PATHFINDING

# Qu’est-ce qu’un pathfinding ?

Pathfinding est un terme anglais qui peut se traduire en « **Trouver son chemin** ».

Notre véhicule devra donc être capable à partir d’un point de départ A de se déplacer jusqu’au point B en utilisant le chemin le plus court tout en suivant la route.

**/!\** Notre véhicule ne peut que tourner à droite ou à gauche dans une intersection, il ne peut aller tout droit.

# Découverte de la map

Tout d’abord pour réaliser notre pathfinding il suffit de faire découvrir le chemin de notre map ou de le faire suivre pour une entité donnée.

Voici la map fourni pour le projet est celle-ci :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **5** |  |  |  |  |  |  |
| **4** |  |  |  |  |  |  |
| **3** |  |  |  |  |  |  |
| **2** |  |  |  |  |  |  |
| **1** |  |  |  |  |  |  |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** |

Figure : Map de base

Nous avons maintenant notre map, il va falloir placer des repères sous formes de numéros correspondants à des coordonnées.

La voiture va pouvoir les emprunter dans les différentes intersections présente dans cette map.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **D** | **E** |  | **H** |  | **I** | **P** |
| **5** | **C** | **F** |  | **G** |  |  |  |
| **4** |  |  |  |  |  |  |  |
| **3** | **B** |  |  |  |  | **J** | **O** |
| **2** |  | **L** |  |  |  | **K** |  |
| **1** | **A** | **M** |  |  |  |  | **N** |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** |  |

Figure : Map détaillée

Maintenant que toutes les intersections sont énumérées de 0 (point de départ) à 16 nous allons pouvoir passer au comportement du véhicule.

Cependant quelques points sont à souligner :

* 16 n’est pas forcément l’arrivée
* L’ordre des intersections importe peu, cet ordre tel qu’il est pour permettre de repérer
* Le point 0 n’existe plus juste après le départ véhicule, il est présent pour représenter le point de départ

# Comportement du véhicule

On considère que chaque intersection est un évènement. Le véhicule s’arrêtera donc à chaque intersection numérotée et prendra une décision programmée dans l’Arduino.

Nous allons donc réaliser un tableau qui va permettre de répertorier toutes les intersections.

|  |  |  |
| --- | --- | --- |
|  | **Abscisse** | **Ordonnée** |
| **A** | 0 | 0 |
| **B** | 0 | 2 |
| **C** | 0 | 4 |
| **D** | 0 | 5 |
| **E** | 1 | 5 |
| **F** | 1 | 4 |
| **G** | 3 | 4 |
| **H** | 3 | 5 |

|  |  |  |
| --- | --- | --- |
|  | **Abscisse** | **Ordonnée** |
| **I** | 5 | 5 |
| **J** | 5 | 4 |
| **K** | 5 | 1 |
| **L** | 1 | 1 |
| **M** | 1 | 0 |
| **N** | 6 | 0 |
| **O** | 6 | 2 |
| **P** | 6 | 5 |

