université PARIS-SACLAY | IUT DE CACHAN

S1-BUT-GEii APP-0 Sept..-2025	**APP-zéro** Semaine d'initiation à l'Apprentissage Par Problèmes (APP) en petits groupes tutorés. **Livret étudiant**	**Livret étudiant**

Concours de robotique : MR2D2 total reboot

Ce livret contient les informations pour réaliser
les activités prévues dans le cadre de l'APP-0 durant la semaine 1

Denis Pénard – Jacques-Olivier Klein – Gilles Raynaud – Yannick Le Paih – Patrick Ruiz

Cet APP-0 est adapté de celui pratiqué en septembre 2017 à l'ECL (UCL, Louvain-la-Neuve).
La situation-problème originale à l'UCL, intitulée « Game of Spies » était inspirée de l'univers d'une célèbre série.

Acquis d'apprentissage visés

Méthodologiques

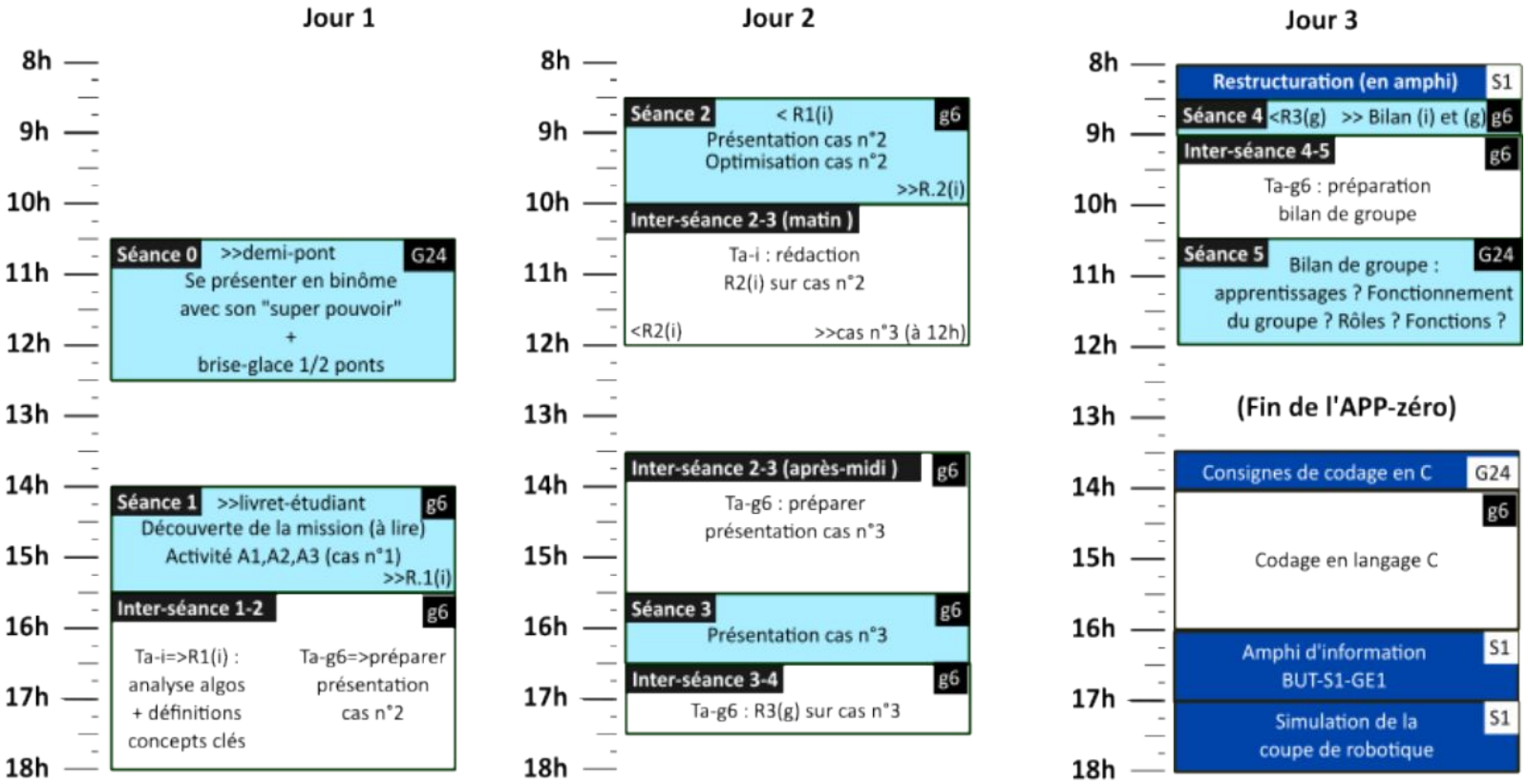
1. Décrire le principe de l'apprentissage par problèmes, rôle des enseignants, rôle de l'équipe,
2. Faire fonctionner votre équipe dans un environnement d'APP ou de projet,
3. Réaliser un bilan réflexif (bilan méta-cognitif) pour en tirer des leçons profitables aux apprentissages avenir,

Disciplinaires

1. Définir des concepts de base en informatique, algorithme, instruction, condition, boucle, test, primitive, opérateur, itération.
2. Décomposer un problème en sous-programmes,
3. Traduire un algorithme simple en pseudo-code,
4. Prévoir l'effet de l'exécution d'un algorithme,
5. Implémenter un algorithme en langage C.

Table des matières

Planning APP-zéro



Légende

Séance tutoré | Séance en autonomie | Cours

- g6 : Travail en petits groupes (6)
- G24 : Travail en groupe TD (24)
- S1 : Amphi avec toute la promo S1
- >>R.j : Les étudiants recoivent le document R.j
- <R.j(i) : Chaque étudiant livre son rapport individuel (i) n°j
- <R.j(g) : Les étudiants livrent leur rapport de groupe (g) n°j

Ta-i : Travail individuel en autonomie

Ta-g6 : Travail de petit groupe (6) en autonomie

Séance 1 – Découverte de la mission Lundi (jour 1) 14h-15h30

Le but de cette première séance tutorée est de :

- Découvrir la situation-problème qui va être utilisée durant toute la semaine,.
- Découvrir quelques notions de base de l'informatique,
- Prendre vos marques dans l'organisation du travail de groupe.

Étape	Déroulement de l'étape
Étape 0	SI vous le jugez nécessaire, définissez des rôles ou des fonctions pour travailler efficacement en groupe. Inscrivez les rôles ou fonctions attribués dans le kit de travail d'équipe, page 32. Ce n'est pas définitif. Vous aurez l'occasion d'y revenir.
Étape 1	Prendre connaissance de la situation problème et de ses annexes 1 à 5 (pages 5-11).
Étape 2	Réaliser les exercices de la section activité A.1 et A.2 (pages 12-13), puis mettre en commun.
Étape 3	Proposer une solution au problème de base (Activité A.3, cas n°1, pages 14-), individuellement, puis en groupe
Étape 4	Construire la solution du groupe et rédiger l'algorithme retenu page 18.
Étape 5	Le tuteur donne les consignes pour le travail à faire de manière autonome durant l'après-midi (cf. pages 19-20).

Situation problème : « MR2D2 : Total reboot ! »

La compétition est dans une semaine et plus personne ne sait comment fonctionne l'algorithme. En général, au bout de 30 secondes, le robot se met à tourner en rond sur lui même indéfiniment, ou il s'arrête bêtement devant un obstacle. C'est d'ailleurs ce drôle de comportement qui lui a valu son nom : MR2D2 ! Hebba a remarqué que la moitié du code est commentée et ne sert à rien. Youssef a toujours dit qu'il fallait une démarche rigoureuse pour valider l'algorithme, mais personne ne l'a écouté. Maintenant c'est trop tard, l'algorithme est devenu beaucoup trop compliqué. Il fallait prendre une décision radicale.

