

TIPE : LA GÉNÉRATION PROCÉDURALE

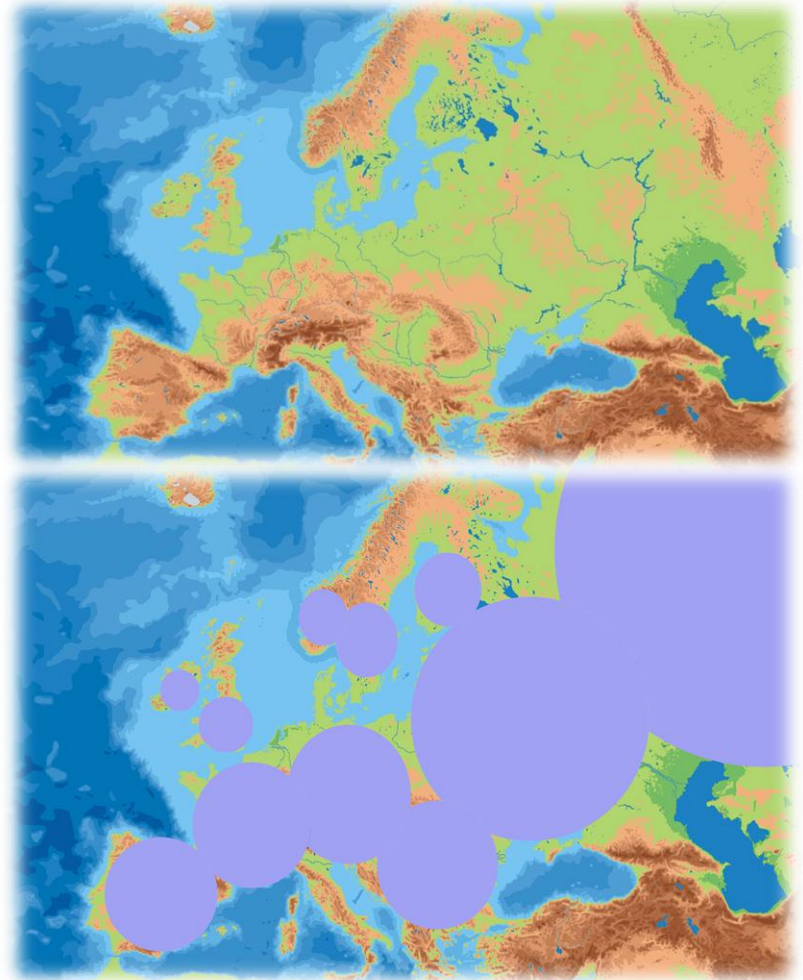
HENRARD Stanislas
SEGUIN Béranger
MPSI2

Principe de notre travail

- **Génération procédurale** : générer aléatoirement des objets conformes à certains critères.
- **Méthode** :
 - Étudier l'objet à générer, en tirer les principales caractéristiques
 - Établir un modèle et en déduire un algorithme
 - Mettre l'algorithme à l'épreuve
 - Améliorer le temps d'exécution
- **Exigences** : ressemblance à l'objet étudié, vraisemblance, esthétique, complexité acceptable...
- **Objet de notre étude** : génération de cartes avec reliefs (océan, terre, montagnes)

Étude d'une carte

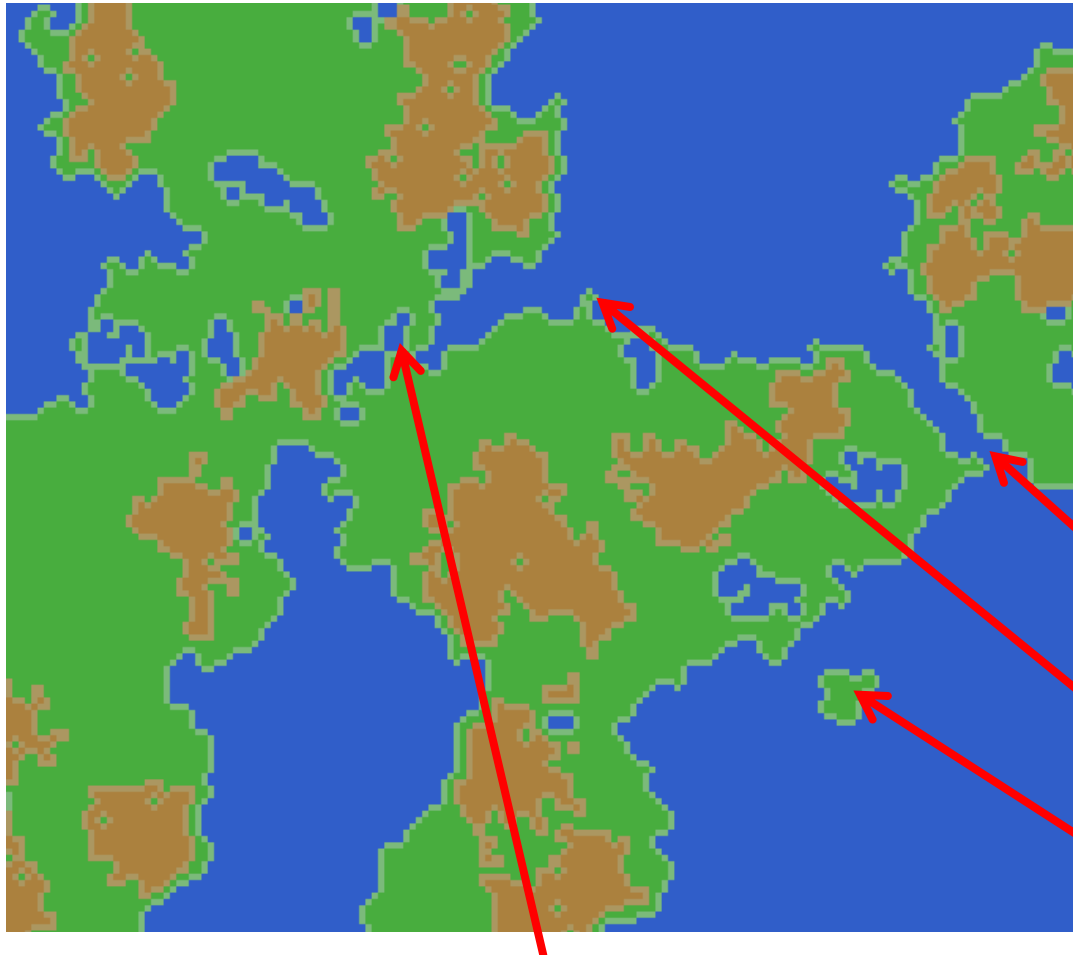
- Sur une carte de l'Europe, on fait quelques observations :
 - Les continents ont des formes plutôt lisses. Les contours sont cependant irréguliers.
 - On reconnaît dans les continents des ensembles quasi-circulaires se superposant
 - Certaines structures sont identifiables : isthmes, détroits, îles, lacs...
 - Les montagnes représentent environ 30% des continents.



