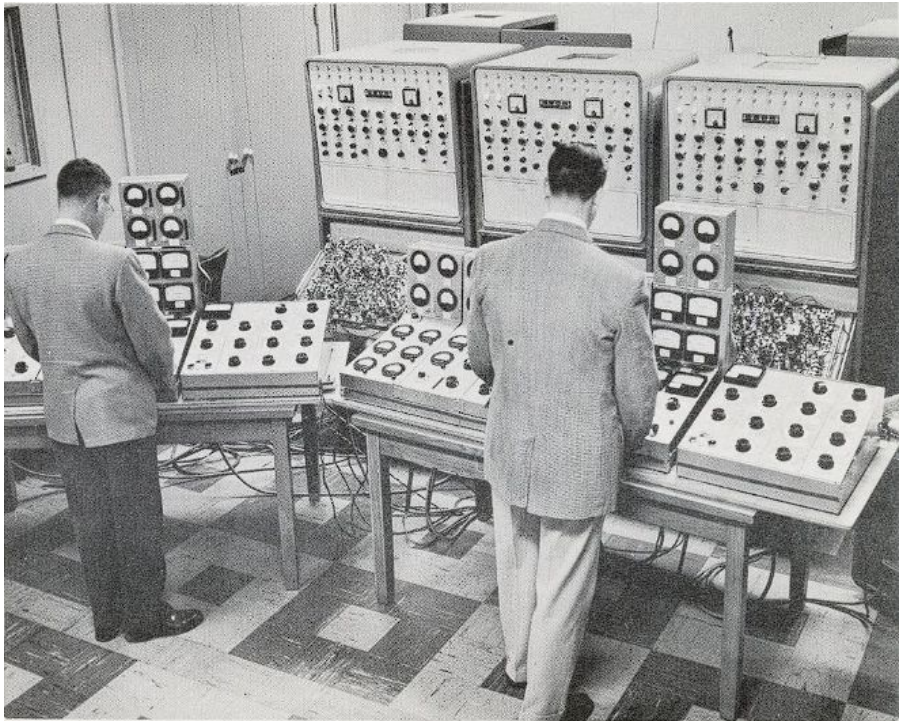
L’apprentissage par renforcement

Comment les machines apprennent à gagner

benoit.crespin@unilim.fr

17 avril 2025

Le jeu de Nim



- 11 allumettes au départ
- Chaque joueur prend à son tour **une** ou **deux** allumettes
- Celui qui **prend la dernière allumette a perdu**

11

https://simplecounter.app/

Comment choisir entre 1 et 2 allumettes en fonction d'une probabilité donnée ?

Probabilité de choisir
1 allumette (0%)
ou
2 allumettes (100%)

Comment gagner au jeu de Nim ?

Nombre d'allumettes restantes	Probabilité de choisir 1 allumette (0%) ou 2 allumettes (100%)
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

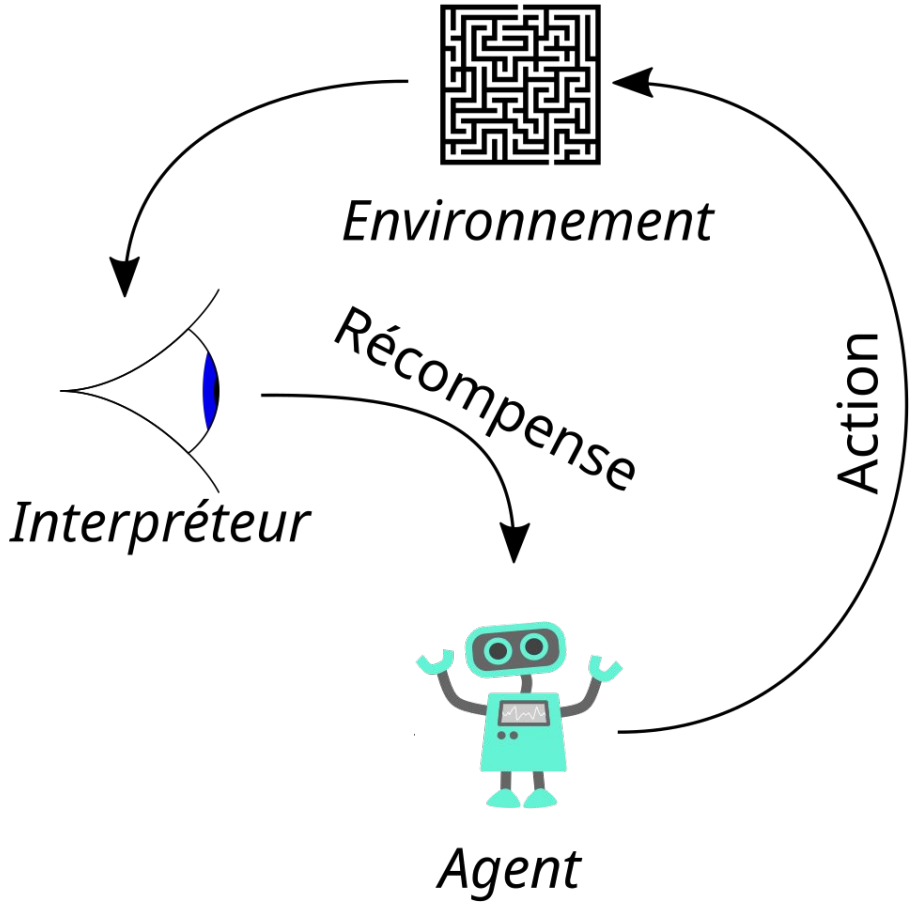
Mise à jour progressive des probabilités

https://scratch.mit.edu/projects/511538914/fullscreen/

Démonstration avec 11 allumettes

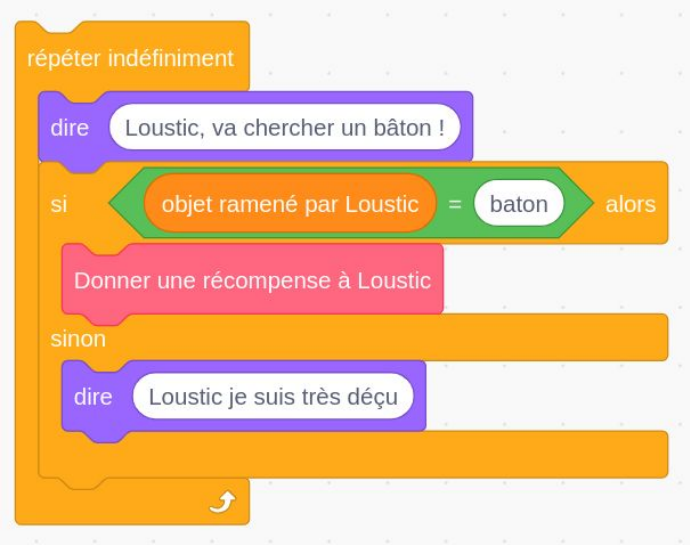
https://scratch.mit.edu/projects/513372592/fullscreen/

