

GÉNÉRATION PROCÉDURALE

Ismail Aïssi et Daniel Zhu



Introduction

Comment générer aléatoirement et de manière satisfaisante du contenu pour les jeux vidéo ?

Sommaire

I - Donjons

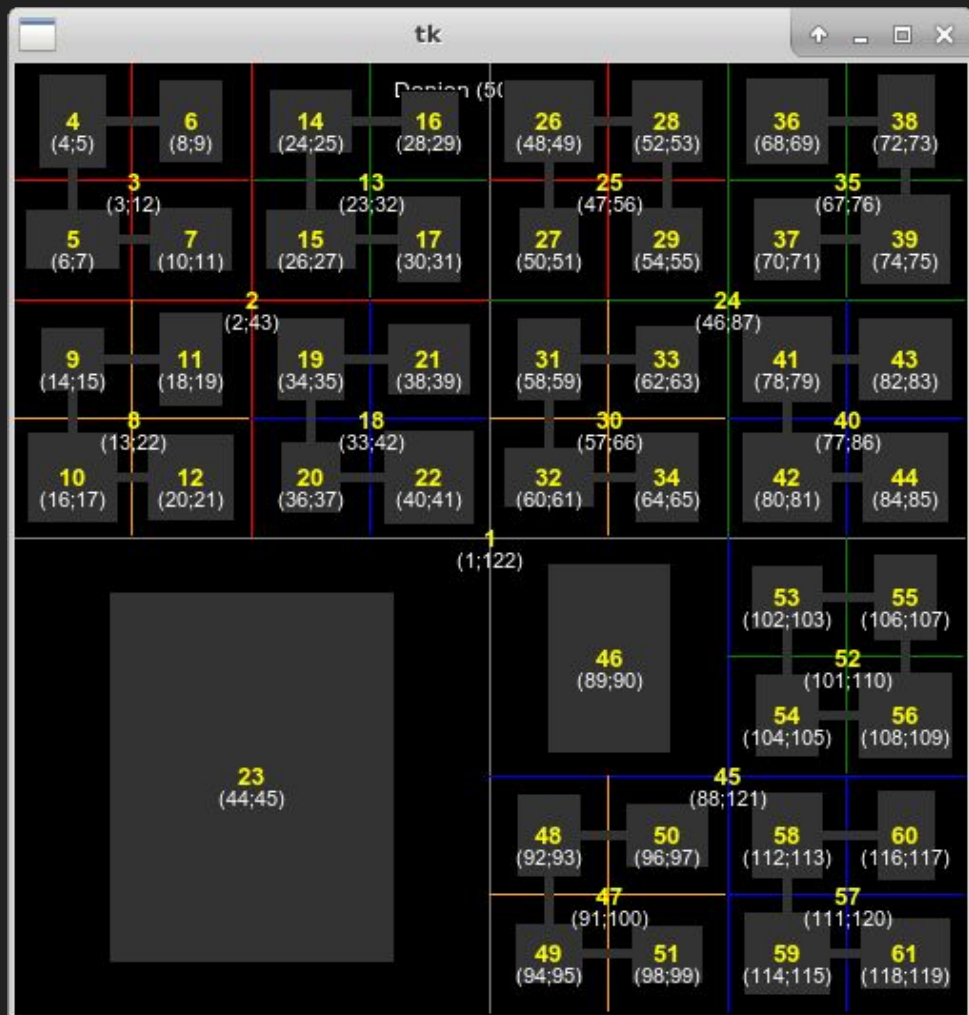
- Méthode du *quadtree*
- L'exemple de *Spelunky*

II - Reliefs montagneux

- L'algorithme du diamant-carré
- Implémentation

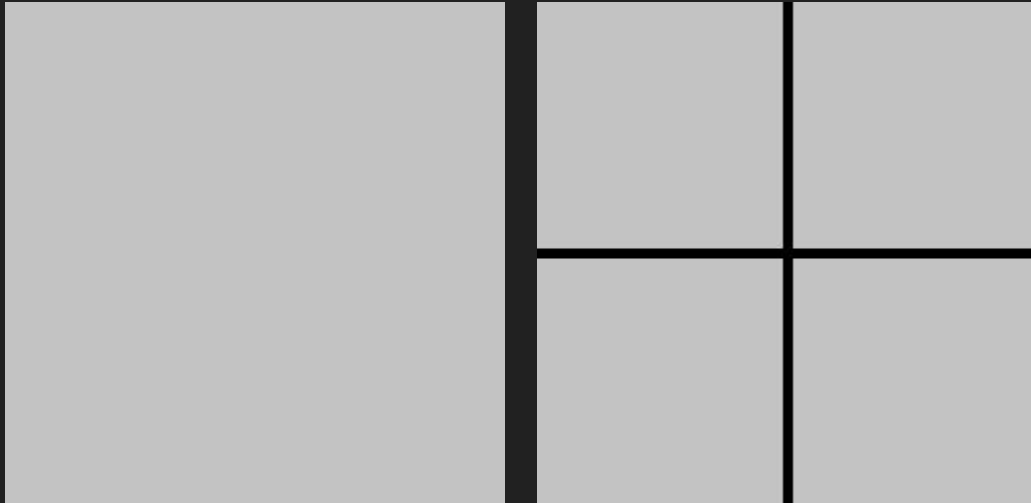
III - Cohérence

- Étude statistique
- Vérification algorithmique



I - Donjon > Méthode du *quadtree*

Partitionnement spatial (1/3)



(i) Début

(i) Partitionnement

I - Donjons

— **Méthode du *quadtree***

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

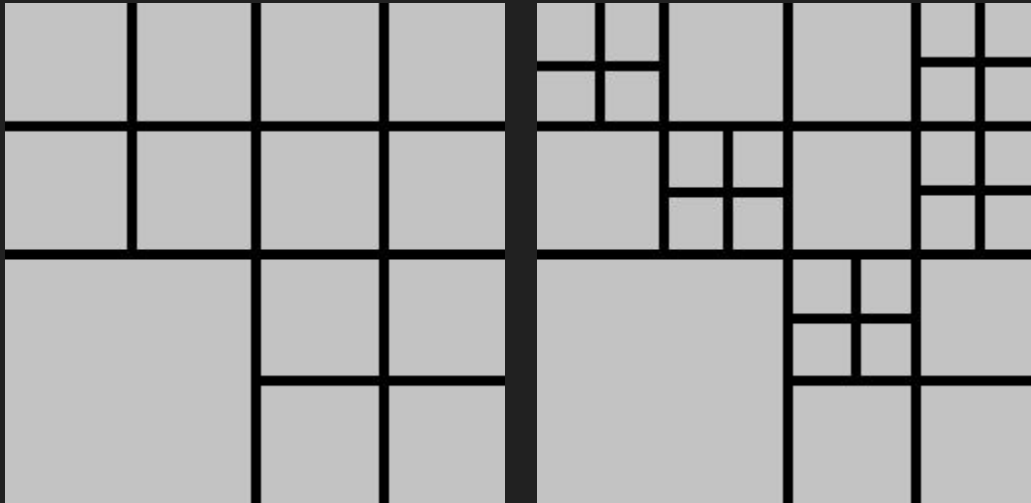
III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > Méthode du *quadtree*

Partitionnement spatial (2/3)



(iii) Sous-partitionnement
de certaines partitions
aléatoirement choisies

(iv) Et ainsi de suite
jusqu'à un certain niveau

I - Donjons

— **Méthode du *quadtree***

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

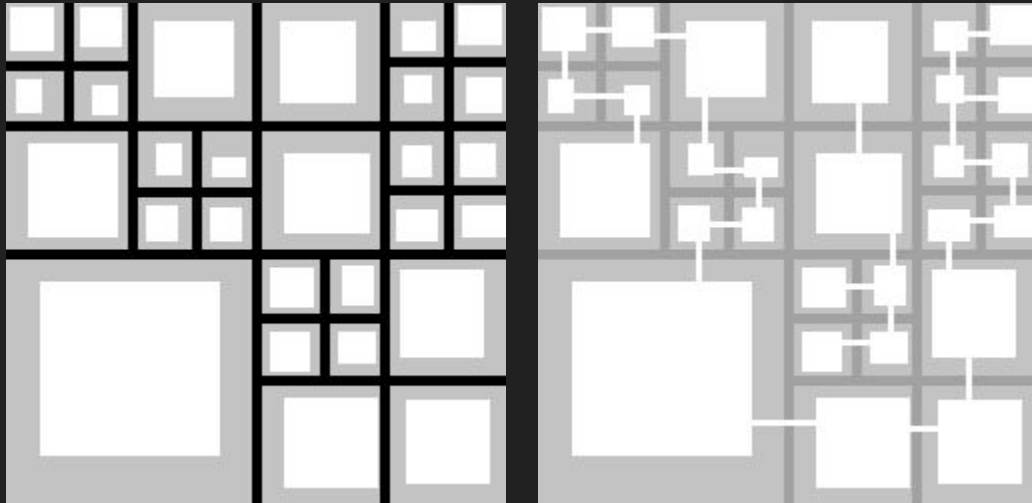
III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > Méthode du *quadtree*