C'est fait ! 5 pour, 2 contre, 1 abstention. C'est Darren et Melissande qui ont eu cette drôle d'idée, le total reboot : donner l'algorithme à faire à partir de zéro, à des yeux neufs : la nouvelle promotion de S1. Autant dire que Jonas n'y croit pas du tout. Ça fait un an qu'il travaille sur cet algorithme, ce ne sont pas des S1 qui ne connaissent rien au langage C qui vont faire mieux en 1 semaine. Ils ne savent même pas ce qu'est une instruction, une variable, une fonction… Mais pour Romain, ils peuvent y arriver. D'ailleurs, il suffit qu'ils se limitent à deux structures de contrôle : la boucle (while) et le test (if).

Aline, elle, est convaincue. Les capteurs, fonctionnent parfaitement, on détecte les obstacles devant le robot à 60 cm comme la zone blanche d'arrivée. Yanis le confirme, les moteurs et l'asservissement fonctionnent aussi : si on se limite à trois fonctions, avancer de 50 cm, tourner à gauche ou à droite de 90°, c'est jouable, ça revient à se déplacer sur un quadrillage, et c'est facile de faire des simulations. Justement, Jonas avait écrit un simulateur, il va peut-être enfin servir à quelque chose. Pour **Chafik, le plus important, c'est que tout le monde soit capable d'expliquer comment fonctionne l'algorithme.** Si c'est encore un.e étudiant.e, tout.e seul.e, qui le fait entièrement dans son coin, ça va recommencer, le syndrome de l'usine à gaz.

Youssef explique son idée de validation progressive : il faut montrer que l'algorithme parvient à atteindre l'arrivée pour des configurations de complexité croissante : d'abord sans obstacle, puis avec des obstacles isolés, puis avec des lignes d'obstacles isolées, sans angles, puis en gérant les obstacles sur les bords et éventuellement les obstacles en angles, etc. Jonas est sceptique, mais au point où on en est, qu'est-ce qu'on risque ?

Annexe 1 : Règlement simplifié du concours de robotique GEii, édition 2009-2017 (article 5) .

- Les robots évoluent dans un carré de 8m x 8m entouré d'une bordure. L'espace de 8m x 8m est recouvert d'une moquette bleue foncée. Les quatre coins du carré, correspondant aux zones de départ et d'arrivée, sont recouverts d'un papier autocollant blanc.
- Chaque robot part d'un coin du carré et doit parvenir au coin opposé en moins de 90 secondes. Il doit, pour cela, éviter les obstacles, les autres robots et les zones blanches qui ne correspondent pas à son coin d'arrivée.
- Pour valider son arrivée, le robot doit crever le ballon qu'il transporte.
- Le premier robot arrivé qui fait crever son ballon est le vainqueur.
- Si un robot fait crever son ballon ailleurs que dans sa zone d'arrivée, ou s'il passe sur une zone d'arrivée qui n'est pas la sienne, ou s'il n'est pas sorti de sa zone de départ après 9 secondes, le robot est déclaré perdant et doit être retiré de la piste.

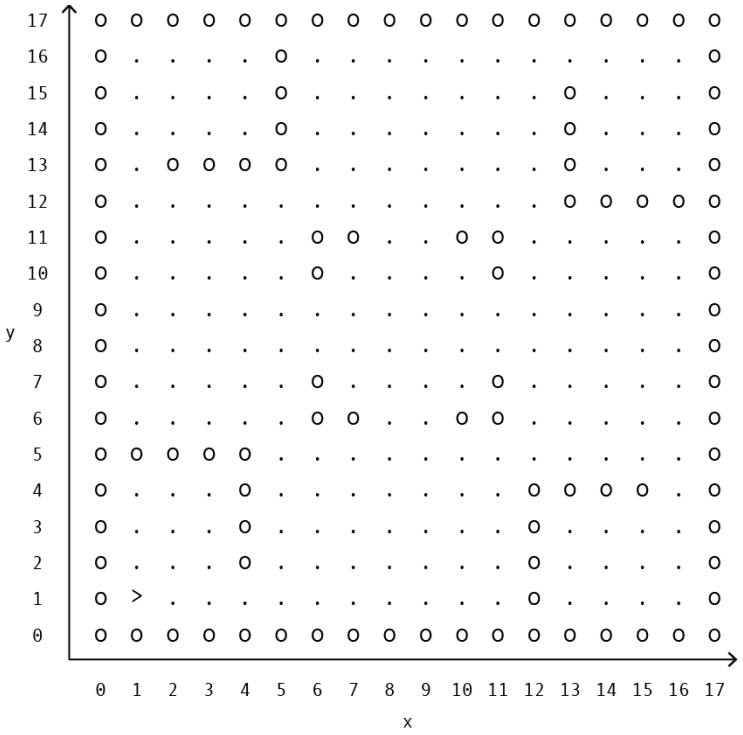
Le règlement complet du concours est ici : https://bit.ly/reglement_vierzon_2017

(il faut être connecté à l'environnement numérique de travail ecampus.paris-saclay.fr)

Annexe 2 : Le simulateur de robot développé par Jonas

Pour vous permettre de valider votre programme, Jonas a développé un simulateur. Jonas nous a promis que nous pourrions l'utiliser Mercredi (jour 3), mais, manque de chance, il est pour l'instant en vacances en Patagonie, sans réseau. Aucun essai ne sera donc possible avant la validation finale de Mercredi (jour 3). Aussi, il faudra vous convaincre, par une réflexion collective approfondie, que votre programme sera correct et efficace.

Dans le simulateur de Jonas, les robots se déplacent par rotation à gauche ou à droite de 90° ou en avançant de 50cm. Tout se passe donc, comme s'ils se déplaçaient sur un quadrillage de 16x16 cases. Le simulateur considère en fait un quadrillage 18x18 incluant les bords inaccessibles. L'origine du système se situe dans le coin inférieur gauche, de coordonnées (0,0). Les case d'abscisses ou d'ordonnées 0 ou 17 correspondent aux bords et ne sont pas accessibles.



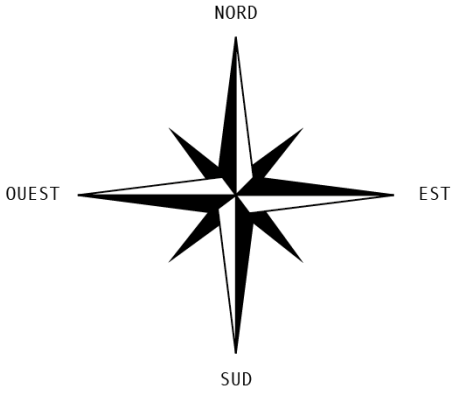


Figure 1: Exemple de piste comprenant des obstacles et système de coordonnées du simulateur.

Chaque robot est repéré par un signes (^, v, >, <) indiquant dans quelle direction (Nord, Sud, Est, Ouest) il est dirigé. Chaque rotation ou déplacement dure une demi-seconde, chaque robot dispose donc de 180 déplacements (90 secondes) pour parvenir au coin opposé.

Annexe 3 : Déplacement du robot

Que ce soit en vrai, sur le terrain, ou dans le simulateur, le pilotage du robot est obtenu avec une séquence d'ordres de déplacement très simples, nommés « opérations primitives ». Chaque déplacement dure une demi-seconde.

Voici la liste des ordres auxquels obéissent les robots :

Opérations primitives	Description
ROBOT_Avance ( ) *	Le robot tente d'avancer d'une case devant lui. S'il rencontre un obstacle ou une bordure, rien ne se produit. S'il rencontre un autre robot, le choc est fatal, les deux robot s'arrêtent définitivement.
ROBOT_Tourne_a_gauche ( ) *	Le robot effectue un quart de tour à gauche. S'il était dirigé vers le Nord, il termine dirigé vers l'Ouest.
ROBOT_Tourne_a_droite ( ) *	Le robot effectue un quart de tour à droite. S'il était dirigé vers le Nord, il termine dirigé vers l'Est.
ROBOT_Creve_ballon_et_stop ( ) *	En plus des déplacements, le robot arrivé à destination doit signaler son arrivée en faisant crever son ballon. A cet effet, cette opération déclenche un dispositif crève-ballon.

* Les parenthèses sont nécessaires en langage C. Elles ne sont pas nécessaires quand on décrit un algorithme en pseudo-code. Voir annexe 5.