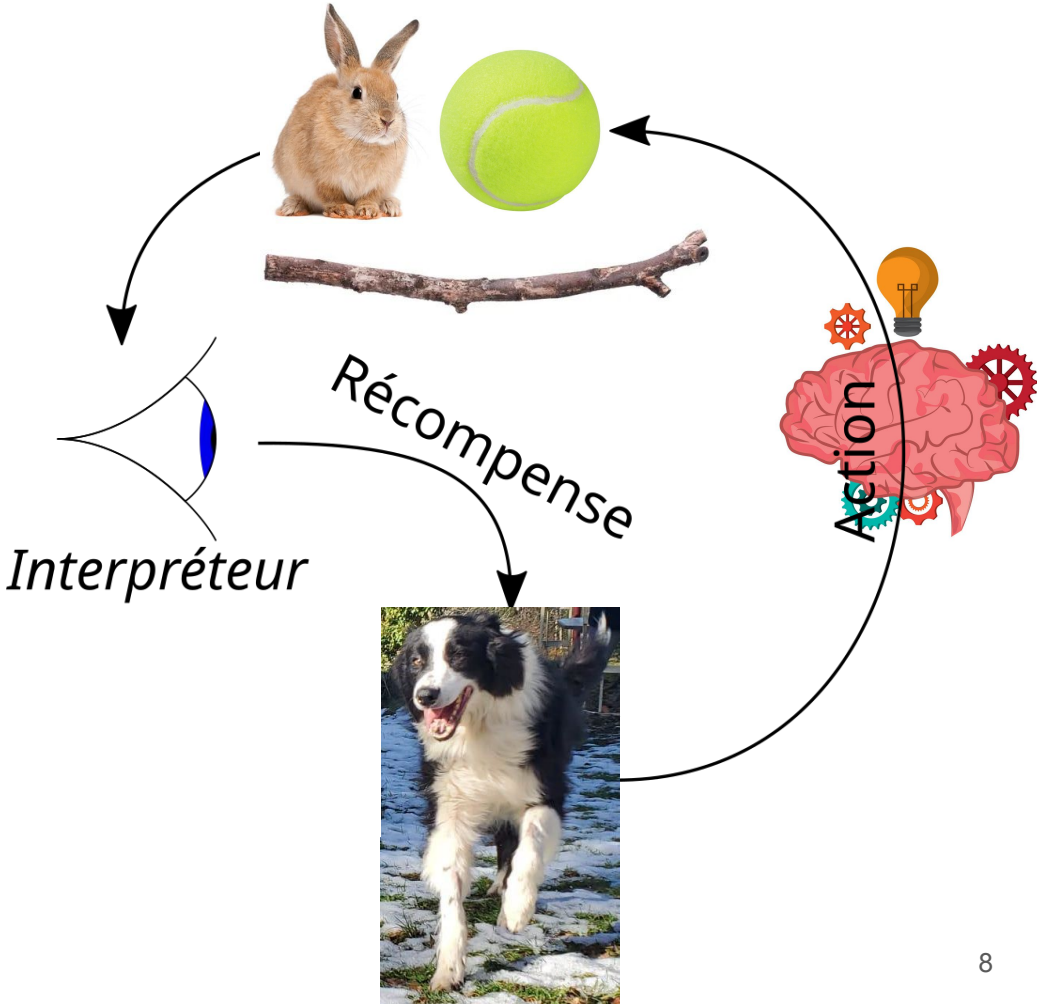
Méthode de résolution



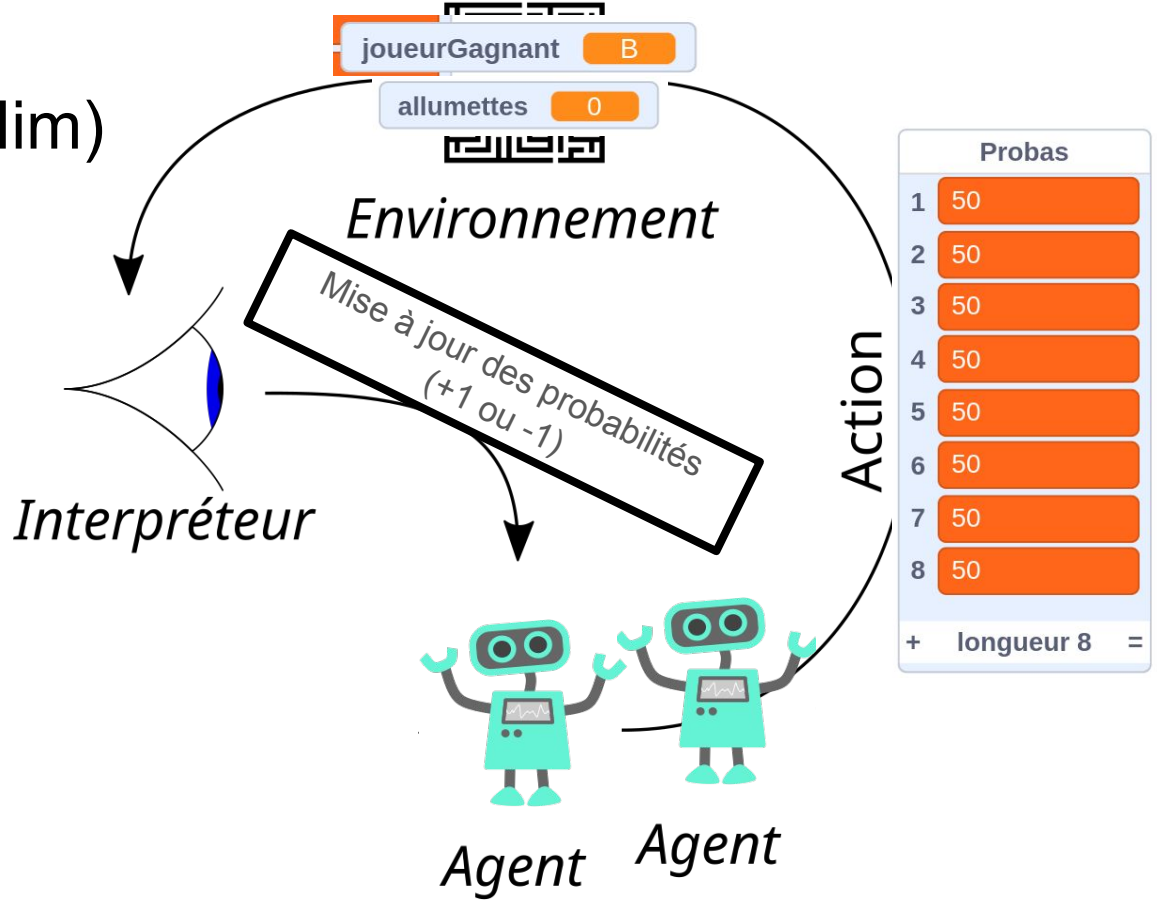
Méthode de résolution (principe original)

Comment entraîner Loustic à ramener un bâton ?



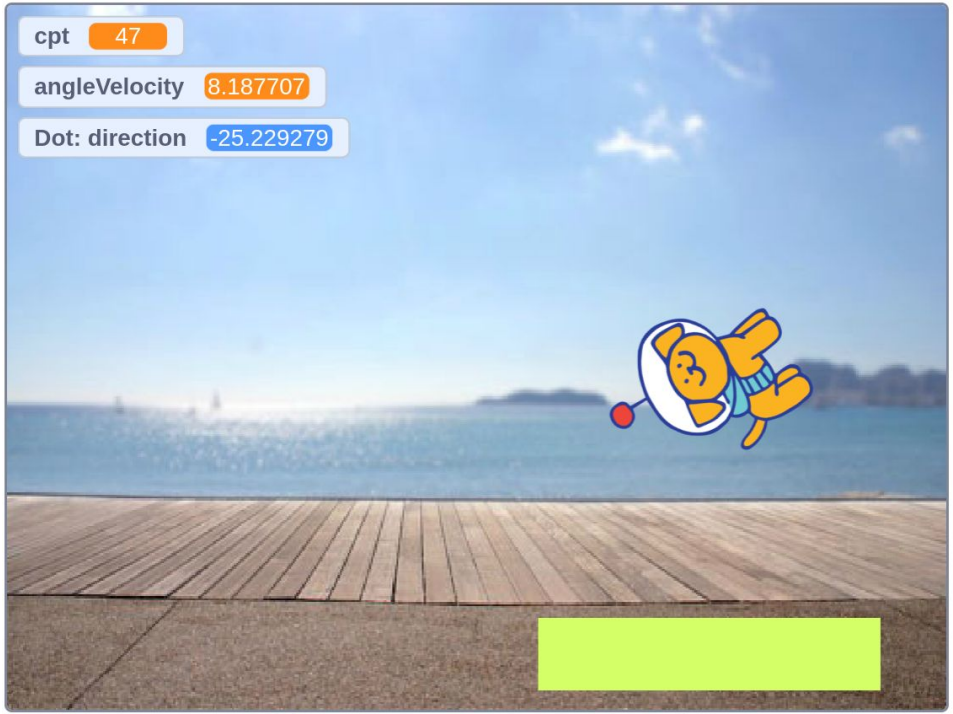


Méthode de résolution (appliquée au jeu de Nim)




TARS

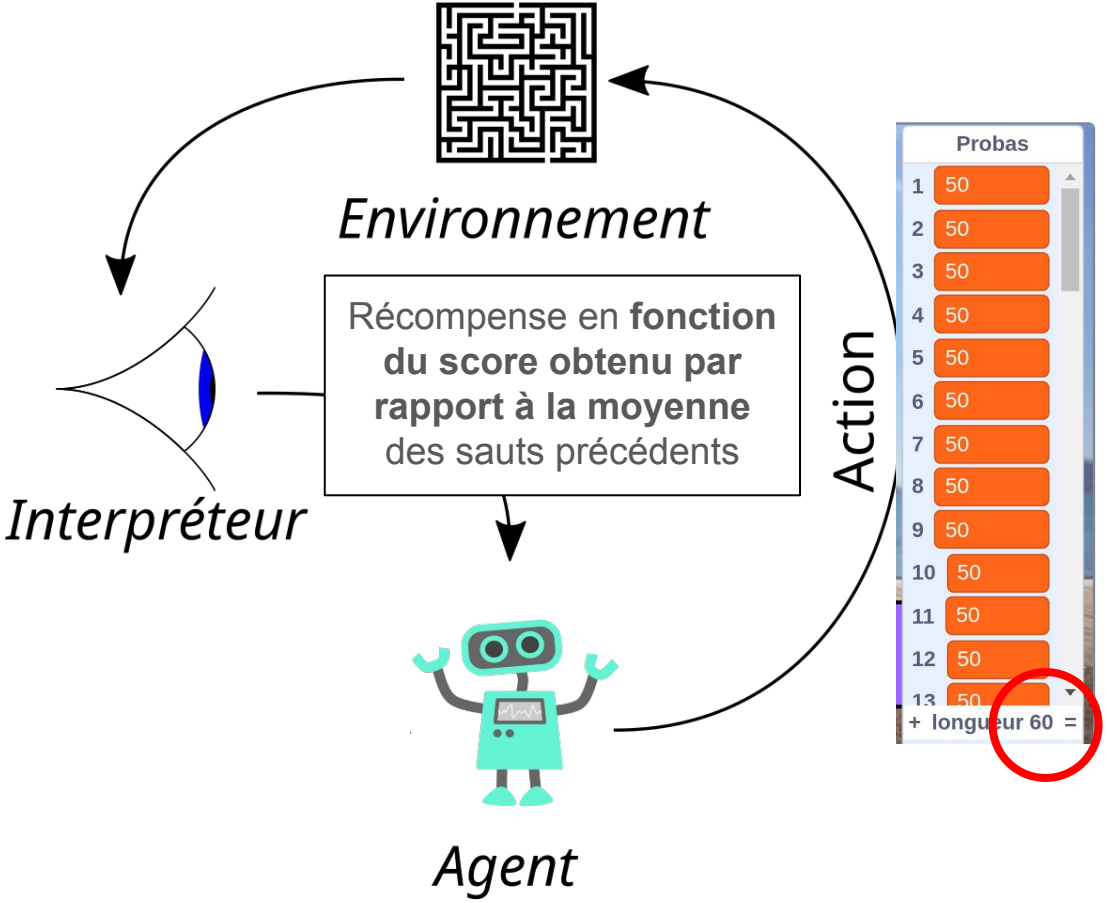
Crazy flip



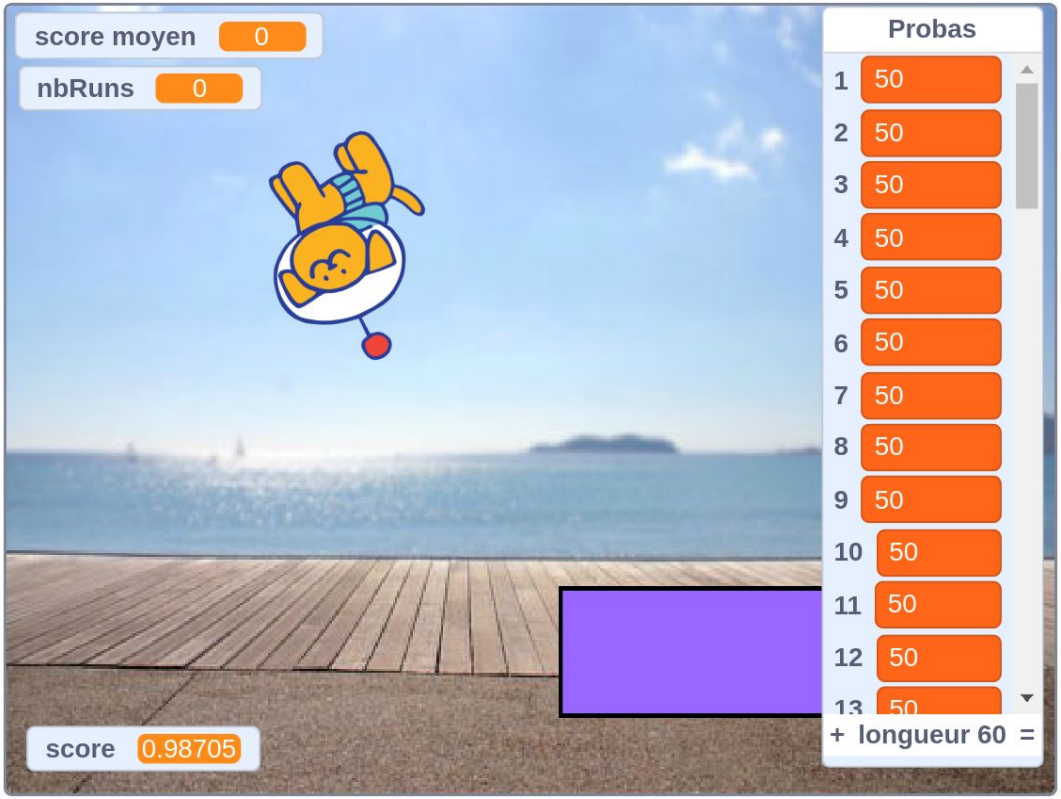
- Le but du jeu est de retomber bien droit
- 2 actions possibles pendant le vol :
  - cliquer pour accélérer la rotation
  - ne rien faire

