

Projet de Programmation

Génération de planètes

Alexey Zhukov
Tony Wolff
Baptiste Bedouret
Alexis Marec
Antoine Fredefon
Thomas Mercier

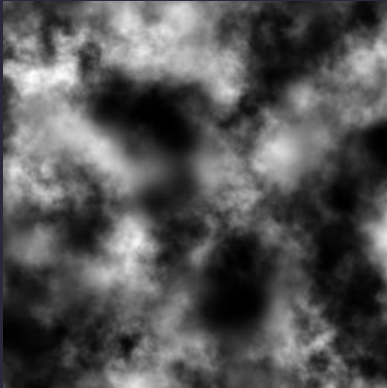
Professeur accompagnant: M. Mansencal

université
de **BORDEAUX**



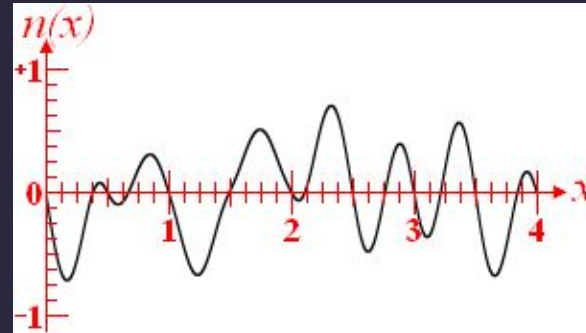
Heightmap

Image 2D



<http://upload.wikimedia.org/wikipedia/commons/5/57/Heightmap.png>

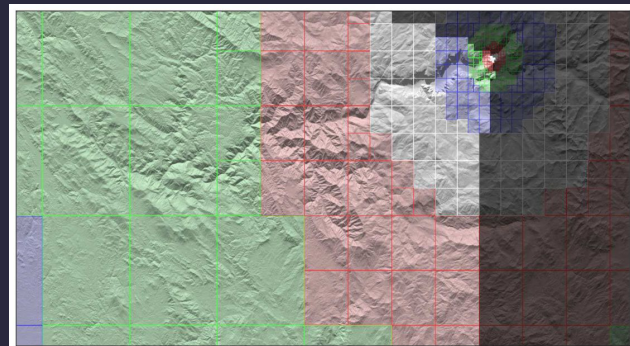
Bruit de Perlin



<http://libnoise.sourceforge.net/examples/complexplanet/index.html>

Continuous Distance-Dependent Level of Detail

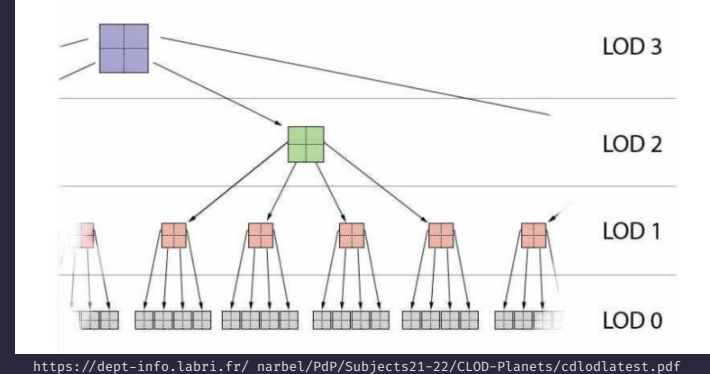
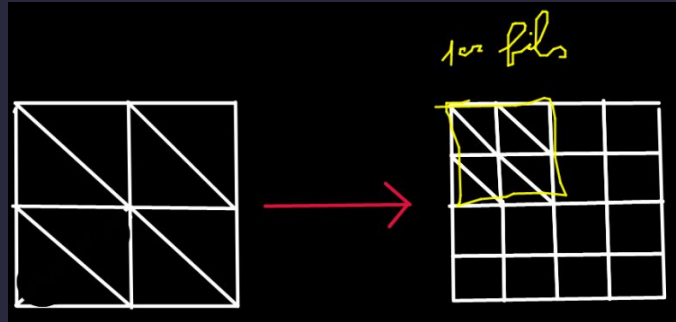
- Différents niveaux de précision
- Effectuée à chaque déplacement



<https://dept-info.labri.fr/narbel/PdP/Subjects21-22/CLOD-Planets/cdlodlatest.pdf>

Quadtree

- Subdivisions du maillage de la surface
- Noeuds
- Quadrants
- triangles



<https://dept-info.labri.fr/~narbel/PdP/Subjects21-22/CL00-Planets/cd lodlatest.pdf>

The background of the slide is a dark blue field filled with a complex, white wireframe pattern. This pattern consists of numerous interconnected lines that form a series of overlapping, irregular polygons and shapes, creating a sense of depth and movement, similar to a wireframe mesh or a stylized, abstract landscape. The lines are thin and white, contrasting sharply with the dark blue background.

