

Mini Rapport TP1 Dijkstra & A*

Version Unity : 2020.1.6.f1

Le script **PathGrid** représente la grille, les noeuds, utilisés pour le calcul des plus courts chemins à la fois par Dijkstra et A*. Dans l'inspecteur Unity on peut indiquer si nous utilisons les diagonales (i.e. si nous sommes en 4 ou 8 voisinage).

Le script **Node** représente donc chaque noeud de la grille. Il contient les différents coûts :

- *gCost* : la distance de Manhattan (avec 4 ou 8 voisins) depuis la source.
- *hCost* : la distance de Manhattan du noeud courant vers la destination
- *fCost* : simple cumul $gCost + hCost$.

AStar utilise *fCost* pour calculer le plus court chemin. Dijkstra utilise *gCost* uniquement.

Le script **Player** permet de contrôler le mage "francis" avec les touches Z/Q/S/D ou les flèches. Il se déplace de case en case. Dans l'inspecteur associé à ce *gameobject* on peut définir un délai entre chaque déplacement. C'est uniquement pour éviter que le personnage se déplace trop vite.

Les scripts **AStarEnemy** & **DijkstraEnemy** sont rattachés à notre ennemi, le prince crapaud. Ces scripts calculent le chemins (selon l'algorithme) en direction d'un *gameobject* défini dans l'inspecteur. **Attention à bien désactiver un des deux scripts lors des tests.** Comme pour le joueur, on peut définir un délai entre chaque déplacement.

Parfois l'ennemi semble parcourir 2 cases à chaque déplacement. C'est un bug qui n'a pas encore été trouvé mais est certainement lié au recalcul du chemin.

Il n'y a pas de conditions de victoire dans ce jeu.