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.

Juan Carrillo *juansc / juanscarrillo23@gmail.com*

Your scene tells a simple, fun story, with action that is very nicely timed and manages some genuine dramatic tension—tough to do in a fully-programmed animated scene. As for the code…

1. *** 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 *cog.js*, *smileyFace.js*, and *snake.js* lying around. Potentially confusing and ultimately not good practice. (*4b*, *4e*)
2. This line is too long—current convention is to have a maximum of 120 characters for a single line of code (it used to be 80, but screens are wider these days **:)**). (*4c*)
3. Some inconsistent spacing here. Not as prevalent as in the previous assignment, at least. Note that, with complex literals or repeated structures, this is bound to happen, so best to format-as-you-go, or have a good autoformatting configuration. (*4c*)
4. Now, this isn't spacing but incorrect indentation. (*4c*)
5. *** A bit of a separation of concerns breakdown here—you certainly got the background function mechanism in there, but for this specific scene you are relying on a background that really does not belong inside the generalized keyframe tweener object. This function should be *outside* of the generalized library and passed as part of the options. (*4b*)
6. Nice, clean structure you have here. Good scaffolding for the tweening code. (*4b*)
7. This code is functionally OK, but unusually tight. Not sure why…it seems that, in the middle of concentrating on getting this function to work right, you forgot about proper presentation. (*4c*)
8. *** This is a common pattern that has a pretty elegant solution in JavaScript. When you have multiple possibilities based on the value of a single string, consider another helper object whose properties are those very values, associated with functions that do the desired work. Look at the block comment below the note to see a sketch of the code. Note how that structure is very easy to extend and revise, and keeps the central logic—which is that you are tweening some property—clear and uncluttered. (*4b*)
9. Why is `setInterval` here? I suspect you might be using an older version of the sample code. The latest code uses the now-recommended `requestAnimationFrame`. (*4b*)
10. This is another separation of concerns "leak." In a sense, the tweening library should not enforce which properties are which; instead, this should be passed into it from the caller. That way, your sprites have complete freedom, and you can designate new subproperty objects in the future without having to change the core library. (*4b*)
11. This is a leak as well. As a rule, if a value can vary based on the content of the program (not its logic), then that value should not be hardcoded into the general library. Pass it as an option. (*4b*)

1a — **+**

2a (max **|**) — **|**

3a (max **|**) — **|**

3b (max **|**) — **|**

4a — **+** …Very rich tweening functionality here, some beyond the expectations of the assignment.

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.

4b — / …However, although the code does its job, internally it does not separate concerns very well, 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. Strive to write code that is both functional *and* well-designed.

4c — | …Code is presented better than in the previous assignment, but still with enough hiccups to be noticed. Incorrect indentation sticks out in particular, as well as odd swaths of code where the presentation seems to break down. Your task here is definitely to start learning how to code immediately in a manner that reflects its structure and intent, so that you are not worried about going back to it later. You should also consider some automation.

4d — +

4e — +

4f — +