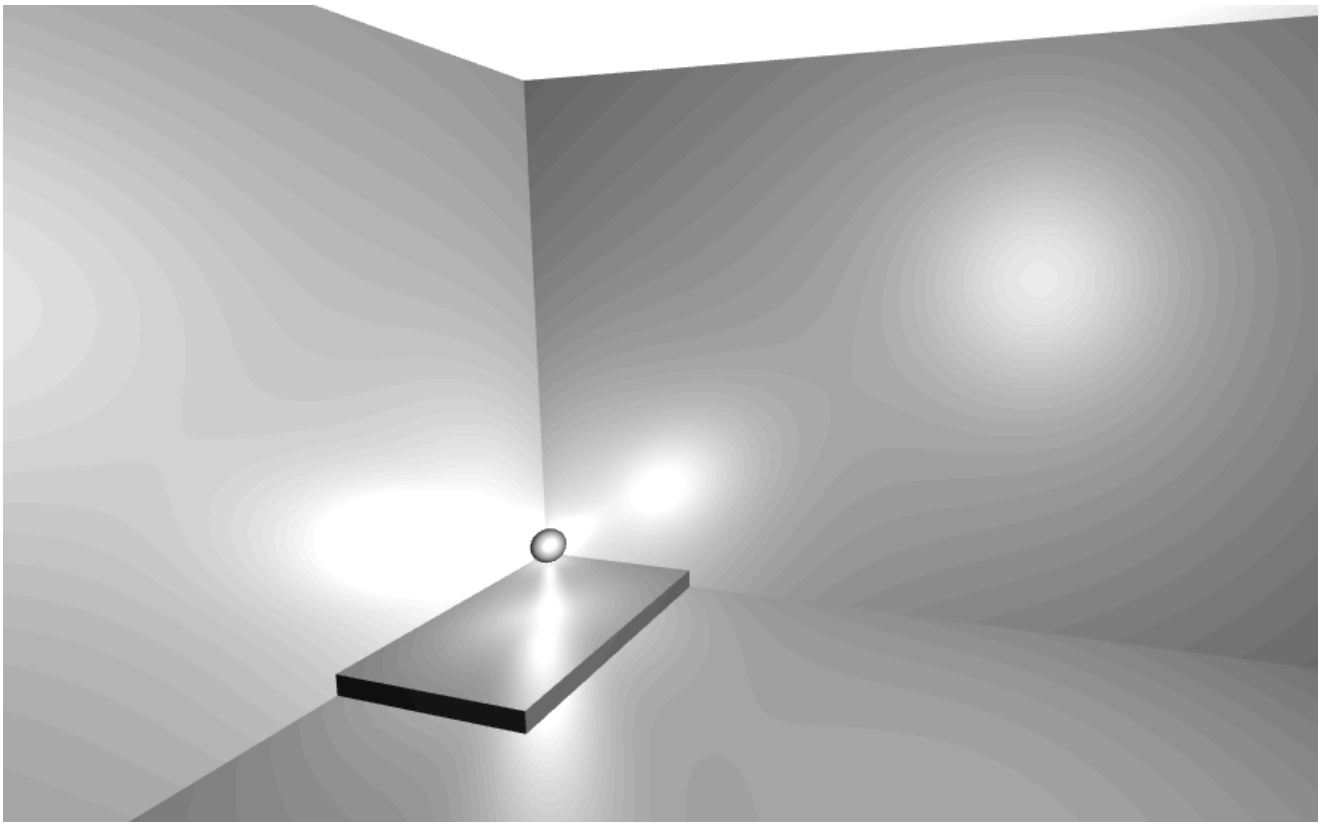
cgviz assignment 1

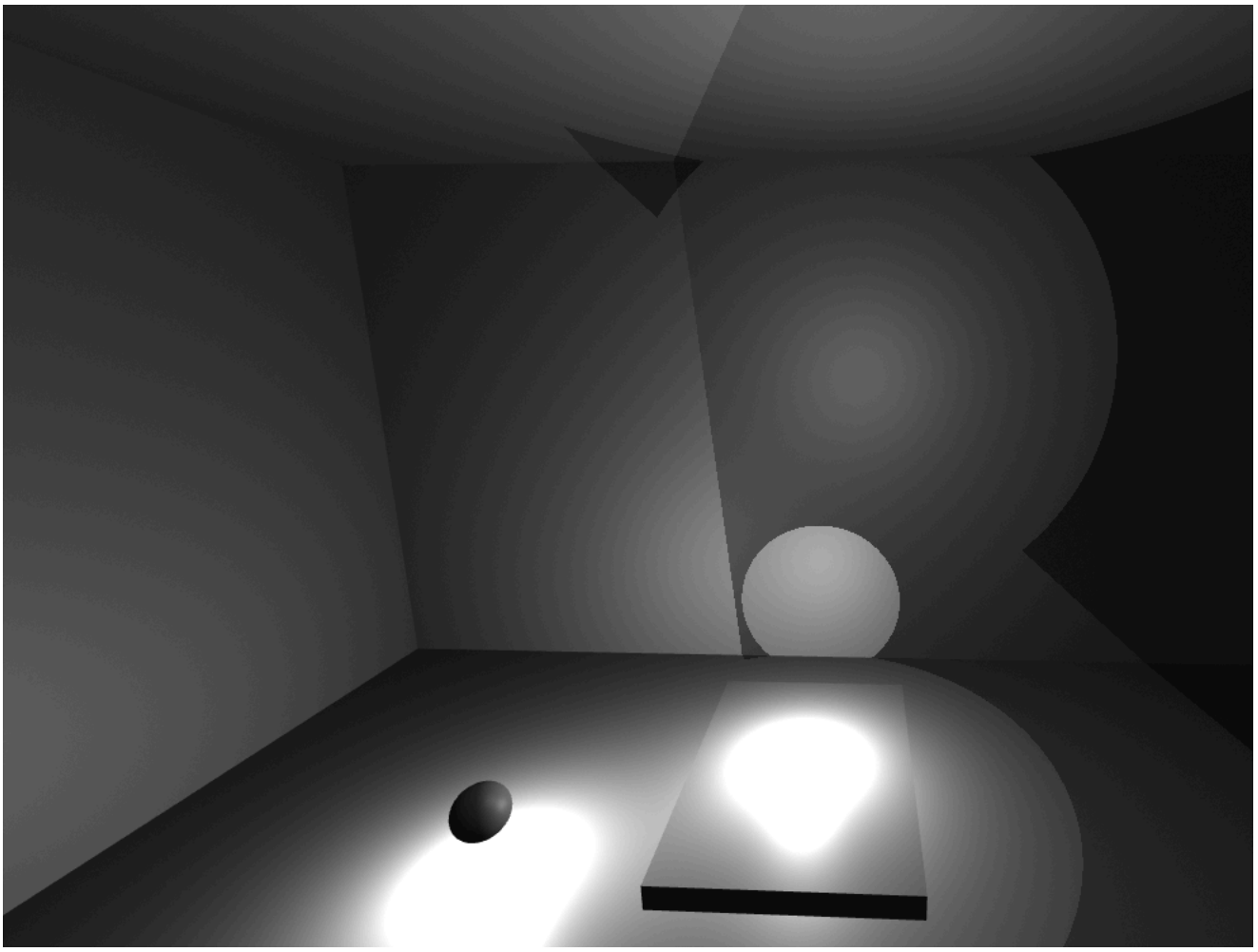
report

task 1



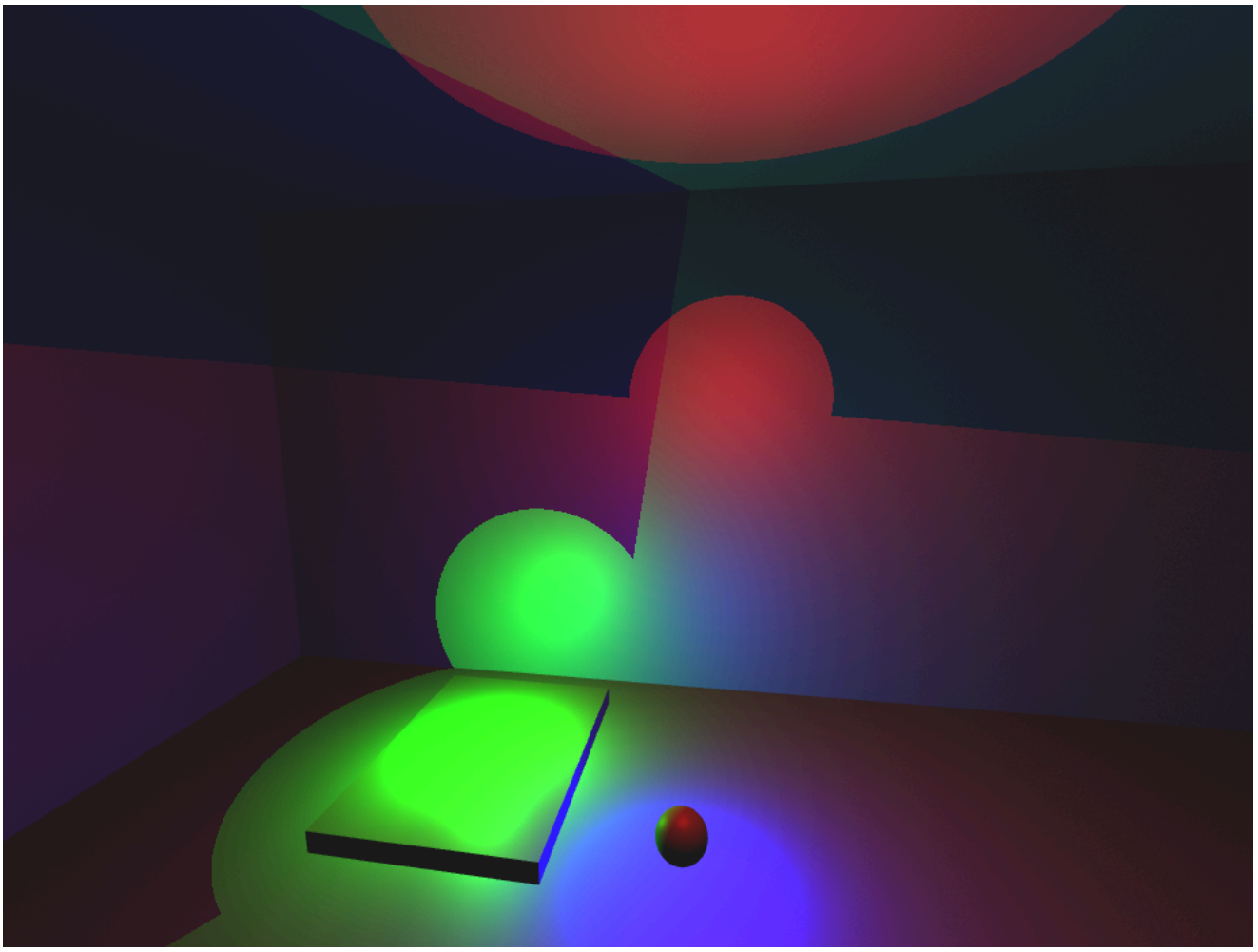
i chose to place three lights in the scene. one in the ceiling, one inside the ball and one inside the pad. using the scene graph, i attach the ball and pad lights as a child to the ball and pad, so that they follow suit.
having the ball light inside the ball causes exceptional shadow calculations later.

