

# The Memo

Jesse Chick      Benjamin Martin      Keenan Johnson  
Nickoli Londura      Jiaji Sun

March 5, 2018

## Contents

<b>1</b>	<b>Contributors and ONIDs</b>	<b>1</b>
<b>2</b>	<b>Product Release</b>	<b>2</b>
<b>3</b>	<b>User Stories</b>	<b>2</b>
<b>4</b>	<b>Design Changes and Rationale</b>	<b>3</b>
<b>5</b>	<b>Tests</b>	<b>5</b>
<b>6</b>	<b>Meeting Report</b>	<b>10</b>

## 1 Contributors and ONIDs

LINK TO THE GITHUB REPO AT THE CORRECT BRANCH: [https://github.com/Tonyenike/CS361-001-W2018/tree/the\\_memo-assignment-6/projects/martinb3/assignment-6](https://github.com/Tonyenike/CS361-001-W2018/tree/the_memo-assignment-6/projects/martinb3/assignment-6)

- 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, not tested. This is incomplete, the links for the actual tutorials must be coded into the website. Coded by Ben/Nickoli. Took roughly 1 week to complete this far. No problems encountered so far.
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.
3. **Scramble Cube** not implemented, no testing. No work has been done on this yet. No problems have been encountered yet as there is no work done on it. Jesse/Phillip are working on this.
4. **Switch Page** implemented, no testing. 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.
5. **Generate Letters** not implemented, no testing. 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 Ben/Nickoli.

6. **Check solution** not implemented, no testing. No work has been done on this yet. No problems have been encountered yet as there is no work done on it. Jesse/Phillip are working on this.
7. **Reset Cube** implemented, no testing. 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.
8. **Main Page Redirect** not implemented, no testing. Although the client-side interface has been implemented, no link redirect has been added to the website yet.
9. **Clear input bar** implemented, no testing. This is a client-side feature, so testing on this user story will be very limited. Coded by Ben. Took roughly 1 day to complete so far. No problems in implementing this feature.
10. **404 Content** implemented, no testing. This is a client-side feature, so testing on this user story will be 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 issues at the present moment.

## 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.

**\*Note:** If it were not clear at this point, our "customer" is our very own Keenan. As it happens, Keenan is the only qualified customer that any of us know for a project such as this one, as he is actually a blindfolded cuber, so he knows what the application should look like and which actions it should perform; he is the man with the vision. As such, the communication between customer and developer is less of a rhetorical dialog and more of a socratic discussion.

## 5 Tests

We currently have 1 unit test for a major piece of our system that is also a user story: the rotateCube method. This method is what rotates the cube object. The second unit test (called unit test2) tests the functionality of the rotateCube method.

At the moment, all client-side functionality (the in-browser session) does not exist. We have back-end code and client-side pages, but we do not have the engine linked up to the webpages yet. This means that there is nothing to test on the client side yet. The testing is completed through test.js, which tests exclusively our back-end code.

One test that is not yet implemented is the test that checks for the checkSolution user story. There is no validation step to see if the user is actually correct or not based on input and there is no testing on this method yet. Since this method does not exist, we cannot test it. To create a unit test though, we would do the following:

1. Generate a psuedo-scrambled cube (i.e., we already know the solution to this cube)
2. The test would enter multiple incorrect answers as well as the correct answer.
3. Check that the function returned false for all incorrect solutions and true for the correct solution.
4. If the answer was false, the test would also check to see that a correct solution was given as a wrapper object as well (to give the user the correct answer after being incorrect)

Some of the user stories cannot be tested through unit testing means. The View Tutorials story, for example, cannot be tested through conventional means as it is a hyperlink to another website, which a js unit test cannot validate.

/ \*

```

*   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 eight. On this week we have met three times on campus. Including three times in library and one meeting after class. Our customer has met us in three meetings. We think his requirements are reasonable for this project. We basically implement his requirements in our project.

Our first meeting is on Tuesday after class. 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. Benjamin Martin is working on test section, Jesse Chick is working on design changes and rationale. Jiaji Sun is working on meeting report. Keenan Johnson is working on product release. Since we are close to the dead week, we decided to try to finish this assignment as soon as possible. Because everyone has so many assignments to do, and some of them are

very difficult to finish. Due to this issue, we decided we will just meet everyone three times this week. However, we will keep track our assignment process by texting each other. After we decided our plan for this week, we started to work on product release and user story. Our product will release on GitHub, it's the best place to release our project. For user stories part, we will keep working on our user stories from last assignment. At the end of this meeting, we will have a second meeting on Thursday class time. Because we don't have class on Thursday.

Our second meeting is on Thursday, during our class time. We met in the library again. On this meeting, we started to discuss the third part of this assignment, which is design changes and rationale. This part is the most complex part of this assignment. Since our customer showed up at our meeting every time. We have asked him a few questions and introduced the new design of this project. The customer is satisfied our new design and answered our questions.

Our last meeting is on Friday. We met in the library. On this meeting, our major topic is to finish rest of this assignment, which is testing. This part is the most difficult part of this assignment. The reason is we have to write some codes to create module testing. We are not only writing some codes for the testing module but also make those codes are working for our module. We have discussed a series of possible problems with our codes. However, our project is a unique project, which means there is no similar example online. We have to create our code by ourselves. This is a big challenge for us. Jesse wrote testing codes, and we use those codes to test our project modules. We got lots of issues at the beginning, such as the output of the testing code is not right. We have adjusted our codes many times. Therefore, we keep changing our code in two hours. We finally make testing codes working correctly at the end. Then we started to use those codes to test our project module. We spend another three hours to make testing codes working on our project. After five hours working, we finally finished this part at the end.

Our plan for next week keeps working on our project and try to find any bugs in our project. The new schedule for next week will be we are trying to finish these user stories: "View tutorials", "Rotate Cube", "Scramble Cube", "Switch Page" and "Generate Letters". We are planning to be done these user stories in five weekdays. Then we will use two days weekend to testing each story.