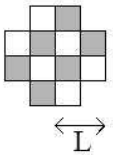
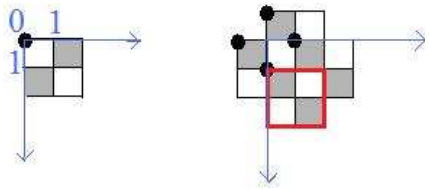
Pavage au hasard d'un diamant aztèque par des dominos

On part d'un échiquier ayant la forme dite de diamant aztèque : comme ici un diamant aztèque d'ordre $L = 2$ (L est sa demi-longueur centrale). On veut le recouvrir par des dominos occupant chacun deux cases horizontales ou deux cases verticales. On sait qu'il existe $2^{L(L+1)/2}$ pavages possibles. Par exemple pour $L = 100$, il y a 2^{5050} cas, soit environ mille milliards de cas.



L'objectif est de construire un de ces pavages au hasard. Une méthode consiste à établir une bijection entre les pavages et les nombres en binaire de longueur $L(L+1)/2$, car ceux-ci sont aussi au nombre de $2^{L(L+1)/2}$, puis de tirer au hasard un de ces nombres en binaire, afin d'en déduire le pavage correspondant.[1] Une autre méthode consiste à fabriquer progressivement un pavage au hasard à partir d'un diamant aztèque d'ordre L=1, puis de passer à l'ordre L=2, en allant ainsi progressivement d'un diamant aztèque d'ordre $L - 1$ à un diamant aztèque d'ordre L, suivant le procédé élaboré par J. Propp puis par T. de la Rue et E. Janvresse.

- On commence par déterminer les pavés carrés occupant deux cases sur deux cases situés à l'intérieur du diamant aztèque d'ordre L, en ne prenant que ceux dont la case en haut à gauche a la même couleur (noire ou blanche) que celle du pavé situé en haut du diamant aztèque. Notons que ces pavés peuvent se chevaucher partiellement, et qu'ils sont au nombre de L^2. Par exemple pour $L = 1$, il existe un pavé unique (la case en haut à gauche ayant par exemple la couleur blanche). Puis on passe au diamant aztèque d'ordre deux, dont la case en haut à gauche sur sa bordure est noire, et l'on prend tous les pavés carrés dont la case en haut à gauche est elle aussi noire, ce qui donne quatre pavés (leur sommet en haut à gauche est dessiné sous forme de disque dans le dessin ci-dessous). Notons que pour les valeurs paires de L, les pavés carrés ont leur case supérieure gauche noire, et que celle-ci est blanche pour les valeurs impaires de L.



Passage du diamant aztèque d'ordre 1, avec un seul pavé carré (case en haut à gauche blanche) au diamant aztèque d'ordre 2, avec quatre pavés carrés dont la case en haut à gauche est noire (un des quatre pavés est indiqué en rouge). On a aussi placé un repère en bleu, dont l'origine est toujours la même, celle prise pour le diamant aztèque initial d'ordre $L = 1$, et dont les graduations sont celles données par les cases.

Voici le morceau de programme qui permet de connaître les coordonnées $x[k]$,$y[k]$ du sommet supérieur gauche de chaque pavé carré (de numéro k) pour le diamant d'ordre L. On utilise le fait que ces points sont disposés sur des parallèles aux deux bissectrices du repère, d'équation $y = – x + cte$ ou $y = x + cte$.

[1] C'est cette méthode qu'a utilisée A. Fathi, *Pavages du diamant aztèque par des dominos*, mémoire de maîtrise, département informatique, Université Paris 8, 2001.

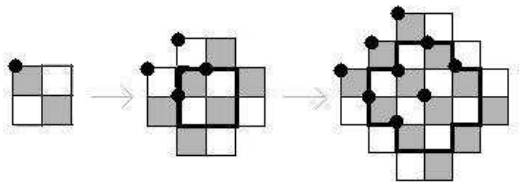
```
k=0;
for(xo=-L+1;xo<=0;xo++)
{ yo=-xo-L+1;
for(xg=xo;xg<xo+L;xg++)  /* point xg, yh ou x[k], y[k] et sur l'écran xeg[k], yeh[k]  */
{ yh=xg+yo-xo;
        x[k]=xg; y[k]=yh;   xeg[k]=xorig+zoom*xg ; yeh[k]=yorig+zoom*yh;
       cercle(xeg[k],yeh[k],4,noir);    k++;
     }
  }
```

Par exemple pour L=2, le programme donne les quatre points (-1,0), (0,1), (0,-1), (1,0).

