Intelligence Artificielle

Machine learning

Age : 10-14 ans

Le jeu du labyrinthe pour comprendre le Q-learning

Objectifs :

- ✓ Faire découvrir quelques caractéristiques et principes essentiels de l'apprentissage par renforcement s'appuyant sur l'algorithme du Q-learning à partir d'un jeu
- ✓ Faire émerger les principes fondamentaux de l'apprentissage par renforcement avec l'algorithme du Q-learning

Notions abordées : Q-Learning, Apprentissage par renforcement, sanction, récompense, essais/erreurs

- **Phase 1 :** le jeu du labyrinthe
- **Phase 2 :** Cipy, le robot dans un labyrinthe

Durée : 2h (phase 1) + 1h30-2h (phase 2)

Dispositif pédagogique : par groupe de 2 à 4 (phase 1) et en mode conférence (phase 2)

Matériel

- 1 jeu par équipe comprenant : le plateau, 48 caches carrés amovibles numérotés, 1 pion, 1 dé à 4 faces, 1 dés à 6 faces
- Papeterie (feuilles, post-it, stylos, feutres, …)
- 1 ordinateur (animatrice.eur)
- 1 vidéo projecteur (animatrice.eur)
- 1 présentation pdf « 6.2.1 – Cipy, le robot dans un labyrinthe »
- 1 fichier Scratch « 6.2.2 - Q-learning en Scratch » (Bonus)

Annexes

- Annexe 1 : Plateau de jeu
- Annexe 2 : Consignes du jeu du labyrinthe
- Annexe 3 : Q-learning en Scratch

Références & liens utiles

- **Cours introduction à l'IA – Université Berkeley :**

  http://ai.berkeley.edu/home.html

  http://inst.eecs.berkeley.edu/~cs188/fa19/projects/#projects-overview
- **Towards data science – Percy Jaiswal :**

  https://towardsdatascience.com/getting-started-with-reinforcement-q-learning-77499b1766b6
- **Site ActuIA, 'Le portail de l'intelligence artificielle et des startups IA'**

  https://www.actuia.com/contribution/thibault-neveu/lapprentissage-par-renforcement/
- **Data Science Central - William Vorhies – 13/09/2016**

  https://www.datasciencecentral.com/profiles/blogs/reinforcement-learning-and-ai
- **Wikipedia – Apprentissage par renforcement**

  https://www.wikiwand.com/fr/Apprentissage_par_renforcement
- **Baptiste Saintot – 20/02/2019 - L'apprentissage par renforcement démystifié**

  https://blog.octo.com/lapprentissage-par-renforcement-demystifie/
- **Video Aibo le chien robot** : https://www.youtube.com/watch?v=aN9jjw1zLSY

Droits d'auteur

Description détaillée

Phase 1 : le jeu du labyrinthe

- *But du jeu*

Le but du jeu consiste à trouver une coupe qui se cache dans un labyrinthe en évitant les obstacles comme le ferait une IA.

- *Confection du plateau de jeu*

Il est possible de réaliser le jeu (plateau + caches carrés numérotés de 1 à 48) par impression sur papier ou carton.

Une autre option est d'utiliser une graveuse laser sur bois (FabLab) et de la peinture pour les couleurs.

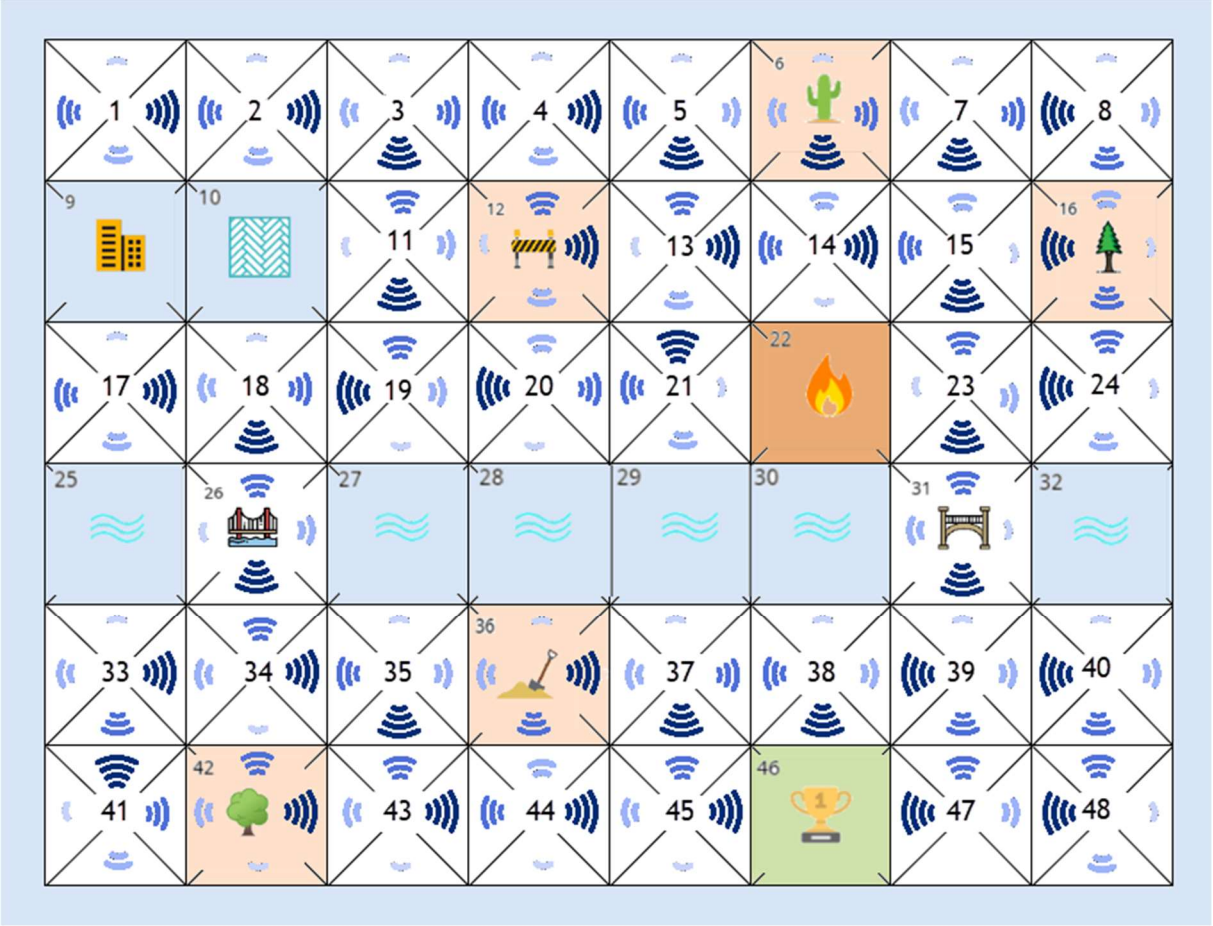
Le modèle se trouve en annexe 1.

- *Explications quant aux cases du labyrinthe qui seront progressivement dévoilées*

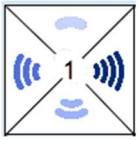
Les cases en bleu ainsi que l'extérieur du labyrinthe (liseré bleu) sont infranchissables ou inaccessibles.

Les cases roses représentent des éléments qui ralentissent la progression de l'IA.

Les cases de couleur verte (coupe) et rouge foncé (incendie) sont des cases qui entraînent le retour à la case départ.



- *Déroulement du jeu*
    - Au début du jeu, chaque case est recouverte par un cache carré portant le numéro de la case
    - Le pion est positionné sur la case de départ N° 1 (coin supérieur gauche) qui est dévoilée.
    - Le pion est déplacé d'une case à la fois dans la direction déterminée.
    - Au cours des 20 premiers coups, la direction à suivre est déterminée au hasard, de manière aléatoire.
    - À partir du 21ème coup, la direction du déplacement est déterminée soit au hasard, de manière aléatoire soit en suivant les indications des arcs de cercle de couleur bleue.

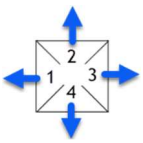


C'est le symbole  qui indique la direction qui doit être suivie lorsque l'on ne joue pas au hasard (4 arcs de cercles bleu foncé – l'orientation du symbole montre la direction à suivre – Ici vers la droite)

- Les cases du labyrinthe sont progressivement dévoilées en ôtant le cache numéroté. Le jeu se termine lorsque la coupe est atteinte.
- Le nombre de coups joués est noté par la personne désignée à cet effet.
- Un élément d'émulation peut être introduit si nécessaire : la première équipe qui arrive à la coupe a gagné.

- *Déroulement d'un coup*

  - Pendant les 20 premiers coups, on joue avec le dé à 4 faces pour déterminer une direction au hasard

    Résultat du jet du dé à 4 faces 

✓ 1 = le pion est déplacé vers la gauche
✓ 2 = le pion est déplacé vers le haut
✓ 3 = le pion est déplacé vers la droite
✓ 4 = le pion est déplacé vers le bas

Si par ce déplacement, on arrive:

✓ sur une case non encore découverte, celle-ci est dévoilée en ôtant le cache ;
✓ en dehors de la grille ou sur une case bleue, le pion reste sur la position initiale ;
✓ sur la case orange foncé: le pion revient à la case de départ N° 1 ;
✓ sur la case verte (coupe) : le pion revient à la case de départ et le jeu est terminé.

  - Après les 20 premiers coups, on joue au hasard ou en suivant les indications données par les 4 arcs de cercle de couleur bleue

Jet du dé à 6 faces pour déterminer si on joue au hasard :

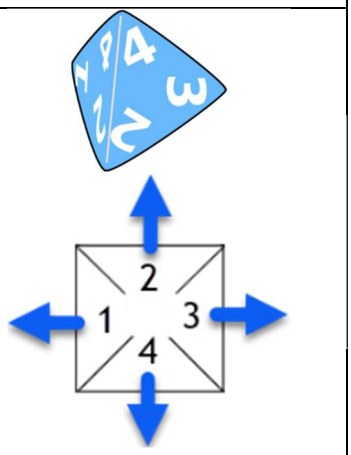
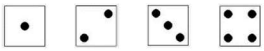
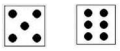
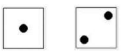
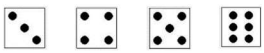
✓ Pour les coups 21 à 40 : on joue au hasard si on obtient 1, 2, 3 ou 4 ou selon la direction indiquée par les 4 arcs de cercle de couleur bleue si on obtient 5 ou 6
✓ Pour les coups 41 à 60 : on joue au hasard si on obtient 1, 2 ou 3 ou selon la direction indiquée par les 4 arcs de cercle de couleur bleue si on obtient 4, 5 ou 6
✓ À partir du 61ème coup : on joue au hasard si on obtient 1 ou 2 ou selon la direction indiquée par les 4 arcs de cercle de couleur bleue si on obtient 3, 4, 5 ou 6

- Si on ne joue pas au hasard, le pion est donc déplacé dans la direction indiquée par les 4 arcs de cercles bleu foncé de la case sur laquelle on se trouve.

