# Dice Battle

## 1 Description du jeu

Deux joueurs s'affrontent dans un jeu de dés. Le but est d'être le premier à atteindre au moins $N$ points (généralement $N = 100$). A chaque tour, on choisit de lancer entre 1 et $D$ dés à six faces (par exemple, $D = 10$). Le nombre de points marqués est 1 si l'un des dés au moins tombe sur 1, dans le cas contraire c'est la somme des dés. Nous allons étudier deux variantes de ce jeu :

- variante séquentielle : les joueurs jouent à tour de rôle (le premier joueur a donc bien sûr un avantage) ;
- variante simultanée : les joueurs jouent simultanément à chaque tour. Dans ce cas si les deux joueurs atteignent $N$ points ou plus lors du même tour, c'est le joueur qui dépasse le plus les $N$ points qui l'emporte ; si les deux joueurs obtiennent le même score, alors ils sont ex-aequos.

Le but du projet est de déterminer une stratégie de jeu optimale (et de tester cette stratégie par rapport à d'autres). On dit qu'un joueur a un gain égal à 1 s'il remporte la partie, un gain égal à 0 si la partie est nulle, un gain égal à -1 s'il perd. C'est donc en particulier un jeu à somme nulle.

## 2 Travail demandé

Il vous est demandé de réaliser le projet en répondant aux questions et en faisant les implantations et tests demandés. Une interface de jeu (pour choisir les stratégies à mettre en œuvre, mais aussi pour permettre à un humain de jouer contre l'ordinateur, ou à deux humains de jouer ensemble) sera appréciée.

Le travail se fait obligatoirement **en binôme de 2 étudiants du même groupe de TD**. Les binômes seront communiqués au chargé de TD. Le langage de développement est libre.

### 2.1 Rapport

Chaque binôme doit rédiger un rapport (de 5 pages environ, et dans tous les cas moins de 10 pages) répondant aux différentes questions de l'énoncé (quelques explications et commentaires peuvent ne pas être superflus). Il n'est pas utile de mettre le code dans le rapport.

Le rapport (format pdf) et le code (commenté) sont à remettre sur le moodle de l'ue avant le vendredi **6 décembre** 23h59. Vous soumettrez un fichier .zip (contenant le rapport et le code), le nom du fichier contenant le nom de famille des 2 membres du binôme. **Attention à bien soumettre dans votre groupe de TD.**

### 2.2 Soutenance

Une soutenance aura lieu la semaine du 16 décembre en salle machine (sur le créneau de la dernière séance de TD-TME). Chaque binôme doit donc avoir son code prêt à être exécuté sur son compte à la PPTI ou sur une machine personnelle. Il faut prévoir de pouvoir afficher une stratégie optimale à partir de valeurs de $N$ et $D$.

Chaque binôme présentera en 5 minutes son travail (simplement sur la base de son code, pas de transparent ou autre), puis répondra aux questions. Le planning sera établi pour chaque groupe par le chargé de TD.

**Il est impératif de remettre le rapport et d'effectuer la soutenance dans le groupe où vous êtes inscrit(e).**

## 3 Probabilités

Afin de pouvoir calculer des stratégies optimales, nous devons d'abord évaluer la probabilité de marquer un certain nombre de points lorsque l'on lance les dés.

Lorsqu'un joueur lance $d \geq 1$ dés, soit il obtient 1 point (si au moins un dé tombe sur 1), soit il obtient entre $2d$ et $6d$ points (il n'est donc pas possible d'obtenir entre 2 et $2d-1$ points).

Soit $P(d,k)$ la probabilité qu'un joueur qui lance $d$ dés obtienne $k$ points. Nous avons facilement :

- $P(d,1) = 1 - \left(\frac{5}{6}\right)^d$ : c'est la probabilité qu'au moins un dé tombe sur 1.
- $P(d,k) = 0$ pour $2 \leq k \leq 2d-1$ et pour $k > 6d$.

Pour $2d \leq k \leq 6d$, le calcul de $P(d,k)$ est plus compliqué. Nous pouvons l'écrire $P(d,k) = \left(\frac{5}{6}\right)^d Q(d,k)$ où $Q(d,k)$ est la probabilité d'obtenir $k$ points en jetant $d$ dés sachant qu'aucun dé n'est tombé sur 1. $Q(d,k)$ peut se calculer à l'aide de la formule de récurrence suivante pour $d \geq 2$ et $2d \leq k \leq 6d$ :

$$Q(d,k) = \sum_{j=2}^{6} \frac{Q(d-1,k-j)}{5}$$

**(Q1)** *Expliquer brièvement la relation de récurrence ci-dessus. Donner les cas d'initialisation. Implanter une méthode permettant de calculer le tableau des $P(d,k)$ pour tout $d \geq 1$, $k \geq 1$.*

Ces probabilités étant utilisées plusieurs fois, la méthode calculera (et renverra) un tableau contenant l'ensemble des $P(d,k)$.

