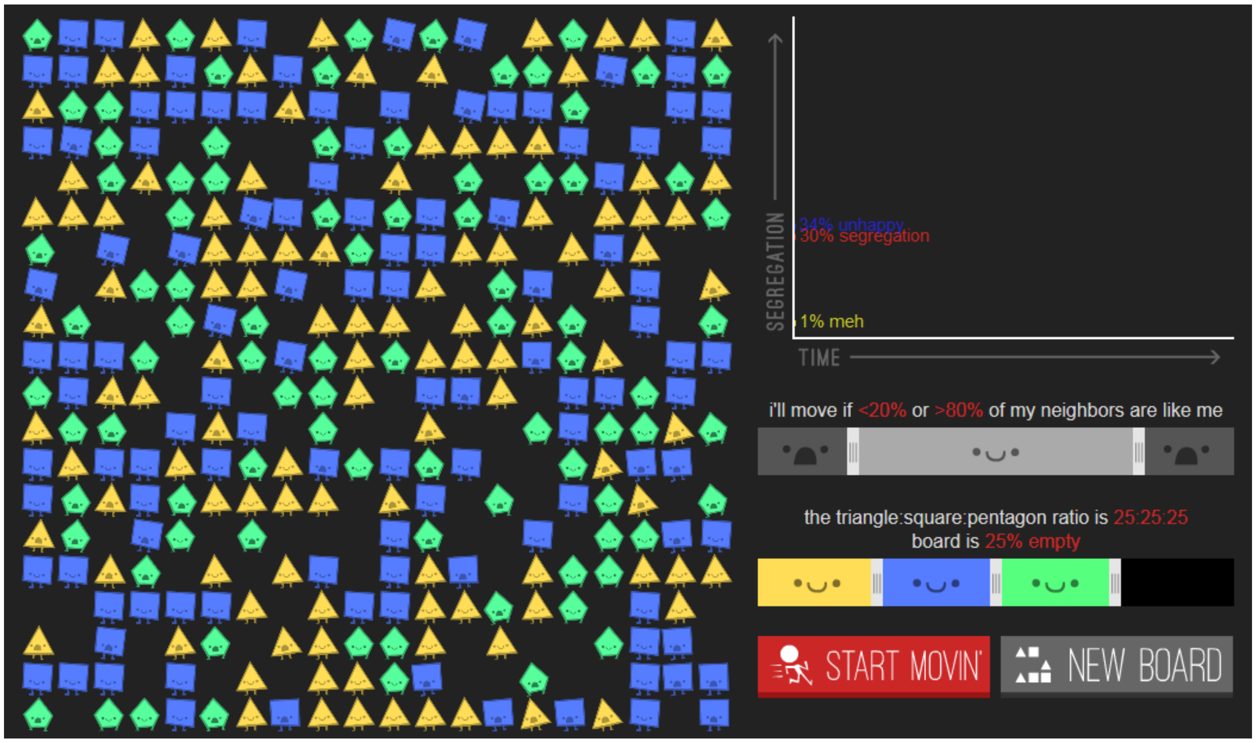
CSCI capability requirements

- There must be a third polygon
- The polygon must be a circle
- The circle must be red
- The circle must behave in the same manner as the triangle and square
- The ratio slider must be modified to add in the red circle
- There must be 3 relationship sliders
- Each slider must show the relationships between 2 types of polygons
- These sliders must form a triangle
- There must be 2 new placement algorithms added
- One of these must place a polygon based on its own happiness
- The other must place a polygon based on its happiness, as well as the happiness of its potential new neighbors
- There must be 3 radio buttons
- These 3 radio buttons will allow the user to choose which algorithm to use