Exemple : Ici, c'est la direction vers la droite qui doit être suivie

- Si on joue au hasard, le dé à 4 faces est jeté pour déterminer la direction à suivre.

Il est conseillé d'imprimer le tableau récapitulatif des règles ci-dessous (cf annexe 2) pour chaque groupe de joueurs :

Coup	Jet du dé à 6 faces **pour déterminer la manière de jouer**		Jet du dé à 4 faces **pour déterminer la direction au hasard**
	On joue au hasard si le résultat est	On joue en suivant la direction indiquée par le symbole  si le résultat est	On suit la direction correspondant au résultat du jet de dé
1 à 20	-	-	
21 à 40			
41 à 60			
61 et plus			

- *Collaboration*

Il s'agit d'un jeu collaboratif qui peut se jouer par équipe de 4 :

- 1 personne détermine la direction à suivre au hasard avec le dé à 4 faces
- 1 personne tient le compte du nombre de coups joués et, à partir du 21ème coup, détermine la manière de jouer le coup avec le dé à 6 faces
- 1 personne ôte les carrés de la grille
- 1 personne déplace le pion sur la grille

Il est possible d'alterner les rôles au cours d'une même partie. Deux rôles seront attribués par personne dans le cas d'équipes de 2.

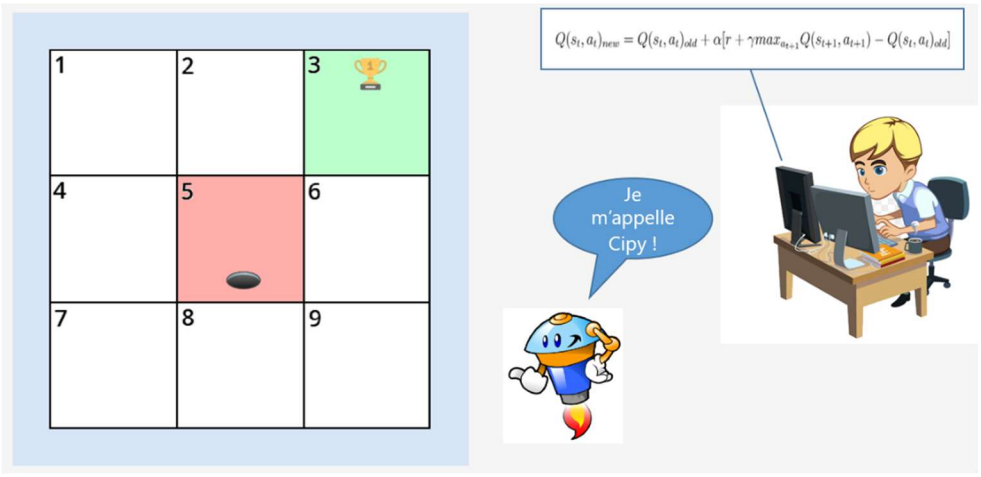
Il est possible que, par manque de temps, des équipes n'arrivent pas au bout du jeu. Ce n'est pas grave. Le but poursuivi est surtout de faire percevoir, de faire sentir le mécanisme général d'apprentissage utilisé par une IA.

Phase 2 : Cipy, le robot dans un labyrinthe

Pour cette activité, l'animatrice-eur va utiliser la présentation pdf « 6.2.1 – Cipy, le robot dans un labyrinthe » comme support pour poser des questions et faire participer les jeunes.

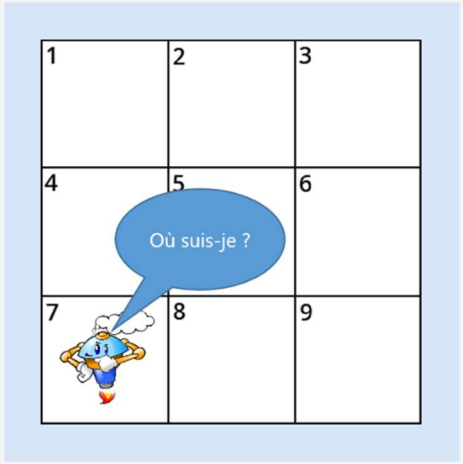
- Slide 2

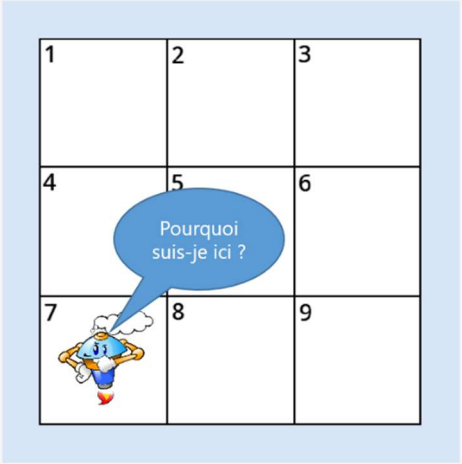
Il était une fois Cipy, un labyrinthe et un algorithme …



- Slides 3 et 4

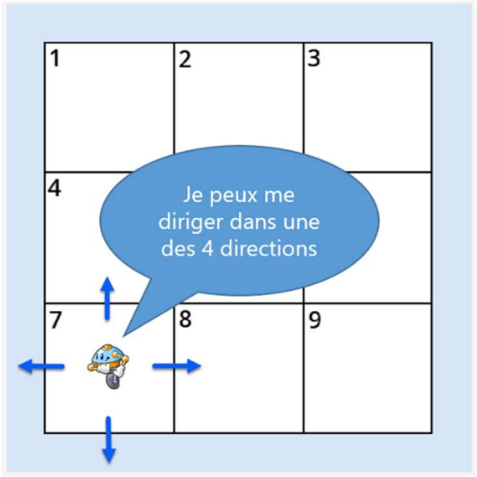
Cipy n'a aucune connaissance du labyrinthe … et ne sait pas ce qu'il fait là.





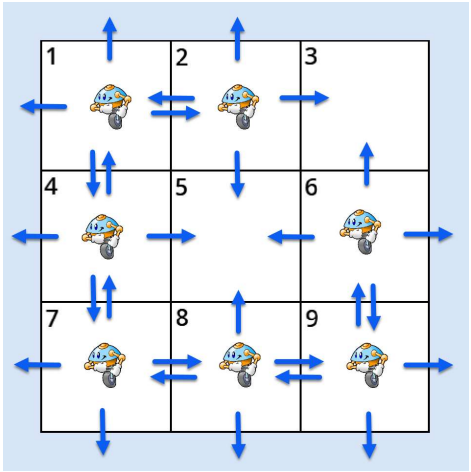
- Slide 5

Il « sait » seulement qu'il peut se déplacer dans une des 4 directions permises.



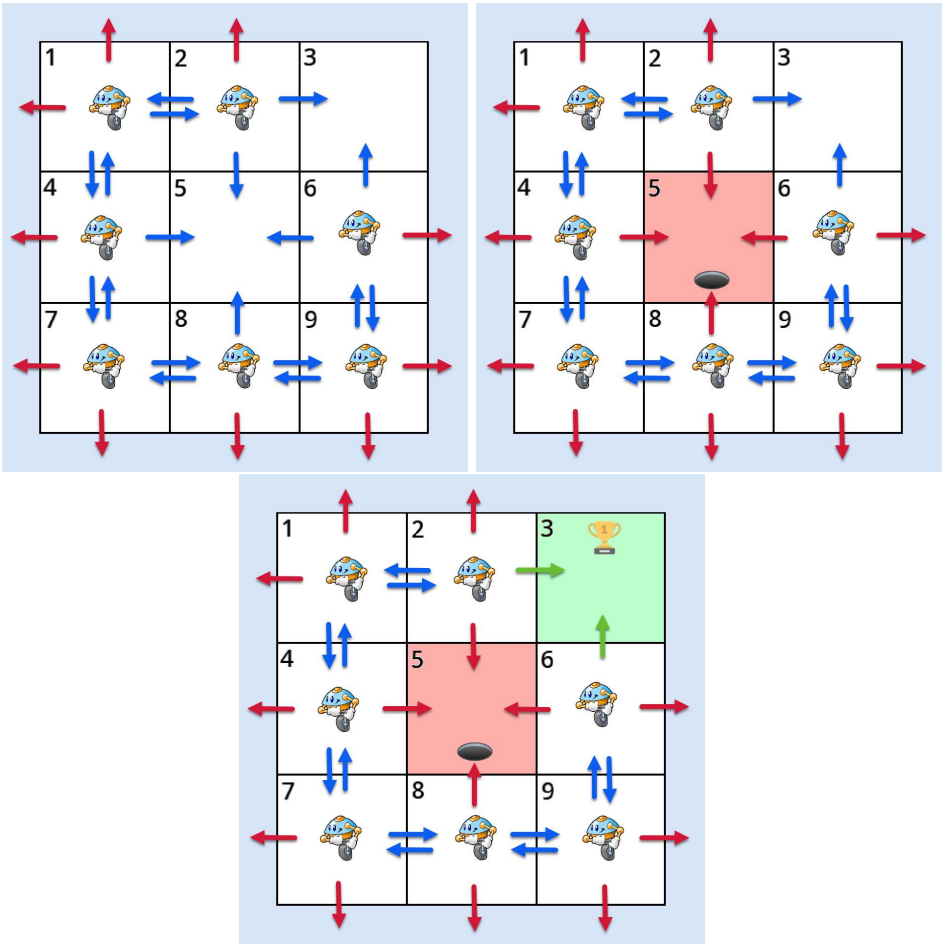
- Slide 6

Cipy se déplace dans toutes les directions possibles.



