

Projet V1.2.0



Cette œuvre est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](#).

Document en ligne : www.mickael-martin-nevot.com

Date de remise du livrable intermédiaire 1 : **30 juin 2016 à 08h30**
Date de remise du livrable intermédiaire 2 : **05 juillet 2016 à 13h30**
Date de rendu : **10 juillet 2016 à minuit**
Travail : **binôme**

1 Assistance

Vous pouvez contacter l'enseignant en cas de besoin en formalisant et en ciblant précisément votre demande. Pour ce faire, vous devez respecter les règles de communication et d'envoi (ci-dessous).

2 Communication et envoi

2.1 Généralités

En joignant vos coordonnées (*e-mail* et téléphone portable notamment) à un message ou à votre livraison, vous pourrez être joint en cas de problème.

2.2 Communication

Chaque communication devra être faite :

- à l'adresse électronique mmartin.nevot@gmail.com ;
- en faisant figurer [G4][Algorithmique avancée] en début de sujet.

2.3 Livrables intermédiaires

Vous devez rendre en présentiel directement à votre enseignant **deux livrables intermédiaires** aux dates de rendues correspondantes.

2.4 Livraison finale

Vous devez envoyer le livrable final de votre projet avant la date de rendu.

Votre livrable final devra être :

- nommé de la manière suivante (Nom1, Nom2 étant vos noms et Prénom1, Prénom2 vos prénoms) : **Nom1 Prénom1 – Nom2 Prénom2** ;
- compressé dans une seule archive au format ZIP ;

- enregistré, avant la date de rendu, sur **Novae Cloud Echanges Documents G4** (<https://www.novae.info/edn/>) à l'adresse suivante : Rendus Novae / Travail à rendre / Marseille / Mickaël Martin Nevot / Algorithmique avancée.

2.5 Evaluation

Votre projet sera évalué selon les critères suivants :

- algorithme présentant des caractères valides de **correction**, **déterminisme** et **terminaison** ;
- **vitesse d'exécution relative** de l'algorithme par rapport aux autres groupes d'apprenants.

3 Sujet



Figure 1 – Terrain généré algorithmiquement

Générer un terrain le plus réaliste possible (modélisation, textures, etc.), en se démarquant au maximum des autres groupes d'apprenants. Idéalement, vous y incorporerez un cours d'eau et généreriez des terrains complexes.

Vous réaliserez une implantation de votre algorithme en **Java** dans une classe nommée `TerrainGenerator` et une méthode de signature `tgNom1Nom2()` (`Nom1`, `Nom2` étant vos noms). Elle devra, en outre, intégrer une **visionneuse** de terrain permettant d'observer le résultat.

Voici quelques pistes algorithmiques à rechercher pour vous aider dans l'accomplissement de ce travail :

- carte d'élévation ;
- carte de surplomb ;
- terrain fractal ;

- algorithme diamant-carré ;
- algorithme *square scheme* ;
- algorithme de Perlin ;
- algorithme Fault-Formation.

4 Conseils

Voici quelques conseils :

- allez à l'essentiel : il ne s'agit pas de faire une application surchargée mais bien de pouvoir accéder facilement aux fonctionnalités essentielles demandées dans le sujet ;
- préalablement à l'implantation de l'algorithme, faites-en la conception en effectuant toutes les recherches préalables nécessaires ;
- faites des recherches sur le Web (et idéalement ailleurs également) en prenant soin de vérifier que les informations trouvées soient correctes ;
- en cas de problème trop complexe, veillez à le découper en autant de plus petits problèmes que nécessaire ;
- améliorez la complexité de votre algorithme, notamment en réalisant des itérations successives.