En plus des quatre opérations primitives, des informations peuvent être obtenues grâce aux capteurs du robot. Ces opérations produisent une valeur booléenne (vrai / faux) ou une valeur entière (un nombre de 1 à 16) ou une valeur symbolique (NORD, SUD, EST, OUEST).

Opération-question	Type de donnée produite	Description
ROBOT_Peut_avancer ( )	Booléen (vrai / faux)	Produit une valeur VRAI s'il n'y a pas d'obstacle devant le robot. Produit une valeur FAUX si une bordure, un obstacle ou un autre robot se trouve devant le robot.
ROBOT_Arrive ( )	Booléen (vrai / faux)	Produit la valeur VRAI si le robot se situe au dessus de sa zone blanche d'arrivée. Renvoie la valeur FAUX dans toutes les autres situations.
ROBOT_Pos_x ( )	Entier (comprise entre 1 et 16)	Produit une valeur entière indiquant la position en X du robot (dans le quadrillage 18 x 18). Les positions d'abscisse 0 et 17 correspondent à des bordures.
ROBOT_Pos_y ( ):	Entier (comprise entre 1 et 16)	Produit une valeur entière indiquant la position en Y du robot (dans le quadrillage 18 x 18). Les positions d'abscisse 0 et 17 correspondent à des bordures.
ROBOT_Direction ( )	Symbolique (parmi NORD, SUD, EST, OUEST)	Produit la direction courante vers laquelle est dirigé le robot (NORD, SUD, OUEST, EST).

Annexe 4 : Eléments du langage autorisés

Les instructions sont exécutées l'une après l'autre, sauf dans les deux cas suivants.

1 : Des instructions **conditionnelles**, dont l'exécution dépend d'une condition.

La syntaxe est la suivante :

```
si (condition) alors
 |                    bloc d'instruction(s) conditionnel
```

Ou, avec un bloc d'instruction alternatif :

```
si (condition) alors
 |                    bloc d'instruction(s) conditionnel
sinon
 |                    bloc d'instruction(s) alternatif
```

Par exemple :

```
si (ROBOT_Direction ==  EST) alors
 |                    ROBOT_Tourne_a_gauche        ← bloc d'instruction(s) conditionnel
 |                    ROBOT_Avance
suite…                                            ← fin du bloc d'instruction(s) conditionnel
```

Si la condition est VRAI, le bloc d'instruction conditionnel est exécuté, si la condition est FAUX, le bloc d'instruction conditionnel est sauté et l'exécution se poursuit à l'instruction qui suit le bloc conditionnel.

Autre exemple avec bloc d'instructions alternatif :

```
si (ROBOT_Direction ==  EST) alors
 |                    ROBOT_Tourne_a_gauche        ← bloc d'instruction(s) conditionnel
sinon
 |                    ROBOT_Tourne_a_droite        ← bloc d'instruction(s) alternatif
suite…
```

2 : Des **boucles tant-que** dont la syntaxe est la suivante :

```
tant que (condition) faire
 |                    bloc d'instruction(s) à répéter
```

Par exemple :

```
tant que ( ROBOT_Pos_x  <  8  ET NON ROBOT_Arrive)  faire
 |                    ROBOT_Avance
… suite
```

L'exécution se répète, tant que la condition, évaluée avant chaque exécution du bloc d'instruction, est VRAI. Si la condition est évaluée à la valeur FAUX, la répétition cesse et le programme se poursuit à l'instruction qui suit le bloc.

- Dans une **condition**, on peut trouver une combinaisons d'opérateurs logiques (ET, OU, NON), d'opérateurs arithmétiques (+, -, /) et des d'opérateurs relationnels (<, >, ≤, ≥, ==, ≠).

  Notez le « double-égal » (==) pour tester l'égalité, à ne pas confondre avec le « simple-égal » (=) utilisé pour un autre usage, et interdit à ce stade.
- Les instructions conditionnelles, les boucles, et les blocs d'instructions peuvent se combiner ou s'imbriquer pour décrire un **algorithme**. C'est ce que nous allons voir tout de suite.

Annexe 5 : Notion d'algorithme

Un algorithme est une description d'une procédure systématique précisant les instructions à exécuter. L'encadré ci-dessous montre un exemple d'algorithme utilisé pour illustrer ce concept. Sur la gauche, vous trouvez une description de l'algorithme en pseudo-code et sur la droite le code en langage C. L'algorithme présenté traduit le comportement suivant : « Tant le que le robot peut bouger, avancer d'une case dans la direction courante ».

### Algorithme n°1	
TANT QUE ROBOT_Peut_avancer FAIRE	**while (ROBOT_Peut_avancer ( ))**
|	**{**
| ROBOT_Avance	** ROBOT_Avance();**
|	**}**

La figure 2, montre la situation initiale. Le robot est matérialisé par le signe >. Dans ce cas, si on exécute l'algorithme 1, le robot ne pourra se déplacer vers la droite que de quatre cases. Ensuite, comme il n'y a pas d'autre instruction, l'algorithme n°1 se termine.

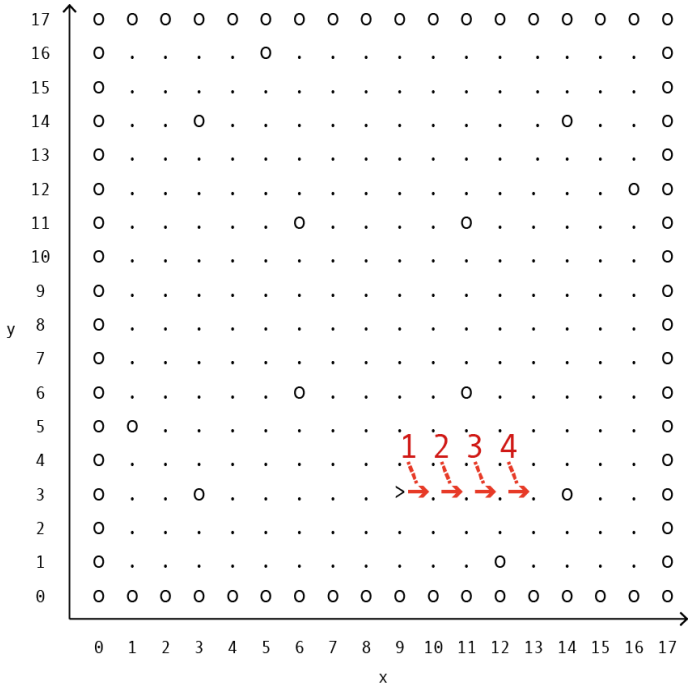


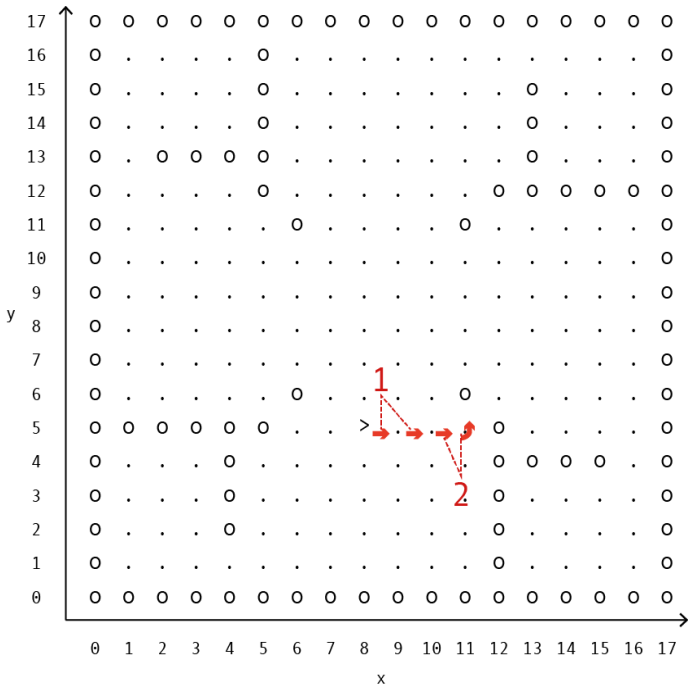
Figure 2 : Robot initialement en position (9,3) orienté vers l'EST et déplacements lors de l'exécution de l'algorithme n°1.

