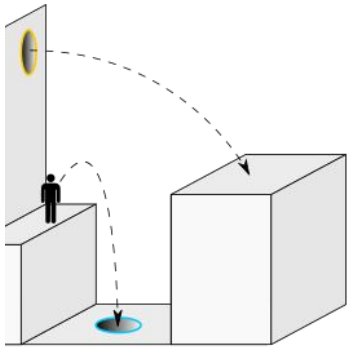
Portal 3

Thomas Kamm, Sunny Li, Edgar Baudry

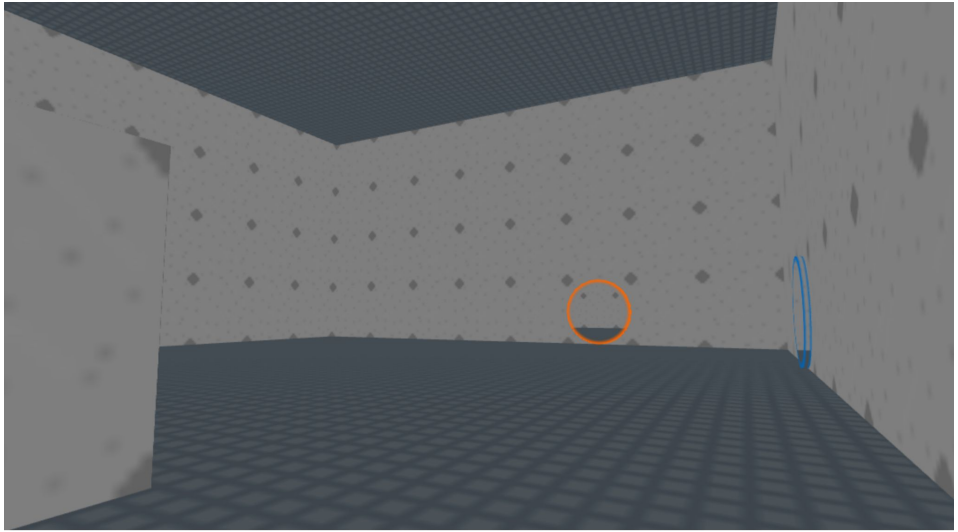
Overview

- Simple clone of the popular game ‘Portal’
- Players can create two portals on various surfaces with the ability to see through and move between the portals.
- Players will operate in a basic environment

