Activity 2: Fill Algorithms

Contributors include Keane Dalisay, Nel Alanan and Prince Alexander Malatuba.

🎉 Cheers!

Using numpy.rand.randint()

This is to create a random array given a `max_range` integer as basis for the highest integer inside the array.

```
class createPolygon:
    def __init__(self, max_range, rows, columns):
        self.two_d_arr = np.random.randint(max_range, size=(rows, columns))
```

```
print('\nNumber of rows for polygon...')
rows = int(input(': '))

print('\nNumber of columns for polygon...')
columns = int(input(': '))

choice = True
while choice:
    print('\nMax value range to fill polygon...')
    max_range = int(input(': '))

    polygon = createPolygon(max_range, rows, columns)
```

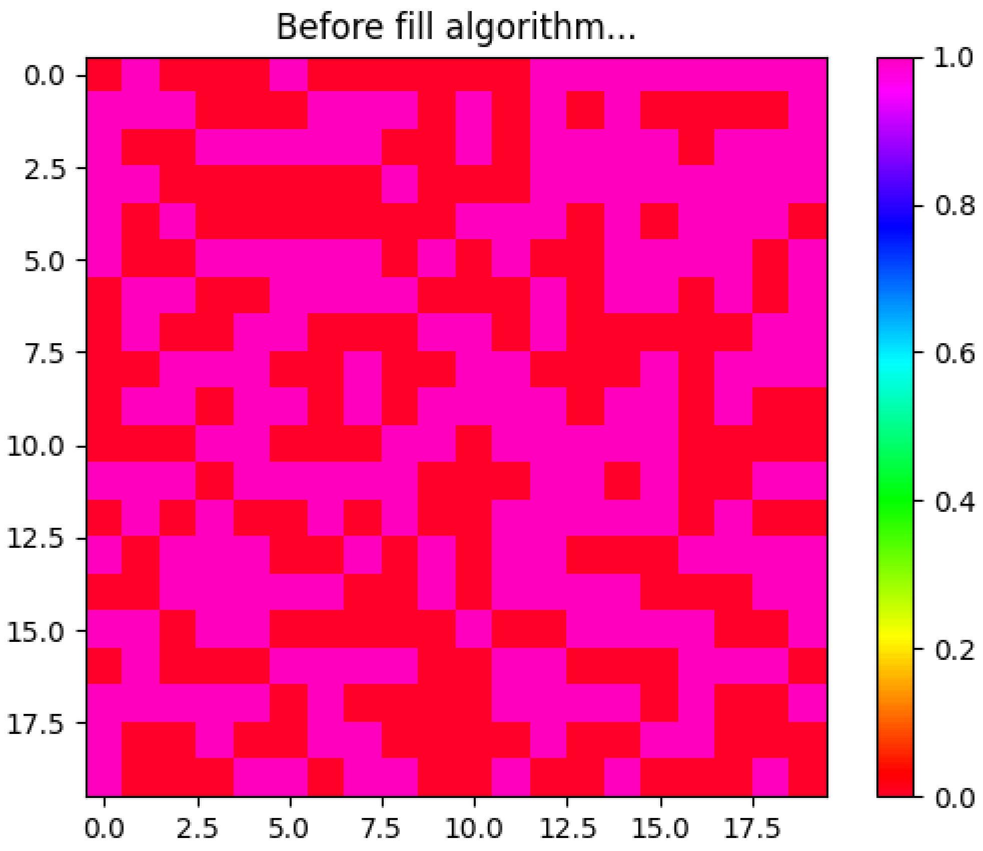
Boundary-Fill Algorithm

Fills all empty blocks inside a single colored border.

