**Initial Planning**

For this phase, we segmented our task into various sub-problems and assigned at least one task to each of the members on the team. To co-ordinate our individual activities and ensure that we reached our deadlines, we held meetings – either on skype or in person – to discuss our progress and next steps. The frequency and time of these meetings were decided by the scrum master. Additionally, the scrum master was responsible for taking notes at each of our meetings and recording the progress of the group. The scrum master also analyzed our team meetings and synchronized and adjusted our plans as necessary. Our team elected Daniel to be our scrum master, however due to personal matters, this was later changed to Allen.

In order to estimate the size of each task, our group took into account several factors. These factors included, but were not limited to, how long we perceived the task would take, how much knowledge the task would require, and how abstract the topic seemed. Since we all had varying degrees of understanding and background in web development, this meant that, while some tasks could have been perceived as “easier” according to our requirements listed previously, they were not easy for the individual executing the task. Therefore, experience was also a factor when estimating task size (but not its difficulty).

**Sprint Backlog**

Initially we planned to build a website that would be capable of teaching users C through a variety of methods. This website would have had to have been capable of compiling and testing C code, arranging topics in a hierarchical tree structure that linked to other pages, have interactive games, and more. After we designated tasks that we would have liked to accomplish in order to finish our product (listed above), we assigned deadlines and individuals to each tasks. We decided to implement a subset of our initial tasks in order to produce a functional MVP for this phase.

The tasks for this phase were divided and estimated as follows:

1. **User Account System** [Left for later phase]
   1. This should contain a log-in page for existing users
   2. Should also contain a sign-up page for new users
   3. Needs a database to store usernames
   4. Implement user interface for user to see their profile, settings, etc
2. **Home Page Interface** [Left for a later phase]

* Using the interface implemented for the entire webpage, implement the conditions such that the user is able to see and access the Programming Tree interface when they login or register
* Include in content for what the Programming Tree webpage is about, and what is offers to users who are interested in learning C.

1. **Programming Tree** [Shamama and Daniel].

* Shamama had some experience in HTML & CSS, but she was not familiar with JavaScript. Daniel was not familiar with any of the three languages. Therefore, despite the difficulty of the task, the assigned deadlines were longer than expected as they two had to pick up the languages to implement the tree node.
* ***Estimated Task Difficulty:*** Medium – There is no predefined method of implementing a visual tree on the web that the group knew of. Despite this, it was evident that this would only require the knowledge of HTML, JavaScript and CSS to complete. Therefore this task was evaluated to be of medium difficulty.
* ***Intermediate Deadlines*:** 
  1. Implement nodes that are capable of linking to new pages: Small – (February 26, 2016)
     + The two needed time to learn additional languages so despite the fact this is the easiest task, it was given the longest amount of time
  2. Arrange them in hierarchical structure as specified by the administrator: Medium – (February 29, 2016)
     + This is a somewhat difficult task to finish as it is a little abstract so they were given more time for this one.
  3. Create visual links (e.g., lines) to show relation between different nodes: Small – (March 3, 2016)
     + This is should be quite simple to implement as there are methods of doing this listed online, therefore it was given the shortest amount of time.

1. **Content Material** [Mostly to be left for another phase – memory allocation game implemented]

* Set up content material and pages for teaching users C
* Come up with one logical C Node Tree structure that will teach one full concept with a minimum of at least three nodes in the tree.
* Create ideas for activities and games that can be used to help teach students C (node content)
* Create ways to test C understanding by using C compiler and asking user to solve specific problems.
* Test the code obtained from the user to ensure that the user code functions as expected
* **Implement memory allocation game:** [Completed by Justin] – set up page with canvas element as the base for the game

1. **Setting up webpage layout** [Flora]

