

# The Memo

Jesse Chick      Benjamin Martin      Keenan Johnson  
Nickoli Londura      Jiaji Sun

March 12, 2018

## Contents

### 1 Contributors and ONIDs

LINK TO THE GITHUB REPO AT THE CORRECT BRANCH: [https://github.com/Tonyenike/CS361-001-W2018/tree/the\\_memo-assignment-7/projects/martinb3/assignment-7](https://github.com/Tonyenike/CS361-001-W2018/tree/the_memo-assignment-7/projects/martinb3/assignment-7)

- Jesse Chick ~ chickj
- Benjamin Martin ~ martinb3
- Keenan Johnson ~ johnsoke
- Nickoli Londura ~ londuran
- Jiaji Sun ~ sunji

### 2 Product Release

Currently, we have no place where we are serving our website. The link to access our product GitHub is here: <https://github.com/Tonyenike/CS-361-Group-Project>. Clone the repository and follow the instructions included in the instructions.txt file.

### 3 User Stories

1. **View Tutorials** implemented and tested. This is incomplete, the links for the actual tutorials are now coded into the website. Coded by Ben/Nickoli. Took roughly 1 week to complete this far. No problems encountered so far. This story is complete.
2. **Rotate Cube** implemented, testing included. This function is completed, the only thing left is bug testing. Coded by Ben. Took roughly 5 days to complete at this point, including testing. Bug testing on this method currently consists of one unit test. One of the problems that we encountered was finding out how to rotate a face clockwise and counterclockwise, as that would invariably affect other faces on the cube in unforeseen ways. This was solved by implementing a special face function called rotateFace which could be rotate clockwise or counterclockwise as the cube engine desired. This story is complete.
3. **Scramble Cube** We are abandoning this feature from the visual perspective. This means that we have no implementation or testing on this feature anymore, see details below (see **Design Changes and Rationale**). This was discussed at great length with our customer Keenan. Story abandoned.
4. **Switch Page** implemented, tested. Buttons to switch to different pages still needs to be cleaned up. Coded by Ben /Nickoli. Problems encountered: button resizing in the dynamic layout is not optimal, text size is not matching the button size. These problems are still unsolved and must be fixed. Almost complete.
5. **Generate Letters** implemented, no testing. No client-side implementation yet, only implemented on the back-end. The only work that has been done on this is the client-side webpage has the input field and submission button, but no operating back-end code yet. Coded by Keenan. Not yet complete.
6. **Check solution** implemented, no testing. Code packages are installed and ready to go. No problems have been encountered yet, as it is not fully implemented. Keenan is working on this. It has taken two days so far. Not yet complete.
7. **Reset Cube** implemented, tested. Coded by Ben. Took 1 day to complete. This function is completed, only thing left is bug testing. There were no

problems in implementing this method. This story is complete.

8. **Main Page Redirect** implemented and tested. Although the clientside interface has been implemented with this main page redirect. No issues. Took less than a day to complete. Coded by Ben Martin. This story is complete.
9. **Clear input bar** implemented and tested. This is a client-side feature, so testing on this user story was limited to a by-hand basis. Coded by Ben. Took roughly 1 day to complete so far. No problems in implementing this feature. This story is complete.
10. **404 Content** implemented, tested. This is a client-side feature, so testing on this user story was very limited, and will be on a personal basis (testing manually by testing different pages). Coded by Ben. Took roughly 1 day to complete so far. There was an issue with 404 errors being sent for content that the user did not have permission to, this was fixed by using `chmod` commands on all content inside the public folder. No further issues. This story is complete.

## 4 Design Changes and Rationale

Fortunately, we have the opportunity to work very closely with our customer\*, who oversees development in an advisory capacity. This allows for very strong communication of needs from the customer and reporting from the developers with regards to the requirements for the project. In broad strokes, our customer has approved resoundingly of the layout of our webpage and proposed cube UI. However, given the rigors of integrating the various components (details to be discussed later) and approaching deadlines, there may have to be some corner cutting in the form of simplifying the application and its functionality.

The development of the core cube engine is in the process of being implemented in C++. This is because development began on this part of the application before any other javascript had been written, and javascript is a pain to work with without a server or other infrastructure in place already. At this point in the application development process, continuing to implement in C++ would be unnecessary and would make subsequent tasks bigger, such as transforming the C++ code into javascript (and merging it with the existing javascript cube code we have). This is a task that was left off all of our planning documents up until now, as it

was not anticipated as a task too long back in this process.