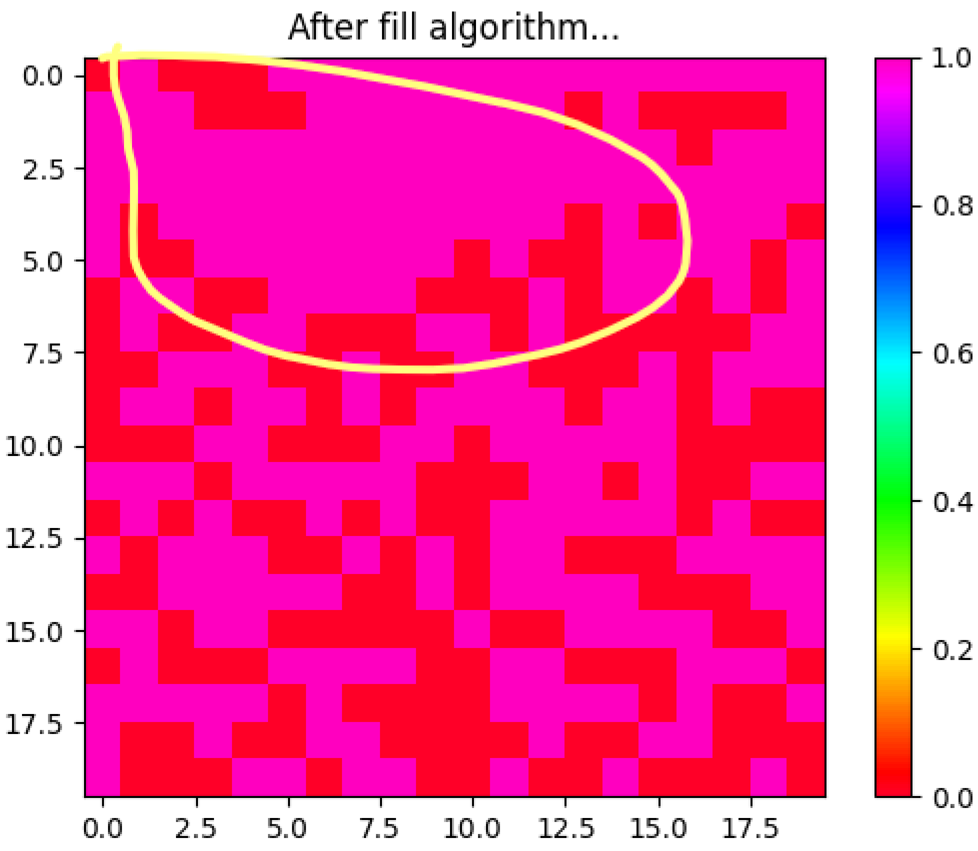
```
def boundaryFill(self, row, column):
    arr_dim = self.two_d_arr.shape
    if  row >= arr_dim[0] or  row < 0 : return
    if  column >= arr_dim[1] or column < 0: return

    value = self.two_d_arr[row][column]
    if value == 1: return
    # Check if target block is already assigned 1
    # This denotes the border to stop filling

    self.two_d_arr[row][column] = 1
    # Assign target block with 1
    # If cmap is gray_r, 1 is equivalent to black

    self.boundaryFill(row - 1, column) # Top block
```

```
        self.boundaryFill(row, column + 1) # Right block
        self.boundaryFill(row + 1, column) # Bottom block
        self.boundaryFill(row, column - 1) # Left block
        return
```

