

# II.1102 – Projet : Robot Turtles

Patrick Wang

Année 2019 – 2020

## Table des matières

<b>1</b>	<b>Informations générales</b>	<b>1</b>
<b>2</b>	<b>Présentation de Robot Turtles</b>	<b>2</b>
2.1	Description de Robot Turtles . . . . .	2
2.2	Matériel de jeu . . . . .	3
2.3	Règles du jeu . . . . .	4
2.4	Préparation du jeu . . . . .	4
2.5	Déroulement d'un tour de jeu . . . . .	4
2.5.1	Compléter le programme . . . . .	5
2.5.2	Construire un mur . . . . .	5
2.5.3	Exécuter le programme . . . . .	5
2.6	Règles de déplacement . . . . .	5
2.7	Fin du jeu . . . . .	5
2.8	Règles additionnelles et optionnelles . . . . .	6
2.8.1	Trois à la suite . . . . .	6
2.8.2	Bug! . . . . .	6
<b>3</b>	<b>Travail attendu</b>	<b>6</b>
3.1	Fonctionnalités attendues . . . . .	6
3.2	Fonctionnalités facultatives . . . . .	6
<b>4</b>	<b>Livrables</b>	<b>7</b>
<b>5</b>	<b>Optionnel : Concours d'IA</b>	<b>7</b>

## 1 Informations générales

Le module II.1102 propose aux étudiants de travailler sur un projet de développement en Java. Historiquement, le projet a toujours concerné le développement d'un jeu et cette année n'échappe pas à la règle : vous devrez programmer le jeu Robot Turtles. Plus d'informations sur ce jeu en Sections 2.

Avant de décrire ce jeu ainsi que son règlement, voici quelques consignes générales concernant la réalisation de votre projet :

1. Vous devrez former des équipes de **2 ou 3 étudiants**. Vous pouvez renseigner vos groupes dans ce fichier : [https://docs.google.com/spreadsheets/d/1ywMeBekpFozXFY\\_11pBwwudv7Shz2u1nfTcKfdDwA5Y/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1ywMeBekpFozXFY_11pBwwudv7Shz2u1nfTcKfdDwA5Y/edit?usp=sharing).

2. Faites en sorte de former des groupes d'étudiants ayant le cours II.1102 **sur le même créneau**. Aucun changement d'emploi du temps ne sera toléré lors des séances de projet.
3. Des séances de TP seront consacrées à ce projet. Ce sera l'occasion de montrer votre avancement à votre tuteur et de demander de l'aide si nécessaire.
4. Les codes sources de vos projets vont être comparés afin de détecter toute tentative de partage de code. Si le moindre soupçon de triche est présent, vous passerez en audition devant vos enseignants pour prouver que vous êtes les auteurs de vos propres projets.
5. Des bonus pourront être attribués pour des fonctionnalités originales de votre choix. Attention, veillez à implémenter entièrement les fonctionnalités de base avant de réfléchir à des fonctionnalités supplémentaires.

## 2 Présentation de Robot Turtles

### 2.1 Description de Robot Turtles



FIGURE 1 – Logo du jeu Robot Turtles.

Robot Turtles est un jeu de plateau conçu pour introduire des notions élémentaires de l'algorithmique à des jeunes enfants. Robot Turtles se joue de 2 à 4 joueurs, et chaque joueur incarne une tortue se déplaçant sur un plateau de taille  $8 \times 8$ .

L'objectif de chaque joueur est d'amener sa tortue sur un joyau placé sur le plateau en construisant un petit algorithme. Cet algorithme est construit à l'aide de cartes qui permettent de faire avancer la tortue ou de la faire tourner d'un quart de tour vers la gauche ou la droite. Des cartes supplémentaires permettent de créer des obstacles ou de les détruire.

## 2.2 Matériel de jeu

La Figure 2 illustre l'ensemble du contenu du coffret de jeu. Dans le cadre de ce projet, nous n'allons pas forcément utiliser tout ce qui est décrit en Figure 2.



FIGURE 2 – Contenu du jeu complet.

