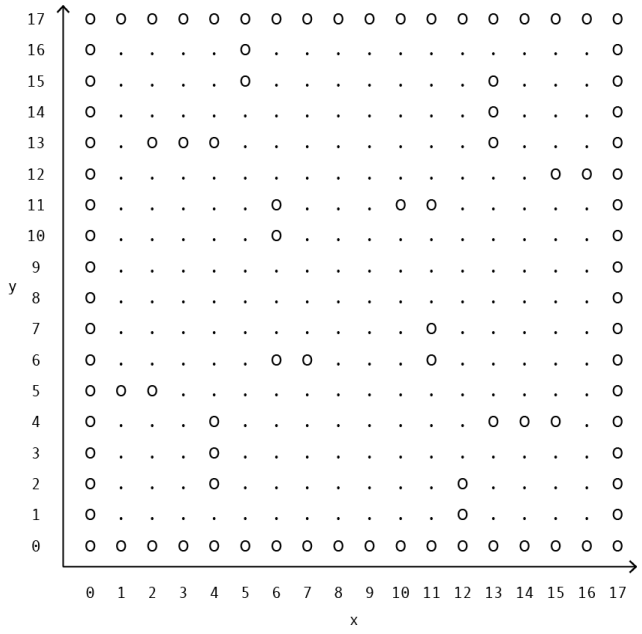
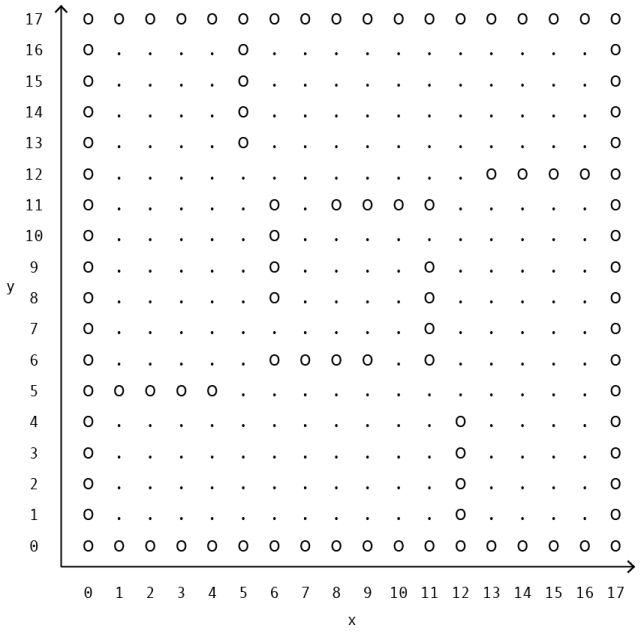
université PARIS-SACLAY | IUT DE CACHAN

| S1-BUT-GEii<br>APP-0<br>Sept.-2024 | **APP-zéro**<br>**Cas n°3, programme complet**<br>Rapport d'équipe à remettre au tuteur<br>Mercredi matin à 8h30, avant l'évaluation formative. | **Rapport R-3**<br>**Cas n°3** |
|---|---|---|

Nous ne considérons plus des obstacles isolés. Les obstacles peuvent former des lignes horizontales ou verticales qui ne se touchent pas. Les lignes peuvent cependant toucher les bords par une de leurs extrémités (mais pas sur toute leur longueur). Voici des exemples de pistes.




*Exemple de configuration pour le cas n°3*




*Exemple de configuration pour le cas n°3*

On ne connaît pas la direction initiale des robots.

En fonction du numéro de l'équipe, le robot démarre d'un des 4 coins. Plus précisément :

- ➔ Équipe 1 (robot n°1): départ du coin Nord-Ouest, case de coordonnées 1,16
- ➔ Équipe 2 (robot n°2): départ du coin Nord-Est, case de coordonnées 16,16.
- ➔ Équipe 3 (robot n°3): départ du coin Sud-Est, case de coordonnées 16,1.
- ➔ Équipe 4 (robot n°0) : départ du coin Sud-Ouest, case de coordonnées 1,1.

Pour le code en C final utilisé dans le simulateur (Mercredi (jour 3) uniquement) :

Les structure de contrôle algorithmiques **SI-SINON** et **TANT-QUE** seront remplacées par leur équivalent en C « **if-else** » et « while » :

**if (condition){instruction(s)} else {instruction(s) alternative(s)}**

et

**while (condition){instruction(s)}**

Les robots utiliseront les fonctions en langage C qui portent leur numéro après le préfixe « ROB », par exemple ROB3 pour le robot n°3 :

**SI ( ROB3_Direction ( ) != OUEST) { ROB3_Tourne_a_gauche ( ) ;}**

Le code commencera par l'inclusion d'un fichier d'entête (.h) contenant le déclaration des fonctions utilisables pour ce robot, et définira ainsi la fonction associée à l'algorithme :

| Pour le robot n°0 (équipe 4) : | Pour le robot n°1 : |
|---|---|
| //---------- fichier mr2d2_0.c ----------<br>#include "mr2d2_0.h"<br><br>void ROB0_AlgorithmeSO(void) {<br>//décrire l'algorithme du robot n° 0 ici ...<br><br>} | ---------- fichier mr2d2_1.c ----------<br>#include "mr2d2_1.h<br><br>void ROB1_AlgorithmeNO(void) {<br>// décrire l'algorithme du robot n° 1 ici …<br><br>} |
| **Pour le robot n°2 :** | **Pour le robot n°3 :** |
| ---------- fichier mr2d2_2.c ----------<br>#include "mr2d2_2.h"<br><br>void ROB2_AlgorithmeNE(void) {<br>// décrire l'algorithme du robot n° 2 ici …<br><br>} | ---------- fichier mr2d2_3.c ----------<br>#include "mr2d2_3.h"<br><br>void ROB3_AlgorithmeSE(void) {<br>// décrire l'algorithme du robot n° 3 ici …<br><br>} |

La solution proposée doit répondre aux contraintes suivantes :

- Au moins un sous-programme est utilisé.
- L'algorithme est parfaitement **compris par tous** les étudiants du groupe. Chacun est capable d'expliquer le rôle de chaque partie.