L'encadré suivant montre un exemple d'algorithme plus complexe qui combine une boucle et un test. L'algorithme présenté traduit le comportement suivant : « Tant que le robot peut bouger, il effectue séquentiellement les deux actions suivantes : il avance d'une case selon leur direction courante ; ensuite, deux situations peuvent se produire : soit il peut bouger et il avance encore d'une case, soit il est coincé et tourne sur lui-même de 90 degrés dans le sens anti-horaire».

```
### Algorithme 2
TANT QUE  ROBOT_Peut_avancer  FAIRE
|
|    ROBOT_Avance
|    SI   ROBOT_Peut_avancer  ALORS
|    |
|    | ROBOT_Avance
|    |
|    SINON
|    |
|    |  ROBOT_Tourne_a_gauche
|    |
|
```

```
while ( ROBOT_Peut_avancer() )
{
  ROBOT_Avance();
  if ( ROBOT_Peut_avancer())
  {
     ROBOT_Avance();
  }
  else
  {
     ROBOT_Tourne_a_gauche();
  }
}
```

La figure ci-dessous montre l'évolution d'un robot, avec cet algorithme, dans une configuration particulière de terrain.



1. Le robot initialement en position (8,5) dirigé vers l'EST est libre d'avancer. Il va donc entrer dans la boucle et avancer d'une case. Étant toujours capable d'avancer, il va exécuter l'instruction conditionnelle et avancer d'une nouvelle case. Suite à la première itération de la boucle, il aura donc avancé de deux cases en tout et se trouve en position (10,5), toujours dirigé vers l'EST.
2. Après cela, étant donnée qu'il peut toujours avancer, la boucle sera exécutée de nouveau. Il avance alors d'une case et cette fois-ci, se retrouve coincé par un obstacle. Il va donc exécuter le bloc d'instruction alternatif (sinon) et tourner sur lui-même vers la gauche, en restant sur la même case. A ce stade, le robot est sur la case (11,5) dirigé vers le NORD.
3. Avant de commencer une éventuelle 3^{e} itération de la boucle tant-que, le robot ne peut pas avancer. Il ne reprendra donc pas l'exécution de la boucle et l'algorithme n°2 se termine ici, avec le robot en position (11,5) dirigé vers le NORD.

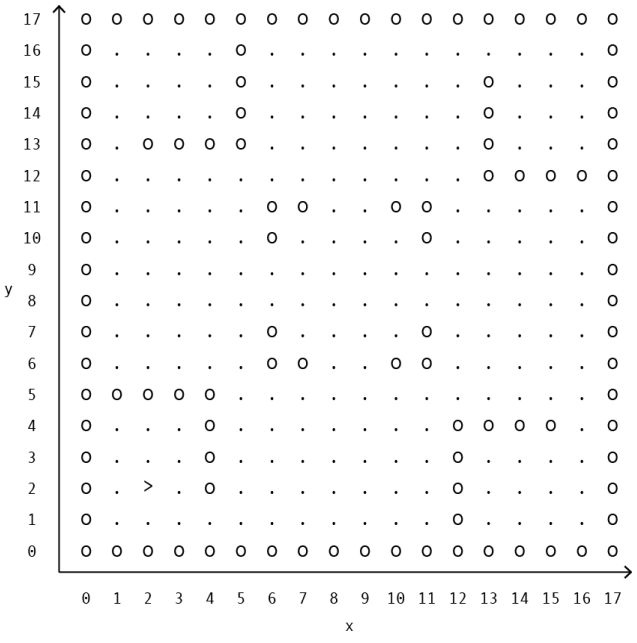
Activités de la séance n°1 : Concept d'algorithme Lundi (jour 1) 14h-15h30

Activité A.1 : Simuler un algorithme et prévoir le déplacement du robot

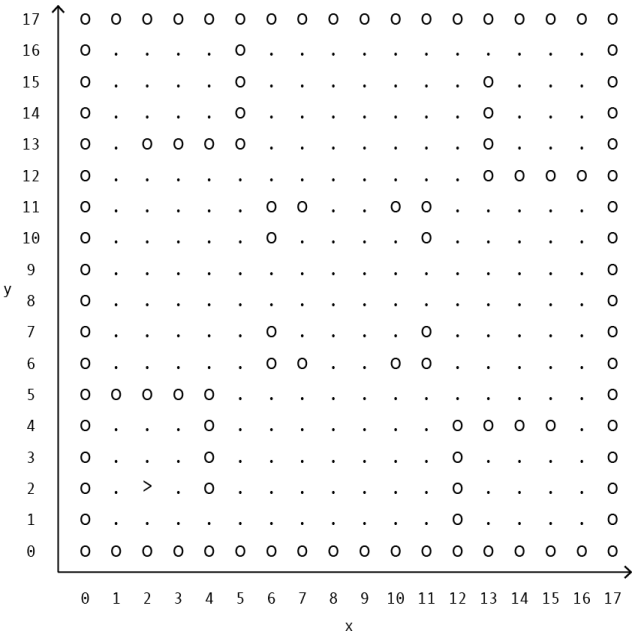
Pour la situation suivante, indiquez sur le dessin de gauche le résultat de l'exécution de l'**algorithme n°1**, en supposant que le robot est initialement orienté vers l'est. Faire de même avec l'**algorithme n°2** sur le dessin de droite.

```
### Algorithme n°1
TANT QUE  ROBOT_Peut_avancer  FAIRE
|
|   ROBOT_Avance
|
```

```
### Algorithme 2
TANT QUE  ROBOT_Peut_avancer  FAIRE
|
|   ROBOT_Avance
|   SI   ROBOT_Peut_avancer  ALORS
|   |
|   | ROBOT_Avance
|   |
|   SINON
|   |
|   |  ROBOT_Tourne_a_gauche
|   |
|
```



*Résultat de l'exécution de l'**algorithme n°1***



*Résultat de l'exécution de l'**algorithme n°2***

Activité A.2 : Ecrire un algorithme à partir d'une description en français

Traduisez le comportement suivant en un algorithme :
L'algorithme doit continuer à s'exécuter tans que le robot ne se trouve pas dans la ligne tout en haut.
Si le robot peut se déplacer dans la direction courante, il se déplace.
Quand il ne peut plus avancer à cause d'un obstacle ou d'un bord, il tourne sur lui-même de 90 degrés vers la droite.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Activité A.3 : Concevoir un algorithme : cas des obstacles simples (cas n°1)

Dans un premier temps, on considère un problème simplifié :

- Le terrain n'est constitué que d'obstacles occupant exactement une case (mais possiblement au bord du terrain).
- Votre robot est seul sur la piste. Il ne risque donc pas de rencontrer d'autres robots.
- Le robot démarre en bas à gauche (1,1) dirigé vers l'est.

Il n'y a pas d'obstacle dans les trois cases à proximité de la zone de départ.

Attention, il ne faut pas passer sur la zone de départ des autres robots, à part celle qui constitue votre zone d'arrivée, bien sûr.

La figure 5 illustre un exemple de cette situation simplifiée. Attention, ce n'est qu'un exemple !