Partitionnement spatial (3/3)



(v) Création des salles
aux dimensions
aléatoires

(iv) Puis on relie les
salles aléatoirement

I - Donjons

— **Méthode du *quadtree***

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

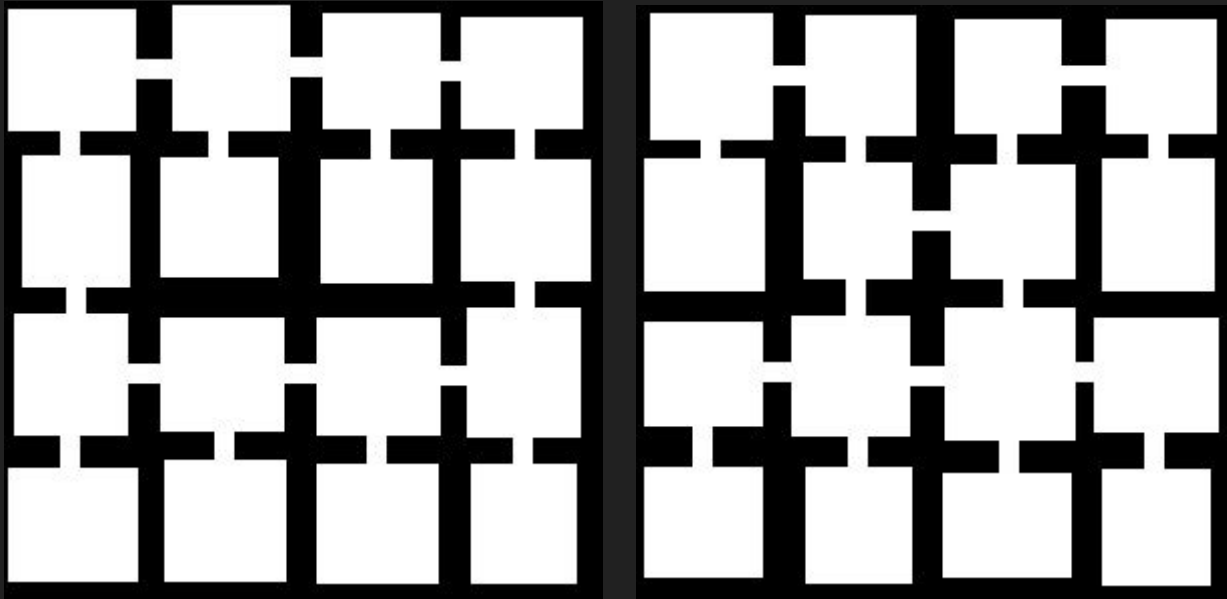
III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > Méthode du *quadtree*

Exemples (1/2)



I - Donjons

— **Méthode du *quadtree***

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

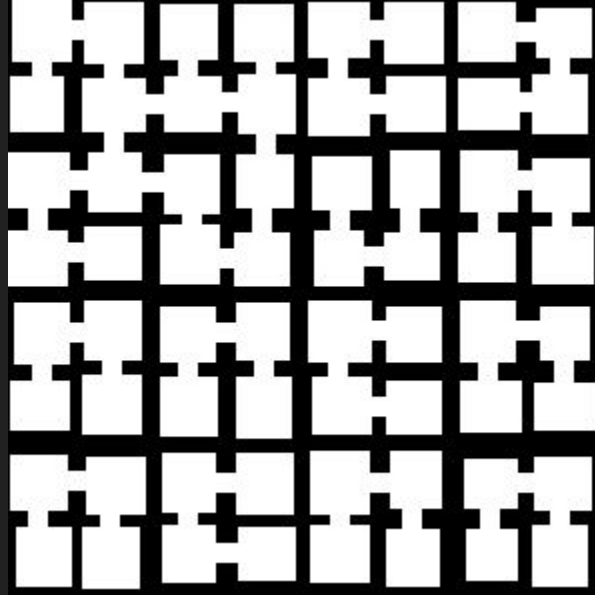
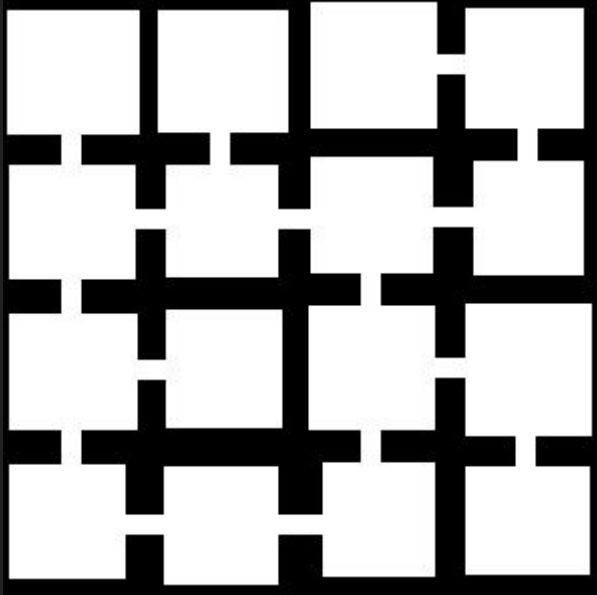
III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > Méthode du *quadtree*

Exemples (2/2)



Pour des donjons trop profonds, on ne parvient pas à relier chaque partition

I - Donjons

— **Méthode du *quadtree***

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*



- I - Donjons
 - Méthode du *quadtree*
 - **L'exemple de *Spelunky***
- II - Reliefs montagneux
 - L'algorithme du diamant-carré
 - Implémentation
- III - Cohérence
 - Étude statistique
 - Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Exemple : création de 4 salles > Le donjon est créé à l'aide d'une matrice dont tous les coefficients sont initialement nuls.

$$\begin{bmatrix} [0 & 0 & 0] \\ [0 & 0 & 0] \\ [0 & 0 & 0] \end{bmatrix}$$

I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

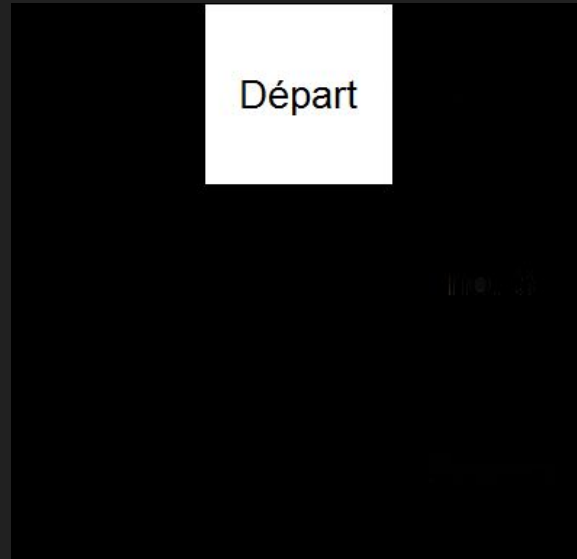
III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

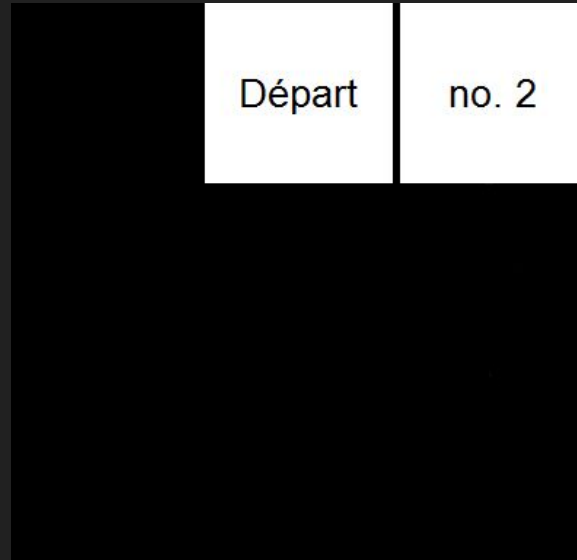
Exemple : création de 4 salles > Placement aléatoire d'un point de départ dans la matrice.

$$\begin{bmatrix} [0 & 1 & 0] \\ [0 & 0 & 0] \\ [0 & 0 & 0] \end{bmatrix}$$


- I - Donjons
 - Méthode du *quadtree*
 - **L'exemple de *Spelunky***
- II - Reliefs montagneux
 - L'algorithme du diamant-carré
 - Implémentation
- III - Cohérence
 - Étude statistique
 - Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Exemple : création de 4 salles > Placement de la seconde salle dans une case voisine (non diagonale)

$$\begin{bmatrix} [0 & 1 & 2] \\ [0 & 0 & 0] \\ [0 & 0 & 0] \end{bmatrix}$$


- I - Donjons
 - Méthode du *quadtree*
 - **L'exemple de *Spelunky***
- II - Reliefs montagneux
 - L'algorithme du diamant-carré
 - Implémentation
- III - Cohérence
 - Étude statistique
 - Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Exemple : création de 4 salles > Ainsi de suite...

