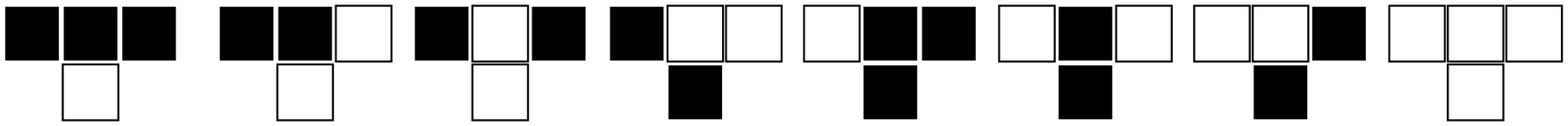
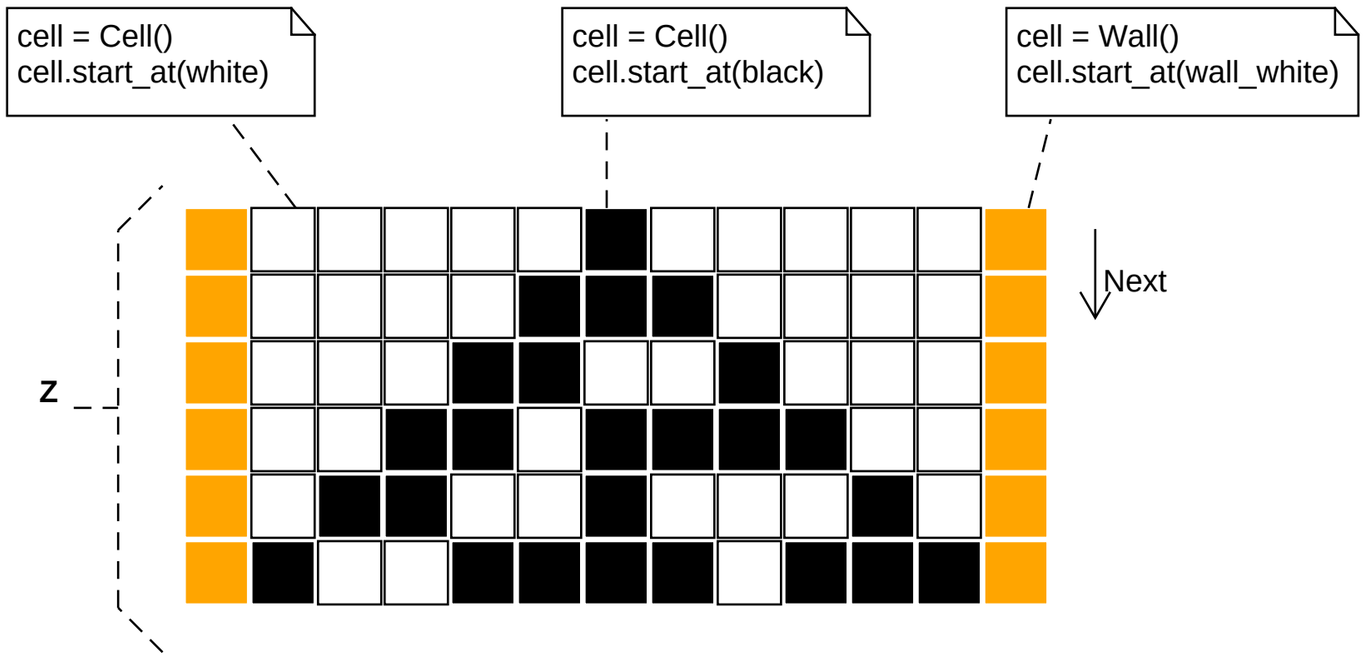
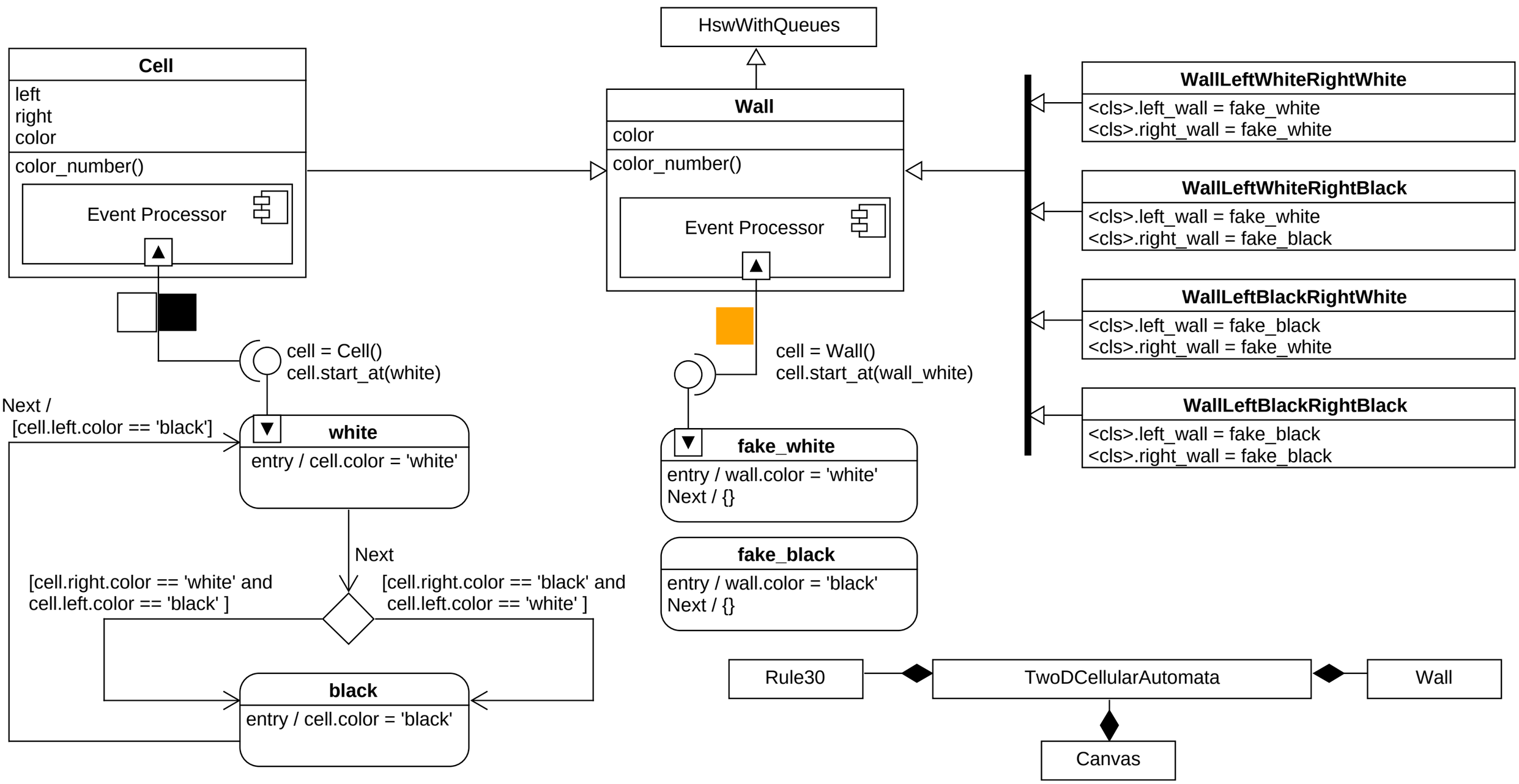



cell = Cell()
cell.start_at(white)
cell = Cell()
cell.start_at(black)
cell = Wall()
cell.start_at(wall_white)
Next
z


HswWithQueues
Cell
left
right
color
color_number()
Event Processor
Wall
color
color_number()
Event Processor
WallLeftWhiteRightWhite
<cls>.left_wall = fake_white
<cls>.right_wall = fake_white
WallLeftWhiteRightBlack
<cls>.left_wall = fake_white
<cls>.right_wall = fake_black
WallLeftBlackRightWhite
<cls>.left_wall = fake_black
<cls>.right_wall = fake_white
WallLeftBlackRightBlack
<cls>.left_wall = fake_black
<cls>.right_wall = fake_black
cell = Cell()
cell.start_at(white)
cell = Wall()
cell.start_at(wall_white)
Next /
[cell.left.color == 'black']
white
entry / cell.color = 'white'
fake_white
entry / wall.color = 'white'
Next / {}
fake_black
entry / wall.color = 'black'
Next / {}
Next
[cell.right.color == 'white' and
cell.left.color == 'black']
[cell.right.color == 'black' and
cell.left.color == 'white']
black
entry / cell.color = 'black'
Rule30
TwoDCellularAutomata
Wall
Canvas