import random

class Cell:

# creates data fields for cell

def \_\_init\_\_ (self, row, col, value, flipped):

self.row = row

self.col = col

self.value = value

self.flipped = flipped

#defines flipped/clicked

def flip(self):

self.flipped = True

#allows to view data field of value

#can’t be manipulated

def display(self):

return self.value

# creates data field for grid

class Grid:

grid= [data type]

def \_\_init\_\_ (self, num\_rows, num\_cols, num\_mines,):

self.num\_rows = num\_rows

self.num\_cols = num\_cols

self.num\_mines = num\_mines

def create\_grid(self):

for row in range(self.num\_rows):

self.grid.append([])

for col in range(self.num\_cols):

self.grid[row].append(Cell(row, col, ' ', False))

def place\_mines(self):

for m in range(self.num\_mines):

mine\_row = random.randrange(self.num\_rows)

mine\_col = random.randrange(self.num\_cols)

# prevent any mines from overlapping

while (self.grid[mine\_row][mine\_col].value == '\*'):

mine\_row = random.randrange(self.num\_rows)

mine\_col = random.randrange(self.num\_cols)

self.grid[mine\_row][mine\_col].value = '\*' #assigns value ‘\*’ if cell has mine

def assign\_value(self):

for row in range(self.num\_rows):

for col in range(self.num\_cols):

if (self.grid[row][col].value != '\*'):

num\_border\_mines = 0

ur = row - 1

cr = row

dr = row + 1

lc = col - 1

cc = col

rc = col + 1

if row >= 1 and col >= 1:

if (self.grid[ur][lc].value == '\*'):

num\_border\_mines += 1

if row >= 1:

if (self.grid[ur][cc].value == '\*'):

num\_border\_mines += 1

if row >= 1 and col < self.num\_cols - 1:

if (self.grid[ur][rc].value == '\*'):

num\_border\_mines += 1

if col >= 1:

if (self.grid[cr][lc].value == '\*'):

num\_border\_mines += 1

if col < self.num\_cols - 1:

if (self.grid[cr][rc].value == '\*'):

num\_border\_mines += 1

if row < self.num\_rows - 1 and col >= 1:

if (self.grid[dr][lc].value == '\*'):

num\_border\_mines += 1

if row < self.num\_rows - 1:

if (self.grid[dr][cc].value == '\*'):

num\_border\_mines += 1

if row < self.num\_rows - 1 and col < self.num\_cols - 1:

if (self.grid[dr][rc].value == '\*'):

num\_border\_mines += 1

if num\_border\_mines != 0:

self.grid[row][col].value = str(num\_border\_mines)

#code portion doesn’t quite work as planned Maybe????

def flip\_unassigned\_tiles(self):

for row in range(self.num\_rows):

for col in range(self.num\_cols):

if (self.grid[row][col].value == ' '):

self.grid[row][col].flip()

def hit\_mine(self):

print("GAME OVER!")

for row in range(self.num\_rows):

for col in range(self.num\_cols):

if (self.grid[row][col].value == '\*'):

self.grid[row][col].flip()

def win(self):

#if all assigned and unassigned tiles are flipped except mines then you win

#must be called after all clicks

for row in range(self.num\_rows):

for col in range(self.num\_cols):

if (not (self.grid[row][col].value == '\*' or self.grid[row][col].flipped)):

return False

return True

def display\_grid(self):

# called after every click and reupdates the grid

for row in range(self.num\_rows):

print('|', end='') #makes it so “|” goes to |||||| and not “|” in vertical line

for col in range(self.num\_cols):

if (self.grid[row][col].flipped):

print(self.grid[row][col].display(), end='|')

else:

print('#', end='|')

print('')

def play\_game(self):

self.create\_grid()

self.place\_mines()

self.assign\_value()

while (True):

self.display\_grid()

input\_col = int(input('Enter a Column: '))

input\_row = int(input('Enter a Row: '))

print('=' \* 25)

self.grid[input\_row][input\_col].flip()

if (self.grid[input\_row][input\_col].value == ' '):

self.flip\_unassigned\_tiles()

if (self.grid[input\_row][input\_col].value == '\*'):

self.hit\_mine()

self.display\_grid()

break

if (self.win()):

print('You Win!')

grid = Grid(9, 9, 10)

grid.play\_game()