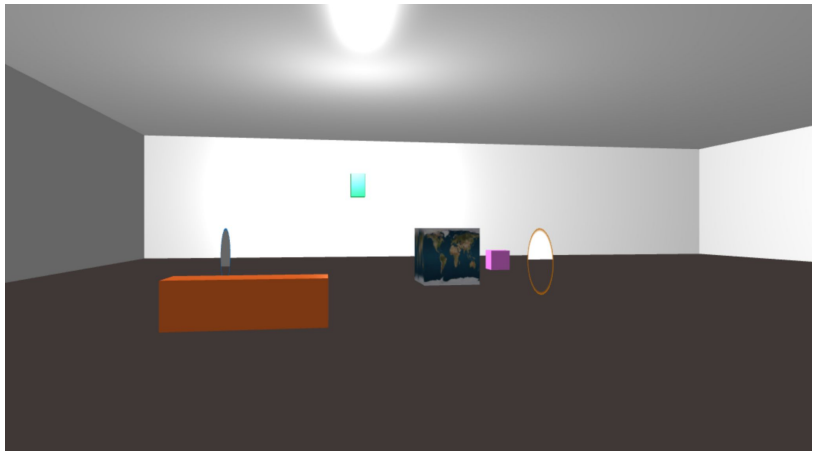


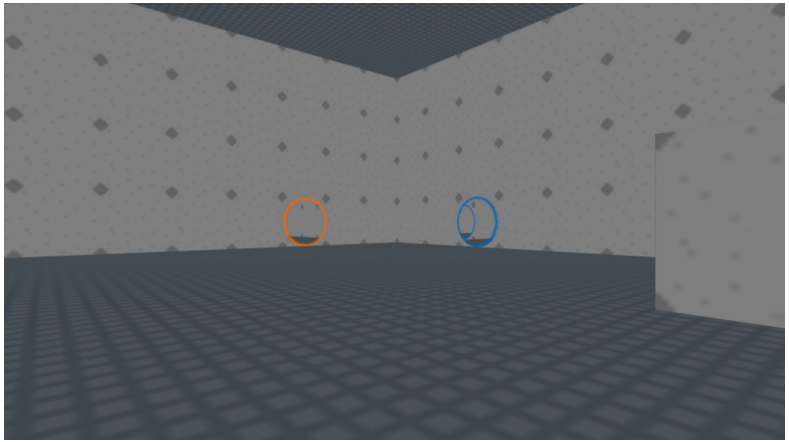
Current Features - Environment

- Player can move around in a multi-level, fully walled space
- Environment is readily changeable
- All walls stored in multidimensional array according to the walls' spatial coordinates, allowing for quick lookup



