

```

# Tobias cgol

import random as r
import time


# the game board
def buildBoard(numRows, numCols):
    board = []
    for x in range(numRows):
        row = []
        for y in range(numCols):
            row.append(randState())
        board.append(row)
    return board


def randState():
    v = r.random()
    prob = 0.8

    if(v > prob):
        return 1
    else:
        return 0


# print the board
def printBoard(board):
    alive = "X"
    dead = "O"

    for row in board:
        rowString = ""

```

```

    for col in row:
        if col == 0:
            rowString += dead
        else:
            rowString += alive
    print(rowString)

# separate boards
print()
print("-----")
print()

# set each cell
def setCell(b, r, c, v):
    b[r][c] = v

# check neighboring cells
def checkNeighbors(b, r, c):
    count = 0
    for i in range(r-1, r+2, 1):
        for j in range(c-1, c+2, 1):

            # only check valid indices and don't count cell being checked
            if validIndex(b, i, j) and not(i == r and j == c):
                count += b[i][j]
    return count

# is the index valid
def validIndex(l, r, c):
    if 0 <= r < len(l) and 0 <= c < len(l[r]):

```

```
try:
    l[r][c]
    return True
except IndexError:
    return False
else:
    return False
```

```
def getNewCell(l, r, c):
    newValue = l[r][c]
    count = checkNeighbors(l, r, c)
    if newValue:
        if count < 2 or count > 3:
            newValue = 0
    else:
        if count == 3:
            newValue = 1
    return newValue
```

```
def getNewBoard(l):
    newBoard = [x[:] for x in l]
    for r in range(len(l)):
        for c in range(len(l[r])):
            newBoard[r][c] = getNewCell(l, r, c)
    return newBoard
```

```
board = buildBoard(10, 10)
while True:
    printBoard(board)
```

```
board = getNewBoard(board)
```

```
time.sleep(1)
```

```
#OMG it works!
```