Design Document

1. The constructor will take the grid as input. It will initialization the self._square_size, self._grid, self._n, self._e , self._s, self._w

```
def __init__(self, grid, square_size):
        self._grid = grid
        self._square_size = square_size
        self.n = None
        self.e = None
        self.s = None
        self.w = None
```

2. It will provide private data grid, square_size
   It will provide public data n, e, s, w

3.

(1)draw the board.

I will use a for loop to read the string in the grid. So that i know what string is in each cell.

For example, if grid is [[“sw”, “kw”],[“ew”, “e+”]], then i can know that in (0,0) cell, the string is “sw”.I use a for loop to read the string so that i can draw them out. For instance, if (i,j) is ‘ew’, i will first get ‘e’. I draw the pipe from the east side of the cell to the middle.

```
gui.line(i*size+size, (j*size+size/2), (i*size+size/2), (j*size+size/2),'black',(size/10))
```

Size is the lenght of each cell, which come from square_size. Size/10 is the width of the pipe.

Then i can get the ‘w’. i will draw a pipe from the west of the cell to the middle.

```
gui.line(i*size, (j*size+size/2), (i*size+size/2), (j*size+size/2),'black',(size/10))
```

Then the shape of the pipe of the cell (0,0) finish.

I will draw the cell one by one and then the whole pipe will show out.

(2) fill state

I will get the self, x,y and state as input data. From x,y i can know that which cell will change the state. From state i can know the pipe has water or not. True = has water. False = no water. I draw a smaller blue line when drawing the pipe. All the cell have a if statement.

```
if get_fill_state(self, i, j) == 'True':
```

Then the blue line show out. The pipe fill with water.

```
if get_fill_state(self, i, j) == 'False':
```

Then the blue line will not show out. The pipe is still black.

(3) Rotate_cw(self,x,y), rotate_ccw(self,x,y)

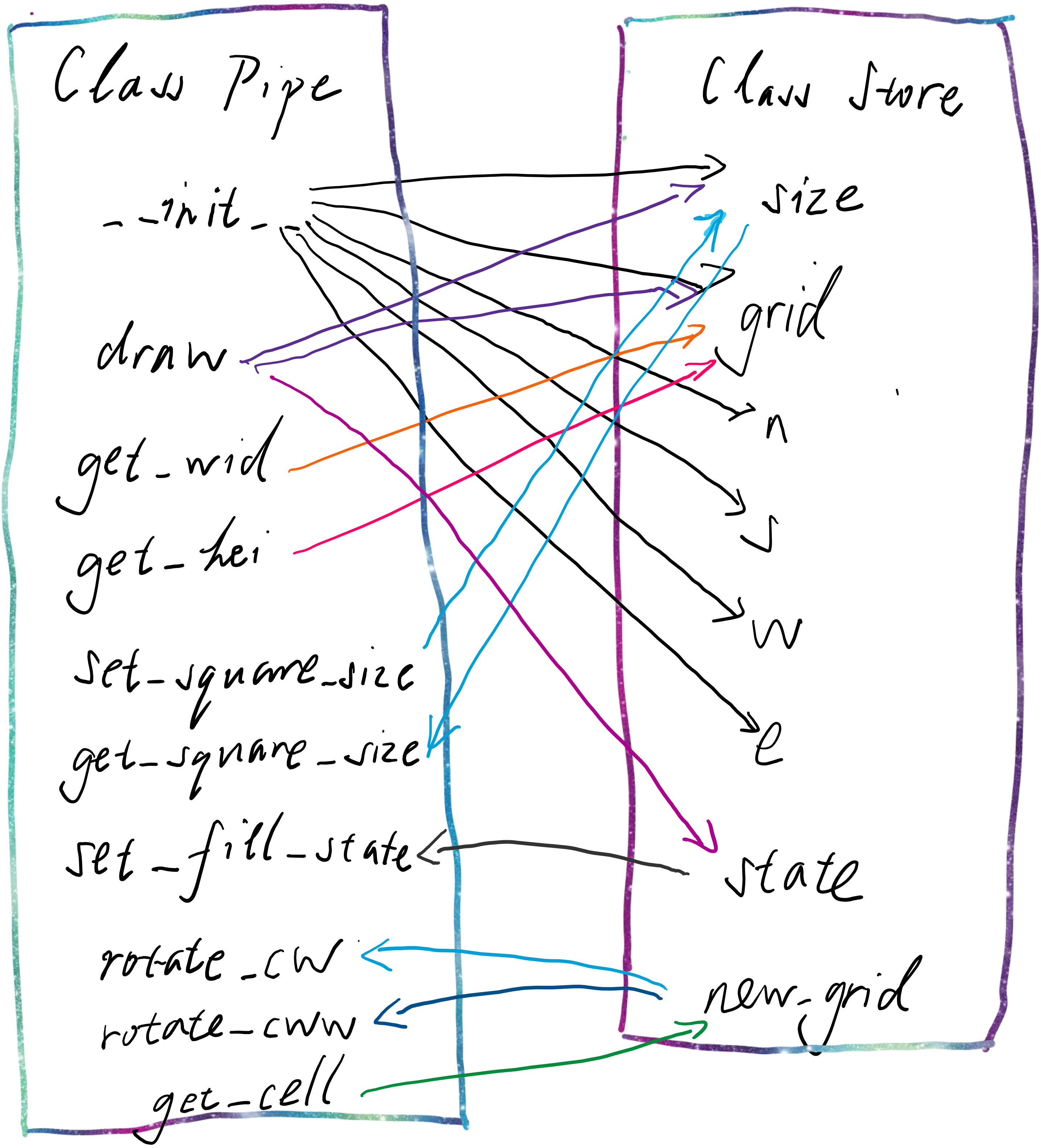
Firstly, i will get the string in grid[j][i]. then i use the for loop to write a new string which are the result of the rotate and then replace it.then i recall the draw function. It will rotate.

(4) get_cell(self, x,y)

I will get the string from grid[y][x] and then resort it by the order n e s w +.(f) then return it.

4. It will store the original input string. It will check the n s w e whether they are linked.

For example, if the string is ‘we’ then it will check whether the grid[y][x].w has ‘e’ whether grid[y][x].e has string ‘w’. If it has, change the state to True. it will keep checking until n s w e all become None.



Class Pipe
__init__
draw
get_wid
get_hei
set_square_size
get_square_size
set_fill_state
rotate_cw
rotate_cww
get_cell
Class Store
size
grid
n
s
w
e
state
new_grid