https://scratch.mit.edu/projects/1058913446/fullscreen/

Méthode de résolution

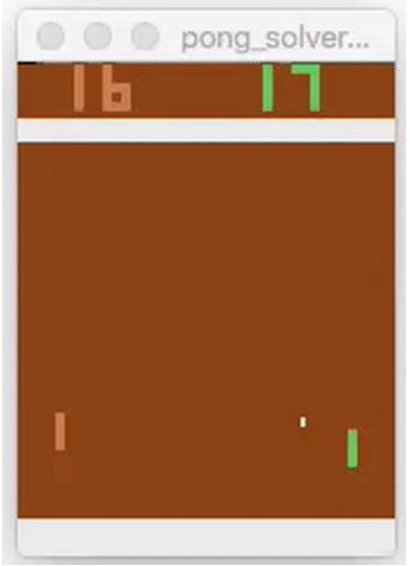


A votre avis, quelle est la stratégie gagnante ?

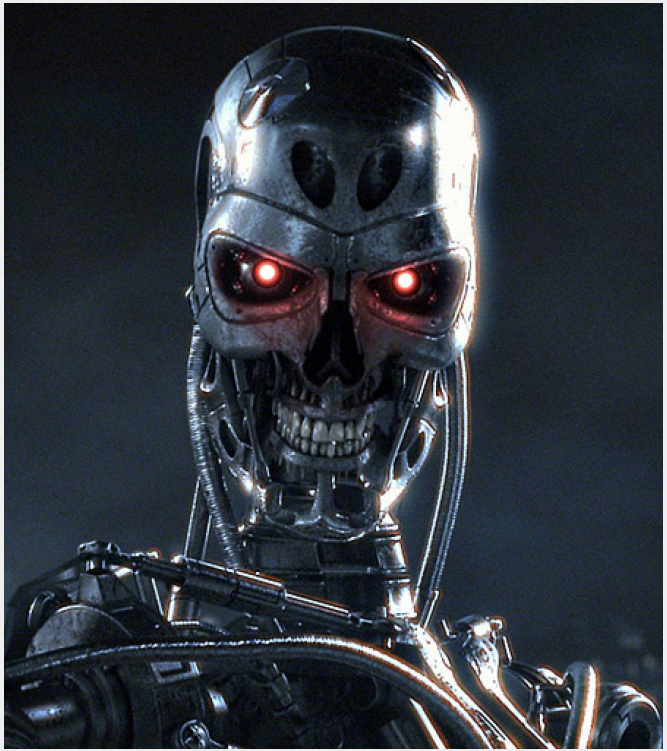


https://scratch.mit.edu/projects/1059008383/fullscreen/

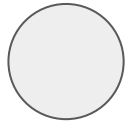
L’apprentissage par renforcement…



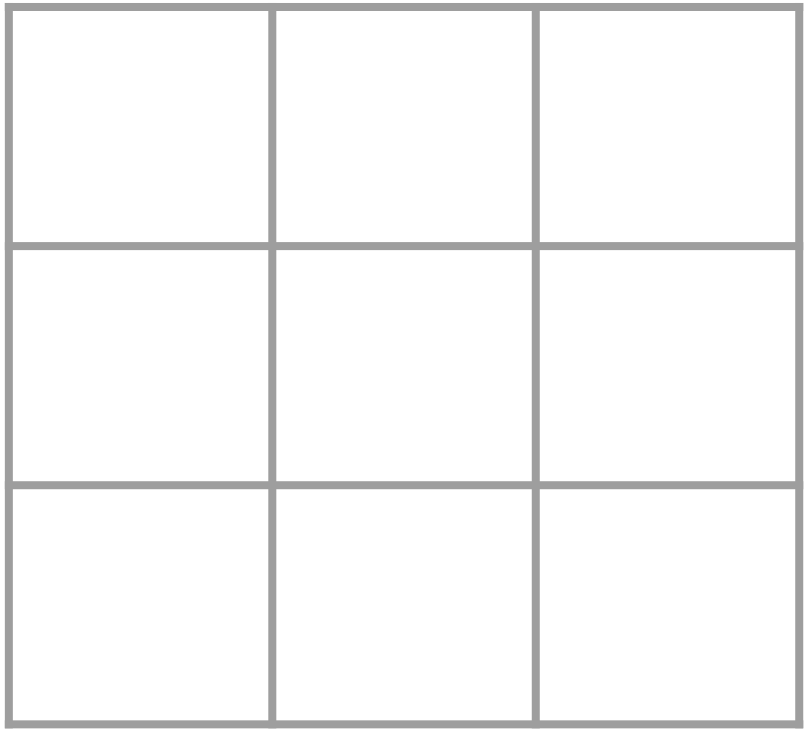
- **L’ordinateur apprend par lui-même**, pas besoin d’un humain
- Une **suite d’actions** fait varier **l’état du système**, et permet d’obtenir un **score** (positif ou négatif)
- Découverte progressive de la meilleure stratégie (**exploration** vs **exploitation**)
- Exemples :
  - jeux vidéos ou jeux traditionnels
  - systèmes mécaniques/robotiques
  - “brique de base” dans des applications plus complexes



Tic-Tac-Toe version “misère”

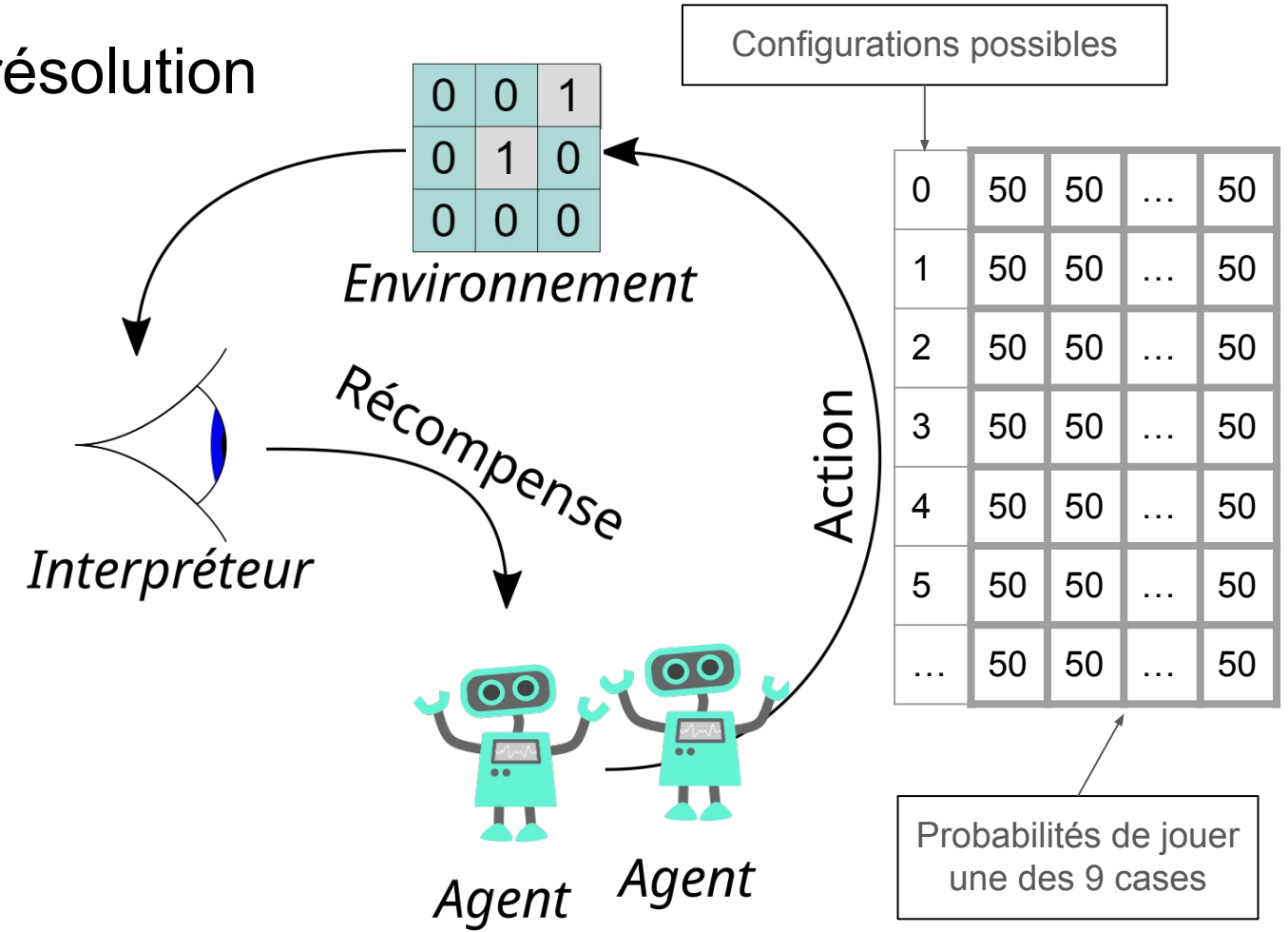


- **Un seul type de pion**
- Chaque joueur place un pion à son tour sur une grille 3x3
- Le joueur **qui complète une ligne** (horizontale, verticale ou diagonale) **a perdu**