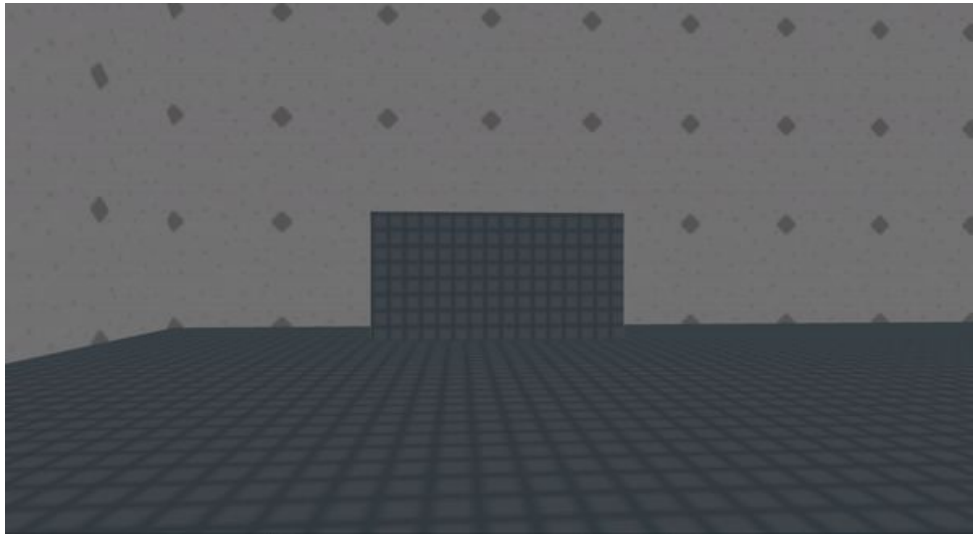
Update Since Midpoint



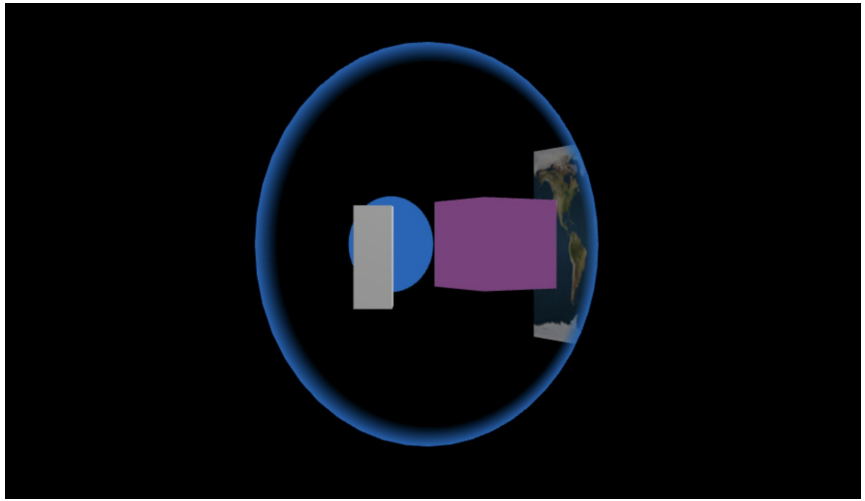


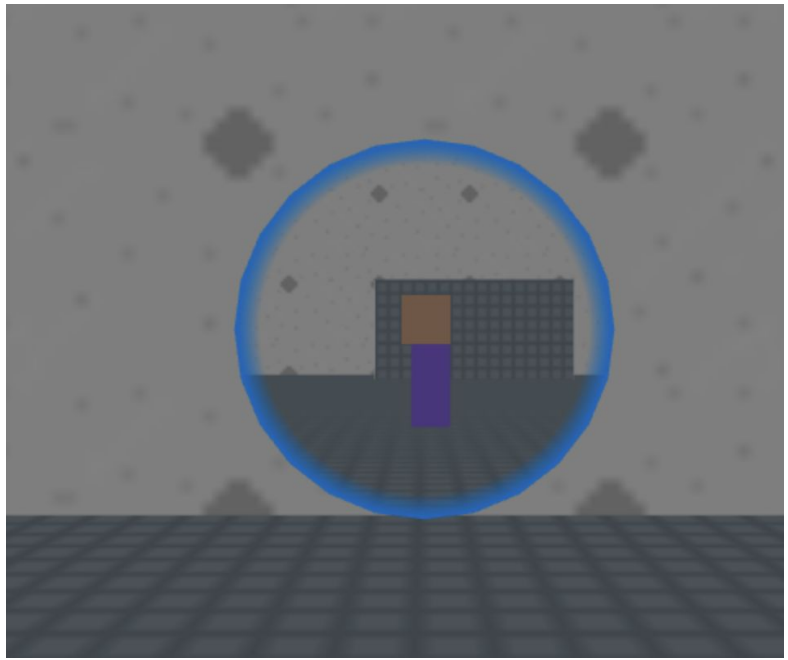
Current Features - Player movement, model, jumping & gravity

- Basic movement with WASD
- Look around with mouse movement
- Player can jump (no double jumping) and fall according to gravity
- Player confined to environment



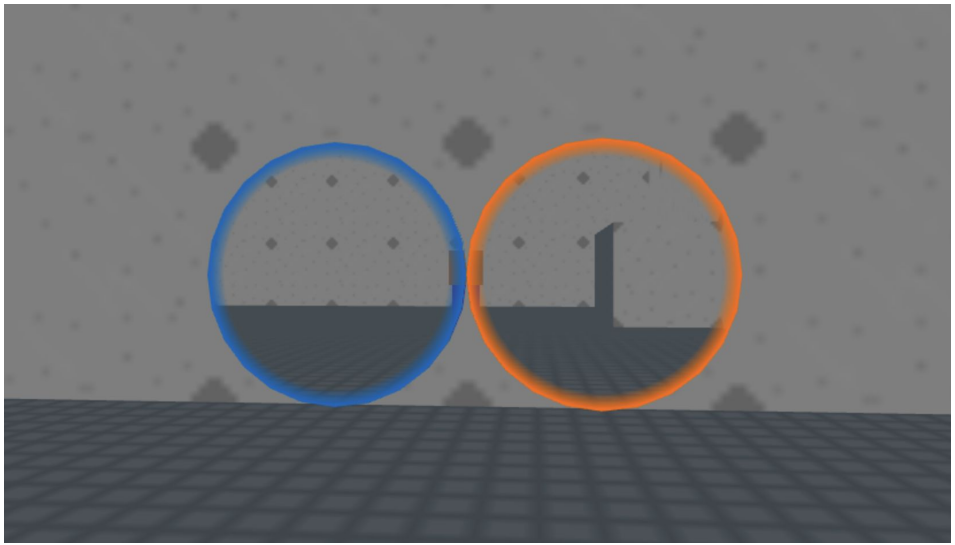
Update Since Midpoint: Player Model





Current Features - Portals, Portal Movement

- Blue and orange portal in environment
- Custom shader and texture gives portals lit outer ring
-
- Portal projectiles can move portals around the environment