- Slide 7

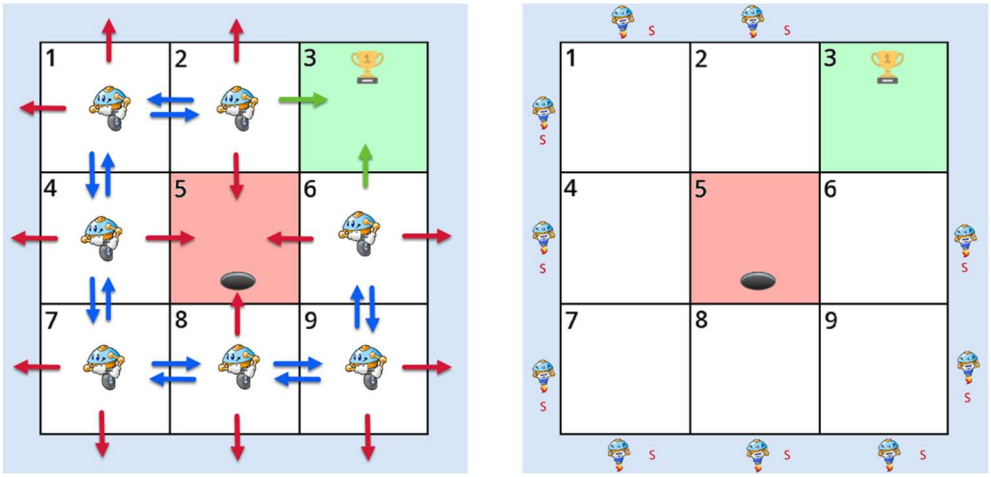
Il commet parfois des erreurs (sortir de la grille, tomber dans le trou) ou atteint parfois la coupe.



Question : Comment peut-il « comprendre » qu'il a commis une erreur ou qu'il a fait bonne action ?

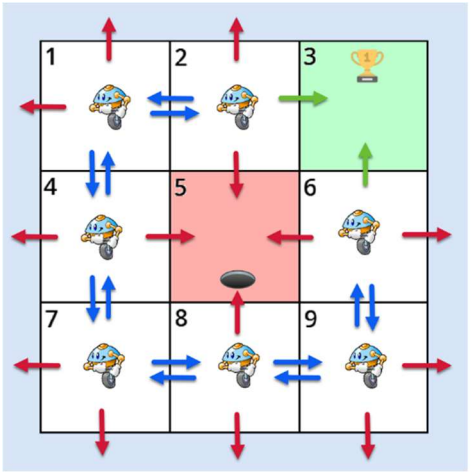
- Slide 11

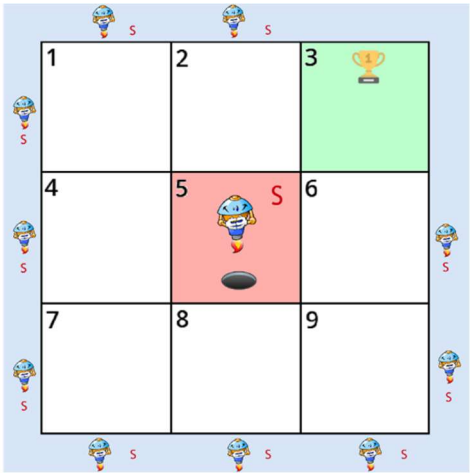
Quand Cipy commet une petite erreur (essaye de sortir de la grille), il reçoit une petite sanction et reste sur place.



- Slide 12

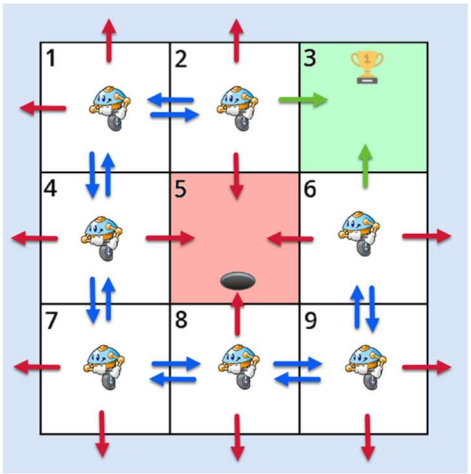
Quand Cipy commet une grosse erreur (tombe dans le trou), il reçoit une grosse sanction et revient à la case départ.

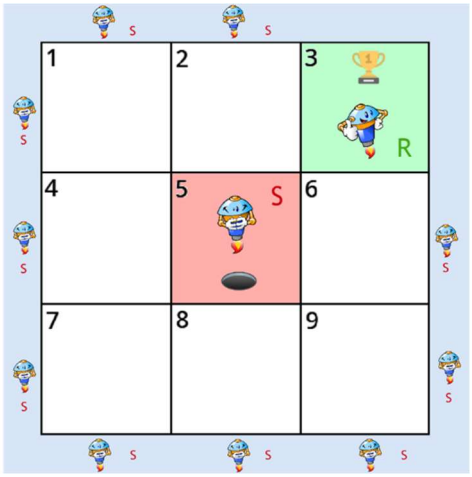




- Slide 13

Quand Cipy fait une bonne action (trouve la coupe), il est récompensé et revient à la case départ.





Pour le reste, Cipy se déplace simplement de case blanche en case blanche.

On peut aussi représenter les différentes attitudes de Cipy confronté à ces événements comme suit :

Cipy explore

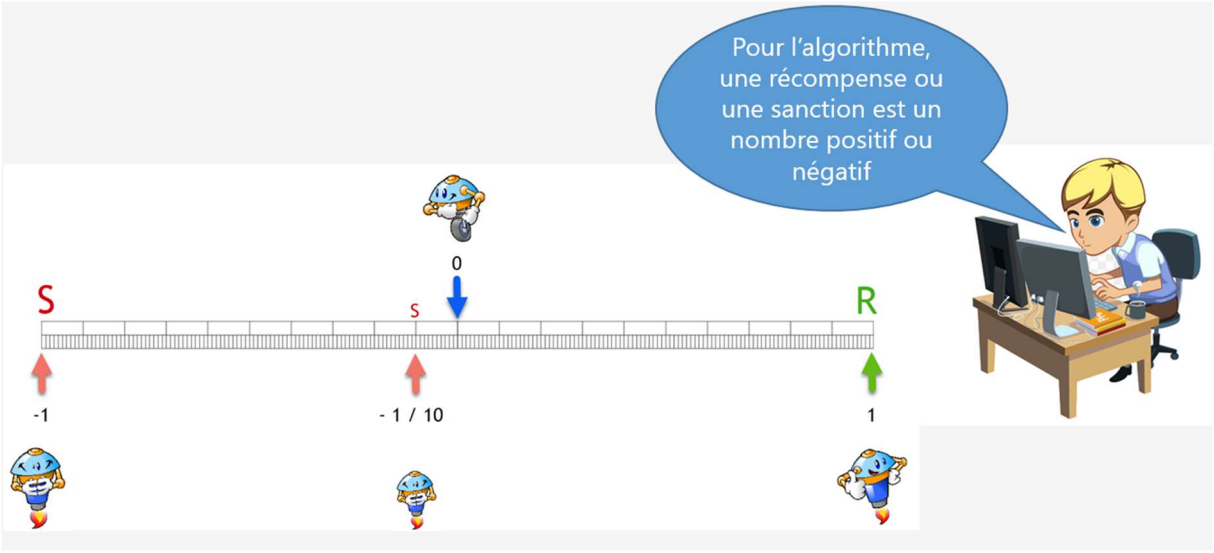
 Cipy a commis une petit erreur  Cipy a commis une grosse erreur

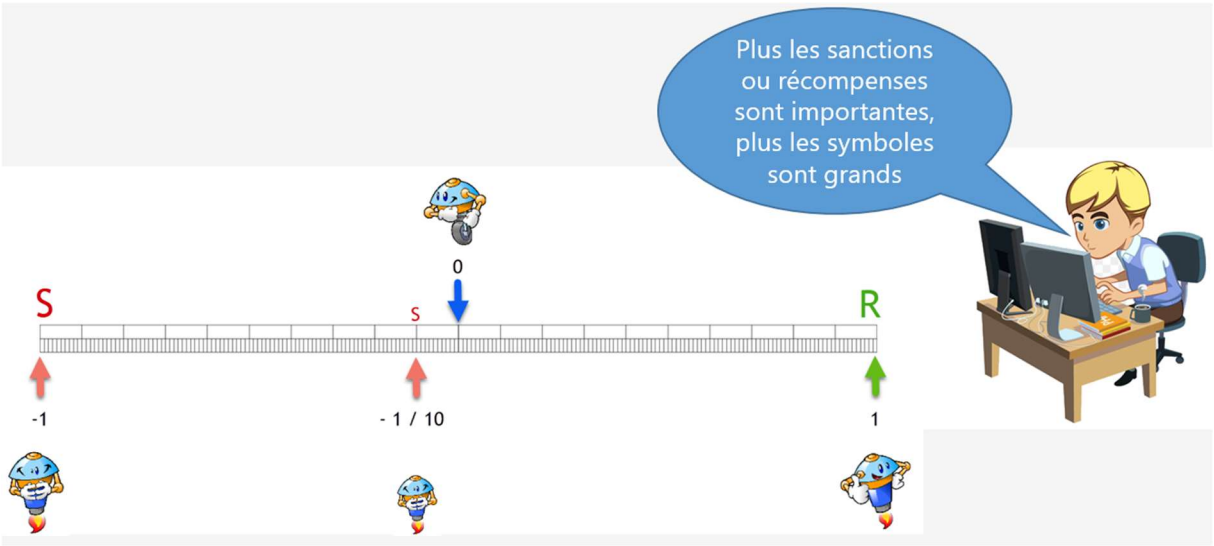
 Cipy est content

- Slide 14 - 16

Question : Au fait, ces récompenses et sanctions, qu'est-ce que c'est ?

Une sanction est un nombre négatif, une récompense un nombre positif ! En valeur absolue, plus la sanction (ou la récompense), est importante, plus les symboles utilisés sont grands.





