William Woodard
Project 1Report

Question 1: Before executing it, what do you expect to see?

Answer 1: I expect to see the canvas start with color red, then turn green, then turn blue, then turn black and stay black.

Question 2: What do you see?

Answer 2: I see a black canvas, only.

Question 3: What do you see?

Answer 3: I see a red canvas and an alert, then a green canvas and an alert, then a blue canvas and an alert, and finally a black canvas and an alert.

Question 4: Using your knowledge of how double buffering works (discussed in class) and using the discussion in WebGL book Appendix A, explain why, in the first case (vii), you only see a black square, but in the second case (ix) you see 4 different colored squares—even though in the first case there is a delay loop that <u>should</u> allow you to see the other 4 colors, each for a brief moment.

Use a diagram (neat, scanned hand sketched pictures are ok; or use your preferred drawing tool) to help explain your answer.

Answer 4: In WebGL, the browser is in control of what is displayed, so double buffering is not needed (though in a sense, it really seems like one buffer, the WebGL color buffer, and one display "buffer", the browser, which is very similar to double buffering).

In the first case, we only see black because the black color was in the buffer at the time when the control was passed to the browser as `main()` exited. All that time used to delay the for loop was time where `main()` was in control, not the browser, so the browser never had control to draw to the canvas. The colors all cycled and then we ended up with black color (the last color in the array) when `main()` exited. After `main()` exited, control passed back to the browser, and then it drew to the canvas a black color.

In the second case, we see four different colors because each time when we call `alert()`, control is passed back to the browser until the user clicks "OK". When control is returned to the browser, it draws to the canvas, so this leads us to seeing all the colors in the color buffer as the alerts are called.