task 2



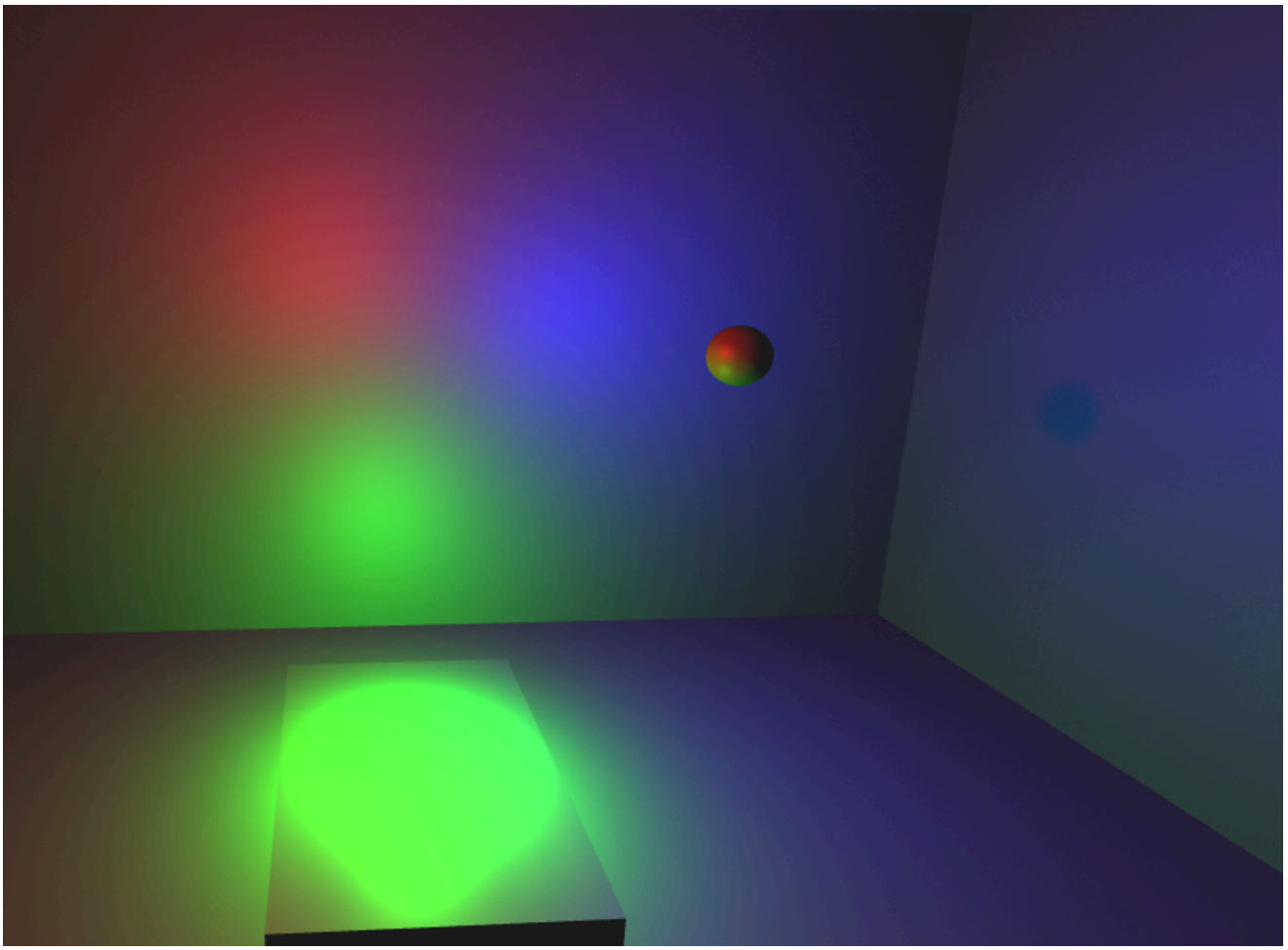
note: these shadows are kinda wrong. i fix them in task 4. the shadow logic was kind of backwards, causing this effect which is almost interesting.