[[0 1 2]
[0 0 3]
[0 0 4]]



I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

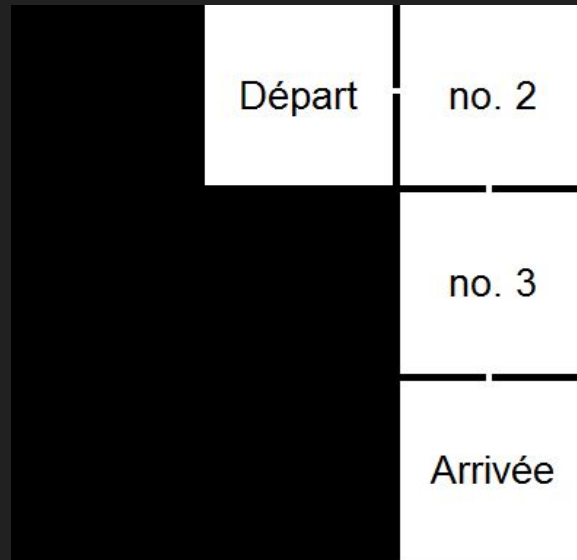
— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Exemple : création de 4 salles > Puis on relie les salles.

```
[ [ 0  1  2 ]  
  [ 0  0  3 ]  
  [ 0  0  4 ] ]
```



I - Donjons
— Méthode du *quadtree*
— **L'exemple de *Spelunky***

II - Reliefs montagneux
— L'algorithme du diamant-carré
— Implémentation

III - Cohérence
— Étude statistique
— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Que faire si on est bloqué : exemple avec 5 salles

$$\begin{bmatrix} [0 & \mathbf{1} & 0] \\ [0 & 0 & 0] \\ [0 & 0 & 0] \end{bmatrix}$$

Voisins :

[(1, 1) ,
(1, 3) ,
(2, 2)]

Interdits :

[]

Mémoire :

[(1, 2) ,]

I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Que faire si on est bloqué : exemple avec 5 salles

$$\begin{bmatrix} [0 & 1 & 0] \\ [0 & \mathbf{2} & 0] \\ [0 & 0 & 0] \end{bmatrix}$$

Voisins :

[(2, 1) ,
(2, 3) ,
(3, 2)]

Interdits :

[]

Mémoire :

[(1, 2) ,
(2, 2)]

I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Que faire si on est bloqué : exemple avec 5 salles

$$\begin{bmatrix} [0 & 1 & 0] \\ [3 & 2 & 0] \\ [0 & 0 & 0] \end{bmatrix}$$

Voisins :

[(1, 1) ,
(3, 1)]

Interdits :

[]

Mémoire :

[(1, 2) ,
(2, 2) ,
(2, 1)]

I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Que faire si on est bloqué : exemple avec 5 salles

Problème !

$$\begin{bmatrix} [4 & 1 & 0] \\ [3 & 2 & 0] \\ [0 & 0 & 0] \end{bmatrix}$$

Voisins :

[]

Interdits :

[]

Mémoire :

[(1, 2),
(2, 2),
(2, 1),
(1, 1)]

I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Que faire si on est bloqué : exemple avec 5 salles

$$\begin{bmatrix} \cancel{4} & 1 & 0 \\ 3 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Voisins :

~~[(1,1),~~
(3,1)]

Interdits :

[(1,1)]

Mémoire :

[(1,2),
(2,2),
(2,1),
~~(1,1)~~]

I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Que faire si on est bloqué : exemple avec 5 salles

$$\begin{bmatrix} [0 & 1 & 0] \\ [3 & 2 & 0] \\ [4 & 0 & 0] \end{bmatrix}$$

Voisins :

[(3, 2)]

Interdits :

[]

Mémoire :

[(1, 2) ,

(2, 2) ,

(2, 1) ,

(3, 1)]

I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Que faire si on est bloqué : exemple avec 5 salles

Terminé !

```
[ [ 0  1  0 ]  
  [ 3  2  0 ]  
  [ 4  5  0 ] ]
```

Voisins :

```
[ ]
```

Interdits :

```
[ ]
```

Mémoire :

```
[ (1, 2),  
  (2, 2),  
  (2, 1),  
  (3, 1),  
  (3, 2) ]
```

I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Exemples (1/2)

	Départ	no. 18	no. 19	Arrivée
no. 3	no. 2	no. 17	no. 16	
no. 4			no. 15	no. 14
no. 5	no. 8	no. 9	no. 10	no. 13
no. 6	no. 7		no. 11	no. 12

no. 10	no. 11	no. 12	no. 13	no. 14
no. 9	no. 8	no. 19	no. 18	no. 15
	no. 7	Arrivée	no. 17	no. 16
	no. 6	no. 5	no. 2	Départ
		no. 4	no. 3	

I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

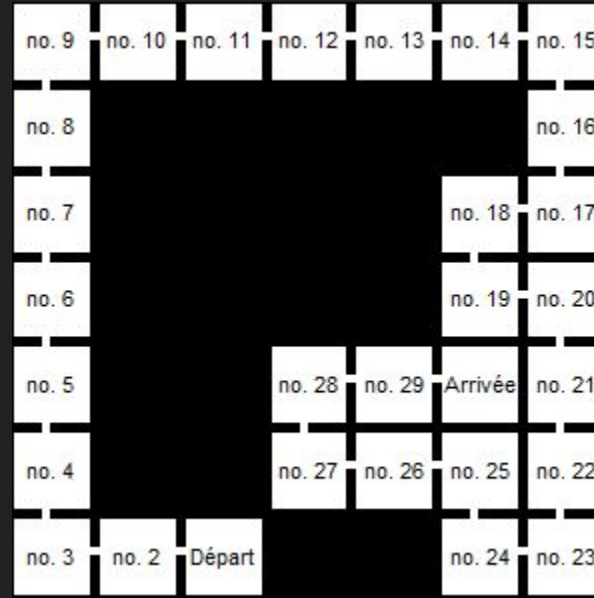
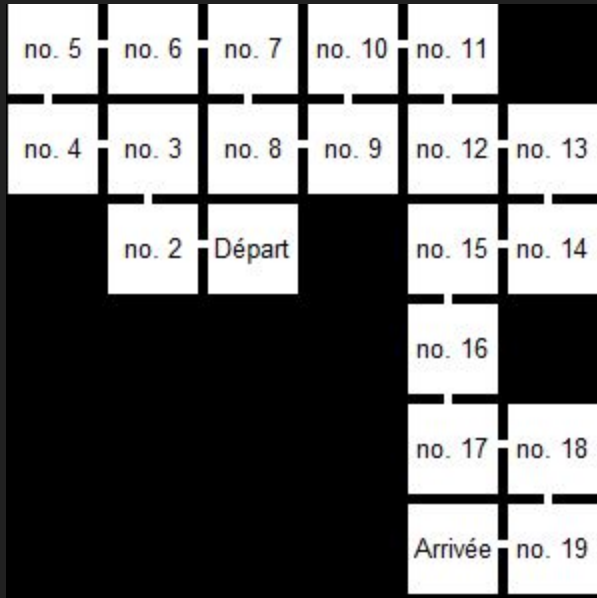
III - Cohérence

— Étude statistique

— Vérification algorithmique

I - Donjon > L'exemple de *Spelunky*

Exemples (2/2)



