Collision Response

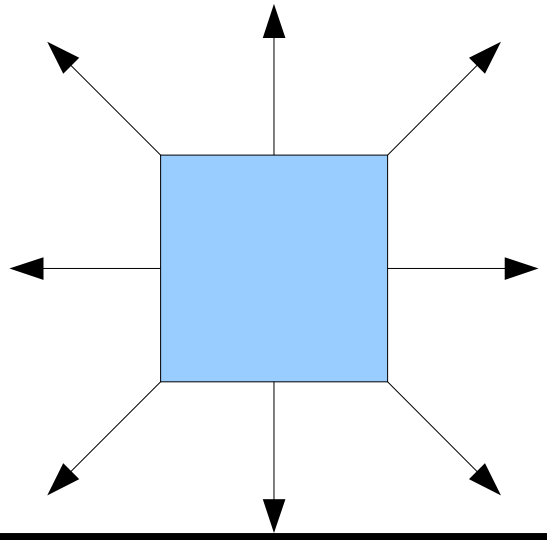
Dealing with those pesky walls

BACKGROUND

- In the simple ball game, we'll eventually want to handle walls.
- Collision response with said walls is surprisingly tricky.

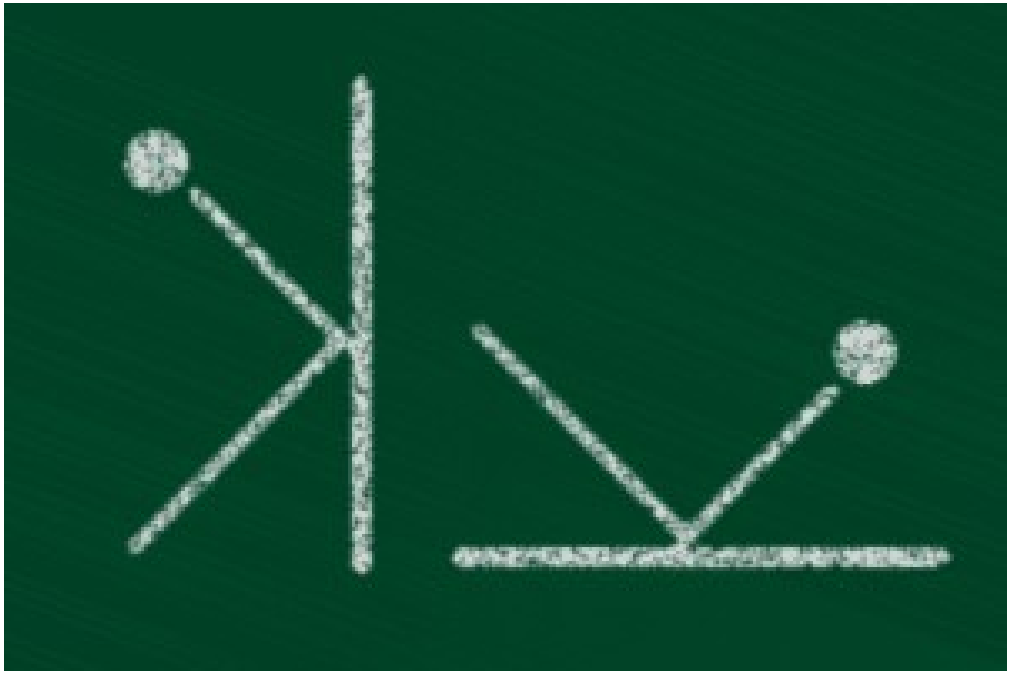
COLLISION DETECTION

- First step is figuring out the collision.
- We'll first augment the tile data with collider information.
- Basically what we're after is the collision surface normal vector.



