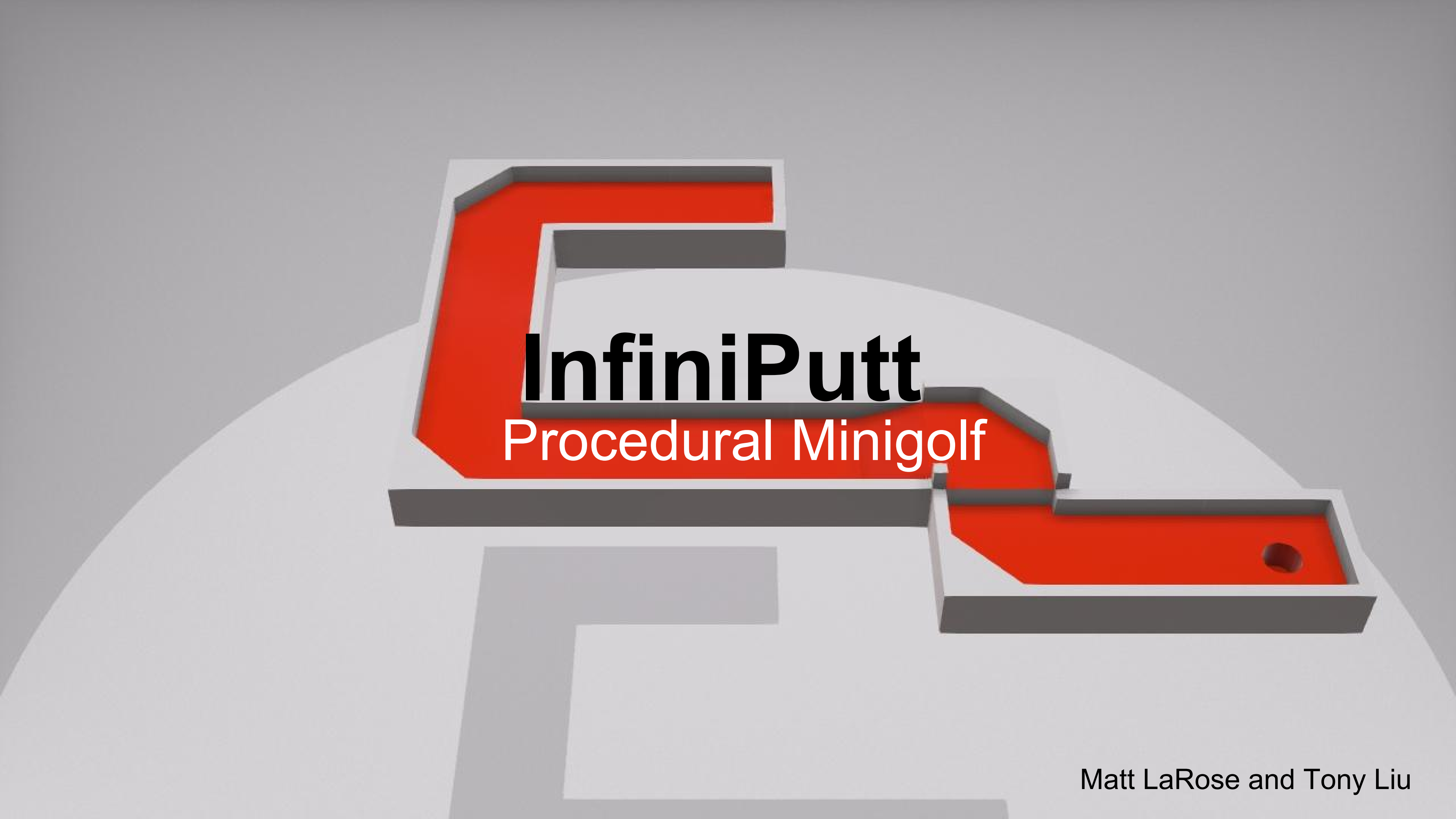


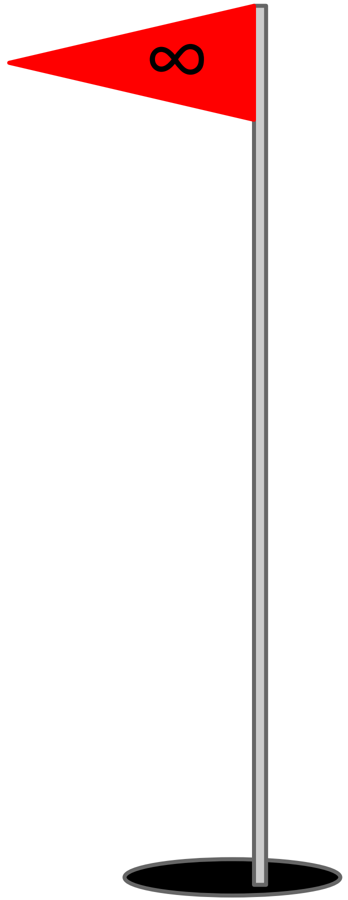
InfiniPutt

Procedural Minigolf

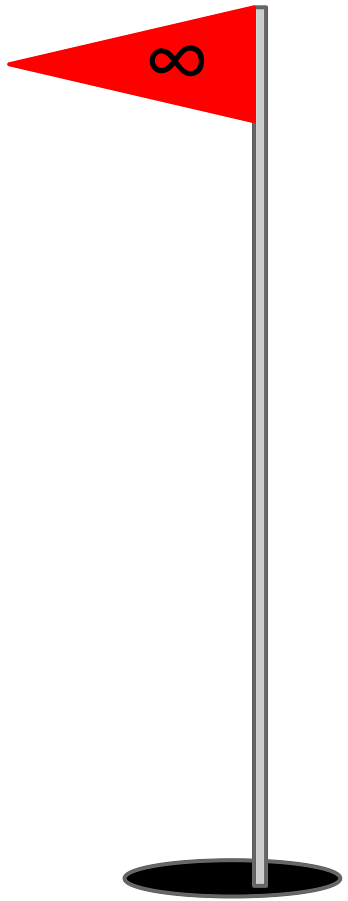
Matt LaRose and Tony Liu

Overview

1. Inspiration
2. Goals
3. Visuals
4. Physics
5. Course Generation



Inspiration



