

**TME numéro 4**

(communication via les sockets)

**Exercice 1** (entraînement) Créez une liste de nombres ainsi qu’une fonction qui sauve ces nombres dans un fichier (un nombre par ligne), et une fonction qui lit ce fichier et affiche son contenu à l’écran. Modifiez vos fonctions de manière à faire des lectures/écritures de chaînes de caractères similaires à celles du protocole d’OCAMLPILOT.

**Exercice 2** Reprenez le TME précédent. Écrivez la fonction `get_balles_a_afficher : in_channel -> out_channel -> Jeu.eltJeu list` qui récupère la liste des balles se trouvant sur l’espace de jeu en interrogeant le serveur via les canaux `in_channel` et `out_channel`, et qui renvoie cette liste. Vous vous conformerez au protocole OCAMLPILOT. Vous remplacerez alors l’appel à la fonction `aide_get_balles_a_afficher` que vous aviez fait dans le TME précédent par un appel à votre nouvelle fonction. Vous devez alors obtenir un nouveau client OCAMLPILOT. Pensez bien à effectuer des `flush` pour transmettre vos requêtes au serveur. Testez bien votre application et ne passez pas à l’exercice suivant tant que tout n’est pas correct.

**Exercice 3** Faites de même avec les fonctions `get_vaisseaux_a_afficher`, `get_plateau_a_afficher` et `get_obstacles_a_afficher`. Assurez-vous après chaque écriture de fonction que vous obtenez bien un client qui fonctionne et, surtout, n’essayez pas d’intégrer plusieurs fonctions `get_*` en même temps avant de faire vos tests, c’est le meilleur moyen d’obtenir un programme qui ne fonctionne plus correctement. N’oubliez pas les `flush` ni les commentaires OcamlDoc.

**Exercice 4** Écrivez une fonction `get_couleur : unit -> Graphics.color` qui demande à l’utilisateur quelle couleur il désirerait avoir parmi les suivantes : noir, rouge, vert, bleu, jaune, cyan et magenta et qui renvoie cette couleur (comme un `Graphics.color`, pas comme une chaîne de caractères). Notons que les couleurs citées ci-dessus sont prédéfinies dans le module `Graphics`. Rappelez-vous que, lorsque vous testez dans le « `caml-toplevel` » de xemacs, il faut valider vos couleurs avec la touche « entrée » du pavé numérique.