Procedural Town

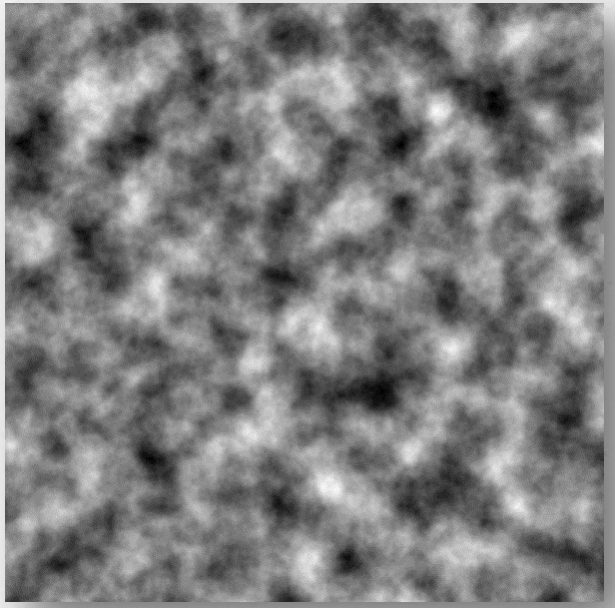
Création d'une ville de manière procédurale

Elric Mathure – Vincent Langlois

Création d'un « perlin noise » :

```
mapgrid = new int[mapWidth, mapHeight];

//generation donnée map
for (int h = 0; h < mapHeight; h++)
{
    for (int w = 0; w < mapWidth; w++)
    {
        mapgrid[w, h] = (int) (Mathf.PerlinNoise(w / 10.0f, h / 10.0f) * 10);
    }
}
```



× 10 =

10	6	7	10	9	9	8	7	6	6
9	5	6	10	10	7	7	6	6	5
9	9	9	10	10	8	6	5	5	4
8	9	8	10	9	8	6	5	6	4
8	7	8	9	9	8	6	5	6	3
8	7	8	9	9	8	7	5	4	3
7	6	8	9	9	8	7	4	3	3
7	6	8	6	7	6	5	4	2	2
7	5	8	6	7	2	3	3	1	1
7	6	6	6	5	1	1	2	1	1

Création des rues et carrefours :

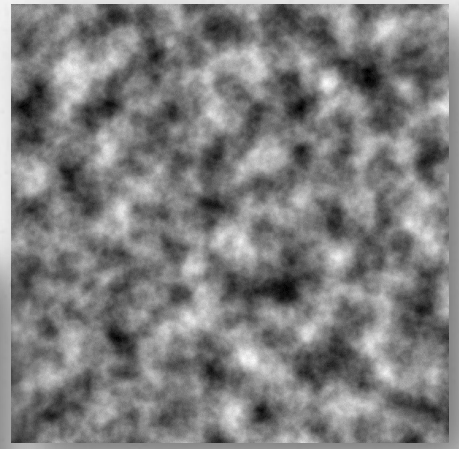
```
//build streets
int x = 0;
for(int n = 0; n < 50; n++)
{
    for (int h = 0; h < mapHeight; h++)
    {
        mapgrid[x, h] = -1;
    }

    x += Random.Range(3, 3);
    if (x >= mapWidth) break;
}



int z = 0;
for(int n = 0; n < 10; n++)
{
    for (int w = 0; w < mapWidth; w++)
    {
        if (mapgrid [w, z] == -1)
            mapgrid [w, z] = -3;
        else
            mapgrid [w, z] = -2;
    }

    z += Random.Range(2, 20);
    if (z >= mapHeight) break;
}
```



10	6	7	10	9	9	8	7	6	6
9	5	6	10	10	7	7	6	6	5
9	9	9	10	10	8	6	5	5	4
8	9	8	10	9	8	6	5	6	4
8	7	8	9	9	8	6	5	6	3
8	7	8	9	9	8	7	5	4	3
7	6	8	9	9	8	7	4	3	3
7	6	8	6	7	6	5	4	2	2
7	5	8	6	7	2	3	3	1	1
7	6	6	6	5	1	1	2	1	1

10	6	7	10	-2	9	8	7	6	6
-1	-1	-1	-1	-3	-1	-1	-1	-1	-1
9	9	9	10	-2	8	6	5	5	4
-1	-1	-1	-1	-3	-1	-1	-1	-1	-1
8	7	8	9	-2	8	6	5	6	3
8	7	8	9	-2	8	7	5	4	3
7	6	8	9	-2	8	7	4	3	3
7	6	8	6	-2	6	5	4	2	2
7	5	8	6	-2	2	3	3	1	1
7	6	6	6	-2	1	1	2	1	1

Génération des « blocs » de la ville :

```
//Génération city
for (int h = 0; h < mapHeight; h++)
{
    for(int w = 0; w < mapWidth; w++)
    {
        int result = mapgrid[w, h];
        Vector3 pos = new Vector3(w * buildingFootprint, 0, h * buildingFootprint);


        System.Random rnd = new System.Random();
        angleChoisi = rnd.Next(0, 4);



        if (result < -2)
            Instantiate(crossroad, pos, crossroad.transform.rotation);
        else if (result < -1)
            Instantiate(xstreets, pos, xstreets.transform.rotation);
        else if (result < 0)
            Instantiate(zstreets, pos, zstreets.transform.rotation);


        else if (result < 1)
            Instantiate(buildings[0], pos, Quaternion.Euler(new Vector3(0, angles[angleChoisi], 0)));
        else if (result < 2)
            Instantiate(buildings[1], pos, Quaternion.Euler(new Vector3(0, angles[angleChoisi], 0)));
        else if (result < 4)
            Instantiate(buildings[2], pos, Quaternion.Euler(new Vector3(0, angles[angleChoisi], 0)));
        else if (result < 6)
            Instantiate(buildings[3], pos, Quaternion.Euler(new Vector3(0, angles[angleChoisi], 0)));
        else if (result < 7)
            Instantiate(buildings[4], pos, Quaternion.Euler(new Vector3(0, angles[angleChoisi], 0)));
        else if (result < 10)
            Instantiate(buildings[5], pos, Quaternion.Euler(new Vector3(0, 0, 0)));
```

Problèmes rencontrés :

- Orientation des « blocs »
- Orientation des routes
- Dimensions des maisons