- Slide 17

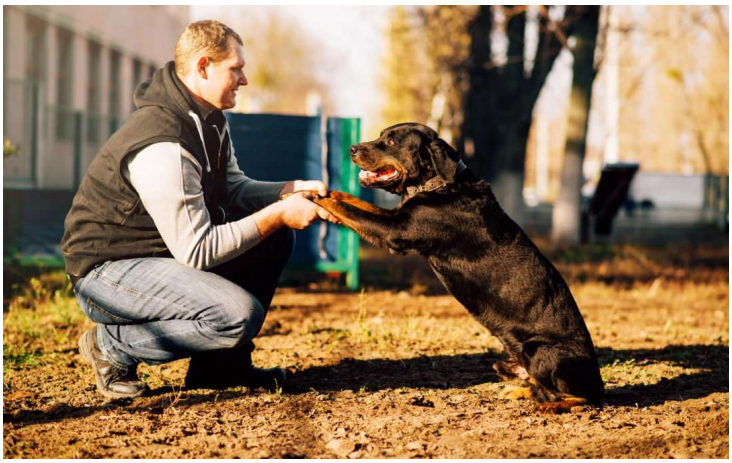
Question : Utiliser des sanctions et récompenses pour apprendre … D'autres exemples ?

- Slide 18

Dans le jeu de Nimm, la « machine » apprend de ses essais et erreurs. Si la « machine » perd ou gagne, elle en « déduit » que son coup est fautif et la solution est retirée (sanction). C'est le même mécanisme ici. Cipy apprend également de ses essais et erreurs mais dans la situation plus compliquée d'un labyrinthe où il se dirige dans 4 directions possibles.

- Slide 19

Un parallèle peut être également fait avec le dressage d'un chien. Son maître lui apprend un comportement spécifique. S'il exécute bien l'ordre de son maître, le chien reçoit une récompense (caresse, etc.). S'il n'exécute pas bien les ordres de son maître, il ne reçoit rien.



- Slide 20

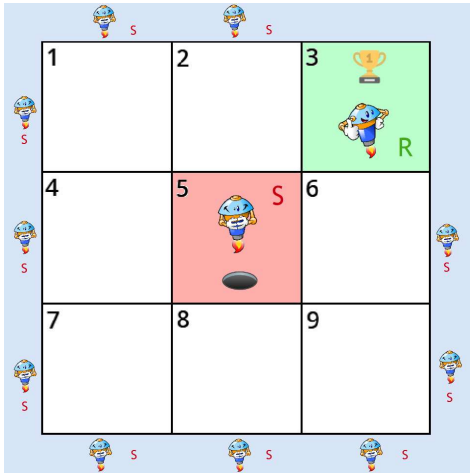
Un chien robotisé qui apprend à marcher peut utiliser le même mécanisme d'apprentissage par renforcement. Au début, il va essayer des mouvements aléatoires anarchiques et va tomber. Mais très rapidement, il va apprendre de ses erreurs et à marcher correctement en synchronisant ses actions. Chaque fois qu'il va commettre une erreur, il va être sanctionné et chaque fois qu'il fera un mouvement correct, il sera récompensé.

Cette technique permet d'obtenir des chiens-robot qui savent mieux marcher que ceux qui ont été spécifiquement programmés. Celui qui est pré-programmé ne peut pas s'améliorer alors que celui qui apprend par renforcement va constamment s'améliorer et s'adapter.

Aibo de Sony : https://www.youtube.com/watch?v=aN9jjw1zLSY

- Slide 21

Question : Comment Cipy va-t-il se rappeler du fait qu'il a reçu telle récompense ou telle sanction ?



- Slide 22 - 23

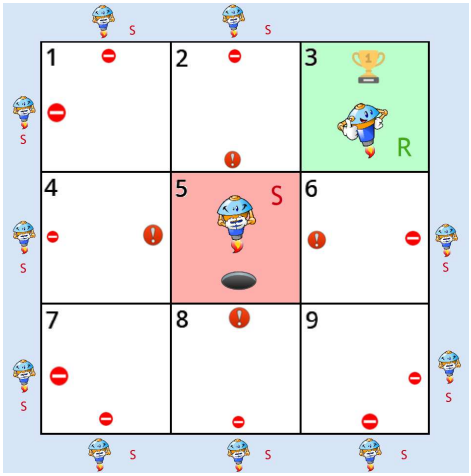
En mémorisant (enregistrant) des nombres (on visualisera cela avec le programme scratch) qui tiennent notamment compte de l'importance de la sanction ou de la récompense qu'il a reçue lorsqu'il a emprunté une certaine direction et du nombre de fois que cette sanction ou que cette récompense a été obtenue.

- Directions menant à des sanctions

On va tout d'abord représenter par des symboles rouges ces nombres pour les directions à éviter

⛔ Un nombre qui « tient compte du fait » que Cipy va recevoir une petite sanction s'il suit une direction qui est interdite (sortir de la grille)

❗ Un nombre qui « tient compte du fait » que Cipy va recevoir une grosse sanction s'il suit une direction qui le mène au trou



Les symboles en rouge représentent des directions à éviter. **Plus les symboles rouges sont grands au sein d'une même case, plus ces directions doivent être évitées.**

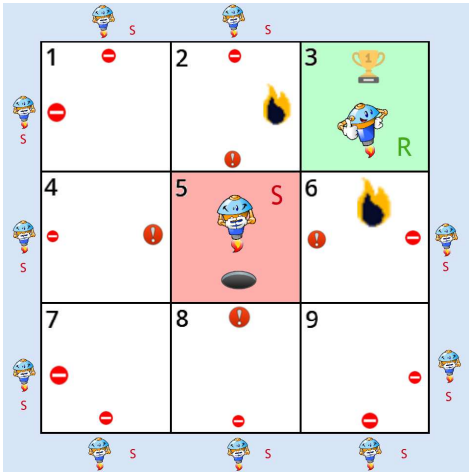
Lorsqu'on se trouve sur les cases 2, 4, 6 et 8, on constate que ce sont les directions vers le trou (case 5) qui sont à éviter. Pour les autres cases, la grandeur des nombres (des symboles) dépend essentiellement du nombre d'expériences aléatoires dans l'une ou l'autre direction. Les sanctions sont des nombres négatifs. Les symboles sont donc représentés ici en valeur absolue !

- Directions menant à la récompense

Il y a seulement 2 cases qui y mènent directement (cases 2 et 6).

On va représenter les nombres qui « rendent compte du fait » que Cipy obtient la récompense (R) s'il suit les directions qui le mènent à la coupe par une flamme

Les symboles qui représentent des directions à suivre pour obtenir la récompense ont des tailles différentes car ce ne sont pas des nombres identiques. Plus ils sont grands, plus ces directions doivent être suivies. La grandeur des nombres (des symboles) dépend essentiellement ici du nombre d'expériences aléatoires dans l'une ou l'autre direction et de l'importance de la récompense (Ici une seule récoimpense).



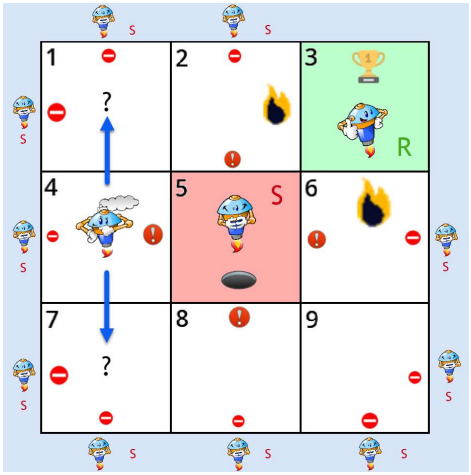
À ce stade, il est possible de représenter des directions à suivre ou à ne pas suivre en mémorisant des nombres (on pourra les visualiser avec le programme scratch) qui tiennent compte de la sanction ou de la récompense obtenue, de son importance et du nombre de fois que ces sanctions ou récompenses ont été obtenues au cours de l'apprentissage (renforcement).

- Slide 24 - 25

Question : Cipy peut-il se diriger dans le labyrinthe et trouver la direction menant à la coupe ?

À ce stade, Cipy pourrait éviter de sortir du labyrinthe ou de tomber dans le trou mais n'est pas encore capable de se déplacer dans la bonne direction pour atteindre la coupe !

S'il se déplace vers une autre case blanche, il ne reçoit aucune récompense ou sanction !

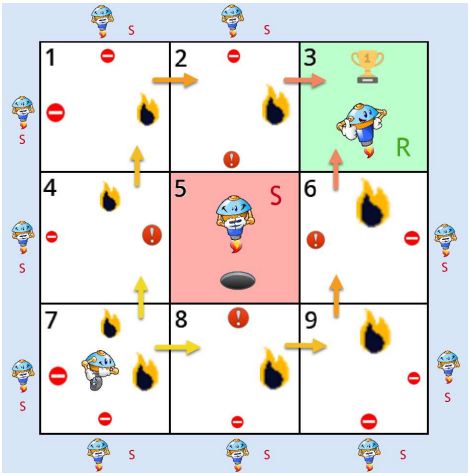


- Slide 26 - 30

Les directions à suivre sont également représentées par l'algorithme par des nombres. Pour symboliser ces nombres, on peut penser à faire une analogie avec le jeu du **« Chaud-Froid ».** Plus Cipy va prendre une direction qui le rapproche de la coupe, plus cela va devenir « chaud ». Plus Cipy va prendre une direction qui l'éloigne de la coupe, plus cela va devenir « froid ».

On va symboliser « plus chaud » par une flamme plus ou moins grosse selon l'intensité de la chaleur.

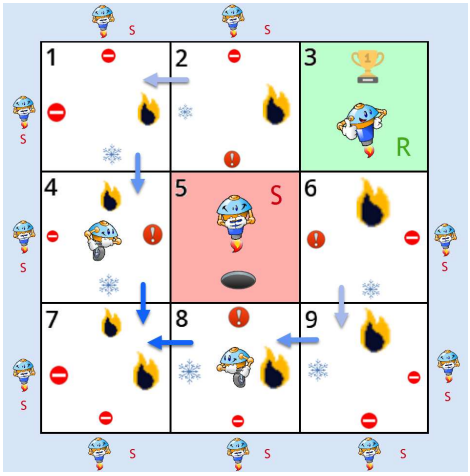
On part des cases 2 et 6 pour lesquelles, on a déjà deux symboles . Ceux-ci représentent la chaleur maximale (on se trouve sur des cases contigües à la récompense). Si l'on suit la direction indiquée par ces symboles, on obtient directement la récompense. Si on recule sur une case plus lointaine (cases 1 et 9 par exemple), la chaleur ressentie sera moindre en direction de la coupe. On peut donc représenter ce ressenti en diminuant la taille de la flamme lorsqu'on s'éloigne de la coupe.



