

## TD/TP 01 - Décisions sous incertitude

### Prise en main de Zombie Dice

IMT Lille Douai – UV-MAD

Nom(s) et prénom(s) : \_\_\_\_\_

## 1 Prise en main et compréhension

Comprendre le jeu *Zombie Dice* et son implémentation :

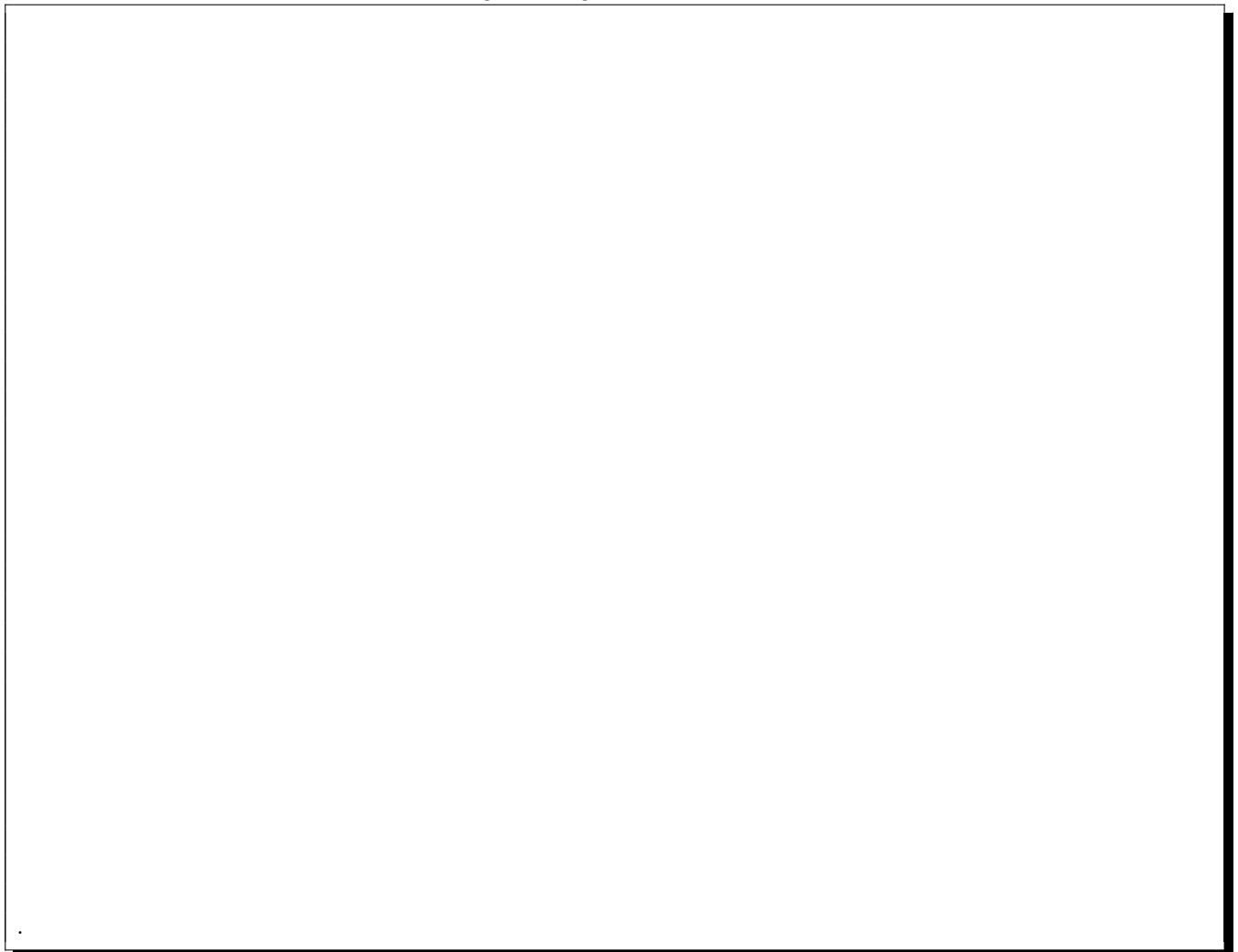
- ☐ Se connecter et cloner <https://bitbucket.org/Num-ILD/pyzombie>
- ☐ Regarder le code et faire quelques parties pour comprendre les règles.
- ☐ Représenter schématiquement les classes du projet sous forme d'un diagramme de classes.

