

```
1 - Create Turkmite class, ctor takes image, Step() method (image as dependency injection)
2
3 - Color constants
4
5 (commit)
6
7 - Direction with modulo: (d + 4) % 4
8 - Direction with dxy from array, X += ..., X and Y limiter with Min, Max
9 - extract Step() method
10 - extract: private (Vec3b newColor, int turn) GetStep(Vec3b
    currentColor)
11
12     public void Step()
13     {
14         Vec3b currentColor = Indexer[Y, X];
15
16         (Vec3b newColor, int turn) = GetStep(currentColor);
17         Indexer[Y, X] = newColor;
18         direction += turn;
19
20         direction = (direction+4) % 4;
21
22         (int, int)[] dxy = [(0, -1), (1, 0), (0, 1), (-1, 0)];
23         (var dx, var dy) = dxy[direction];
24         X += dx;
25         Y += dy;
26         X = Math.Min(Math.Max(X, 0), 199);
27         Y = Math.Min(Math.Max(Y, 0), 199);
28     }
29
30     private (Vec3b newColor, int turn) GetStep(Vec3b currentColor)
31     {
32         return currentColor == black ? (white, 1) : (black, -1);
33     }
34
35 (commit)
36
37 - Create base class for turkmites, original turkmite into BasicTurkmite
38 - Create unit test for BasicTurkmite using a BasicTurkmiteMock
39
40 (commit)
41
42 - Create a new turkmite with a different color
43
44         return currentColor == black ? (red, 1) :
            ( currentColor==red ? (yellow, -1) : (black, -1));
45
46 - Add PreferredStepCount
47
48     public abstract int PreferredStepCount { get; }
49
50 (commit)
```