11759 COËRCHON Colin

Optimisation de trajectoires dans un environnement simple

La recherche de plus court chemin est une thématique plongée au cœur des nouvelles technologies. Les réflexions qui en ressortent sont applicables dans le domaine artistique, et notamment dans la conception de jeux vidéo. Cette perpétuelle recherche d'optimisation et les moyens algorithmiques mis en jeu ont alors attisé ma curiosité.
L'optimisation de trajectoires concerne les déplacements. Elle relève d'un enjeu capital quant à la prévention de situations complexes comme, par exemple, la gestion du trafic routier. On retrouve aussi cette optimisation de déplacement dans le domaine de la santé, notamment dans les endroits publics pour limiter la propagation de maladies.

Positionnement thématique (ETAPE 1)

INFORMATIQUE (Informatique pratique), INFORMATIQUE (Informatique Théorique), MATHEMATIQUES (Géométrie).

Mots-clés (ETAPE 1)

Mots-Clés (en français)	**Mots-Clés** (en anglais)
Trajectoire optimale	*Optimal pathfinding*
Algorithmes de recherche	*Search algorithms*
Triangulation	*Triangulation*
Quadrillage	*Grid*
Visibilité	*Visibility*

Mots-clés (ETAPE 2)

Mots-Clés (en français)	**Mots-Clés** (en anglais)
Trajectoire optimale	*Optimal pathfinding*
Algorithmes de recherche	*Search algorithms*
Parcours de graphe	*Graph traversal*
Triangulation	*Triangulation*
Quadrillage	*Grid*

Bibliographie commentée

La problématique de l'optimisation d'une trajectoire entre deux points est un sujet récurrent qui touche de nombreux domaines, comme l'informatique, la robotique, et plus globalement l'algorithmique (comme les GPS par exemple).

Il survient alors une question naturelle qui est le choix de la représentation à adopter en fonction des caractéristiques du milieu et de la rapidité d'exécution attendue. Il existe alors de nombreux modèles, dont on en étudiera deux des plus simples : la **représentation polygonale** et le **quadrillage de l'espace**.

En s'intéressant en premier lieu à un espace polygonal simple, on cherche à relier deux points contenus dans le polygone. L'article [1] nous présente alors une idée cruciale dans un tel espace : la triangulation. Cette idée permet alors de simplifier notre recherche d'optimisation de trajectoire.

En considérant ainsi l'espace comme une succession de triangles, on parvient rapidement à un algorithme qui fournit un tracé optimisé en un temps très faible. Mais ce tracé s'avère non optimal. Le chapitre 9 de l'ouvrage *Computational Geometry* [2] fournit alors de nombreux arguments et preuves théoriques sur le choix d'une triangulation qui maximise les angles des différents triangles et qui parvient à un découpage plus pertinent du polygone : **la triangulation de Delaunay**. Cette triangulation est unique, et l'article [3] donne alors différentes méthodes pour obtenir cette triangulation dans un espace 2D (et même 3D).

Il est impensable de se contenter d'une solution rapide mais non optimale. Il vient alors différents algorithmes qui répondent à cette problématique dans ce type de modélisation. Les plus connus sont alors décrits dans les documents [4] et [5]. On s'intéressera ici particulièrement à Dijkstra et A*.

Le principe de **l'algorithme de Dijkstra** est présenté dans les documents [6] et [8], et on traite alors notre polygone comme un graphe pondéré. La solution obtenue passe par différents sommets du graphe et se révèle être optimale.

L'algorithme A* est très bien décrit et détaillé dans l'article [7] de Joseph Razik et dans le document [8]. Il ne parcourt pas l'entièreté de l'espace comme l'algorithme de Dijkstra mais effectue une sélection des trajectoires à analyser. Pour cela, il dispose d'informations sur la position d'arrivée et utilise en conséquence une fonction de coût qui lui indique pour une case donnée, les cases les plus pertinentes à parcourir.

La deuxième approche pour modéliser notre problème est de quadriller l'espace en une grille de déplacements verticaux et horizontaux (voire diagonaux). En ce sens, augmenter la dimension de notre grille a alors pour conséquence d'améliorer la précision des résultats, mais de potentiellement limiter la vitesse d'exécution des algorithmes.

On souhaite toujours déterminer le plus court chemin entre deux points, et cette modélisation offre une plus grande flexibilité dans les choix des éléments présents dans l'environnement (exemple : murs, rivière, montagne, téléporteur, etc...). L'incorporation de ces éléments permettent ainsi de considérer une carte présentant des disparités dans les coûts de déplacement.

