

# Final Project Peer Review: Julia Feldman

Paul Plew

April 20, 2021

## ▼ User Story

### ▽ Step-by-step

1. Click the maze button
2. Use the arrow keys to navigate the mouse through the maze
3. Return to the start screen
4. Take the cheese quiz
5. Finish the cheese quiz
6. Return to the start screen

### ▽ Features and Screens

The program begins on a start screen with two options, since this is an extension of the previous project the top option allows the user to play a cheese trivia game. The bottom button is where the new features live. When clicked this bottom button takes the user to a mouse maze (fitting with the cheese theme). In the maze the user is a mouse and will navigate themselves through a maze. Right now the maze is not fully generated and the mouse can move into some walls, but everything major works well.

### ▽ Purpose

The purpose of this program is to give the user a fun maze either before or after they take the cheese quiz.

## ▼ Review

I am going to skip over all the parts that are from the old project and only review the code that has been added since then.

### ▼ Variables

The use of variables is good to me. I think that the names are descriptive enough. On thing I noticed in the rectangle class is that the variables are not accessed using the Java convention `this.variableName` this would make the code more accessible to someone reading it for the first time.

### ▼ Setup Function

The setup function first sets up the canvas and the minim player. Then it loads some sound files and loads the XML file. Up until this point everything makes sense and is clear, however after this there are a lot of variables instantiated that should be done in their respective functions. One quick fix for this is to set up the PShape and walls in their own function, or take advantage of Java's object oriented nature and store them as an object.

### ▼ Draw Function

I really like how simple the draw function is. It determines the current state of the game and calls the correct function to render the scene. It is good in its current state.

### ▼ Class Structure

The class structure has some parts that are confusing, all I would change is to add comments for each of the classes and functions that describe what they do in more detail.

### ▼ Most Difficult Part

The most difficult part of the code to understand is the `mouse.move(...)` function. I can't figure out why it is passed an array of Rectangles, and believe there might be an easier way to implement this. You could store the rectangles locally and access them that way, and pass the move function the direction of the move.

### ▼ Best Written Part

I think the best written part of the code would either be the rectangle class or the draw function. Both of these are short and simple while still getting the work needed done.

## ▼ Adjusted Plan

Julia has stuck to her plan well and is on track with it. There have only been some slight adjustments to the way she implemented features such as changing from an array of cells to an array of walls (the functionality is still the same though). I would not have to change the plan at all.