

Project 4: WebGL 3D Project

Samuel B. Scalf

School of Cybersecurity and Information Technology, University of Maryland Global Campus

CMSC 405: Computer Graphics

Dr. Cynthia V. Marcello

8 March 2022

Test Cases

Table 1*Test Cases for WebGL-HTML Controls*

Control	Function	Expected Output	Actual Output	Pass/Fail
Animation Checkbox	Toggles Animation	Animation should play/pause	Animation played and paused	Pass
Speed Slider	Adjusts the speed of the animation	The speed of the animation should change	The animation speed changes, but sometimes has a rewind/fast-forward effect	Pass
Scale Slider	Adjusts the scale of the objects (zoom in/out)	The scale of the objects changes to match the slider.	While the animation is running objects scale smoothly, but only update when mouse is released otherwise.	Pass
Sunlight Checkbox	Turns off the sunlight	The sun should no longer emit light but should still be visible. Objects can block remaining light.	The sun no longer emits light and is still visible. Objects can block remaining light.	Pass

Note. The effects of the controls are persistent, except in the case of a page refresh. The effects of the Speed Slider control are only apparent if the Animation Checkbox is checked, meaning the animation is running. Changing the position of the Speed Slider while the animation is running can cause a rewind or fast-forward effect due to the speed being multiplied to the frame number. The effects of the Scale Slider are only apparent after releasing the mouse button, unless the animation is running, in which case the effects are immediate.

Source Code

The source code for this project contains four JavaScript files and one HTML file: `basic-object-models-IFS.js`, `gl-matrix-min.js`, `trackball-rotator.js`, `scalf-homework4.js`, and `scalf-homework4.html`. The first three JavaScript files were included in the example files and have been unchanged. The final two files were originally extracted from the provided “`diskplay.html`” file, with the main script data being moved to `scalf-homework4.js`. This extraction was necessary to change all “`var`” keywords to “`let`.” The original methods for displaying lights, the soccer ball, and the box man, were all replaced by my own methods and helper methods. Some additional global variables were added, such as `solarSpeed` and `solarScale` so multiple methods could access their values.

Only one method was used to generate each of the planets, using parameters to modify their attributes. The asteroid belt was by far the most difficult to create, with each asteroid being its own sphere in a random location. Finding a random location within a torus was proving difficult. It was less difficult to get a random location within a sphere, so that method was used, then duplicated around the orbit, creating a torus effect. Since the points for each asteroid within their respective spheres was calculated independently, using the polar coordinates for the sphere, it is rare for two asteroids to be in the exact same location.

Screen Captures

Due to the nature of animations being incompatible with still images or screen captures, testing the actual animation and speed of animation must be observed directly. Regardless, some screen captures were taken to display different times or states of the program. Also note the clear color is not purely black, but a very dark grey to allow the user to see “shadows” of objects in the dark. The first screen capture was taken to display the top of the HTML page but was not necessary for the rest of the captures. All other screen captures will display the entire WebGL Canvas. A final screen capture shows the footer of the HTML page.