Before

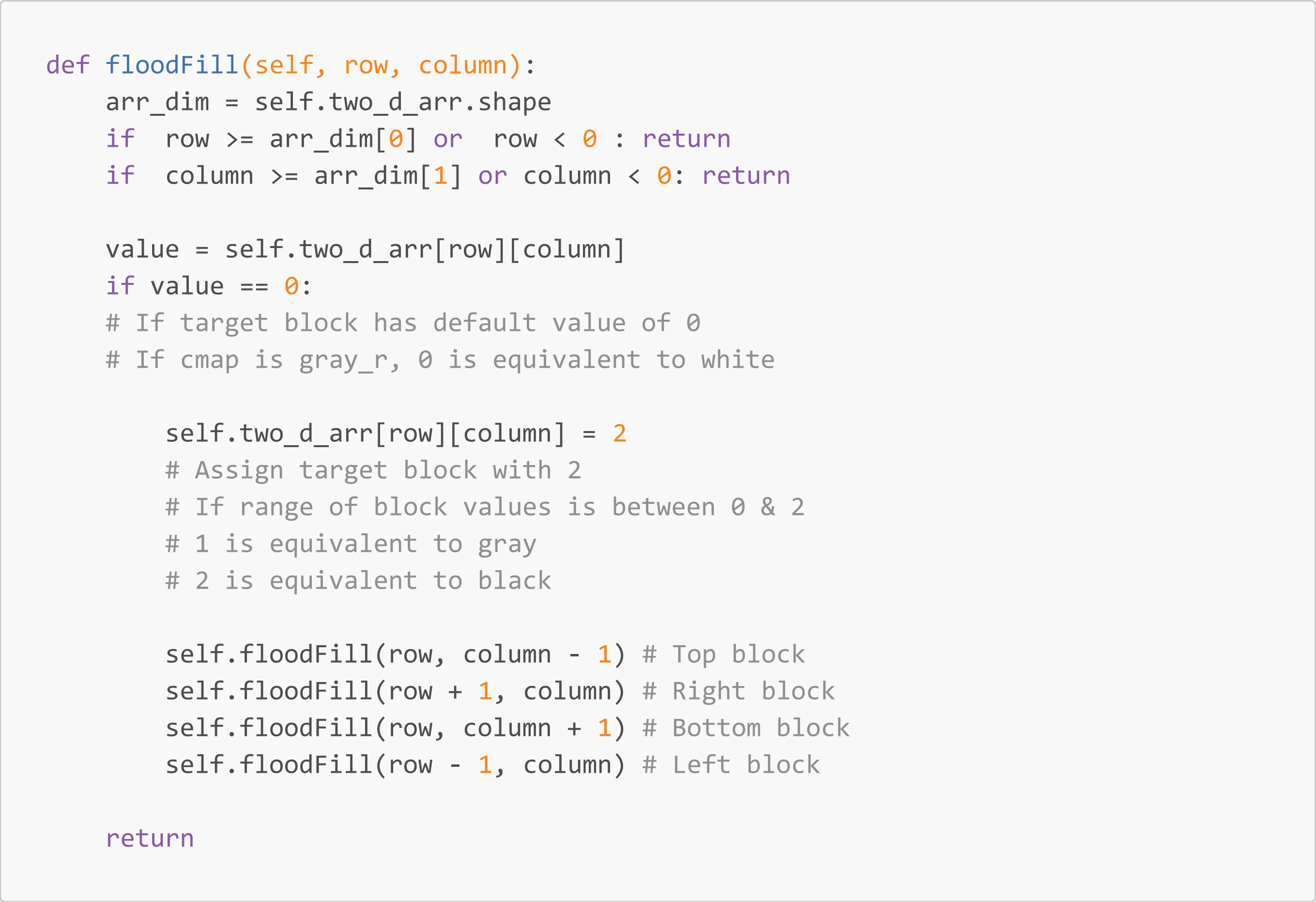


After



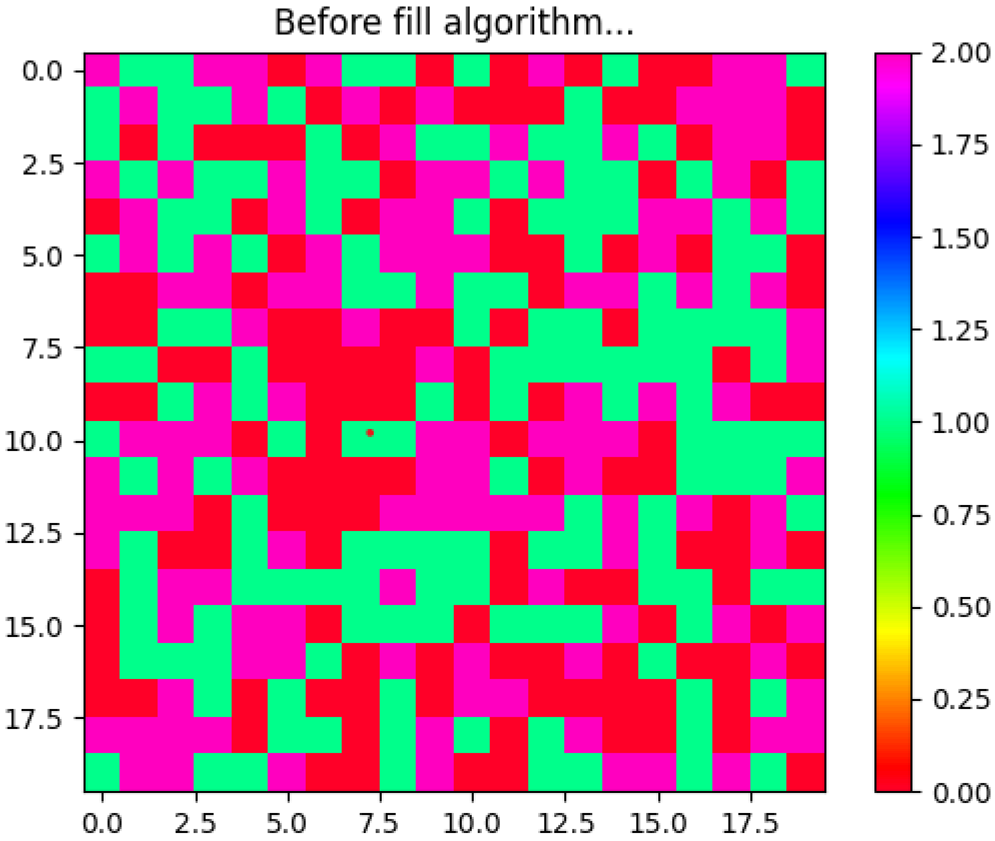
Flood-Fill Algorithm

Fills all empty blocks inside a multi-colored border.

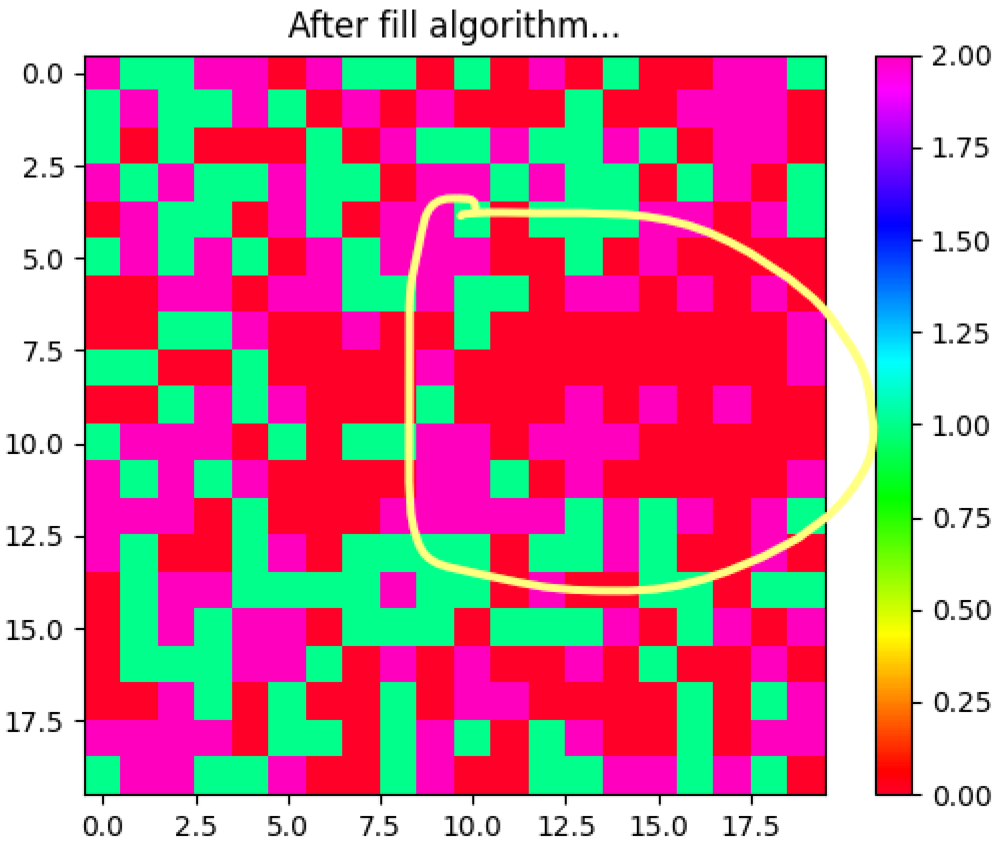


```
def floodFill(self, row, column):
    arr_dim = self.two_d_arr.shape
    if  row >= arr_dim[0] or  row < 0 : return
    if  column >= arr_dim[1] or column < 0: return

    value = self.two_d_arr[row][column]
    if value == 0:
    # If target block has default value of 0
    # If cmap is gray_r, 0 is equivalent to white

        self.two_d_arr[row][column] = 2
        # Assign target block with 2
        # If range of block values is between 0 & 2
        # 1 is equivalent to gray
        # 2 is equivalent to black

        self.floodFill(row, column - 1) # Top block
        self.floodFill(row + 1, column) # Right block
        self.floodFill(row, column + 1) # Bottom block
        self.floodFill(row - 1, column) # Left block

    return
```

Before



After


After fill algorithm...
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00