task 3

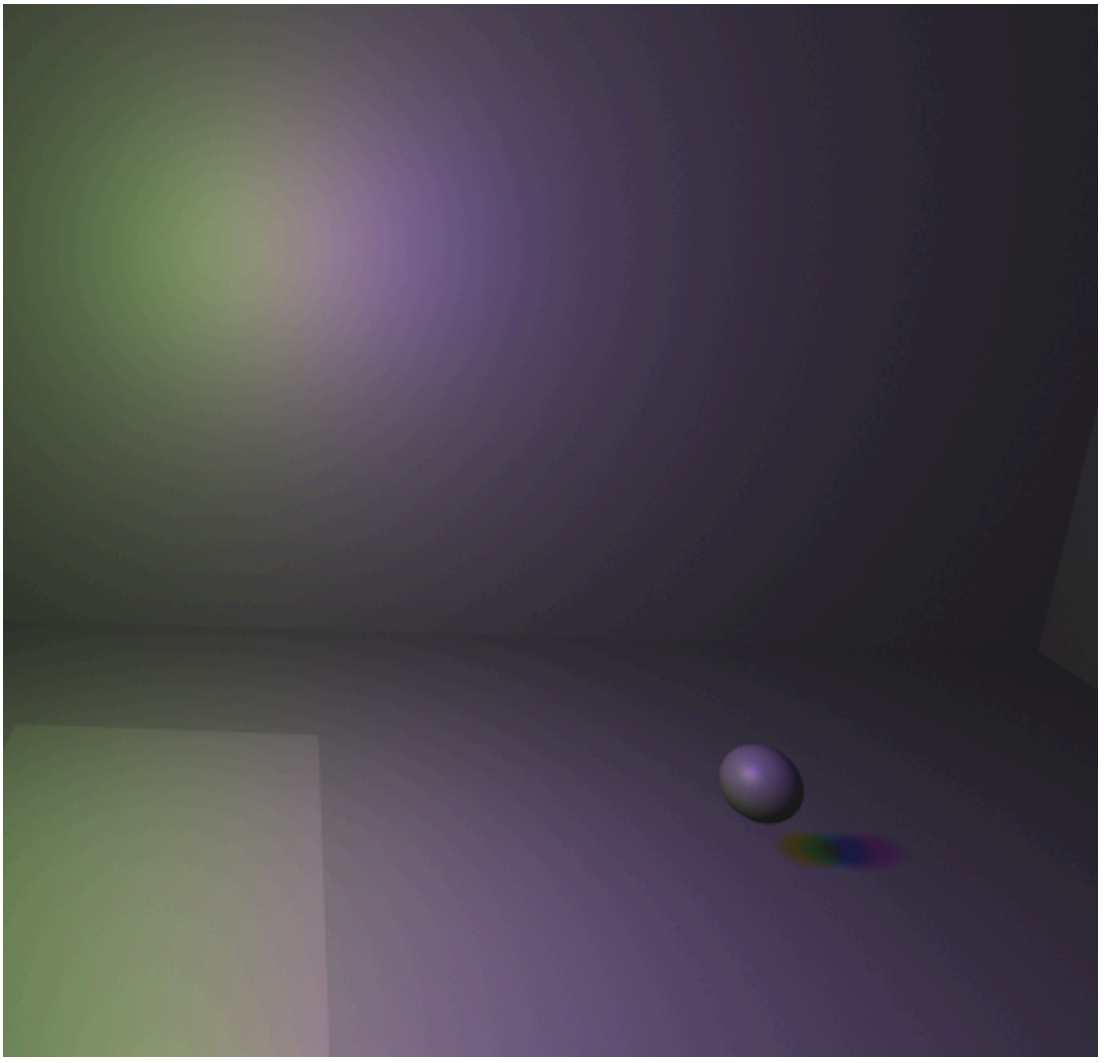


here is the same weird shadow, but with colors!

task 4



shadows are now fixed with proper shadows with blurred edges.



i placed the lights statically in the room, giving a chance to better see the shadows mix. i could've fiddled more with their placements, but i'm content with this.

ai usage & reflection

i'm sharing my prompts.

https://kagi.com/assistant/079b3a17-b318-4496-9912-ecc373e2a9e0

i started the assignment late due to having been quite busy, so i felt the need to rush with the use of ai as an aid. however, i think it might have caused equally many headaches, as evident from the length of the above thread.

it worked well for giving me correct (seemingly) equations and implementations of phong, since it is such a widely used illumination technique. the issue lay more in that the ai had no overview over my whole code-base (i haven't tried any of the new "agent modes" that are all the rage these days), thus it couldn't properly debug. i pulled my hair out for a long while, trying to figure out why my normals were all messed up, causing black walls. the issue lay in a completely different file, namely in SceneNode, which i had not given the ai for debugging. thus i went in circles for a long time trying to scour the GameLogic and fragment shader scripts for any bugs, only for my ai assistant to blabber about non-existant problems -- though it did on some occasions find pretty terrible bugs i had overlooked.

in short, ai helped debug when it had access to the right files where the bugs lay. without access to those modifications i had made that introduced a bug, it would only look for bugs in the handout code, which is wholly unproductive.

after task 1, it wasn't that useful anymore, as i felt the scaffolding we had built and the understanding i had (re)gained, as well as the straight-forward formulations of the later tasks, allowed me to make the necessary implementations without any ai help. by far the greatest challenge was getting phong hooked up to begin with.

if i had slowed down, and instead just tackled task 1 without ai, only maybe consulting it to refresh my knowledge on phong etc, then i could've avoided some headaches, but i am not sure it would have allowed me to be more productive in the sense of completing the assignment earlier and with a greater understanding of the code and phong in general.