Software Design Description (SDD)

Software It Counts (SWIC)

CMSC 447

Scope

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 process we will follow in order to create the requested modifications. It will also detail exactly how we will go about following this process to maximize productivity and create the best product possible

Referenced documents

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

https://github.com/dncnmcdougall/polygons

CSCI-wide design decisions

a. Design decisions regarding inputs the CSCI will accept and outputs it will produce, including interfaces with other systems, HWCIs, CSCIs, and users (4.3.x of this DID identifies topics to be considered in this description). If part or all of this information is given in Interface Design Descriptions (IDDs), they may be referenced.
   - The CSCI will accept input, which shall describe the relationships between the various polygons and the desired sorting algorithm, from the user from either radio buttons (algorithms) or sliders (relationships among the polygons). The produced output shall depend entirely on what the user has

given as input; but speaking broadly, the CSCI will show the user the results of the algorithm and polygon relations combination. The CSCI will not take any input from any other interface, other CSCI, or HWCI (although the speed of the output user's computer's processing power), for the code is more dependent on what one individual user has to say.

b. Design decisions on CSCI behavior in response to each input or condition, including actions the CSCI will perform, response times and other performance characteristics, description of physical systems modeled, selected equations/algorithms/rules, and handling of unallowed inputs or conditions.
    - The first algorithm will be the one already provided by Nicky Case. The second algorithm, Collective Happiness, will make polygons move to different places based on the amount of polygons surrounding them, as well as the relationship ratios the user has indicated. The calculations will be performed as soon as the user presses the enter button; however, the actual board itself will not 'stop' until all of the polygons are happy where they are.
    - The sliders will not allow for any unpermitted inputs such as negative happiness relationships or relationships between nonexistent polygons (e.g. a star, diamond, etc). If for whatever reason an error occurs during calculation, an error message will appear, stating what ill condition caused the error.

c. Design decisions on how databases/data files will appear to the user (4.3.x of this DID identifies topics to be considered in this description). If part or all of this information is given in Database Design Descriptions (DBDDs), they may be referenced.
    - Not applicable since the program is not storing any data from the user.

d. Selected approach to meeting safety, security, and privacy requirements.
    - Not applicable since The Parable of The Polygons does not use/collect any sensitive user data.

e. Other CSCI-wide design decisions made in response to requirements, such as selected approach to providing required flexibility, availability, and maintainability.
    - The code shall be available to all via GitHub, and the code will work on FireFox, Internet Explorer, and Chrome.

CSCI architectural design

This section shall be divided into the following paragraphs to describe the CSCI architectural design. If part or all of the design depends upon system states or modes, this dependency shall be indicated. If design information falls into more than one paragraph, it may be presented once and referenced from the other paragraphs. Design conventions needed to understand the design shall be presented or referenced.

CSCI components

a. Software Units that make up the CSCI
    a. The Graph
    b. The Sliders
    c. The Polygon Grid

    - d. The Radio Buttons
- b. Relationship between units
    - a. The sliders dictate the ratio of every polygon on the grid and how they will sort themselves when the simulation is started.
- c. Purpose of each software unit
    - a. The Graph
        - i. Display segregation versus time for the simulation
    - b. The Sliders
        - i. Allow the user to select different ratios for each polygon
        - ii. Allow the user to select how the polygons will segregate themselves
    - c. The Polygon Grid
        - i. Display the results and the segregation in real time
    - d. The Radio Buttons
        - i. Allow the user to select which algorithm to use
- d. Software Unit Development Status
    - a. The Graph
        - i. Base implementation exists, needs to be updated to support new polygon
    - b. The Sliders
        - i. Base implementation exists, needs to be updated to support new polygon
        - ii. The segregation slider will be implemented from scatch
    - c. The Polygon Grid
        - i. Base implementation exists, needs to be updated to support new polygon
    - d. The Radio Buttons
        - i. No base code exists.