## 4 Variante séquentielle

### 4.1 Stratégie aveugle

En première approche, on peut simplement lancer un nombre de dés qui nous assure en moyenne un nombre de points maximum. Si l'on lance $d$ dés, l'espérance $EP(d)$ du nombre de points obtenus s'obtient directement par la formule suivante :

$$EP(d) = 4d\left(\frac{5}{6}\right)^d + 1 - \left(\frac{5}{6}\right)^d$$

**(Q2)** *Implanter une méthode permettant de calculer en fonction de $D$ le nombre de dés $d^*(D)$ maximisant l'espérance du nombre de points.*

Nous appellerons dans la suite *stratégie aveugle* la stratégie consistant à lancer $d^*(D)$ dés.

**(Q3)** *Expliquer pourquoi cette stratégie n'est pas toujours la meilleure. On pourra donner un exemple précis où un joueur aurait intérêt à ne pas lancer $d^*(D)$ dés.*

## 4.2 Programmation dynamique

Une stratégie optimale pour la variante séquentielle peut être calculée par programmation dynamique. Soit $(i, j)$ l'état où le premier joueur a cumulé $i$ points, le deuxième joueur a cumulé $j$ points, et c'est au joueur 1 de jouer. Soit $EG(i, j)$ l'espérance de gain du joueur 1 dans l'état $(i, j)$, en supposant que son adversaire et lui-même jouent de façon optimale.

**(Q4)** *Donner une formule de récurrence pour calculer l'espérance de gain $EG(i, j)$ en un état $(i, j)$ donné. Comment initialiser la récurrence ?*

On rappelle que le jeu est à somme nulle.

**(Q5)** *Expliquer comment adapter l'algorithme pour pouvoir retourner une stratégie optimale.*

**(Q6)** *Expliquer quelle serait la difficulté algorithmique supplémentaire si l'on supposait que le nombre de points marqués est 0 (et non plus 1) si on obtient au moins un 1.*

## 4.3 Mise en œuvre

**(Q7)** *Programmer les stratégies aveugle et optimale.*

Pour la stratégie optimale, on pourra remarquer que le total de points d'un joueur ne peut jamais dépasser $N - 1 + 6D$.

On peut maintenant simuler des parties à deux joueurs, les joueurs utilisant l'une ou l'autre des stratégies. On peut par exemple s'intéresser à évaluer expérimentalement l'espérance de gain $G_1(a, b)$ du joueur 1 si le joueur 1 joue avec la stratégie $a$ et le joueur 2 avec la stratégie $b$.

**(Q8)** *Evaluer expérimentalement $G_1(a, b)$ pour $a \in \{aveugle, optimale\}$ et $b \in \{aveugle, optimale\}$. On pourra faire varier $D$ et/ou $N$.*

**(Q9)** *Proposer d'autres stratégies, les coder et évaluer expérimentalement leur qualité (espérance de gain face à d'autres stratégies) .*

On pourra par exemple utiliser des stratégies aléatoires (prendre $d$ au hasard entre 1 et $D$ étant la plus simple).

Comme indiqué précédemment, une interface permettant de jouer contre l'ordinateur sera appréciée.

# 5 Variante simultanée

Dans la variante séquentielle, il existait une stratégie optimale déterministe : en fonction de l'état du jeu, on détermine le nombre optimal de dés à lancer. Si les deux joueurs jouent à chaque tour de façon simultanée, alors la stratégie optimale dans un état $(i, j)$ donné n'est plus nécessairement déterministe : elle peut être randomisée. Ainsi, nous considérons des stratégies randomisées (dites mixtes), consistant à donner un vecteur $p = [p(1), p(2), \ldots, p(D)]$ où $p(d)$ est la probabilité avec laquelle nous lançons $d$ dés. Bien sûr, $p(d) \geq 0$ et $\sum_{d=1}^{D} p(d) = 1$.

## 5.1 Jeu en un coup

On considère en première approche une version simplifiée où l'on ne lance qu'une fois les dés. Le gagnant est simplement celui qui remporte le plus de points (toujours selon la même règle : 1 point si au moins un des dés tombe sur 1, la somme des dés sinon). En cas d'égalité, la partie est nulle. On note $EG_1(d_1, d_2)$ l'espérance de gain du joueur 1 s'il lance $d_1$ dés alors que le joueur 2 en lance $d_2$.

**(Q10)** *Comment calculer $EG_1(d_1, d_2)$ en fonction des probabilités $P(d, k)$ ? Donner la matrice des gains pour $D = 3$.*

Supposons que le joueur 1 adopte la stratégie consistant à lancer $d \in \{1, \ldots, D\}$ dés avec probabilité $p_1(d)$.

