

# COMPTE-RENDU BATAILLE

*Groupe 4 : Arthur CHEVALIER, Pierre LOPES, Arthur COTTARD*

## 1. Projet et cahier des charges

Le projet était de créer un jeu de bataille où deux joueurs dits "robots" s'affrontent à l'aide d'un tirage aléatoire des cartes.

Il s'agissait également de découper du déroulement du jeu à l'aide d'un schéma, disponible à la racine du code source (schema\_bataille\_\*.jpg).

Enfin, la rédaction d'un carnet de bord décrivant le parcours au fur et à mesure des séances, permet de retracer les choix mais aussi les avancées du projet et la répartition du travail du début jusqu'à l'achèvement du projet. (carnet\_de\_bord.html)

## 2. Présentation du projet achevé

### 2.1. Principales fonctionnalités

- Classes
  - Paquet
    - ajouter une carte, assembler deux paquets, ...
  - Carte
    - Comparer deux cartes (surcharge de `__lt__()`), ...
  - Exceptions
- Fonctions
  - création paquet initial
  - affrontement (entre deux cartes)
  - bataille (lorsqu'il y a égalité à l'affrontement)

### 2.2. Comment exécuter le programme ?

Tout d'abord, il s'agit de comparer la version du code source en local avec la version la plus récente sur Github (lien en section 6).

1. Importer en console ou dans un fichier python les fonctions `main` et `main_simulation` pour pouvoir les exécuter en changeant les paramètres ou en récupérant les valeurs retournées.
2. Exécuter le fichier `bataille.py` (pas de simulation et paramètres par défaut).

### **3. Organisation du projet et problèmes surmontés**

#### **3.1. Outils**

Nous avons choisi d'uniformiser nos environnements de travail pour permettre une productivité accrue et une entraide optimale. Nous nous sommes alors tourné vers VSCode qui présente plusieurs avantages comme un catalogue fourni d'extensions (suggestions intelligentes, ...) et aussi une interface graphique pour le système de gestion de versions Git.

Aussi, le projet est hébergé sur Github (lien en section 6) et associé à l'utilisation de Git, cela a permis de limiter les contraintes dûes au travail de groupe (synchronisation des fichiers, suivi des modifications).

#### **3.2. Découpage du projet en sous-problèmes et déroulement algorithmique**

Le jeu est composé de deux grandes phases : la mise en place et le déroulement

La première constituant principalement de la création des paquets telle qu'elle se fait dans la réalité et la deuxième faisant appel à des concepts algorithmiques plus poussés avec des boucles et des conditions multiples.

Plus de détails : voir schéma en section 7 ou code source

Aussi, il est important de noter que le projet a été développé avec un aspect collaboratif : pour recréer un jeu de poker ou de solitaire, n'importe qui pourrait récupérer le code source des fichiers `carte.py` et `paquet.py` entre autres.

Cela explique aussi l'utilisation d'exceptions adaptées à la situation afin de permettre une gestion des exceptions (`try.. except`) plus poussée dans le cas d'un fork du projet.

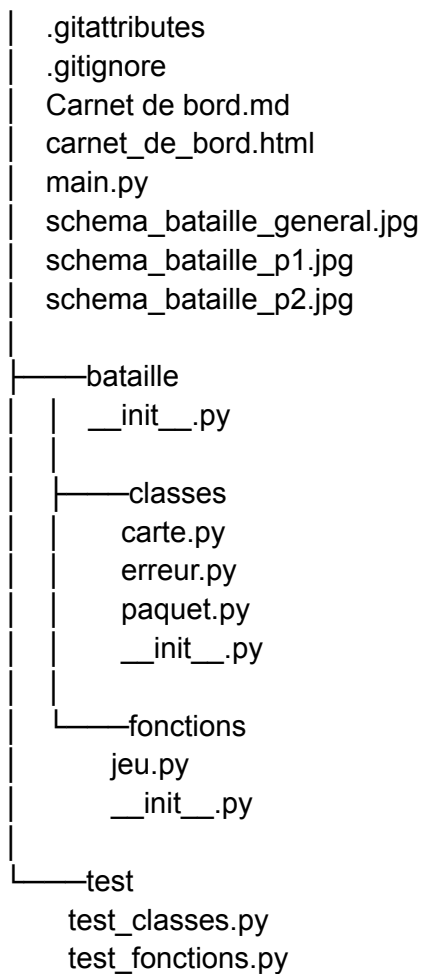
#### **3.3. Tests**

Afin de vérifier le bon fonctionnement du code, des tests ont été mis en place. A l'aide du package `unittest`, chaque fonction ou méthode a pu être testé afin d'assurer un bon fonctionnement global.