Figure 3 : Répertoire de position des intersections

À la suite de cela il sera possible de faire un algorithme de déplacement du robot d’un point connu à un autre. Cependant, il est difficile de dire comment circuler entre deux points. Pour pouvoir remédier à ce problème, on créer un tableau qui va nous permettre de gérer facilement la notion de points et d’indices.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** | **N** | **O** | **P** |
| **A** |  | **1** |  |  |  |  |  |  |  |  |  |  | **1** |  |  |  |
| **B** |  |  | **1** |  |  |  |  |  |  |  |  | **1** |  |  |  |  |
| **C** |  |  |  | **1** |  | **1** |  |  |  |  |  |  |  |  |  |  |
| **D** |  |  |  |  | **1** |  |  |  |  |  |  |  |  |  |  |  |
| **E** |  |  |  |  |  | **1** |  | **1** |  |  |  |  |  |  |  |  |
| **F** |  |  | **1** |  |  |  | **1** |  |  |  |  |  |  |  |  |  |
| **G** |  |  |  |  |  |  |  | **1** |  |  |  |  |  |  |  |  |
| **H** |  |  |  |  |  | **1** |  |  | **1** |  |  |  |  |  |  |  |
| **I** |  |  |  |  |  |  |  |  |  | **1** |  |  |  |  |  | **1** |
| **J** |  | **1** |  |  |  |  |  |  |  |  | **1** |  |  |  | **1** |  |
| **K** |  |  |  |  |  |  |  |  |  |  |  | **1** |  |  |  |  |
| **L** |  |  |  |  |  |  |  |  |  |  |  |  | **1** |  |  |  |
| **M** | **1** |  |  |  |  |  |  |  |  |  |  |  |  | **1** |  |  |
| **N** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **1** |  |
| **O** |  |  |  |  |  |  |  |  |  | **1** |  |  |  |  |  | **1** |
| **P** |  |  |  |  |  |  |  |  | **1** |  |  |  |  |  |  |  |

Figure 4 : Tableau de circulation dans la map

Grâce aux indices de tableau, on peut voir facilement si un déplacement est possible ou non.

Malheureusement à ce stade, on ne peut pas indiquer au robot quelle manœuvre il doit effectuer pour atteindre le point suivant. Il ne sait pas s’il doit tourner à gauche, à droite ou avancer tout droit.

Pour réaliser ceci, il suffit de définir un plan avec sa propre échelle et d’utiliser des vecteurs. Dans ce cadre, on va créer un nouveau tableau qui va contenir les coordonnées de tous les points.

Donc, voici le tableau qui contient les cordonnées : (L’origine de notre repère est 4)

|  |  |  |
| --- | --- | --- |
| **Points** | **Abscisses** | **Ordonnées** |
| 1 | 1 | 3,5 |
| 2 | 0 | 3,5 |
| 3 | 0 | 2,5 |
| 4 | 0 | 1,5 |
| 5 | 0 | 0 |
| 6 | 1 | 0 |
| 7 | 1 | 1 |
| 8 | 1 | 2,5 |
| 9 | 3 | 2,5 |
| 10 | 3 | 3,5 |
| 11 | 4 | 1 |
| 12 | 4 | 1,5 |
| 13 | 4 | 3,5 |
| 14 | 4,5 | 0 |
| 15 | 4,5 | 1,5 |
| 16 | 4,5 | 3,5 |

Figure 5 : Vecteurs de déplacement

Une fois ce tableau que nous avons réalisé ce tableau il suffit simplement de faire le calcul des vecteurs d’un point à un autre pour avoir la direction du robot.

UML

Une image contenant capture d’écran

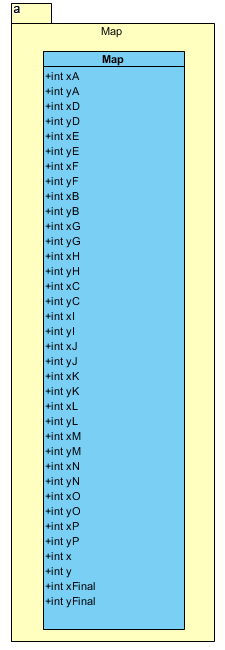
Description générée avec un niveau de confiance très élevé

Ce diagramme définie l’énumération Move pour les déplacements vers la gauche et la droite

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

Ce diagramme montre les « interfaces » de movement Turn, RedresseRight/Left, Stop, ToutDroit, ChoixAction, Check et contient également les char Intersec, StackIntersec, Nextintersec



Map contient les coordonnées des différentes intersections (A à P).

Une image contenant capture d’écran

Description générée avec un niveau de confiance élevé

Diagramme du main dépendant de Movement, Enum, Map et Grove\_I2C

Une image contenant texte, carte

Description générée avec un niveau de confiance très élevé

Diagramme de packages montrant les differents imports vers le main