- Slide 31 - 34

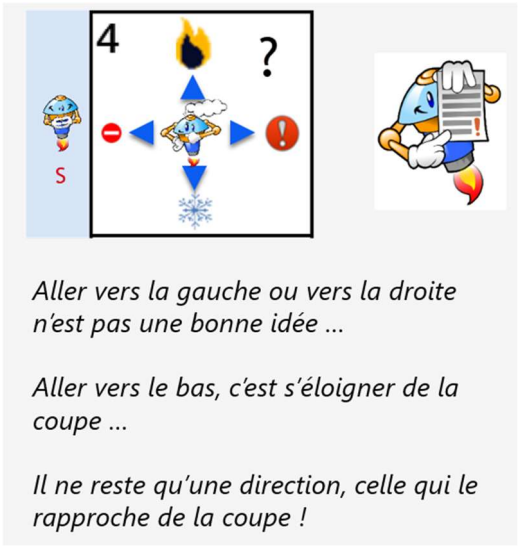
On va maintenant symboliser « plus froid » par un cristal de glace

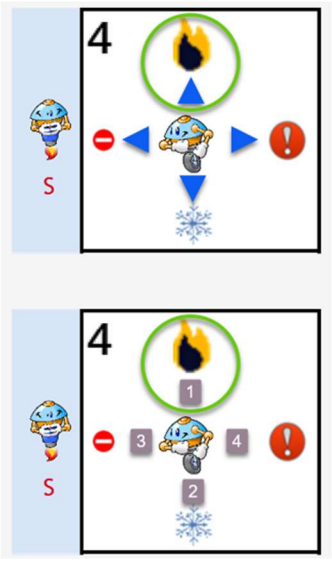
Si Cipy prend une direction qui l'éloigne de la coupe, il va ressentir du froid. Ce froid va être tout d'abord symbolisé par sur les cases 2 et 6 les plus proches de la récompense. Plus, il va s'éloigner de la récompense et plus le ressenti de froid va s'accentuer. Cela se concrétise par une augmentation de la taille du cristal de glace.



- Slide 35 - 39

Question : Si l'on prend en compte toutes les informations disponibles, Cipy peut-il maintenant se déplacer dans le labyrinthe pour atteindre la coupe ?





Les différents symboles dans une case du labyrinthe permettent d'établir un classement des directions et de déterminer la meilleure option permettant d'atteindre la coupe. Les directions vers la droite et la gauche mènent à des sanctions et la direction vers le bas

éloigne Cipy de la récompense. La meilleure option (N°1) est donc d'aller vers le haut, dans la direction qui rapproche de la récompense (chaleur ressentie). Cipy classe continuellement les directions possibles pour se diriger.

Lorsqu'il exploite ses connaissances, Cipy emprunte la direction classée N°1.

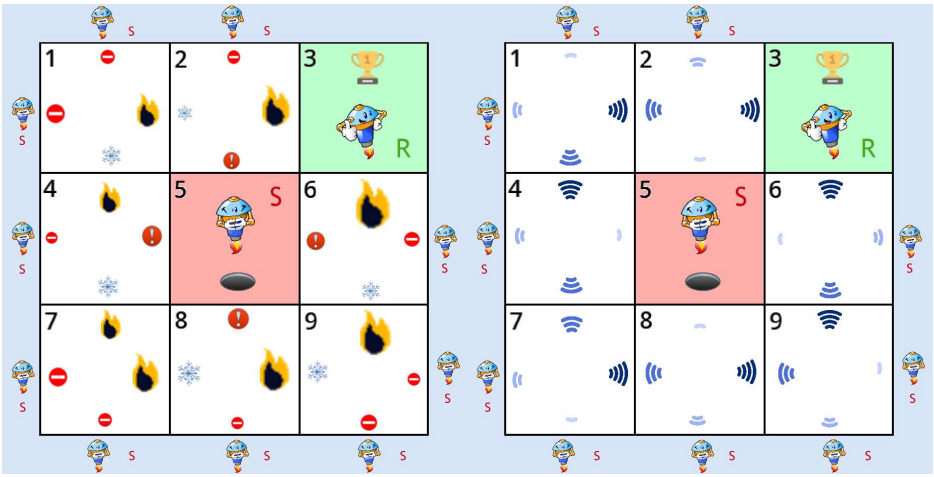
- Slide 41

Question : Quel est le lien avec les symboles en arcs de cercle utilisées dans le jeu ?

Les symboles en arcs de cercles représentent un classement des directions différent basé sur la valeur réelle des nombres. La direction indiquée par les 4 arcs de cercle est le plus grand nombre des 4. Il indique la meilleure direction à suivre si l'on veut atteindre la récompense.

Les autres symboles  indiquent des directions moins indiquées pour atteindre la récompense.

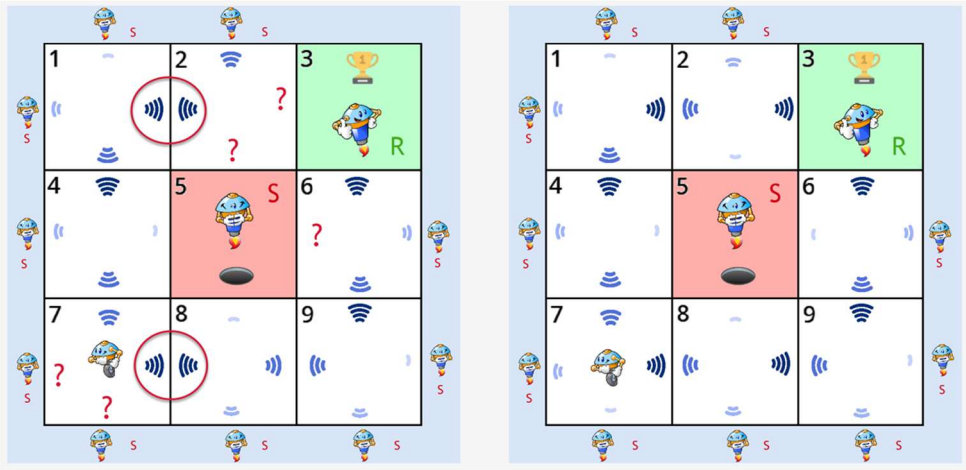
Si on observe attentivement le labyrinthe, les symboles  pointent toutes vers des directions qu'il n'est pas vraiment souhaitable d'emprunter : cases inaccessibles, extérieur de la grille, incendie, travaux, marche arrière sur l'itinéraire, etc. Si l'on suit ces directions, il est vraiment douteux qu'on puisse arriver à la récompense.



- Slide 42-43

Question : La direction classée N°1 est-elle toujours la meilleure direction à suivre ?

Les résultats de l'apprentissage n'indiquent pas toujours la meilleure direction, surtout au début de l'apprentissage. Il faut du temps et de multiples essais et erreurs pour que l'algorithme puisse établir un classement correct. Il faut notamment que toutes les directions possibles aient été empruntées.

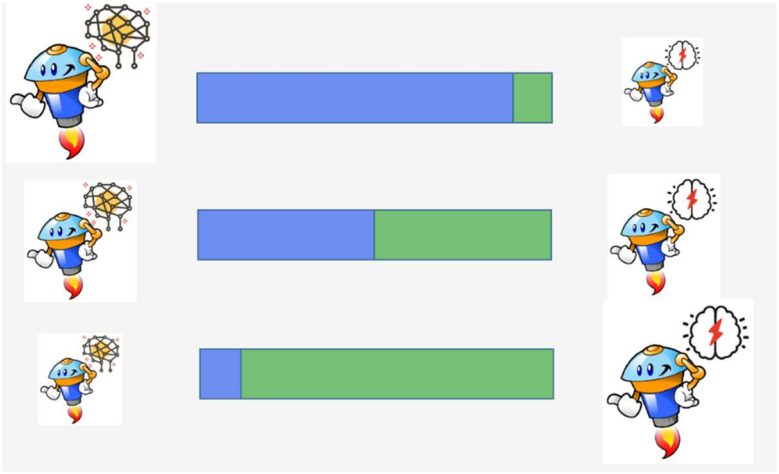


La direction classée en 1 pour chaque case (4 arcs de cercle) est la meilleure direction à suivre à un certain moment pour **espérer** obtenir la récompense. **Ce n'est pas une certitude**. Progressivement, après un apprentissage suffisant, on pourra considérer que le chemin optimal aura été trouvé.

L'algorithme met à jour et classe constamment les résultats obtenus lors de son parcours dans le labyrinthe de la plus haute espérance d'atteindre la coupe à la plus petite.

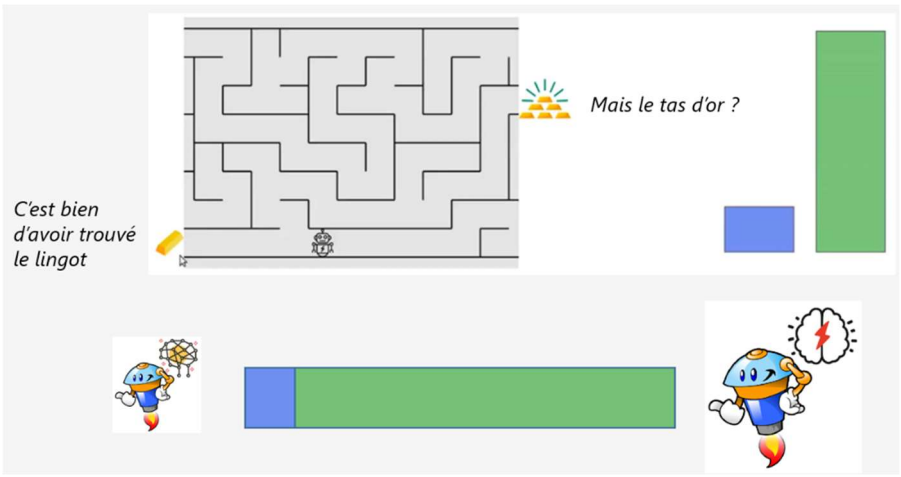
- Slide 44

En fait, en début d'apprentissage, l'IA explore beaucoup le labyrinthe de manière aléatoire car elle n'a pas encore fait suffisamment d'expériences. Ensuite, l'IA peut exploiter de plus en plus les résultats obtenus. L'évolution de la proportion entre exploration aléatoire de l'environnement et exploitation des données issues de l'apprentissage au fil du temps peut-être représentée comme suit.