CSCI external interface requirements

Interface identification and diagrams

The project shall contain the following interfaces: a polygon simulation board with the 3 required shapes, the required radio buttons (3) to specify which algorithm to implement on the board, and a 3-sided slider to accommodate each individual shapes' biases. The redesigned polygon board should be almost identical to the existing board but with a third shape (red circle) included. The board will be a configuration item within the interface it is placed in, and be based on Nicky Case and Vi Hart's original "Parable of the Polygons" game board design. Here is a sample screenshot taken from Duncan McDougall's pentagon remix of "Parable of the Polygons" of what our intended board should look like, except instead of green pentagons, it will include/be adjusted for red circles:


SEGREGATION
TIME
34% unhappy
30% segregation
1% meh
i'll move if <20% or >80% of my neighbors are like me
the triangle:square:pentagon ratio is 25:25:25
board is 25% empty
START MOVIN'
NEW BOARD

(Project-unique identifier of interface)

a. Priority that the CSCI must assign the interface
   1) High priority.

b. Requirements on the type of interface (such as real-time data transfer, storage-and-retrieval of data, etc.) to be implemented
   1) Real-time display update

c. Required characteristics of individual data elements that the CSCI must provide, store, send, access, receive, etc., such as:
   1) Names/identifiers
      I. Project-unique identifier
         • CMSC-447-5
      II. Non-technical (natural-language) name
         • The Parable of The Polygons
      III. DoD standard data element name
         • Not applicable as this project is not being made under military standards.
      IV. Technical name (e.g., variable or field name in code or database)
         • Not applicable.
      V. Abbreviation or synonymous names
         • TPP (abbreviation)
   1) Data type (alphanumeric, integer, etc.)
      a) The Parable of Polygons will use given integers from the user slide input.
   2) Size and format (such as length and punctuation of a character string)

a)Not applicable as the code will use the default on the site's font and board size.

3) Units of measurement (such as meters, dollars, nanoseconds)

a)It will consider (un)happiness and the algorithm will be calculated in seconds (even though the finished board may take any unpredictable amount of seconds/minutes.)

4) Range or enumeration of possible values (such as 0-99)

a)The segregation percentages will range from 0 to 100. Each shape will have a bias percentage towards each shape.

5) Accuracy (how correct) and precision (number of significant digits)

a)Accuracy will be 100% in the sense that each polygon will move once they feel unhappy in a spot. Precision will technically be one significant digit; however, the code does not extensively consider digits regardless.

6) Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the data element may be updated and whether business rules apply

a)Actual processing time may depend on the user's browser.

7) Security and privacy constraints

a)Not applicable as The Parable of The Polygons does not deal with user privacy nor security concerns.

8) Sources (setting/sending entities) and recipients (using/receiving entities)

a)Not directly applicable, as The Parable of The Polygons is not directly sending anything to the user (other than output produced on the screen).

d. Required characteristics of data element assemblies (records, messages, files, arrays, displays, reports, etc.) that the CSCI must provide, store, send, access, receive, etc., such as:

1) Names/identifiers

I.Project-unique identifier

•Yellow triangle, red circle, blue square

II.Non-technical (natural language) name

•N/A

III.Technical name (e.g., record or data structure name in code or database)

•N/A

IV.Abbreviations or synonymous names

•The Polygons (general name)

2) Data elements in the assembly and their structure (number, order, grouping)

3) Medium (such as disk) and structure of data elements/assemblies on the medium

a. Not applicable as the software is entirely online, on GitHub. No physical medium is needed.

4) Visual and auditory characteristics of displays and other outputs (such as colors, layouts, fonts, icons and other display elements, beeps, lights)

a. The Parable of The Polygons will produce no sounds. The shapes must be yellow triangles, red circles, and blue squares. The layout will be what is already in the source code (black grid with invisible lines, black background in general, etc). All 3 shapes will have 3 basic faces: happy, neutral, and unhappy.