I - Donjons

— Méthode du *quadtree*

— **L'exemple de *Spelunky***

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— Étude statistique

— Vérification algorithmique

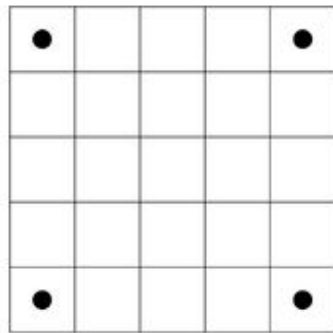
II - Reliefs montagneux > Diamant-carré

L'algorithme - Prérequis, présentation :

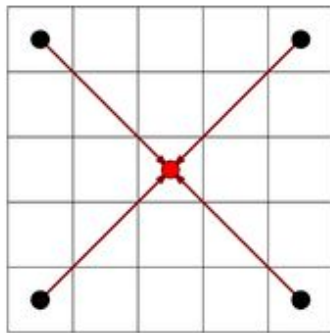
Taille de la matrice : $2^n + 1$

$$M_{x,y} = \frac{M_{1,1} + M_{1,n} + M_{n,n} + M_{n,1}}{4} + R$$

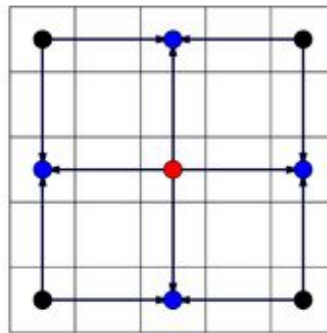
$$R \in] - 1; 1[$$



Initialisation des quatre coins



Phase du diamant avec un pas de 4



Phase du carré avec un pas de 4

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— **L'algorithme du diamant-carré**

— Implémentation

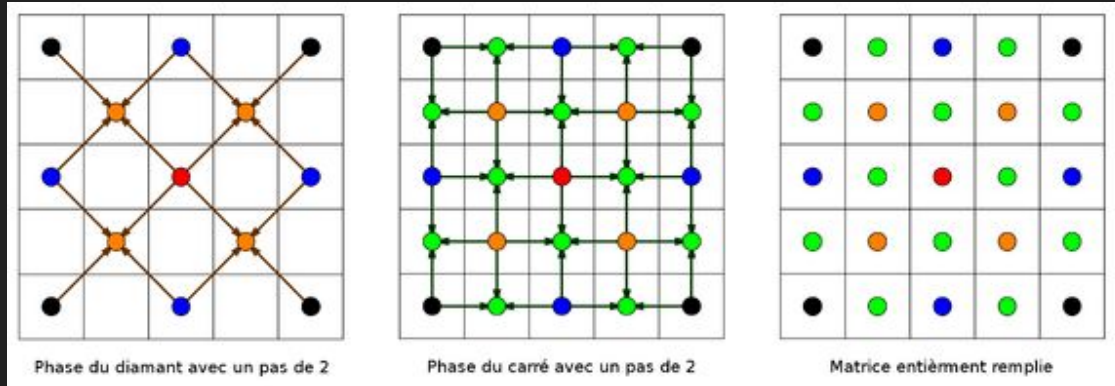
III - Cohérence

— Étude statistique

— Vérification algorithmique

II - Reliefs montagneux > Diamant-carré

L'algorithme - Présentation :



- I - Donjons
 - Méthode du *quadtree*
 - L'exemple de *Spelunky*

- II - Reliefs montagneux
 - **L'algorithme du diamant-carré**
 - Implémentation

- III - Cohérence
 - Étude statistique
 - Vérification algorithmique

II - Reliefs montagneux > Diamant-carré

> Algorithme récursif : `diamant()`, `carre()`, `diamant_carre()`

> Principe du « **diviser pour régner** » sur chaque matrice extraite (non asynchrone)...

[[X X X X X]	[[X X X X X]	[[X X X X X]
[X X X X X]	[X X X X X]	[X X X X X]
[X X X X X]	[X X X X X]	[X X X X X]
[X X X X X]	[X X X X X]	[X X X X X]
[X X X X X]] 1	[X X X X X]] 2	[X X X X X]] 3

[[X X X X X]	[[X X X X X]	[[X X X X X]	
[X X X X X]	[X X X X X]	[X X X X X]	
[X X X X X]	[X X X X X]	[X X X X X]	
[X X X X X]	[X X X X X]	[X X X X X]	
[X X X X X]] 4	[X X X X X]] 5	[X X X X X]] 6	Etc

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— **Implémentation**

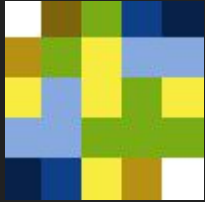
III - Cohérence

— Étude statistique

— Vérification algorithmique

II - Reliefs montagneux > Diamant-carré

Représentation visuelle : palier de hauteur



```
[[ 10.    7.9    5.2    2.2    0.    ]
 [  7.1    5.8    4.7    3.3    3.4    ]
 [  4.7    3.4    4.0    5.6    4.7    ]
 [  3.0    2.6    5.0    5.6    6.0    ]
 [  0.     2.1    4.2    6.2    10.   ]]
```

Palier	Légende	Couleur
0.0	Océan profond	(bleu foncé)
1.25	Océan	(bleu)
2.5	Mer	(bleu clair)
3.75	Plage	(jaune)
5.0	Plaine	(vert)
6.25	Colline	(marron clair)
7.5	Montagne	(marron foncé)
8.75	Neige éternelle	(blanc)

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— **L'algorithme du diamant-carré**

— Implémentation

III - Cohérence

— Étude statistique

— Vérification algorithmique

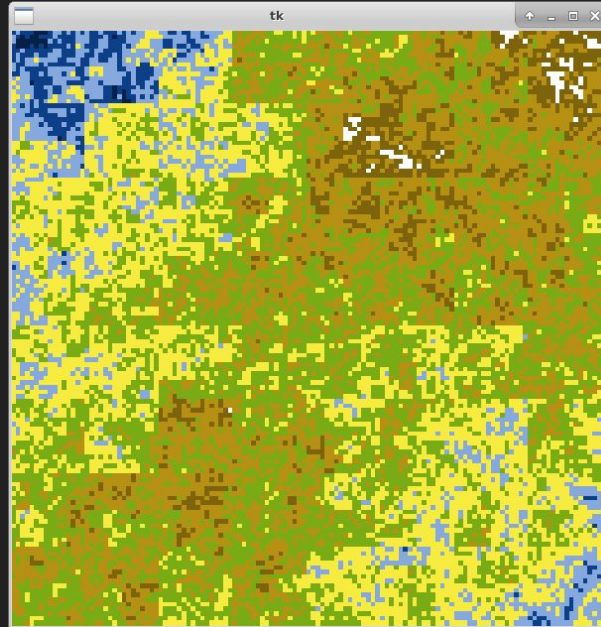
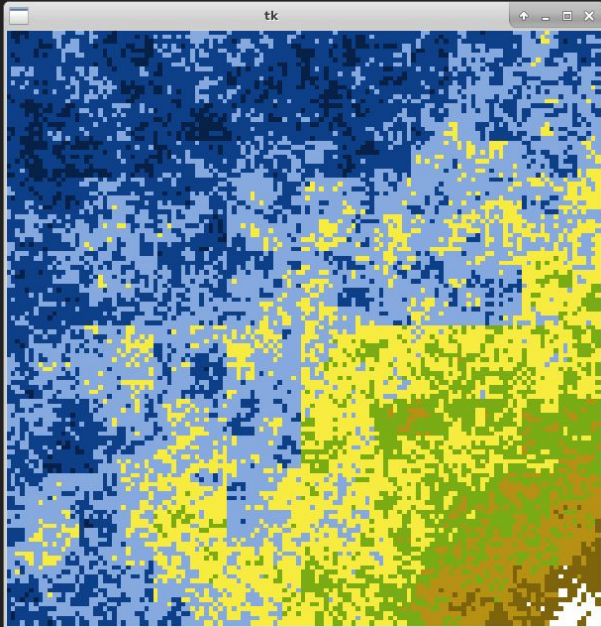
Pas de division :

$$a = \frac{\max M - \min M}{N}$$

Avec $N = 8$ (nombre de couches) et M la matrice.