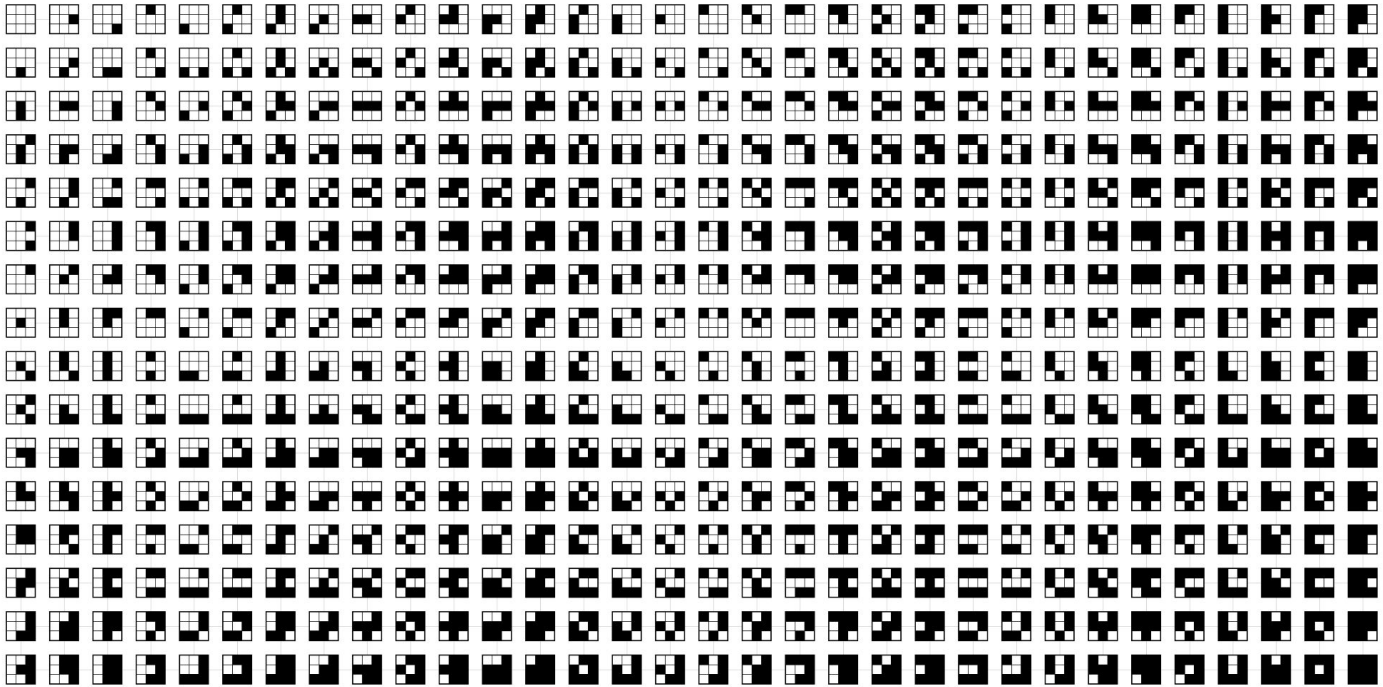


- Stratégie gagnante ?

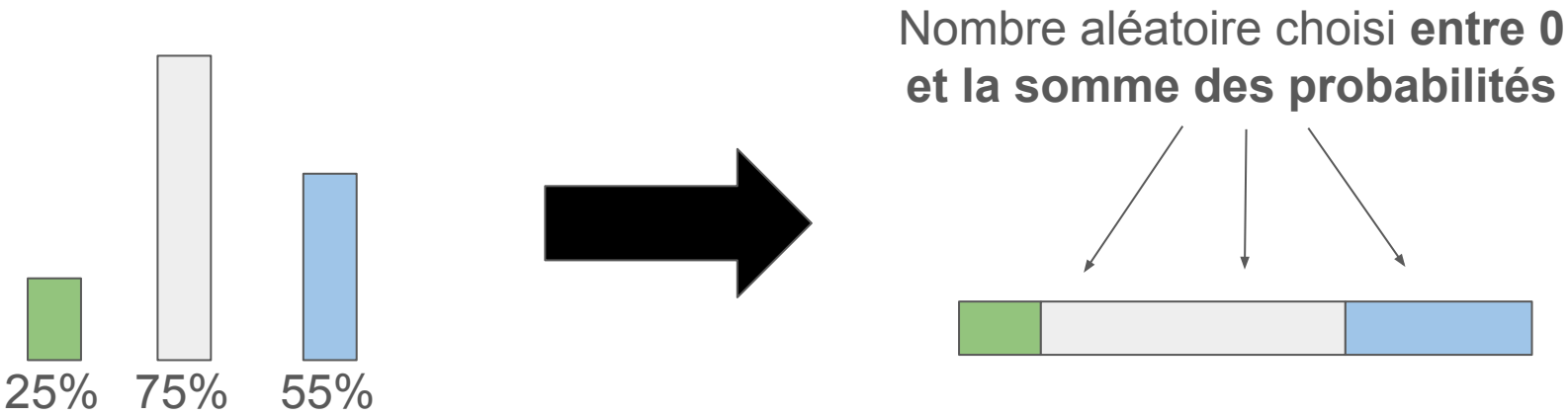
Méthode de résolution



Combien de configurations possibles ?

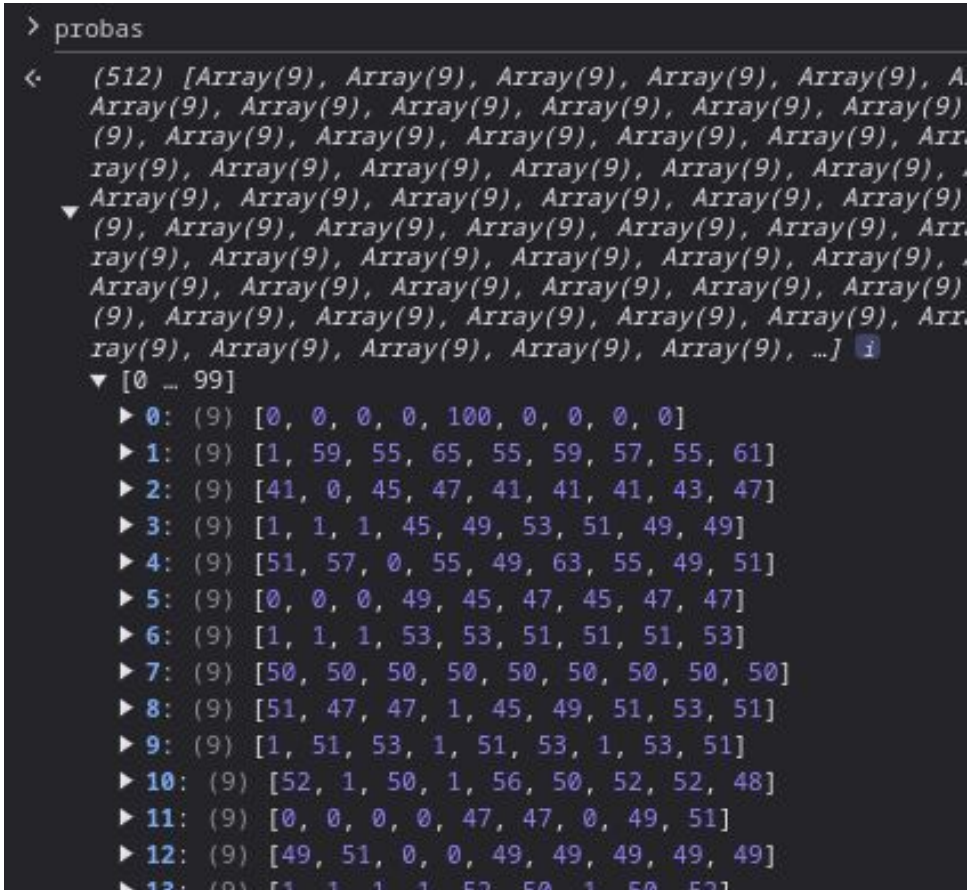


Comment choisir un nombre en fonction de plusieurs probabilités ?

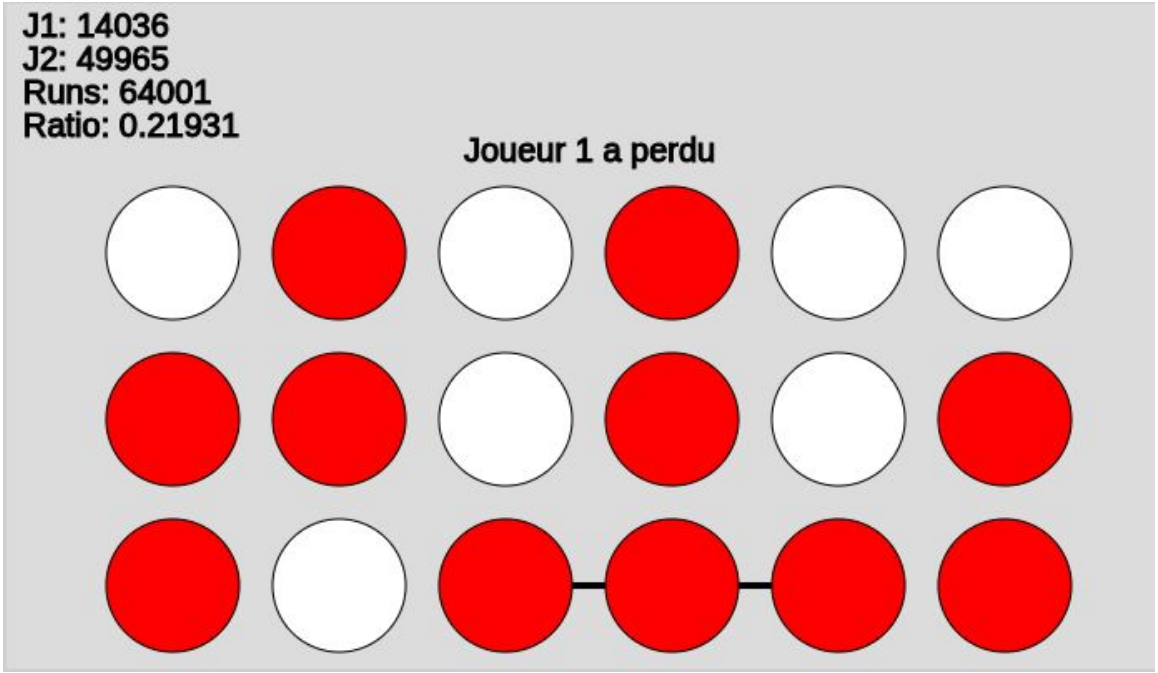


https://openprocessing.org/sketch/2521786

Probabilités obtenues



Version 6x3



https://openprocessing.org/sketch/2611108

Stratégie gagnante ?

Après 25 millions d’itérations…

```
> probas[0]
  ▸ (18) [99, 99, 99, 99, 99, 99, 99, 98, 99, 99, 99, 99, 99, 99, 99, 99, 100, 99]
> probas[1]
  ▸ (18) [0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 0, 0]
> probas[2]
  ▸ (18) [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0]
> probas[3]
  ▸ (18) [1, 99, 95, 100, 100, 99, 1, 97, 100, 91, 100, 100, 1, 100, 100, 100, 93, 100]
```

Ratio = 0.02251

Combien de configurations possibles ?

