Alex Verdin

CMSI 402

Prof. Johnson

January 29, 2018

Homework 1

* 1. The basic tasks that all engineering projects must handle are: project requirements, high-level design, low-level design, development, testing, deployment, maintenance, and wrap-up.
  2. The requirements task has developers gathering information from the customer as to what needs to be implemented in the project. The high-level design is a description of the various aspects of the project such as platform, data design, and interfaces that the project will use. Low-level design is a mockup of how the project will function. Development is the actual writing of code for the project. Testing means checking the project code for bugs either by self-testing or by outsourcing testing to someone else. Deployment is the stage where the project is released to the customer. Maintenance is the task where developers may need to update the code after the project has been released as new bugs or features may be revealed by the customer. The wrap-up stage is where the developers reflect on what worked and did not work with the project in order to make future projects more successful.
  3. JBGE stands for “Just Barely Good Enough” and it is a philosophy that is sometimes applied to code documentation. Some developers do not like to waste time writing comments to explain their code so they implement the JBGE philosophy meaning they only write as few comments to get by explaining their code. This is not always a good philosophy, but it is better than writing no comments. Developers really only implement this philosophy for code documentation and not actual development and testing itself.
  4. The tasks on the critical path are G (Rendering Engine), D (Character Editor), E (Character Animator), M (Character Library), and Q (Character Testing). The expected duration of the project in working days is 32 days.
  5. See Back.
  6. Ways to manage unpredictable problems are known as risk management. Risk management involves assessing how difficult a problem may be, as well as various risks that may come from the task. A part of risk management is allotting time for unexpected problems that may slow progress and tracking the progress of tasks. This means giving more time to tasks than may be needed. This adds flexibility in the schedule in case something does go wrong with the task. If the task is finished ahead of the schedule, the schedule can allocate more time to the next task.
  7. One of the biggest mistakes you can make while tracking tasks is being too confident in being able to make up lost time and ignoring the problem altogether. It is better to predict that you may fall behind if a problem arises and to properly allocate more time to the problem ahead of time rather than last minute. The second biggest mistake is to add new developers to tackle the problem to try to shorten time. This may actually slow down production if the new developers are constantly having to be brought up to speed by the developers already working on the project. It is better to just plan early that the task may fall behind schedule.
  8. The five characteristics of good requirements are that they are clear, unambiguous, consistent, prioritized, and verifiable.
  9. a. user, functional

b. user, functional

c. user, functional

d. user, functional

e. user, functional

f. nonfunctional

g. nonfunctional

h. nonfunctional

i. nonfunctional

j. functional

k. functional

l. user, functional

m. user, functional

n. user, nonfunctional

o. functional

p. functional

There is no business and integration categories probably because these requirements are primarily focused with explaining how the project works and what it does rather than explaining potential business goals or what platforms it will be implemented on.

* 1. Depending on the timeframe given for the game, the game can be even more simplified or made more complex. However, I prioritized the game mechanics based off the MOSCOW method.

**Must**: The game must have some way for the player to choose a word for play whether that be through the game selecting a random word from its list or a user inputted word. The game must allow the players to choose letters a-z. The game must give an amount of chances for guessing the word as well as keep track of when the player wins or looses the game. The game must also indicate how long a word must be. Based off the simple must features, a version of this game could be made with dialog only, making one user input a word into the game to be guessed, and another user having to guess the word. The game will indicate how long the word is and will tell the player if their guess of letter is right or wrong and indicate where in the word the letter belongs. In this version, it will be up to the player to keep track of letters used.

**Should**: The game should use visuals, not just text, to indicate to the player how many letters are left as well as how many chances they have left. The game should have a menu that allows the player to enter or quit the game. A game based off of the should features would be similar to the one described in the question, having very basic features and visuals.

**Could/Won’t**: This category contains features that are not necessary to making the game function, but could make it more interesting. Therefore, the game could have a difficulty setting as well as some kind of multiplayer mode which allows more than one player to compete against one another, and the game would keep track of the players scores. The game could also have extra animations and sounds to make it more visually appealing and interesting. Going along with the multiplayer aspect, there could be online leaderboards or online functionality where others could compete against people via the internet.