

# CMSI 371-01

## COMPUTER GRAPHICS

Spring 2015

### Assignment 0217 Feedback

Outcomes that ultimately cover both 2D and 3D max out at | for now because we are dealing only in 2D. They will expand to their full potential with the 3D course work.

Daniel Levine

*dannymlevine / dannymlevine@gmail.com*

Bender would be proud! Your animated scene has some unique tweens with full-screen backgrounds...and music! Nicely done, artistically; now let's look at that code...

1. Indent is off by one space here. No big deal, but just letting you know that I noticed >:-) (4c)
2. \*\*\* You shouldn't have copied your sprites here. Instead, you should have modified your sprites "in place," in the *sprites* directory. Now, you have multiple copies of *Mr.Meeseeks.js*, *PlanetExpressSpaceship.js*, and *bender.js* lying around. Potentially confusing and ultimately not good practice. (4b, 4e)
3. Good job staying almost tab-free...except for *background.css*. (4c)
4. The lack of linebreak here is somewhat unusual; at a minimum it messes with the indentation scheme. Best to just linebreak between the comment end and the code. (4c)
5. You don't need these functions anymore! You should have just blown them away. You could have always grabbed them from the repository again if needed. (4b, 4c, 4e)
6. I can see what you did here, and I appreciate the conscious choice, but ultimately I would disagree with this indentation decision. Lining up array members with the array itself loses their very containment within an array. I know the indentation may go deep but it would have been more accurate. (no, I won't let this hit the proficiency because I see the rationale; but I am just registering my disagreement) (4c)
7. Note how the indentation scheme kinda breaks here. (4c)
8. I know that you got your parameters to tween, so having done that, why not play with intermediate values in the keyframes themselves? I can't tell if you just didn't realize that, or if this was a conscious artistic decision. (3b, 4a)
9. \*\*\* I appreciate the credit given here, and the resourcefulness at this alternative way to do backgrounds is noted, but for future reference, *don't just adapt code blindly*, especially if that code shows some bad practices like this does (e.g., [a] top-level scope; [b] function statement; [c] one-line if statement; [d] implicit use of eval). Bring in the code, review it against our best practices, and refactor as necessary so that it doesn't stick out badly. (4a, 4b, 4d)
10. \*\*\* You got the basic tweening functionality done here, but the design is not ideal and will not scale. With this approach, you will be adding more and more code as sprites gain an increasing number of customizable parameters. I mentioned the need for this refactoring in class multiple times. (4b)
11. The arguments to these function calls should be indented inward, because they are "contained" within the function call itself. (4c)
12. \*\*\* Here is another instance where the functionality was accomplished but the design needs improvement. You are effectively overloading the second argument and that just won't scale. Further, it prevents a sprite from having separate *leftLeg*, *land*, and *hand* properties at the same time. You should really have gone with a jQuery-plugin-style options object here, as mentioned in the assignment. (4b)
13. Good, you got the idea for how to make this tweenable. There is one point of improvement: if you applied *legPosition* using a *translate* rather than adding it to every drawing call, you would have needed muuuuch less code. (3b, 4a, 4b)
14. Same idea here; this would have been a *scale*. (3b, 4a, 4b)

# CMSI 371-01

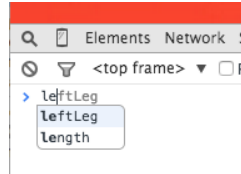
## COMPUTER GRAPHICS

Spring 2015

### Assignment 0217 Feedback

Outcomes that ultimately cover both 2D and 3D max out at | for now because we are dealing only in 2D. They will expand to their full potential with the 3D course work.

15. There is no `var` keyword here! That means all of your variables have top-level scope, as can be seen in the following screenshot: (4b)



16. Fortunately, here you did not forget about the `var` keyword in *this* file. (4b)

1a — +

2a (max |) — | ...The transforms were more manual, but that hits 4b more. We'll see what happens when we go into 3D.

3a (max |) — | ...Library of sprites functionally did what was expected but did not have the best design. That issue was not serious enough to drag you down to a /, but it will keep you from going to a + if your 3D libraries also have that shortcoming.

3b (max |) — |

4a — +

4b — / ...The overall characteristic of the code is that it does the job, but not internally in the best way, and that is reflected here by the seeming contradiction of + in 4a but / in 4b. In fact, this is how a lot of real-world software is—it works, but it could have been designed much better.

4c — + ...Despite the number of 4c notes, in the end they were all fairly small and did not add up enough to make for a ding in the proficiency.

4d — +

4e — +

4f — +