II - Reliefs montagneux > Diamant-carré

(Algorithme récursif) Échantillons A



- I - Donjons
 - Méthode du *quadtree*
 - L'exemple de *Spelunky*
- II - Reliefs montagneux
 - L'algorithme du diamant-carré
 - **Implémentation**
- III - Cohérence
 - Étude statistique
 - Vérification algorithmique

II - Reliefs montagneux > Diamant-carré

> Sans « lissage » : $R \in]-1;1[$

$$M_{x,y} = \frac{M_{1,1} + M_{1,n} + M_{n,n} + M_{n,1}}{4} + R$$

> Avec « lissage » : $R \in]-1;1[$

$$M_{x,y} = \frac{M_{1,1} + M_{1,n} + M_{n,n} + M_{n,1}}{4} + R \times \text{lissage}$$

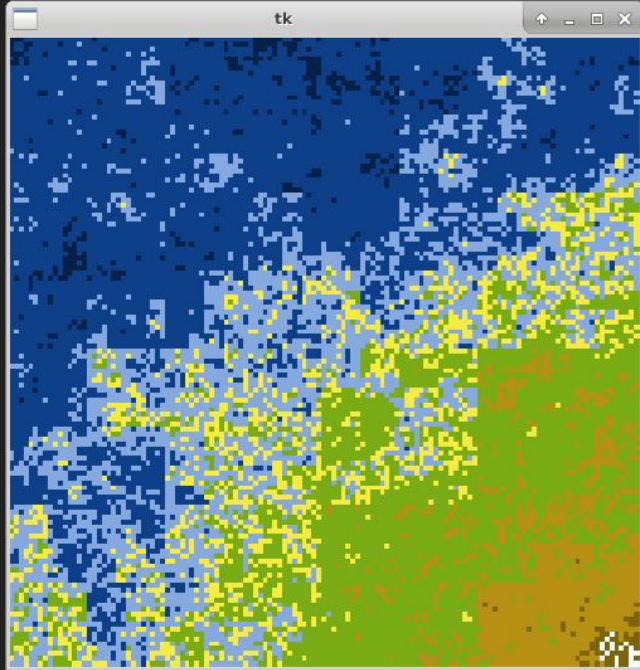
I - Donjons
— Méthode du *quadtree*
— L'exemple de *Spelunky*

II - Reliefs montagneux
— L'algorithme du diamant-carré
— **Implémentation**

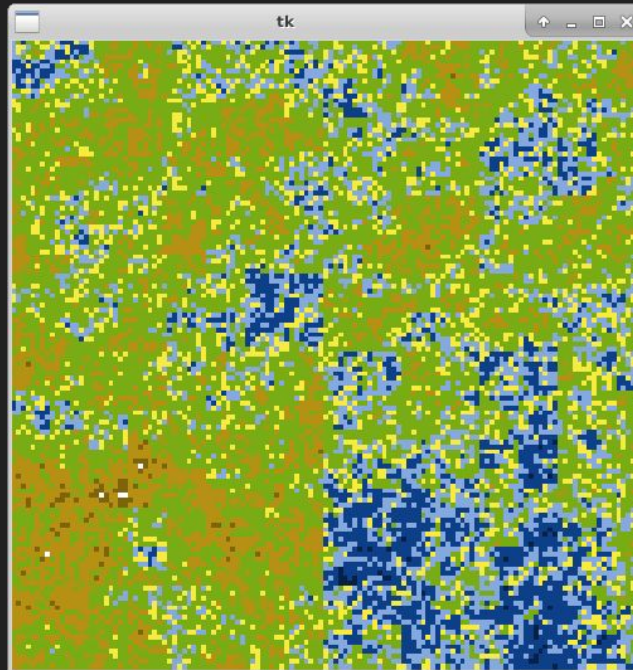
III - Cohérence
— Étude statistique
— Vérification algorithmique

II - Reliefs montagneux > Diamant-carré

(Algorithme récursif) Échantillons B



Lissage : 0.05

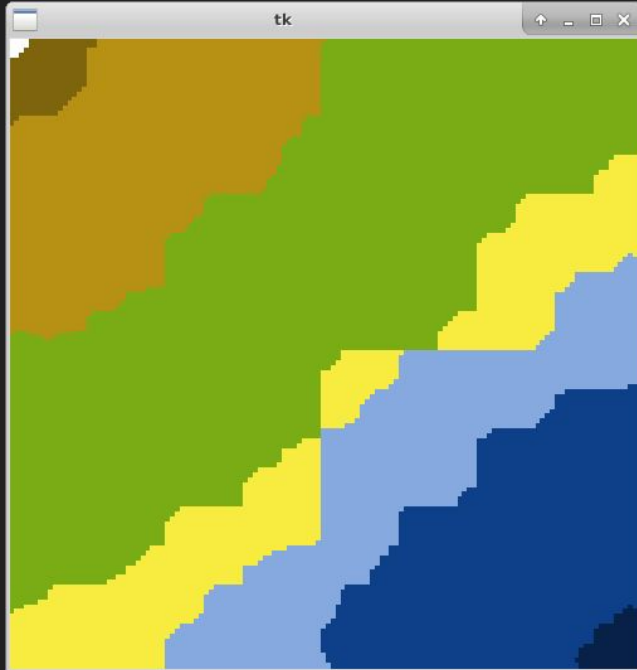


Lissage : 0.2

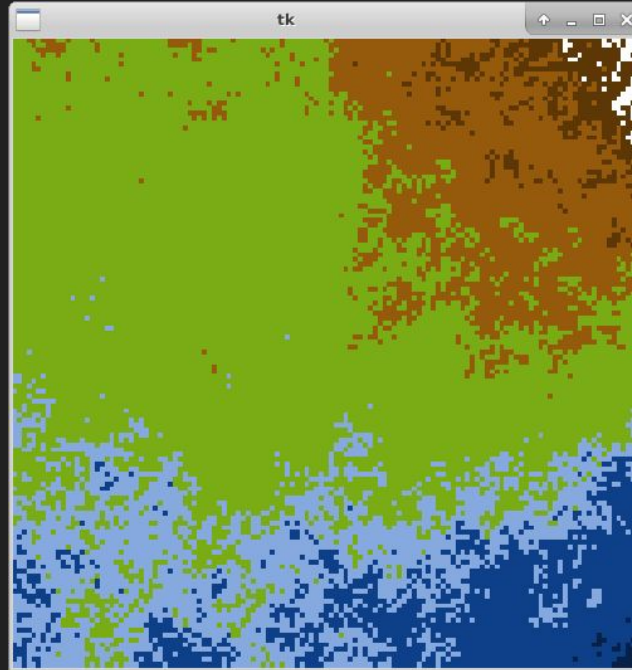
- I - Donjons
 - Méthode du *quadtree*
 - L'exemple de *Spelunky*
- II - Reliefs montagneux
 - L'algorithme du diamant-carré
 - **Implémentation**
- III - Cohérence
 - Étude statistique
 - Vérification algorithmique

II - Reliefs montagneux > Diamant-carré

(Algorithme récursif) Échantillons B



Lissage : 0.00

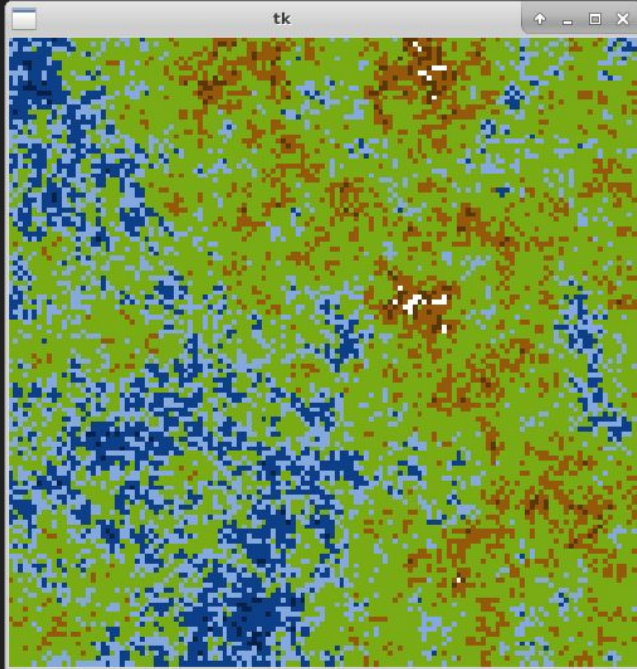


Lissage : 0.05

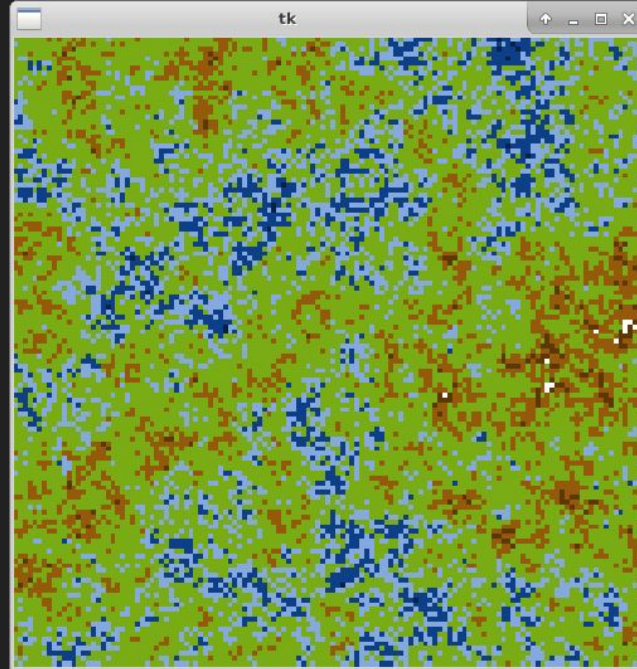
- I - Donjons
 - Méthode du *quadtree*
 - L'exemple de *Spelunky*
- II - Reliefs montagneux
 - L'algorithme du diamant-carré
 - **Implémentation**
- III - Cohérence
 - Étude statistique
 - Vérification algorithmique

