The Memo

Jesse Chick   Benjamin Martin   Keenan Johnson

Nickoli Londura   Jiaji Sun

February 5, 2018

Contents

1 Contributors and ONIDs

LINK TO THE GITHUB REPO AT THE CORRECT BRANCH: https://github.com/Tonyenike/CS361-001-W2018/tree/martinb3-assignment-3/projects/martinb3/assignment-3

- Jesse Chick ∼ chickj

- Benjamin Martin ∼ martinb3

- Keenan Johnson ∼ johnsoke

- Nickoli Londura ∼ londuran

- Jiaji Sun ∼ sunji

2   User Interface Prototypes

Each page is divided into three sections regardless of where the user is on the website. On the top of the page highlighted in black is the title bar. The title bar will contain the title of the site "Rubik's Cube Practice Tool", and if clicked on, will link the user back to the home page of the website. The highlighted red region on the left is the menu bar. The menu bar will allow the user to navigate throughout the website. There are three buttons on the menu: one that will link to the home page (the Tutorials Page), one that leads to the Rubik's Cube practice page, and one that leads to the Letter memorization page. The third highlighted region, which is in blue, is the canvas. The canvas serves the content specific to each page (tutorial/Rubik's Cube practice page/letter memorization page). Since the canvas serves the content unique to each page, we will focus specifically about discussing what the content on the canvas does for the user.

2.1 Tutorials Page
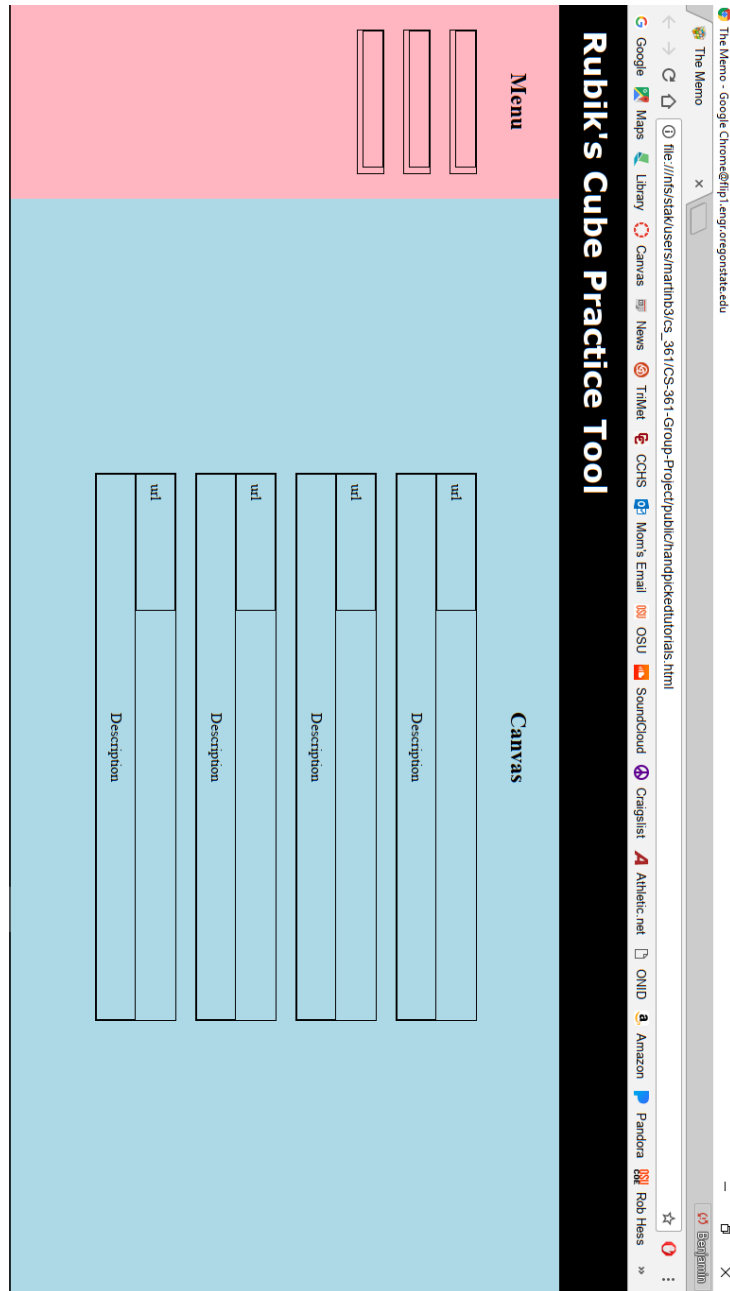
Instructions: On the canvas region of the tutorials page, there will be hotlinks to direct the user to tutorials to learn about how to solve the cube blindfolded.

