



PROGRAMMATION PROCEDURALE - Année 2019/2020

CLASSE CYCLE D'INGENIEUR ING1G04

INTRODUCTION AU C :

**TP6**

---

Made by :  
Lilian Naretto

```

1 #include "tp6.h"
2 /*!author Lillian Naretto <naretto11@eisti1.eu>
3 \date 10 novembre 2019
4 \file tp6.c
5 \brief tp6
6 \version 0.1 premier jet*/
7 void init(int ttint_plateau[N][N]){
8     int i,v;
9     for ( i = 0; i < N; i++){
10         {
11             for ( v = 0; v < N; v++){
12                 {
13                     ttint_plateau[i][v] = -1;
14                 }
15             }
16         }
17     }
18 }
19
20 void affichage(int ttint_plateau[N][N]){
21     int i,v;
22     printf("  ");
23     for ( i = 0; i < N; i++){
24         {
25             printf(" %d ",i);
26         }
27     }
28     printf("\n");
29
30     for ( i = 0; i < N; i++){
31         {
32             printf(" +-----+-----+-----+\n");
33             printf("%d ",i);
34             for ( v = 0; v < N; v++){
35                 {
36                     if (ttint_plateau[i][v]== -1)
37                     {
38                         printf("|  ");
39                     }
40                     if (ttint_plateau[i][v]==1)
41                     {
42                         printf("| X ");
43                     }
44                     if (ttint_plateau[i][v]==2)
45                     {
46                         printf("| 0 ");
47                     }
48                 }
49             }
50             printf("\n");
51         }
52     }
53     printf(" +-----+-----+-----+\n");
54 }
55
56 int jouer(int ttint_plateau[N][N],int int_joueur, int int_x){
57     int i;
58     int k=N;
59     for (i = 0; i < N+1; i++)
60     {

```

```

25 int jouer(int ttint_plateau[N][N],int int_joueur, int int_x){
26     int i;
27     int k=N;
28     for (i = 0; i < N+1; i++)
29     {
30         if (ttint_plateau[k][int_x]== -1)
31         {
32             if (int_joueur==1)
33             {
34                 ttint_plateau[k][int_x]=1;
35                 return 1;
36             }
37             if (int_joueur==2)
38             {
39                 ttint_plateau[k][int_x]=2;
40                 return 1;
41             }
42             else
43             {
44                 return 0;
45             }
46         }
47         else
48         {
49             if (k<0)
50             {
51                 return 0;
52             }
53             else
54             {
55                 k--;
56             }
57         }
58     }
59     return 0;
60 }
61
62 int agagne(int ttint_plateau[N][N]){
63     int i,v,j,k,m,x,y;
64     int diag;
65     int accum;
66     int val;
67     /* horizontale et verticale */
68     for ( i = 0; i < N; i++)
69     {
70         for ( v = 0; v < N; v++)
71         {
72             if (ttint_plateau[i][v]==1)
73             {
74                 if (ttint_plateau[i][v]==ttint_plateau[i+1][v] && ttint_plateau[i][v]==ttint_plateau[i+2][v] && ttint_plateau[i][v]==ttint_plateau[i+3][v])
75                 {
76                     return 1;
77                 }
78             }
79         }
80     }
81     return 0;
82 }

