

**TME numéro 6****(fin de l'écriture du client)**

**Exercice 1** Reprenez le TME précédent et écrivez une fonction :

```
connexion_serveur : unit -> Graphics.color * in_channel * out_channel
```

qui demande à l'utilisateur la couleur qu'il désire avoir pour son vaisseau spatial, qui crée la connexion avec le serveur et qui, si tout s'est bien passé, renvoie un triplet contenant la couleur du vaisseau ainsi que les canaux d'entrée et de sortie permettant de communiquer avec le serveur. En revanche, si la couleur est déjà utilisée (si c'est le cas, le serveur vous l'indiquera), votre fonction fermera proprement les sockets et se terminera. Si une erreur survient lors de la tentative de connexion (problème de réseau, etc) votre fonction terminera votre programme.

**Exercice 2** Écrivez une fonction `main` qui lance la connexion au serveur, initialise les variables de taille, puis boucle indéfiniment sur l'affichage des objets graphiques et l'envoi au serveur des instructions du joueur (ces deux fonctions pourront être lancées à intervalle de 1/30ème de seconde).

**Exercice 3** Vérifiez maintenant tout votre code pour vous assurer qu'il n'y a plus de bug. En particulier, assurez-vous que, lorsque l'on ferme le client ou que la communication avec le serveur est interrompue, votre application se termine en fermant proprement les sockets qu'elle a ouvertes (lancez un `netstat -tau` dans une console pour vous en assurer).

**Exercice 4** Supprimez maintenant le « `open Aide_client` » que vous avez écrit en début de fichier. Votre programme ne fonctionne plus car la fonction `nanosleep` n'est plus connue d'Ocaml. Pour pallier cela, insérez la ligne suivante dans votre fichier ml :

```
external nanosleep : float -> unit = "ocamlpilot_nanosleep";;
```

Cette ligne indique à Ocaml le type de la fonction `nanosleep` et lui précise que c'est en fait la fonction C « `ocamlpilot_nanosleep` » qu'il faut exécuter lorsque l'on appelle `nanosleep`. Recompilez dans `xemacs` : maintenant tout devrait fonctionner.

**Exercice 5 (fin de la partie client)** Soumettez le source (commenté) de votre client sur l'URL :

<http://www-sysdef.lip6.fr/~gonzales/teaching/li260-2005fev/soumission.php>

**Note :** pour obtenir un exécutable de votre client, et en supposant que le source de ce dernier s'appelle `client.ml`, vous pouvez utiliser les commandes suivantes :

```
ocamlc -c jeu.mli
gcc -I /usr/lib/ocaml/ -c ocamlpilot_stub.c
ocamlc -custom -o ocamlpilot unix.cma graphics.cma ocamlpilot_stub.o client.ml
```

Vous pouvez aussi utiliser le `Makefile` qui vous est fourni dans la rubrique « ressources » de la page oùbe du module ou bien dans le répertoire `/Infos/lmd/2004/licence/ue/li260-2005fev/g7/ml` et taper « `make ocamlpilot` ». Le `Makefile` suppose que votre fichier s'appelle `client.ml`.

Cela dit, je rappelle que ce sont les `SOURCES` que vous devez transmettre via l'URL ci-dessus.