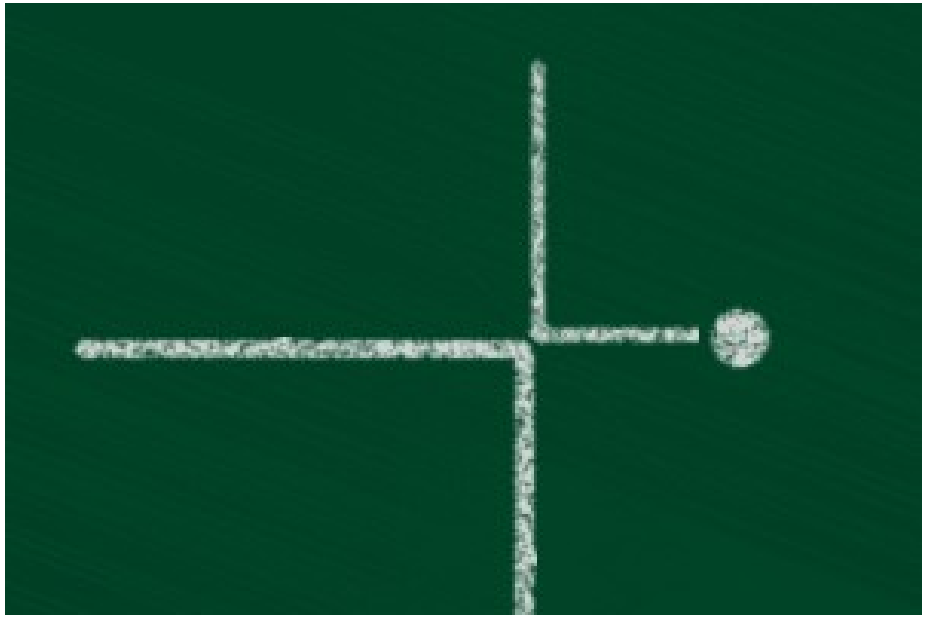
REASON FOR NORMALS

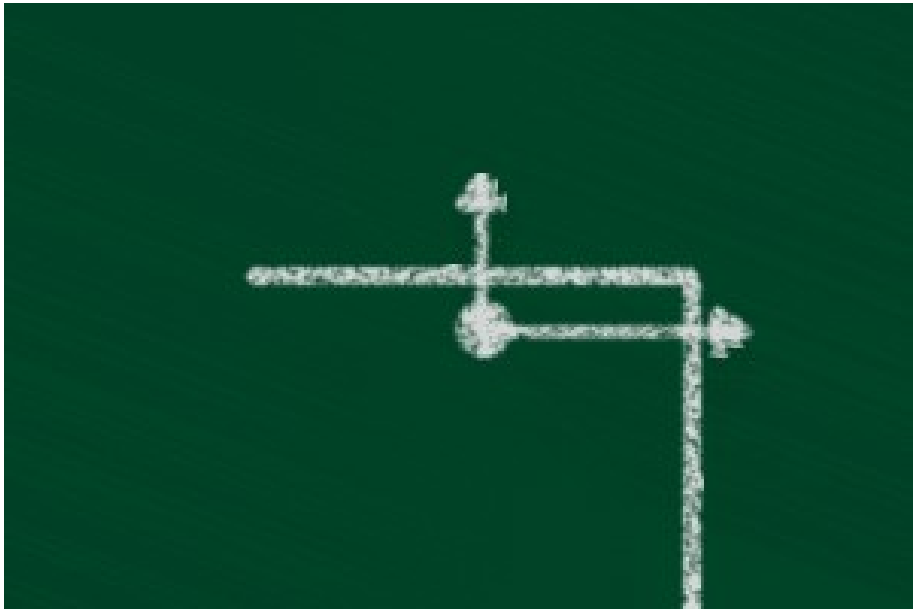
- While simple cases work without trickery..



REASON FOR NORMALS

- Corners and other special cases are annoying.



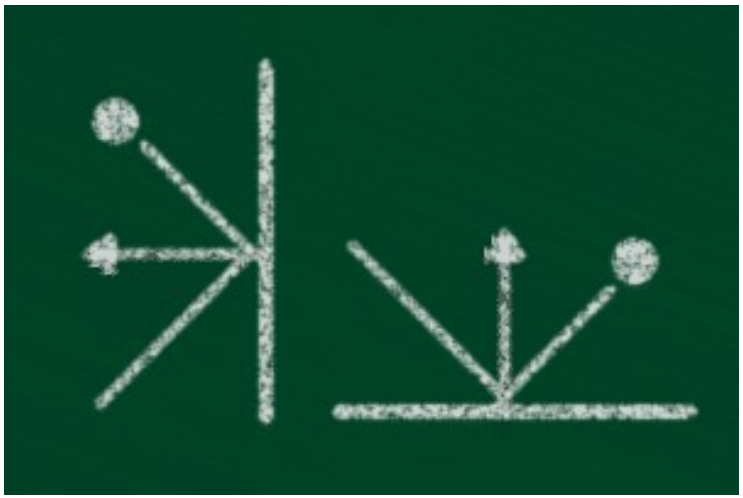


Enter the Dot Product

- Dot product is:
  - d = x1 * x2 + y1 * y2
- Or, in our case:
  - d = motionx * normalx + motiony * normaly

THE SIMPLE CASE

- d = mx * nx + my * ny
  - D = 1 * -1 + 0 * 1 = -1
- If we multiply normal with dot product and substract the from motion vector twice, we'll get
  - mx = mx - d * nx * 2 = 1 - (-1 * -1 * 2) = -1
  - my = my - d * nx * 2 = 1 - (-1 * 0 * 2) = 1



THE CORNER CASE

- d = mx * nx + my * ny
  - d = 0 * 0.71 + -1 * 0.71 = -0.71
- Again..
  - mx = mx - d * nx * 2 = 0 - (-0.71 * 0.71 * 2) = 1
  - my = my - d * nx * 2 = -1 - (-0.71 * 0.71 * 2) = 0