Concept of execution

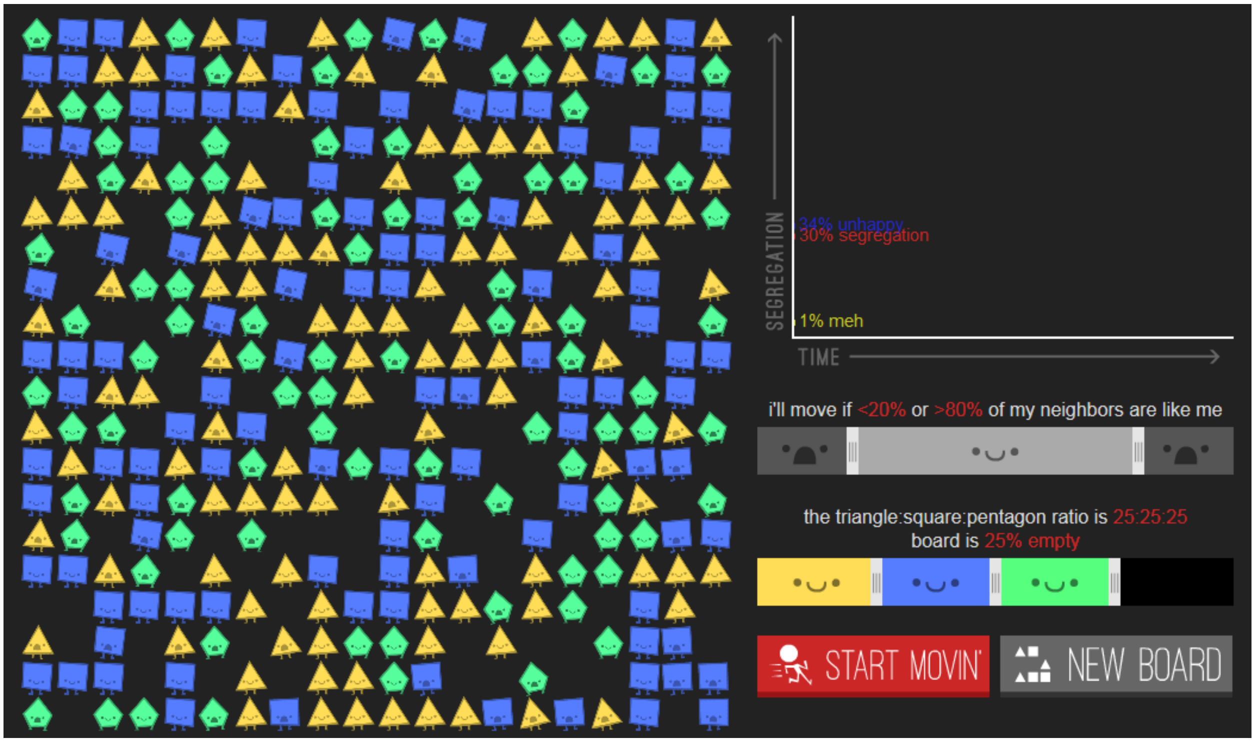
The user will load the webpage and adjust the sliders depending on what simulation will be run. They will be able to adjust how many of each polygon is on the grid and to what degree they will segregate themselves. After the sliders are adjusted the user can then start the simulation when the polygons will begin sorting themselves in real-time while the graph updates.

Interface design

The interface provided by the project shall be mostly based off Nicky Case and Vi Hart’s existing “Parable of the Polygons” platform. As such, similar data elements and structures will be developed as our design builds on what was already available. The subsequent interface/interfacing entities are devised from requirements and intended design plans.

