**Lab 3 Observations**

**Chapter 4, Problem 1:**

This question asked me to comment out the code that changes the value of lastAnimationFrameTime in the togglePaused() method. The JavaScript for problem #1 has this code commented out. With this code commented out, it causes the game to “jump forward” when you resume the game. Pausing the game will appear to freeze the game, but unpausing will jump you forward to the point you would have been at had you not paused at all. Uncommenting the code fixes this issue, and you resume from the same point] where you paused.

**Chapter 4, Problem 2:**

This question asked me to replace instances of this in the code with snailBait. You can’t do this in the constructor because, at that point, snailBait is undefined. You will never be able to define snailBait before the SnailBait constructor because the snailBait variable relies on the object existing to initialize itself. However, you can do this in the prototype. Replacing all this instances in the prototype methods has no noticeable effect. The game works as it should. This may be inadvisable, because now the prototype methods are directly referencing the SnailBait object assigned to your variable. If you ever change that variable name, you will need to update all instances of snailBait in the code. Using a self variable with this assigned to it is a better option.

**Chapter 4, Problem 3:**

This question asked me to use the self technique described in section 4.2 of the textbook to improve the initializeImages() method. This technique involves creating a variable called self in the method and assigning this to it. This effectively results in a this variable that never changes regardless of context. This is a better solution to the above problem. Updating the method has no noticeable effect. The game works as it should.

**Chapter 4, Problem 4:**

This question asked me to change the countdown for when the window is refocused from three seconds to five seconds. This is easily achieved by adding two additional setTimeout() functions to the window focus event listener that displays the numbers five and four. Indeed, doing so has the desired effect and the countdown goes from five instead of three.

**Chapter 5, Problem 1:**

This question asked me to create a different animated GIF for the game’s loading screen and integrate it. I created a very small (~800 bytes) loading animation and updated the code to use that GIF instead of the snail one. It has the intended effect of updating the loading animation to my new GIF.

**Chapter 5, Problem 2:**

This question asked me to clear the browser’s cache and reload the game to see the loading screen for a longer time. I used the same files as chapter 5, problem 1 to do this. When both the cache is present and the cache is cleared, I only see the loading screen for about a second. This is likely because the files are stored locally instead of being transferred by a web server. If this were hosted somewhere else, especially on a particularly slow server, the slower load would be more noticeable.

**Chapter 5, Problem 3:**

This question asked me to update the fadeInElements() and fadeOutElements() methods so that they did not change the display CSS property. Additionally, it asked me to change the CSS for the canvas, chrome, and toast so that the display property was set to block. Doing this altered the page significantly. The top chrome remained the same, but most of the bottom chrome disappeared. The copyright text moved into the canvas. Additionally, the loading screen didn’t appear at all. The game itself still functions correctly.