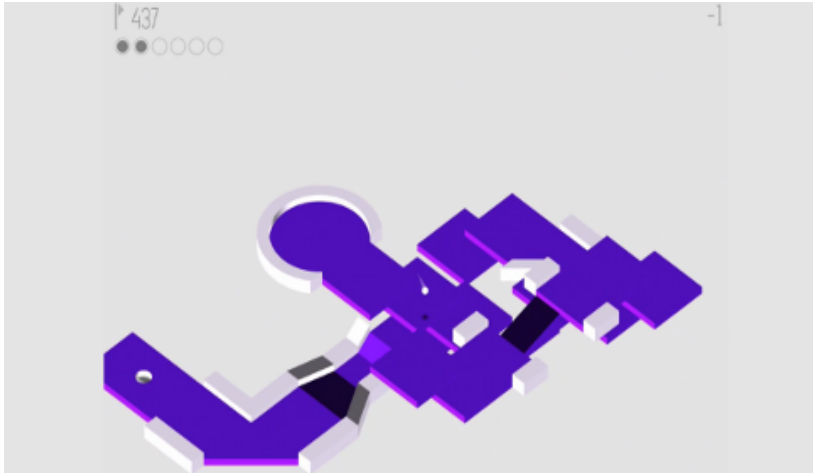
Golfinity

Good:

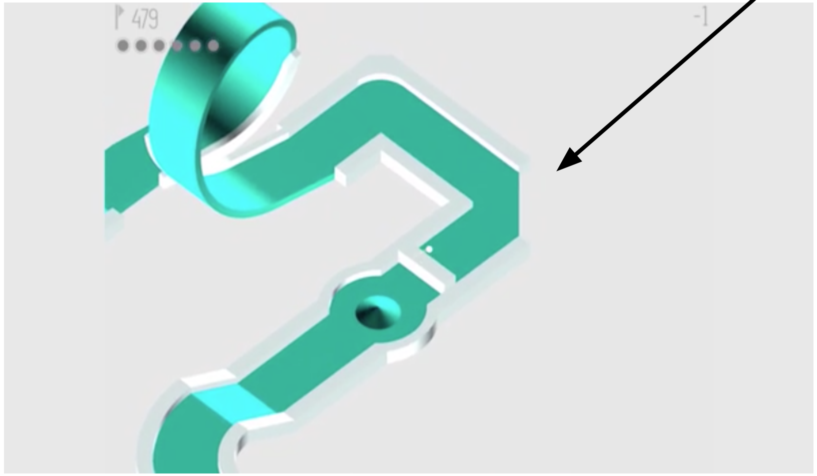
- Large variation in level generation
- Colorful, sleek model design

Bad:

- Clunky interaction
- Poor lighting and shading

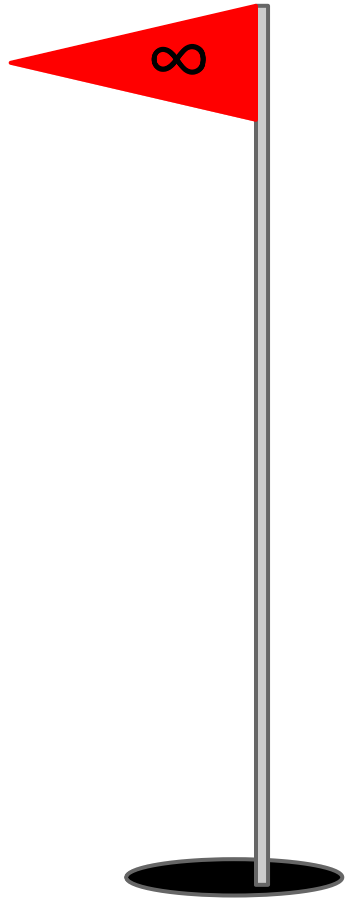


Where's the wall?