Plan général de l'algorithme

- Des observations d'une carte, un algorithme se dégage:
 - On génère un nombre fixé m de points suffisamment éloignés qu'on place dans une liste *bordures*.
 - De chaque bloc de *bordures*, on rajoute du continent dans un bloc d'eau adjacent choisi aléatoirement et on met à jour la liste *bordures* pour qu'elle contienne à tout instant l'ensemble des blocs de terre en contact avec un bloc d'eau. On répète cette opération jusqu'à atteindre une proportion de continents sur la carte suffisante.
 - On lisse ensuite les contours en supprimant les blocs « erratiques » : blocs de terre ou d'eau isolés, entourés de moins de 2 blocs identiques.
 - On procède de manière analogue pour rajouter des montagnes et des lacs au sein des continents.

Résultats obtenus



- Cette carte (200*150) a été obtenue en 13s.
- On y identifie bien les structures qu'on avait repérées sur une vraie carte

Lacs

Détroit

Péninsule

Île

Isthme

Extraits du code

- **Algorithme d' « expansion continentale » :**

```
while (nbTerre < n*l*T/100):  
    for (x,y) in bordures:  
        (xN,yN)=randomAdjacent(x,y)  
        if isBloc(xN,yN,eau):  
            placer(xN,yN,terre)  
            nbTerre+=1  
            updateBordure2(xN,yN)  
            for (xD,yD) in obtenirAdjacents(xN,yN):  
                updateBordure1(xD,yD)
```

⇒ À chaque passage dans la boucle: la complexité est majorée par la taille de *bordures*

- **Algorithme de lissage:**

```
bordersToSmooth=True  
while bordersToSmooth:  
    bordersToSmooth=False  
    for bloc in bordures:  
        (x,y)=bloc  
        if compteAdjacent(x,y,terre)<=1:  
            bordures.remove((x,y))  
            placer_ocean(x,y)  
            bordersToSmooth=True  
        for (xN,yN) in obtenirAdjacents(x,y):  
            if isBloc(xN,yN,terre):  
                updateBordure2(xN,yN)
```

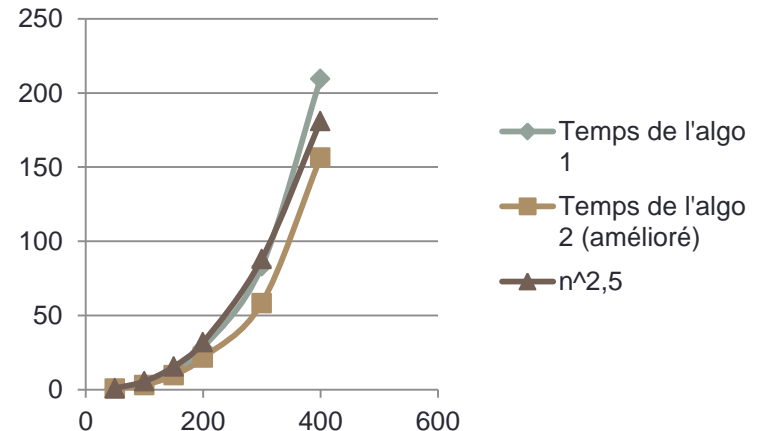
⇒ Jusqu'à ce qu'il n'y ait plus de blocs de terre « erratiques », on les enlève.

⇒ À chaque passage dans la boucle, la complexité est majorée par la taille de *bordures*.

Temps mesurés et amélioration

- Notre algorithme initial (décrit précédemment) générait des cartes de manière relativement lente.
- En optimisant la recherche d'un bloc d'eau aléatoire lors de l'expansion des continents, nous avons réussi à réaliser un gain d'environ 25%.
- Le tracé des courbes nous permet de conjecturer une complexité générale en $n^2\sqrt{n}$.

Taille de la grille	50 ²	100 ²	150 ²	200 ²	300 ²	400 ²
Temps sans amélioration	1.0 s	4.678 s	11.403 s	28.285 s	83.151 s	209.61 s
Temps avec amélioration	0.886 s	3.224 s	9.753 s	21.663 s	58.402 s	156.45 s



Conclusion

- En développant cet algorithme, nous avons réussi à mieux cerner la diversité des reliefs continentaux.
- Nous avons aussi eu l'occasion de faire face à la rapide augmentation du coût temporel et à la nécessité de l'endiguer.
- Les résultats obtenus sont visuellement très satisfaisants et seraient exploitables (par exemple pour un jeu vidéo)