[https://fr.wikipedia.org/wiki/Diagramme\\_de\\_classes](https://fr.wikipedia.org/wiki/Diagramme_de_classes)

[https://fr.wikipedia.org/wiki/UML\\_%28informatique%29](https://fr.wikipedia.org/wiki/UML_%28informatique%29)

L'idée consiste à dessiner une boîte pour chaque classe dans le programme avec son nom et les méthodes principales puis de lier les classes entre elles, s'il existe des connexions (typiquement : un attribut de la classe A et une instance de la classe B).

Listing 1 – Diagramme de classes



## 2 Première IA

Écrire une première intelligence artificielle (IA) qui décrit des règles d'action sous forme d'un Script simple. Dans un premier temps l'IA se contentera de toujours jouer puis développera un comportement plus complexe composé d'une cascade de *if..then..else* :

- Dans un nouveau fichier *Python*, créer une nouvelle classe *ScriptIA* héritant de *Player*.
- Implémenter la méthode *act* de façon à ce qu'elle retourne toujours la chaîne de caractère "play".
- Modifier la fonction *main* de façon à ce qu'un *Player ScriptIA* remplace un des joueurs. Lancer le jeu.
- Modifier votre IA avec un script simple pour qu'elle ait un comportement un minimum cohérent. Récupérer les informations nécessaires via les méthodes d'accès aux variables *game* passée en paramètre.

## 3 L'arbre de décision

Il est possible de représenter des règles d'action sous la forme d'un arbre de décision. Un arbre de décision définit une structure permettant de descendre jusqu'à l'action qui sera choisie par l'agent. La succession définit les tests sur les valeurs des variables de l'état.

[https://fr.wikipedia.org/wiki/Arbre\\_de\\_d%C3%A9cision](https://fr.wikipedia.org/wiki/Arbre_de_d%C3%A9cision)

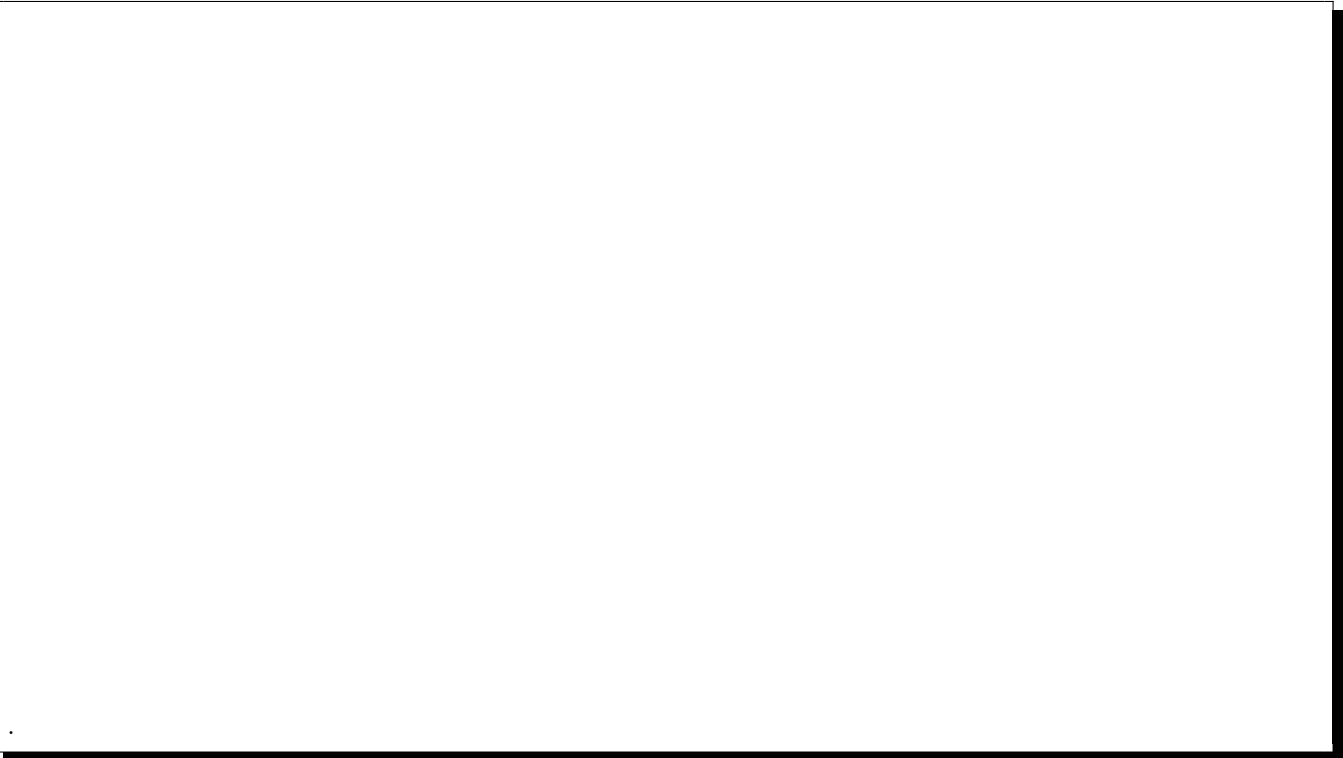
Les feuilles de l'arbre (nœuds sans descendant) représentent une décision (ici "play" ou "score"). Les autres nœuds reprennent une variable et orientent la descente dans une branche de l'arbre en fonction de la valeur de cette variable.