Since any new visitor that comes to this website may not be familiar with how to solve a Rubik's Cube blindfolded, this page is set as the HOME PAGE. This will be the first thing a user sees when they enter our website. If they are familiar with how to solve a Rubik's Cube blindfolded, they will proceed by clicking on a menu button on the left to navigate to other pages to practice their blindfolded skills. If they are not familiar, then they will click on the links to learn about how to solve a Rubik's Cube blindfolded. The choice is theirs to make on judging the sufficiency of their knowledge.

2.2   Rubik's Cube practice page
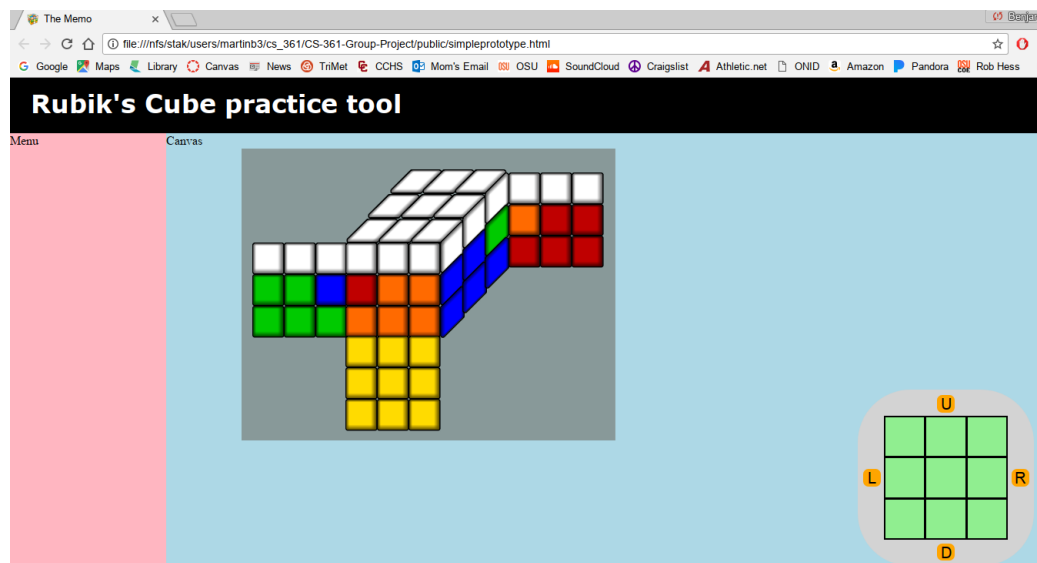
Instructions: This page will have a Rubik's Cube display (shaded in a dark grey background), and a "viewtoggle" display (the display in the bottom right hand corner). The viewtoggle will have four orthognal buttons to allow the user to rotate the face of the cube to see other sides. The Rubik's Cube display will also change accordingly to the button pressed on the viewtoggle. If, for example, the user were to press the "U" button, the cube display would rotate up, shifting all of the squares on each face down by one face. The squares colored on the viewtoggle would change as appropriate as well. The behavior of the changing colors of the cube would be determined by the cube object which will be described in detail in our diagrams.

2.3 Letter memorization page

Instructions: The letter memorization page is simpler. There is a text box area where the user will input their string of letters. The user will then press the