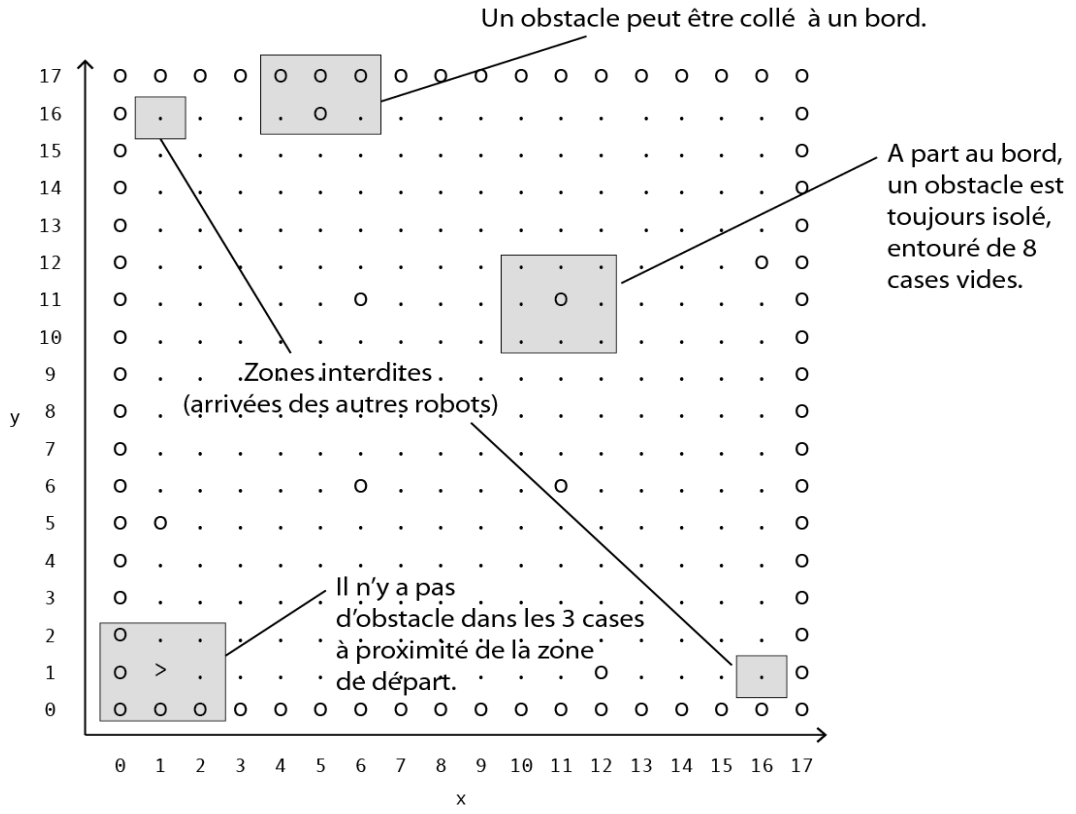


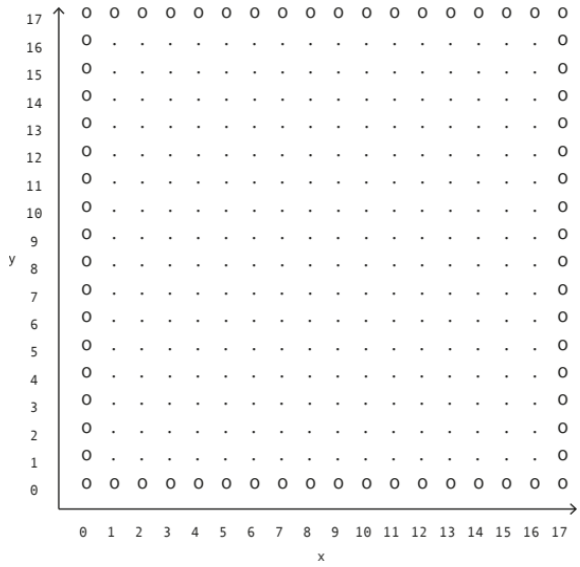
Figure 5 : Robot en position (1,1) orienté vers l'est

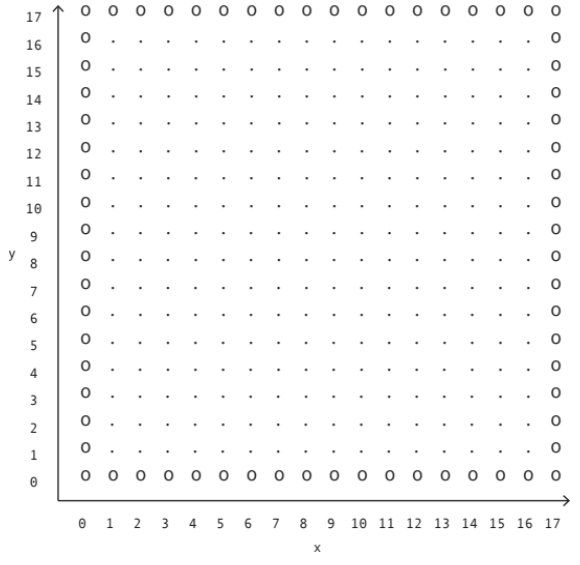
Durant la première phase, vous travaillerez en groupes. L'objectif à atteindre est d'aboutir **collectivement** à un algorithme permettant à votre robot de rejoindre le coin opposé.

1. Réfléchissez **d'abord individuellement** à un algorithme et rédigez-le ci-dessous. Dessinez la trajectoire que devraient suivre votre robot s'il exécutait l'algorithme que vous avez élaboré.
2. Donnez ensuite votre algorithme à la personne se trouvant à votre gauche.
3. Exécutez manuellement l'algorithme que vous avez reçu de la personne se trouvant à votre droite.

Ma proposition d'algorithme pour le cas n°1.

Quatre tableaux vierges pour tester votre algorithme.





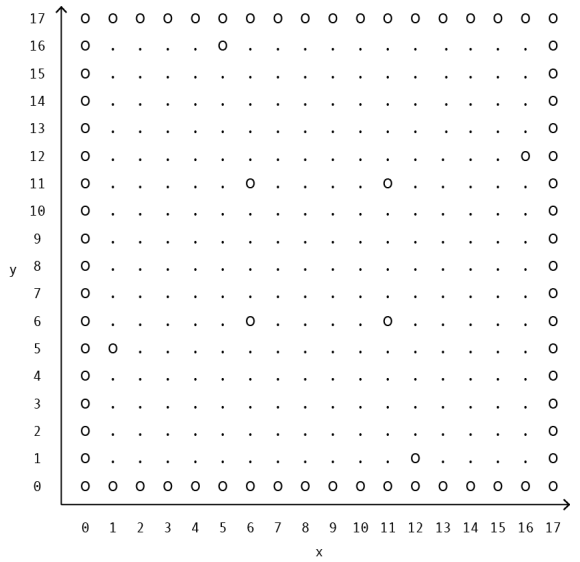
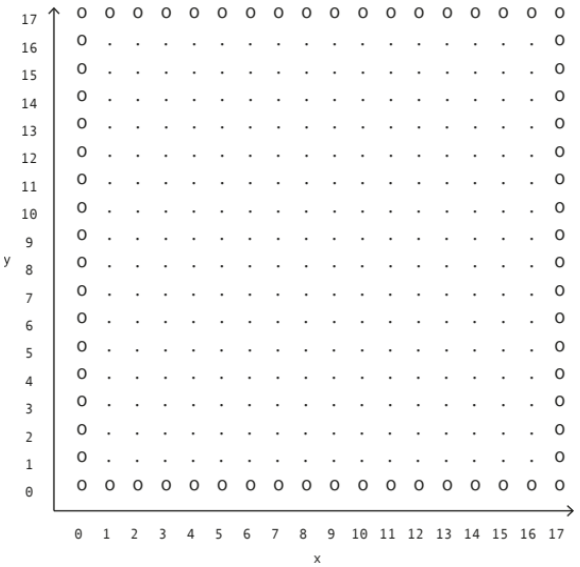
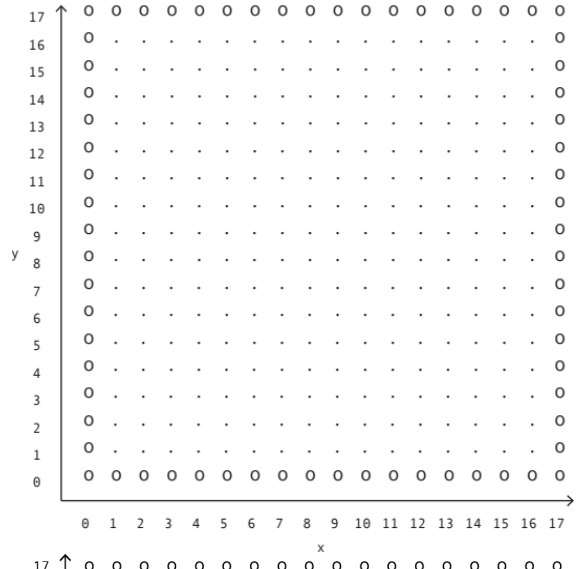