- Quel est l'arbre de décision correspondant au script simple suivant :

Listing 2 – Script 2 variables

```
if game.brain() > 0 :
    if game.shot() == 0 :
        return "play"
    else :
        return "score"
else :
    return "play"
```

Listing 3 – Arbre de Decision



- Cette IA simple repose sur deux variables d'état : le nombre de cerveaux mangés et le nombre de balles prises. En considérant qu'il est possible de manger au maximum 13 cerveaux et de prendre 3 balles, l'IA est définie par

un maximum de combien d'états (les états non inatteignables sont tout de même comptabilisés dans le nombre d'états) ?

## 4 Des arbres de décision plus complets

Un état de l'agent joueur représente une affectation complète de toutes les variables décrivant l'agent dans son environnement à un instant  $t$ . Si on rapporte la notion d'états aux arbres de décision, le nombre d'états correspond au nombre maximum de feuilles que peut contenir l'arbre.

L'arbre de décision permet de factoriser les choix d'action en regroupant ensemble les états qui conduisent à la même décision pour une même conjonction de valeurs variables. Dans *ZombieDice* par exemple, la variable *brain* qui comptabilise le nombre de cerveaux mangés est prédominante. La variable *brain* égale à 0 va systématiquement impliquer de jouer (pourquoi scorer un score nul?).

