COP3502 - 9/13/21 Floodfill

Click pixel
fill region new color.

y →
x ↓
mc - 1 do
this weird!

```
// R x C grid, grid
// Don't fill if already filled, Don't send bogs == -1
floodfill (int x, int y, int color) {
    if(!inbounds(x,y)) return;
    if(grid[x][y] == -1) return; if (grid[x][y] == color) return;
    grid[x][y] = color;
    for (i = 0; i < NUMDIR; i++) {
        floodfill(x+DX[i], y+DY[i], color);
    }
}
```

fewer base case add if stmt.

DX, DY array

optional

```
const int DX[4] = {-1, 0, 0, 1};
const int DY[4] = {0, -1, 1, 0};
// for up down left, right
```

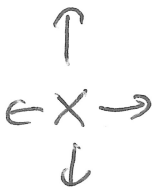
If I am at (x, y), I could go to (x-1, y), (x, y-1), (x, y+1) or (x+1, y)

```
// i is direction
newx = x + DX[i];
newy = y + DY[i];
```

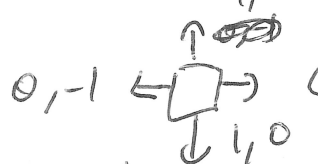
COP 3502 9/13/21 Floodfill

1) Click pixel
2) fill region (until boundary) w/a diff color.

DX/DY array




```
const int DX[4] = {-1, 0, 0, 1};
const int DY[4] = { 0,-1, 1, 0};
#define NUMWAYS 4
```




```
// (x,y) is where I am
for (int i=0; i<NUMWAYS; i++) {

    int newx = x + DX[i];
    int newy = y + DY[i];

    { process (newx, newy);

}
```

```
floodfill(int x, int y, int color) {
    if(!inbounds(x,y)) return;
    if( grid[x][y] == color) return;
    grid[x][y]
    if (grid[x][y] == -1) return;
    grid[x][y] = color;
    for (int i=0; i<NumWays; i++)
        floodfill(x+DX[i], y+DY[i], color);
}
```

Could have added an if and removed a base case.