

Rendering Non-Euclidean Space

Riley Slotboom

K.M.Slotboom@student.tudelft.nl

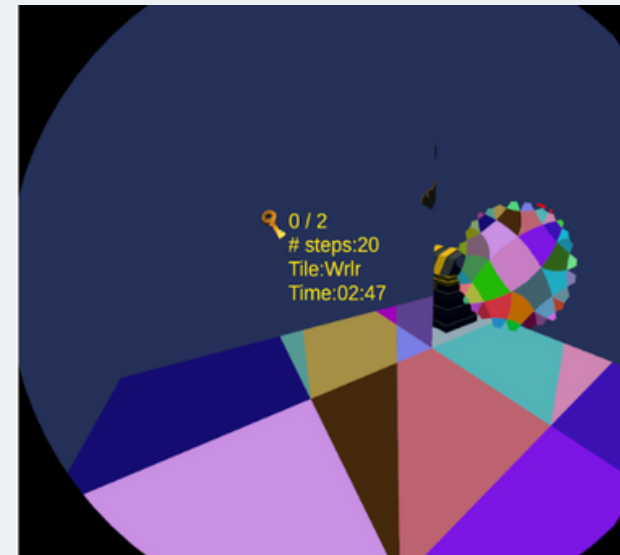
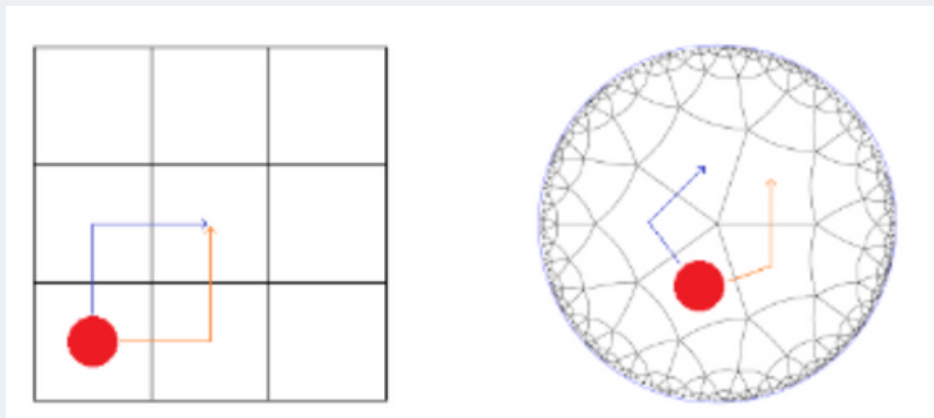
1. Background

Non-Euclidean space is any space that doesn't follow Euclid's postulates

We are interested in hyperbolic space in particular

Important property: five squares can be placed around a single vertex, rather than the usual four

This means some tiles of our own tiling would overlap in Euclidean space



2. Research Question

How do we render hyperbolic space in a performance-friendly manner?

The focus lies on two methods: the current implementation using render textures, and a new implementation using stencil polygons

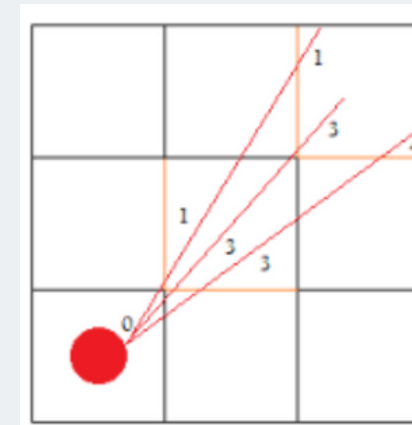
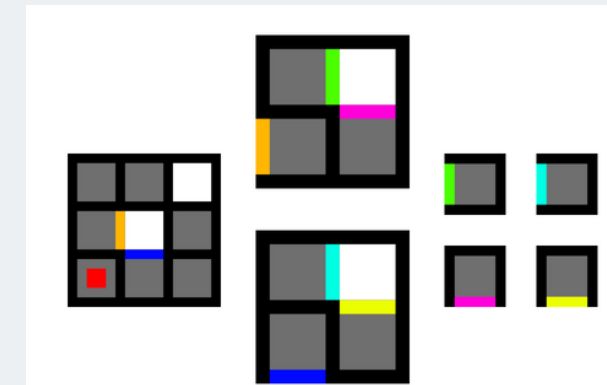
These methods will be compared in terms of performance and technical limitations

3. Overview of methods

Recursively generate "subgrids" where conflicts occur, and "portals" connecting them to the main grid

For render textures: portals are implemented by rendering a camera view onto them

For stencil polygons: portals write to the "stencil buffer"; geometry reads this buffer to decide if it should render



4. Results

Experiment: rotate a camera in place for five test cases; measure FPS

Stencil polygons have much less overhead when adding additional portals, but render textures are less constrained

	Average FPS for render textures	Average FPS for stencil polygons	Speedup factor
Center tile	143.843830	333.679134	2.32
Corner tile	258.604970	332.965919	1.29
Corner tile (90 degrees)	92.67712857	367.6417286	3.97
Edge tile	216.967100	313.425702	1.44
Edge tile (180 degrees)	146.4999928	342.3625739	2.34