

# Description du projet : un jeu de quilles

---

## Projet python PeiP1 : Jeu de quilles

---

Le but de ce projet est d'écrire un jeu de quilles avec comme interface graphique le module Turtle.

**L'objectif pédagogique** est de vous faire travailler sur un projet assez conséquent où vous serez obligés de mettre en œuvre les bonnes pratiques de programmation vues pendant le semestre, afin d'arriver au bout du projet. Les points importants seront la décomposition (modules et fonctions), la bonne organisation du code, et la gestion de projet (répartitions des tâches entre les membres du binôme, organisation dans le temps).

---

## Le jeu

### Règles du jeu

Le joueur joue contre l'ordinateur. Au départ, un nombre de quilles est tiré au hasard. Le joueur commence puis à chaque tour, le joueur ou l'ordinateur choisit de tirer au milieu, à gauche ou à droite d'une ligne de quilles. Quand on tire au milieu, 2 quilles sont descendues, quand on tire à droite ou à gauche, une seule quille est descendue. Quand on descend 2 quilles au milieu, la ligne de quille est coupée en deux. Celui qui descend la dernière quille a gagné.

### Interface

Pour que cela soit plus agréable, les quilles seront représentées par un dessin fait avec turtle, et elles seront situées dans un décor. Chaque projet sera différent, vous devrez laisser parler votre créativité pour choisir un thème. Il vous faudra choisir un dessin simple pour les quilles debout, et un dessin modifié pour les quilles couchées. Par exemple, ceux qui aiment la nature, pourront couper des arbres (arbres entier = quille debout, arbre coupé = quille tombée), alors que ceux qui aiment les fast-food, pourront manger des frites (frite entière = quille debout, frite avec moitié de ketchup = quille tombée).

Votre décor devra inclure les éléments suivants:

- une zone pour afficher ce que vient de jouer l'ordinateur,
  - une zone pour afficher les quilles,
  - un élément du décor devra se trouver à deux endroits différents (par exemple, 2 montagnes, ou deux tables)
  - un élément du décor (différent des quilles) devra être répétitif (par exemple, une forêt qui contient plusieurs arbres, une tribune de foot qui contient plusieurs spectateurs ...).
- 

## Le code python

Votre code devra contenir toutes les fonctions et procédures nécessaires. Il ne suffira pas que votre projet marche, il faudra qu'il soit **bien écrit** !.

### Boucle de jeu

Le programme principal suivra l'algorithme suivant :

```
Tirer un nombre aléatoire de quilles
Tant qu'il n'y a pas de gagnant :
    Faire jouer le joueur
    Mettre à jour le nombre de quilles
    Afficher les quilles
    S'il n'y a plus de quilles : le joueur a gagné
    S'il reste des quilles :
        Faire jouer l'ordinateur
        Mettre à jour le nombre de quilles
        Afficher les quilles
    S'il n'y a plus d'allumettes, l'ordinateur a gagné
```

## Structure de données

Pour ce jeu, le choix d'une structure de données adaptée est un point délicat. En effet, il faut non seulement connaître le nombre de quilles mais aussi quelles quilles sont debout et quelles quilles sont tombées.

Nous vous **imposons la structure de données** suivante. Une liste *quilles* représente l'ensemble des lignes de quilles. Cette liste est formée de listes *[deb,fin]* telles que il y a des quilles debout entre *deb* et *fin* et partout ailleurs il y a des quilles tombées. Il faut également connaître le nombre initial des quilles noté *nlnit*. Par exemple, si *quilles=[[1,4],[7,9]]* et *nlnit=13*, il y a des quilles de 1 à 4 et de 7 à 9, et les quilles tombées sont : la quille 0, les quilles 5 et 6 et les quilles 10,11 et 12. A l'état initial on a *quilles=[[0,nlnit-1]]* puisqu'il y a une seule ligne de quilles.

Nous vous **imposons** aussi la **syntaxe d'entrée** : pour indiquer que l'on veut jouer au milieu de la ligne *i* il faudra écrire *i:M*, pour la droite on mettra *i:D* et pour la gauche on mettra *i:G*.

## Interface graphique

Quelques conseils pour mener à bout votre interface graphique. Nous nous restreignons à une utilisation simple de la tortue afin d'être sûr que tous les projets aboutissent.

- commencer par faire un dessin papier (une maquette) de votre interface en plaçant les différentes zones,
- faites des dessins simples à base de formes élémentaires (c'est la structure et la décomposition qui seront valorisées dans la notation),
- la zone qui contient le dessin des quilles est effacée à chaque tour de jeu (par exemple en affichant un grand rectangle qui couvre la zone) puis recouverte des nouvelles quilles.
- la réponse de l'utilisateur est lue avec **textinput** qui ouvre une fenêtre popup et renvoie une chaîne de caractères avec le texte tapé par le joueur (par exemple, 4:D)

En suivant ces consignes, vous pourrez mener à bien l'interface graphique du jeu.

---

## Travail à réaliser

Ce projet sera réalisé en binômes, le choix des binômes est libre mais les deux membres doivent être dans le même groupe.

Au terme de ce projet, vous devrez fournir :

1. un fichier *quilles.py* qui permet de lancer votre jeu
2. (optionnel mais recommandé) des modules (fichiers .py que vous importerez) permettant de dessiner les éléments du décor, les formes de base ...

3. un rapport *rapport.pdf* (**PAS** de .docx) entre 150 et 250 lignes contenant les règles du jeu, expliquant la structure de votre projet (décomposition en fonctions, en modules, organisation du code, ...), la répartition du travail entre les deux membres du binôme, les difficultés rencontrées ainsi que les solutions proposées, comme par exemple, l'utilisation de fonctions complexes du module *turtle*), et le travail qui reste à faire,
4. un fichier *ecran.jpg* qui est une copie d'écran de votre jeu pris au 3ème tour de jeu.



**La première ligne de chaque fichier .py doit contenir le nom des deux membres du binôme.**



**Ne sous-estimez pas la difficulté du projet, car vous avez peu de temps, il vous faut donc travailler régulièrement et bien partager les travail entre les deux membres du binôme.**

**Nota :** Il existe une stratégie gagnante pour ce jeu, que vous verrez au 2d semestre. Ceux qui sont intéressés peuvent regarder du côté des jeux de Nim.

---

## Critères d'évaluation

**Nous nous réservons le droit de mettre une note différente aux deux membres du binôme en fonction de ses réponses à l'oral.**

Votre projet sera évalué selon les critères suivants :

- Respect des dates de rendu (déclaration binômes, rendu intermédiaire, livraison du projet)
- Respect des consignes (nom des fichiers, nom des membres du binôme dans les fichiers .py ...)
- Suivi de projet : version textuelle et description écrite du projet
- Qualité des commentaires
- Qualité du rapport
- Respect des conventions de code (choix des identificateurs)
- Fonctionnalités (testées en séance)
- Structure du code (fonctions, paramètres, valeurs de retour, utilisation de paramètres par défaut,...)
- Bonne utilisation des modules
- Clarté du code : bonne factorisation, écriture correcte des boucles, ...
- Oral (attribué de façon individuelle)
- Créativité

Modifié le: mercredi 11 décembre 2019, 12:02