Images from Golfinity by NimbleBit, LLC

Goals



Our Goals

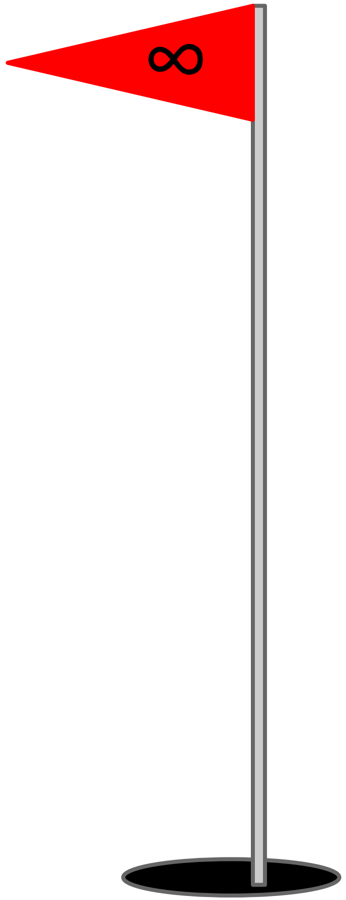
Gameplay:

- Infinite number of random, compelling minigolf courses
- Fix some issues in Golfinity

Minimalist Style:

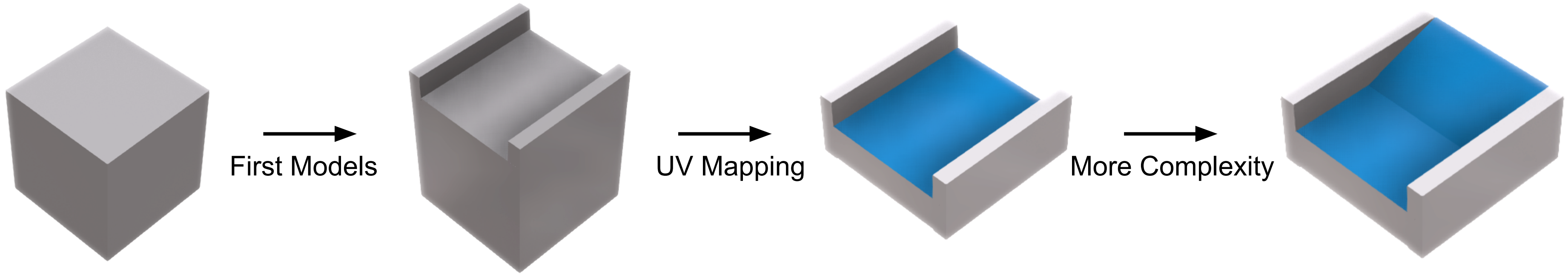
- Few, simple colors
- Grid layout with sharp models