5) Relationships among assemblies, such as sorting/access characteristics
   a. Other than that the shapes sort themselves based on a given algorithm, not applicable because there will not be multiple CSCIs.
6) Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the assembly may be updated and whether business rules apply
   a. There must be the same amount of polygons; in other words, the user cannot add nor take away a certain polygon. The algorithms working correctly will be of the upmost importance.
7) Security and privacy constraints
   a. Not applicable as the software does not directly deal with sensitive information.
8) Sources (setting/sending entities) and recipients (using/receiving entities)
   a. The program, aside from user input, does not really interact with other entities such as databases or network connections.

e. Required characteristics of communication methods that the CSCI must use for the interface, such as:
   1) Project-unique identifier(s)
      a)Not applicable for the scope of this project.
   2) Communication links/bands/frequencies/media and their characteristics
      a)Technically anything that the user's computer is using to connect to the Internet; but otherwise, not applicable.
   3) Message formatting
      a)None as The Parable of the Polygons is not sending messages.
   4) Flow control (such as sequence numbering and buffer allocation)
      a)None as the software does not require dynamic memory nor flow control.
   5) Data transfer rate, whether periodic/aperiodic, and interval between transfers
      a)The "data", if defined as the results calculated for each polygon, will be delivered in real time as it is being calculated (so constantly until the board has reached 100% happiness). Otherwise there is nothing specific, such as "100 bits per second."
   6) Routing, addressing, and naming conventions
      a)None as The Parable of The Polygons will not have a specific/unique network nor routing table.
   7) Transmission services, including priority and grade
      a)None as The Parable of The Polygons will not be transmitting things of that nature.
   8) Safety/security/privacy considerations, such as encryption, user authentication, compartmentalization, and auditing
      a)The Parable of The Polygons can use whatever encryption/security the browser uses, but it will not have its own encryption so to say.

f. Required characteristics of protocols the CSCI must use for the interface, such as:
   1) Project-unique identifier(s)
   2) Priority/layer of the protocol
   3) Packeting, including fragmentation and reassembly, routing, and addressing

a)The software will not use/provide anything special in particular (i.e., its own) and will use whatever is being used by default in the system.

4) Legality checks, error control, and recovery procedures

a)The software will not use/provide anything special in particular (i.e., its own) and will use whatever is being used by default in the system.

5) Synchronization, including connection establishment, maintenance, termination

a)The Parable of The Polygons will use whatever synchronization methods the current browser is using.

6) Status, identification, and any other reporting features

a)Not applicable as The Parable of The Polygons will not allow one to send user reports about statuses.

g. Other required characteristics, such as physical compatibility of the interfacing entities (dimensions, tolerances, loads, plug compatibility, etc.), voltages, etc

1) None directly applicable.

CSCI internal interface requirements

The internal interfaces within the CSCI must include radio buttons, a redesigned polygon board, and a redesigned slider to adjust individual shapes' biases. The new designed interfaces within the CSCI must incorporate radio buttons (3) for each algorithm used to run the simulated polygon optimizations; the radio buttons should be implemented next to the board where the algorithms are implemented. Likewise, the interfaces must facilitate a new polygon board that captures a new, third shape (red circle) to be analogous to the previous polygons (yellow triangle and blue square); the board should be almost identical to the previous board in shape/size. Finally, the internal interfaces of the CSCI must accompany a custom-made slider that helps to adjust the individual shapes' biases respective to all three shapes (the interface of the slider should be left to the design)

CSCI internal data requirements

All the data internal to the CSCI should be left to the design. However, most of the data used to initialize the polygon board, sliders, etc. should be based primarily off the data files/databases already setup by the previous (original) version of the "Parable of the Polygons."

Adaptation requirements

There are no requirements, or designs for the CSCI, entailing any installation-dependent data or operational parameters.

Safety requirements

There are no CSCI requirements concerned with safeguarding property, personnel, or flaws (possibly exploited within the CSCI).

Security and privacy requirements

Each group member will generate an SSH key which will be linked to ther git account. Our repository will be configured to only accept the SSH keys from members of the group.

