① box center (cx, cy), type OABCX

② grid point position orientation type

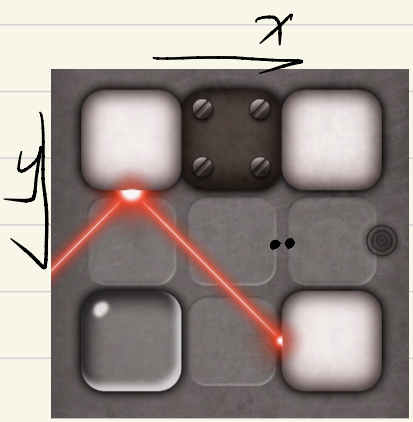
(x, y) (dx, dy) OABCX

(1, 0) (-1, 0)
(0, 1) (0, -1)

③ ray point (x, y) (Vx, Vy)

incident	box ori	reflection
(1, -1)	x = -1	(-1, -1)
	y = 1	(1, 1)
(1, 1)	x = -1	(-1, 1)
	y = -1	(1, -1)
(-1, 1)	x = 1	(1, 1)
	y = -1	(-1, -1)
(-1, -1)	x = 1	(1, -1)
	y = 1	(-1, 1)

1. readd and import bft file.
2. box:(cx, cy), type put ABC in O.
3. arranged abc, covert to the nearest 4 grid points.
4. function on how to reflect and transmit
5. put start point in, generate all light-passed point
6. check if the required points are passed



incident point: (4, 7), (−1, −1)
save position passpoint[ ]
predict the wall and find
(4, 7), (1, 0) ? type

how to predict?

(irx, iry), (vx, vy)

irx { even / odd } look for
(irx, iry), (−vx, 0)
(irx, iry), (0, −vy)

1. in-rpoint[] List [入射光]
2. save position,
3. generate next incident

trans (4,7),(-1,-1)
refle (4,7), (1,-1)
$\Rightarrow$ (irx+Vx, iry+Vy) (V_x, V_y)

out point ⟶ next incident point

4. ray pass the boundary / absorbed
   terminate this ray

5. When no more incident ray point
   finished.