

FICHE PROFESSEUR

- Niveau : 2^{nde} (SNT), durée : 1h30

- Objectifs pédagogiques :

Travailler le traitement de données numériques à l'aide d'un tableur.

Travailler la lecture des données dans un tableur.

- La situation-problème :

On présente le jeu « Une course stratégique ». On cherche à élaborer une stratégie pour gagner.

- Matériel

- Un plateau de jeu
- Un jeu de 52 cartes
- Un pion

- Les règles du jeu :

- On joue à deux (joueur A, joueur B).
- Chaque joueur dispose des quatre cartes As, 2, 3 et 4 d'une même couleur.
- Le pion est placé sur la case « Départ ».
- À tour de rôle (en commençant par le joueur A), chaque joueur pose une carte et avance le pion du nombre de cases indiqué sur sa carte.
- Le joueur gagnant est celui qui arrive le premier à faire franchir la ligne d'arrivée au pion.

- Pré-requis :

Python : fonctions et boucle for

Notion de données structurées, format .csv

Tableur : importer des données au format .csv

- Déroulement :

Partie A : À la main

En binômes, les élèves jouent quelques parties et essaient de trouver une stratégie gagnante.

Partie B : Avec une « IA »

→ Générer les données

On définit la fonction `jouer()` qui permet de simuler aléatoirement une partie (une liste avec le joueur gagnant et la suite des cartes posées).

```
from random import randint

def jouer():
    A=[1,2,3,4] #cartes du joueur A
    B=[1,2,3,4] #cartes du joueur B
    case=0      #position initiale du pion sur la case 0
    liste=""    #on va mémoriser dans une liste les cartes posées par chaque joueur
    (feuille de match)

    while case<=10: #On joue tant que la ligne d'arrivée n'est pas franchie
        h=randint(0,len(A)-1) #A choisit au hasard une carte restante de son jeu
        liste.append(A[h])    #Je mémorise dans ma liste la carte que A vient de
    jouer
        if case+A[h]>10: #Si le pion avancée par A dépasse la ligne d'arrivée
    alors ...
```

UNE COURSE STRATEGIQUE

```
liste[0]="A"      #A gagne et au début de la liste, je précise le nom du
gagnant, ici A
return liste     #Je renvoie la partie jouée et mémorisée
else:           #sinon :
    case=case+A[h] #A avance le pion du nombre de cases indiqué sur la
cartejouée
    A.remove(A[h]) #Dans les deux cas, A retire de son jeu la carte qu'il
vientde jouer

    h=randint(0,len(B)-1)
    liste.append(B[h])
    if case+B[h]>10:
        liste[0]="B"
        return liste
    else:
        case=case+B[h]
    B.remove(B[h])
```

On simule avec Python un grand nombre de parties (5000) pour essayer d'obtenir toutes les parties possibles.

Pour cela, on peut utiliser le notebook Capytale : <https://capytale2.ac-paris.fr/web/c/e9c9-611613>

On peut ensuite copier les résultats obtenus et les coller dans un bloc-note. On enregistre ce fichier au format .csv pour pouvoir l'importer dans un tableur.

→ Organiser les données

On supprime les doublons et on trie les données suivant l'ordre des cartes jouées. On obtient alors les 264 parties possibles (si ce n'est pas le cas, il faut simuler d'autres parties et les ajouter au fichier).

La marche à suivre est détaillée dans la fiche élève.

→ Établir une stratégie

On cherche à « fabriquer une IA ». On veut établir, à l'aide du tableur, un enchaînement de cartes à jouer pour gagner à coup sûr. Pour cela, on peut colorier les cellules du tableur pour choisir les cartes à jouer.

Une méthode est détaillée dans la vidéo suivante :

<https://youtu.be/Yg07TJr3LOU>.

On propose de déterminer une stratégie gagnante pour le joueur A (celui qui joue en premier).

