

Mange-moi si tu peux !

Introduction

Manger est une activité essentielle dans le monde animal. L'absorption de nourriture va permettre d'avoir de l'énergie qui sera utile pour le fonctionnement musculaire et pourra également contribuer à la prise de poids.

Prenons l'exemple d'un mouton, l'herbe naturelle lui convient très bien. Néanmoins, il s'accommodera de chardons qui lui provoquent quelques désagréments. Bien évidemment il préférera la marguerite qui lui donne plus d'énergie !

Chaque espèce animale mange, mais pas la même chose : chacun son régime alimentaire ! Ainsi, le loup ne mange pas de végétaux point d'herbe ou de fleur à son menu ! En revanche, le loup raffole du mouton. Dès qu'il en voit un, il ne peut s'empêcher de le chasser. Chaque jour son but est d'attraper et dévorer un mouton.

L'observation sur le terrain nous a permis de conclure qu'après avoir ingéré du cactus le mouton avance deux fois moins vite qu'après avoir ingéré de l'herbe classique, par contre après avoir ingéré une fleur de marguerite il avance deux fois plus vite.

Le mouton est un animal paisible qui aime consacrer sa journée à paître : il avance aléatoirement puis ingurgite le végétal (quel qu'il soit) où il se trouve, puis change de place. Par contre, quand il détecte un prédateur son objectif est de s'éloigner le plus vite. Néanmoins, après chaque déplacement, il ne peut s'empêcher de manger. Dès qu'il quitte la parcelle sur laquelle il était le végétal brouté repousse immédiatement.

Le loup est un prédateur en perpétuelle quête d'une proie. Nos observations nous ont permis de comprendre que le loup se déplace aléatoirement, puis quand il détecte une proie il se dirige vers elle pour la manger.

Ni le loup ni le mouton ne sont capables de passer au-dessus d'un rocher.

Pour faire avancer les connaissances scientifiques dans le domaine et voir l'impact de la réintroduction du loup dans certaines régions, nous aimerions faire une étude pour comprendre l'efficacité des stratégies des moutons et des loups dans certains environnements contraints.

Développement d'un simulateur

Pour éviter la mort atroce de cactus, de marguerites et bien sûr de moutons. Nous vous demandons de développer un simulateur qui permettra de faire évoluer un loup et un mouton dans un environnement s'apparentant à un labyrinthe.

La Figure 1 donne un exemple de visualisation du labyrinthe.