Figure 6 : Résultat de l'exécution de mon algorithme





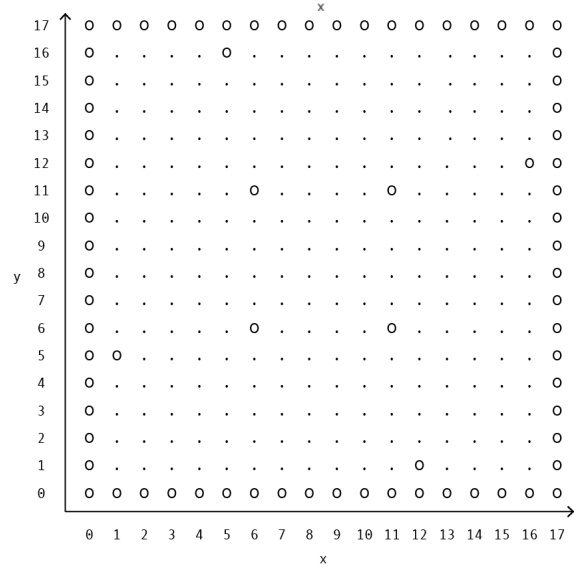


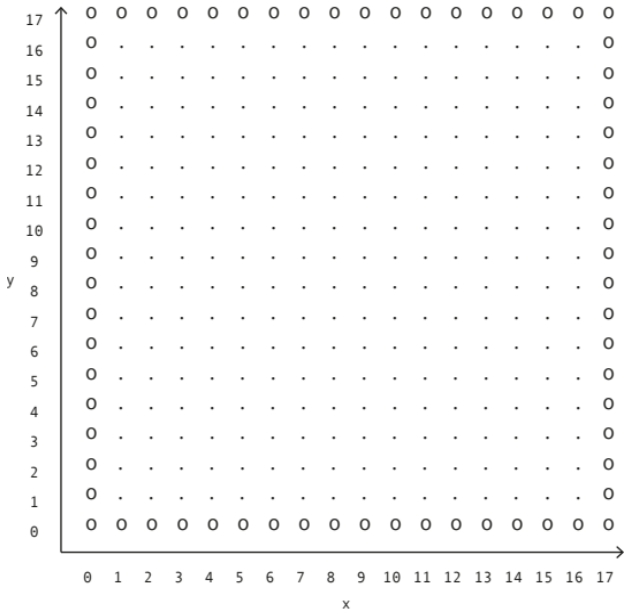
Figure 7 : Résultat de l'exécution de l'algorithme de mon voisin de droite

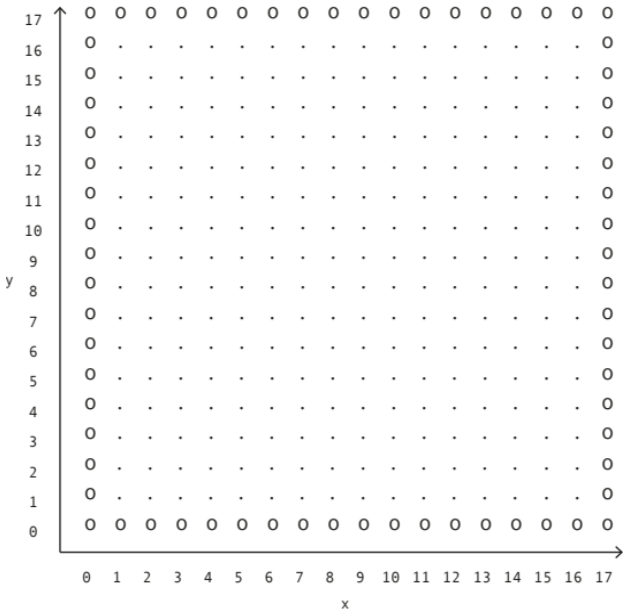
Grille de comparaison d'algorithmes

1.

A : Tout à fait B : Plutôt oui C : Moyennement D : Plutôt non E : Pas du tout	Solution n°					
Critères	1	2	3	4	5	6
Concision – L'algorithme est concis.						
Complexité – L'algorithme reste simple, il n'y a pas trop de boucles ou tests imbriqués.						
Sécurité – Il est vraiment évident à sa lecture que l'algorithme atteint bien l'objectif.						
Efficacité – L'algorithme rejoint le coin opposé en faisant un minimum de mouvements, dans tous les cas.						

Quatre tableaux vierges pour tester votre algorithme.





Solution du groupe pour le cas n°1.

Inter-séance 1-2 – Rapport R1 et Cas n°2. Lundi (jour 1) 15h30-18h

Attention ! Ce travail doit être fait AVANT de vous rendre à la séance 2.

Consignes pour travailler en autonomie

Durant la séance 1, vous avez travaillé sur le problème de base (cas n°1) et votre tuteur vous a assisté durant toute la séance. Vous allez maintenant travailler sur une variante (cas n°2) et devoir, en groupe et de manière autonome, poursuivre la résolution du problème.

- Le travail **individuel** à réaliser en vue de la séance 2 consiste à :
  1. Compléter le rapport de mission **individuel** (R.1) à rendre le Mardi à 8h30,
  2. Lire l'énoncé du cas n°2. → Si votre équipe ne parvenait pas à trouver rapidement une solution satisfaisante au cas n°2 (voire au cas n°1), vous pouvez commencer par résoudre un problème plus simple, par exemple, sans obstacle, ou avec des obstacles connus.
- Le travail **de groupe** à réaliser en vue de la séance 2 consiste à :
  1. S'organiser pour élaborer une réponse argumentée au cas n°2, réponse à laquelle tous les membres du groupe adhèrent,
  2. Rédiger la réponse du groupe,
  3. Préparer au tableau une présentation de l'état d'avancement de vos travaux pour le tuteur,
  4. Désigner à l'avance, un étudiant pour cette présentation, qui sera informelle et interactive **mais doit être préparée.**

Différentes étapes facilitent la résolution, en groupe, d'une situation-problème. Vous pouvez vous en inspirer pour organiser votre travail en autonomie. Nous vous invitions à appliquer cette démarche pour le cas n°2.

Des étapes pour faciliter le travail en groupe

L'apprentissage par problème (APP) est une approche pédagogique basée sur l'analyse et la résolution d'un problème qui s'inspire d'une situation de la vie quotidienne ou professionnelle.

C'est une méthode structurée d'apprentissage, qui alterne des temps de travail en groupe (supervisé par un tuteur ou autonome) et des périodes de travail individuel (recherche personnelle, lectures, étudie, rédaction, etc.).

Le déroulement d'un APP suit systématiquement les 7 étapes suivantes :

Étape 1 : Comprendre la tâche. En groupe, les étudiants prennent connaissance de l'énoncé du problème. Ils analysent et clarifient l'énoncé, en vue de le traiter correctement par la suite. Les étudiants reformulent ensuite l'énoncé, pour s'assurer mutuellement qu'ils ont bien compris le problème à traiter.

Étape 2 : Faire le point. Les étudiants font le point sur les connaissances dont il disposent déjà pour traiter le problème.

Étape 3 : Formuler des pistes. Ils définissent le champ de connaissances à explorer pour combler leurs lacunes, formulent des pistes pour traiter le problème et établissent un plan d'action pour s'assurer que chaque membre du groupe sache ce qu'il a à faire durant le temps de travail individuel.

Étape 4 : Réaliser le plan d'action. Conformément au plan établi, les étudiants réalisent, individuellement des activités de recherche, d'étude et, le cas échéant, de conception et de réalisation. Ce travail se déroule normalement sur 6 à 8 heures.

Étape 5 : Mettre en commun. Les membres du groupe se retrouvent ensuite pour mettre en commun le apports/les acquis individuels.

Étape 6 : Construire une solution. Cette étape consiste à construire, ensemble, une solution au problème.

