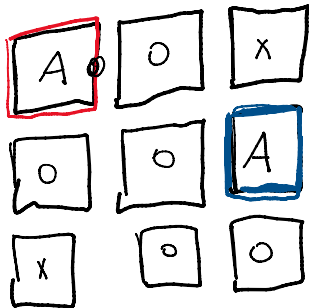


## Board



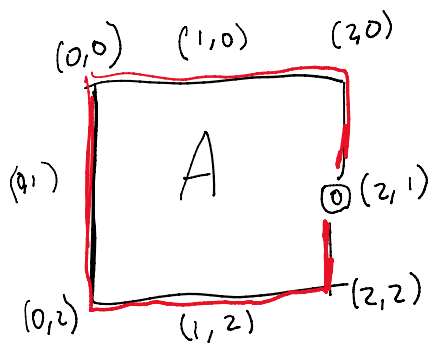
Right now the code reads each pre-existing block as an element in an array, with each row being a new array.

Each block can be represented by a set of coordinates that correspond to their matrix indices.

Examples:

(0,0) (2,1)

## Blocks and Points



The points (and lasers) use a different co-ordinate system. As each block has a different "wall" that they can have either a point or a laser reflection happening, each block object will need four wall attributes.

Example:

Block (0,0)

- Left Wall = (0, 1)
- Right Wall = (2,1)
- Top Wall = (1,0)
- Bottom Wall : (1,2)

To convert a block co-ordinate to a get the four wall co-ordinates, use this formula

Block (x,y)

- Left Wall = (2x, 2y+1)
- Right Wall = (2(x+1),2y+1)
- Top Wall = (2x+1,2y)
- Bottom Wall : (2x+1,2(y+1))

## Lasers

- Each laser has an origin point and a trajectory.
- We can give each laser an end point based on its direction.
  - If the trajectory is (1,1), that means the direction is Bottom Left.
  - The end points are anything beyond (x, len(columns)) and (len(rows), y), where x and y are any value
- On its way to the endpoint, we can create a set of all the wall co-ordinates the laser passes through. If there is an intersection between the co-ordinates in this set and all the co-ordinates in the set of points, then we have reached our solution

# Blocks

- When a laser encounters a block, three things can happen
  - If the block is opaque, the laser finds a new endpoint with co-ordinates being the wall of the opaque block where the laser intersected
  - If the block is a reflect block, the laser finds a new endpoint with co-ordinates being the wall of the opaque block where the laser intersected and a new laser is made with origins at the reflect wall and a new direction depending on the original direction wall that was hit
  - If the block is the refract block, the laser keeps its original endpoint but a new laser is still made with origins at the refract wall and a new direction depending on the original direction wall that was hit
- Therefore there should be a function that tells us which wall is the one being hit. Since each block already has each wall as an attribute, we can make four functions:
  - `getLeftWall()`
  - `getTopWall()`
  - `getBottomWall()`
  - `getRightWall()`
- Each laser should also have the functions,
  - `getDirection()`
  - `getEndpoints()`
  - `getIntersects()`
- There should be a function that counts all the lazors we have, count all the points and walls they intersect, and cross-references those interections with the "goal" points (defined in the BFF file as "P x y")