
ART OF WAR

Synthèse des modifications apportées au projet de Nathan Traineau et Lucas Mas

Remarques générales :

- Beaucoup d'erreurs de syntaxe ont été notées, beaucoup de `false` à la place de `False`, le nom de variable « `type` » utilisé alors qu'il sert au langage, etc...
- L'utilisation d'un identifiant pour repérer une carte en particulier est forcé par le code et impacte donc sur la structure de donnée à utiliser.
- Il y a 42 cartes, si on rentre une autre valeur, le programme ne prend pas en considération cette erreur.

Le fichier principal `artofwar_T_M.py` :

- `retirer_de_main()` et `placer_champs()` utilisent un identifiant et non pas une carte.
- Modification de l'input pour la récupération dans le deck à partir de l'identifiant

Le fichier `main.py` dédiée aux fonctions de la main :

- La fonction `placer_dans_main(carte, joueur)` a été remplacée par `placer_dans_main(carte, main)`

Le fichier `royaume.py` :

- La fonction `est_dans_royaume(carte)` ne prends pas en paramètre de royaume, seulement une carte. Il est pourtant nécessaire de savoir dans quelle royaume se trouve la carte à rechercher.

Le fichier carte.py :

- La fonction `est_en_posture_defensive(carte)` prenait seulement une carte, mais elle nécessite également de prendre en paramètre un royaume : `est_en_posture_defensive(carte, royaume)`.
- La fonction `touche(carte1, carte2)` ne prenait pas en compte les champs de batailles dédiés à la carte 1 et à la carte 2. Or il était nécessaire pour obtenir les positions de carte et donc vérifier que les cartes étaient à portée pour l'attaque : `est_en_posture_defensive(c1, c2, cdb1, cdb2)`
- `getID()`, `aEteTouche()` sont des fonctions ajoutées car manquantes dans les spécifications fonctionnelles du type `Carte`.

Le fichier Joueur.py :

- La fonction `reinitialiser_cartes(joueur)` avait besoin d'un accès à la structure de donnée du champ de bataille et du royaume. On l'a alors séparé en deux fonctions dans les fichiers respectifs afin d'éviter l'accès à la structure de donnée par l'usage de fonctions propres au type.

Le fichier test unitaires :

- `test_is_front()`
`test_get_position_utilisables()`
Sont faux car ils ne testent pas la fonctionnalité principale de la fonction

- ```
def test_piochercarte():
 DeckGen = creerDeck()
 pioche = creer_pioche(1, DeckGen)
 PiocheDebut = pioche ### == ERREUR == ###
 carte = piocher_carte(pioche)
 return (PiocheDebut != pioche)
```

**### ERREUR car en modifiant pioche on modifie également PiocheDebut, le test renverra donc toujours False.**