Étape 7 : Faire un bilan des apprentissages. Le groupe établit un inventaire des apprentissages réalisés et des points à approfondir. Les étudiants établissent également un bilan du fonctionnement du groupe, ainsi qu'un bilan du travail individuel.

Variante du problème initial (Cas n°2)

Nous considérons toujours des obstacles isolés, mais nous allons maintenant prendre en compte :

- la présence d'autres robots sur la piste. Contrairement à un obstacle, si votre robot tente d'avancer vers l'un d'eux, il le percutera et sera détérioré. Dans ce cas, il s'arrête définitivement. Il convient donc d'éviter absolument d'avancer si la voie n'est pas libre.
- la possibilité de démarrer dans n'importe qu'elle direction (NORD, SUD, EST , OUEST) et non plus seulement dirigé vers l'EST.
- Les robot ne partent plus tous du coin Sud-Ouest (1,1). 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 ce faire, il est possible de définir des **sous-programmes** et les utiliser dans l'algorithme du cas n°2.

Par exemple, un sous-programme **SP1 : ROBOT_Tourne_toi_a_l_EST** qui oriente le robot à l'EST, quelle que soit sa direction initiale pourrait être utile.

Voici quelques idées de sous-programmes :

- SP1 : A partir de n'importe quelle orientation, diriger le robot vers l'est.
- SP2 : Avancer d'une case uniquement si la voie est libre de tout obstacle et de tout autre robot.
- SP3 : A partir d'une position où il n'y a pas de bord à gauche du robot, contourner un obstacle isolé par la gauche.
- SP4 : Partant de n'importe quelle case de la ligne d'abscisse x=8, se rendre sur une case d'ordonnée y=16 et d'abscisse x≥8, en contournant les obstacles par la droite.
- SP5 : Partant de la position (1,1), se rendre au point (16,16) en suivant la diagonale et en contournant les obstacles par la droite.

Vous pouvez évidemment imaginer d'autres sous-programme. Attention, ces exemples **ne sont pas fournis,** c'est à vous de les écrire si vous souhaitez les utiliser.

Les figures 8 et 9 montrent des exemple de configurations pour le cas n°2.

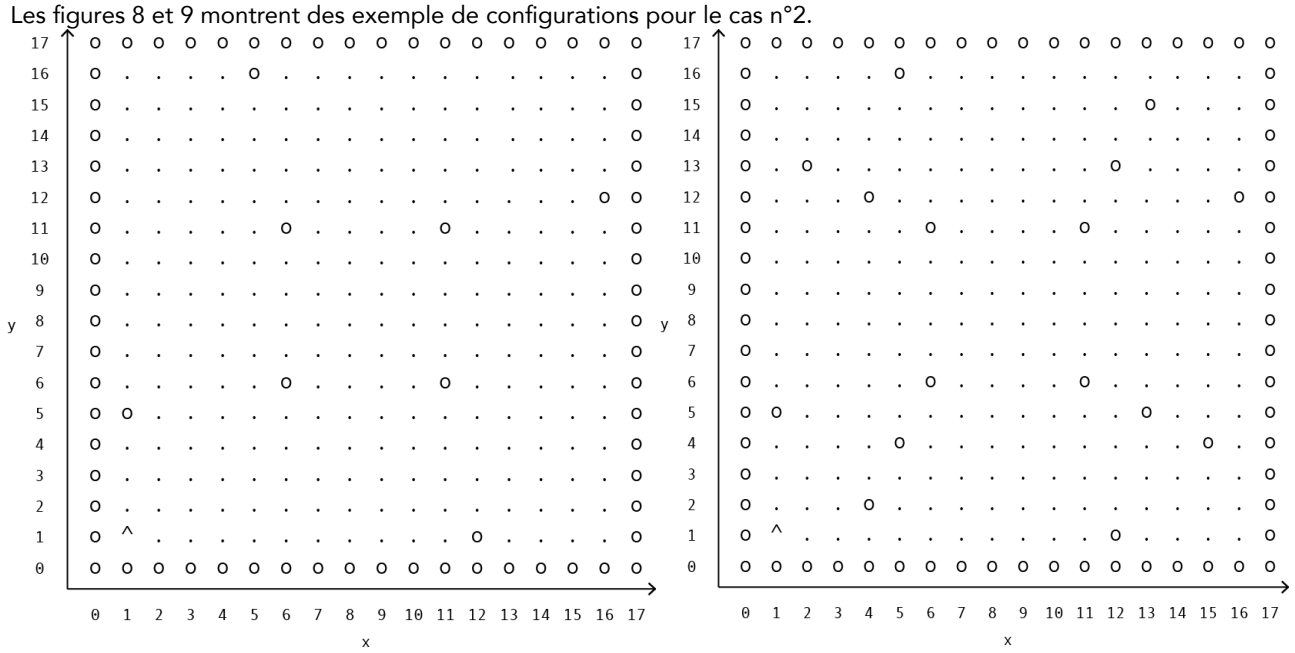


Figure 8 : Exemple de configuration simple pour le cas n°2. Figure 9 : Configuration plus complexe pour le cas n°2.

Proposez une solution globale pour le cas n°2 utilisant au moins deux sous-programmes.

Séance 2 – Faire le point sur le cas n°2 Mardi (jour 2) matin

Étape	Déroulement de l'étape
Étape 1	Présentez à votre tuteur la réponse du groupe au cas n°2. Argumenter vos choix avec des exemples en explicitant le cheminement qui vous a amené à cette solution. Organisez la présentation selon les rôles que vous aurez définis.
Étape 2	Discutez des alternatives à votre algorithme et de leur efficacité.
Étape 3	Reprendre le kit **de travail d'équipe** page 32 et identifier s'il y a des améliorations à réaliser pour mieux fonctionner (identifiez les améliorations pour votre kit).
Étape 4	Au terme de la séance tutorée, vous devrez rédiger **individuellement** un rapport R-2 sur la solution proposée pour le cas n°2, à remettre aujourd'hui (mardi en fin de matinée à 12h).

Inter-séance 2-3 – Rapport R2 et cas n°3 Mardi (jour 2) 10h-15h30

- Travail individuel (fin de matinée) : Ce mardi en fin de matinée à 12h, vous rendrez à votre tuteur votre rapport individuel R-2 et vous récupérez la description du cas n°3.
- Travail en groupe (début d'après-midi) : Préparez-vous à expliquer la solution de votre groupe au cas n°3 au tuteur cet après-midi. (Qui présente ? Quel rôle, quelle fonction ?).

Pour la prochaine séance, rédigez **individuellement** un rapport R-2 à propos de la solution que vous proposez pour résoudre la seconde variante du problème initial (cas n°2). Ce rapport, qui fera **deux pages A4 au maximum**, doit contenir les éléments suivants :

— Une description de haut niveau de la solution (description en français de la solution et des sous-programmes)

— Votre algorithme donné en pseudo-code ou en code C (chaque sous-programme doit être défini séparément)

Indiquez vos nom, prénom et numéro de groupe sur le rapport et remettez-le à l'heure. Vous pouvez utiliser le squelette de rapport présenté ci-dessous. Pour le travail de demain, **gardez une copie du rapport pour vous !**

Rapport R.2 à remettre pour le Mardi (jour 2), en fin de matinée

Squelette du rapport :

Prénom : _____ Nom : _____ N°équipe : _____

Ce rapport est a réaliser individuellement, vos propositions seront discutées en groupe Mardi (jour 2) après-midi.

Introduction

Le but de cette première phase du travail est de ...

Description générale

Ma proposition est de décomposer le problème principal en XXX sous-programme. Le premier sous- programme SP1 a pour but de ...

Algorithme

SP1 : TANT QUE Le Robot ...

SP2 : SI Le Robot ...

Test

Dans ces différentes configurations de terrain, l'algorithme se comporte de cette façon…

Conclusion

Ma proposition d'algorithme est la plus générale possible, mais il se trouve que si ... alors le chemin parcouru n'est pas le plus court possible. Le souci est que …

- ✔ Le rapport est complet : il répond de façon précise aux questions. Aucune information n'est manquante.
- ✔ Le rapport est propre et bien rédigé : le texte est formé de phrases et non d'une liste de mots-clés.