"Generate words" button, where the algorithm will then generate a random series of words from the letters given and display them on the canvas below the text input. The user can then repeat as many times as they feel like. If numbers or symbols are input there will be an alert from the webpage saying "Error" input is not valid!" and will NOT generate a string of words.
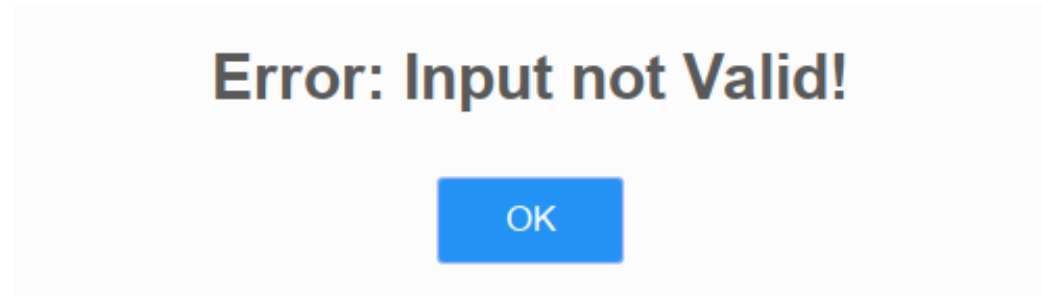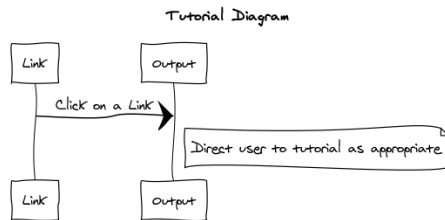

3 Class Diagrams
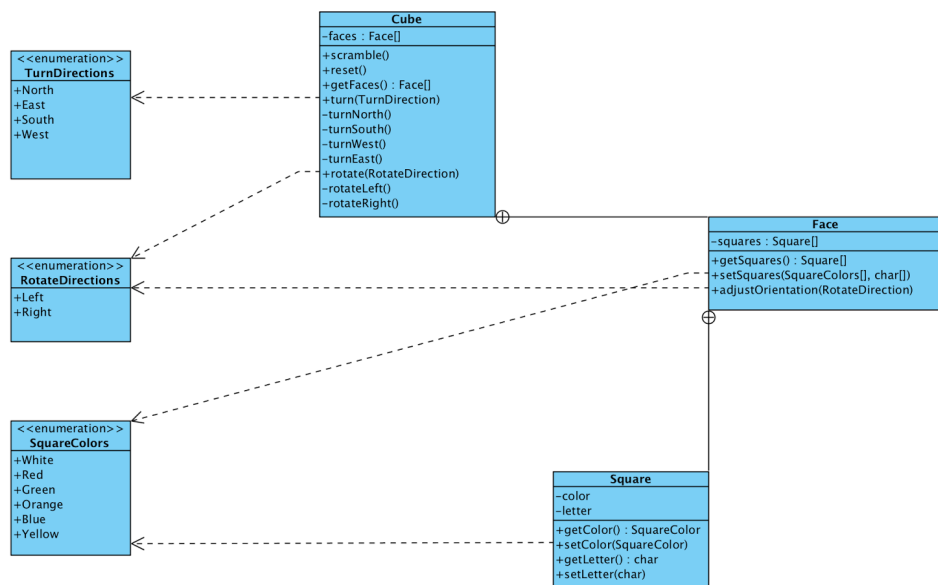
Notes:

1. In Visual Paradigm, the application we used to generate the diagram, composition is denoted by a plus sign in a circle, not a black diamond.

2. Given that we only have composition relationships represented (where the proportions of objects is implicit), and Visual Paradigm makes it nigh impossible to attribute multiplicity to a relationship between classes without making the whole thing ugly, we opted not to include them in our diagram. The proportions are constant and are as follows:

   - 1 Cube: 6 Faces
   - 1 Face: 9 Squares

3. Considering that the cube object (and the objects that compose it) is the only complex javascript object in our application, with the other javascript implementations existing entirely as their own tiny entities in other files, we chose to focus exclusively on the structure and relationships of and between the different classes comprising the cube.

4 Sequence Diagrams

4.1 Case: Tutorial Page

7

4.2 Case: Rubik's Cube Practice Tool

Cube display Sequence Diagram

Viewtoggle · Viewtoggle

Change the direction that we view the cube

Returns a cube object

Cube · Cube

Cube processes directional data.

Collect contents from all 6 face objects.

Returns contents of the face (the 9 squares).

Face · Face

Collects data from all 9 squares

Returns the color of itself

Square · Square

Contains contents of its color and its respective letter.

4.3   Case: Letter Generation

Letter Diagram

Genwords → Genwords

Input a string of letters

Wordgen → Wordgen

Generate a string of words from letter input.