Interface identification and diagrams

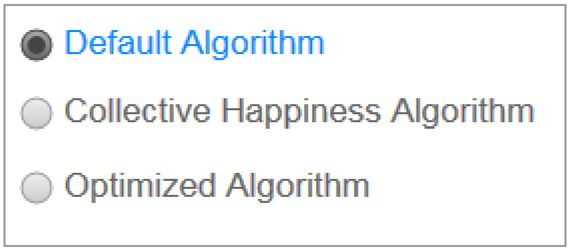
The project shall contain the following interfaces: a polygon simulation board with the 3 required shapes, the required radio buttons (3) to specify which algorithm to implement on the board, and a 3-sided slider to accommodate each individual shapes' biases. The redesigned polygon board should be almost 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 the 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:



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

The custom made radio buttons shall be a configuration item of simple buttons placed near the simulated polygon board interface. There shall be 3 radio buttons each matching an according algorithm as previously outlined within the requirements: default, collective happiness, and optimized algorithms. Note: only one radio button shall be selected at a given time, meaning only one algorithm may be implemented per simulation. The following design shall roughly depict the interface item holding the suggested radio buttons:

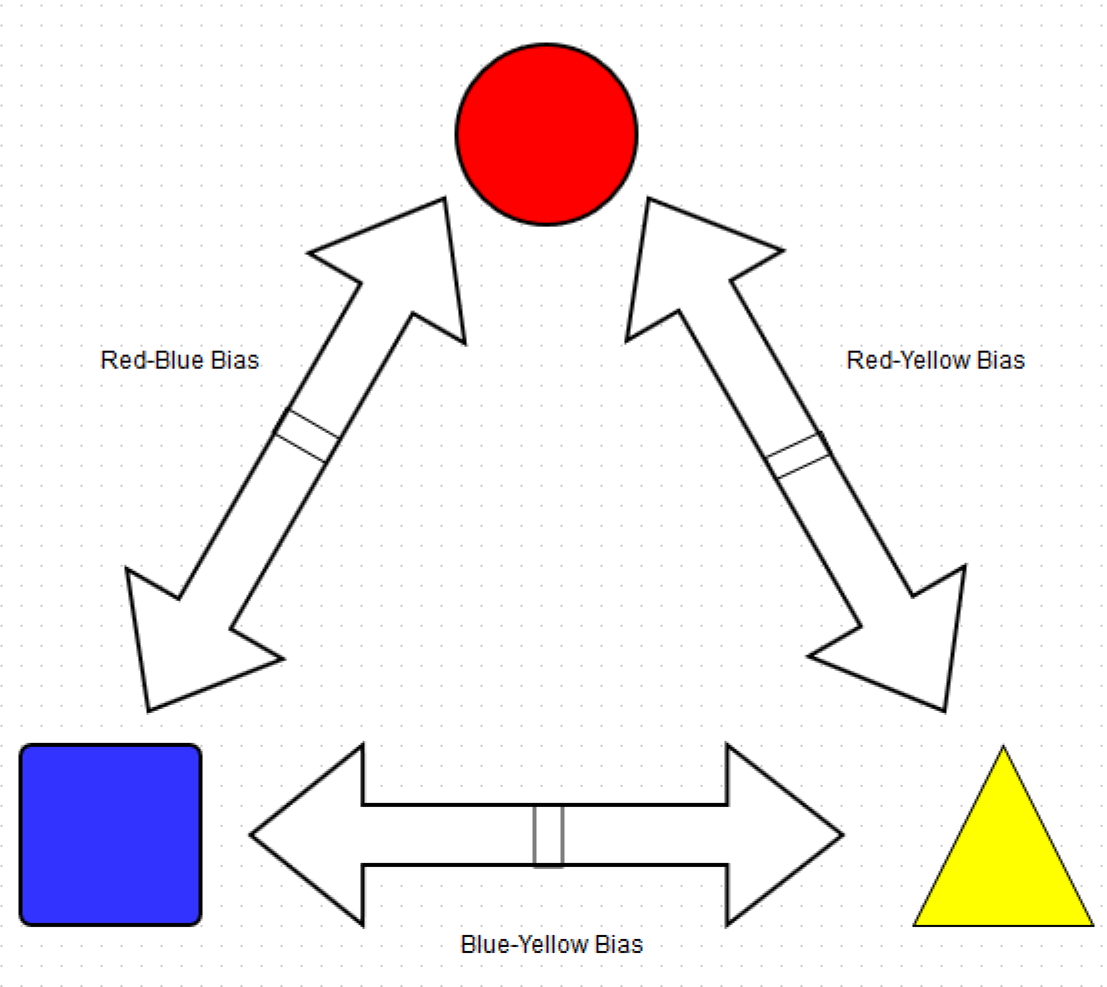


● Default Algorithm

○ Collective Happiness Algorithm

○ Optimized Algorithm

The custom made slider shall be comprised of 3 separate bias slides called Red-Blue Bias, Red-Yellow Bias, and Blue-Yellow Bias. Each of these slides will be configurable upon mouse click-slide (similar to the slides existing in the original “Parable of the Polygons”), and shall be a configuration item in an interface near the simulated polygon board. The custom 3-sided slider will look similar to the design below:



(Project-unique identifier of interface)

For the polygon simulated board, the user should be to click and drag any of the shapes within it to any available sections of the board. Likewise, the polygons within the board should respond to its according surroundings based on their respective shape biases. All functioning and mechanics should operate almost exactly how the original “Parable of the Polygons” game board does. Thus, the board interface must implement real-time data transfer of the appropriate polygon data elements: circles, triangles, squares.

For the given radio buttons assigned to the three algorithms as stated above, the user should be able to select only one radio button (algorithm) at a time while performing a simulation on the polygon board. The “default” algorithm shall be used by default when the interface is first created and thereafter changed by the user according to their preference. The interface using the radio buttons must use storage-and-retrieval of data when assigning the appropriate algorithm for a given simulation.

