

**Forum: Moteur de jeu GameBlender et alternatives****Topic: système Sid Meyer like (Civilization)****Subject: Re: système Sid Meyer like (Civilization)**

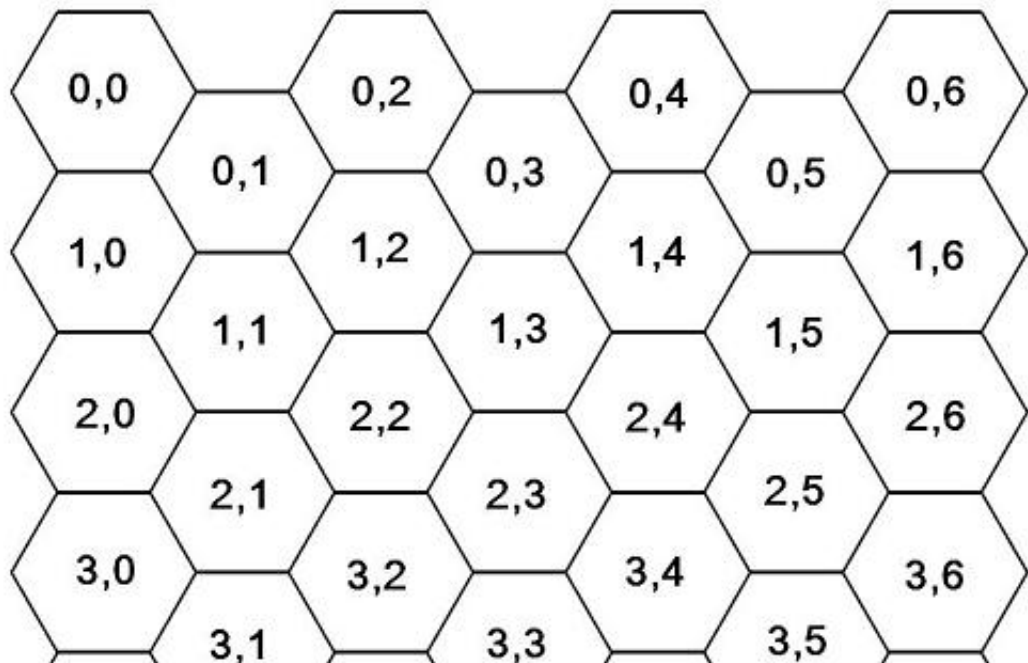
Posté par: Bibi09

Contribution le : 9/4/2020 12:05:48

Je crois qu'on s'est compris.

Il y a un décalage entre ce que tu vois (et donc la façon dont tu structures une matrice d'hexagones) et la façon de représenter les données en interne.

Une matrice d'hexagone, c'est un peu comme une matrice de carrés. Il faut juste glisser les cases impaires vers le haut ou vers le bas d'une demi-case :



Mais tu vois sur cette image que si tu mets les cases sur une même ligne, ça te fait toujours un carré. Tu peux en déduire les tuiles adjacentes à tester.

\*\*Tu génères le diagramme de Voronoï. Ca te donnera des cellules. Tu remplis ces cellules avec les hexagones.

Tu utilises les mêmes hexagones pour A\*.

Quant à l'opérateur "<>" (différent), il a été remplacé par "!=" dans Python3. Tu peux donc faire cette modification sans que ça ne change le fonctionnement de l'algorithme.

\*\*

EDIT: pour la génération de la map, je vais essayer d'être plus clair.

Tu génères un diagramme de Voronoi de la même taille que tu veux avoir ta map. Si ta map fait 64x48 tuiles, tu génères un diagramme de 64x48 pixels. Chaque pixel représente le type de terrain (terre, montagne, mer). Tu as alors ta map qui est constitué de tuiles et chaque tuile représente déjà le terrain.

Tu utilises ta map de données pour lancer A\*.  
Pas besoin de faire plusieurs tableaux.