

```
/*
 * Fonctions spécifiques à la gestion de l'affichage de la barrière
 */

#ifndef _AFFICHEUR_H
#define _AFFICHEUR_H

#include <Socket.h>
#include <string.h>

// Efface le contenu des deux lignes de l'afficheur
inline
void clear(Socket &barriere)
{
    barriere.envoyer("Afficheur:clear");
    barriere.recevoir();
}

// Place le curseur en haut à gauche de l'afficheur
inline
void home(Socket &barriere)
{
    barriere.envoyer("Afficheur:home");
    barriere.recevoir();
}

// Rendre visible ou pas le curseur de l'afficheur
inline
void setCursor(Socket &barriere, bool visible)
{
    char chaine[30];
    strcpy(chaine, "Afficheur:setCursor:");
    strcat(chaine, (visible ? "true" : "false"));
    barriere.envoyer(chaine);
    barriere.recevoir();
}

// Place le curseur de l'afficheur à l'endroit spécifié
inline
void positionCursor(Socket &barriere, int x, int y)
{
    char chaine[40];
    sprintf(chaine, "Afficheur:positionCursor:%d:%d", x, y);
    barriere.envoyer(chaine);
    barriere.recevoir();
}

// Affiche un message aux coordonnées spécifiées
inline
void affiche(Socket &barriere, int x, int y, char *message)
{
    char chaine[40];
    sprintf(chaine, "Afficheur:affiche:%d:%d:%s", x, y, message);
    barriere.envoyer(chaine);
    barriere.recevoir();
}

// Affiche un caractère à l'endroit où se trouve le curseur de l'afficheur
inline
void afficheCaractere(Socket &barriere, char caractere)
```

```
{
    char chaine[40];
    sprintf(chaine, "Afficheur:afficheCaractere:%c", caractere);
    barriere.envoyer(chaine);
    barriere.recevoir();
}

// Affiche deux messages sur chacune des lignes de l'afficheur
inline
void afficheMessages(Socket &barriere, char *premier, char *deuxieme)
{
    char chaine[80];
    sprintf(chaine, "Afficheur:afficheMessages:%s:%s", premier, deuxieme);
    barriere.envoyer(chaine);
    barriere.recevoir();
}

// Affiche deux messages avec un effacement préliminaire
inline
void affiche(Socket &barriere, char *premier, char *deuxieme)
{
    clear(barriere);
    afficheMessages(barriere, premier, deuxieme);
}

// Affiche deux messages centrés avec un effacement préliminaire
inline
void centre(Socket &barriere, char *premier, char *deuxieme)
{
    clear(barriere);
    int position = (16 - strlen(premier)) / 2;
    affiche(barriere, position, 0, premier);
    position = (16 - strlen(deuxieme)) / 2;
    affiche(barriere, position, 1, deuxieme);
}

#endif
```