Visuals



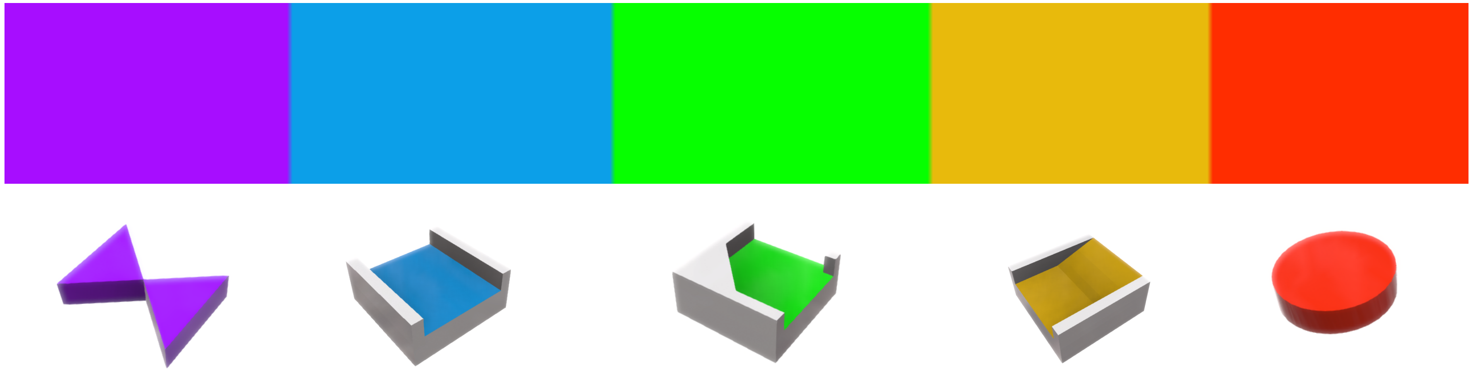
Modeling

Iterating from scratch via Blender



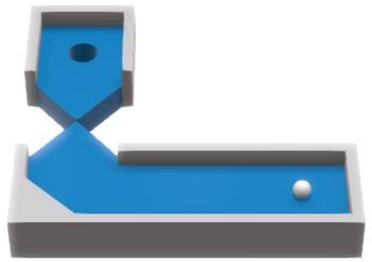
Color Palette

Generated custom palettes with Adobe Color CC



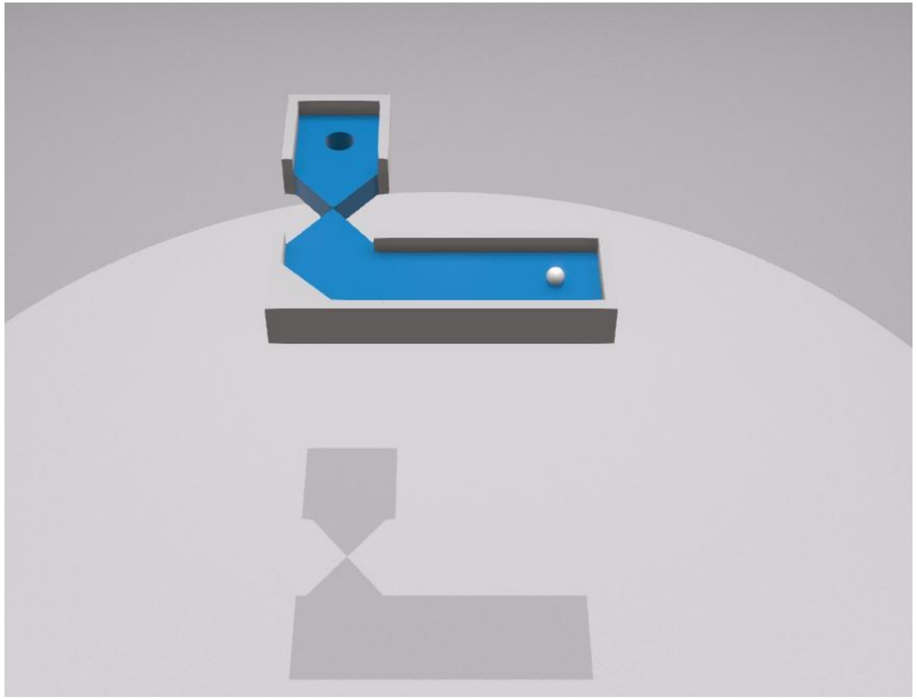
Lighting

Lighting placed to accentuate wall shadows

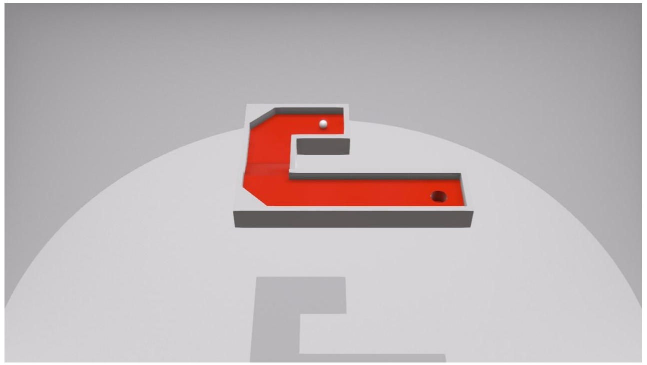
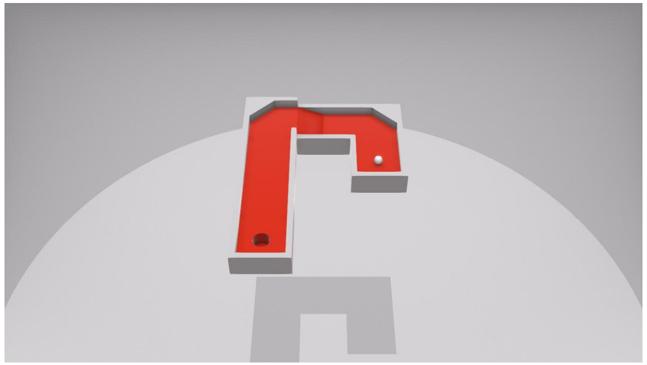
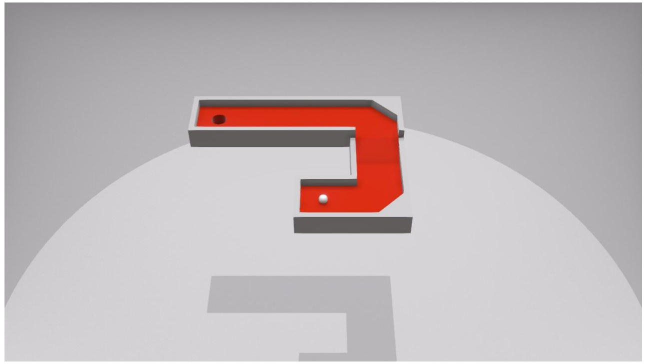
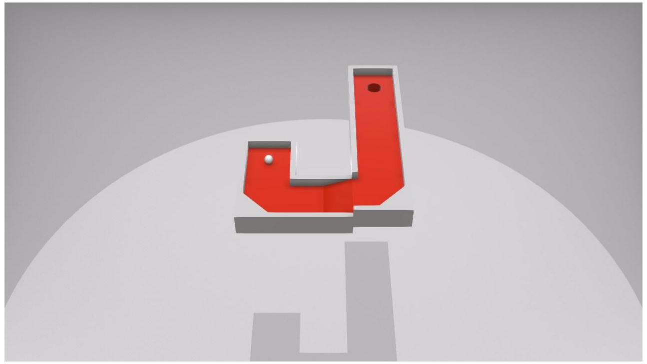