Séance 3 – Présentation cas n°3 et auto-évaluation Mardi (jour 2) 15h30-16h30

Étape	Déroulement de l'étape
Étape 1 20 minutes Expliquer	Un étudiant (quel rôle ? quelle fonction ?) explique l'algorithme que vous avez retenu pour contourner les lignes isolées (cas n°3). Comme ce matin, ces présentations seront l'occasion d'interactions avec le tuteur et au sein du groupe (quels rôles /fonctions sont mobilisés ?)
Étape 2 10 minutes Confronter	Discutez des différentes solutions apportées, tentez de généraliser et simplifier la solution.
Étape 3 15 minutes Tester	Des situations de test sont passées en revue. Leur description est l'occasion de discuter de leur portée et de leurs limites.
Étape 4 10 minutes Évaluer	Une évaluation formative (auto-évaluation) vous est proposée, celle-ci a pour but de vous aider à vérifier que vous êtes bien en train d'acquérir les compétences visées.
Étape 5 5 minutes Rédiger un rapport	Au terme de la séance vous rédigerez un rapport **de groupe** R.3 sur le cas n°3, à remettre Mercredi matin à 8h30.

Inter-séance 3-4 – Travail en autonomie Mardi (jour 2) 16h30-17h30

Pour préparer la séance de Mercredi (jour 3) matin :

Organisez-vous (quel rôle / fonction ? Qui ?) pour rédiger le rapport de groupe R.3 sur le cas n°3.

Mettre à jour le kit de travail d'équipe page 32.

Individuellement, remplir le page bilan du travail de groupe page 34 ; puis comparer vos réponses et complétez les si nécessaire.

Séance 4 – Bilan individuel et de groupe Mercredi (jour 3) 8h30-9h

Étape	Déroulement de l'étape
Étape 1 5 minutes Expliquer	Rendre à votre tuteur votre rapport d'équipe R.3.
Étape 2 10 minutes Faire un Bilan	Remplir individuellement le bilan de groupe (page 35). Comparer vos réponses. Comment expliquez-vous les éventuelles différences ?
Étape 3 10 minutes Comparer	Présenter à votre tuteur votre kit des rôles pour travailler efficacement en équipe (page 32 et votre bilan de travail en groupe (page 34). Quelle amélioration dans votre organisation vous paraît utile ?
Étape 4 5 minutes Penser	Prenez connaissance avec votre tuteur des consignes pour le poster et la présentation à préparer pour la fin de la matinée.

Inter-Séance 4 à 5 – Préparation posters et bilan Mercredi (jour 3) 9h-10h30

Préparer le bilan de la séquence d'APP-zéro. Répartir les rôles.

Productions incontournables : un poster présentant ces trois points.

- Poster présentant vos apprentissages en informatique, les **concepts** clés. L'algorithme n'est PAS à présenter.
- Bilan de l'analyse individuelle et collective pour l'efficacité des réunions et le fonctionnement de l'équipe. Aidez-vous du bilan de travail de groupe page 34.
- Votre kit de fonctions et rôles pour un travail efficace en groupe.

Séance 5 – Présentation des posters et bilan Mercredi (jour 3) 10h30-12h

Étape	Déroulement de l'étape
Étape 1 1h Faire un bilan	Pour faire le bilan de cette semaine, présenter à l'ensemble du groupe et aux tuteurs : • Les apprentissages que vous avez réalisés au cours de ce cycle d'APP, • Votre kit des rôles pour travailler efficacement en équipe, • Un bilan réflexif sur votre travail en équipe. Ce qui a bien fonctionné, ce qui a manqué. Vous recevrez un retour de la part de votre tuteur.

Séance 6 – Codage des algorithmes Mercredi (jour 3) 13h30-16h

Pour cette séance, les équipes se retrouvent en salle informatique (bat H).

Des instructions vous serons données sur place pour produire et fournir une version codée en langage C de votre algorithme. Celui-ci pourra être compilé pour vérifier qu'il ne contient pas d'erreur de syntaxe. Les fichiers doivent impérativement être envoyé avant 16h. Le programme ne sera testé qu'à l'occasion du concours de robot virtuel, de 17h à 18h.

Kit de travail d'équipe

Des rôles (ou fonctions) pour travailler efficacement en équipe

Ce document est à compléter / améliorer à mesure que vous découvrirez ou inventerez des nouvelles fonctions ou de nouveaux rôles utiles pour travailler efficacement en équipe. Vous présenterez ces rôles ou ces fonctions Mercredi (jour 3) matin (10h30-12h), à la séance de bilan.

Rôle ou fonction	
Description, Objectif	
Actions à réaliser	
• • • •	

Rôle ou fonction	
Description, Objectif	
Actions à réaliser	
• • • •	

Rôle ou fonction	
Description, Objectif	
Actions à réaliser	
• • • •	

Rôle ou fonction	
Description, Objectif	
Actions à réaliser	
• • • •	

Rôle ou fonction	
Description, Objectif	
Actions à réaliser	
• • • •	

Rôle ou fonction	
Description, Objectif	
Actions à réaliser	
• • • •	

Bilan du travail groupe

Bilan d'équipe	
1. Quels sont les rôles et fonctions que vous avez identifiés et que vous avez effectivement utilisés ?	
2. Quels sont les rôles et fonctions que vous aviez identifiés et qui n'ont finalement pas été utilisés ?	Pourquoi ?
3. Quels sont les rôles ou fonctions qui ont été les plus utiles ?	
4. Quels sont les rôles et fonctions qui vous semble avoir manqué ?	Pourquoi ?
5. Qu'est ce qui a le mieux fonctionné dans votre groupe, selon vous ?	
6. Qu'est-ce qui pourrait être amélioré dans le fonctionnement de votre groupe selon vous ?	
7. Comment avez vous fait pour vous assurer que la solution proposée à la situation-problème soit bien collective ?	
8. Comment avez vous fait pour vous assurer que la priorité a bien été donnée à la qualité d'apprentissage de chacun et non à l'efficacité de la solution proposée ?	

Bilan de groupe, à remplir individuellement.

Indiquez sur chacun des 6 axes figurant sur l'étoile ci-dessous votre niveau d'appréciation générale entre 1 et 4 :

4 signifie « Très satisfaisant »	**2** signifie « Peu satisfaisant »
3 signifie « Satisfaisant »	**1** signifie « Très insatisfaisant »

Ensuite, reliez les points.

Les axes (quelques critères d'évaluation)

- **La production du groupe** (le groupe a produit quelque chose de satisfaisant, cette production est réellement le résultat d'un effort collectif, les réunions étaient efficaces, les échanges ont permis de faire émerger des points de vue différents pour traiter le problème.
- **L'ambiance dans le groupe, le climat de travail** (l'entente entre les membres du groupe a été bonne, les participants s'aident et s'encouragent mutuellement, le groupe est arrivé à surmonter ses divergences de vue, personne n'est arrivé à imposer son point de vue…)
- **L'organisation du travail** (le groupe est parvenu à coordonner ses activités, le groupe est resté centré sur la tâche à accomplir, le groupe a fait un bon usage du tableau, un secrétaire a gardé des traces des échanges, un animateur a joué son rôle, le timing a été respecté…)
- **L'implication et l'expression de chacun des membres** (chacun des participants a contribué de manière significative à l'efficacité du groupe, le groupe a donné l'occasion à chacun de ses membres d'exprimer son point de vu, les participants en retrait ont été sollicités, tous les membres du groupe ont fait leur part de travail individuel entre les deux séances…)
- **La relation avec le tuteur** (de façon générale, le groupe a bien exploité la présence du tuteur comme une ressource pour l'aider à avancer dans son travail)
- **La relation à la situation problème** (le groupe s'est laissé prendre au jeu, il a été motivé à travailler le thème ; le groupe a trouvé que la situation problème était bien adaptée au public, qu'elle était suffisamment complexe, qu'elle était riche à exploiter).