Les différents algorithmes précédemment cités (notamment Dijkstra et A*) peuvent alors être implémentés dans un tel environnement, comme le présente le document [8].

De plus, il pourra être pertinent d'étudier des algorithmes plus efficaces mais qui fournissent des solutions non-optimales. Ainsi, le document [4] évoque, par exemple, l'algorithme de recherche « **Best-first** » qui semble allier optimisation de la solution et rapidité d'exécution, sans pour autant viser la solution idéale.

Enfin, si l'on s'intéresse au cas plus concret du déplacement d'un personnage (ou d'un robot) dans une grille, on peut considérer que ce personnage avance en terrain inconnu mais a une idée de la position moyenne de l'arrivée. En ce sens, on rencontre un problème de visibilité qui est largement abordé dans la thèse [9] de Stéphane Riviere. En faisant varier l'incertitude du positionnement de l'arrivée et la zone de visibilité du personnage, on arrive à un problème plus complet, massivement présent dans le domaine des jeux-vidéo par exemple.

Problématique retenue

Comment déterminer avec efficacité et réalisme le plus court chemin reliant deux points d'un environnement donné ?
Comment évaluer la pertinence du choix de la représentation ?

Objectifs du TIPE

1 - Étudier la recherche d'un chemin optimisé dans le cas d'un espace polygonal. Commencer par des méthodes impliquant la triangulation de l'espace, puis comparer avec des algorithmes plus optimaux.
2 - Déterminer le plus court chemin entre deux points distincts dans un environnement présentant de nombreuses disparités.
3 - Essayer de modéliser une découverte progressive de l'espace en tenant compte d'un certain degré de vision.
4 - Évaluer la pertinence des modèles et des représentations choisies vis-à-vis de situations plus concrètes.

Références bibliographiques (ETAPE 1)

[1] FRÉDÉRIC LEGRAND : Triangulation d'un polygone simple : *https://www.f-legrand.fr/scidoc/docmml/graphie/geometrie/polygone/polygone.html*
[2] MARK DE BERG, OTFRIED CHEONG, MARC VAN KREVELD, MARK OVERMARS : Computational Geometry : Algorithms and Applications (Chapitre 9) : *ISBN : 978-3540779735*
[3] GUENDALINA PALMIROTTA : Triangulation de Delaunay : *https://math.uni.lu/eml/projects/reports/MathExp_Palmirotta.pdf*
[4] AMIT PATEL : Introduction to the A* Algorithm : *https://www.redblobgames.com/pathfinding/a-star/introduction.html*
[5] MOHAMED HENY SELMI : Intelligence Artificielle - Algorithmes de recherche : *https://fr.slideshare.net/mohamedhenyselmi/intelligence-artificielle-algorithmes-de-recherche*
[6] DENIS CONDUCHÉ : Algorithme de Dijkstra :

https://www.normalesup.org/~dconduche/informatique/PT/Cours/Dijkstra.pdf
[7] Joseph Razik : Algorithme A* : *https://razik.univ-tln.fr/algorithme-a.html*
[8] Amit Patel : Implementation of A* : *https://www.redblobgames.com/pathfinding/a-star/implementation.html*
[9] Stéphane Riviere : Calculs de visibilité dans un environnement polygonal 2D : *https://tel.archives-ouvertes.fr/tel-00528854/document*

DOT

[1] *Septembre 2021 : Découverte de l'algorithme de recherche A*, et de quelques aspects théoriques sur la triangulation d'un polygone.*
[2] *Octobre 2021 : Première implémentation sur Python d'un algorithme de triangulation de polygone.*
[3] *Novembre 2021 : Amélioration de l'algorithme de triangulation, et correction de l'algorithme de recherche de chemin dans le cas d'un polygone triangulé.*
[4] *Décembre 2021 – Janvier 2022 : Implémentation sur Python du parcours en largeur glouton (GBFS), de Dijkstra, et de A* dans le cas d'un environnement polygonal.*
[5] *Février 2022 : Implémentation sur Python des mêmes algorithmes de recherche dans le cas d'un environnement quadrillé.*
[6] *Mars 2022 : Premiers essais de ces algorithmes sur une carte de Paris simplifiée.*
[7] *Mai 2022 : Finalisation du travail d'optimisation de trajectoire dans les rues de Paris, et applications des algorithmes sur d'autres exemples variés.*
[8] *Juin 2022 : Étude d'un algorithme très intéressant découlant de A* : l'algorithme WA*.*