Given the rigor of our backend developers' work- and school schedules, we are running a little bit behind schedule in the development of the cube engine. Couple this with the aforementioned need to transfer the existing C++ code to javascript, there needs to be a change to our schedule and expectations for the project. As far as getting the cube engine up and running and able to be tested thoroughly, we will have one or a few of the front end guys take time getting that cube working (this has already begun).

In a way, these developments have affirmed a decision that we, as a group, made very early on in our planning process, that being the decision to keep our expectations limited to something we knew we could get done and block off a few weeks for growth, time permitting. Since we knew we were all busy people with jobs, full schedules, and commutes, we foresaw the time crunch at the end, hence the preemptive measures. That is not to say that we did not aspire to hit every benchmark that we set in this whole process, because we surely did; rather, we geared our approach towards safely having a solid product rather than dangerously having a spectacular product. I am not sure how well this orientation towards product development transfers into the industry, but I find the lesson regarding planning with realistic (rather than idealistic) expectations in mind is important. As such, it will likely be the plan of this group to cut the iOS migration portion of the project and focus on getting our existing application performing (and looking) perfectly, and practicing the software engineering skills associated with these tasks.

Scramble Cube is no longer being implemented visually; we have opted to change the programming so that a string of moves for the scramble to the user is recommended instead. This will ease our functionality so that we don't have to represent the Rubik's Cube scrambled visually.

## 5 Tests

This time around, testing was put on a lesser emphasis.

### **Client side**

All testing that needs to be done by hand (on the client-side HTML) has all been verified.

## **Back-end**

We have testing for:

1. Rubik's Cube rotational methods
2. Cube object creation/default construction
3. Cube color scheme verification
4. Cube setFace functions

At the moment, we have no back-end testing for lettergen, cube solving or cube scrambling. All test parameters test various input points on the methods described, and all return in success. This verifies that all tested methods are bug-free to our knowledge.

```
/*
 * test.js
 *
 *
 * Write any unit tests here to test our backend code!
 * I have written one as a quick example. Type
 * 'node test.js' to run the tests.
 *
 */

var cubeEngine = require('./engine.js'); // This lets us use the funct

/*
 * Styling points for unit testing!
 * Only works on *NIX operating systems though.
 * Run this on the ENGR server for best results.
```

```

*
*/

Reset = "\x1b[0m"
Bright = "\x1b[1m"
Dim = "\x1b[2m"
Underscore = "\x1b[4m"
Blink = "\x1b[5m"
Reverse = "\x1b[7m"
Hidden = "\x1b[8m"

FgBlack = "\x1b[30m"
FgRed = "\x1b[31m"
FgGreen = "\x1b[32m"
FgYellow = "\x1b[33m"
FgBlue = "\x1b[34m"
FgMagenta = "\x1b[35m"
FgCyan = "\x1b[36m"
FgWhite = "\x1b[37m"

BgBlack = "\x1b[40m"
BgRed = "\x1b[41m"
BgGreen = "\x1b[42m"
BgYellow = "\x1b[43m"
BgBlue = "\x1b[44m"
BgMagenta = "\x1b[45m"
BgCyan = "\x1b[46m"
BgWhite = "\x1b[47m"

function unit_test1()
{
// This test checks to see if the moveFace function is working
// as expected.
var cube = new cubeEngine(); //The cube variable is our cube object.

console.log("");
console.log(FgCyan + "=== Beginning Unit test 1 ===" + Reset);

```

```

console.log("-----");

cube.moveFace("top", "bottom");
console.log("=== setting the bottom face to be equal to the top face")

var topface = cube.getFace("top");
var botface = cube.getFace("bottom");

var midbotcol = botface.getSquare("mm").getColor();
var midtopcol = topface.getSquare("mm").getColor();

console.log("=== bottom square is now:", midbotcol);
console.log("=== top square is now:", midtopcol);

var stat;
//Assertion
if(midbotcol === midtopcol)
{
stat = FgGreen + "SUCCESS" + Reset;
}
else
stat = FgRed + "FAILURE" + Reset;
console.log(FgYellow + "=== TEST STATUS:" + Reset, stat);
console.log("");
}

function unit_test2()
{
// This test checks to see if the rotateCube function is working
// as expected. This test uses the "up" direction.
var cube = new cubeEngine(); //The cube variable is our cube object.

console.log("");
console.log(FgCyan + "=== Beginning Unit test 2 ===" + Reset);
console.log("-----");