- Slide 45

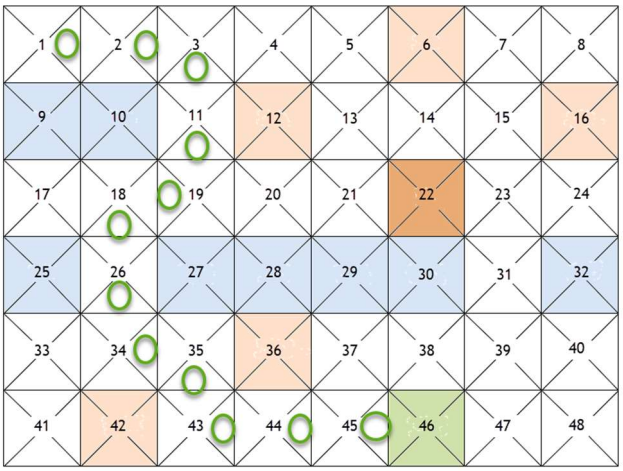
Même après beaucoup d'entraînement, l'IA essayera encore de s'améliorer pour découvrir de meilleures solutions ou un trésor plus important …



Question : Revenons quelques instants au jeu. Quel est le chemin trouvé par l'IA pour se rendre de la case 1 à la coupe ?

N.B. : Le tableau complet du labyrinthe avec les symboles (arcs de cercle de couleur bleue) est dévoilé ou est projeté sur écran.

Chaque équipe détermine, si nécessaire, le chemin trouvé par l'IA pour atteindre la coupe à partir de la position 1 en suivant les directions indiquées par les symboles : 1 – 2 – 3 – 11 – 19 – 18 – 26 – 34 – 35 – 43 – 44 – 45 – 46



A retenir en synthèse

L'IA utilise un algorithme qui lui dit comment apprendre mais pas ce qu'elle doit apprendre. Petit à petit, l'IA est capable de trouver toute seule son chemin vers la coupe en apprenant de ses erreurs et de ses succès, en recevant des récompenses lorsqu'elle tombe sur la coupe ou des sanctions lorsqu'elle fait des choses interdites ou moins indiquées. Elle parvient progressivement à déterminer pour chaque direction qu'elle peut suivre l'espérance qu'elle a d'atteindre la récompense si elle suit cette direction.

La direction classée en 1 est la meilleur direction à suivre pour espérer atteindre la récompense à un moment donné de l'apprentissage. L'apprentissage peut durer plus ou moins longtemps. Les solutions trouvées par l'IA peuvent être erronées ou peuvent être susceptibles d'amélioration. Même après un long apprentissage, une IA continue à explorer son environnement au hasard afin de trouver une éventuelle meilleure solution. Elle peut donc progresser.

- Slide 46 (bonus)

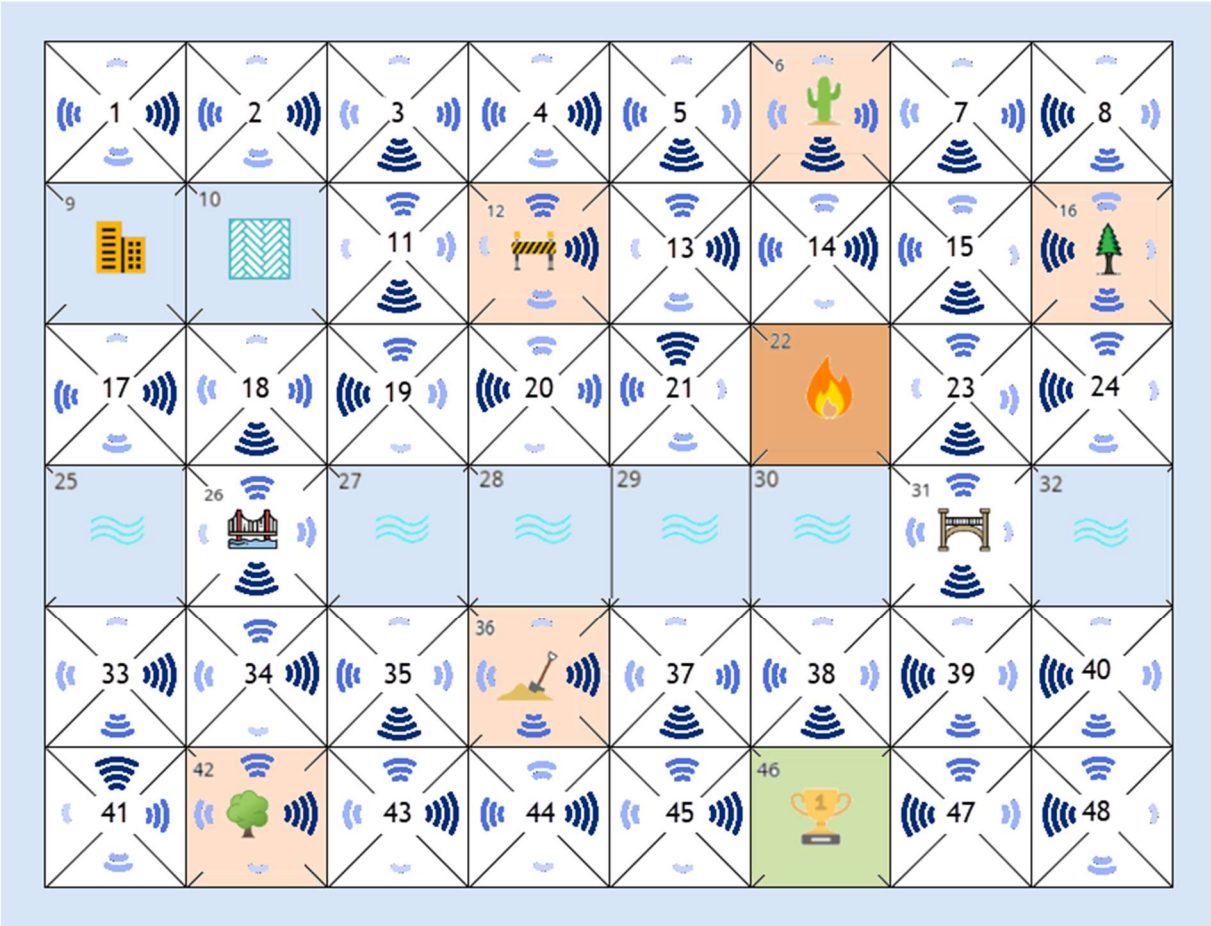
Montrer en vidéo un exemple d'application du Q-learning : la voiture autonome

- Slide 47 (bonus)

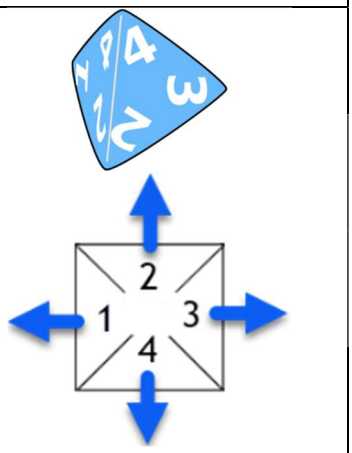
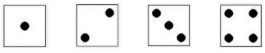
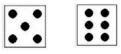
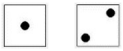
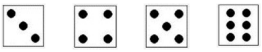
On utilise l'application scratch, en l'occurrence, les labyrinthes 1 (le labyrinthe simple utilisé pour faire émerger les principes du Q-learning) et 4 (celui du jeu).

Cf. la fiche 'Q-learning en scratch' en annexe 3

ANNEXE 1 : Plateau de jeu



ANNEXE 2 : Consignes du jeu du labyrinthe

Coup	Jet du dé à 6 faces **pour déterminer la manière de jouer**		Jet du dé à 4 faces **pour déterminer la direction au hasard**
	On joue au hasard si le résultat est	On joue en suivant la direction indiquée par le symbole  si le résultat est	On suit la direction correspondant au résultat du jet de dé
1 à 20	-	-	
21 à 40			
41 à 60			
61 et plus			

ANNEXE 3 : Q-learning en Scratch

Présentation

L'application propose 7 labyrinthes ayant un niveau de complexité croissant. Cipy, l'intelligence artificielle (IA), apprend à trouver la récompense (coupe ou lingots d'or) placée dans le labyrinthe en évitant les pièges et embûches.

Commandes et icônes


Cipy, l'intelligence artificielle (IA)


Cipy est en train d'explorer l'environnement

 Cipy est en train d'exploiter les données qu'il a acquises par l'apprentissage

Valeur actuelle du facteur ?-greedy

Épisode d'apprentissage en cours

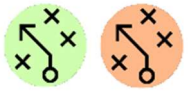
 Afficher / Cacher la liste des paramètres d'apprentissage

Liste des paramètres avec leur valeur par défaut (peuvent varier en fonction de la grille) :

1 : numéro de la grille utilisée : 1
2 : nombre d'épisodes d'entraînement : 0
3 : facteur ?-greedy : 1
4 : valeur maximale du facteur ?-greedy : 1
5 : valeur minimale du facteur ?-greedy : 0,05
6 : facteur de réduction d'?-greedy : 0.996
7 : nombre d'épisodes après lesquels le facteur ?-greedy est diminué : 10
8 : facteur alpha : 0,8
9 : facteur gamma : 0,9
10 : pénalité d'étape : -0,01
11 : pénalité de sortie de la grille : -0,1

 Actualisation de la liste « Parametres » avec les données actuelles

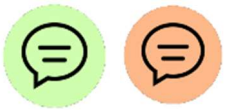
 Initialisation de la poursuite d'un apprentissage à partir des données de la liste « Parametres »

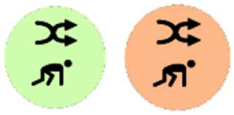
 Montrer / Cacher les déplacements de Cipy

 Activer / Désactiver la mise à jour des triangles / carrés de couleur représentant les Q-Values

 Désactiver (éclair vert) / Activer (éclair rose) un retard de 0,5 seconde dans l'affichage des déplacements de Cipy (Par défaut l'état est dans le mode activé)

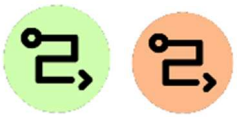
 Activer / Désactiver l'affichage d'une flèche rouge indiquant la direction suivie par Cipy

 Activer / Désactiver l'affichage des messages de Cipy lorsqu'il tombe sur un état terminal

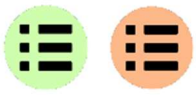
 Activer / Désactiver le départ d'un épisode d'apprentissage à partir d'une position aléatoire de la grille (p.m.)