Lighting

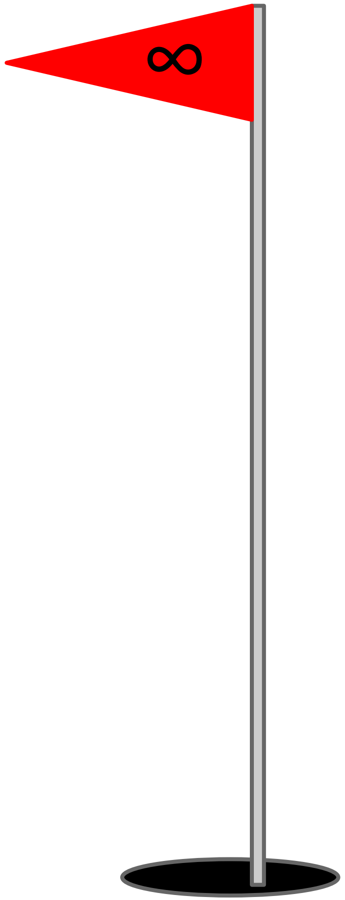
Ground plane for extra depth



Cameras

Physics



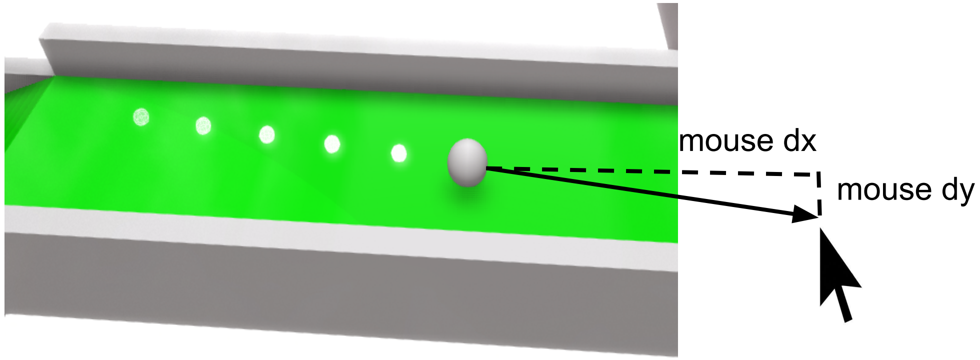
Physics

- Creating a real physics engine is hard
- Integrating an existing physics engine is also hard
- Faking physics is easy and sufficient
- Collision detection already implemented in G3D

