Time log

28/1/2019: 1 hour spent building the framework of the game: the phrase and letter classes.
29/1/2019: 5 hours spent building the GameWindow and HangmanGame classes that control the GUI.
1/2/2019: 1 hour spent building the lists of phrases for the Movies and TV Series categories from IMDB data.table files in RStudio.
7/2/2019: 2 hours spent writing the project plan.
20/2/2019: 1 hour on use case
22/2/2019: 3 hours making state and class diagrams
3/3/2019: 5 minutes writing test plan
4/3/2019: 3.5 hours writing manual tests
6/3/2019: 6.5 hours writing automated tests
7/3/2019: 1 hour writing test report
16/3/2019: 1.5 hours polishing button and background appearance.
18/3/2019: 5 hours adding sound effects and background music.
20/3/2019: 3 hours making and animating the game over ghost.

Time log reflections:

In total iteration 1 took much longer than I thought it would. However, the project plan stage took only went 10 minutes over estimation. This is because I had to redo the plan because of misunderstanding the instructions. The bulk of the extra time came from development of the code. It took a long time to implement the code as planned because I only had experience using swing to develop applications and decided to use JavaFX because of its relevance to my other course. This meant I spent much of my time learning.

Iteration 2: My overall estimate was exactly right but the time allocation for each specific task was wrong. I used a program called Dia to make the diagrams and it took some time to learn it, and I had to redo diagrams I thought I was done with. If I did it again now, knowing what I know, it would probably take half the estimated time.

Iteration 3: This iteration went way over time estimate. I state in my iteration 3 report that I planned on using a certain framework to test my GUI but after hours of trying, I had to give up. Writing the manual tests also went over time. Writing

more testable code and learning a mocking framework will go a long way toward making testing go better.

Iteration 4: Again, I went over estimate. I spent a long time trying to format my sound effects to make it easy to run my game as an executable jar but time constraints led me to switch my implementation back to a simpler one that works only in an IDE. The other thing that added a lot of time was animating my game over ghost. Thanks to a bug I introduced to make my game wait before changing to the game over screen, all of the code I was using to animate the ghost was acting very strange. Once I changed the screen change to wait for the sound to end before switching to the game over screen, the animation worked exactly as intended.