Visualisation OpenGL

Description des besoins - CDLOD

- lecture de heightmap : Format, Précision, Stockage
- Stockage de la heightmap sous forme de Quatree
- Sélection des noeuds visibles en fonction de la distance avec l'observateur
- Transfert des données à la partie OpenGL

Description des besoins - Visualisation

- Fenêtres de visualisation : GLFW
- Création de la fonctionnalité nécessaire et intégration de l'existant : Eigen, Glbinding
- Initialisation de la caméra
- Les données de terrain : Frustum Culling

<https://dept-info.labri.fr/narbel/PdP/Subjects21-22/CLOD-Planets/cdlodlatest.pdf>

Architecture

-

Architecture

-

Point technique (CDLOD)

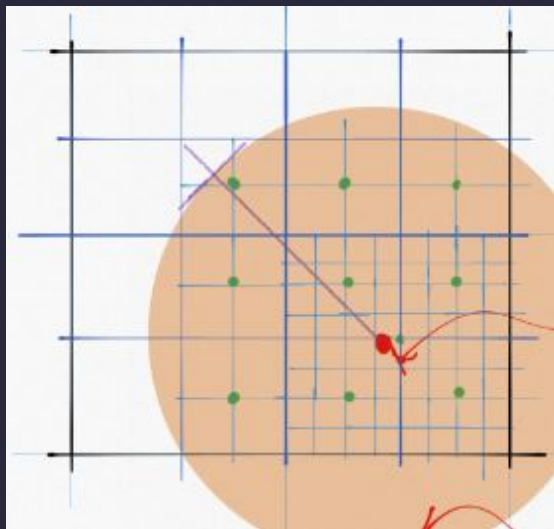
- Création du quadtree :
 - Parcours des sommets présents dans la heightmap
 - Définir le nombre de niveaux de détail (i.e profondeur de l'arbre)
 - Tant que le nombre de sommets dans le noeud courant est supérieur à ce qui est demandé, on découpe le noeud en quatre fils.

Point technique (CDLOD)

```
/*  
 * Fils en haut à gauche  
 */  
childNode.minX = currentNode.minX;  
childNode.minZ = currentNode.minZ;  
childNode.maxX = currentNode.minX + childNode.width;  
childNode.maxZ = currentNode.minZ + childNode.height;  
  
childNode.level = currentNode.level + 1;  
  
nodeIndex++;  
currentNode.childrenIndex[0] = nodeIndex;  
createNode(childNode);
```

Point technique (CDLOD)

Définition des distances de transition entre les niveaux de détail (i.e. le rayon du cercle entourant l'observateur) :

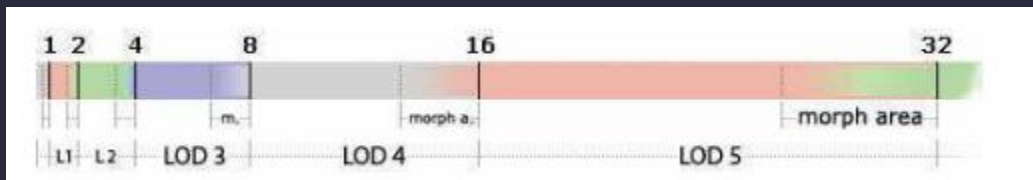


- A chaque déplacement de l'observateur, on calcule la distance entre la caméra et chaque noeud.
- On affiche les sommets des noeuds visibles.

Point technique (CDLOD)

Transition entre les niveaux de détail :

Théoriquement, l'algorithme CDLOD fait en sorte que la transition entre deux niveaux de détail se passe de manière fluide.



F. Strugar, 11 Juillet 2010. *CDLOD*

Point technique (CDLOD)

Passage à OpenGL :

- Création du QT avec la HeightMap
- Création d'un vecteur de vertex contenant l'information pour dessiner les quads et triangles + calcul des normales et coordonnées de textures
- passage à OpenGL avec glBindBuffer, utilisation du GPU et accélération du rendu

```
vx = qtTest->nodearray.at(i).vertices[x][z].x;
vz = qtTest->nodearray.at(i).vertices[x][z].z;

//std::cout << vx << " -- " << vz << endl;
mVertices[indexVertices] = Vertex(Vector3f(vx* heightmap_x, (heightmapMat(vx, vz) * 255) * heightmap_y, vz* heightmap_z);
mVertices[indexVertices].texcoord = Vector2f(vx* heightmap_tex_x, vz* heightmap_tex_z);
mVertices[indexVertices].visible = true;
```

Thomas Mercier, 3 days ago • Adding terrain class and modifying came

Description des tests - Quadtree

Description : Vérifier si le nombre de noeuds créés correspond à la formule définie dans l'algorithme.

Données en entrée : Une heightmap

Déroulement du test : Modifier le nombre de niveaux de détail attendu et comparer le nombre de noeuds créés avec le nombre théorique.

Résultats attendus : Entre 0 et 9 niveaux de détails : résultat identique à la formule.

Description des tests - Quadtree

test 3

Map Height : 512

Map Width : 512

vertexNum : 262144

facesNum : 524288

----- QuadTree build -----

341 nodes created

341 nodes expected

```
void QuadTree::createNode(QTNode currentNode)
{
    // A choisir
    int maxLevel = 5;
    float threshold = width * pow(0.5, maxLevel);

    // Une fois le cap de triangle atteint
    if ((currentNode.height) == threshold)
    {
        currentNode.leaf = true;
        return;
    }
}
```

Résultats obtenus



Conclusion et points d'améliorations

- Algorithme pas très intuitif, problème stimulant mais CLOD n'est pas très documenté
 - L'analyse de l'existant est très utile
 - Démarrage lent (en racine carrée)
-
- Continuous DLOD
 - Application de l'algorithme sur une sphère (CLOD sur les 3 axes)
 - Frustum Culling
 - Tests unitaires