Jeu	Taille du problème	Résultat
Tic-Tac-Toe “misère” v1	3x3 cases, 2 valeurs possibles	
Tic-Tac-Toe classique	3x3 cases, 3 valeurs possibles	
Tic-Tac-Toe “misère” v2	6x3 cases, 2 valeurs possibles	
Puissance 4	7x6 cases, 3 valeurs possibles	
Echecs	8x8 cases, 7 valeurs possibles	
Nombre d’atomes dans l’univers	L’univers	
Go	19x19 cases, 3 valeurs possibles	

Combien de configurations possibles ?

Jeu	Taille du problème	Résultat
Tic-Tac-Toe “misère” v1	3x3 cases, 2 valeurs possibles	$2^9 = 512$
Tic-Tac-Toe classique	3x3 cases, 3 valeurs possibles	
Tic-Tac-Toe “misère” v2	6x3 cases, 2 valeurs possibles	
Puissance 4	7x6 cases, 3 valeurs possibles	
Echecs	8x8 cases, 7 valeurs possibles	
Nombre d’atomes dans l’univers	L’univers	
Go	19x19 cases, 3 valeurs possibles	

Combien de configurations possibles ?

Jeu	Taille du problème	Résultat
Tic-Tac-Toe “misère” v1	3x3 cases, 2 valeurs possibles	$2^9 = 512$
Tic-Tac-Toe classique	3x3 cases, 3 valeurs possibles	$3^9 = 19683$
Tic-Tac-Toe “misère” v2	6x3 cases, 2 valeurs possibles	
Puissance 4	7x6 cases, 3 valeurs possibles	
Echecs	8x8 cases, 7 valeurs possibles	
Nombre d’atomes dans l’univers	L’univers	
Go	19x19 cases, 3 valeurs possibles	

Combien de configurations possibles ?

Jeu	Taille du problème	Résultat
Tic-Tac-Toe “misère” v1	3x3 cases, 2 valeurs possibles	$2^9 = 512$
Tic-Tac-Toe classique	3x3 cases, 3 valeurs possibles	$3^9 = 19683$
Tic-Tac-Toe “misère” v2	6x3 cases, 2 valeurs possibles	$2^{18} = 262144$
Puissance 4	7x6 cases, 3 valeurs possibles	
Echecs	8x8 cases, 7 valeurs possibles	
Nombre d’atomes dans l’univers	L’univers	
Go	19x19 cases, 3 valeurs possibles	

Combien de configurations possibles ?

Jeu	Taille du problème	Résultat
Tic-Tac-Toe “misère” v1	3x3 cases, 2 valeurs possibles	$2^9 = 512$
Tic-Tac-Toe classique	3x3 cases, 3 valeurs possibles	$3^9 = 19683$
Tic-Tac-Toe “misère” v2	6x3 cases, 2 valeurs possibles	$2^{18} = 262144$
Puissance 4	7x6 cases, 3 valeurs possibles	$3^{42} = 1.0941899e+20$
Echecs	8x8 cases, 7 valeurs possibles	
Nombre d’atomes dans l’univers	L’univers	
Go	19x19 cases, 3 valeurs possibles	

Combien de configurations possibles ?

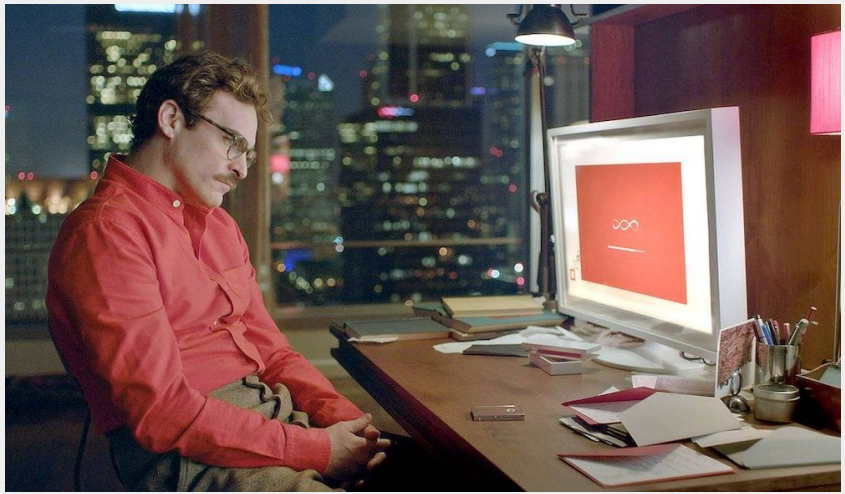
Jeu	Taille du problème	Résultat
Tic-Tac-Toe “misère” v1	3x3 cases, 2 valeurs possibles	$2^9 = 512$
Tic-Tac-Toe classique	3x3 cases, 3 valeurs possibles	$3^9 = 19683$
Tic-Tac-Toe “misère” v2	6x3 cases, 2 valeurs possibles	$2^{18} = 262144$
Puissance 4	7x6 cases, 3 valeurs possibles	$3^{42} = 1.0941899e+20$
Echecs	8x8 cases, 7 valeurs possibles	$7^{64} = 1.2197605e+54$
Nombre d’atomes dans l’univers	L’univers	
Go	19x19 cases, 3 valeurs possibles	

Combien de configurations possibles ?

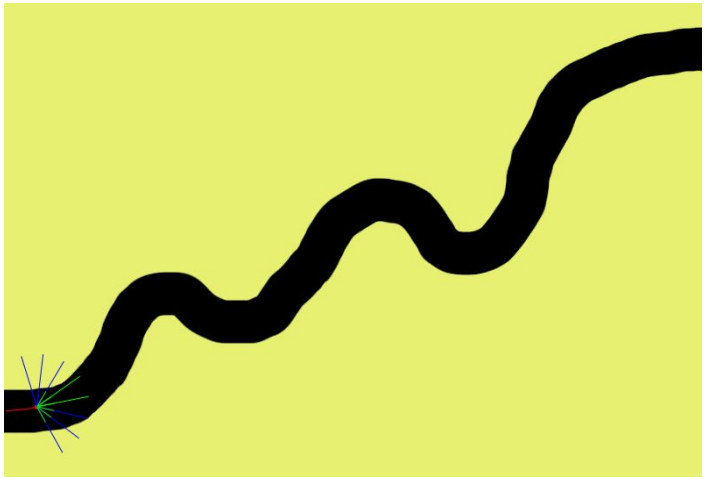
Jeu	Taille du problème	Résultat
Tic-Tac-Toe “misère” v1	3x3 cases, 2 valeurs possibles	$2^9 = 512$
Tic-Tac-Toe classique	3x3 cases, 3 valeurs possibles	$3^9 = 19683$
Tic-Tac-Toe “misère” v2	6x3 cases, 2 valeurs possibles	$2^{18} = 262144$
Puissance 4	7x6 cases, 3 valeurs possibles	$3^{42} = 1.0941899e+20$
Echecs	8x8 cases, 7 valeurs possibles	$7^{64} = 1.2197605e+54$
Nombre d’atomes dans l’univers	L’univers	~1e+80
Go	19x19 cases, 3 valeurs possibles	

Combien de configurations possibles ?

Jeu	Taille du problème	Résultat
Tic-Tac-Toe “misère” v1	3x3 cases, 2 valeurs possibles	$2^9 = 512$
Tic-Tac-Toe classique	3x3 cases, 3 valeurs possibles	$3^9 = 19683$
Tic-Tac-Toe “misère” v2	6x3 cases, 2 valeurs possibles	$2^{18} = 262144$
Puissance 4	7x6 cases, 3 valeurs possibles	$3^{42} = 1.0941899e+20$
Echecs	8x8 cases, 7 valeurs possibles	$7^{64} = 1.2197605e+54$
Nombre d’atomes dans l’univers	L’univers	~1e+80
Go	19x19 cases, 3 valeurs possibles	$3^{361} = 1.740897e+172$



Voiture autonome

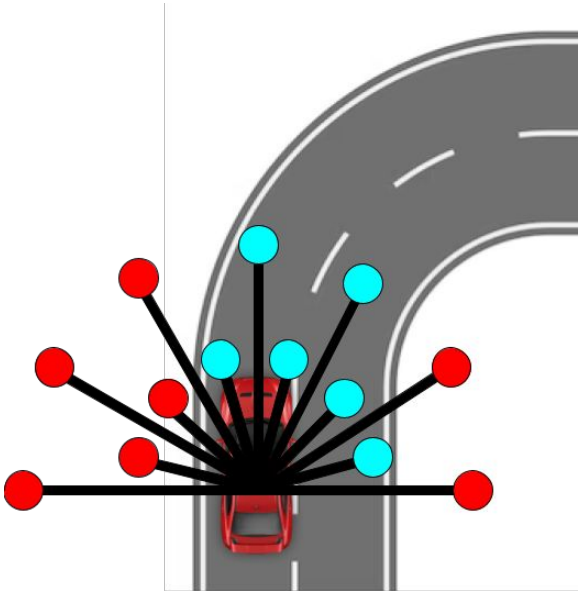


Conduite manuelle
https://openprocessing.org/sketch/2521815

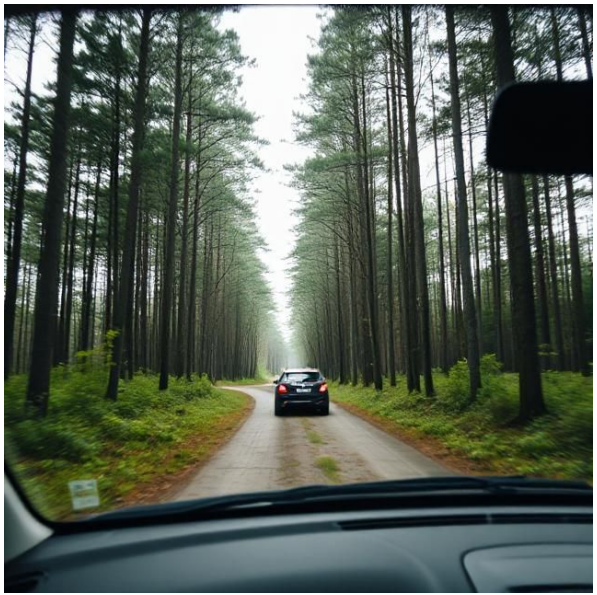
- Vitesse constante
- Score = distance parcourue avant de sortir de la route
- 3 actions possibles : avancer tout droit, tourner vers la droite ou vers la gauche

- Apprentissage : https://openprocessing.org/sketch/2522130
- Vidéo : https://mediaserver.unilim.fr/videos/06022025-171217/