```

```
115     }
116
117     }
118     if (ttint_plateau[i][v]==2)
119     {
120         if (ttint_plateau[i][v]==ttint_plateau[i+1][v] && ttint_plateau[i][v]==ttint_plateau[i+2][v] && ttint_plateau[i][v]==ttint_plateau[i+3][v])
121         {
122             return 2;
123         }
124         if (ttint_plateau[i][v]==ttint_plateau[i][v+1] && ttint_plateau[i][v]==ttint_plateau[i][v+2] && ttint_plateau[i][v]==ttint_plateau[i][v+3])
125         {
126             return 2;
127         }
128     }
129 }
130
131 }
132
133 /* diagonale droite */
134 k = (N-3);
135 m = k/2;
136
137 for(i = 0; i < k; i++){
138
139     diag = N-abs(m-1);
140
141     x = 0;
142     y = 0;
143     if(i <= k/2){
144         x = k/2-1;
145     } else {
146         y = 1-k/2;
147     }
148
149     accum = 0;
150     val = -1;
151
152     for(j = 0; j < diag ; j++){
153         if(ttint_plateau[x+j][y+j] != -1){
154             if(accum == 0){
155                 val = ttint_plateau[x][y];
156             }
157             if(ttint_plateau[x+j][y+j] != val){
158                 accum = 1;
159                 val = ttint_plateau[x+j][y+j];
160             } else {
161                 accum++;
162                 if(accum == 4){
163                     return (val);
164                 }
165             }
166         } else {
167             accum = 0;
168         }
169     }
170 }
171 }
```

```
172  /* diagonale gauche */
173  for(i = 0; i < k; i++){
174
175      diag = N-abs(m-1);
176
177
178      x = 0;
179      y = N-1;
180
181      if(i <= k/2){
182          x = k/2-1;
183      } else {
184          y = N-1-(i-k/2);
185      }
186
187
188      accum = 0;
189      val = -1;
190
191      for(j = 0; j < diag ; j++){
192          if(ttint_plateau[x+j][y-j] != -1){
193              if(accum == 0){
194                  val = ttint_plateau[x+j][y-j];
195              }
196              if(ttint_plateau[x+j][y-j] != val){
197                  accum = 1;
198                  val = ttint_plateau[x+j][y-j];
199              } else {
200                  accum++;
201                  if(accum == 4){
202                      return (val);
203                  }
204              }
205          } else {
206              accum = 0;
207          }
208      }
209  }
210
211
212  for ( i = 0; i < N; i++)
213  {
214      for ( v = 0; v < N; v++)
215      {
216          if (ttint_plateau[i][v]==-1)
217          {
218              return -1;
219          }
220      }
221  }
222
223  }
224  return 0;
225 }
```

```
227 void tourdejeu (int ttint_plateau[N][N]){
228     int parite = 0;
229     int x;
230     while (agagne(ttint_plateau) !=0 && agagne(ttint_plateau) != 1 && agagne(ttint_plateau)!=2)
231     {
232         if (parite % 2 ==0)
233         {
234             printf("joueur1, donne une colonne :");
235             scanf("%d",&x);
236             if (x>=N)
237             {
238                 printf("mauvais numero, recommence \n");
239                 tourdejeu(ttint_plateau);
240             }
241             else
242             {
243                 if (jouer(ttint_plateau,1,x)==0)
244                 {
245                     printf("Erreur, recommence \n");
246                 }
247                 else
248                 {
249                     affichage(ttint_plateau);
250                     parite++;
251                 }
252             }
253         }
254         else
255         {
256             printf("joueur2, donne une colonne :");
257             scanf("%d",&x);
258             if (x>=N)
259             {
260                 printf("mauvais numero, recommence \n");
261                 tourdejeu(ttint_plateau);
262             }
263             else
264             {
265                 if (jouer(ttint_plateau,2,x)==0)
266                 {
267                     printf("Erreur, recommence \n");
268                 }
269                 else
270                 {
271                     affichage(ttint_plateau);
272                     parite++;
273                 }
274             }
275         }
276     }
277     if (agagne(ttint_plateau)==1)
278     {
279         printf("joueur1 a gagné !\n");
280     }
281     if (agagne(ttint_plateau)==2)
282     {
```

```
284     }
285     if (agagne(ttint_plateau)==0)
286     {
287         printf("match nul !\n");
288     }
289 }
290
291 }
292 /*! \fn int main(int argc, char** argv)
293 \param argc nombre d'arguments en entrée
294 \param argv valeur des arguments en entrée
295 \brief sert à appeller les fonctions nécessaires*/
296 int main(int argc, char** argv){
297     int plateau[N][N];
298     init(plateau);
299     affichage(plateau);
300     tourdejeu(plateau);
301     return 0;
302 }
```

```
1  #ifndef __TP6_H_
2  #define __TP6_H_
3
4  #define N 5
5
6
7  #include <stdio.h> /*Autorise l'emploi de printf et de scanf.*/
8  #include <time.h>
9  #include <stdlib.h>
10 #include <math.h>
11
12 /*! void init(int ttint_plateau[N][N])
13 \param int ttint_plateau[N][N]
14 \brief initialise à -1 la grille de jeu*/
15 void init(int ttint_plateau[N][N]);
16
17 /*! void affichage(int ttint_plateau[N][N])
18 \param int ttint_plateau[N][N]
19 \brief affiche la grille sous forme de print, remplace les valeurs -1 1 et 2*/
20 void affichage(int ttint_plateau[N][N]);
21
22 /*! int jouer(int ttint_tableau[N][N],int int_joueur, int int_x)
23 \param int ttint_tableau[N][N]
24 \param int int_joueur
25 \param int int_x
26 \brief verifie si une case est vide sinon, pose le chiffre du joueur correspondant*/
27 int jouer(int ttint_tableau[N][N],int int_joueur, int int_x);
28
29 /*! agagne(int ttint_plateau[N][N])
30 \param int ttint_tableau[N][N]
31 \brief verifie les combinaisons possibles de victoires*/
32 int agagne(int ttint_plateau[N][N]);
33
34 /*! void tourdejeu (int ttint_plateau[N][N])
35 \param int ttint_tableau[N][N]
36 \brief fait jouer les joueurs*/
37 void tourdejeu (int ttint_plateau[N][N]);
38
39 #endif
```