For the designed slider to be implemented with 3-way shape bias configuration, the user should be able to click-move any/all of the 3 different bias measurements shown in the above slider diagram from 4.3.1. By default, the interface of the slider shall initialize all biases to 50% and thereafter change accordingly as the user manipulates such ratios. The interface of the custom slider will use real-time data transfer to affect the different shapes’ bias and affect happiness within the board state. This, the slider must also interact dynamically with the same interface that utilizes the simulated polygon board.

CSCI detailed design

This section shall be divided into the following paragraphs to describe each software unit of the CSCI. If part of all of the design depends upon system states or modes, this dependency shall be indicated. If design information falls into more than one paragraph, it may be presented once and referenced from the other paragraphs. Design conventions needed to understand the design shall be presented or referenced. Interface characteristics of software units may be described here, in Section 4, or in Interface Design Descriptions (IDDs). Software units that are databases, or that are used to access or manipulate databases, may be described here or in Database Design Descriptions (DBDDs).

(Project-unique identifier of a software unit, or designator of a group of software units)

A. The algorithms used will be selectable by the user between the Random, Happiness, and Collective Happiness algorithm with the Random algorithm selected by default.
B. The system must be fully functional on Firefox, Chrome, and Internet Explorer.
C. The languages to be used for the system are HTML, CSS, and Javascript.
D. The system will have radio buttons that will allow the user to select which sorting algorithm to use. The system will also feature sliders that will allow the user to select the conditions for happiness for each polygon.
E. The system will update the graph and polygon grid as its output.

F. The system will start with control being given to the option panel and when the user presses the “Start Movin’” button, control will be transferred to the polygon grid as it sorts itself.

G. Errors and exceptions are handled by only allowing input we accept to come in so it is unlikely to receive unexpected input.

Requirements traceability

a. Red Circle tracked via the HTML page hosted on github

b. Random Algorithm tracked via the HTML page hosted on github

c. Collective Happiness tracked via the HTML page hosted on github

d. Sliders tracked via the HTML page hosted on github