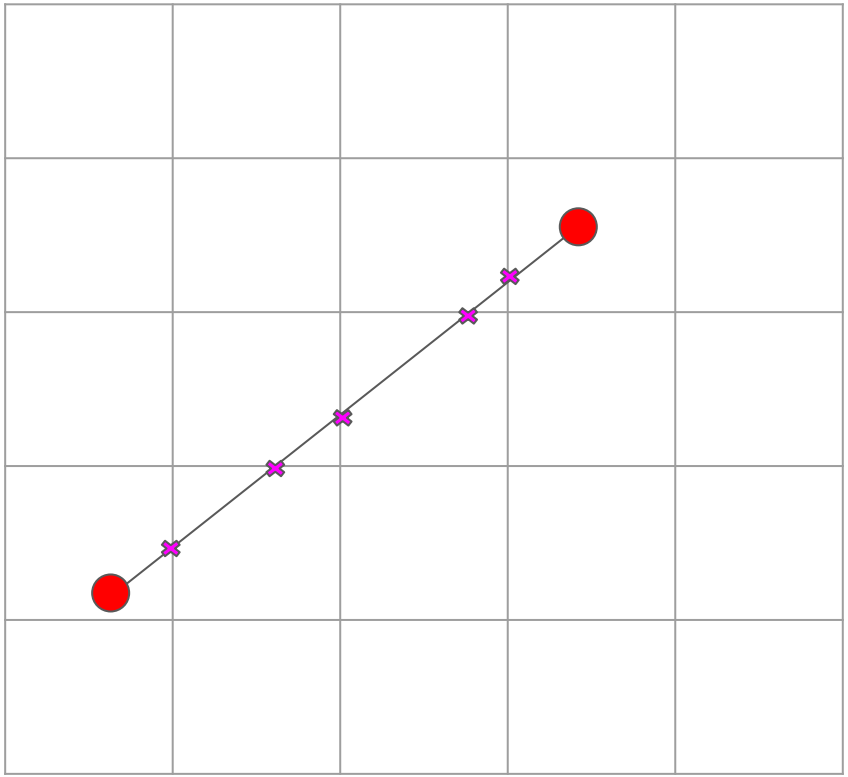


Current Features - Portal Projectiles

- Player can shoot projectiles (orange or blue), which travel in a straight line in the direction the player is looking
- Disappear after max lifetime or wall collision
- Upon touching a wall, will cause the corresponding portal to move to that wall

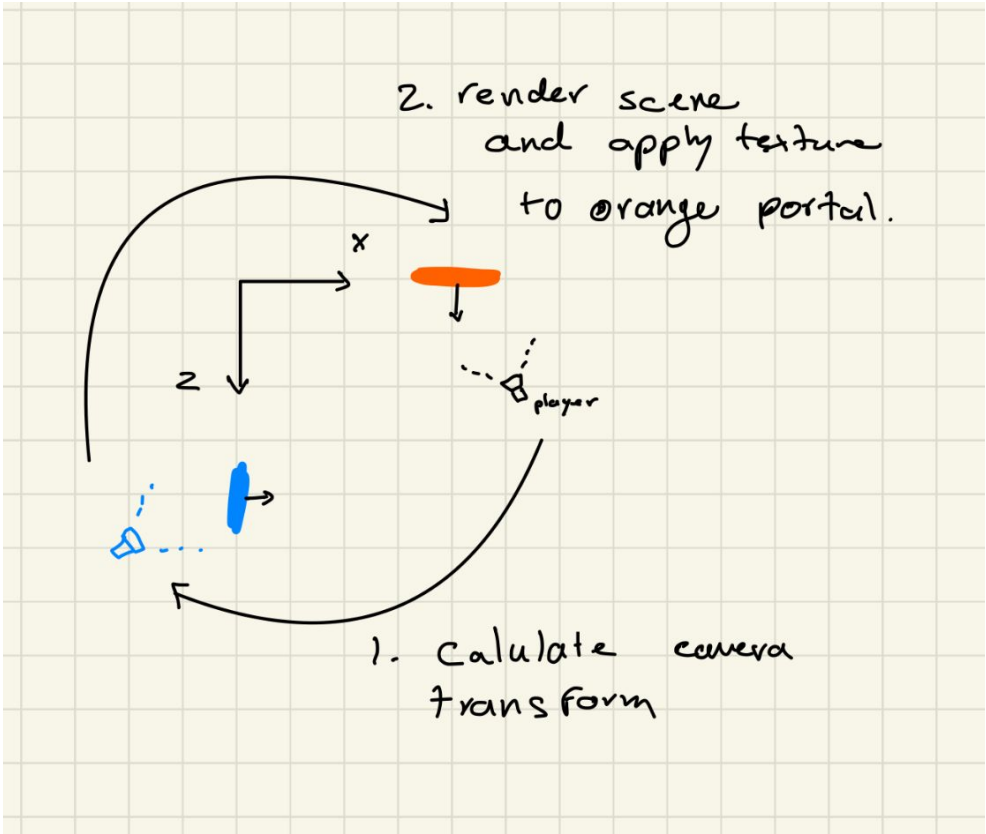
Current Features - Collision Detection

- Collision detection between
  - Projectiles and walls
  - Player model and environment surfaces
  - Player and portals
- Point-based collision detection
- Since wall placement is discreet, could create $O(n^{1/3})$ algorithm for most collisions.