● Analyse du dispositif :

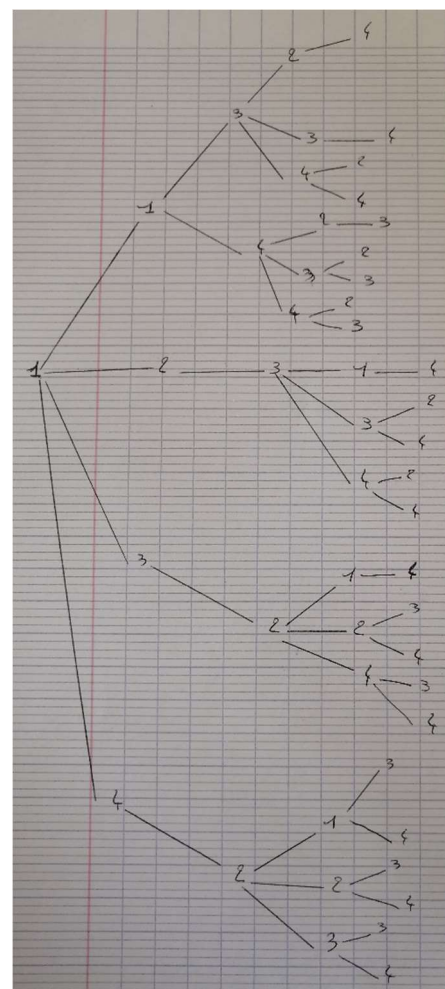
Les élèves ont apprécié le jeu.

Dans la partie B, les élèves ont réussi sans grande difficulté à générer les données et à les organiser dans le tableur en suivant les instructions détaillées sur la fiche élève.

Il est très difficile pour les élèves d'élaborer seuls une stratégie à l'aide des données représentées dans le tableur. Il est utile de proposer une méthode pour les analyser.

La vidéo <https://youtu.be/Yg07TJr3LOU> peut aider les élèves. Pour certains, il a été nécessaire de redonner des explications. Les élèves ont réussi à utiliser le tableur ou un arbre pour gagner à coup sûr (dans le cas du joueur A).

La stratégie optimale pour le joueur B est beaucoup plus fastidieuse à établir.