Figure 1

Screen Capture of Top of Page

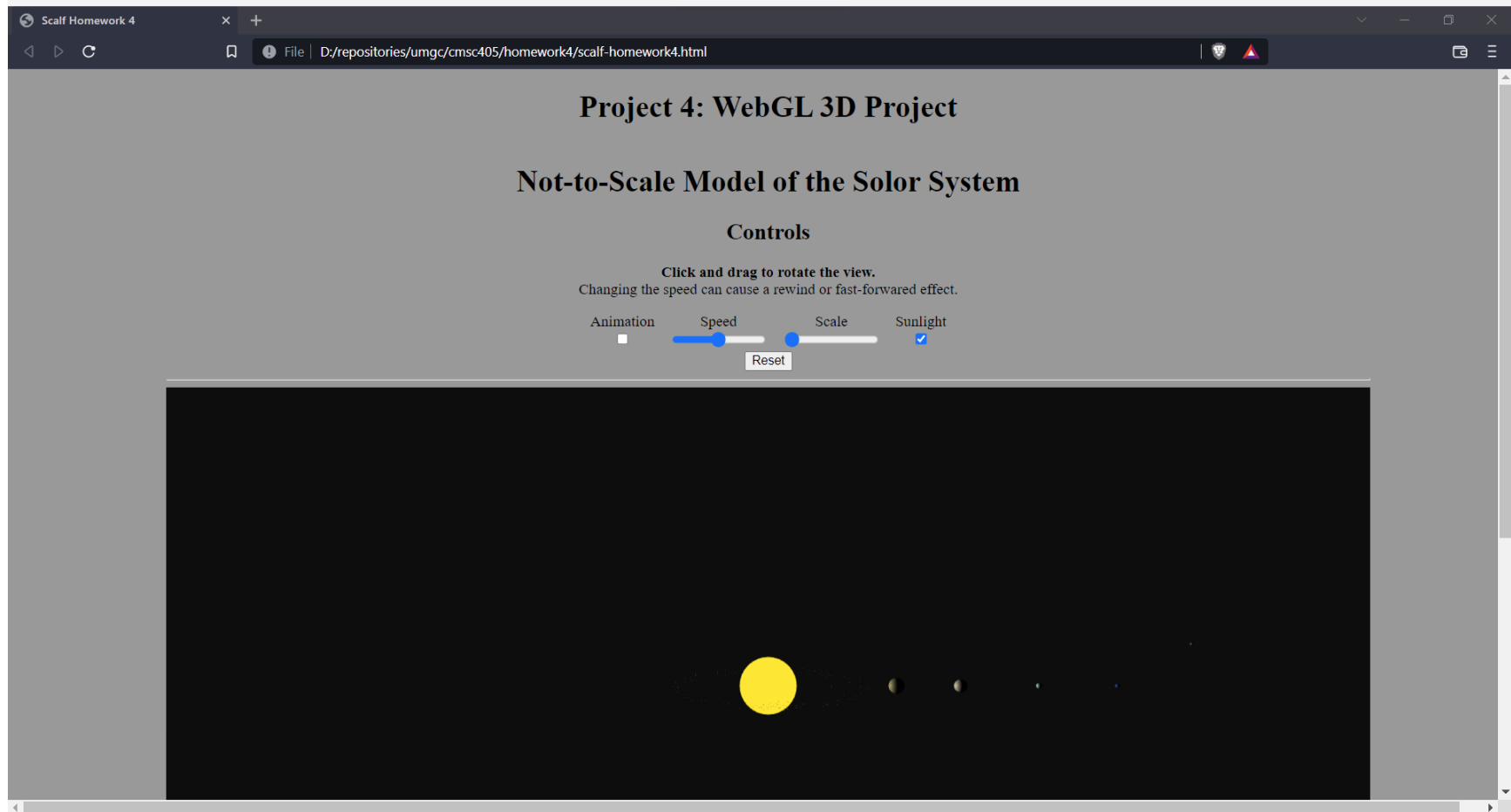


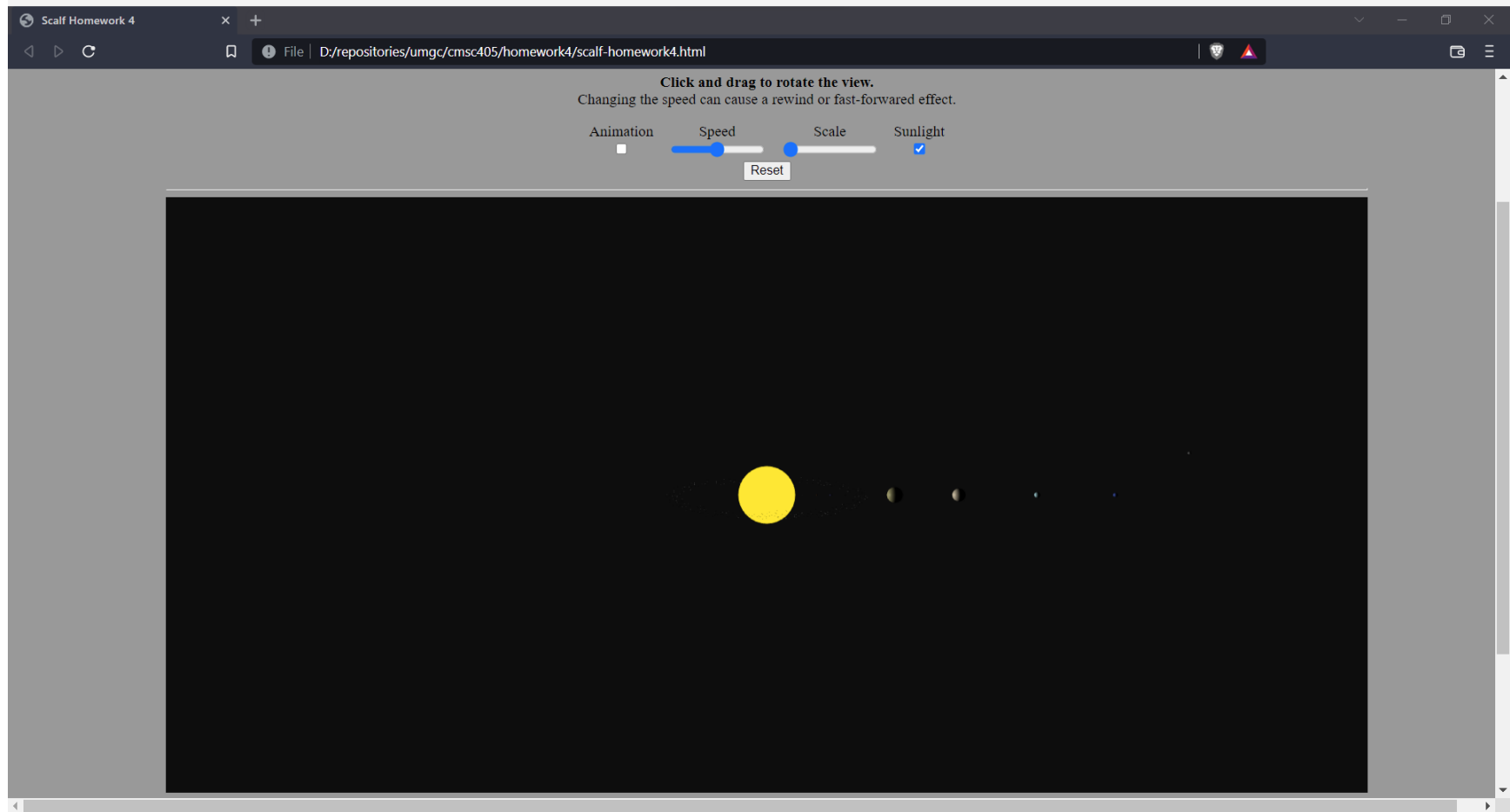
Figure 2*Screen Capture of Starting Point*

Figure 3

Screen Capture of Sunlight Turned Off (No Animation)