Pour ce projet, nous aurons besoin de :

- Un plateau de jeu de taille  $8 \times 8$  ;
- Quatre tuiles représentant les tortues (une bleue, une rouge, une verte, et une rose) ;
- Quatre tuiles représentant les joyaux (les couleurs sont les mêmes que pour les tortues) ;
- Quatre jeux de cartes contenant chacun 37 cartes (un jeu par couleur). Ces jeux de cartes contiennent quatre types de cartes :
  - Les cartes bleues ( $\times 18$ ) font avancer la tortue d'une case. C'est la seule carte qui peut faire se déplacer une tortue.
  - Les cartes jaunes ( $\times 8$ ) font tourner la tortue de  $90^\circ$  dans le sens anti-horaire. Attention, la tortue reste sur la même case.
  - Les cartes violettes ( $\times 8$ ) font tourner la tortue de  $90^\circ$  dans le sens horaire. Attention, la tortue reste sur la même case.
  - Les cartes "Laser" ( $\times 3$ ) permettent aux tortues d'utiliser un laser pour attaquer tout ce qui se trouve devant elles.
- 36 tuiles représentant des obstacles qui peuvent être placées sur le plateau. Ces obstacles sont de 3 types :
  - Les murs en pierre ( $\times 20$ ) sont des obstacles inamovibles et indestructibles ;
  - Les murs en glace ( $\times 12$ ) sont des obstacles inamovibles mais qui peuvent être détruits par des lasers ;

- Les caisses en bois ( $\times 4$ ) sont des obstacles amovibles et indestructibles. Pour les déplacer, une tortue doit simplement pousser la caisse. Si un autre obstacle quelconque se trouve derrière une caisse, une tortue ne pourra pas la pousser.

### 2.3 Règles du jeu

Pour ce projet, nous allons jouer à Robot Turtles en utilisant les règles “Galapagos”. L’objectif final est simple : atteindre un joyau avant les autres pour gagner un maximum de points ! Attention, un même joyau peut être atteint par plusieurs joueurs.

### 2.4 Préparation du jeu

Selon que l’on joue à 2, 3, ou 4 joueurs, le plateau doit être disposé comme illustré en Figure 3.



FIGURE 3 – Initialisation du jeu pour les règles “Galapagos”.

Chaque joueur doit récupérer sa tortue, 3 murs de pierre, 2 murs de glace, ainsi que le jeu de cartes correspondant. Bien que la Section 2.2 mentionne des “caisses en bois”, nous n’allons pas les utiliser ici.

Les tuiles représentant les murs sont mises de côté tandis que le jeu de cartes est mélangé puis placé face caché devant chaque joueur. Au début de la partie, chaque joueur pioche 5 cartes.

### 2.5 Déroulement d’un tour de jeu

On va choisir au hasard le joueur qui va commencer la partie. À chaque tour, un joueur a le choix entre trois options :

- Compléter le programme ;
- Construire un mur ;
- Exécuter le programme.

À la fin de son tour, et quel que soit l’option choisie, le joueur peut défausser sa main et piocher des cartes jusqu’à en avoir à nouveau 5.

### 2.5.1 Compléter le programme

Chaque joueur doit construire son programme afin de déplacer sa tortue sur un des joyaux présents sur le plateau. Un programme est construit en plaçant des cartes les unes à la suite des autres, afin de créer une “file d’instructions”. Attention, ces cartes doivent être placées face cachées. Il est donc important que le joueur mémorise son programme en même temps qu’il le construit.

À chaque tour, le joueur peut donc ajouter à la fin de son programme 1–5 cartes afin de le compléter.

### 2.5.2 Construire un mur

Le joueur peut placer un de ses murs (en pierre ou en glace) sur le plateau. Attention, il est **interdit de bloquer l'accès aux joyaux**. En conséquence, il est aussi interdit d’encercler un joyau ou un autre joueur.

### 2.5.3 Exécuter le programme

Le joueur peut décider d’exécuter le programme qu’il a construit. La tortue du joueur va donc lire la “file d’instructions” et les effectuer les unes à la suite des autres. Lorsque l’exécution du programme est terminée, les cartes constituant le programme sont ajoutées à la pile de défausse.

## 2.6 Règles de déplacement

Le jeu de cartes est constitué de quatre types de cartes :

- Les cartes bleues font avancer la tortue d’une case ;
- Les cartes jaunes font tourner la tortue de 90° dans le sens anti-horaire ;
- Les cartes violettes font tourner la tortue de 90° dans le sens horaire ;
- Les cartes “Laser” touchent la première tuile se trouvant en face de la tortue. Si la tuile touchée est un mur de glace, celui-ci va fondre et disparaître.