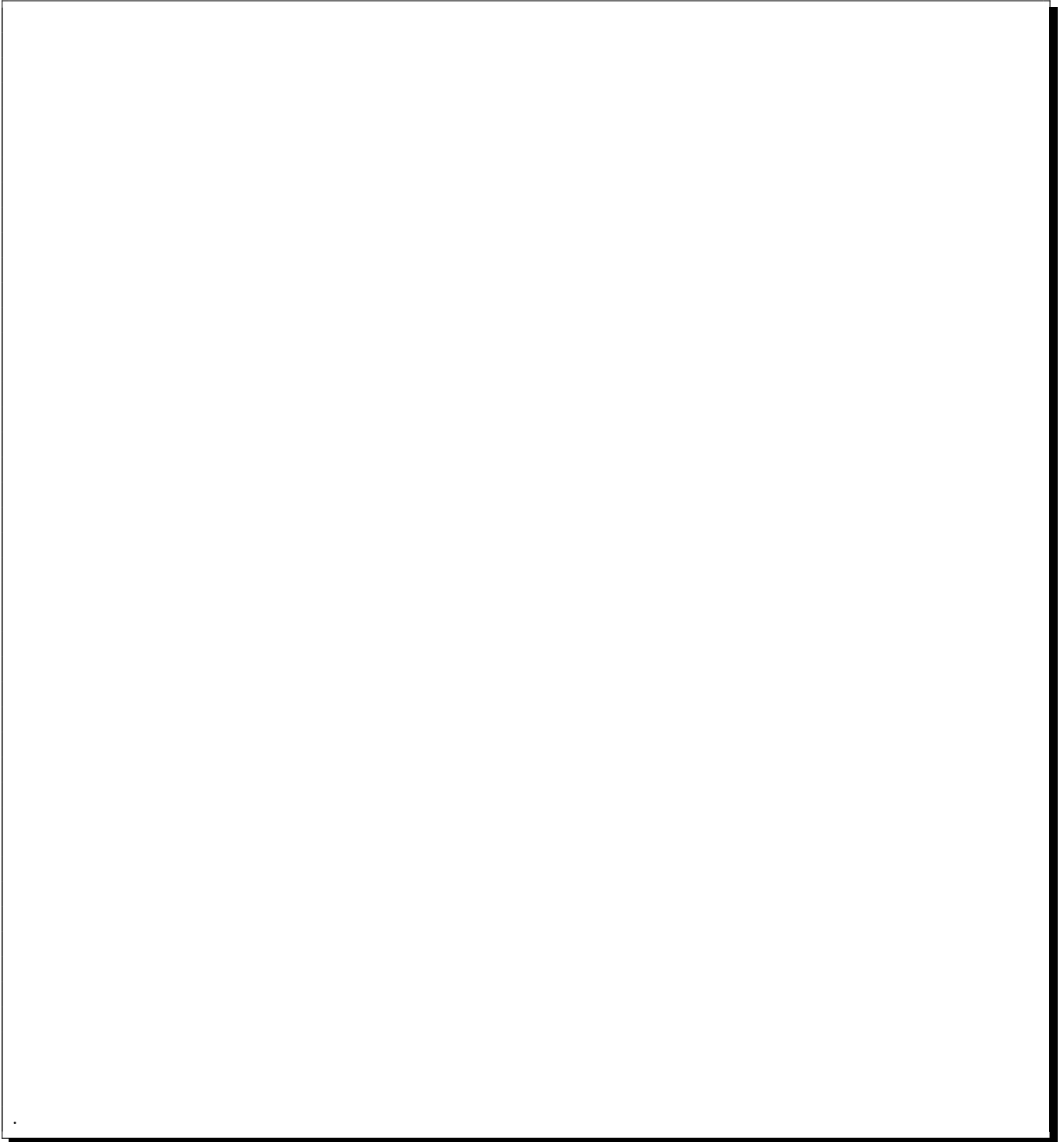
Poser les bases pour permettre la génération d'IA consiste, dans un premier temps, à bien identifier ce qu'est l'état du système à contrôler et donc l'ensemble des variables et les domaines de variation (ensemble des valeurs que peut prendre chaque variable).

- Lister l'ensemble des variables que vous identifiez appartenir à l'état du jeu, permettant à un joueur de prendre une décision. Établir les domaines de variation.

- Sans tenir compte des dépendances qui existent entre ces variables, l'agent joueur est défini par un maximum de combien d'états ? (Donner le calcul et le résultat.)

- Proposer un arbre de décision sur un minimum de 3 niveaux.

## Listing 4 – Votre Arbre de Decision



- ☐ Implémenter l'arbre précédent sur un nouveau PNJ (*TreeIA*).

## 5 Tournoi (optionnel)

Dans la mesure où plusieurs IA heuristiques sont proposées. Le plus simple pour les évaluer consiste à les faire s'affronter (évaluation empirique).

- ☐ Implémenter la nouvelle IA que vous avez proposée.
- ☐ Modifier le main de façon à faire jouer les deux IAs l'une contre l'autre en alternant le premier joueur (100 parties successives pour chacune des IAs en premier joueur)
- ☐ Relever et afficher différentes statistiques (% des parties gagnées, minimum, moyenne et maximum sur les nombres de cerveaux mangés...)