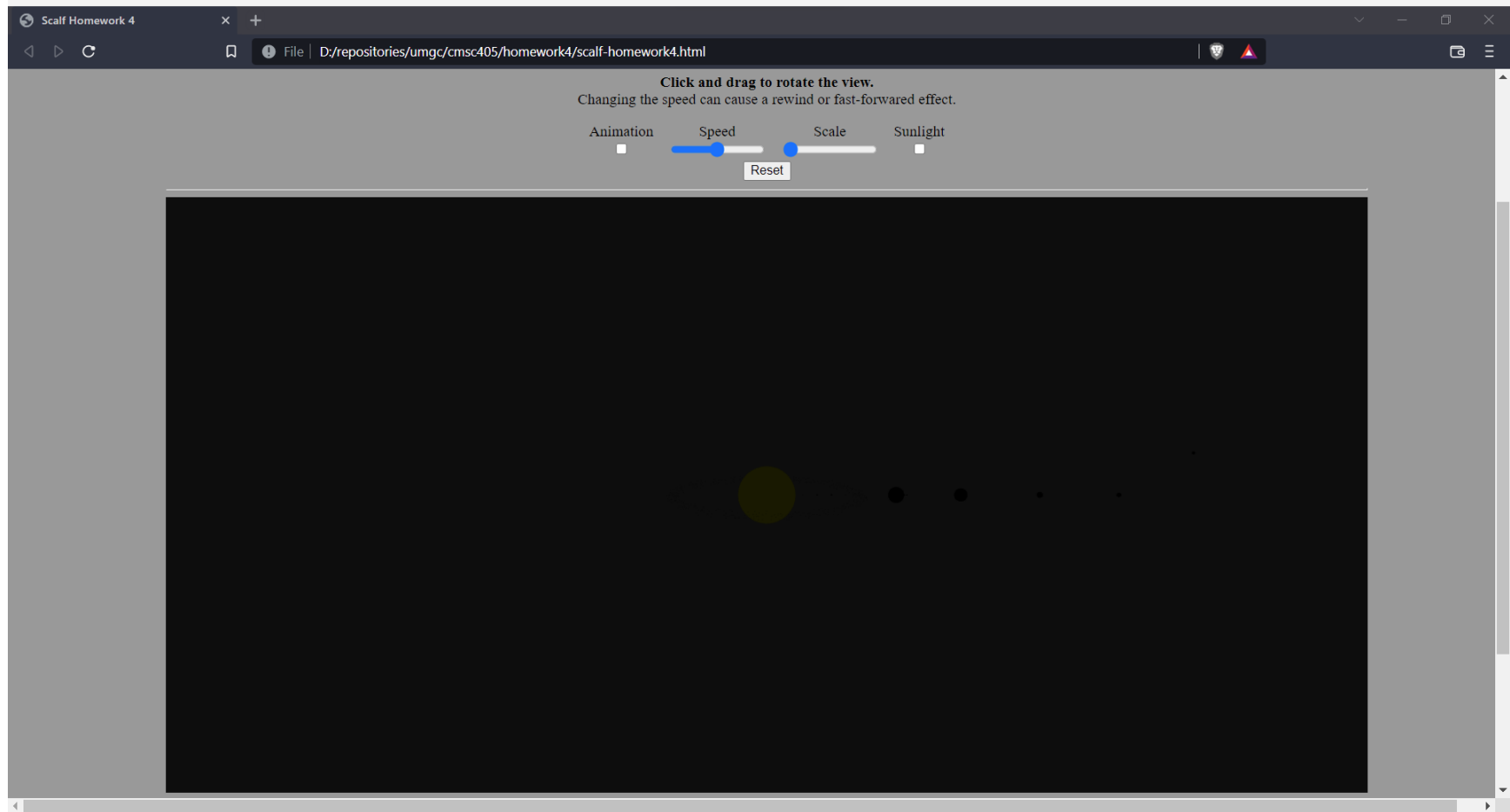


Figure 4

Screen Capture of Scale Slider at Halfway Point

