JEU de NIM

(application des graphes et découverte de l’IA)

* Jouer au jeu de Nim avec 12 allumettes ou feutres à 2 :

On prend la version celui qui prend le dernier perd, essayer de trouver une stratégie gagnante. Expliquer la stratégie gagnante



* Jeu de nim et machine learning :

Rechercher la définition de Machine Learning (une branche de l’IA) et comprendre la différence avec Deep Learning

Comment apprendre à l’ordinateur à jouer au jeu de Nim et à gagner, visionner la vidéo suivante :

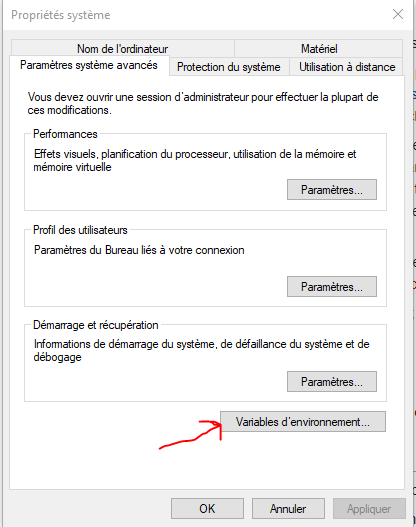
<https://youtu.be/q8-LhLKc8wY>

Reproduire en classe cette activité d’apprentissage en mode débranchée (sans ordinateur !)

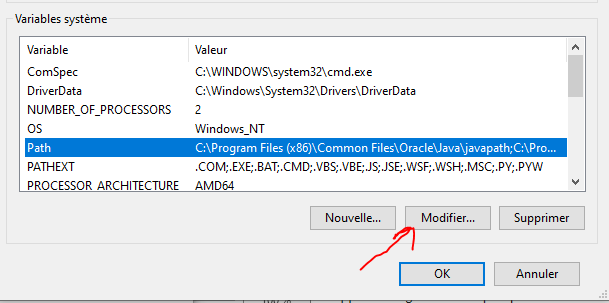
* Création du graphe de toutes les possibilités de jeu

Installation de Graphviz un outil de visualisation de graphes programmable avec Python :

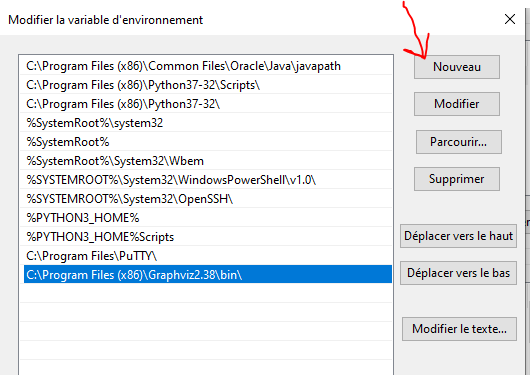
Site pour télécharger Graphviz-2.38 : <https://www.npackd.org/p/org.graphviz.Graphviz/2.38>

Ajouter un chemin au PATH pour que l’ordinateur trouve bien Graphviz depuis Python :

1. Ouvrir les propriétés système :
2. Dans variables environnement :

Sélectionner Path pour Modifier

1. Ajouter nouveau chemin :



Voici ce que vous devez y copier :

C:\Program Files (x86)\Graphviz2.38\bin\

Puis Ok, Ok, Ok

1. Redémarrer l’ordinateur et vous pouvez utiliser Graphviz

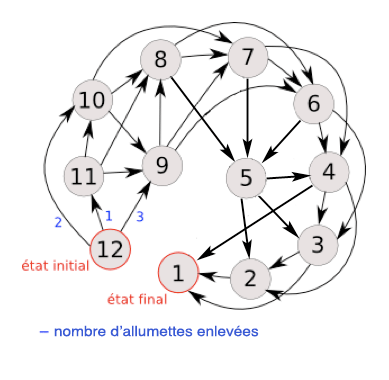
* Graphe associé au jeu de Nim :

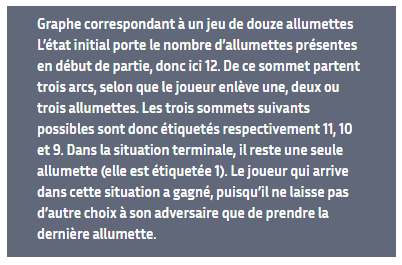
Dans la grande majorité des cas, l’ensemble des situations d’un jeu, même s’il est très grand, est fini. Il est alors possible de constituer un graphe de leur enchaînement, traduisant l’ensemble des coups possibles à jouer.

Un graphe est une structure mathématique définie par un ensemble de sommets et un ensemble d’arcs orientés reliant deux sommets. **Dans le graphe associé à un jeu de Nim, les sommets représentent les situations et les arcs sortant d’un sommet et entrant dans un autre représentent les coups qui font passer d’une situation à une autre.**

Partant d’une situation initiale, représentée dans le graphe par un sommet sans arc entrant, le jeu évolue vers une situation finale représentée par un sommet sans arc sortant, un « puits » dans le langage de la théorie des graphes.

