

INFO-H-100 : Projet 2017-2018

Jeu de Poker arrangé

Université libre de Bruxelles

Introduction

On vous demande de réaliser un jeu de poker dont les règles ont été modifiées, en Python. Il se joue avec un jeu de 52 cartes classique. Pour rappel, un jeu de 52 cartes est composé de 13 symboles dans 4 couleurs différentes. Les 13 symboles sont les suivants : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, Valet, Dame, Roi. Le projet se déroulera en 3 étapes qui correspondent à trois manières différentes de jouer au poker. Pour chacune des trois étapes, le jeu se déroulera comme suit :

1. Deux cartes sont distribuées à chaque joueur de façon individuelle. C'est à dire que chaque joueur voit ses deux cartes mais ne voit pas les deux cartes des autres joueurs.
2. Le 'flop', composé de trois cartes, est dévoilé. Cela veut dire que 3 cartes sont déposées de façon visible sur la table. Chaque joueur voit donc son jeu composé des deux cartes qu'ils ont en main et du flop, déposé sur la table.
3. Le 'turn', composé de une carte, est dévoilé. Il y a donc maintenant 4 cartes déposées et visibles sur la table qui peuvent composer le jeu de tous les joueurs.
4. La 'river', composée de une carte, est dévoilée.

Une fois toutes les cartes distribuées, chaque joueur aura donc le choix parmi 7 cartes (5 visibles et communes à tous les joueurs + 2 personnelles à chacun) pour avoir la meilleure combinaison de 5 cartes possible.

Les combinaisons prises en compte sont les suivantes :

- La paire ('p')
- La double-paire ('dp')
- Le brelan ('b')
- La suite ('s')
- La couleur ('c')
- Le full ('f')
- Le carré ('ca')
- La suite-couleur (encore appelée quinte flush) ('sc')

Pratiquement, elles seront référencées par le joueur par la chaîne de caractère mentionnée en vert entre parenthèse. Pour plus d'information sur la nature de chacune de ces combinaisons, voici une référence qui décrit chacune de ces combinaisons : https://fr.wikipedia.org/wiki/Main_au_poker#Ordre_des_niveaux_des_mains

1 Programmes à effectuer

1.1 Poker à un joueur se basant sur des mises

Le premier programme à effectuer sera un jeu de poker à un seul joueur où le joueur doit miser sur un résultat parmi la liste des combinaisons possibles. Le joueur commence avec une somme de 100 Bit coins (<https://fr.wikipedia.org/wiki/Bitcoin>). Selon le résultat final, la somme mise et le type de mise, le joueur gagnera une somme en retour.

Voici comment le jeu se déroulera :

- Le joueur reçoit deux cartes.
- MISE 1 : le joueur peut miser sur une combinaison de cartes finale. Pour que le joueur mise une certaine quantité de jetons sur une combinaison précise, le joueur répondra à deux 'input'. D'abord la combinaison sur laquelle il veut miser lui sera demandée. Il introduira la chaîne de caractère correspondant à la combinaison comme sus-mentionnée. Par exemple pour miser sur une paire, le joueur répondra à un input par 'p'. Ensuite, il devra répondre à un deuxième input qui lui demandera le nombre de jetons à miser. Il répondra par un nombre entier. Exemple : 4.
- Le flop, composé de trois cartes, est dévoilé.
- MISE 2 : le joueur peut miser sur une combinaison de cartes finale.
- Le turn, composé de une carte, est dévoilé
- MISE 3 : le joueur peut miser sur une combinaison de cartes finale.

- La river, composée de une carte, est dévoilée
- Suite au dépôt de la dernière carte, les différents résultats misés sont analysés (voir formule du calcul des gains ci-dessous) et les éventuels gains distribués au joueur si les mises correspondent aux résultats finaux.
- Si le joueur n'a plus de bit coins, le jeu s'arrête. Sinon, on demande au joueur s'il désire continuer ou pas. S'il veut continuer, on recommence à l'étape 1, avec de nouvelles cartes. Sinon, le jeu s'arrête.

En cas de succès d'une ou plusieurs des trois mises, les gains sont calculés comme suit :

$$Gain(Mise_i) = Somme_i \times \frac{1}{P(Mise_i) * i}$$

Où $Somme_i$ est la somme mise lors de la mise i . Le facteur i intervient car il est logique d'avoir un gain plus grand lorsque peu de cartes sont dévoilées. Le terme $P(Mise_i)$ représente la probabilité d'avoir le résultat de la mise i parmi 7 cartes. Ces probabilités, associées à chacune des combinaisons possibles seront mentionnées dans un fichier auquel vous accèderez sur Upylab via le chemin '/pub/data/probabilites.txt'. Pour que vous vous entraîniez, nous vous fournirons ce fichier texte et nous vous laissons le soin d'analyser ce fichier pour connaître son contenu exact. De nouveau, il est logique que le gain soit d'autant plus grand que le résultat misé soit peu probable.