Voiture autonome

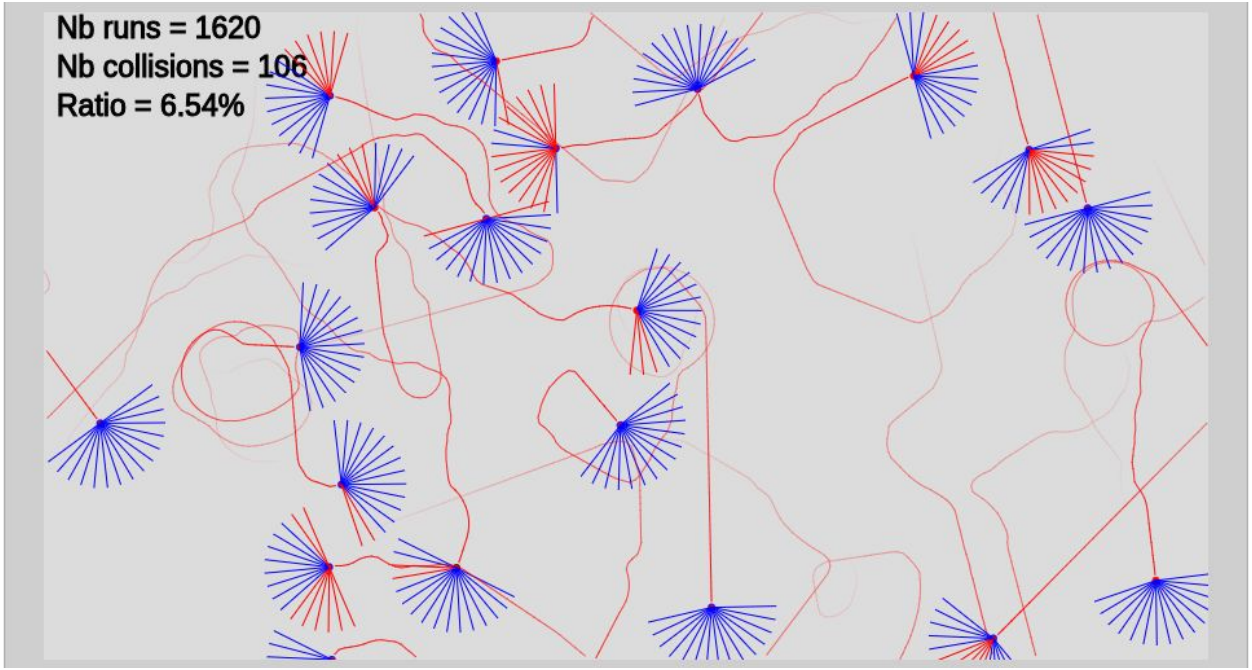


16 capteurs => Nombre de configurations possibles ?



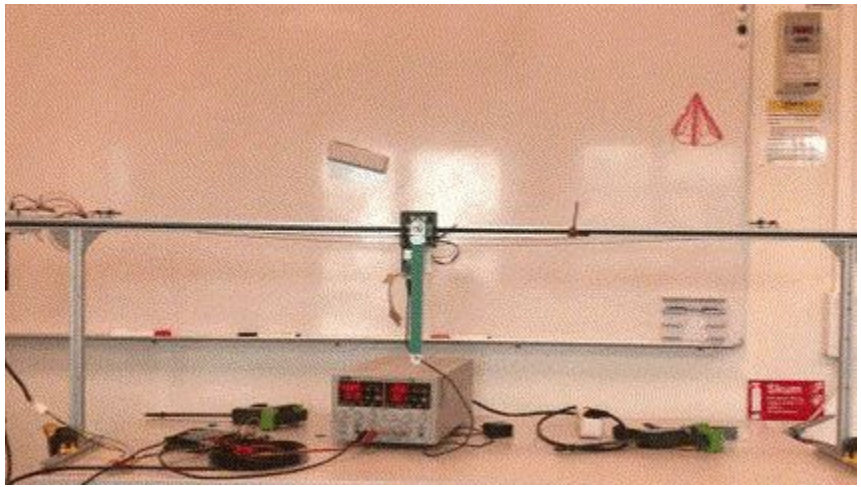
En pratique: [MIT Autonomous Vehicle: Learning Robust Sim-to-Real Control Policies]

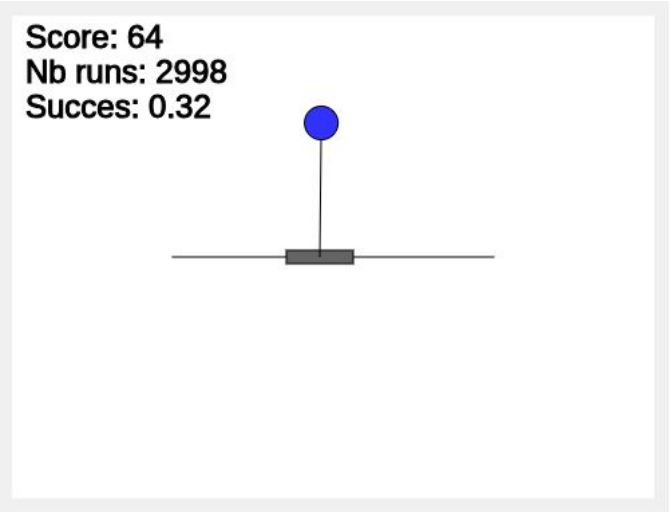
Voitures autonomes



https://openprocessing.org/sketch/2536996
https://mediaserver.unilim.fr/videos/08022025-122136/

Le pendule inversé

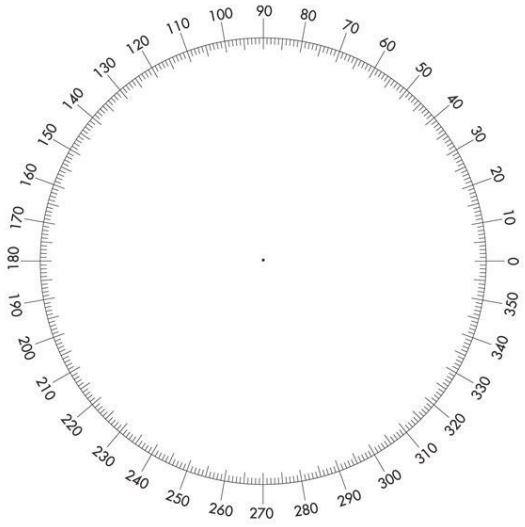




https://openprocessing.org/sketch/2525186

Vidéo:
https://mediaserver.unilim.fr/videos/15032025-183057/

Le pendule inversé : échantillonnage



360 valeurs d’angle

*

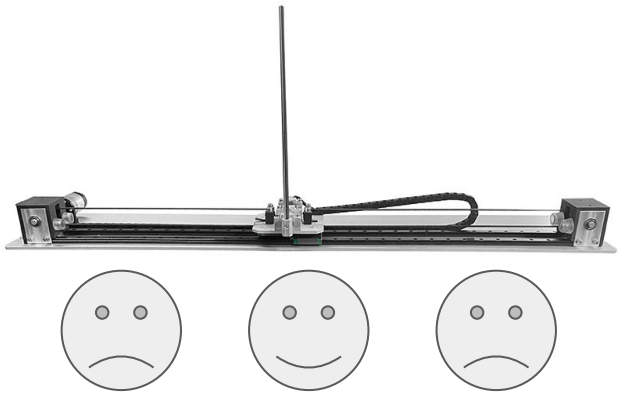
1000 valeurs de vitesse

*

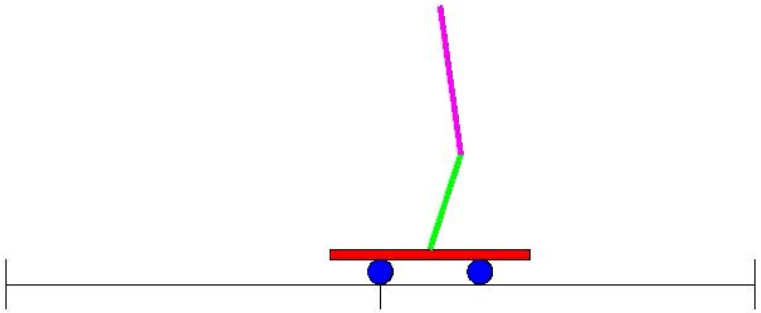
3 valeurs de “région”

