**CMSI 4072 HW1**

1. What are the basic tasks that all software engineering projects must handle?
   1. Requirements gathering
   2. High-level design
   3. Low-level design
   4. Development
   5. Testing
   6. Deployment
   7. Maintenance
   8. Wrap-up.
2. Give a one sentence description of each of the tasks you listed in Exercise 1
   1. Planning and figuring out requirements - what the customer wants and needs
   2. Systems are being used in the project, splitting into major sections
   3. How things will work, smaller details from high-level design
   4. Coding work
   5. Testing for bugs
   6. Software rollout, and reacting to potential issues following rollout
   7. More bug fixing and adding features based on users interactions with the software
   8. Reflect on process, prepare for future undertakings
3. Changes are shown in a green highlight, this applies to new text and lines added. However, it does not show changes to font size and color. Additionally, the green highlight makes it impossible to see changes in font color. Versions show information such as the name given to them by the user, the name of the user, and date plus time of when the version was saved. In comparison to GitHub this works in a pretty similar fashion, looking at a branches commit log similarly highlights the changes made to a file – added lines are highlighted in red, and removed lines and highlighted in red. On the other hand the classification of changes is done differently, rather than letting users name the version commits are assigned a unique commit number. At the same time GitHub displays the commit message users chose and the name of the user who made the commit. This is a better and more advances system as it makes it impossible for users to have 2 version with an identical identifier.
4. What does JBGE stand for and what does it mean?  
   Just Barely Good Enough, the idea that one should not spend too much time of documentation, making it just good enough where you can move on and work on other aspects of the process.
5. Use critical path methods to find the total expected time from the project's start for each task's completion. Find the critical path. What are the tasks on the critical path? What is the total expected duration of the project in working days?  
   Critical path: Start → C, G, H, A → L, B, J, O → K, N→ R → Finish  
   Tasks: Texture editor, Rendering engine, Humanoid base classes, Robotic control module, Test environment editor, Texture library, Zombie classes, Zombie editor, Test environment, Zombie library, Zombie testing  
   Duration: 0 + 4 + 6 + 3 + 5 + 6 + 5 + 3 + 5 + 5 + 15 + 4 + 0 = 61 days
6. Build a Gantt chart for the network you drew in Exercise 3. Start on Wednesday, January 1, 2024, and don't work on weekends or the following holidays:  
   2nd longest critical path: Start → C, H, A, G → B, J, O → N  
   Gantt chart: [Uploaded as Excel sheet]
7. In addition to losing time from vacation and sick leave, projects can suffer from problems that just strike out of nowhere. Sort of a bad version of deus ex machina. For example, senior management could decide to switch your target platform from Windows desktop PCs to the latest smartwatch technology. Or a pandemic, hurricane, trade war, earthquake, alien invasion, and so on could delay the shipment of your new servers. (Not that anything as far-fetched as a pandemic might occur.) Or one of your developers might move to Iceland. How can you handle these sorts of completely unpredictable problems?  
   To plan ahead for unpredictable events you should add some extra time to your planning for any setbacks. When creating a timeline, there should be free time at the end such that a delay does not leave you behind.
8. What are the two biggest mistakes you can make while tracking tasks?
   1. Ignoring the problem, hoping to make up time elsewhere.
   2. Adding more people to a task that is behind, it only slows it down more.
9. List five characteristics of good requirements.
   1. Clear
   2. Unambiguous
   3. Consistent
   4. Prioritized
   5. Verifiable
10. Suppose you want to build a program called TimeShifter to upload and download files at scheduled times while you're on vacation. The following list shows some of the applications requirements. For this exercise, list the audience-oriented categories for each requirement. Are there requirements in each category?
    1. Allow users to monitor uploads/downloads while away from the office. - User
    2. Let the user specify website log-in parameters such as an Internet address, a port, a username, and a password. - User
    3. Let the user specify upload/download parameters such a number of retries if there's a problem. - User
    4. Let the user select an Internet location, a local file, and a time to perform the upload/download. - User
    5. Let the user schedule uploads/downloads at any time. - User
    6. Allow uploads/downloads to run at any time. - Business
    7. Make uploads/downloads transfer at least 8 Mbps. - User
    8. Run uploads/downloads sequentially. Two cannot run at the same time. - User
    9. If an upload/download is scheduled for a time whan another is in progress, it waits until the other one finishes. - Functional
    10. Perform schedule uploads/downloads. - Business
    11. Keep a log of all attempted uploads/downloads and whether the succeeded. - Functional
    12. Let the user empty the log. - User
    13. Display reports of upload/download attempts. - Functional
    14. Let the user view the log reports on a remote device such as a phone. - Business
    15. Send an e-mail to an administrator if an upload/download fails more than its maximum retry number of times. - Nonfunctional
    16. Send a text message to an administrator if an upload/download fails more than it's maximum retry number of times. – Nonfunctional
11. Figure 5-1 [right] shows the design for a simple hangman game that will run on smartphones. When you click the New Game button, the program picks a random mystery word from a large list and starts a new game. Then if you click a letter, either the letter is filled in where it appears in the mystery word, or a new piece of Mr. Bones's skeleton appears. In either case, the letter you clicked is grayed out so that you don't pick it again. If you guess all the letters in the mystery word, the game displays a message that says, "Contratulations, you won!" If you build Mr. Bones's complete skeleton, a message says, "Sorry, you lost.". Brainstorm this application and see if you can think of ways you might change it. Use the MOSCOW method to prioritize your changes.
    1. M – Must not pick the same word for another 15 games. Must not have inappropriate language in the word pool.
    2. S – Should have a responsive keyboard, and visuals
    3. C – Could have a nicer graphic, preferably one with no white background
    4. W – Won’t have background music