The Physics of Minigolf

- Hitting
- Acceleration
  - gravity
  - friction
- Collisions
  - bouncing

Hitting



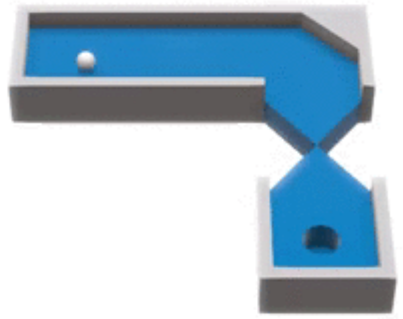
Acceleration

Gravity

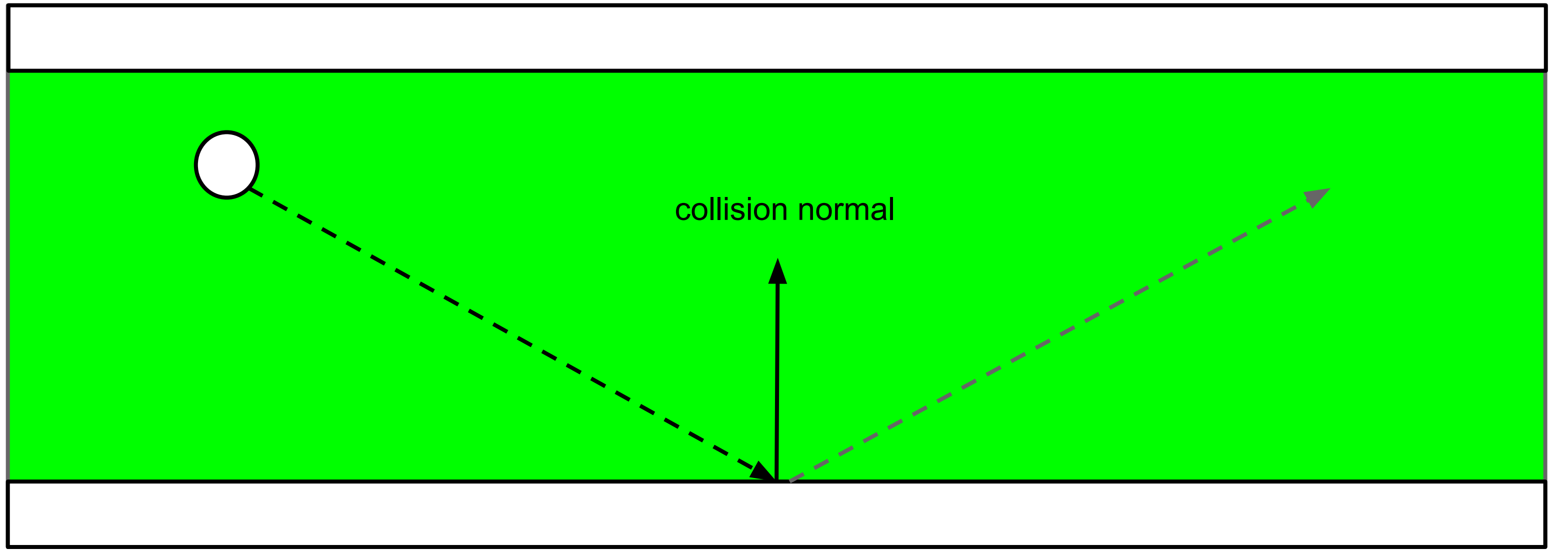
- Just add gravity to velocity vector in each call in game loop

Friction

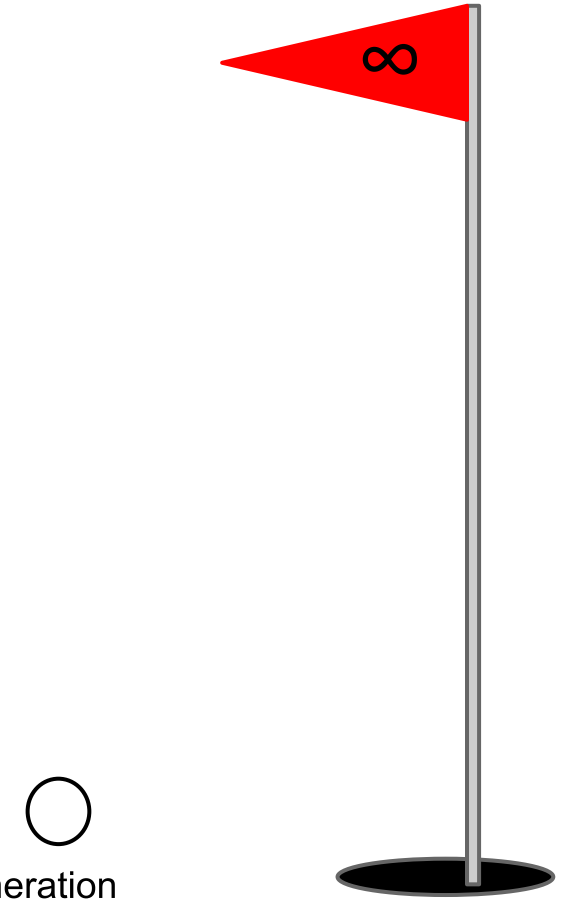
- Reduce velocity by a constant factor each time
- Clamp to 0 when length of velocity vector below threshold



