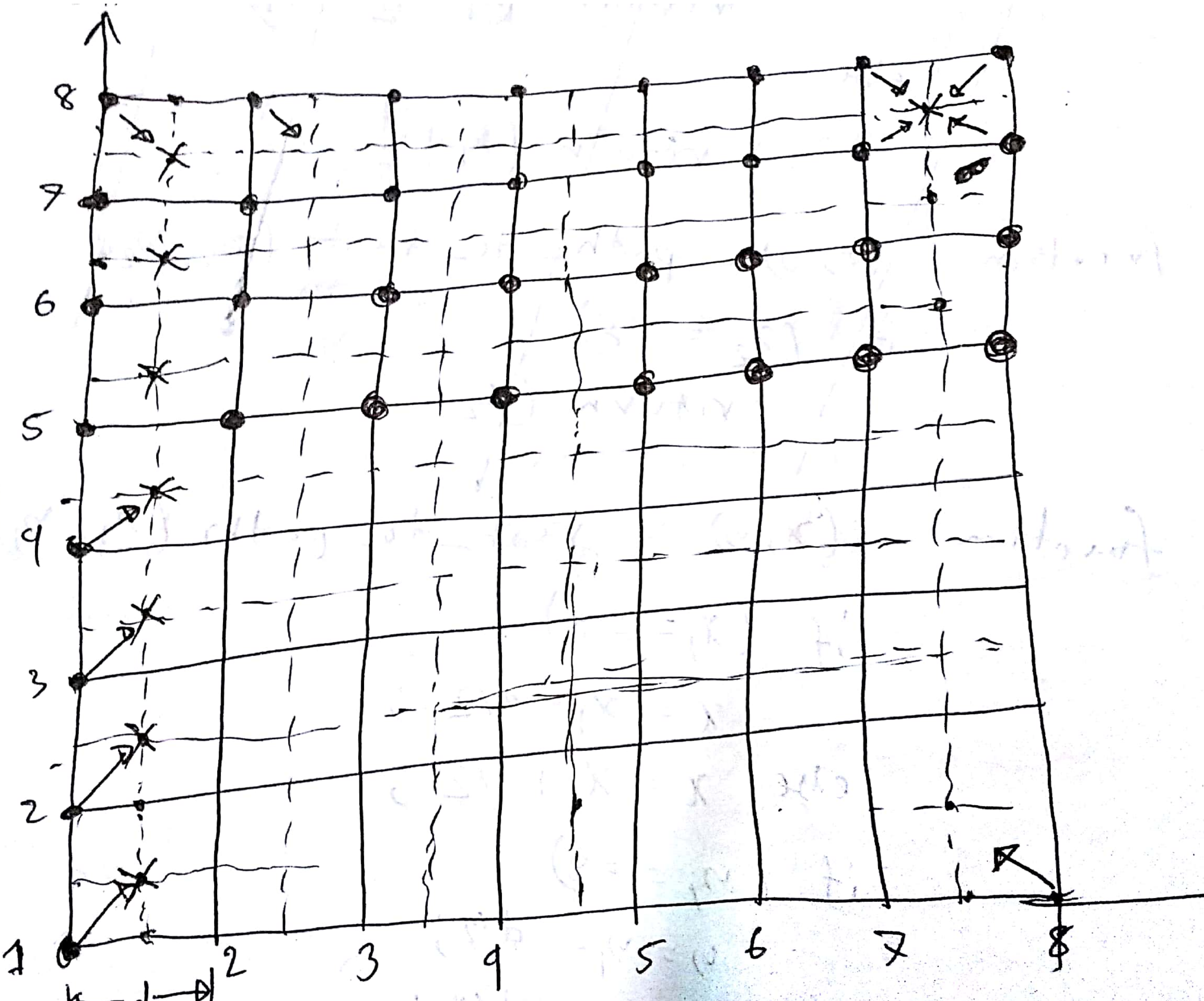
8
7
6
5
4
3
2
1
0
2
3
4
5
6
7
8
d

Say we have to move a piece from (x_1, y_1) to (x_2, y_2). The big dots are the places where the pieces ~~=~~ rest. The dotted lines are paths to move the pieces. At first we move the piece to resting place to path, then we transfer the piece, then we move it back to the resting place. We define ~~two~~ a functions;

```
~~function (x, y) = rest_to_path (x1, y1)~~
        ~~if (x1 == 8)~~
                ~~return (x1 - d/2, y1)~~
        ~~else~~
                ~~return (x1 + d/2, y1)~~

~~function (x, y) = path_to_rest (x2, y2)~~
        ~~if (x2 == 8)~~
                ~~return (x2~~
```

```
function (x, y) = rest_to_path (x1, y1)
        if (x1 == 8)
                x = x1 - d/2 ;
        else  x = x1 + d/2 ;
        if (y1 == 8)
                y = y1 - d/2;
        else  y = y1 + d/2 ;
    return (x, y) ;
```

main progam:

```
    $(x_1', y_1')$ = rest_to_path $(x_1, y_1)$
    $(x_2', y_2')$ = rest_to_path $(x_2, y_2)$

    move from $(x_1, y_1)$ to $(x_1', y_1')$;
    move from $(x_1', y_1')$ to $(x_2', y_2')$;
    move from $(x_2' y_2')$ to $(x_2, y_2)$;
```

——x——

Moving a piece from any two points (x_1, y_1) to (x_2, y_2) will be a function like this:

```
move $(x_1, y_1, x_2, y_2)$
{
    ~~move servo~~
    move from $(x_1, y_1)$ to $(x_2, y_1)$;
    move from $(x_2, y_1)$ to $(x_2, y_2)$;
}
```

In this method two piece will never collide ~~due~~ because we move them ~~but~~ in the path between two pieces.

we can add further instructions to the above function like this:

```
move ($x_1 ; y_1 , x_2 , y_2$)
{
        move servo to ($x_1 , y_1$);
        catch the piece;
        move from ($x_1 , y_1$) to ($x_2 , y_1$);
        move from ($x_2 , y_1$) to ($x_2 , y_2$);
        release piece;
}
```

"current_position" of servo may be a global variable.

we can write ~~funct~~ more functions to do specific tasks:

```
move_servo ($x_1 , y_1$)
{
        moves servo from "current_position"
        (global variable) to ($x_1 , y_1$);
}

~~catch piece~~
catch-piece ()
{
}
```

```
move_x (x1, x2)
{
    moves servo from x1 ~~x~~ to x2 by
    operating the single/ the double
    motor;
}
move_y (y1, y2)
{
    does the same for y direction
}
release_piece()
{
}
```

move_x() and move_y() functions ~~actually~~ actually do nothing but ~~operation~~ operating the motors. So, we can use them in move_servo() function as well.

In move() function we can add

```
move ( )
{
    . . .
    . . .
    . .
    update current_position to (x2, y2);
}
```

So that after each move the current position of the servo ~~upto~~ updates.