Bataille Navale - TP4

Katleen Blanchet

katleen.blanchet@gmail.com

16 mai 2017

TO DO

1. Finir le jeu de base pour ceux qui sont TRÈS en retard
2. Intégrer le client/serveur en respectant le format donné à la fin des slides
3. Programmer l'IA (mode "Jouer contre ordinateur")
   1. Chercher une stratégie simple pour jouer au jeu (choix des bateaux, et tir)
   2. Implémenter cette stratégie
   3. Complexifier la stratégie pour réduire les possibilités de tir

   Même avec la meilleure stratégie, une IA de bataille navale peut difficilement gagner à toutes les parties. Ce jeu contient une grande part de hasard.
4. Création de l'interface graphique pour les plus avancés avec Tkinter

Librairie Tkinter

Documentation

- Documentation : `http://effbot.org/tkinterbook/`
- Documentation plus succincte : `http://www.fil.univ-lille1.fr/~marvie/python/chapitre6.html`
- Exemple MVC : https ://sukhbinder.wordpress.com/2014/12/25/an-example-of-model-view-controller-design-pattern-with-tkinter-python/

Vérification de l'installation de la librairie

- Taper 'python3' dans un terminal
- Taper la commande 'import tkinter'
- Attention, avec Python3, utiliser **t**kinter et non pas **T**kinter

Interface graphique I

Première fenêtre

- Créez votre première fenêtre
- Donnez lui un titre
- Ajoutez votre premier widget (composant de l'interface) : un label
- Ajoutez un bouton "Quitter"
- Ajoutez un canvas
- Créez plusieurs rectangles côte à côte
- Créez un ovale s'étendant sur les rectangles

Interface graphique II

Affichage du jeu (sans interaction)

- Observez le modèle MVC (Model-View-Controller) donné slide précédent. Il permet de faire la distinction entre le jeu ("Model"), l'interface ("View"), et le client dans votre cas ("controller") qui fait le lien entre les deux.
- Dans un premier temps, votre interface représente l'état du jeu pour un joueur. Elle doit contenir sa grille (avec ses bateaux et les tirs de l'adversaire) et la grille de l'opposant (avec les tirs du joueur).
- Vous pouvez éventuellement écrire le nom du gagnant à la fin de la partie et d'autres informations qui vous semblent utiles au cours du jeu.
- ATTENTION : ici vous ne faites qu'afficher des informations sur l'interface, le programme a besoin de continuer son exécution séparément de l'affichage. Il faut donc remplacer $w.mainloop()$ par $w.update\_idletasks()$

Format d'échange à respecter

- Tous les échanges s'effectuent en bits.
  - conversion d'un string $s = "text"$ en bits : $s.encode()$ ou $b'text'$
  - conversion d'une chaîne de bits b en une chaîne string : $b.decode()$
- Le serveur envoie :
  - 'VS' pour savoir si le joueur veut jouer contre un autre joueur ou contre l'ordinateur
  - 'NAME' pour obtenir le nom du joueur
  - 'BOATS' pour obtenir la position des bateaux du joueur
  - 'CELL' pour obtenir la cellule visée par le joueur, suivi une fois la réponse reçue de 'number' pour envoyer le résultat du tir avec 'number' = 0, 1 ou 2 (raté, touché, coulé)
  - 'ATTACK,cell,number' pour indiquer au client que son adversaire a joué, avec 'cell' de la forme : $A1$ et 'number' valant 0, 1 ou 2 (raté, touché, coulé)
  - 'WINNER,winnerName' pour indiquer le nom du gagnant
  - 'END' quand la partie est terminée

Format d'échange à respecter (suite)

- Le client envoie :
  - 'PLAY' pour débuter une partie
  - 'P' ou 'C' suivant s'il veut jouer contre un autre joueur ou l'ordinateur respectivement
  - son nom
  - sa liste de bateaux sous cette forme (sans espace) : *porteAvion* : $J6J10$, *croiseur* : $C1F1$, *etc*.
  - la cellule visée sous cette forme : $A1$