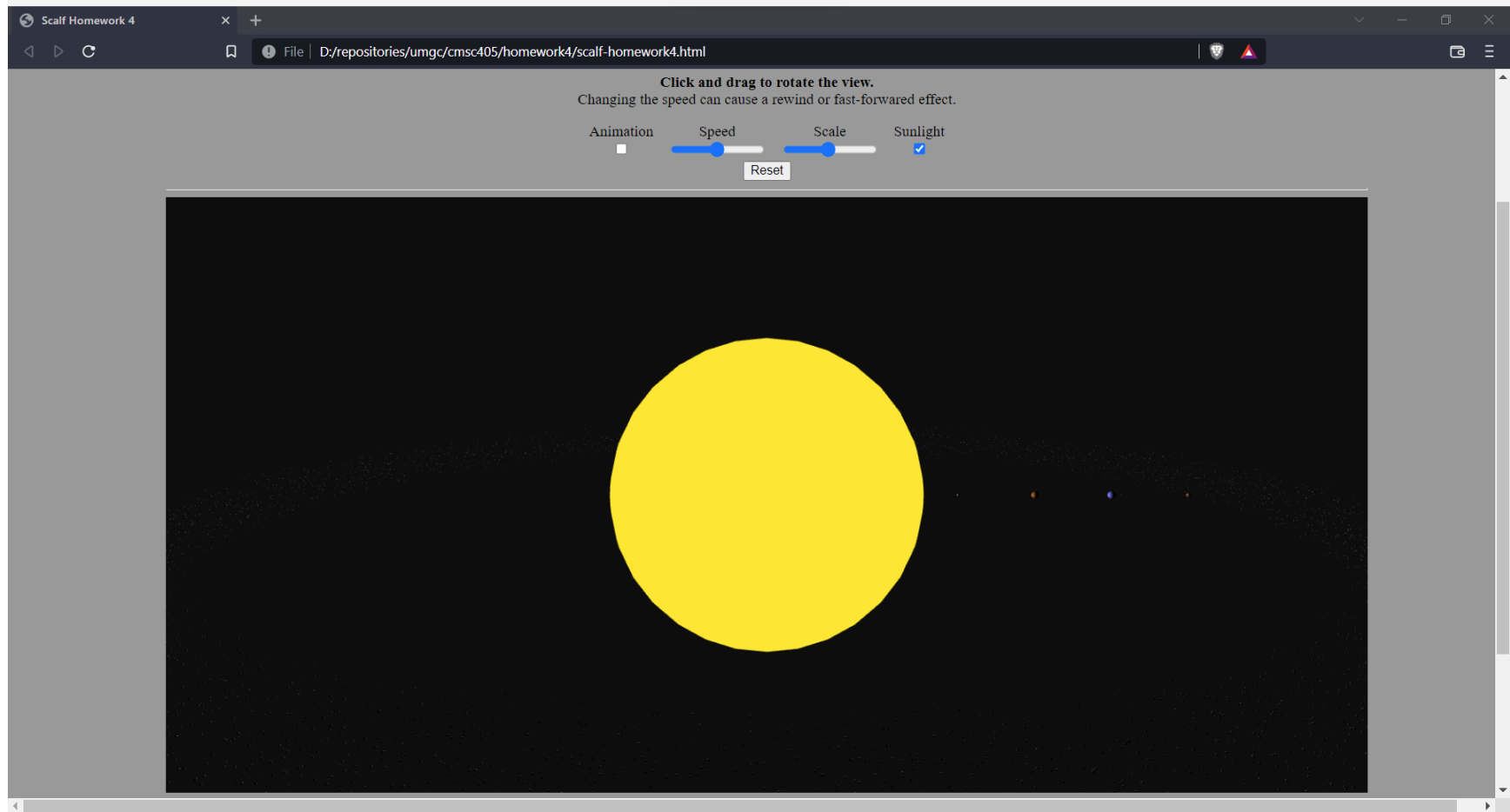
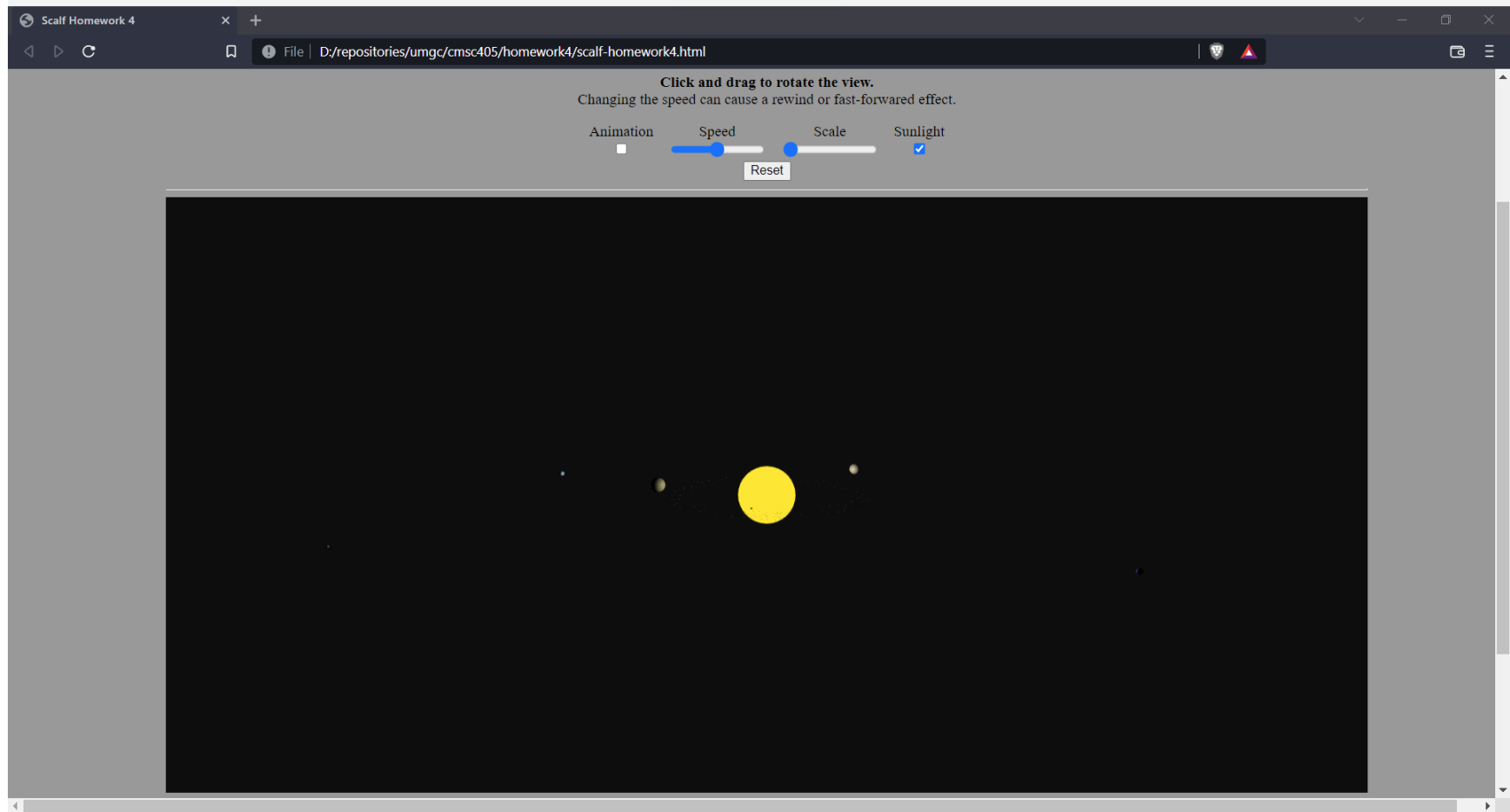