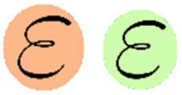
 Effacer les triangles / carrés de couleur symbolisant les Q-values

 Afficher / Rafraîchir la totalité des triangles / carrés de couleur représentant les Q-values

 Activer / Désactiver l'affichage de l'itinéraire trouvé par Cipy

 (Re)Lancer / Mettre en pause l'apprentissage

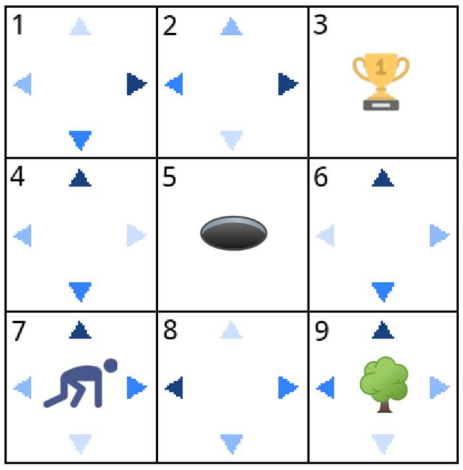
 Afficher / Cacher les listes de données d'apprentissage « Gauche », « Haut », « Bas », « Droite »

 Aficher / Cacher la variable ?-greedy

Algorithme de Q-learning

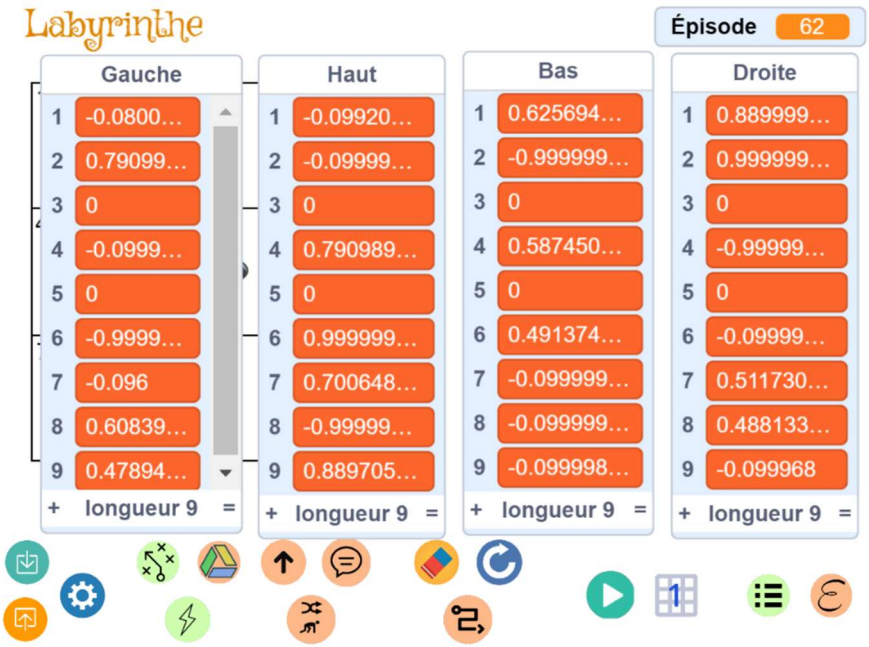
L'algorithme Q-learning utilisé dans l'application est détaillé dans la fiche 'Q-learning'.

Les Q-values sont représentées ici graphiquement par des triangles de direction ou des carrés de couleur (grilles 5 à 7) allant du bleu très foncé (la plus haute Q-value) au bleu très clair (la plus petite Q-value). Les triangles gris représentent les directions qui n'ont pas encore été testées par l'IA.



Les Q-values réelles peuvent être affichées dans les listes par simple pression sur l'icône  .

Les numéros d'ordre dans les listes correspondent aux numéros des cases du labyrinthe.



Processus d'apprentissage

Déroulement d'un apprentissage

Le processus d'apprentissage peut être long notamment pour les labyrinthes les plus complexes. Il est donc possible de récupérer les données d'apprentissage et de les réutiliser ultérieurement afin de poursuivre l'entraînement.

Pour sauvegarder les données d'apprentissage :

 Mettre l'apprentissage en pause sans arrêter l'application

 Afficher la liste « Parametres »

 Mettre à jour la liste « Parametres » avec les données les plus récentes

Passer en mode « réduction de fenêtre »

Enregistrer les données de la liste « Parametres » sur l'ordinateur au moyen de la commande « Exporter » (clic droit avec la souris dans la liste).

 Afficher les listes « Gauche », « Haut », « Bas », « Droite » et enregistrer les données de ces listes sur l'ordinateur au moyen de la commande « Exporter ».

 Arrêter l'application

Pour importer des données d'entraînement et reprendre l'apprentissage:

 Lancer l'application

 Afficher la liste « Parametres »

Passer en mode « réduction de fenêtre »

« Importer » les données sauvegardées précédemment dans la liste « Parametres » au moyen de la commande « Importer » (clic droit avec la souris dans la liste).

 Presser l'icône afin d'appliquer les paramètres d'apprentissage et la grille ad hoc s'affiche.

 Afficher les listes « Gauche », « Haut », « Bas », « Droite ».

« Importer » dans chaque liste les données sauvegardées précedemment au moyen de la commande « Importer ».

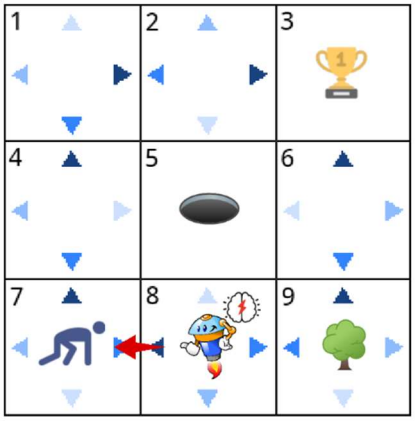
 Relancer l'apprentissage.

NB : Si nécessaire, certains paramètres peuvent être modifiés directement dans le fichier .txt reprenant les données à importer dans la liste « Parametres » avant leur importation.

Visualiser l'apprentissage

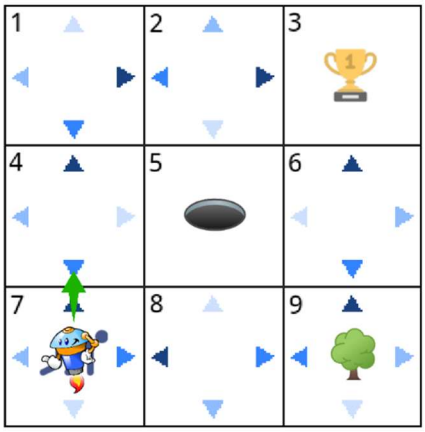
Il est possible de visualiser le processus d'apprentissage de l'IA. L'affichage graphique plus complexe ralentit toutefois la progression de Cipy.

Montrer les directions suivies par Cipy en cours d'apprentissage



 Cette commande active l'affichage d'une flèche rouge indiquant la direction vers laquelle Cipy se dirige ou tente de se diriger.

Montrer l'itinéraire vers la récompense



Après interruption de l'apprentissage, la pression sur ce symbole entraîne la progression d'une flèche verte à partir de la position de départ en suivant la direction indiquée par les Q-values maximales de chaque case. Des messages d'erreur peuvent être générés : « Itinéraire en boucle », « Apprentissage insuffisant », etc. Dans ce cas, l'apprentissage doit être poursuivi pendant un certain nombre d'épisodes.

Des listes de données d'apprentissage sont disponibles pour chaque labyrinthe. Il suffit d'importer les listes selon la procédure mieux décrite supra et de cliquer sur l'icône.

Si une position de départ spécifique n'est pas entrée, l'itinéraire commence à la position de départ par défaut (ici la case 7).

Caractéristiques des labyrinthes

Les cases ordinaires entraînent une petite pénalité d'étape de - 0,01. Il s'agit d'une méthode classique pour encourager l'agent à trouver le chemin le plus court parmi ceux qui lui permettent d'atteindre l'objectif final.

Les tentatives de l'agent (Cipy) pour sortir de la grille ou atteindre un élément infranchisable ou inaccessible entraînent une pénalité de - 0,1. Lorsque Cipy tente d'accéder à un élément infranchissable, inaccessible ou de sortir du labyrinthe, il reste sur place.

Les éléments qui ralentissent sa progression peuvent être traversés moyennant pénalité.

Les cases terminales entraînent la fin d'un épisode d'apprentissage et le retour à la case départ.

Pour les 3 premiers labyrinthes, le nombre d'épisodes après lequel le facteur ?-greedy est diminué est de 1. Pour les autres, le nombre d'épisodes est de 10.