**(Q11)** *Si le joueur 2 connaît ce vecteur de probabilités $[p_1(1), p_1(2), \ldots, p_1(D)]$, quelle est sa stratégie optimale (en fonction des $p_i(d)$) ?*

**(Q12)** *Comment le joueur 1 doit-il choisir sa stratégie (son vecteur de probabilités $[p_1(1), p_1(2), \ldots, p_1(D)]$) de manière à maximiser son espérance de gain (en supposant que le joueur 2 réponde optimalement, selon la question précédente) ? Exprimer cela sous forme d'un programme linéaire.*

**(Q13)** *Implanter une méthode qui prend en entrée la matrice des gains (pour un certain $D$) et qui résoud ce programme linéaire. Donner une stratégie optimale pour différentes valeurs de $D$.*

**(Q14)** *Comparer expérimentalement cette stratégie optimale par rapport à la stratégie (déterministe) aveugle.*

## 5.2 (Facultatif) Cas général

###### Cette section est facultative. ######

Nous revenons au jeu initial décrit dans la section 1 où il s'agit de dépasser $N$ points. Ce cas est bien entendu plus compliqué car les stratégies optimales dépendent de l'état $(i, j)$ (état où le joueur 1 a accumulé $i$ points et le joueur 2 a accumulé $j$ points) dans lequel on est.

**(Q15)** *Donner l'espérance de gain $EG_1(i, j)$ du joueur 1 lorsque $i \geq N$ ou $j \geq N$.*

Nous voulons maintenant déterminer une stratégie randomisée optimale caractérisée par les probabilités $p_1^{i,j}$ (distribution de probabilité sur le nombre de dés à jeter) pour le joueur 1 lorsque l'on est dans l'état $(i, j)$ (avec $i < N$ et $j < N$). Nous supposons pour cela que nous connaissons déjà l'espérance de gain du joueur 1 $EG_1(k, \ell)$ lorsque $k > i$ et $\ell > j$ (lorsque les joueurs jouent de manière optimale).

Supposons que le joueur 1 lance $d_1$ dés et que le joueur 2 lance $d_2$ dés.

**(Q16)** *Exprimer l'espérance de gain $E_1^{d_1,d_2}(i,j)$ du joueur 1 en fonction des valeurs $P(d,k)$ et $EG_1(k,\ell)$.*

La matrice de jeu en un état $(i,j)$ donné est donc une matrice $E_1(i,j)$ avec $D$ lignes (stratégie de 1) et $D$ colonnes (stratégie de 2). La case $(d_1,d_2)$ de $E_1(i,j)$ correspond à l'espérance de gain $E_1^{d_1,d_2}(i,j)$ du joueur 1.

| $d_1, d_2$ | 1 | ... | $D$ |
|---|---|---|---|
| 1 | $E_1^{1,1}(i,j)$ | ... | $E_1^{1,D}(i,j)$ |
| ... | $\vdots$ | | $\vdots$ |
| $D$ | $E_1^{D,1}(i,j)$ | ... | $E_1^{D,D}(i,j)$ |

**(Q17)** *Implanter une méthode qui, étant donnés $i < N$, $j < N$, et un tableau contenant les valeurs $EG_1(k,\ell)$ pour $k > i$ et $\ell > j$, renvoie la matrice des gains $E_1(i,j)$.*

La méthode de la question (Q13) permet alors, étant donnée la matrice des gains $E_1(i,j)$, de calculer une distribution de probabilité optimale $p_1^{i,j}$ sur le nombre de dés à lancer dans un état $(i,j)$ donné pour le joueur 1.

**(Q18)** *Expliquer alors comment obtenir par programmation dynamique l'espérance de gain du joueur 1 $EG_1(i,j)$ pour chacun des états $(i,j)$. Le calcul de $EG_1(i,j)$ est fonction de valeurs $EG_1(k,\ell)$ lorsque $k > i$ et $\ell > j$, et l'algorithme résoud un PL en chaque état.*

**(Q19)** *Expliquer comment adapter l'algorithme pour pouvoir retourner une stratégie optimale. Implanter cet algorithme.*

Comme précédemment, la méthode renverra un tableau contenant une stratégie optimale pour chaque état $(i,j)$, afin de ne pas avoir à recalculer cela lors des simulations.

Supposons que le joueur 1 adopte cette stratégie optimale.

**(Q20)** *Evaluer expérimentalement l'espérance de gain du joueur 1 lorsque (1) le joueur 2 joue également optimalement (2) lorsqu'il joue la stratégie aveugle (3) lorsqu'il joue la stratégie séquentielle optimale.*