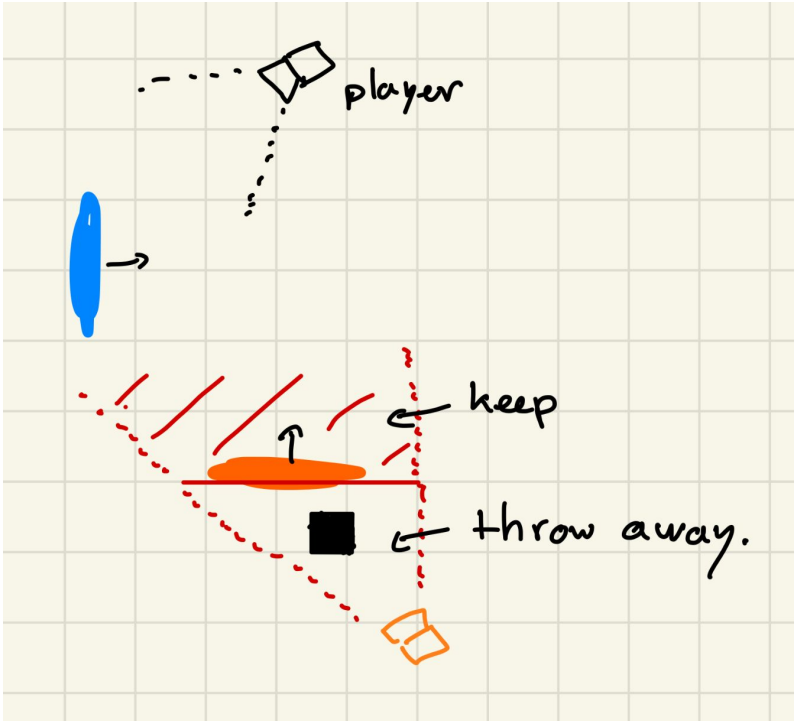
Current Features - Portal Rendering

- In order to achieve the effect on portal orange, we need to position another camera behind portal blue
- calculate transform using change of basis
- Render scene, and save to a texture
- Apply texture to portal orange, using custom shader



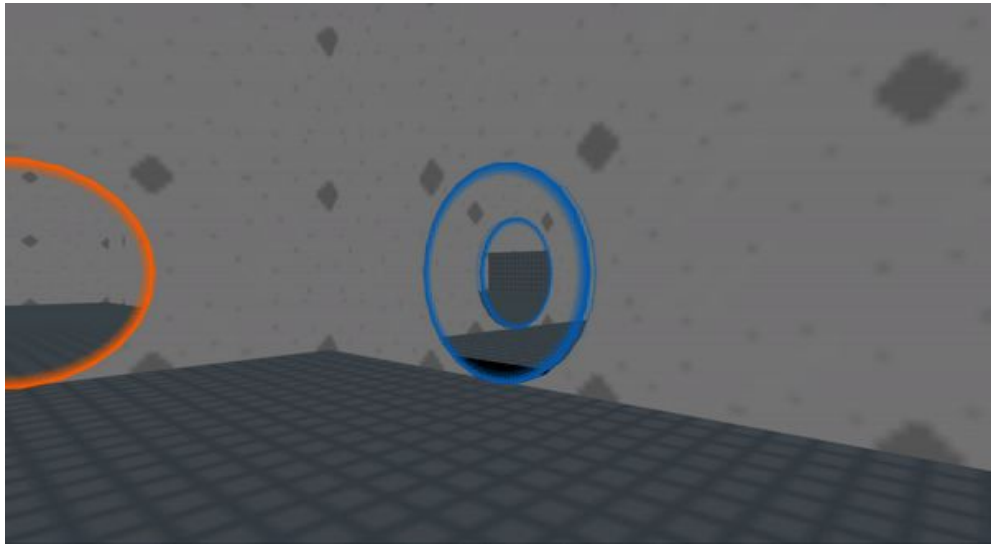
Current Features - Oblique Near Clipping Plane