Figure 5

Screen Capture of Scale Slider at Maximum



Figure 6*Screen Capture at Default Scale After Animation*

Note. Although faint, towards the left side of the screen capture, Pluto can be seen. Neptune is opposite, on the right, giving off a deep blue reflection.

Figure 7

Screen Capture at Medium Scale After Animation

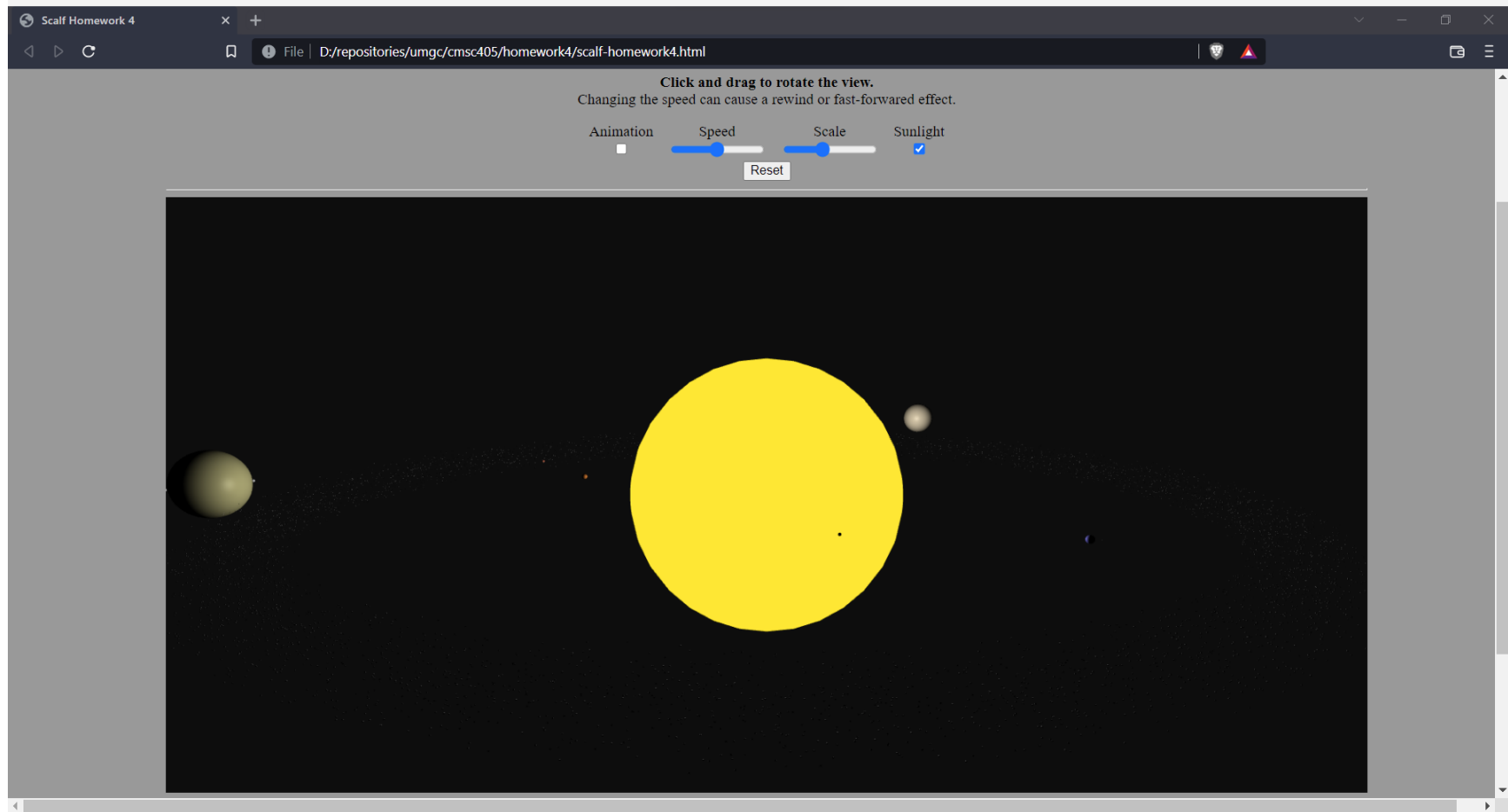
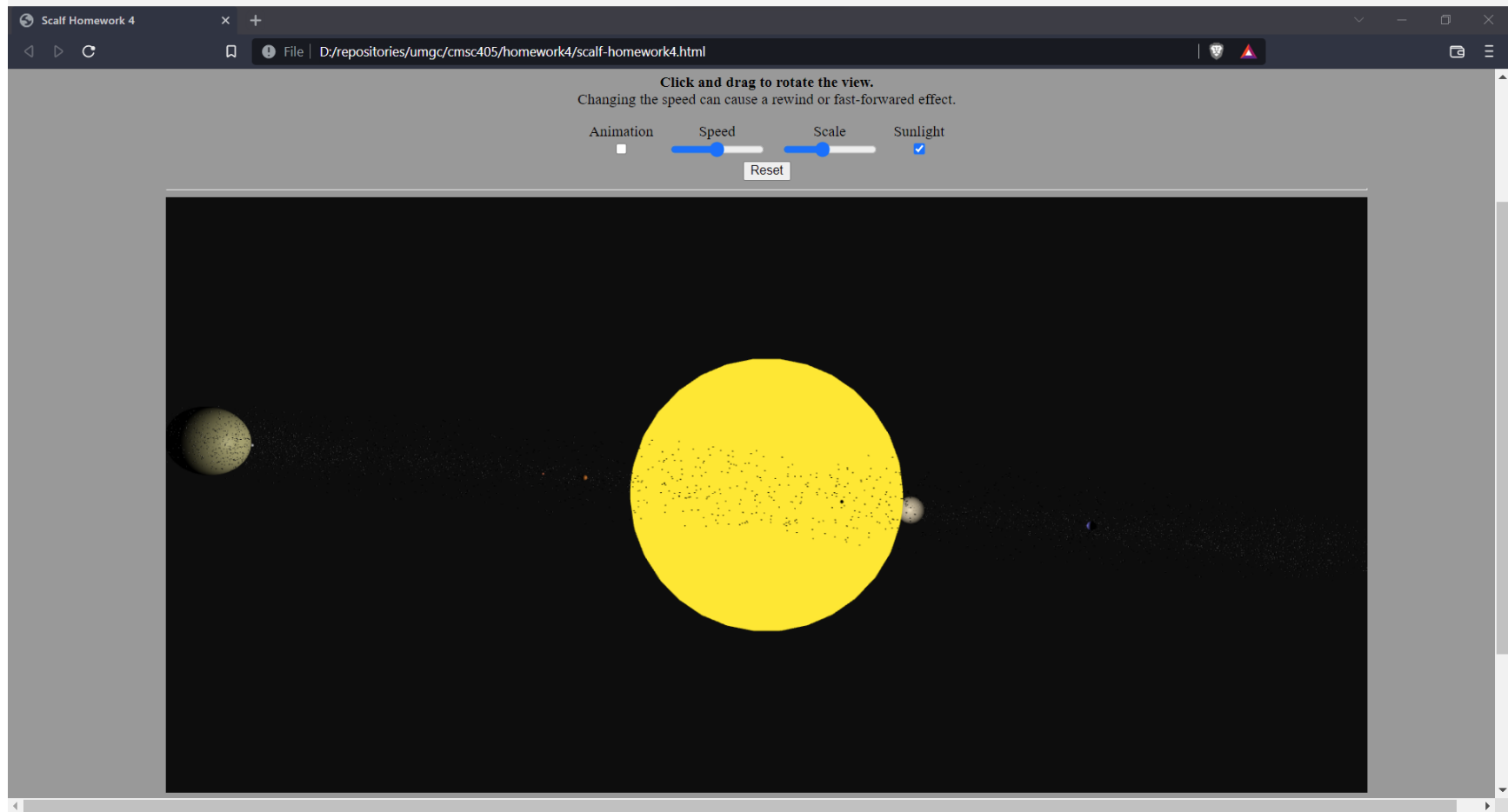