#### **3.4. Carnet de bord**

Organisé par séance, avec l'ordre du jour et la planification de la prochaine séance, le carnet de bord retrace l'avancement de l'équipe en précisant les réflexions, les étapes importantes et les participants à chacune des avancées.

### 3.5. Arborescence du projet



### 3.6. Problèmes rencontrés

- Séparation du projet en différents modules (import, ...)
  - Solution : Organisation en différents dossiers : classes, fonctions, ...
- Incertitude du déroulement du programme à cause du hasard du tirage des cartes.
  - Solution : Encadrement du déroulement à l'aide de conditions

### 3.7. Documentation

Chaque fonction, méthode et classe (sauf les tests) est documentée à l'aide des docstrings, décrivant les arguments et les objets retournés sans oublier le rôle de la fonction, méthode ou classe.

#### 4. Ressentis personnels

- Arthur CHEVALIER : “Je trouve que les débuts du projet ont été plutôt rapides ainsi que la mise en place, ce qui a laissé du temps pour la partie code. L'ambiance était agréable et on a répondu à mes questions quand j'en avais.”
- Pierre LOPES : “Le groupe a travaillé de façon efficace, on a vite su comment on voulait s'organiser dans notre projet et c'était agréable de travailler ensemble. Peu de difficultés ont été rencontrées ce qui était satisfaisant.”
- Arthur COTTARD : “Un projet mené sans accroc qui a permis à chacun de s'entraîner au travail en groupe. L'objectif étant atteint, chaque membre peut être satisfait d'avoir participé de manière efficace.”

#### 5. Observations sur les simulations du jeu et partie graphique

##### 5.1. Exemple de l'affichage du jeu en console

```
Manche 443
| 8 carreau | | 1 trèfle |
Vainqueur de la manche: 2

Manche 444
| 7 pique | | 7 coeur |
BATAILLE
| 4 carreau | | 3 carreau |
FIN BATAILLE
Vainqueur de la manche: 1

Manche 445
| 1 coeur | | 1 trèfle |
BATAILLE
| Reine carreau | | 8 carreau |
FIN BATAILLE
Vainqueur de la manche: 1

VAINQUEUR DE LA PARTIE: 1
```

## 5.2. Simulation

A l'issue du projet, l'idée nous est venue d'exécuter de nombreuses fois le programme, de manière à observer le nombre de manches, le vainqueur et aussi si l'exécution se déroulait correctement.

Ensuite nous l'avons automatisé afin d'obtenir des statistiques plus précises, la fonction `main_simulation()` a alors vu le jour, affichant la moyenne du nombre de manches et le nombre de victoires par joueur pour le nombre d'itérations désiré.

Ainsi, on remarque qu'en moyenne, une partie compte autour de 300 manches et que les nombres de parties gagnées entre le joueur 1 et le joueur 2 sont à peu près égaux sans pour autant être parfaitement identiques.

## 5.3. Partie graphique

En cours de développement, accessible par la branche "graphique" sur Github.

Consiste en un affichage graphique des cartes avec une exécution contrôlée et une interface plus conviviale.

L'idée est venue de Pierre ayant manifesté sa volonté de faire une partie graphique. On a ajouté une branche du projet contenant les images de cartes et le fond. Arthur COTTARD et Pierre LOPES ont fait des recherches pour essayer de développer la partie graphique en premier temps avec Tkinter puis avec Pygame.

## 6. Lien

- Github : <https://github.com/swarthur/bataille>

## 7. Schéma

