

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

# Simulating Rotation and Gravity

Reed LeBlanc



# Original Project Idea

Entirely different:

Originally was going to build an animated wallpaper  
/background to use on a resume website.

Ended up having fun with replicating gravity  
and object movement/rotation.



# What I built Instead

Created a visual representation of tire movement approaching viewer with attempt to simulate realism involving:

Tire shading through directional light source to represent sun's location

Tire rotation/spin changes relevant to direction heading

Wobble / non rigid positioning

Bouncing effect on rubber tire involving:

- Vertical change

- Hard direction shift at lowest Y position (road/ground)

- Vertical pace slows at specific height cap emulating effect of gravity



# Things I could add

Would like to continue to work on this and add more objects in the scene i.e. something for the tire to run over/break as it travels towards the viewer

Maybe change the background to original with an added broken glass effect as the tire reaches and hits the camera





# Biggest Surprise

Originally wrote this on desktop, fine tuned directional values to convey proper speed.

With identical code on laptop, because of hardware differences, the whole scene and all movements were rendered at a much slower pace:

On desktop, the tire would reach camera position at around 7 seconds

On laptop, the tire would take around 15 seconds to traverse the scene and all movements were slower, making it look like slow motion/the moon's gravity and taking away from the desired effect.

Had to adjust almost all directional values to recreate original visual effect.

Goes to show how difficult optimization must be for developers catering to all types of user hardware.