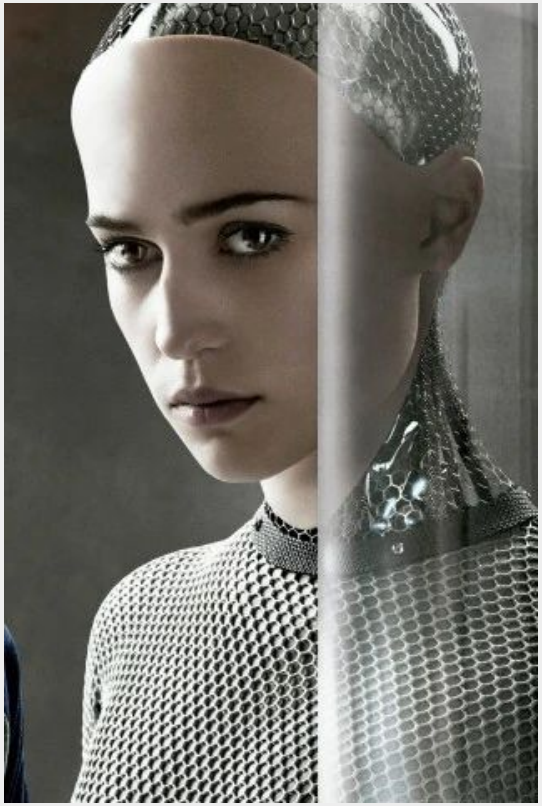
=

1,080,000
configurations possibles



Double pendule, triple pendule, …

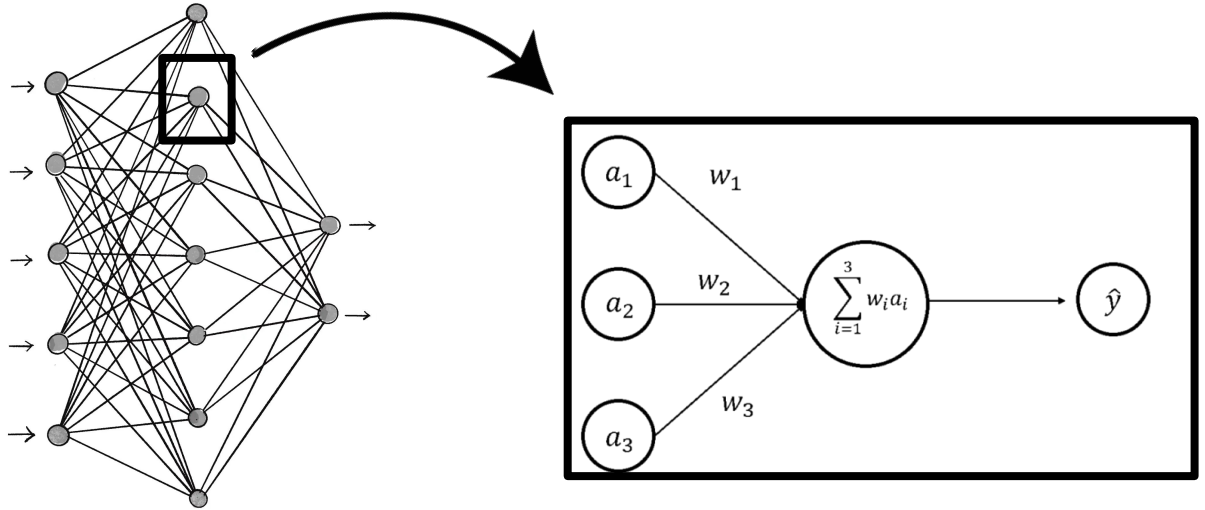




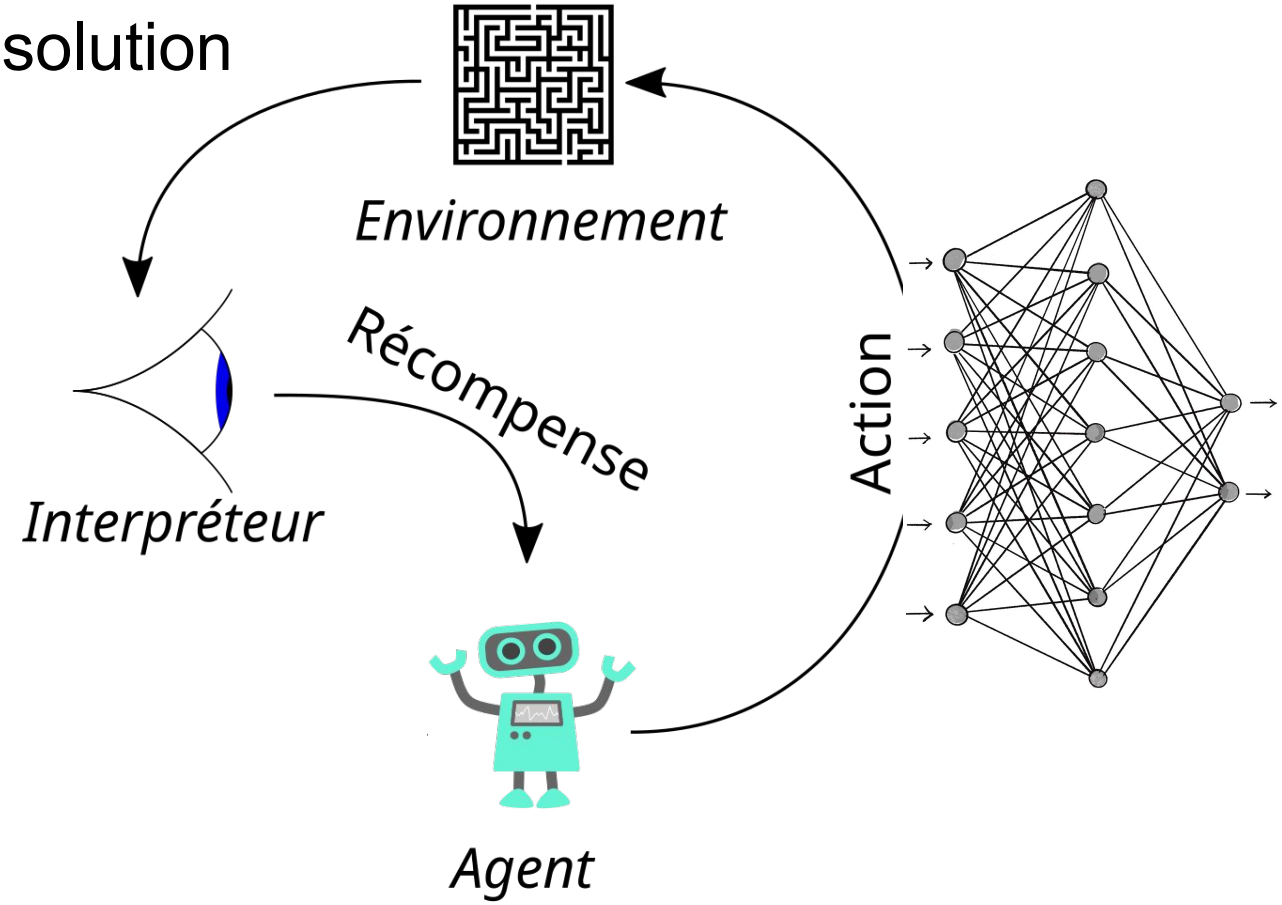
Deep reinforcement learning

Pour éviter d'avoir à stocker toutes les probabilités pour toutes les actions possibles, on peut utiliser un **réseau de neurones** (*"deep" learning*)

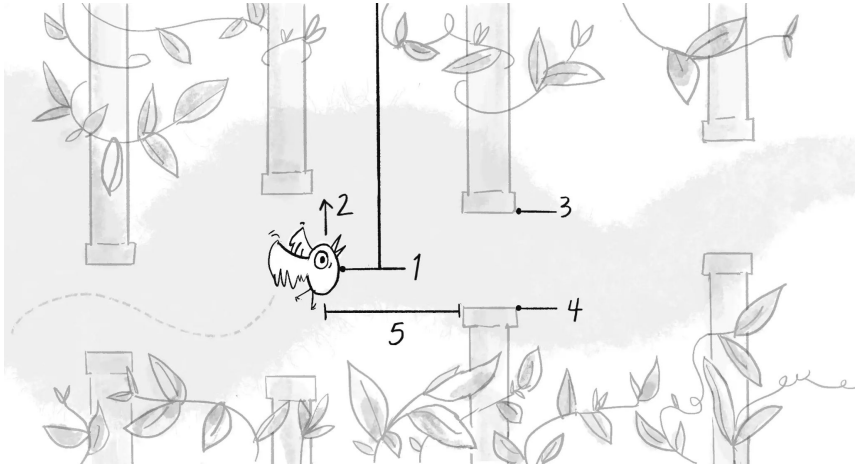
Le but est maintenant de faire "apprendre" à l'ordinateur les valeurs optimales des (très nombreux) paramètres **w1, w2, …** du réseau



