Reviewing the Development of

Mad Math Minute

One of the most common reasons to develop a program is to solve a real-world issue. This may involve a problem that affects the programmer, or could help others. In this case, a game called ***Mad Math Minute***was developed to help a teacher view the progress of their students. User played the game, and then the displayed results would allow a teacher to view how well each student did. The program was developed in about a week, in which the problem was recognized, a design was created, and the program itself coded and tested before implementation. This report will go into detail on the development process of ***Mad Math Minute****,* explaining issues faced, solutions made, the timeline of the project, and any possible improvements.

When the idea for the game was initially brought up, it was important to determine if it was the right solution, viewing how it could actually help and if other options should be viewed. In this *Program Definition* stage, it was determined that the game could assist the teacher and their students. The teacher could receive feedback from the students by presenting them with a pressure-free assessment of sorts, and then use the scores to find the stronger and weaker students in the class. The initial stage was relatively straightforward, without issues of some sort. Following this, the more thorough *Analysis* stage began, where the more intricate details were laid out. The various screens were designed, as were the variables to allow the game to function smoothly. A general layout for the various screens was created in a number of images. Additionally, a Gantt chart, which will be explained later, formed the planned timeline for the project. One major issue skipped in this stage, and left out of future stages, was the prospect of an actual timer. Creating and transferring the timer between multiple dialogs was extremely confusing, but eventually that concept was removed from the program. Another issue was returning both the name of the player and their score from the game method to the main method, and later to the scoreboard, only able to return 1 variable. The solution was to concatenate the various values into a single string, divided by slashes. This would mean that, in the scoreboard method, a combination of searches, using indexOf() and substring(), would allow each item to be extracted from a single variable. Lastly, the issue of restructuring the division question to match the requirements was simply solved by creating a fourth variable to allow the others to move around. With the key issues solved, the code layout could begin.

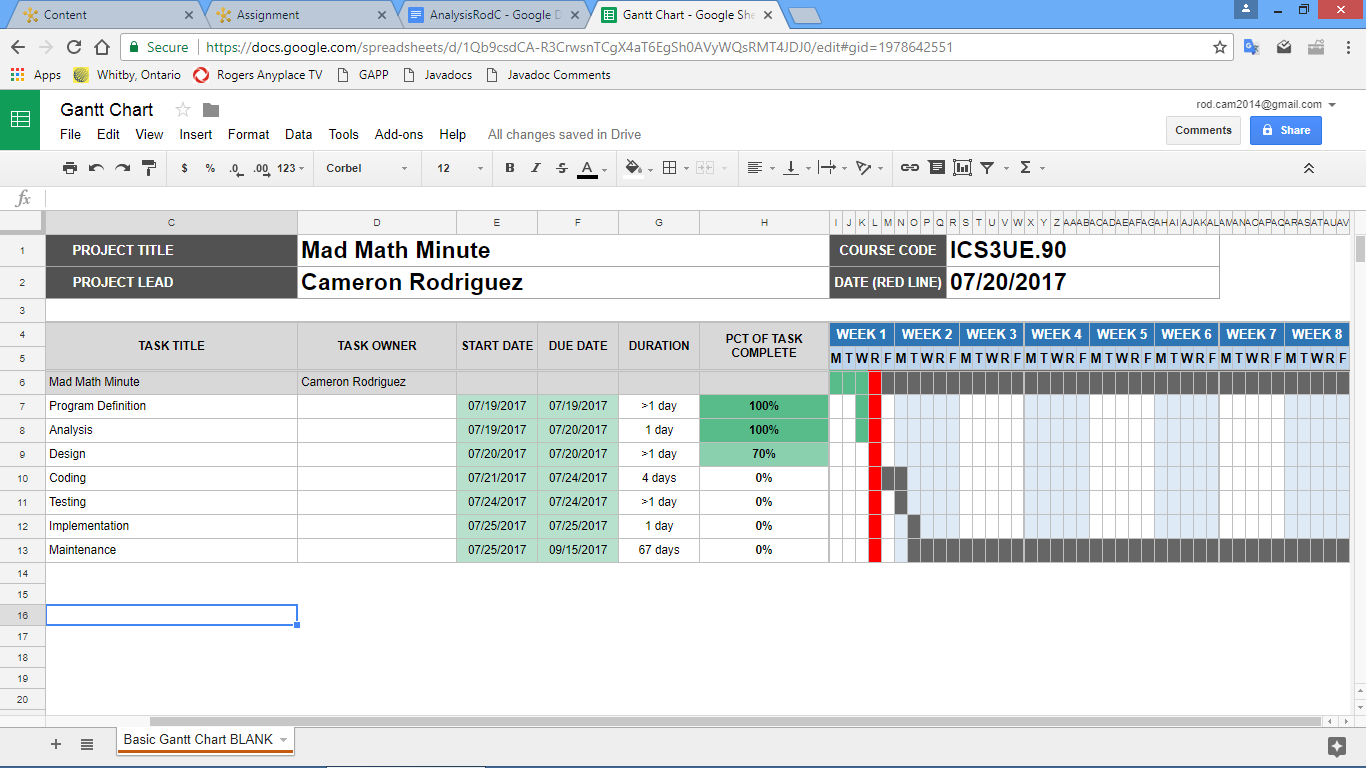
The third stage, *Design*, involved the layout of the program’s specific code. The variables required were determined and named, and a number of flowcharts were created to display the layout of the methods. The main issue was returning data to the main menu. The simplest solution was to return an updated array, fully formatted and completely ready to be displayed on the scoreboard. The completed design stage led to the *Write Code* stage, combined with the *Testing* stage, where the code itself was created and commented, using various design components. A few issues were encountered and successfully solved. The possibility of the program crashing if a letter was inputted on the main menu was solved by using a String and the charAt() search. The name and score were able to be registered separately, by using a method for each part, rather than one large String array. Another issue was keep data in line. This was solved by observing the movement of tabs (based on Monospace 11 font), and then instructing the program to increase or reduce the number of tabs based on the length of the player’s name, in order to keep all the scores in line. Testing of the code throughout the process kept the spacing of user-facing text in line, and pointed out the first of the problems listed above. Following this, the *Implementation* began, and the program was sent off.

Figure 1: The Gantt chart, displaying the planned timeline for the *Mad Math Minute* Project.

The project was able to be completed ahead of time, and the program was delivered a day early. *Figure 1* displays the Gantt chart for this project. *Program Definition* and *Analysis* were both completed on their specific timelines, and were finished on July 19 and July 20, respectively. The *Design* stage, however, took longer than expected, and although it was completed on the 21st, the *Coding* stage did not begin until the following Monday, leaving it a day behind schedule. However, this stage was quicker than expected, and with a relatively quick and bug-free *Testing* period, the final stages were completed on July 24, one day before the deadline. *Maintenance*, for the time being, is unneeded.

Overall, the project was completed ahead of time and without bugs. The program, for the most part, works as expected, and no major issues exist. However, there are possibly a few changes that could be made in future updates. The addition of a timer, even if it wasn’t visible to the user, is a key addition to stay true to the game’s title. Moving the game into the more user-friendly JOptionPane would also be a useful addition. This would also allow for the program to format the scoreboard with a more clear JTextArea. Otherwise, the program works well, and no other changes should be needed.