Collisions



Course Generation



Overview

1. Place start (**T**ee) and end (**H**ole) points
2. Create path from **T** to **H**
3. Generate height changes on the path
4. Put things on the path

Point Placement

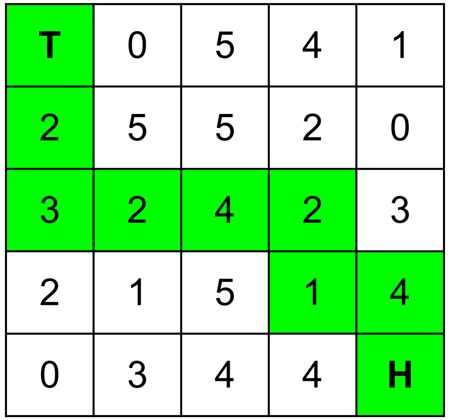
Restrictions:

- Points, like the tee and hole, have to be onscreen
- XZ-plane only
- Integer-value coordinates only

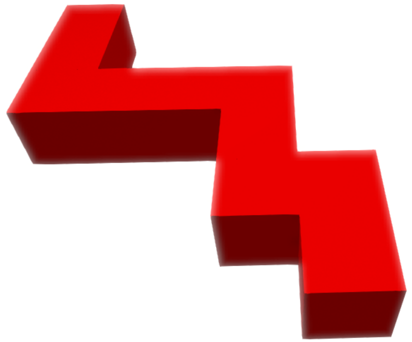
Finding a path

- Give each grid point a random value
- Find shortest path between start and end points

Can adjust “twistiness” by adjusting the randomness of the values



T	0	5	4	1
2	5	5	2	0
3	2	4	2	3
2	1	5	1	4
0	3	4	4	**H**

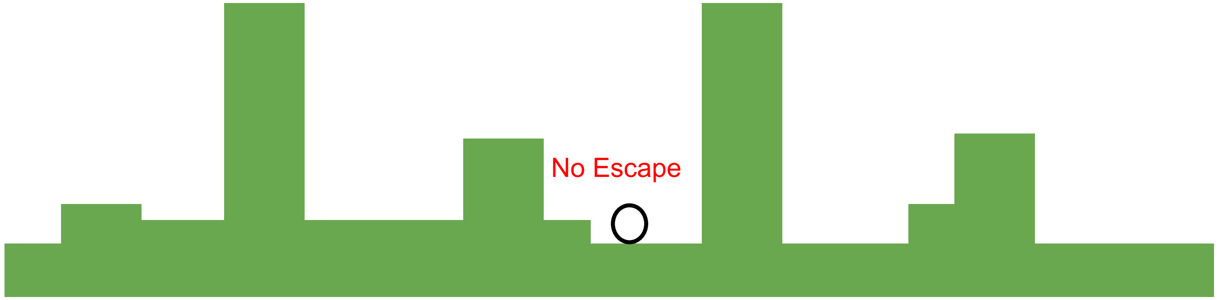


Generating Height Changes

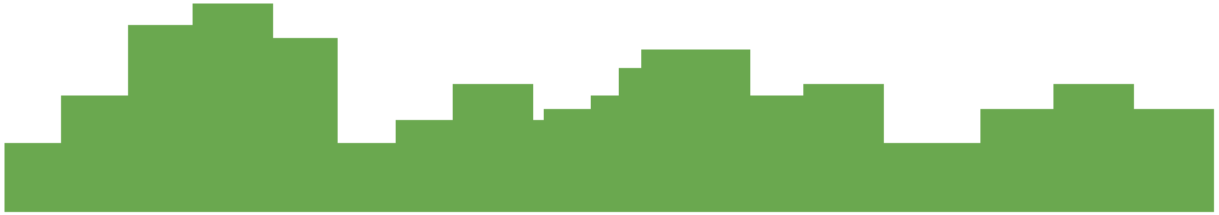
Must be aware of the following design challenges:

- The heights of points in the path are not fully independent
- Easier to go down than up
- Can’t have inescapable ruts