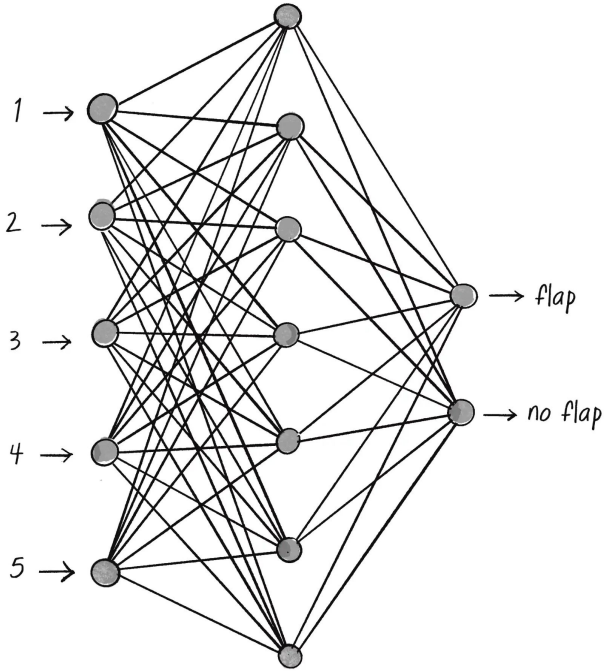
Méthode de résolution

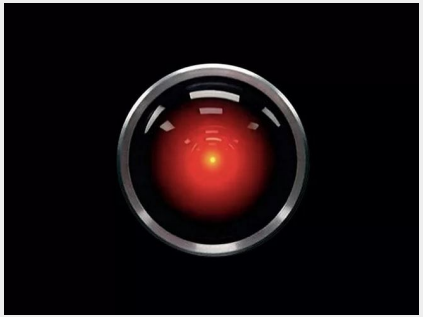


Flappy Bird

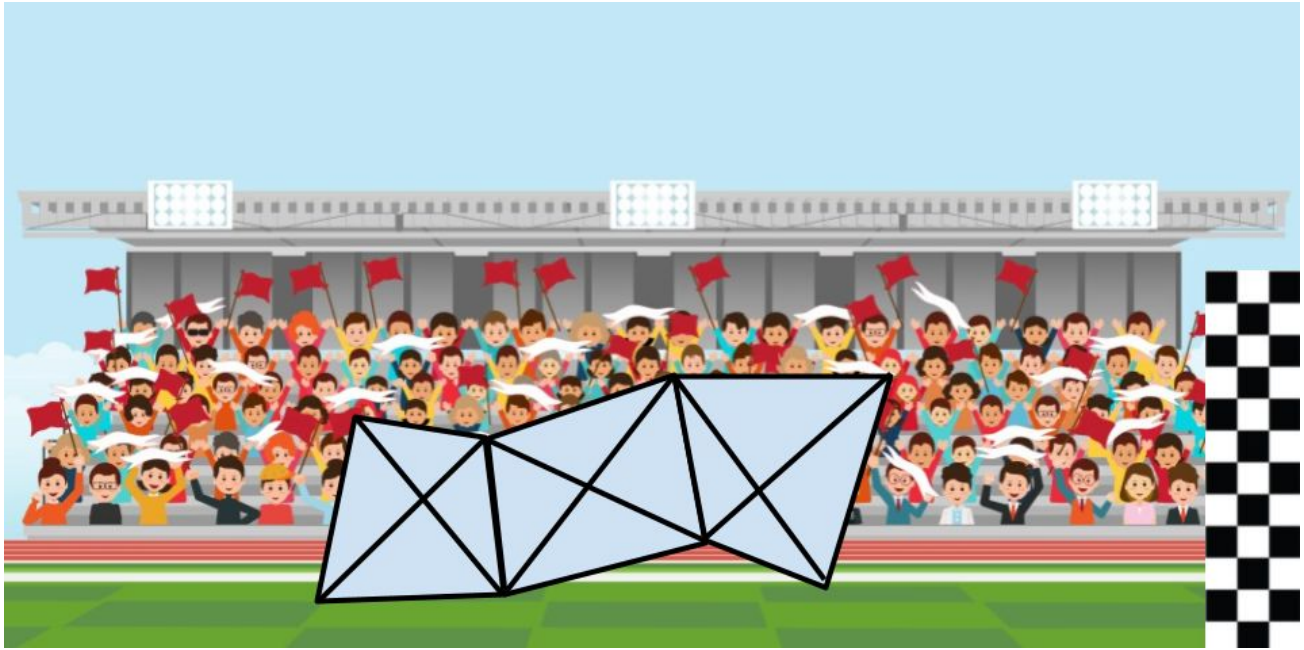


https://natureofcode.com/neuroevolution/





Bouncing Run

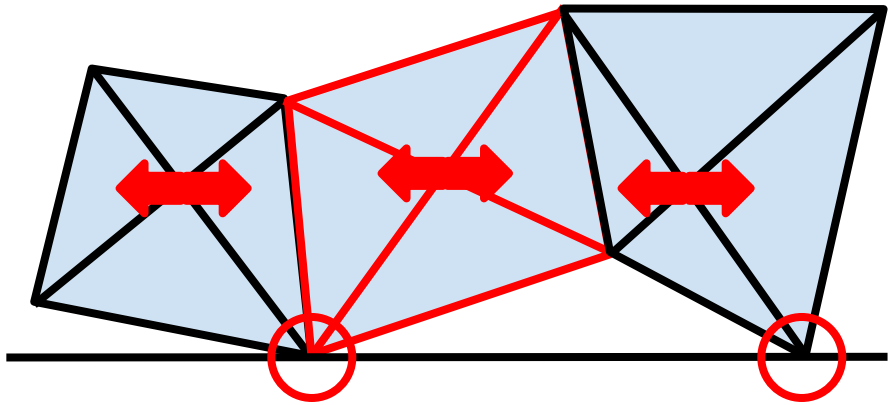


https://openprocessing.org/sketch/2537161

Bouncing Run

- 10 valeurs d'entrée (0 ou 1) :
  - Pour chaque boîte :
    - Ressorts en extension ?
    - Va vers la gauche ou la droite ?
  - Pour chaque sommet inférieur :
    - Touche le sol ?

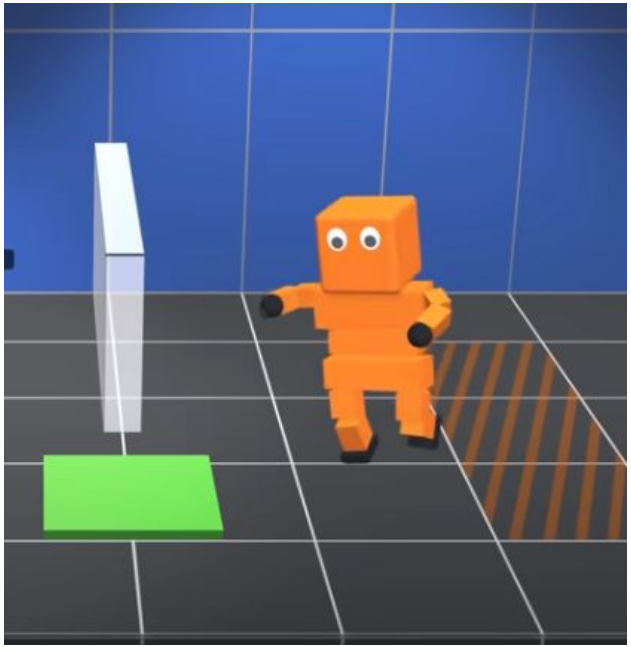
- 4 actions possibles :
  - Étirer les ressorts de la boîte de gauche
  - Étirer les ressorts de la boîte du centre
  - Étirer les ressorts de la boîte de droite
  - Ne rien faire



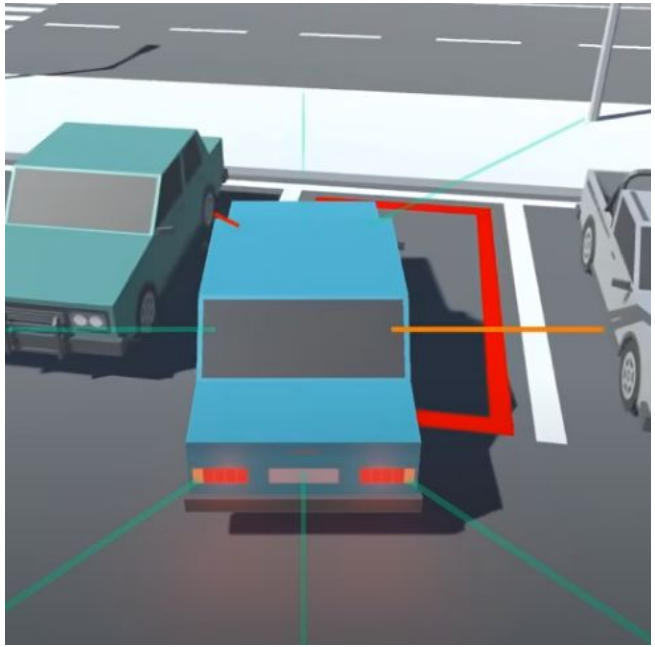
https://openprocessing.org/sketch/2537164

Vidéo:
https://mediaserver.unilim.fr/videos/16032025-174315/

Applications : systèmes mécaniques/robotiques

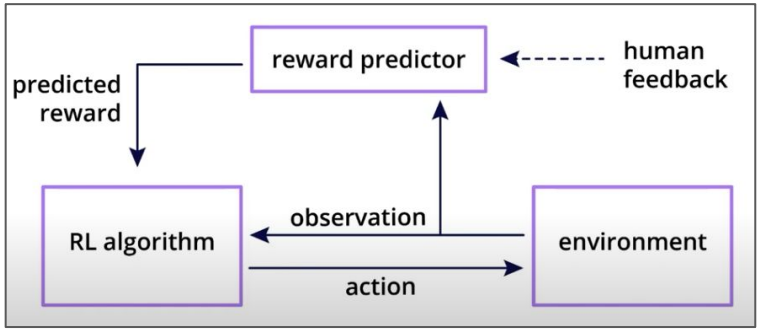


AI Learns to Walk



AI Learns to Park

Applications : feedback humain pour les IA génératives



"portrait of grandpa with hands folded"

Midjourney v2

Midjourney v5.2

As-tu un message pour les spectateurs de la conférence "L'apprentissage par renforcement" ?

"Bienvenue à cette conférence sur l'apprentissage par renforcement ! 🚀 Cette branche fascinante de l'IA permet aux machines d'apprendre par essais et erreurs, comme un joueur qui maîtrise un jeu ou un robot qui découvre son environnement. J'espère que cette exploration vous inspirera et éveillera votre curiosité sur le potentiel de ces algorithmes !"

Applications : multiplier 2 matrices de façon optimale

V. Strassen, 1969

nature

Explore content | About the journal | Publish

nature > articles > article

Article | Open access | Published: 05 October 2022

Discovering faster matrix multiplication algorithms with reinforcement learning

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hube
Mohammadamin Barekatain, Alexander Novikov, Francisc
Swirszcz, David Silver, Demis Hassabis & Pushmeet Kohli

Nature **610**, 47–53 (2022) | Cite this article

Google DeepMind

arXiv > cs > arXiv:2210.04045

Search... Help | Ad

Computer Science > Symbolic Computation

[Submitted on 8 Oct 2022 (v1), last revised 13 Oct 2022 (this version, v3)]

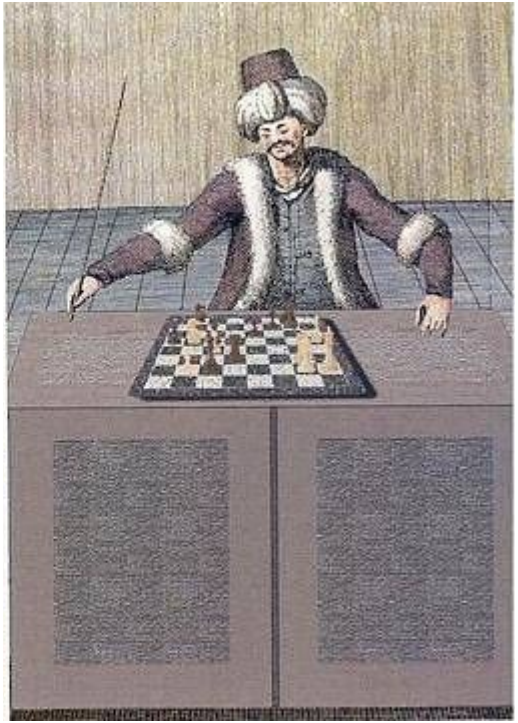
The FBHHRBNRSSSHK-Algorithm for Multiplication in $\mathbb{Z}_2^{5\times5}$ is still not the end of the story

Manuel Kauers, Jakob Moosbauer

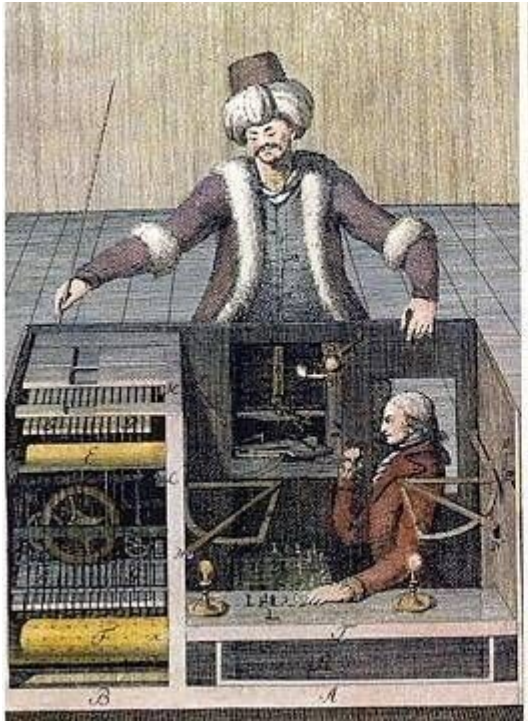
In response to a recent Nature article which announced an algorithm for multiplying 5×5-matrices over $\mathbb{Z}_2$ with only 96 multiplications, two fewer than the previous record, we present an algorithm that does the job with only 95 multiplications.

Subjects: **Symbolic Computation (cs.SC)**; Computational Complexity (cs.CC)
Cite as: arXiv:2210.04045 **[cs.SC]**
(or arXiv:2210.04045v3 **[cs.SC]** for this version)
https://doi.org/10.48550/arXiv.2210.04045

En conclusion…



En conclusion…



Filmographie

- War Games (1983)
- Interstellar (2014)
- Terminator (1984)
- Her (2013)
- Blade Runner (1982)
- Ex Machina (2014)
- 2001, l'Odyssée de l'espace (1968)
- Le Château dans le ciel (1986)