```

```

var frontface = cube.getFace("front");
var midfrontcol = frontface.getSquare("mm").getColor();

console.log("=== front face is color:", midfrontcol);


cube.rotateCube("up");
console.log("=== rotating the cube up");


var topface = cube.getFace("top");
var midtopcol = topface.getSquare("mm").getColor();

console.log("=== top face is color:", midtopcol);


var stat;
//Assertion
if(midtopcol === midfrontcol)
{
stat = FgGreen + "SUCCESS" + Reset;
}
else
stat = FgRed + "FAILURE" + Reset;
console.log(FgYellow + "=== TEST STATUS:" + Reset, stat);
console.log("");
}


function unit_test3()
{
    //This test will check to
    //see that the getFace function is working properly
    //And that default colors are correct.


    var cube = new cubeEngine(); //The cube variable is our cube o


    console.log("");
    console.log(FgCyan + "=== Beginning Unit test 3 ===" + Reset);
    console.log("-----");

```



```

var backface = cube.getFace("back");
var leftface = cube.getFace("left");

var col = backface.getSquare("mm").getColor();
var othercol = leftface.getSquare("mm").getColor();

console.log("=== color of back face:", col);
console.log("=== color of left face:", othercol);

console.log("=== default coloring scheme is:");
console.log("=== front : red");
console.log("=== back  : orange");
console.log("=== left  : green");
console.log("=== right : blue");
console.log("=== top   : white");
console.log("=== bottom: yellow");

//Assertions
if(othercol !== col &&
    col === "orange" &&
    othercol === "green")
{
    stat = FgGreen + "SUCCESS" + Reset;
}
else
    stat = FgRed + "FAILURE" + Reset;
console.log(FgYellow + "=== TEST STATUS:" + Reset, stat);
console.log("");
}

console.log("");
console.log(FgCyan + "=== Beginning Unit Testing ===" + Reset);
console.log("-----");

unit_test1();
unit_test2();
unit_test3();

```

```
console.log("");  
console.log(FgCyan + "=== End of Testing ===" + Reset);  
console.log("-----");  
console.log("");
```

## 6 Meeting Report

This week is week nine. On this week we have met three times on campus. Including two times in library and one meeting after class. Our customer has met us in three meetings. We keep working on his requirement for our project. At this time, we basically finish up implement his requirements in our project.

Our first meeting is on Wednesday. We met in the library this time. On this meeting, we have looked through whole assignment requirement after the assignment come out. Then, we started to divide this assignment into five pieces. Every group member will work on one part. Since our product release and user story doesn't change. We decided to keep using product release and user story from the last phase. Benjamin Martin is working on testing, Jesse Chick is working on design changes and rationale. Keenan Johnson is working on refactoring. In addition, they are also working on slides for presentation next week. Jiaji Sun is working on meeting report. Nickoli Londura is working on the client-side and back-end. Because of next week is the dead week, we decided to try to finish this assignment as soon as possible. Therefore, we decided to keep just meet everyone three times this week. Then, we will keep track our assignment process by texting each other. After we decided our plan for this week, we start to keep working on implement project. We are trying to finish our project as soon as possible this week. So far we finished major feature and interface of this project. We basically implement all customer's requirements at this time.

Our second meeting is after class on Thursday. We discussed all draft testing plans for the whole project. We are planning to do that at next meeting Saturday. After that, we just keep contact and tracking process of this project by texting.

On our third meeting Saturday afternoon on the library, we are working on the testing whole project together. We are trying to find any bugs in our project.

Since we have done the unit test before, we are not doing unit test again. Therefore, we trying to find any problem with entire project side. We pretend regular users and using this website, then trying to find any problems with this website. After our project is done. We are started working on rest sections of the document. This time we discussed the entire document together instead of separate parts. We have discussed design changes and rationale and refactoring first. After we settle down those two sections. We started rest sections. The document is taken most time of that day. We were finally done with that at the end.

Our plan for next week keeps working on our project and try to finish up our project in detail decoration. Then we will finish this project totally.