Figure 8

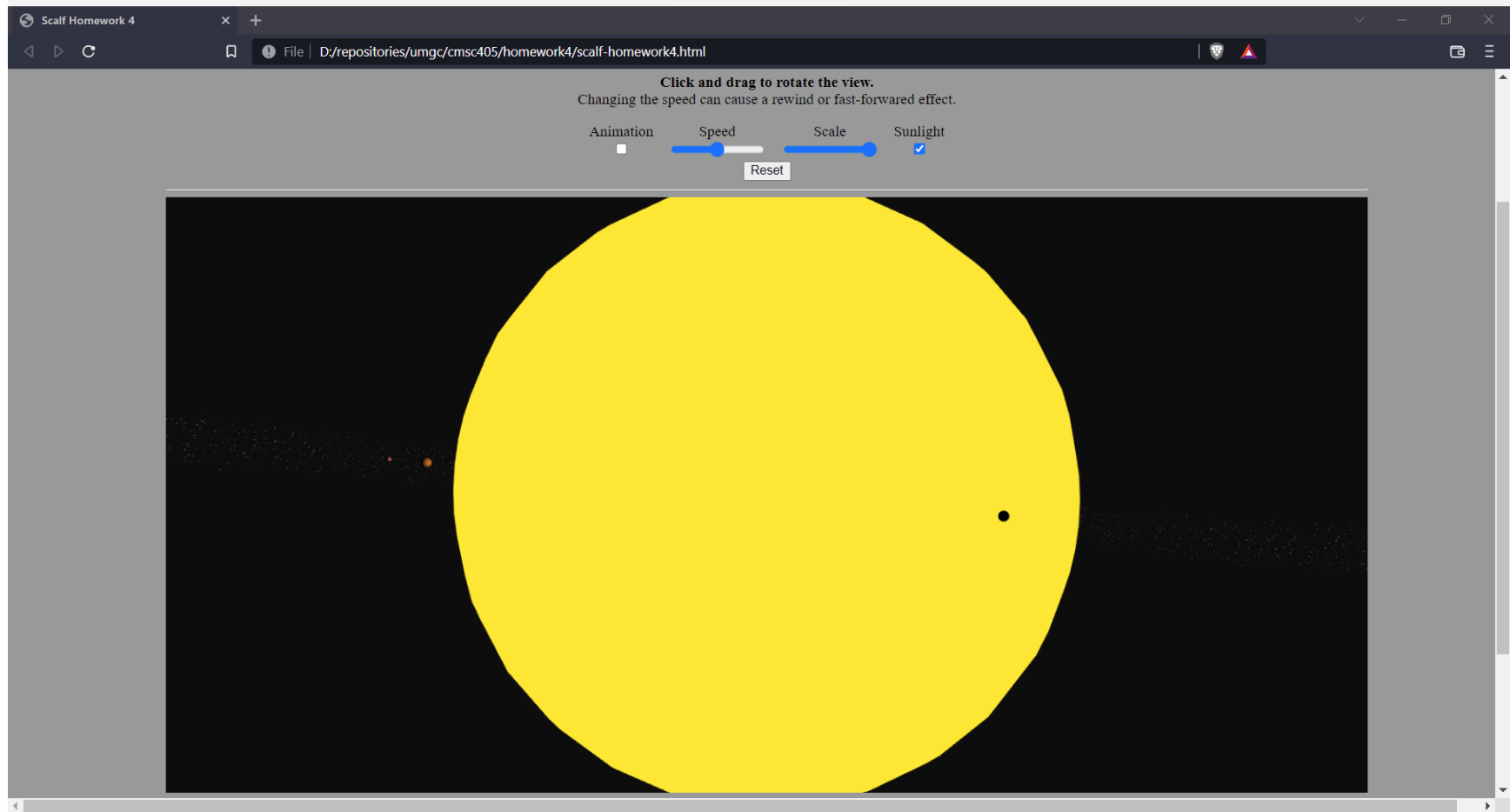
Screen Capture at Medium Scale After Animation (Rotated)



Note. The view is the same as Figure 7, but rotated along the viewer's x-axis to better show the asteroid belt against the Sun.

Figure 9

Screen Capture at Maximum Scale After Animation



Note. The spot on the Sun is Mercury. To the left of the Sun, Venus and Mars can be seen against the asteroid belt.

Figure 10

Screen Capture of Page Footer

