

## Présentation « avancée » de l'IA de Dead Memories

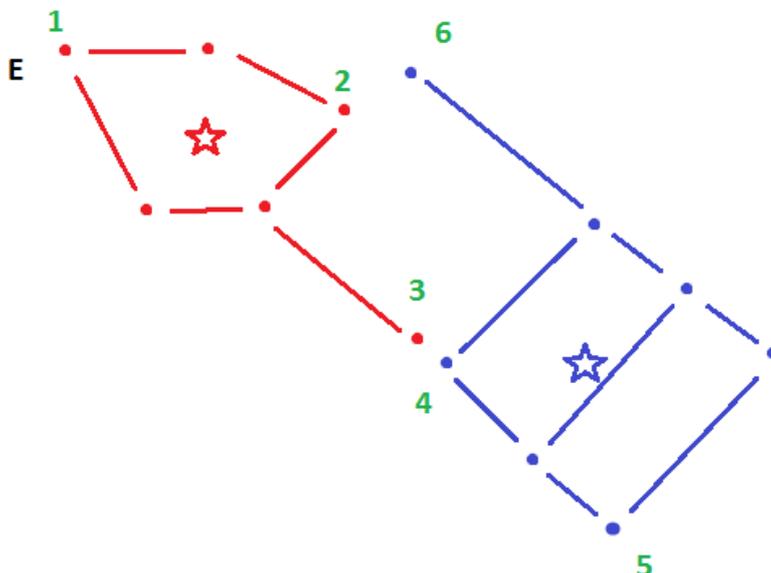
L'IA de Dead Memories définit trois « états » principaux pour les ennemis (patrouille, poursuite et attaque) et gère les transitions entre ces différents états.

Au départ, les ennemis patrouillent. Lorsque l'un d'eux voit le joueur (suivant une distance et un champ de vision défini) ou l'entend (suivant une distance définie, lorsque le joueur se déplace), il se met à sa poursuite et alerte ces voisins dans un rayon déterminé. Ces mêmes voisins ne transmettent pas l'alerte mais se mettent eux-aussi à la poursuite du joueur. Lorsqu'un ennemi est suffisamment proche du joueur, il attaque. Il choisit entre une attaque au corps à corps et une attaque magique suivant différents facteurs expliqués ci-dessous.

La patrouille est probablement l'élément qui a reçu le plus d'attention pour simuler des déplacements naturels. Elle est basée sur l'utilisation de graphes, chaque graphe définissant les points que peuvent relier les ennemis. Un sommet du graphe est choisi comme destination. Le sommet de départ est celui qui est le plus proche de la position de l'ennemi. Le chemin à suivre est calculé à l'aide de l'algorithme de Dijkstra, suivant les arêtes définies entre les sommets. Le chemin pour relier deux sommets disposant d'une arête commune est lui-même calculé à l'aide du NavMesh d'Unity.

Chaque graphe est défini sur une zone donnée du terrain. Plusieurs graphes permettent de définir la patrouille sur tout le terrain. Les ennemis peuvent naviguer de graphe en graphe car à chaque fois qu'ils atteignent leur destination, ils se rattachent à un nouveau graphe, celui qui leur est le plus proche. La position du graphe est définie indépendamment de la position de ses sommets (ie. ce n'est pas le barycentre des sommets).

*Un exemple :*



On prend E la position d'un ennemi. Pour patrouiller, il va d'abord chercher le graphe le plus proche (ici les positions des graphes sont symbolisées par les étoiles). Le graphe le plus proche est le rouge. Il cherche ensuite le sommet de ce graphe le plus proche, le 1, et en choisit un au hasard comme destination, par exemple le 2. Il détermine le chemin le plus court entre les deux (ici en passant par « en haut »). Il rejoint la destination en passant par chaque sommet. Une fois arrivé à destination, il recherche à nouveau le graphe le plus proche (ici toujours le rouge), recherche le sommet le plus proche comme point de départ, choisit une destination aléatoire (par exemple le sommet 3) et s'y dirige, par le chemin le plus court. Il recommence tant qu'il patrouille. Arrivé au sommet 3, le graphe le plus proche est le bleu, et le sommet de ce graphe le plus proche est le 4. Une destination possible est le sommet 5. De même, une fois le sommet 5 atteint, si le sommet 6 est pris comme destination, le graphe le plus proche sera le rouge et le sommet le plus proche le 2.

Lorsque l'ennemi aperçoit le joueur, l'entend ou est alerté, il se met à sa poursuite. S'il perd de vue le joueur, il continue sa traque pour retrouver sa trace pendant un certain laps de temps (fixé à environ 3s dans le jeu). Pour le rechercher, il part dans la dernière direction où il a vu partir le joueur. Si l'ennemi retrouve la trace du joueur, il reprend sa traque, dans le cas contraire, il retourne patrouiller.

Lorsqu'il est suffisamment proche du joueur, l'ennemi l'attaque. Il choisi aléatoirement entre une attaque de mêlée ou une attaque de distance, le facteur de probabilité entre les deux varie suivant la quantité de mana restant à l'ennemi.