Activity logging will be recorded on git as new commits and pushes are demanded.

All code will be pushed from the development branch to the testing branch where quality control will ensure that it passes unit and integration tests which will be set up with the help of Google Test. When the code on the testing branch passes, it will be pushed to the master branch where users may build the program from the repository.

GitHub's existing authentication measures using SSH keys will ensure that only members of SWIC can modify the code.

Our additions to the code data will be the red circle image file and sprites, and the times of the new algorithm. These times will contain the following attributes:

- Date
- Algorithm used w/ version number
- % bias and anti-bias applied
- Time to sort until completion in milliseconds

These will be in the form of an Excel spreadsheet, or similar freeware file type such as that of Libre Office. The analytics program will perform basic descriptive statistics such as median, mean, and range, highlighting the differences between the algorithms and quantifying them. Experimental groups and results from different versions of the algorithm will be stored in separate spreadsheet pages, located on tabs at the bottom of the program.

CSCI environment requirements

The automatic.js object will be able to run on an HTML webpage with minimal modification. Our code is required to pertain to the Open-Closed Policy of software engineering, which states that modules should be open for expansion and closed for modification.

Personnel-related requirements

Due to its simple nature, The Parable of The Polygons' CSCI shall not necessarily possess any requirements such as full-blown, extensive training programs. The user in general should be able to understand (at least) fourth-grade-level English, and also be aware of what the words segregation, diversity, and bias mean in a sociological context.

In terms of error messages, The Parable of The Polygon's scripting error message must be modeled after current browser standards. Criteria for the error messages will be that it produces no sounds, and specifies what went wrong in the code if applicable. Any browser-specific errors (such as a script running too slowly due to browser limits) will be provided by the browser itself, and it is up to the user what to do in response (e.g. stop script, continue script, debug script, etc).

Training-related requirements

As mentioned previously, the provided code already comes with a brief tutorial of how the software operates. This rendition shall keep the provided-tutorials in place to ensure the user knows how to do things such as drag the shapes and understand the software's intention, as well as avoid redundancy or unnecessary redundant additions. Anything beyond that we deem necessary will be provided in text format before the appropriate sandbox. As is the website, the accompanying new text should be written on a level that someone who at least has a fourth grade level understanding of English can understand.

Logistics-related requirements

Because the code is hosted on GitHub, any and all system maintenance relies heavily on GitHub itself. For example, if GitHub is down for maintenance, then the Parable of The Polygons shall consequently also be down. Before the project due date (December 4th 2015), a pseudo system maintenance will be performed at least every other week, where the team analyzes and possibly fixes code.

The team will provide technical support for the project, with Brooke Washington being the primary point of contact for questions or discovered issues in the code.

Other requirements

This paragraph shall specify additional CSCI requirements, if any, not covered in the previous paragraphs.

Packaging requirements

The GitHub repository will technically 'deliver' the code, in the sense of storing it and allowing one to access it at any time in any place. All of the files will be appropriately named as to ensure easy communication, and the GitHub will be available to all (i.e., not private, nor encrypted, and so on).

Precedence and criticality of requirements

The code does not particularly deal with anything especially sensitive nor is at risk of security breaches; each part will have equal importance in terms of safety, security, and privacy. (While the new algorithm working will be the most important aspect in terms of getting it to work, it does not pose any more or less of a threat on safety/security/privacy as do the sliders.

Qualification provisions

a. The Red Circle shall be demonstrated to be functional within the system
b. The Happiness Algorithm shall be demonstrated to be functioning within the system
c. The Collective Happiness Algorithm shall be demonstrated to be functioning within the system
d. The Sliders shall be demonstrated to be functioning within the system

Requirements traceability

a. The Red Circle will be traced by its implementation cycle as it goes from planning to design to construction to testing.
b. The Happiness and Collective Happiness Algorithms will first be implemented in a test branch, then later committed to the master branch as they are finished.