

Projet : jeu de la Bataille

Description du projet :

Ceci est une bataille codé pour un projet scolaire de la spécialité NSI pour le 22 décembre 2023.

Nous avons commencé de 0, le but de ce projet était d'apprendre à coder en python et d'utiliser la POO.

Prérequis :

- Installer le module random :

```
pip install random
```

- Installer le module colorama :

```
pip install colorama
```

Installation :

Pour installer le projet grâce à GitHub, rendez-vous sur ce lien :

<https://github.com/Bessastien/Bataille> et cliquer sur le bouton "<> Code" puis sur "Download ZIP".

Comment lancer le programme :

Si vous êtes arrivés jusque-là vous devez avoir télécharger le dossier et il doit être dézipper. Il vous suffit donc de lancer main.py à la racine du projet grâce à python

Module : pile_file.py

Classe : Pile





Méthodes :	Descriptions :
p = Pile()	Créer une pile vide.
p.est_vide()	Fonction qui renvoie True si la pile est vide, False sinon. Return : bool
p.taille()	Fonction qui renvoie la taille de la pile. Return : int
p.empile(val)	Paramètre val : int Fonction qui empile la valeur val à la pile.
p.depile()	Fonction qui dépile la pile. Return le dernier élément de la pile.
p.__str__()	Fonction qui renvoie une représentation de la pile. Return : str

Classe : File

Méthodes :	Descriptions :
f = File()	Créer une file vide.
f.est_vide()	Fonction qui renvoie True si la file est vide, False sinon. Return : bool
	Fonction qui renvoie la taille de la file. Return : int
f.enfile(val)	Paramètre val : int Fonction qui enfile la valeur val à la file.
f.defile()	Fonction qui dépile la file. Return le premier élément de la file.
f.__str__()	Fonction qui renvoie une représentation de la file. Return : str

Module : Carte.py

Classe : Carte

Méthodes :	Descriptions :
c = Carte(nombre, symbole)	Paramètre nombre : un nombre compris entre 2 et 14 (11 = Valet, 12 = Dame, 13 = Roi et 14 = As) Paramètre symbole : Le symbole de carte ( ,  ,  ou ) Créer une carte avec nombre et symbole comme caractéristique
c.get_nombre()	Fonction qui renvoie le nombre de la carte. Return : int
c.__eq__(autre_carte)	Paramètre autre_carte : Carte() Fonction qui renvoie True si les cartes sont égales, False sinon. Return : bool
c.__lt__(autre_carte)	Paramètre autre_carte : Carte() Fonction qui renvoie True si la carte est inférieure à l'autre, False sinon. Return : bool
c.__gt__(autre_carte)	Paramètre autre_carte : Carte() Fonction qui renvoie True si la carte est supérieure à l'autre, False sinon. Return : bool
c.get_true_nombre()	Fonction qui renvoie le vrai nom de la carte. Return : str
c.__str__()	Fonction qui renvoie une représentation de la carte. Return : str

Classe : Paquet

Méthodes :	Descriptions :
P = Paquet	Créer un paquet de cartes mélangés.
p.generer_paquet()	Fonction qui génère un paquet de cartes. Return une liste de Cartes mélangés
p.diviser_paquet()	Fonction qui divise le paquet en deux. Return un tuple de deux File
p.__str__()	Fonction qui renvoie une représentation d'un paquet. Return : str

Module : Jeu.py

Classe : Bataille

Méthodes :	Descriptions :
b = Bataille()	Créer une instance de la classe Bataille
b.tour(j1,j2)	Paramètre j1 : File représentant la main du joueur 1 Paramètre j2 : File représentant la main du joueur 2 Fonction qui gère un tour de jeu
b.bataille(j1,j2,autre_carte)	Paramètre j1 : File représentant la main du joueur 1 Paramètre j2 : File représentant la main du joueur 2 Paramètre autre_carte : Liste de toutes les cartes qui sont mise en jeu au moment de l'appel de la fonction. Fonction qui gère une bataille
b.jeu	Fonction qui gère le jeu

Module : main.py

Fonction :	Description :
main()	Lance le jeu

Ce module permet de lancer le jeu.