* ***Estimated Task Difficulty:*** Flora was familiar with HTML and CSS and thus she was responsible for this task. Since her portion did not require too much additional learning (as none of these components require JavaScript and she familiar with the rest), she was asked to do more tasks.
* ***Intermediate Deadlines***: Since Flora was familiar with all of the languages she required for her task and all of her tasks were of the same difficulty we decided to split them evenly throughout this phase.
  1. Implement header for webpage name: Small - (February 24, 2016)
  2. Implement a navigation between the different pages in the header: Small (February 27, 2016)
  3. Implement a main section under header for page content: Small (March 1, 2016)

**[Complete the Estimated Difficulty and Intermediate Deadlines below]**

1. **Setting up Node.js and Server** [Allen and Wenfeng] – since these two had the most experience with web programming and were capable of writing code in HTML, CSS and JavaScript, they were asked to complete this task
2. **C Compiler for Exercises** [Allen and Wenfeng] – again as the two had the most experience with web development, they were also asked to research into how to complete this. This is a very integral part of our idea and as such, it was of high importance that it was completed by this phase.

**Update Meetings**

**Meeting One:**

Date: Feb 21 2016, 12:00-1:00PM | Skype | Attendance: Everyone

The group members all collectively went on skype, and discussed the planning stages of the project. In detail we went over what options we could have went with in terms how to implement our project idea.

* Easier implementation with App development, but has a more limited scope in terms of user interaction and small screen interface (and possible challenges in expanding to larger devices such as a tablet)
* HTML, CSS & JavaScript: which more members are not familiar programming with, but have an easier interaction for users.

**Meeting Two:**

Date: Feb 22 2016, 8:00-9:00PM | BA2159 | Attendance: Everyone

1. The group members discussed the concerns we had about our project idea since not all members were yet fully on board with the idea of the Programming Tree website.
2. When talking with the TA the group asked questions on the transitions from phase one to phase two of the project, and listened to the feedback on our submissions from phase one. The group asked if our idea was good to go even though our phase one reports were not detailed enough.
3. Over a few more concerns addressed between the group members the decision to implement the project with HTML, CSS and JavaScript was made.

**Meeting Three:**

Date: Feb 24 2016, 4:00-5:00pm | BA3200 | Attendance: Everyone

1. The group met with the TA in order to ask questions regarding the phase two details and any suggestions to our project website. TA suggested to have a “simple” C compiler implemented to use in our project for the exercises.
2. As the TA left, the group members discussed the basic plans and different components to implement the website, and the different tasks were delegated between the members. A draft of the sprint backlog was created from this discussion.

The front-end that is responsible for the content and activities of the webpage were assigned to Daniel and Justin – as they went on to discuss some details to implement a game for teaching users memory in C.

The back-end members responsible for visual elements and different components of the page were left to Allen, Wenfeng, Flora and Shamama. Allen and Wenfeng decided to take on the task of researching and implementing the C compiler; while Shamama and Flora decided on implementing the Programming Tree data structure. Daniel also agreed to taking down the update minutes and the role of Scrum master.

**Meeting Four:**

Date: Feb 29 2016, 3:00-5:00pm | BA3200 | Attendance: Everyone except Daniel

From discussion of the progress between members:

* Server for the website implemented using Node.js
* Website layout implemented using HTML & CSS
* Basic format and figures for memory game implemented using HTML, CSS and JavaScript
* Nodes with on-click window redirect implemented using JavaScript

During this meeting the group members discussed the progress of what each member did, and continued to work on their delegated tasks. Various discussions such as whether to use the node popup system or node page redirection for the interface were held.

**Meeting Five:**

Date: Feb 29 2016, 8:00-9:00pm | BA2159 | Attendance: Everyone except Daniel

* Group members discussed the project progression with the TA, stating that all was going very smoothly and the basic components were set up. The group members asked questions regarding the demo of phase two, and what was expected of us for the process and product reports.
* Group members then discussed about meeting up the next day in order to work on the two reports of phase two.
* A decision was made to meet with the TA to present the demo of the project on March 7th, 5pm.

