# Iteration One

In the beginning…

## The goal

To study the starting system, then change the player avatar and themed tiles

### Tasks

|  |  |  |
| --- | --- | --- |
| **Task** | **Time Estimate** | **Actual Time Taken** |
| Analysis | 120 | 150 |
| Make a new Avatar | 60 | 60 |

### Analysis

The frame by frame animations of this game are rendered by the CanvasDisplay and State classes. State manages the manipulation of data from frame to frame, while CanvasDisplay manages the visual display based on the data provided by the State instance.

playerSprites is an image DOM Element that is used by CanvasDisplay as the source for the images used to show and animate the player avatar.

Vec (assumed to be short for vector) is a custom data type with two values (x and y), representing one of three in-game data elements:

* The position of actors within a level as x and y grid coordinates.
* The scale of actors within a level as x and y scalars.
* The motion of actors within a level as x and y speed.

### Class Diagram:



## Activity Diagram



### Planning

Because this game is going to have a theme based on the Christchurch rebuild, my player avatar will be a kiwi construction worker.

### Design

The game currently has an animated player sprite, a collectable gold coin and two tile images: a grey brick ‘wall’ and orange ‘lava’.



I will replace these graphics with a Kiwi Construction worker as my player sprite, a yellow hard-hat as my collectible, a red brick wall as my boundary, and ‘bureaucracy’ as my obstacle, represented by mountains of paperwork.



## Planning a Complex Algorithm

Define the problem

*Cannot erase drawings from canvas element*

Inputs to the routine

*status*

Outputs from the routine

*none*

Pre-conditions

*An html canvas element already exists, graphics may already be drawn on the canvas*

Post-conditions

*Canvas will be filled with a new background colour, determined by game status*

Name the Routine

*clearDisplay()*

Think about error handling

*Functionality will be tested in browser*

### Pseudocode

clearDisplay (status) {

if status == won {

colour = gold

} else if status == lost {

colour = red

} else {

colour = blue

}

fill canvas with colour

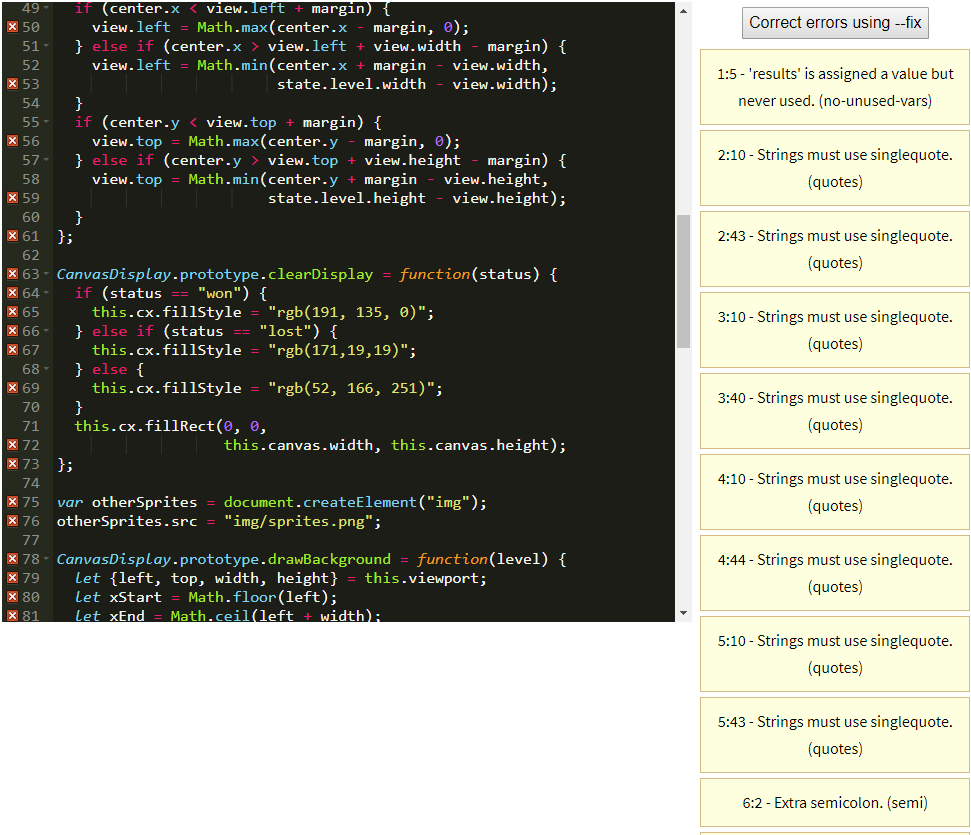
}

### Evaluation

Most of this iteration was about examining the starting system, so there wasn’t much room to make errors. I don’t feel like I have physically achieved much during the course of this iteration, due to the fact that I have made very few modifications to the original code. However, I have no doubt that the time I have spent examining and understanding the system will be very beneficial as I plan and implement changes to come.

I ran the starting system through the standardJS style checker to bring the formatting in line with my preferred style guide. These are the results:

### standardJS Before



### standardJS After

