

```

1  /**
2   * Gestion de l'Affichage sous forme de fenetre graphique
3   * */
4  #include "moteur_graphique.h"
5
6  /**
7   * Affichage des connexions entre les noeuds
8   * @param graphe
9   * @param objet
10  * @param cercle
11  * @param les noeuds
12  * @param le début du tier
13  * @param la fin du tier
14  * */
15  void afficher_voisin(graphe* G, flame_obj_t * fo, cercle_t * c,int sommet, int deb, int fin);
16
17  /**
18   * Sauvegarde des données dans le tableau cercle pour l'affichage
19   * Chaque tiers ont des couleurs différentes (tier 1 : rouge, tier2 :
20   * bleu, tier 3 : vert)
21   * @param cercle
22   * @param le début du tier
23   * @param la fin du tier
24   * @param x représente les coordonnées en abscisse
25   * @param y représente les coordonnées en ordonnée
26   * */
27  void init_affichage_tier(cercle_t * c, int debut,int fin, int tier,int * x,int * y);
28
29  /**
30   * Permet l'initialisation des objets graphiques:
31   * initialisation des cercles et des connexions des noeuds*
32   * @param graphe
33   * @param objet
34   * @param cercle
35   * */
36  void initialisation_objets_graphique(graphe *G,flame_obj_t * fo,cercle_t * c);
37
38  /**
39   * Recherche l'identifiant du cercle dans la fenetre graphique
40   * @param x abscisse
41   * @param y ordonnée
42   * @return identifiant du cercle
43   * */
44  int trouve_id(int x,int y);
45
46  /**
47   * Affiche le plus court chemin
48   * @param objet
49   * @param la table de routage
50   * @param cercle
51   * @param l'identifiant de l'expediteur
52   * @param l'identifiant du destinataire
53   * @param la couleur
54   * */
55  void affiche_chemin(flame_obj_t * fo,routage* R, cercle_t * c,int deb, int fin,enum couleur coul);

```

```
56
57 /**
58  * Permet la gestion de l'interaction utilisateur et machine
59  * @param graphe
60  * @param objet
61  * @param la table de routage
62  * @param cercle
63  * */
64 void interaction_user(graphe * G,flame_obj_t * fo,routage * R,cercle_t * c);
65
66 /**
67  * Fonction principale
68  * Gestion de la fenetre graphique
69  * @param graphe
70  * @param la table de routage
71  * */
72 void gestion_fenetre_graphique(graphe* G, routage *R);
73
```