II - Reliefs montagneux > Diamant-carré

(Algorithme récursif) Échantillons B



Lissage : 0.3

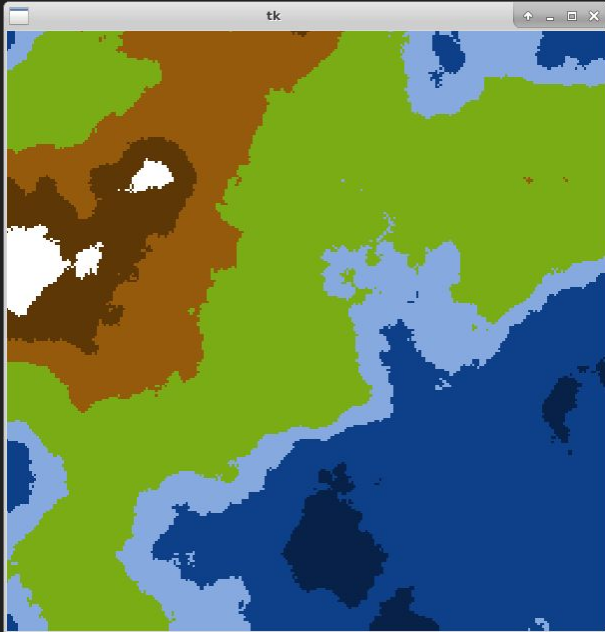


Lissage : 1.00

- I - Donjons
 - Méthode du *quadtree*
 - L'exemple de *Spelunky*
- II - Reliefs montagneux
 - L'algorithme du diamant-carré
 - **Implémentation**
- III - Cohérence
 - Étude statistique
 - Vérification algorithmique

II - Reliefs montagneux > Diamant-carré

(Algorithme itératif) Échantillons C



- I - Donjons
 - Méthode du *quadtree*
 - L'exemple de *Spelunky*
- II - Reliefs montagneux
 - L'algorithme du diamant-carré
 - **Implémentation**
- III - Cohérence
 - Étude statistique
 - Vérification algorithmique

III - Cohérence > Étude statistique

— Trois versions d'échantillons soumis aux votes (A, B et C) ;

— **78** avis récoltés, avec commentaires pour la plupart ;

— Question posée pour chaque échantillon :

[La] carte vous paraît-elle
acceptable dans un cadre fictif ?
(jeu vidéo, dessin animé, etc)

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

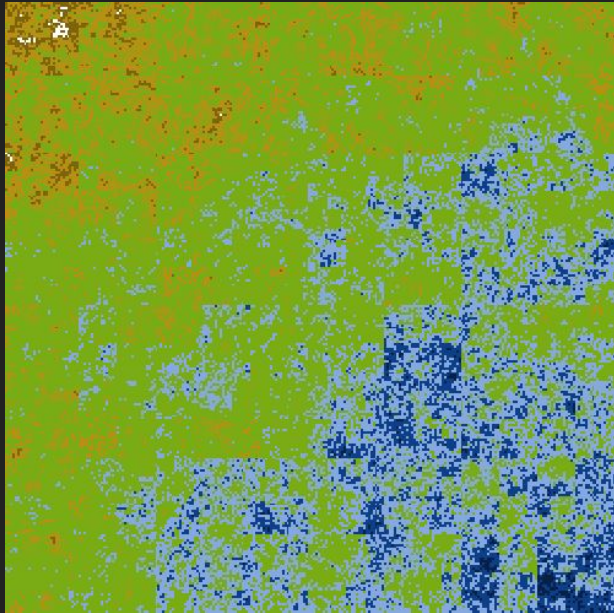
III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Échantillon 1 - A



Avis favorables :

26 / 78

soit **33.3 %**

Lissage : (sans)

Dimension : 2^8+1

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

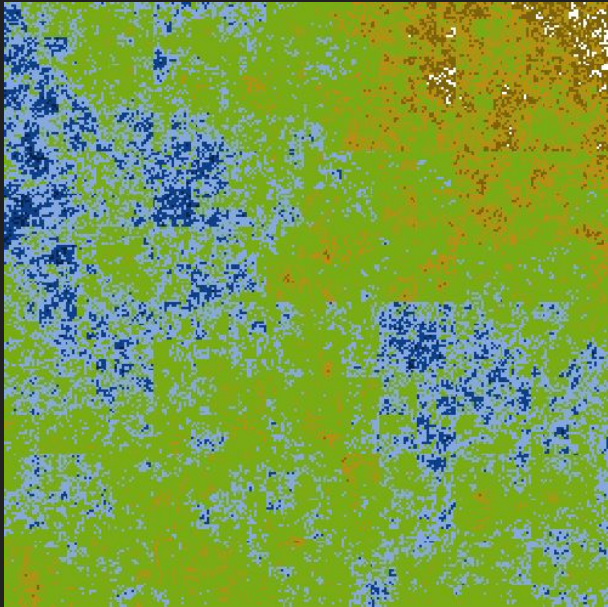
III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Échantillon 2 - A



Avis favorables :

19 / 78

soit **24.4 %**

Lissage : (sans)

Dimension : 2^8+1

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

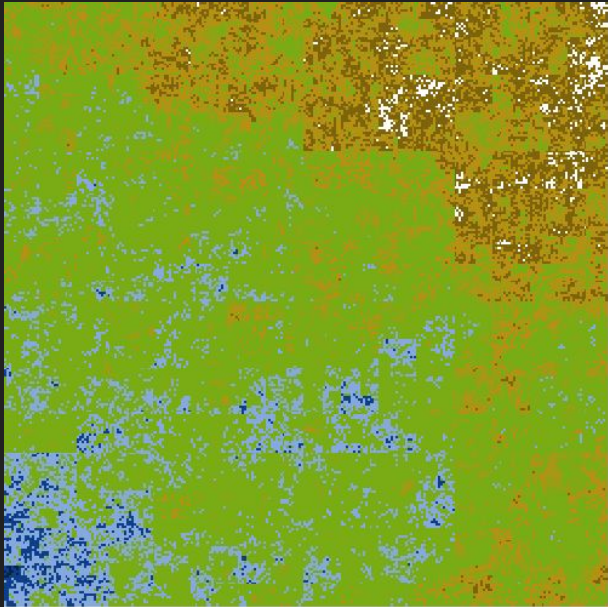
III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Échantillon 3 - A



Avis favorables :

31 / 78

soit **39.9 %**

Lissage : (sans)

Dimension : 2^8+1

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

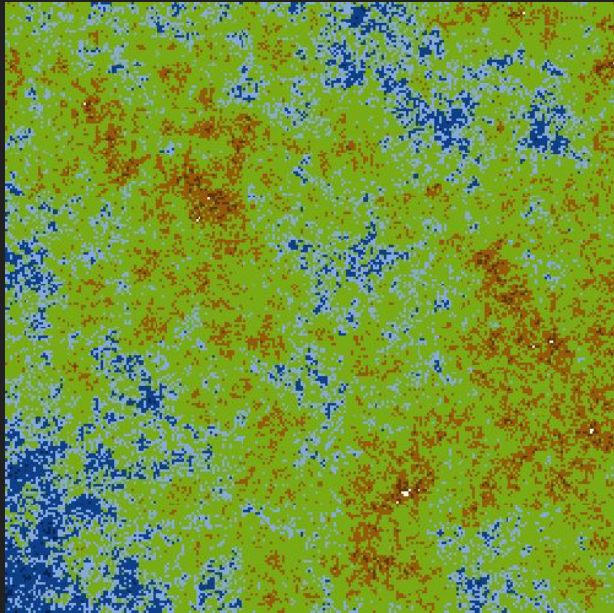
III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Échantillon 4 - B