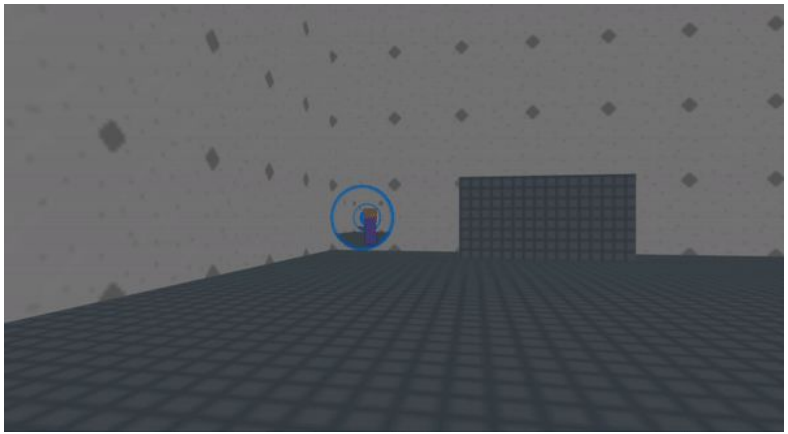
- If we simply do the last slide, can have problem if object is behind portal.
- So using linear algebra, we construct a new projection matrix to make the near clip plane oblique.



Current Features - Recursive Rendering

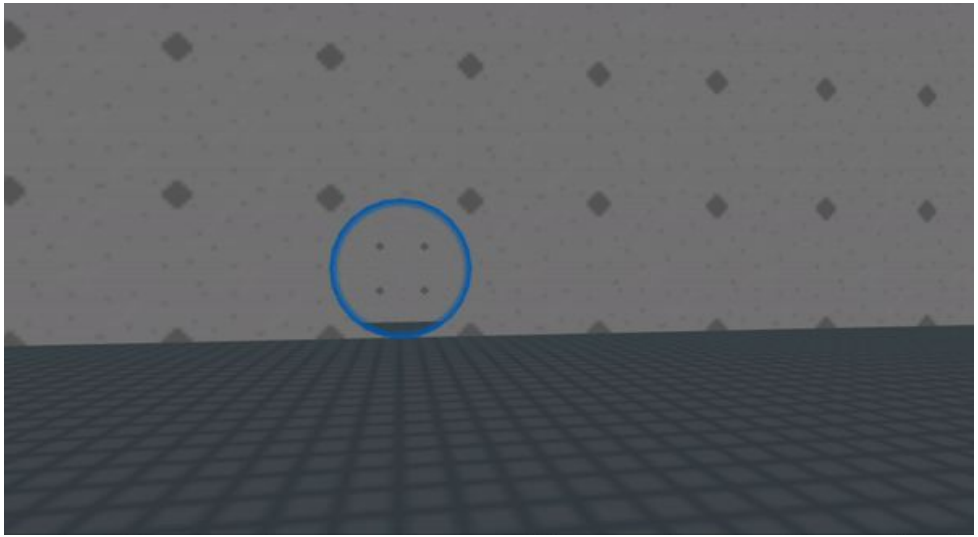
- Do the rendering process twice for each portal
- This means 2 camera to render each portal.
- Disabled rendering if unnecessary, but at most 5 cameras in the scene.
- Toggle between being able to see a portal's view itself in itself, and just being able to see the portal's colored outline in itself for performance



