

SUJET

Le "1000 Bornes™" est un jeu de société utilisant des cartes sur le thème de la course automobile et a été inventé en 1954 par Edmond Dujardin. Le but du "1000 Bornes™" est d'être le premier joueur à effectuer 1000 kilomètres précisément. Pour cela, il vous faudra éviter les pièges de la route (panne de voiture, feux de signalisation, ...). Le détail des règles peut être consulté sur ces sites :

- www.regles-de-jeux.com/regle-du-1000-bornes
- www.regledujeu.fr/regle-mille-bornes

L'objectif de ce projet est

1. de proposer un diagramme de classes UML modélisant ce jeu
2. d'implémenter en Python ce diagramme

Quelques informations concernant le projet

- Le projet (diagramme UML et ses justifications, code Python) est à rendre pour le **15 janvier**
- Le projet est à réaliser **par groupe de 2 en autonomie**.
- Vos chargés de TP sont disponibles pour suivre votre travail, mais c'est à vous de les solliciter

MODALITÉS

1. Diagramme de classes UML :

- (a) Votre diagramme UML devra modéliser le jeu du "1000 Bornes™" tout en utilisant l'ensemble des principes objet (liens entre classes, encapsulation, abstraction, polymorphisme).
- (b) Le projet permet l'utilisation de ces concepts et votre diagramme devra donc contenir
 - au moins une aggrégation ou une composition
 - au moins un lien d'héritage
 - au moins une classe abstraite
- (c) Les choix de lien entre classes (autre que l'héritage) et d'encapsulation devront être justifiés.
- (d) Le diagramme UML devra être lisible, et il est donc conseillé de le réaliser
 - soit à l'aide d'un logiciel (Dia, ArgoUML, ...)
 - soit via un site internet (online.visual-paradigm.com, creately.com/lp/uml-diagram-tool)

2. Implémentation Python :

- (a) Votre implémentation Python devra correspondre à votre diagramme de classe UML
- (b) Votre code devra
 - être commenté de sorte à être compréhensible
 - satisfaire le "Zen of Python" (www.python.org/dev/peps/pep-0020).
 - satisfaire les principes objets (utilisation de constructeur, méthodes de conversion d'objet en chaîne de caractères, comparaison d'objets, ...)
 - être composé de plusieurs fichiers / modules qui seront importés par le programme principal
- (c) Votre programme devra fonctionner
 - à partir d'affichages clairs dans la console (mode textuel, cf image ci-dessous)
 - en mode Joueur contre Joueur (*i.e.* que des humains)
 - à l'aide de valeurs passées par l'utilisateur (carte qu'il souhaite jouer, joueur qu'il souhaite attaquer, ...)

3. Options : en plus de l'implémentation de base décrite en Q2, votre programme devra implémenter au moins une des options suivantes :

- (a) avoir une interface graphique (mais une version console textuelle devra toujours exister)
- (b) implémenter une "IA" pour permettre un jeu (1 à 3) Humain vs (1 à 3) Ordinateur. Plusieurs niveaux de difficulté pourront être proposés.
Une IA consistera simplement en une suite d'instructions permettant de simuler un joueur humain (inutile de recoder AlphaGo ou Watson ...). Par exemple, l'ordinateur essaiera prioritairement de se protéger (à l'aide des bonus) puis d'avancer et enfin d'attaquer le meilleur joueur.
- (c) avoir une fonctionnalité de sauvegarde (dans un fichier texte) pour permettre d'interrompre une partie à la fin d'un tour de jeu. On prévoira évidemment une fonction de chargement de partie pour la reprendre lorsqu'on le souhaite

```
+-----+
| Bienvenue dans le jeu du 1000 bornes! |
+-----+

Combien de joueurs? 4

Veuillez entrer le nom du joueur 1 et s'il s'agit d'une IA (0 / 1): Chuck Norris, 0
Veuillez entrer le nom du joueur 2 et s'il s'agit d'une IA (0 / 1): Maitre Yoda, 1
Veuillez entrer le nom du joueur 3 et s'il s'agit d'une IA (0 / 1): Hairy Otter, 1
Veuillez entrer le nom du joueur 4 et s'il s'agit d'une IA (0 / 1): Tux, 0

-----
Joueur Chuck Norris (0 km, Feu rouge)
IA Maitre Yoda (0 km, Feu rouge)
IA Hairy Otter (0 km, Feu rouge)
Joueur Tux (0 km, Feu rouge)
```

(a) Écran d'accueil

```
-----
C'est a Joueur Chuck Norris (150 km) de jouer:
0: Distance: Escargot, 25
1: Distance: Escargot, 25
2: Distance: Lievre, 100
3: Attaque: Feu rouge
4: Defense: Fin de limitation de vitesse
5: Defense: Feu vert
6: Defense: Roue de secours
7: Defausser une carte

Que souhaitez vous faire (0 a 7)?2
Carte Distance: Lievre, 100 ajoutée! Nouvelle distance parcourue: 250 km
```

(b) Utilisation d'une carte distance

```
-----
Joueur Chuck Norris (1000 km, As du volant, Vehicule prioritaire) a gagne!
Voici le classement des joueurs:
Joueur Chuck Norris (1000 km, As du volant, Vehicule prioritaire)
Joueur Tux (975 km, Feu rouge, Pneus increvables, Limitation de vitesse)
IA Maitre Yoda (800 km)
IA Hairy Otter (225 km)
```

(c) Victoire d'un joueur et classement

```
-----
C'est a Joueur Chuck Norris (0 km, Feu rouge) de jouer:
0: Distance: Escargot, 25
1: Distance: Escargot, 25
2: Distance: Escargot, 25
3: Distance: Canard, 50
4: Distance: Papillon, 75
5: Distance: Lievre, 100
6: Defense: Reparation
7: Defausser une carte

Que souhaitez vous faire (0 a 7)?7
Quelle carte souhaitez vous defausser (0 a 6)?0
Vous defaussez la carte Distance: Escargot, 25
```

(d) Carte défaussée