

Présentation EDUSCOL

Cadre de cette activité :

- Contexte : Programmer les algorithmes de Dijkstra et Astar pour la détermination du plus court chemin entre deux points sur une image en noir et blanc (réseau accessible des pixels blancs et non accessibles des pixels noirs représenté par un graphe en informatique). Plusieurs applications sont proposées, que ce soit pour la résolution d'un labyrinthe, la détermination du plus court trajet dans un réseau autoroutier, ou la détermination de la meilleure trajectoire du godet d'une pelleteuse automatisée en évitant des obstacles (TP ERM « Bras de Pelleteuse Electrique avec IA intégrée).
- Objectifs : Comprendre, programmer et comparer les algorithmes Dijkstra et Astar
- Matière : Sciences de l'Ingénieur et informatique du tronc commun
- Niveau : CPGE SPE : MP-PC-PT-PSI
- Activité : TD réalisé en classe généralement en 4h environ
- Séquence :
 - o En sciences de l'ingénieur : « Expérimentations » en lien, par exemple, avec la pelleteuse, ou tout autre sujet qui s'y rapporte
 - o En informatique du tronc commun : « Bases des graphes, plus courts chemins »
- Matériel : Ordinateur avec logiciel de programmation en langage Python (ex : Pyzo+Anaconda). Modules matplotlib, numpy et math.
- Compétence du programme de SI :
 - o C1.3.7 - Analyser un algorithme. S1
- Compétence du programme d'informatique du tronc commun (les compétences des chapitres précédents sont considérées acquises) :
 - o Recherche du plus court chemin dans un graphe pondéré avec des poids positifs
- Ressources mises à disposition :
 - o Sujet
 - o Dossier des codes corrigés (lire remarque plus bas)
 - o Dossier élèves avec les codes élèves et images du sujet
 - o Cours et résumés sur les images et les graphes
 - o Résultats : Images obtenus pour les différentes applications et les deux algorithmes
 - o Animations : Vidéos de comparaison des calculs réalisés par les deux algorithmes

Remarques :

- Cet algorithme entre dans la famille de l'intelligence artificielle, mais n'est pas spécifiquement au programme de sciences de l'ingénieur. Il peut toutefois bien évidemment créer un lien entre les matières pour être utilisé en SI, par exemple sur le bras de pelleteuse.
- Des codes élèves permettent de gagner du temps et d'organiser les 5 fichiers attendus. Tout ce qui est affichage est fourni, ainsi qu'une base de grille sur laquelle les élèves devront ajouter des obstacles.
- Veiller à bien ouvrir les différents fichiers python dans l'ordre croissant de leur numéro et toujours les exécuter selon le même ordre. Le dossier des codes corrigés contient tous les corrigés possibles, par exemple tous les fichiers commençant par 2 permettent de

créer, au choix, le graphe pour les différentes applications. Il ne faudra choisir qu'un code commençant par 2, un commençant par 4 et un code commençant par 5. J'ai proposé deux versions des codes 4 et 5 selon que je traite une file ou non.

- Des zones des corrigés sont commentées, elles permettent de générer des images à chaque itération des algorithmes dans le sous dossier Animations du dossier du code, afin ensuite de générer des vidéos. Toutefois, je vous mets déjà ces vidéos à disposition
- J'ai partagé les fichiers npy qui sont l'enregistrement au format numpy des images bmp, utiles en cas de soucis avec l'ouverture des images. Il faudra alors les charger avec `np.load(...)`.
- Il y a régulièrement des problèmes de chemins de fichiers, les élèves ayant du mal à comprendre tout ce qui est derrière. Je vous propose donc une « Check Fichiers » permettant de s'en sortir rapidement sur les erreurs « `FileNotFoundError: [Errno 2] No such file or directory: 'test.txt' »`