INF111 – TP6

Lucas Pastor

1 Exercice 5.6

```c
# include "graphsimple.h"

# define LARGEURW   420
# define TAILLE   4
# define MAILLE   (LARGEURW/(TAILLE+2))
# define DEMIMAILLE   (MAILLE/2)

int EcrireEntier (int p, int q, int r);
int Voisines (int p, int q, int r, int s);
int Marge (int p, int q);
int PixelIndice (int p);
void DessinerGrille (void);
void LigneColonneCliquee (int *L, int *C);
void InitialiserTaquin (int Taq[TAILLE + 2][TAILLE + 2], int *lb, int *cb);
void FlipFlop (int Taq[TAILLE + 2][TAILLE + 2], int *lb, int *cb, int lc, int cc);

int main (void)
{
    int T[TAILLE + 2][TAILLE + 2]; /* tableau-taquin */
    int ligcliq, colcliq; /* ligne et colonne cliquees du maillage:
                             entre 0 et Taille+1 */
    int lz, cz;         /* lz et cz ligne et colonne du zero */

    Initialiser (LARGEURW, LARGEURW);
    DessinerGrille();
    InitialiserTaquin (T, &lz, &cz);

    LigneColonneCliquee (&ligcliq, &colcliq); /* Recuperation premier clique */
    while  (! Marge (ligcliq, colcliq) )
    {
        if (Voisines (ligcliq, colcliq, lz, cz))
            FlipFlop (T, &lz, &cz, ligcliq, colcliq);
        LigneColonneCliquee (&ligcliq, &colcliq);   /*
                                                       Recuperation clique suivant */

    }

    /* Le dernier clique est dans la marge : l'utilisateur souhaite terminer ! */
    return 0;
    Clore ();

}


int EcrireEntier (int lig, int col, int r)
{
    /*
     * Cette procedure recoit deux indices lig et col du maillage et un entier r
     * elle ecrit en position centree dans la case "lig, col" de la fenetre
     * graphique l'entier r.
     * Pour cela elle transforme r en chaine de caracteres et calcule la position du
     * pixel centrale de la case "lig, col" de la fenetre graphique
     * */
    char S[3];
    S[2] = '\0';
    if (r == 0)
    {
```

```
58          S[0] = ' ';
59          S[1] = ' ';
60      }
61      else if (r / 10 == 0)
62      {
63          S[0] = ' ';
64          S[1] = 48 + r;
65      }
66      else
67      {
68          S[0] = 48 + r / 10;
69          S[1] = 48 + r % 10;
70
71      }
72      EcrireDessus (col * MAILLE + DEMIMAILLE, lig * MAILLE  + DEMIMAILLE, S);
73      return 0;
74
75  }
76
77  int Voisines (int p, int q, int r, int s)
78  {
79      /* les entiers p, q, r, s sont des  indices du tableau T */
80      /* vraie si et seulement si les cases (p, q) et (r, s) sont
81       * taquin-adjacentes */
82      return  abs(p - r) + abs(q - s) == 1;
83
84  }
85
86  int Marge (int p, int q)
87  {
88      /* les entiers p, q sont des entiers compris entre 0 et TAILLE+1 */
89      /* vraie si et seulement ce ne sont pas des indices de T */
90      return (p == 0) || (p == TAILLE + 1) || (q == 0) || (q == TAILLE + 1);
91
92  }
93
94  int PixelIndice (int p)
95  {
96      /* Transforme un num de pixel en un entier compris entre 0 et TAILLE+1 */
97      return  p / MAILLE;
98
99  }
100
101 void DessinerGrille (void)
102 {
103     /* Dessine le maillage */
104     int i;
105     for (i = 1; i <= TAILLE + 1; i = i + 1)
106     {
107         Ligne (MAILLE * i, MAILLE, MAILLE * i, MAILLE * (TAILLE + 1));
108         Ligne (MAILLE, MAILLE * i, MAILLE * (TAILLE + 1), MAILLE * i);
109
110     }
111
112 }
113
114 void LigneColonneCliquee (int *L, int *C)
115 {
116     /* Recupere un clique et transforme les coordonnees-pixel recupereres  */
117     /* en un numero de ligne et de colonne du tableau taquin */
118     int Px, Py, bouton;     /* Pixel-x et Pixel-y cliquees */
119     CliquerXY(&Px, &Py, &bouton);
120     *L = PixelIndice (Py);
121     *C = PixelIndice (Px);
122
123 }
124
125 void InitialiserTaquin (int Taq[TAILLE + 2][TAILLE + 2], int *lb, int *cb)
126 {
127     /* Initialisation du tableau associe au Taquin */
128     /* Et affichage des chiffres dans la grille */
```

```
129     int i, j;
130     for (i = 1; i <= TAILLE; i = i + 1)
131         for (j = 1; j <= TAILLE; j = j + 1)
132         {
133             Taq[i][j] = ((i - 1) * TAILLE + j) % (TAILLE * TAILLE);
134             EcrireEntier (i, j, Taq[i][j]);
135 
136         }
137     *lb = TAILLE;
138     *cb = TAILLE;
139 
140 }
141 
142 void FlipFlop (int Taq[TAILLE + 2][TAILLE + 2], int *lb, int *cb, int
143                ligcliq, int colcliq)
144 {
145     /* Les cases (*lb, *cb) et (ligcliq, colcliq) sont Taquin adjacentes */
146     /* Echange de ces deux cases dans le tableau associe au Taquin */
147     /* Et aussi dans le graphique */
148     Taq[*lb][*cb] = Taq[ligcliq][colcliq];
149     Taq[ligcliq][colcliq] = 0;
150     EcrireEntier (ligcliq, colcliq, Taq[ligcliq][colcliq]);
151     EcrireEntier (*lb, *cb, Taq[*lb][*cb]);
152     *lb = ligcliq;
153     *cb = colcliq;
154 
155 }
```