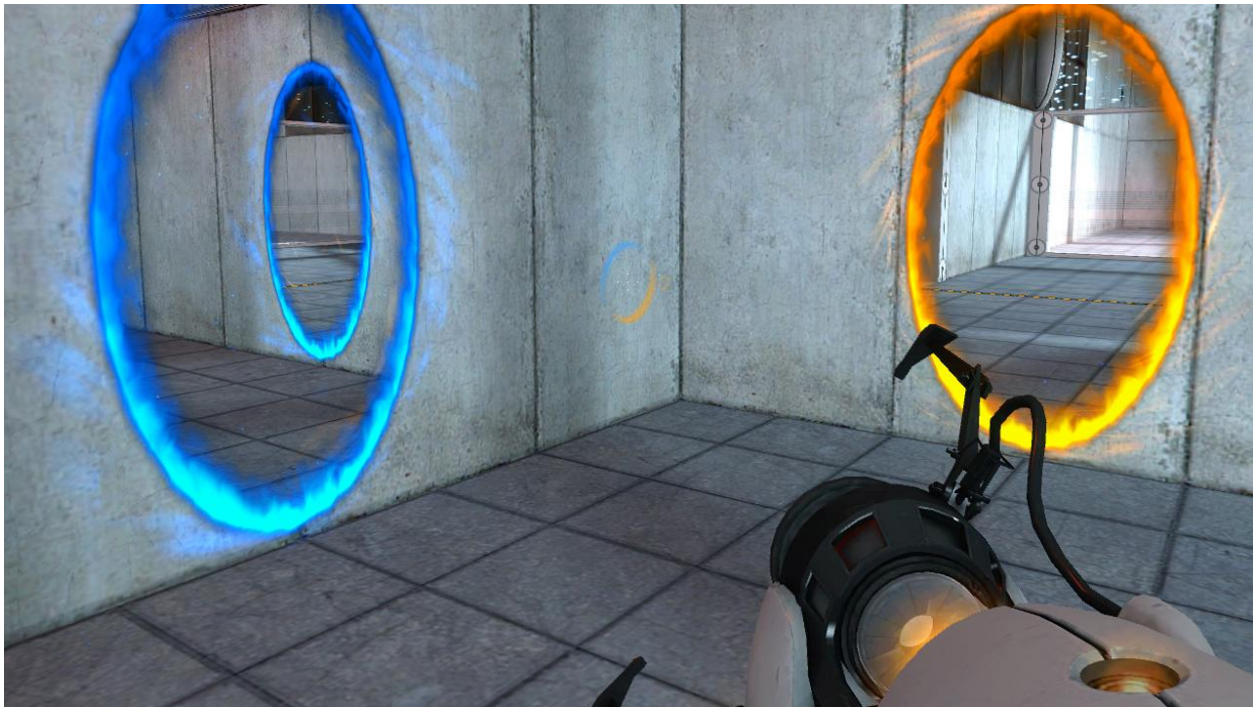
Rendering “Portal”

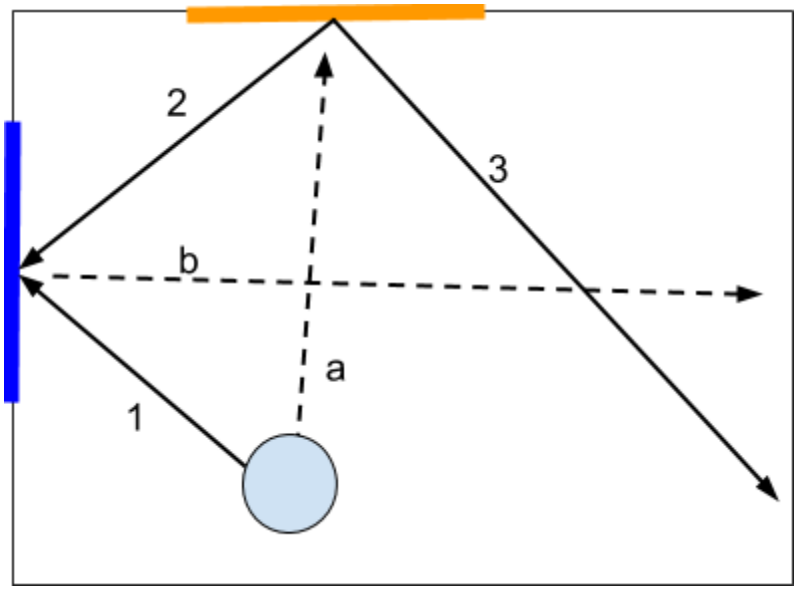
Torin Rudeen (torinmr)

Description

For my project, I am going to render an effect made famous by the video game “Portal.” In that game, players possess a “Portal Gun,” which can be shot to create a blue or orange portal on nearly any surface in the game. The two colors of portal are linked, so that when light or an object passes through the orange portal, it will come out of the blue portal, and vice versa.



This screenshot from the game shows how this effect allows players to see through portals to what is behind them, even recursively. The below diagram shows how the player’s view ray propagates through the blue (solid lines, 1-2-3) and orange (dotted lines, a-b) portals:



For my project, I will use a rasterization-based approach using OpenGL. I will create a simple interior scene using freely available assets, and then add the ability to create portals on the walls, ceiling, and floors, and to see through these portals as shown in the example above. I will focus solely on the graphics-related portions, so I will not implement collision detection, or moving through portals, but simply allow the user to "fly around."

Why is it technically challenging?

This sort of effect is simple to implement in ray traced graphics, but difficult in rasterized graphics. It has some of the same challenges as rendering mirrors, but with additional wrinkles, such as the fact that the view ray can emerge from a different surface than the one it entered, and the fact that portals can be dynamically placed and removed. An additional challenge (shared with mirrors) is that one can look through portals recursively.

Inspiration

My inspiration was, of course, the video game Portal. While I haven't been able to find any papers on this rendering issue, I will be looking at papers on mirror rendering, as well as some blog posts such as (https://th0mas.nl/2013/05/19/rendering-recursive-portals-with-opengl/) which I will be using as a starting point for my work.

Milestones

1. Able to render and move about an interior scene without portals - all non-portal-related work complete. Papers/blog posts on mirror/portal rendering understood.
2. Ability to place portals and see through them, but not recursively.
3. Ability to see through portals recursively.