Labyrinthe 1

Case avec	Dénomination	Type de case		Récompense
 ou rien	Case départ	Case ordinaire		- 0,01
	Trou	Case terminale		- 1
	Coupe	Case terminale		+ 1
	Arbres	Élément ralentisseur		- 0,3

Labyrinthe 2

1	2	3	4
5	6	7	8
9	10	11	12

Case avec	Dénomination	Type de case	Récompense
ou rien	Case départ	Case ordinaire	- 0,01
	Bâtiment	Élément inaccessible	- 0,1
	Incendie	Case terminale	- 1
	Coupe	Case terminale	+ 1

Labyrinthe 3

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

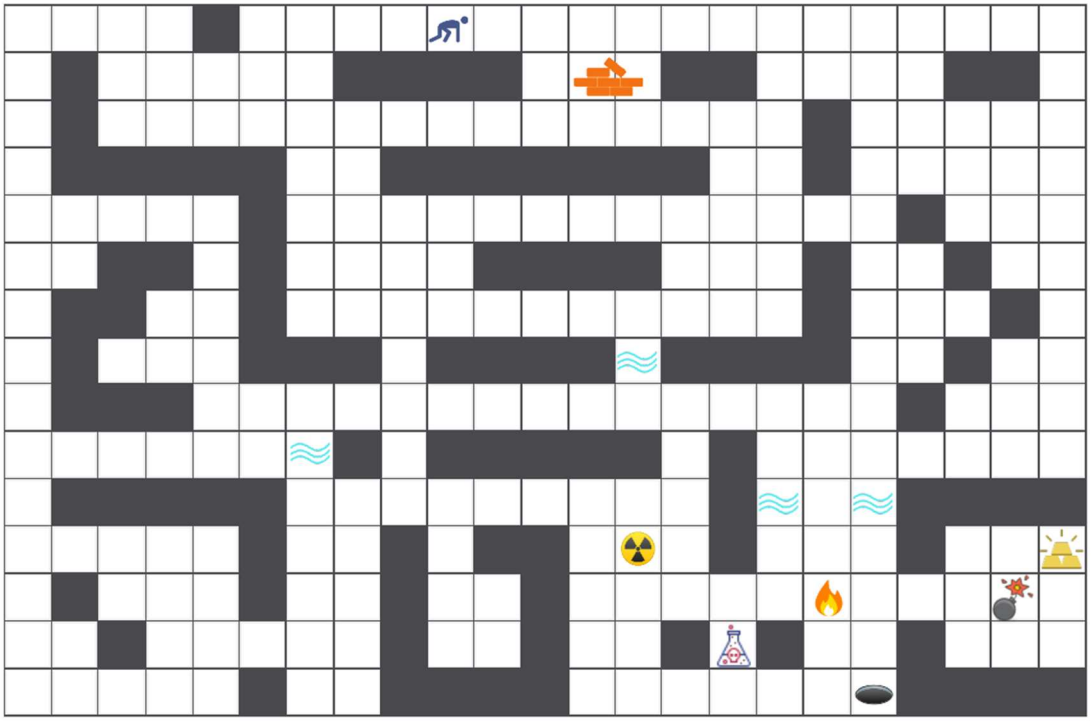
Case avec	Dénomination	Type de case	Récompense
ou rien		Case ordinaire	- 0,01
	Incendie	Case terminale	- 1
	Coupe	Case terminale	+ 1
	Trou	Case terminale	- 1
	Orage	Élément ralentisseur	- 0,3
	Cours d'eau	Élément infranchissable	- 0,1

Labyrinthe 4

Case avec	Dénomination	Type de case	Récompense
ou rien	Case départ	Case ordinaire	- 0,01
	Cactus	Élément ralentisseur	- 0,3
	Building	Elément inaccessible	- 0,1
	Bâtiment	Elément inaccessible	- 0,1
	Travaux	Élément ralentisseur	- 0,3
	Sapins	Élément ralentisseur	- 0,3
	Incendie	Case terminale	-1
	Cours d'eau	Élément infranchissable	- 0,1
	Pont	Case ordinaire	- 0,01
	Pont	Case ordinaire	- 0,01
	Travaux	Élément ralentisseur	- 0,3
	Arbres	Élément ralentisseur	- 0,3
	Coupe	Case terminale	+1

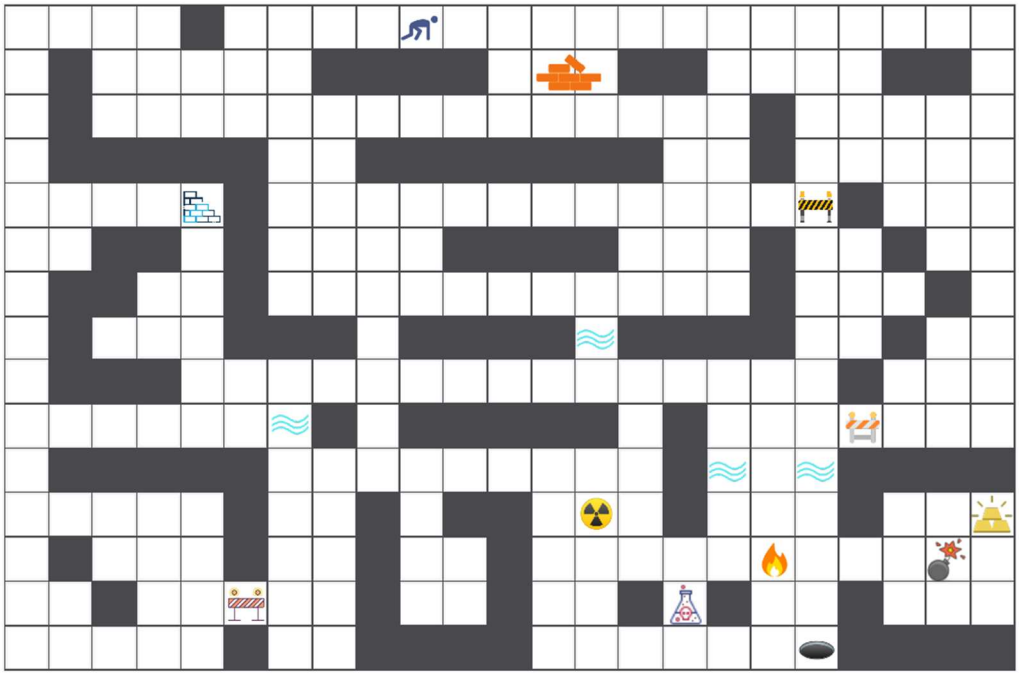
Labyrinthe 5

A partir du labyrinthe 5, les lingots d'or sont protégés par de multiples pièges (Cf. Indiana Jones)



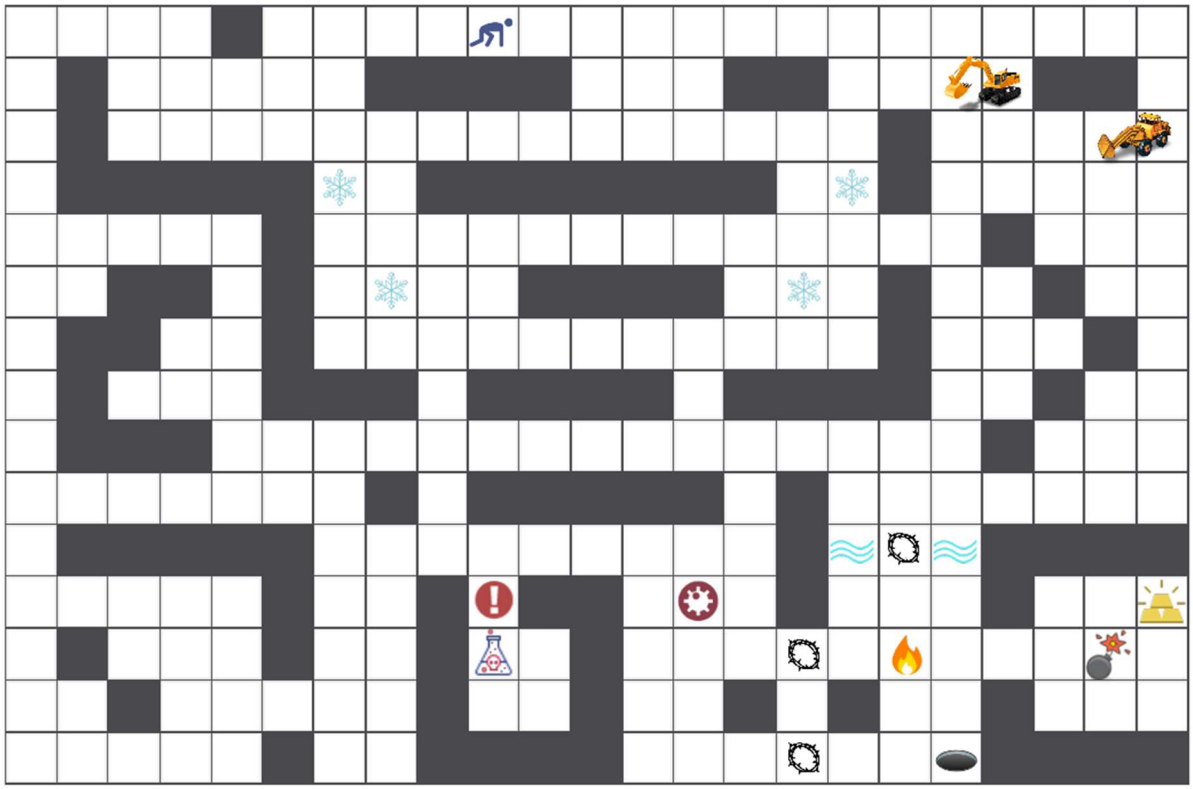
Case avec	Dénomination	Type de case	Récompense
ou rien	Case départ	Case ordinaire	- 0,01
	Cours d'eau	Élément infranchissable	- 0,1
	Mur de briques	Élément infranchissable	- 0,1
	Case noire	Élément inaccessible	- 0,1
	Incendie	Case terminale	- 1
	Radioactivité	Case terminale	- 1
	Produits toxiques	Case terminale	- 1
	Bombe	Case terminale	- 1
	Lingots d'or	Case terminale	+ 1
	Trou	Case terminale	-1

Labyrinthe 6



Case avec	Dénomination	Type de case	Récompense
ou rien	Case départ / …	Case ordinaire	- 0,01
	Cours d'eau	Élément infranchissable	- 0,1
	Mur de briques	Élément infranchissable	- 0,1
	Mur de briques	Élément infranchissable	- 0,1
	Case noire	Élément inaccessible	- 0,1
	Barrière	Élément infranchissable	- 0,1
	Barrière	Élément infranchissable	- 0,1
	Barrière	Élément infranchissable	- 0,1
	Incendie	Case terminale	- 1
	Radioactivité	Case terminale	- 1
	Produits toxiques	Case terminale	- 1
	Bombe	Case terminale	- 1
	Lingots d'or	Case terminale	+ 1
	Trou	Case terminale	-1

Labyrinthe 7



Icône	Dénomination	Type de case	Récompense
	Case départ	Case ordinaire	- 0,01
	Cours d'eau	Élément infranchissable	- 0,1
	Case noire	Élément inaccessible	- 0,1
	Incendie	Case terminale	- 1
	Produits toxiques	Case terminale	- 1
	Bombe	Case terminale	- 1
	Lingots d'or	Case terminale	+ 1
	Trou	Case terminale	-1
	Mine	Élément ralentisseur	- 0,6
	Glace	Élément ralentisseur	- 0,2
	Signal de danger	Élément ralentisseur	- 0,3
	Machine de chantier	Élément ralentisseur	- 0,4

Icône	Dénomination	Type de case	Récompense
	Machine de chantier	Élément ralentisseur	- 0,4
	Barbelés	Élément ralentisseur	- 0,2