Avis favorables :

20 / 78

soit **25.6 %**

Lissage : 0.3

Dimension : 2^8+1

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

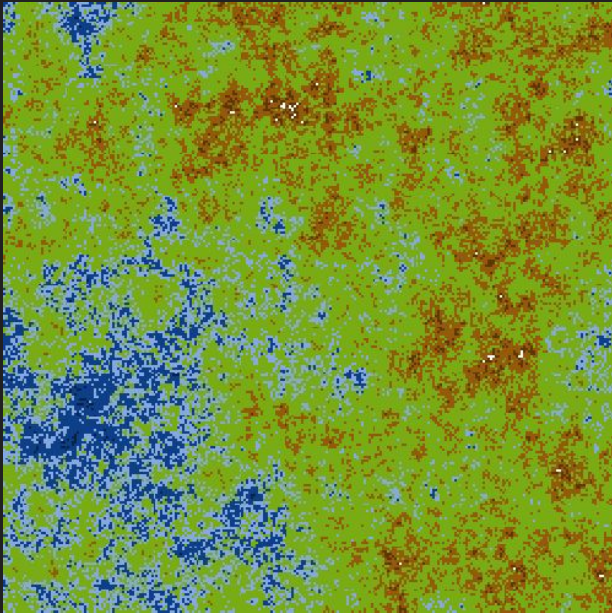
III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Échantillon 5 - B



Avis favorables :

28 / 78

soit **35.9 %**

Lissage : 0.3

Dimension : 2^8+1

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

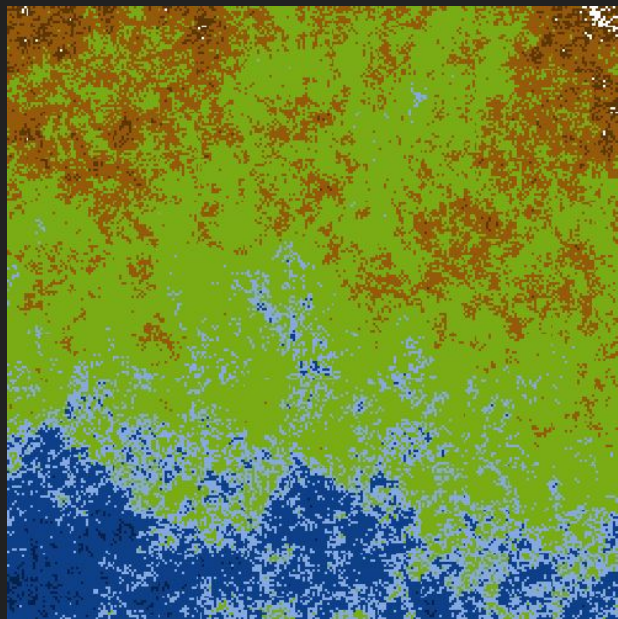
III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Échantillon 6 - B



Avis favorables :

54 / 78

soit **69.2 %**

Lissage : 0.05

Dimension : 2^8+1

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

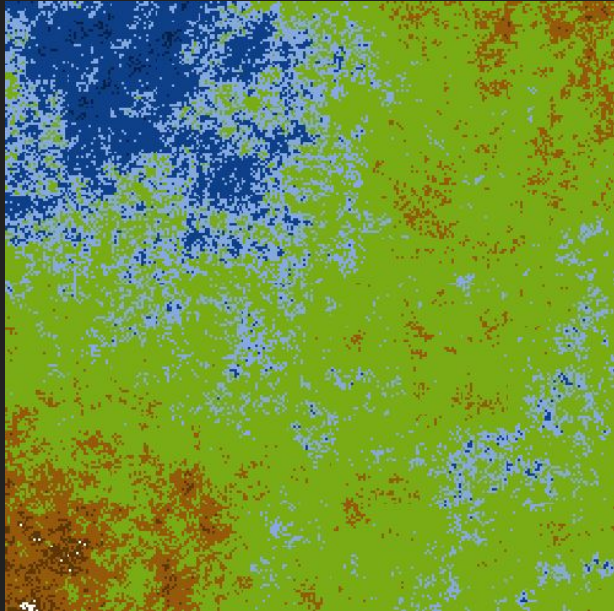
III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Échantillon 7 - B



Avis favorables :

38 / 78

soit **48.7 %**

Lissage : 0.05

Dimension : 2^8+1

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

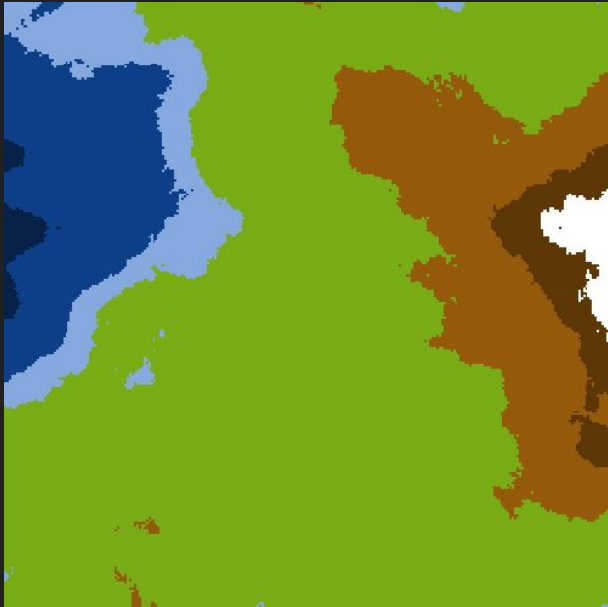
III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Échantillon 8 - C



Avis favorables :

70 / 78

soit **89.7 %**

Lissage : (sans)

Dimension : 2^8+1

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

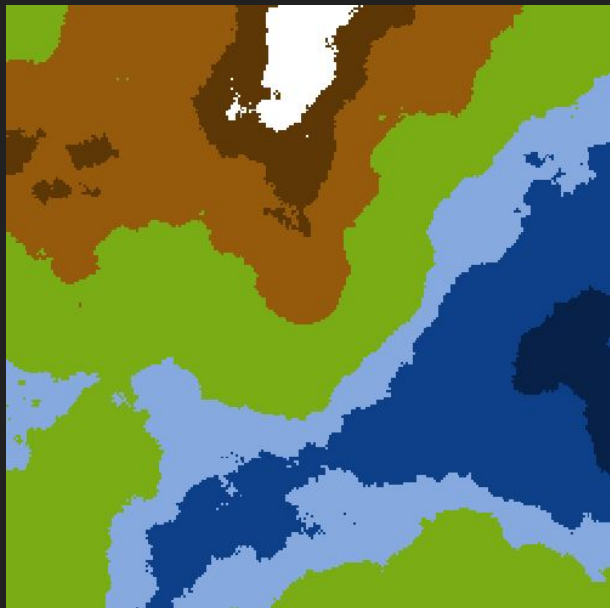
III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Échantillon 9 - C



Avis favorables :

73 / 78

soit **93.6 %**

Lissage : (sans)

Dimension : 2^8+1

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Échantillon 10 - C



Avis favorables :

63 / 78

soit **80.8 %**

Lissage : (sans)

