

Exercice – Path planning

Temps imparti : 4 heures maximum

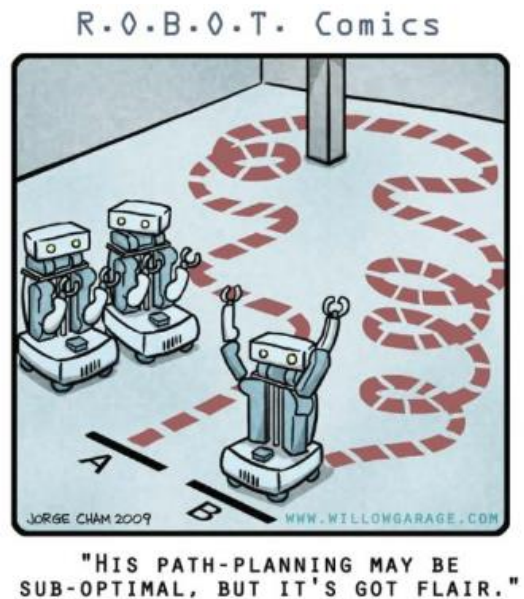
Contraintes :

- Projet : Unity3D 2018.1
- Langage : C# v6
- Plateforme : Standalone Windows/Mac/Linux
- APIs autorisées : Unity3D (UnityEngine.*) et .Net 4.6 (System.*), aucune autre bibliothèque tierce autorisée

Rendu : Fichier compressé contenant les dossiers *Assets/* et *ProjectSettings/*

Autre :

- Accès à Internet autorisé et recommandé (ne jouez pas les héros)
- Des fichiers XML sont fournis en guise d'exemple (voir exercice)



La mission : Charger un ensemble de nœuds, créer un graphe à partir de ces nœuds et établir le plus court chemin entre deux points donnés.

Voici le comportement attendu quand l'utilisateur appuie sur le bouton *Play* :

Étape 1 : Lire un fichier XML pourvu pour charger les nœuds.

Résultat attendu : Un ensemble de nœuds en mémoire.

Étape 2 : Afficher l'ensemble de ces nœuds dans Unity3D sur le plan X-Z selon les règles suivantes :

- Appliquer une couleur différente au nœud avec pour attribut Path=BEGIN et autre nœud avec pour attribut Path=END
- Les nœuds sont définis selon une certaine forme qu'il faut appliquer à l'affichage
- La position des nœuds est définie dans l'espace par des coordonnées à respecter dans Unity3D

Résultat attendu : Un visuel des nœuds dans Unity3D avec les règles appliquées.

Étape 3 : Générer et afficher le graphe en reliant les nœuds selon la règle suivante :

- Pour un nœud x , l'ensemble de ses voisins est défini par les 4 plus proches nœuds (selon une distance euclidienne) de ce nœud x

Résultat attendu : Un visuel du graphe généré dans Unity3D.

Étape 4 : Générer et afficher le plus court chemin depuis le nœud avec pour attribut Path=BEGIN jusqu'au nœud avec pour attribut Path=END. Une distance entre deux nœuds étant égale à la distance euclidienne les séparant.

Résultat attendu : Un visuel du chemin généré dans Unity3D. Mettez une couleur différente.