**Meeting Six:**

Date: Mar 1 2016, 2:00-4:30pm | BA3200 | Attendance: Everyone except Daniel

* Current progress of the website implementations were discussed.
* The process and product reports were discussed and were in works or writing
* Flora worked the burndown chart
* Allen, Wenfeng, Justin and Shamama discussed the write-up of the technical product report.
* After Allen, Wenfeng and Justin left at 4, Shamama and Flora discussed the write up of the process report. The other members are to finish up their version of “reviews” for the process report.

**Meeting Seven:**

Date: Mar 2 2016, 12:00 – 2:30pm | BA3200 | Attendance: Shamama, Allen, and Flora

* The group members continued to work on the process and product reports. Allen worked on the server and other content for the product report. Shamama wrote up the initial planning and finalized version of the sprint backlog. Flora worked on the burndown chart and overview of the process report

**Burndown Chart:**

**To be added once everyone includes in their information. But here’s a sample with random # of tasks. Expected # of tasks at start date is the number of tasks we planned for phase 2**

**Review and Retrospective:**

### Flora

For this phase the basic webpage layout and CSS were implemented. Where the webpage for at this stage just contains a top div element containing the website name and serves as a navigational bar. Below that is the main div element where the content for all the pages of the website will appear on.

Since I had an idea in mind where upon clicking a tree node the user is able to see the content material through an in-browser popup, I proceeded to implement the in-browser popup and have them appear whenever a tree node is clicked. First was just implementing the popups with basic text links, and then changing around HTML and CSS elements to have the text link turned into a tree node form. There were some issues regarding the properties of the node div element conflicting with the popup div element (where the node would still show up even though the popup appears) but that was fixed after some changes to CSS code.

The tasks that were not done was this idea in general, since upon discussion group members decided to have the tree implemented in JavaScript instead and on click redirect to a new page, for a better user interaction with the interface. Tasks that were split was continuing to work on the tree implementation in JavaScript.

Allen

For this phase the compiler is the feature that must be completed in order for the MVP to accomplish its task: teach students how to code in c by testing their code and passing back results. This task was split between two tasks: one is to handle the client side form request to put the code in and process it at the server side. The other was doing research on how to actually implement the compiler in the NodeJs framework. Both of these tasks were completed, but not in the best possible way for software changes. Instead of putting the compiler code in a separate JavaScript file, the code remains in the same file as the post route. The program still works as intended, but if this code is reused many times later there will be a lot of repetitive code. Thus, this is a bad design and is a goal for us in the future to fix in the next phase.

The login system and signup system was also completed using the expressJS server and MongoDB. The schema used to store the user information right now is built to adapt to the changes with an extra array field that will store the progress of the user in their lessons. An improvement that as a team that we will need address for the next phase is to write test cases to test the corner, general, and specifics behaviours of all our function/routes.

Wenfeng Jiang

I and Allen were responsible for the server and C compiler part. We did some research on compilers online and quickly divided up our parts, Allen for building a simple server and I will implement the compiler on local disk using JavaScript. The process was smooth and after the server is finished all I had to do is change the syntax for the commands and merge it with the server code. The codes needed to be inputted and saved to a file before compiling so we broke down the work further, Allen getting the input, and me writing it to a file. File writing was achieved with easy by using the Filesystem module and everything works when we connected everything.

All the outputs were using console.log () so everything is shown on the server, but when I tried to return the output the user (client-side) it would not show. This is the major problem that we have in this phase. We have tried many methods that included change the out to a “Success” or “fail” instead of the raw output, wrap the output with String type but all failed. At the end we decide to not use compile as a module, but to combine it with the main code and it worked. The solution will be temporary until we find a solution.

Overall this phase was smooth; I and Allen were familiar with web developing so we got the approaches to the problems fast. The task splitting was also great, in other courses I tried using nodejs but failed but Allen was experienced with the tool and I worked other parts that involved only JavaScript. I have also learned how nodejs works from Allen during this phase. The next phase we will focus on the content of the website by making question and decide on how we are going to test the results.