String of words

Output → Output

Display string of silly words.
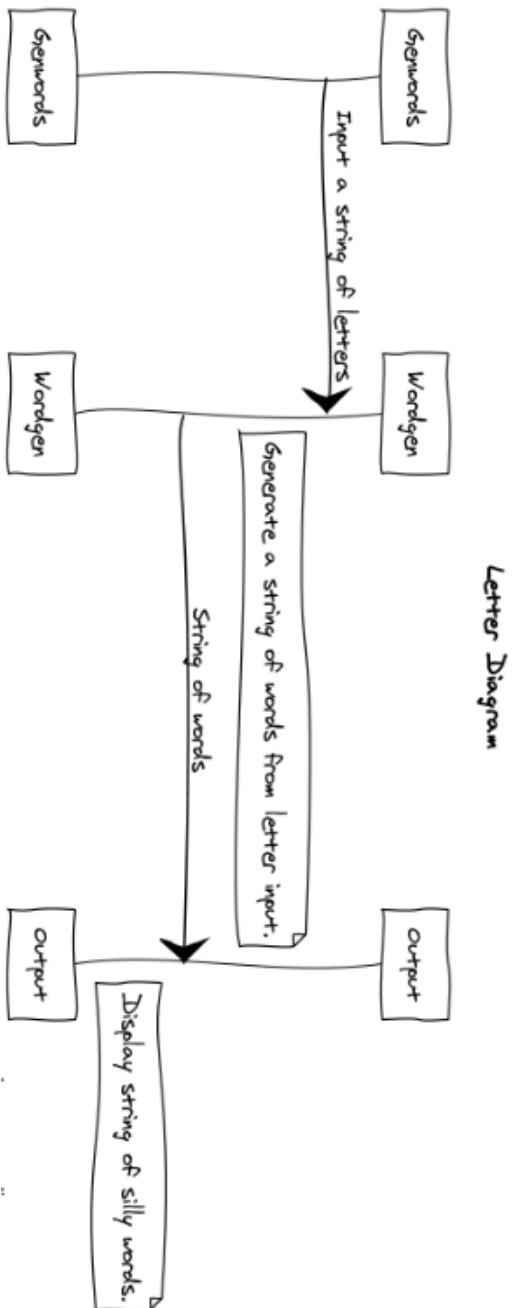
5   Meeting Report

On this week we have met 4 times on campus. Including three times in library and one meeting after class.

On our first meeting on Wednesday. We met everyone in the library. we separate our second phase of the project to every group member. Jesse Chick is working on Class Diagrams. Benjamin Martin is working on Sequence Diagrams. Jiaji Sun is working on meeting report. Nickoli Londura is working on part of User Interface Prototypes. Keenan Johnson is working on scramble code and solution generation. On this meeting, we have drawn basic prototype of our project, and basic design each functions for every function. At the beginning, we drew a basic section of the prototype on paper, then rough design each functions for every function. Then we modify the design multiple times. Finally, we settle down the first version of the project. On this meeting, we find a new feature about our project is we implement one that generates easy to memorize words for given letter pairs. Example: given the letter pair CT would output an easy to memorize word such as 'cat'. At the end of this meeting, we set up next meeting time.

On our second meeting, we met after class. Everyone describes new idea of each other. We are thinking about the new design of our project, and everyone has talked about their new idea of our project. After we discussed our new design. We decide to keep discussed our project design at next meeting. Then we set next meeting time.

On our third meeting, we met everyone in the library. we decide change design for our project. Jesse start modifies class diagrams into a new design. Benjamin also modifies his sequence diagrams into a new design. Nickoli starts working on the first version of prototypes of the project. Keenan is working on try to making code work well and smoothly. After we changed our design, the new version of our project design is come out.

On our fourth meeting, we still met in the library. We have discussed our last version of project design and final version of prototypes. Jesse keeps working on class diagrams. Benjamin keeps working on sequence diagrams. Nickoli keeps working on the prototypes of our project. Keenan compiled every code to make sure it works well. At the end, Jesse and Benjamin finished their diagrams. Nickoli finished the prototypes of the project. Keenan compiled every code, and make

every code working well. We have done our parts nicely.

Our plan for next week keeps working on the prototype of our project. Try to implement every function of each module. Make them be the first version of the code, and test them to find bugs.

[1]

References

[1] *How to Solve the Rubik's Cube Blindfolded.* Ruwix, 2018.