Université de Montpellier
Faculté des sciences

Date : Mardi 2 décembre 2025
Enseignant : Nicolas Lutz

TER : Synthèse universelle de textures temps réel

Contexte

Ce TER est plutôt orienté *recherche appliquée*.

En informatique graphique, l'augmentation de la taille des mondes virtuels rend difficile la création et le stockage de la géométrie et de l'apparence des objets peuplant ces mondes. Pour adresser ce problème, la génération procédurale forme une famille d'algorithmes qui permettent de créer automatiquement les contenus d'un monde virtuel. La synthèse de textures temps réel fait partie de cette famille et permet de générer l'apparence de surfaces de taille non bornée voire infinies sans augmenter l'occupation mémoire.

La synthèse de textures temps réel par l'exemple, en particulier, consiste à étendre un petit échantillon de texture sur la surface à l'aide d'un algorithme contenant de l'aléa, sans les répétitions alignées que causeraient le fait de répéter cet échantillon à l'infini sur la surface (une technique couramment utilisée dans le jeu vidéo qu'on appelle "pavage périodique").

Ces algorithmes sont implémentés directement dans un *fragment shader* pour que la texture soit générée à la volée, uniquement lorsque la surface texturée est visible, et uniquement pour chaque pixel ayant une visibilité directe sur la texture. Ces algorithmes doivent donc être extrêmement rapides à calculer [HN18].

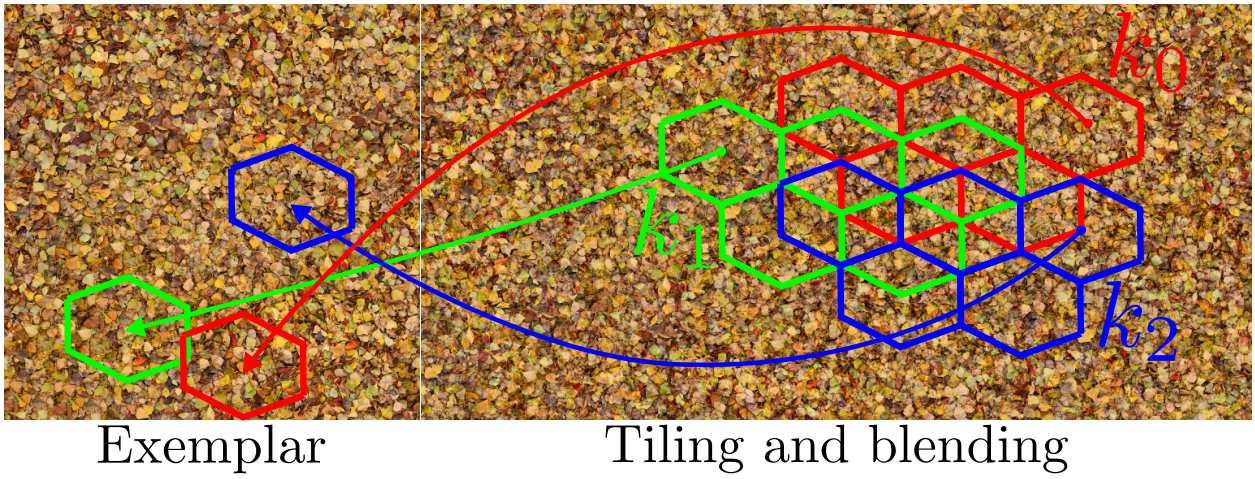

FIGURE 1 – Synthèse de textures de Heitz et Neyret [HN18]. Cette synthèse, calculable en temps réel, consiste à exprimer la valeur en chaque position par le mélange de trois pavages hexagonaux, dont le contenu est pris aléatoirement dans l'exemple.

Problème

Ces derniers temps, la synthèse de textures a eu plusieurs améliorations pour étendre le type de textures pouvant être synthétisées correctement [LSD21, LSD23, LG25] ; cependant, des cas particuliers de textures subsistent toujours ; de plus, choisir le meilleur algorithme pour telle ou telle texture est difficile parce que la qualité des algorithmes de synthèse de textures temps réel est difficile à évaluer et que la vitesse de synthèse varie selon l'algorithme utilisé.

Travail demandé

Plutôt que d'attaquer frontalement le problème en résolvant tous les cas particuliers, vous allez proposer une méthode unique, universelle pour toutes les textures. Cette piste consistera à représenter la texture avec un pavage périodique de près, et l'algorithme du pavage et mélange [HN18] de la Figure 1 de loin. Vous ferez en sorte que la transition soit le moins visible possible et vous argumenterez vos choix d'implémentation.

Vous devrez proposer un prototype de votre travail sur Godot. Vous disposerez déjà de l'implémentation de la synthèse de textures en tant que GDShader, mais il est indispensable que vous compreniez son fonctionnement pour pouvoir travailler avec.

Le TER sera considéré comme un succès si l'une des conditions est respectée :

— La méthode fonctionne telle quelle, et le groupe a prouvé d'une certaine façon que leur algorithme est le meilleur pour de très nombreuses textures.

— La méthode ne fonctionne pas telle quelle, mais le groupe a implémenté des améliorations permettant d'améliorer sa viabilité.

Références

[HN18] Eric Heitz and Fabrice Neyret. High-performance by-example noise using a histogram-preserving blending operator. *Eurographics Symposium on High-Performance Graphics 2018*, 2018.

[LG25] Nicolas Lutz and Guillaume Gilet. Real-time by-example texture synthesis and filtering using local statistics exchange. *Not yet published*, 2025. Submitted to Eurographics 2026.

[LSD21] Nicolas Lutz, Basile Sauvage, and Jean-Michel Dischler. Cyclostationary Gaussian noise : theory and synthesis. In *Eurographics 2021*, Vienna, Austria, May 2021.

[LSD23] Nicolas Lutz, Basile Sauvage, and Jean-Michel Dischler. Preserving the Autocovariance of Texture Tilings Using Importance Sampling. *Computer Graphics Forum*, 2023.