Dans l’exemple du jeu des allumettes, **chaque sommet est renseigné par le nombre d’allumettes** qui restent dans cette situation et **chaque arc est renseigné par le nombre d’allumettes retirées** lors du coup correspondant. Voici par exemple le graphe correspondant à un jeu de douze allumettes.

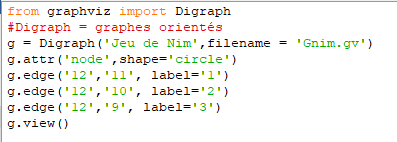




Nous allons créer ce graphe avec le module graphviz de Python :

La doc de graphviz est à consulter à l’adresse : <https://graphviz.readthedocs.io/en/stable/manual.html>

Compléter ce programme pour qu’il affiche le graphe ci-dessus :



* Partie A :Créer le jeu sans IA : 2 joueurs humains :

1. Programmez d’abord les classes dont nous aurons besoin, voici leurs interfaces sous forme de diagramme de classe :

|  |
| --- |
| **GraphOriente** |
| **liste\_sommets** : type list, initialisé vide  **liste\_arcs** : type list initialisé vide elle contiendra des tuples de sommets |
| **ajouter\_sommet(A)** : ajoute le sommet A à la liste\_sommets  **ajouter\_arc(A,B)** : ajoute le couple (A,B) à liste\_arcs  **supprimer\_arc(A,B)** : suprrime le couple (A,B) de liste\_arcs  **liste\_sommets\_issus(A**) : renvoie une liste de couples de sommets commençant par A. |

|  |
| --- |
| **Joueur** |
| **pseudo** : string  **joue** : bool (True si c’est le tour de Joueur et Fasle sinon) initialisé à False  **gagne**: bool (True si le joueur a gané et False sinon initialisé à False  **score**: int initialisé à 0  **etat**: string ‘humain’ ou ‘ordi’ initialisé vide |
| **augmenter\_score() :** augmente de 1 le score quand le joueur gagne  **change\_joueur()** : change l’attribut joue quand on change de joueur  **change\_gagnant()** : change l’attribut gagne à True lorsque le joueur gagne et False quand on refait une autre partie  **set\_etat(valeur)** : affecte à l’attribut etat valeur qui peut être ‘humain’ ou ‘ordi’ |

La classe Joueur sera enregistrée dans un fichier joueur.py et la classe GrapheOriente sera enregistrée dans un fichier graphe\_oriente.py

Charger le fichier : jeu\_nim2humains.py

1. Compléter la zone de gestion des clics des joueurs sur les différents boutons. Tester le jeu.
2. Compléter la partie « fin de jeu » quand le nb d’allumettes vaut 1.
3. Compléter la gestion du gagnant et du score quand c’est player2 qui gagne.

* Partie B : Créer le jeu humain VS ordi avec un ordi qui joue au hasard sans aucun apprentissage !

Charger le fichier : jeu\_nim\_humainVSordi.py

1. Remettre les parties complétées des questions 2) 3) et 4) dans la partie *if humainVShumain == True*
2. Créer le graphe de tous les coups possibles que l’ordinateur utilisera pour choisir son coup.
3. Compléter la zone où c’est au tour du player 2 de choisir son coup en utilisant le graphe de tous les coups

* Partie C : Créer le jeu humain VS ordi en permettant à l’ordi d’apprendre de ses erreurs (machine learning) :

1. Il faut gérer l’apprentissage lorsque c’est le player 1 qui gagne, on doit supprimer le dernier coup du graphe des coups pour que l’ordi ne reproduise plus cette erreur.
2. Il reste une dernière chose à gérer, quand l’ordi choisit son coup mais que la liste des coups possibles est vide (gobelet vide dans vidéo) le jeu doit s’arrêter car c’est le player 1 qui gagne et on doit aussi retirer le dernier coup qui amène à cette situation.

Tester votre IA apprenante, en combien de partie n’arrivez-vous plus à gagner ?

1. Ce jeu comporte bien sûr des défauts (mineurs) qui n’empêche pas de jouer, à vous d’améliorer ce jeu et me le rendre !

Ce long TP (3 semaines de préparation de mon côté) a été inspiré de cet article sur le site ci-dessous ainsi que d’un stage avec un TP sur l’IA présenté par Mme Marie Duflot Kremer (merci à elle pour m’avoir fait comprendre ce que c’est d’apprendre : EVITER de REPRODUIRE les ERREURS ! à méditer )

sitographie : [https://interstices.info/jeux-de-nim/#:~:text=Dans%20le%20graphe%20associ%C3%A9%20%C3%A0,une%20situation%20%C3%A0%20une%20autre](https://interstices.info/jeux-de-nim/%23:~:text=Dans%20le%20graphe%20associ%C3%A9%20%C3%A0,une%20situation%20%C3%A0%20une%20autre).