Height Independence



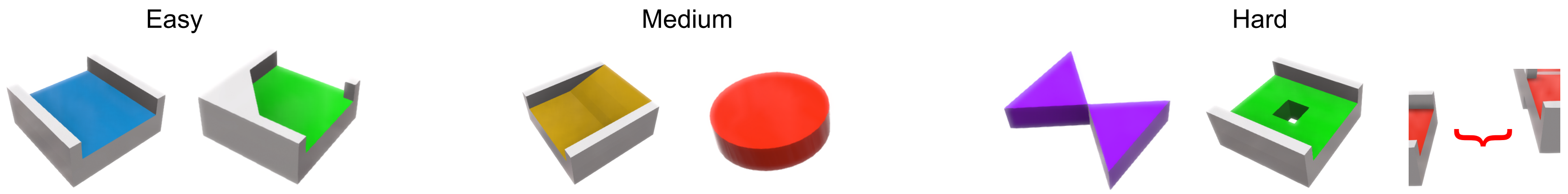
Random
Unnatural
Unplayable



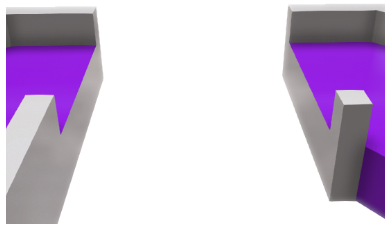
Context-Aware
Smooth
Playable

Course Entities

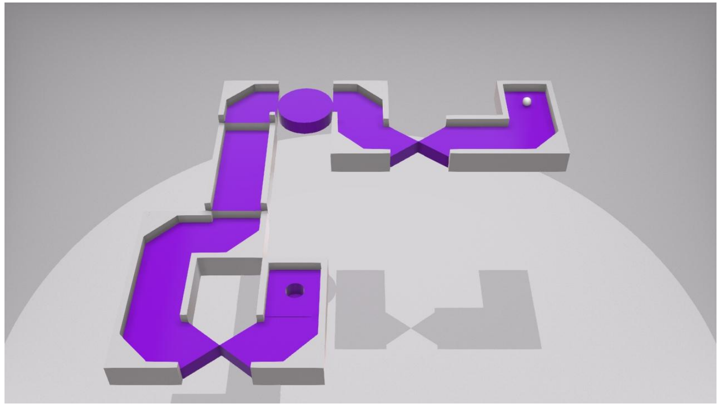
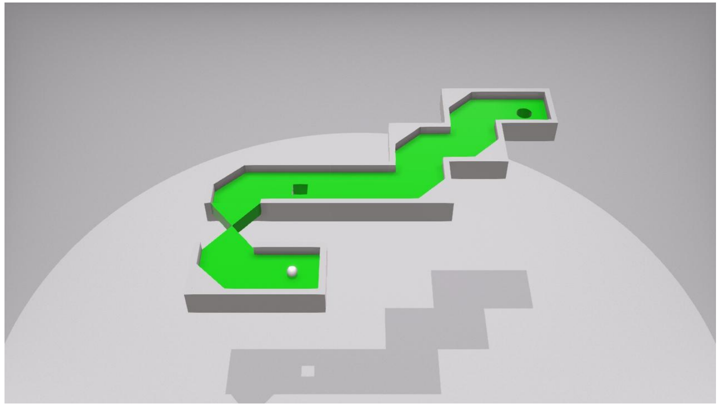
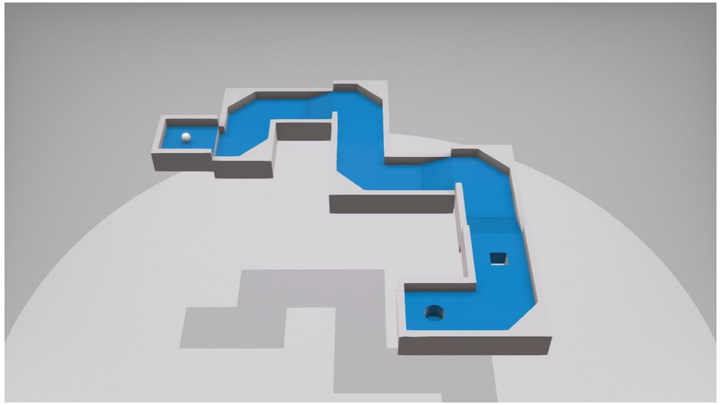
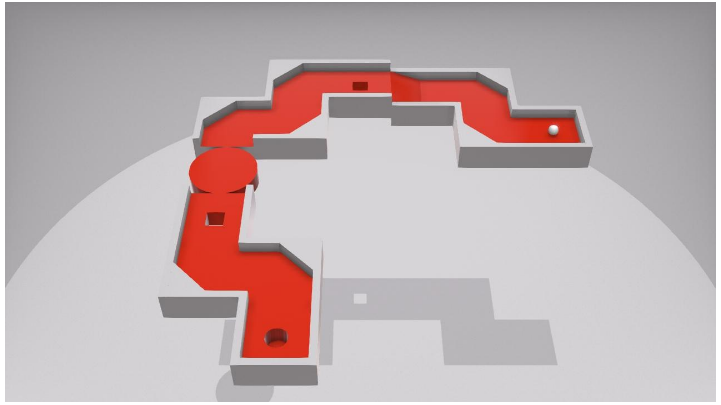
- Certain kinds appear more frequently depending on difficulty



- Not each can be used in every context
  - ex: can't have a gap when the adjacent elevations are equal



Some Representative Courses

Thank you!