**« Backlog » de sprint #3**

Produit : Curved Fractals

Conçu par : JeeGo Squad

## Nom des membres :

Jérôme Pagé – Responsable des livrables

Simon Lepage – Scrum master

Jonathan Simard – Secrétaire

Ludovic D’Anjou-Madore – Directeur de produit

## Échéancier (période):

## Légende :

* Vert, indique que ces items sont réalisés.
* Jaune, indique que ces items sont en cours de réalisation.
* Rouge, problème ou questionnement important qui demande une rencontre d’équipe.
* Aucune couleur, indique que ces items ne sont pas encore faits ou commencés, **on peut toujours les enrichir mais il faut le consentement de toute l’équipe**.

## « Backlog » de sprint

|  |  |
| --- | --- |
| **3** | |
| Acteur ou rôle : | Utilisateur |
| Scénario ou story : | En tant qu’utilisateur, je veux pouvoir spécifier les composantes du champ tensoriel métrique et voir les effets de la courbure de celui-ci sur une fractale |
| Détail ou description : | 1. Recevoir les chaines de caractères de l’utilisateur représentant les composantes du champ tensoriel métrique. 2. Valider la structure et la syntaxe des composantes et les interpréter pour construire le champ tensoriel métrique en Java. 3. Construire une texture 3D (2D Array Textures ou un tableau 2x2 de matrices 2x2 sur OGL 4.3 et +) en évaluant le tenseur à chaque pixel (les composantes x et y de la texture représentent la position de chaque pixel et le niveau (pour le 2DTextures Array) représente la nième composante du tenseur a un point (pour le tableau 2x2, en fixant la position, on obtient une matrice, la représentation locale du tenseur métrique). 4. Envoyer la structure de données choisie sur le shader et calculer la déformation pour chaque pixel. |
| Tests d’acceptation : | Afficher l’ensemble de Mandelbrot et confirmation visuelle de la déformation. |
| Complexité : | 7 |
| Effort : | 3j/personne **ou** 4 |
| Commentaires : | -Lors du changement de tenseur spécifié par l’utilisateur, on recalcule une texture et on met à jour la référence dans le shader  -lors d’un zoom, on doit « zoomer » aussi les coordonnés d’accès à la représentation en mémoire du tenseur. Donc, si on utilise une texture 3d, l’interpolation ce fait automatique pour chaque niveau. Si on utilise un tableau de matrice, dans chaque direction, on doit interpoler entre les composantes. (par exemple : voir l’interpolation linéaire dans la description des concepts) |