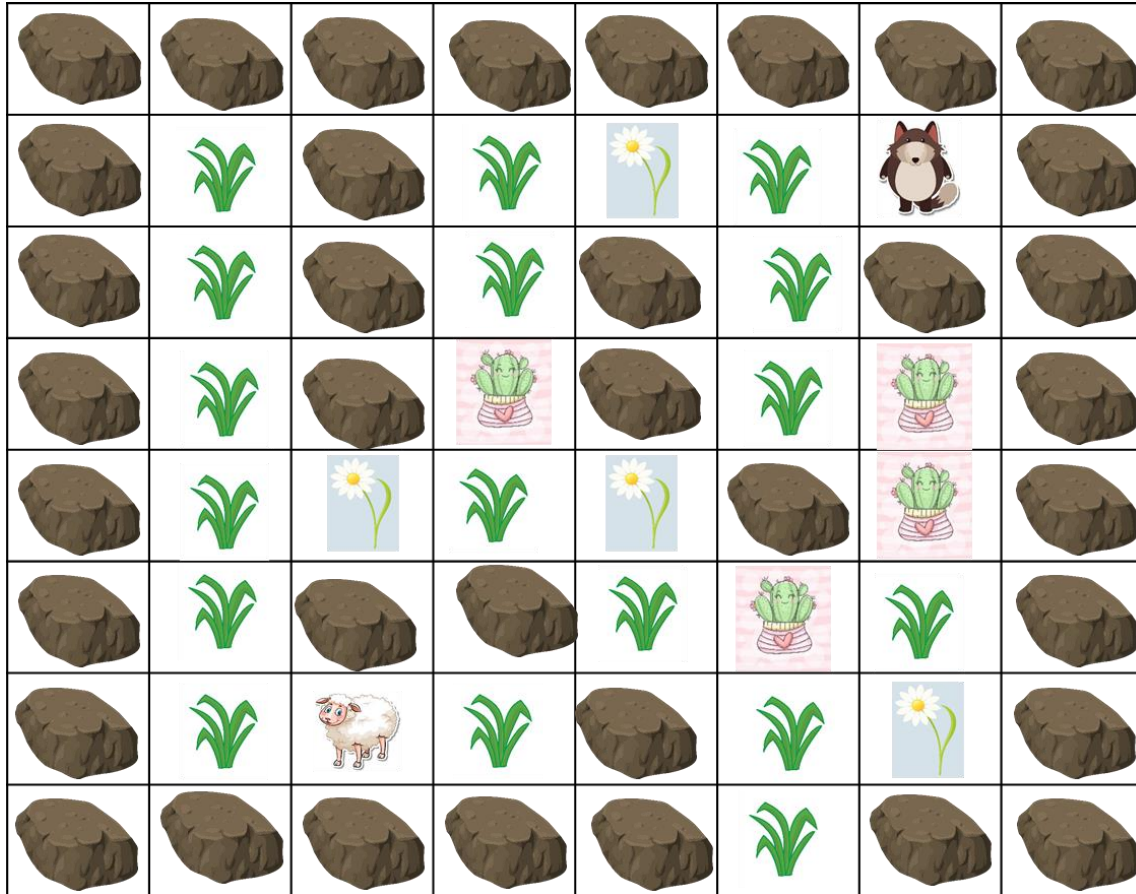


Figure 1 vue schématique du labyrinthe avec les différents acteurs.

Le labyrinthe est délimité par des rochers seule la case de sortie n'est pas occupée par un rocher.

Il ne peut y avoir qu'un seul type de végétal sur une case, il ne peut pas y avoir de végétaux sur un rocher.

Par défaut, le loup avance de 3 cases par tour et le mouton de 2 cases.

Quand il a ingurgité une marguerite le mouton avance de 4 cases et quand il a ingurgité un cactus il n'avance que d'1 case.

La vision d'un animal est de 5 cases (au sens distance de Manhattan).

Le loup ne peut pas aller sur la case de sortie du labyrinthe.

Scénario de la simulation :

- Tant que le mouton n'a pas vu le loup, son objectif est de paître (son objectif est donc d'ingurgiter un maximum de végétaux).
- Quand il voit le loup, il bascule en mode fuite : son objectif est alors de sortir du labyrinthe (atteindre la case de sortie) et bien sûr d'éviter de se faire dévorer par le loup.
- Tant que le loup n'a pas vu le mouton il erre dans le labyrinthe
- Quand le loup voit le mouton il bascule en mode chasse en essayant d'arriver sur la case du mouton.
- Quand le mouton, à la fin d'un déplacement, arrive sur une case il mange le végétal qui s'y trouve, s'il s'agit :
 - o D'herbe : il se déplacera de deux cases au prochain tour,
 - o D'une marguerite : il se déplacera de quatre cases au prochain tour
 - o D'un cactus : il se déplacera que d'une case au prochain tour.

Préparation de la simulation :

L'expérimentateur avant de lancer le scénario va spécifier les données :

- Le nombre de lignes et de colonnes du labyrinthe. Une fois les dimensions fixées, celui-ci sera fermé (rochers sur toutes les cases bordant le labyrinthe), et à l'intérieur des cases contenant de l'herbe seront positionnées. L'application affichera le labyrinthe.
- L'utilisateur devra alors choisir la case de sortie en remplaçant un des rochers des bords par de l'herbe.
- Il pourra ensuite remplacer les contenus des cases du labyrinthe par d'autres éléments :
 - o Rocher
 - o Herbe
 - o Marguerite
 - o Cactus
- L'expérimentateur pourra alors verrouiller la carte du labyrinthe, il devra alors indiquer (voire désigner) la case (qui ne peut pas être un rocher) où commence le loup et celle où commence le mouton. Le système doit vérifier qu'il n'y a pas d'anomalie, i.e. la sortie doit être atteignable de tous les points intérieurs.
- Enfin, l'expérimentateur pourra lancer la simulation avec un des trois algorithmes proposés choisi par animal (Mouton/Loup). La simulation fera une temporisation lors de l'affichage afin que chaque déplacement soit affiché et visible. La simulation s'arrête quand le mouton atteint la sortie ou quand le loup mange le mouton. A l'issue de la simulation les données sont affichées :
 - o Le vainqueur : le loup ou le mouton
 - o Le nombre de tours
 - o Le nombre, pour chaque type de végétaux, ingurgité par le mouton

Pour faciliter l'utilisation de l'application, les données de la carte d'un labyrinthe pourront être sauvegardées dans un fichier et chargées si besoin.

Livrables

- 1) Diagramme de classes
 - Rendu 20/04 à 11h
 - Rendu à la fin de la séance TP avec Mme Oliveira prise en compte des modifications suggérées
- 2) Analyse des tâches (si nécessaire) et Maquettage
 - Rendu le 16/05
- 3) Code de l'application sur compilatio (outil de détection de plagiat)
 - Java, JUnit et JavaFX
 - Affichage du labyrinthe, configuration et déplacements
 - Rendu le 28/05
- 4) Code complet incluant les algorithmes de résolution de problème (les 3 !)
N'oubliez pas les Jeux d'essais prouvant le respect des spécifications et les Jeux de test
Rendu la veille de la démonstration à :23 :59 (12/06, 13/06 ou 14/06 f(date de TP avec M. Bay)).