Planning for Iteration Two

# Iteration Goal

The goal of this iteration is to add 3 features to the game. I want to add a score counter which counts how many cones the player has hit with their car and I also want to add a countdown timer for each level of 180 seconds. This is to fit with the requirement specified for the game. Finally I want to add a construction/roadworker NPC to the game with the goal being to avoid hitting them with the players’ car.

# Planned Tasks

1. Create score counter
2. Create countdown timer
3. Create art for the roadworker
4. Implement the roadworker into the game.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Estimated Time** | Score Counter | Countdown Timer | Art for Roadworker | Implement Roadworker |
| Time (hours) | 1 | 1 | 1 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Actual Time** | Score Counter | Countdown Timer | Art for Roadworker | Implement Roadworker |
| Time (hours) | 1 | 2 | 1 | 1 |

# Planning A Complex Algorithm

## Define the problem

Creating an object oriented Countdown timer class

## Information the routine will hide

Whether the countdown is finished/stopped

## Inputs to the routine

The amount of time to set the countdown timer for

## Outputs from the routine

The routine updates an html heading each second with the revised remaining time

## Pre-conditions

The countdown gets set at level load, so the game needs to have correctly loaded a level.

## Post-Conditions

None

## Name the routine

class Countdown

## Psuedocode

* Constructor function containing the time remaining, whether the countdown has finished, whether the countdown is stopped as well as a setInterval function with a 1 second interval.
* A startCountdown method which checks if the countdown is finished and stops the interval if it has and updates the html headings on the page.
* A tickdown method which gets called each setInterval loop to decrease the time remaining.
* A stopCountdown method which is called to pause/stop the countdown from continuing to countdown.
* A setCountdown method which takes a seconds parameter. This is used when the game resets or when a level is loaded to reset the time to 180 seconds or whatever seconds is specified.

# Planning A Complex Algorithm

## Define the problem

Creating an object oriented Counter class

## Information the routine will hide

Whether the counter has a minimum value and whether it has reached this minimum value, this might be useful for creating some kind of life counter in the future.

## Inputs to the routine

Potentially the minimum value for the Counter in the case of a decreasing counter (Game Lives), the element on the page that the counter will update when its value changes.

## Outputs from the routine

The routine updates an html heading each time the player collects a Roadcone

## Pre-conditions

The counter gets set once the game has loaded, so the game needs to have correctly loaded the levels etc.

## Post-Conditions

None

## Name the routine

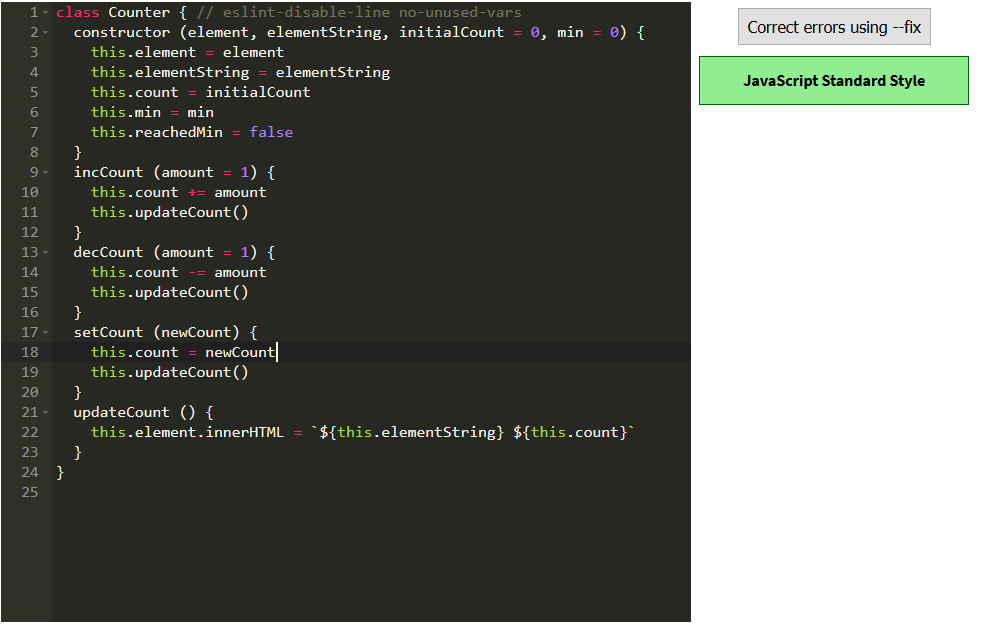
class Counter

## Psuedocode

* Constructor function that sets the element for the HTML id to update, the element string which contains the string that is used to update the HTML heading, the initial count if it exists (default is 0), a minimum count in the case of a countdown and whether it has reached this minimum value.
* An incCount method which adds a set amount to the counter and runs the updateCount method.
* A decCount method which removes a set amount from the counter and runs the updateCount method.
* A setCount method in the case of wanting to manually set a counter to a value and also runs the updateCount method.
* The updateCount method which generates the HTML string and sets it on the page.

## StandardJS

Counter.js



Countdown.js



## Mistakes Analysis

While creating the countdown timer I decided not to use the code given to us and instead creating a separate Class for a countdown. This allowed me to have additional control over the countdown by being able to invoke methods on it from anywhere in the code. I had an issue with setting an interval from within the class constructor as inside of the set interval function I could not access the Classes attributes. To work around this I have the line: var oThis = this which allows me to access the classes attributes using oThis.attribute from within the set interval function.

## Lessons Learned

As mentioned above using an oThis = this to access an objects methods and attributes from within a setInterval function inside of a class constructor.