1.2 Poker à deux joueurs se basant sur des mises

Il s'agit du même principe que le point précédent mais avec un autre joueur géré par l'ordinateur. Tous deux commencent avec une somme de 100 bit coins. Ce joueur va donc également miser à chaque tour. Il le fera en calculant un ratio de probabilité r , à maximiser :

$$r = \frac{P_x(Mise_i)}{P(Mise_i)}$$

$P_x(Mise_i)$ est la probabilité d'obtenir une combinaison i (paire, brelan etc.) en ne connaissant que x cartes dévoilées ($x = 0, 3, 4$ respectivement pour la mise 1, 2 et 3). Par exemple, la probabilité d'obtenir un brelan sera importante si le joueur a dès le départ, une paire en main (ou encore si une paire apparaît une fois que le flop est dévoilé). Ces probabilités $P_x(Mise_i)$ ne doivent pas être calculées de façon exacte et théorique. On vous demande d'utiliser des techniques de simulations de type Monte Carlo. Ces méthodes consistent à effectuer un nombre très élevé (que vous devrez définir) de simulations de jeux finaux possibles, pour calculer combien de fois chacune des combinaisons i possibles sera apparue. Connaissant le nombre de simulations effectuées et le nombre de fois que chacune des combinaisons i sera apparue, vous obtenez $P_x(Mise_i)$. L'optimisation de votre code aura ici une importance toute particulière de manière à ce son exécution ne prenne pas trop de temps.

$P(Mise_i)$ est la probabilité d'obtenir la combinaison i quand toutes les cartes sont connues. Ces probabilités sont indépendantes des cartes distribuées. Ce sont des probabilités inhérentes au jeu du poker. À nouveau, les $P(Mise_i)$ se trouvent dans le fichier 'probabilites.txt' que l'on vous fournit.

1.3 Poker à deux joueurs

Le dernier mode de jeu ressemble au poker classique qui se jouerait avec un autre joueur géré par l'ordinateur. Cette fois-ci, les joueurs ne misent plus sur la combinaison qu'ils pensent obtenir, mais sur le fait qu'ils obtiennent une meilleur combinaison que leur adversaire. À nouveau, ils partent tous les deux avec 100 bit coins chacun.

Les tours de mises se déroulent de la manière suivante :

- Le joueur mise
- L'ordinateur peut se coucher, égaliser la mise du joueur ou miser d'avantage.
- Le joueur peut soit égaliser, soit se coucher.

Lorsque le joueur ou l'ordinateur se couche, celui-ci perd la partie, et son adversaire reçoit l'entièreté de la somme mise par les deux joueurs.

La quantité d'argent mise par l'ordinateur est calculée de la manière suivante :

- L'ordinateur calcule P_i , la probabilité que l'ordinateur gagne la partie, connaissant ses propres cartes ainsi que celles retournées sur la table à l'étape i (avant le flop, avant le turn, etc.) Cette probabilité est à nouveau calculée grâce à des simulations de type Monte Carlo.

— L'ordinateur calcule ensuite la somme :

$$S_i = P_i \times A$$

où A représente l'argent de l'ordinateur. Cette quantité S_i représente la somme que l'ordinateur est prêt à miser au total sur la partie connaissant les cartes dévoilées jusqu'à l'étape i .

— Une fois cette quantité calculée, trois alternatives sont possibles en fonction de S_i , de l'argent misé par le joueur (M_i), d'un seuil d'indifférence c et d'une probabilité p :

1. $M_i + c < S_i$: la somme que l'ordinateur est prêt à miser est significativement plus large que la somme mise par le joueur. L'ordinateur va, dans ce cas, miser S_i et ce sera au joueur d'à son tour égaliser la mise.
2. $M_i - c \leq S_i \leq M_i + c$: la somme que l'ordinateur est prêt à miser est similaire à celle mise par le joueur. L'ordinateur égalise dans ce cas la mise du joueur.
3. $S_i < M_i - c$: la somme que l'ordinateur est prêt à miser est significativement inférieure à celle mise par le joueur. Dans ce cas de figure, l'ordinateur décidera avec une probabilité p de tout de même égaliser la somme mise par le joueur, et décidera donc de se coucher avec une probabilité $1 - p$.

Lorsque la 'river' a été dévoilé, un dernier tour de mise a lieu. Ensuite, les cartes de l'ordinateur sont dévoilées et le joueur ayant la meilleure combinaison remporte l'argent misé.

2 Consignes

Il s'agit d'un projet individuel. Les remises de projet se feront exclusivement via la plateforme 'UpyLaB'.

Le projet sera donc divisé en différentes parties (correspondant à différentes fonctions) dans l'environnement UpyLaB. La plateforme vous permettra donc de vérifier si votre code donne le résultat désiré.

Il est très vivement conseillé de coder les différentes fonctions sur Idle avant de les exécuter dans UpyLaB. Afin d'éviter les problèmes de plagiat, l'ensemble des projets seront comparés grâce à un logiciel spécialisé. En cas de plagiat, les étudiants concernés auront la note de 0/20 pour le projet.

Il est important de tenir compte du style et de l'efficacité de votre code. En effet, chaque code sera analysé par un assistant après passage dans l'environnement UpyLaB.

La pondération de chaque fonction à créer sera donnée via la plateforme UpyLaB. Il est possible que toutes les fonctions ne soient pas mise au même moment sur la plateforme.

Les élèves assistants sont disponibles pour répondre à vos questions lors des guidances organisées sur les temps de midi.

BON TRAVAIL !