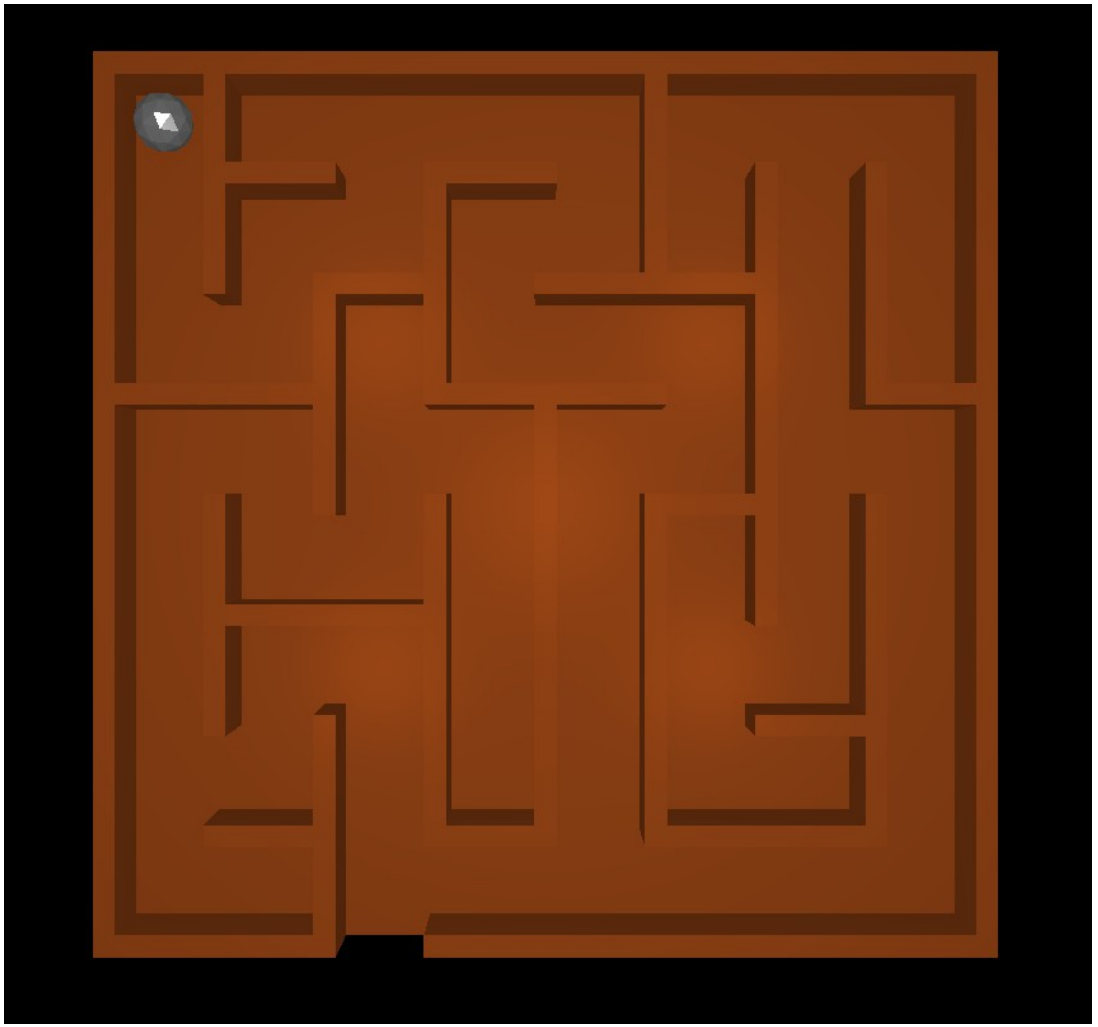
Ball Labyrinth

Introduction

- OpenGL
- SDL2
- GLM



Physics Model

- **3D rigid body model**
- **Ball represented as center with radius**
- **Walls represented as boxes with infinite mass**
- **Symplectic Euler used as update step**
- **Simple ball rolling:**
  - No torque
  - No slip on the surface assumed

Rolling Ball

- **Angular momentum can be seen as reduced force acting on the ball**

$$E_{tot} = E_{kin} + E_{rot} \qquad E_{tot} = \frac{m v^2}{2} + \frac{I \omega^2}{2} \qquad I = \frac{2 m R^2}{5}$$

$$E_{tot} = \frac{m v^2}{2} + \frac{m v^2 R^2}{5 R^2} \qquad m g h = \frac{m v^2}{2} + \frac{m v^2}{5}$$

$$g h = \frac{7 v^2}{5}$$

Rolling

$$v = \sqrt{\frac{5}{7} g h}$$

Sliding

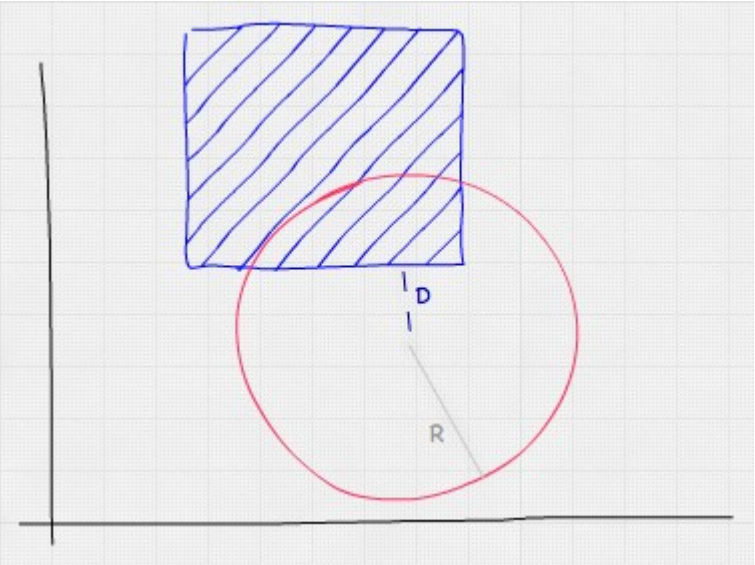
$$v = \sqrt{2 g h}$$

Rolling friction

- **Friction is a constant force dependent on the normal force to the surface**
- $F_r = \mu F_N$

AABB collision detection

- **AABB boxes used to model the maze and floor**
- **Sphere to AABB collision detection**



https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_collision_detection/Bounding_volume_collision_detection_with_THREE.js

Collision Steps

- **Detect if distance to centerpoint < radius**
- **Move ball outside of the collision box**
- **Update velocity according to rigid body model**

Maze Generation

- **Random automatic maze generation**
  http://www.thingiverse.com/thing:24604
- **Python script generates openscad file and .txt file with AABB boxes**
- **Openscad → generates .stl**
- **Blender → .obj from .stl**
- **Tiny Obj Loader used to import the .obj maze**