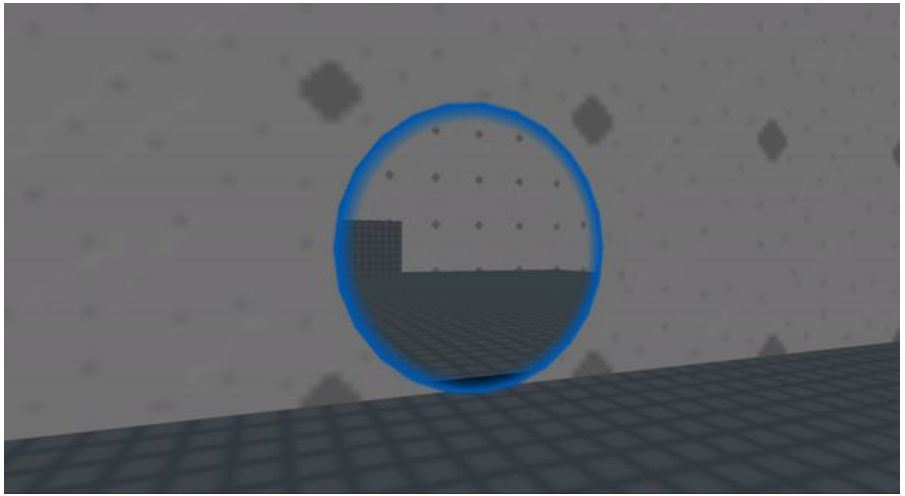


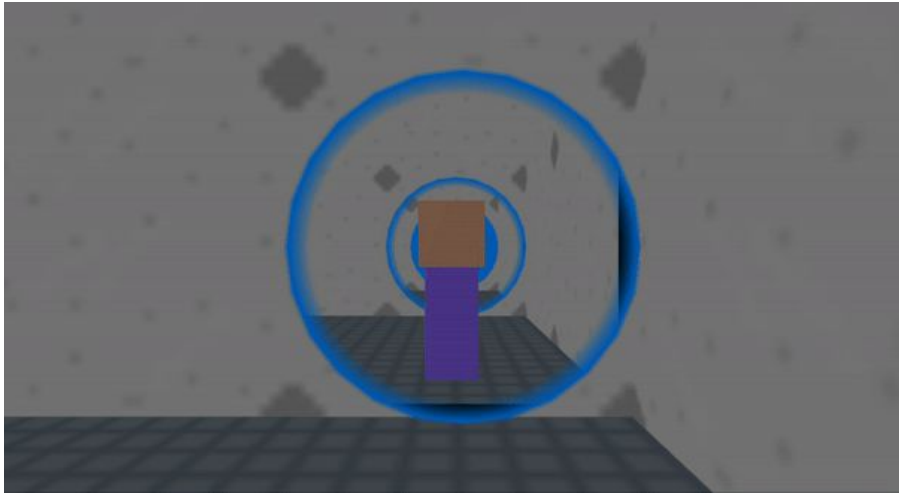
Current Features - Teleporting between Portals

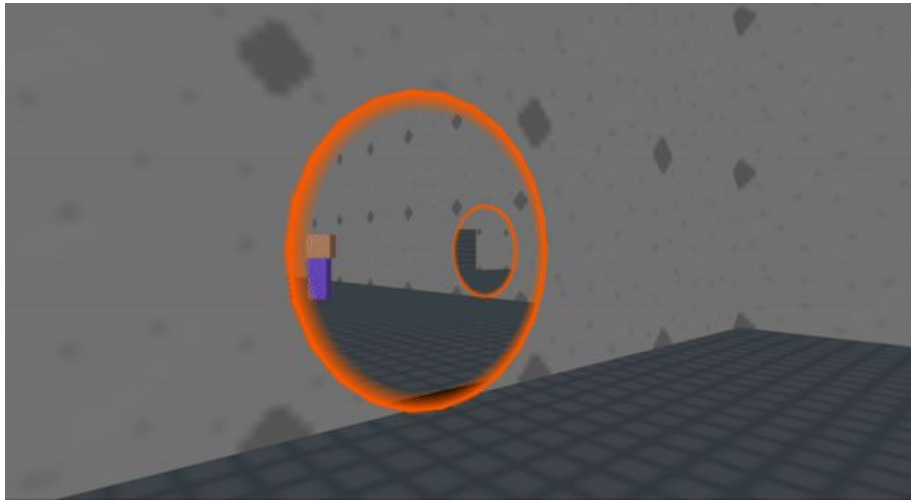
- Relies on collision detection
- Smooth teleport
- Upon touching a portal, player can clip through the wall
- Once player is completely through the wall, teleport it to other portal using change of basis



More Teleportation!







Demo