Dimension : 2^8+1

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

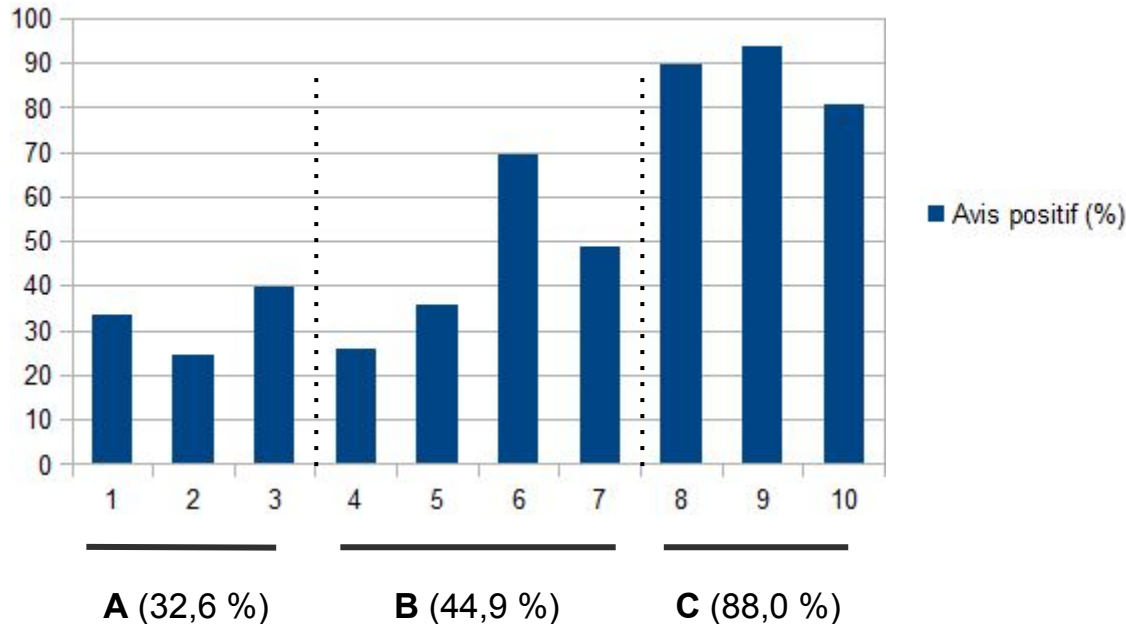
III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Avis positif sur la cohérence de chaque échantillon en pourcentage



I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— **Étude statistique**

— Vérification algorithmique

III - Cohérence > Étude statistique

Commentaires

- Absence d'échelle ;
- Océans *dans* le continent (A et B) ;
- Dispersion des points d'eaux dans le continent et *habitants* : problématique (A et B) ;
- Cartes toutes acceptables selon le contexte choisi (vision détaillée et zoom : A et B vision stratégique : C). [7 avis : tous « oui »]

I - Donjons

- Méthode du *quadtree*
- L'exemple de *Spelunky*

II - Reliefs montagneux

- L'algorithme du diamant-carré
- Implémentation

III - Cohérence

- **Étude statistique**
- Vérification algorithmique

III - Cohérence > Étude statistique & Validation algorithmique

Echt	Moy	Var	Moy (%)	Var (%)	Max	Min	Approbation
01	0.656	0.23	7.461 %	2.66 %	8.787	-1.388	33.3 %
02	0.656	0.23	7.182 %	2.55 %	9.129	-0.632	24.4 %
03	0.656	0.23	6.009 %	2.13 %	10.92	0.054	39.9 %
04	0.188	0.018	9.709 %	0.92 %	1.941	-0.452	25.6 %
05	0.188	0.017	11.98 %	1.13 %	1.566	-0.902	35.9 %
06	0.031	0.00	4.967 %	0.07 %	0.63	0.049	69.2 %
07	0.659	0.24	5.658 %	2.06 %	11.654	-2.874	48.7 %
08	1.137	0.73	0.766 %	0.50 %	148.513	-97.113	89.7 %
09	1.218	0.82	0.931 %	0.63 %	130.788	-115.164	93.6 %
10	1.167	0.78	1.459 %	0.97 %	79.997	-128.19	80.8 %

- I - Donjons
 - Méthode du *quadtree*
 - L'exemple de *Spelunky*
- II - Reliefs montagneux
 - L'algorithme du diamant-carré
 - Implémentation
- III - Cohérence
 - **Étude statistique**
 - **Vérification algorithmique**

A B C

III - Cohérence > Vérification algorithmique

Exemple de vérification algorithmique possible :

— Chaque écart de hauteur est inférieur à $x\%$;

— Jusqu'à $y\%$ des écarts peuvent déroger à la condition ci-dessus

Comment choisir la moyenne x et le taux d'acceptation y ?

Stats : échantillons 6, 8, 9 et 10 ? (moyenne $< 5\%$)

Nécessité de plus d'échantillons et de participations.

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

— Implémentation

III - Cohérence

— Étude statistique

— **Vérification algorithmique**

III - Cohérence > Vérification algorithmique

Exemple : pour « $x = 7\%$ » et « $y = 20\%$ »

Echt	Vérifié	Dépassements	Dépassement (%)
1	False	52518	39.912
2	False	50047	38.034
3	False	38173	29.010
4	False	69690	52.962
5	False	80354	61.066
6	True	26230	19.934
7	False	33878	25.746
8	True	0	0.0000
9	True	0	0.0000
10	True	16	0.0121

I - Donjons

— Méthode du *quadtree*

— L'exemple de *Spelunky*

II - Reliefs montagneux

— L'algorithme du diamant-carré

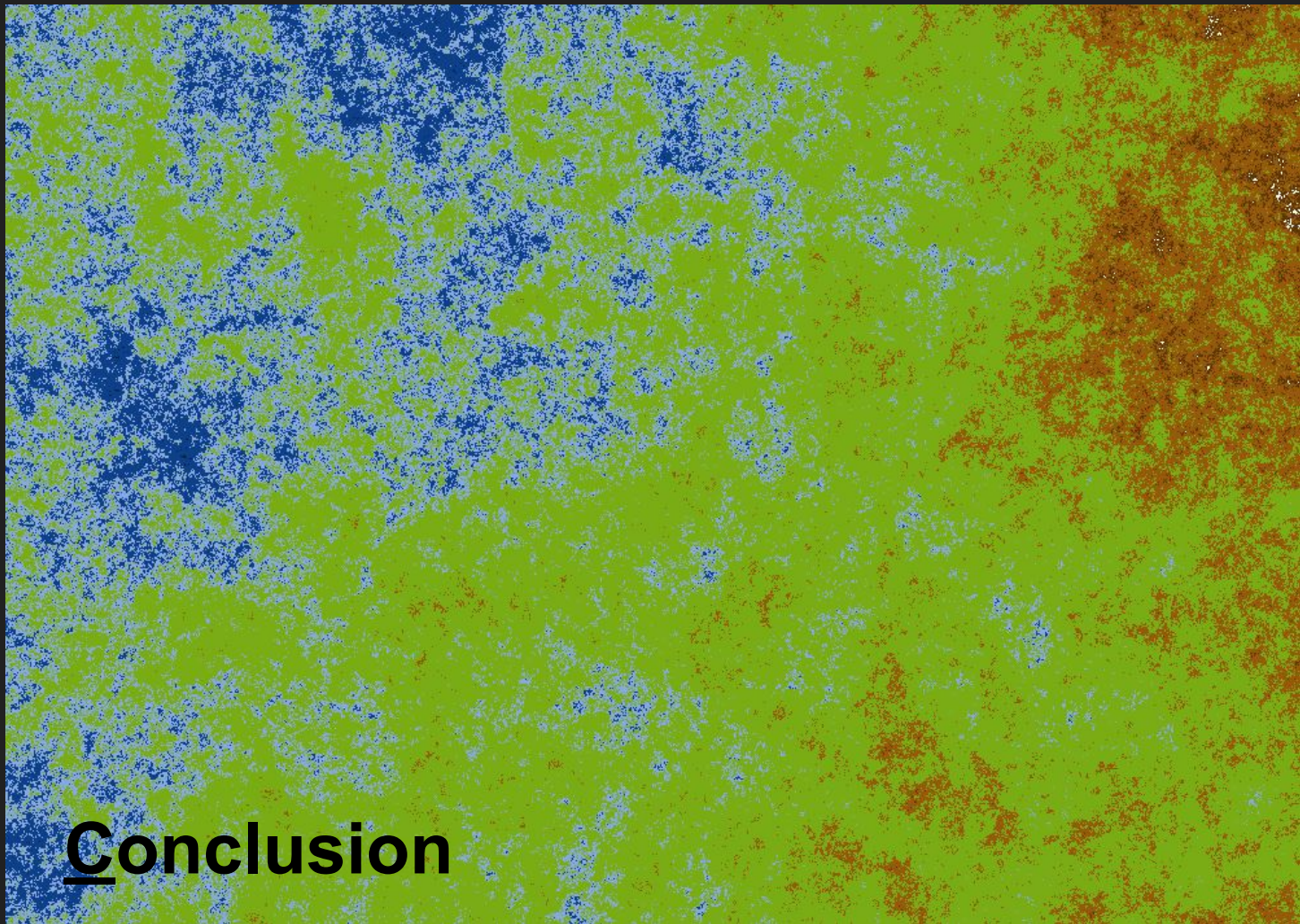
— Implémentation

III - Cohérence

— Étude statistique

— **Vérification algorithmique**

True si moins de 20 % des écarts sont inférieurs à 7 %.



Conclusion