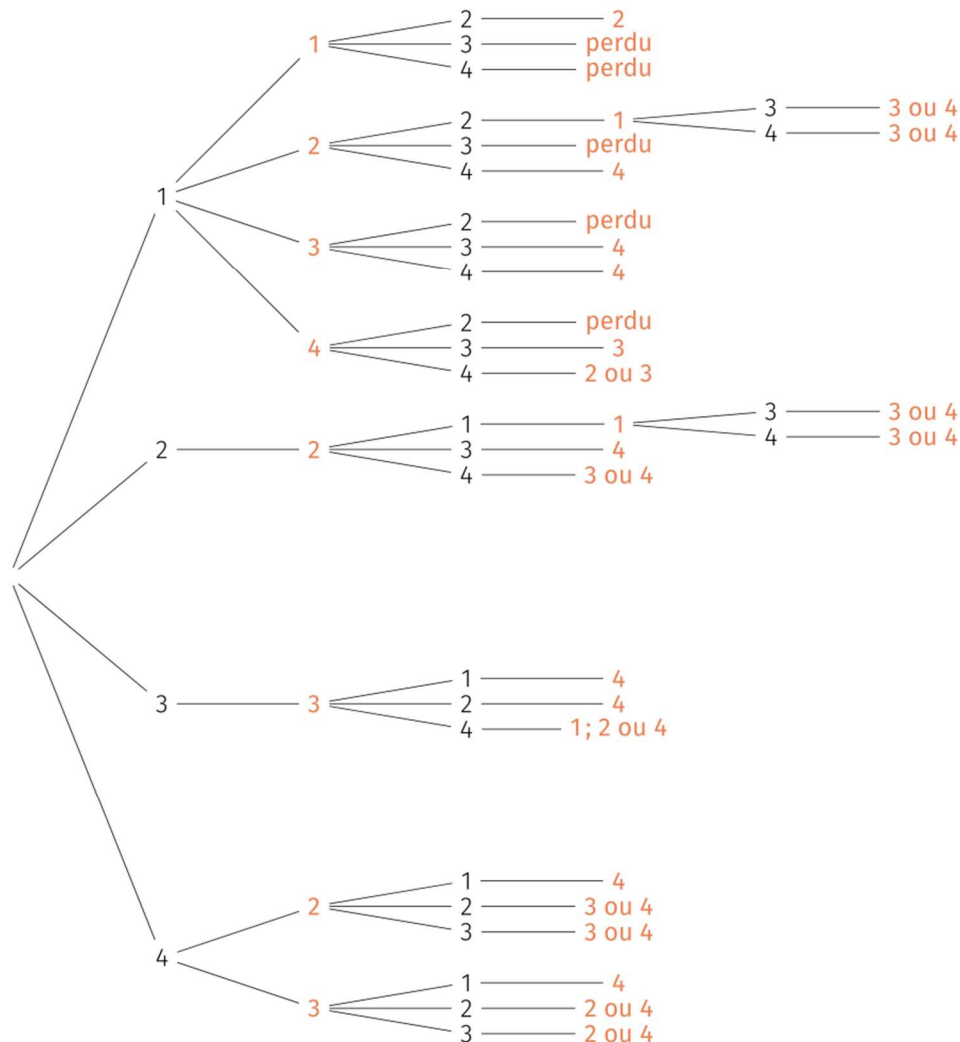
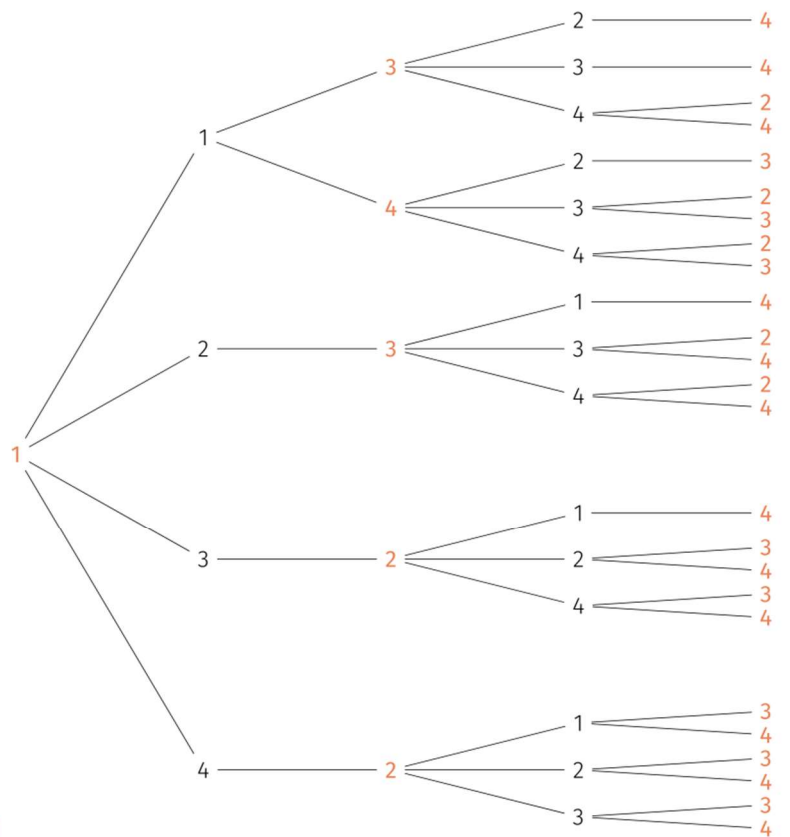
UNE COURSE STRATEGIQUE

• Des stratégies gagnantes :

L'IA joue en premier :

Il existe une stratégie gagnante que l'on peut représenter à l'aide de l'arbre suivant :

Les cartes jouées par l'IA sont représentées en **orange**.



L'IA joue en deuxième

Il n'existe pas de stratégie gagnante à coup sûr. On peut cependant optimiser ses cartes en s'aidant de l'arbre ci-contre.

Les cartes jouées par l'IA sont représentées en **orange**.

UNE COURSE STRATEGIQUE

- **Approfondissement ou prolongement possibles :**

- Proposer d'établir une stratégie optimale pour le joueur B.
- Modifier le nombre de joueurs et/ou le nombre de cartes et/ou la position de la ligne d'arrivée et étudier une stratégie.

- **Dans les programmes du cycle :**

En SNT

Contenus	Capacités attendues
Traitement de données structurées	Réaliser des opérations de recherche, filtre, tri ou calcul sur une ou plusieurs tables.

- **Les six compétences majeures**

Compétences pour le lycée.

- Chercher Domaines du socle : 2, 4

- Extraire, organiser et traiter l'information utile.

- Modéliser Domaines du socle : 1, 2, 4

- Utiliser, comprendre, élaborer une simulation numérique ou géométrique prenant appui sur la modélisation et utilisant un logiciel.

- Raisonner Domaines du socle : 2, 3, 4

- Utiliser différents types de raisonnement (par analyse et synthèse, par équivalence, par disjonction de cas, par l'absurde, par contraposée, par récurrence...).