Des règles complémentaires entrent en jeu :

- Si une tortue se heurte à un mur, elle va faire demi-tour. Le programme continue ensuite son exécution.
- Si une tortue se heurte à une autre tortue, les **deux tortues** retournent à leurs positions de départ, et le programme continue son exécution.
- Si une tortue est touchée par un laser, il y a deux cas possibles :
  - S'il n'y a que deux joueurs, la tortue touchée fait un demi-tour ;
  - S'il y a plus de deux joueurs, la tortue touchée retourne à sa position de départ.
- Si le laser touche un joyau, le laser est réfléchi et se retourne contre la tortue. Celle-ci fait donc un demi-tour (s'il n'y a que deux joueurs) ou retourne à sa position de départ (s'il y a plus de deux joueurs). Puis, le programme continue son exécution.

## 2.7 Fin du jeu

L’objectif étant d’atteindre un joyau, le jeu prend fin lorsqu’il ne reste plus qu’un joueur.

## 2.8 Règles additionnelles et optionnelles

Dans cette section, nous allons présenter un certain nombre de règles additionnelles. Ces règles pourront être implémentées dans vos projets mais constituent des fonctionnalités facultatives.

### 2.8.1 Trois à la suite

Dans ce mode de jeu, on va attribuer des points aux joueurs ayant atteints le joyau. Par exemple :

- S'il y a quatre joueurs, le premier à atteindre un joyau gagne 3 points, le second 2 points, le troisième 1 point, et le dernier ne gagne aucun point.
- S'il y a trois joueurs, le premier à atteindre un joyau gagne 2 points, le second 1 point, et le dernier ne gagne aucun point.
- S'il y a deux joueurs, le premier à atteindre un joyau gagne 1 point et le dernier ne gagne aucun point.

Jouer 3 manches de suite. À la fin des trois manches, le joueur qui a cumulé le plus de points lors des 3 manches remporte la partie.

### 2.8.2 Bug !

Chaque joueur se voit ajouter une carte “Bug” à leur jeu. Cela offre aux joueurs une option supplémentaire à chaque tour : Ajouter la carte “Bug” au programme d'un joueur adverse.

La carte “Bug” a pour effet d'inverser l'exécution d'un programme. Lorsqu'un joueur affecté par la carte “Bug” continue de construire son programme, les nouvelles cartes seront ainsi ajoutées à la fin de la “file d'instructions” inversée. Lorsqu'un joueur affecté par la carte “Bug” exécute son programme, cette carte “Bug” est défaussée jusqu'à la fin de la partie.

## 3 Travail attendu

Pour ce projet, votre travail consistera à développer le jeu Robot Turtles en Java. Ce jeu devra être jouable pour 2 à 4 joueurs sur une interface graphique que vous allez aussi concevoir.

### 3.1 Fonctionnalités attendues

- Développement de l'interface graphique rendant le jeu jouable pour 2 à 4 joueurs humains en tour par tour.
- Implémentation des fonctionnalités de bases décrites en Section 2 : initialisation du jeu, tours de jeu, identification de la fin d'une partie et calcul des scores.
- Rédaction d'un document technique reprenant la conception de votre programme, les différents modèles de données utilisés, et un manuel d'utilisation de votre projet.

### 3.2 Fonctionnalités facultatives

- Implémentation des règles additionnelles décrites en Section 2.8.

- Implémentation d'une IA capable de jouer selon les règles du jeu initiales (les règles additionnelles ne seront donc pas prises en compte par l'IA).

## 4 Livrables

Plusieurs livrables seront à déposer sur Moodle durant le semestre :

- Un premier concerne la modélisation UML de votre projet. Il est important de bien modéliser votre projet pour ensuite concevoir un programme plus robuste et plus facile à modifier si nécessaire.
- En fin de semestre, il faudra rendre un document technique décrivant l'implémentation de votre projet ainsi que le code source du projet.
- Une soutenance sera planifiée durant le semestre durant laquelle vous devrez expliciter la conception de votre projet et en faire une démonstration.

Plus d'informations vous seront communiquées au fur et à mesure sur Moodle.

## 5 Optionnel : Concours d'IA

Le but sera de faire combattre votre IA contre celle de vos camarades et de diffuser en direct les matchs en salle L.012.

S'il y a suffisamment de participants, ce concours d'IA pourra se dérouler une ou deux semaines après la semaine de soutenance. Vous aurez ainsi un peu de temps supplémentaire pour peaufiner votre IA. Le résultat de ce concours n'aura donc pas d'impact particulier sur votre note de projet.

Pour y participer, il faudra faire part de votre motivation en envoyant un mail à Patrick Wang (patrick.wang@isep.fr). Nous vous fournirons ensuite une classe Java permettant à votre programme de communiquer avec le serveur de jeu. Plus d'informations à venir...

Si vous avez des questions sur ce concours d'IA, n'hésitez pas à contacter...