- Que faire pour passer du pavage du diamant aztèque d'ordre L à celui d'ordre L+1 ?

Pour les pavages, les dominos ont quatre couleurs possibles : rouge ou bleu pour un domino horizontal selon que sa case à gauche est noire ou blanche, jaune ou vert pour un domino vertical selon que sa case en bas est noire ou blanche.

On fait grossir le diamant aztèque de la façon suivante :

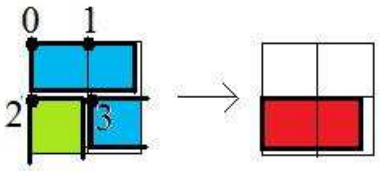


Une fois obtenu un pavage à l'ordre L, on commence par enregistrer les couleurs des points du réseau carré (points à coordonnées entières) dans une large zone autour de l'origine. Les points hors du diamant aztèque ont la couleur fond d'écran (ici noire), et ceux à l'intérieur ont les couleurs des dominos du pavage. Par exemple, on prend les couleurs des points (i,j) de coordonnées comprises entre -10 et +10 (ce qui permet de construire un pavage jusqu'à L=10). Pour cela, on capte la couleur du point correspondant sur l'écran, soit (*xorig+zoom*i*, *yorig+zoom*j*) et on l'enregistre dans $c[i+10][j+10]$ avec ce décalage de 10 de façon que les indices du tableau c soit ≥ 0. Si l'on fait cela, c'est pour conserver ces couleurs lors de la construction du diamant d'ordre suivant, dont les nouvelles couleurs vont dépendre de celles du tableau c, et il ne doit pas y avoir écrasement des anciennes couleurs par les nouvelles tant que le nouveau pavage n'est pas complètement construit.

On prend ensuite chaque pavé carré du nouveau diamant à paver. Il y a trois règles de passage de l'ancien au nouveau pavage :

*Si le pavé est rempli avec deux dominos anciens, on ne fait rien. En fait ces dominos anciens vont être écrasés par de nouveaux provenant des pavés voisins.

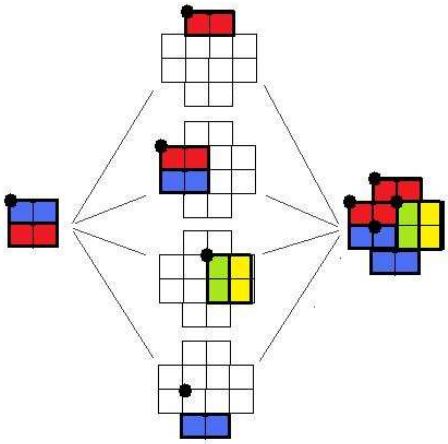
*Si le pavé contient exactement un domino ancien, on le fait coulisser dans la seule autre position possible. L'autre partie sera remplie par le voisinage.



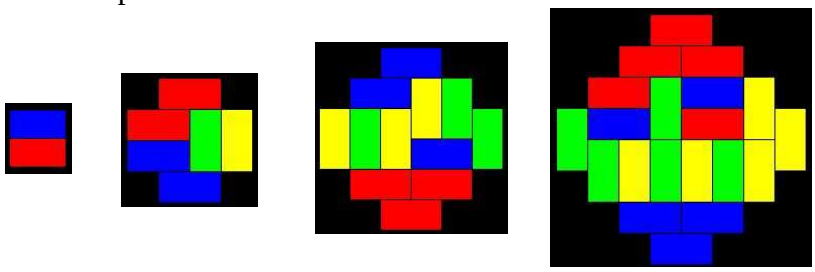
Pour chaque nouveau pavé, on prend les quatre points notés 0, 1, 2, 3. Dans le cas du dessin de gauche on a *c0* = *c1* : les points 0 et 1 ont la même couleur (bleu), mais on n'a pas *c2* = *c3* (les points 2 et 3 n'ont pas la même couleur). Ces couleurs proviennent de l'ancien pavage. Dans le nouveau pavage, on aura encore un domino horizontal, mais il est déplacé d'un cran vers le bas.

*Si le pavé ne contient aucun domino ancien en entier, on le remplit avec deux dominos horizontaux ou deux dominos verticaux, avec une chance sur deux dans chaque cas.

Voici ce que l'on peut obtenir lors du passage de L=1 à L =2, en prenant les quatre pavés à remplir pour L=2 à partir du pavé déjà rempli pour L=1 :



Un exemple d'évolution :



Programme final

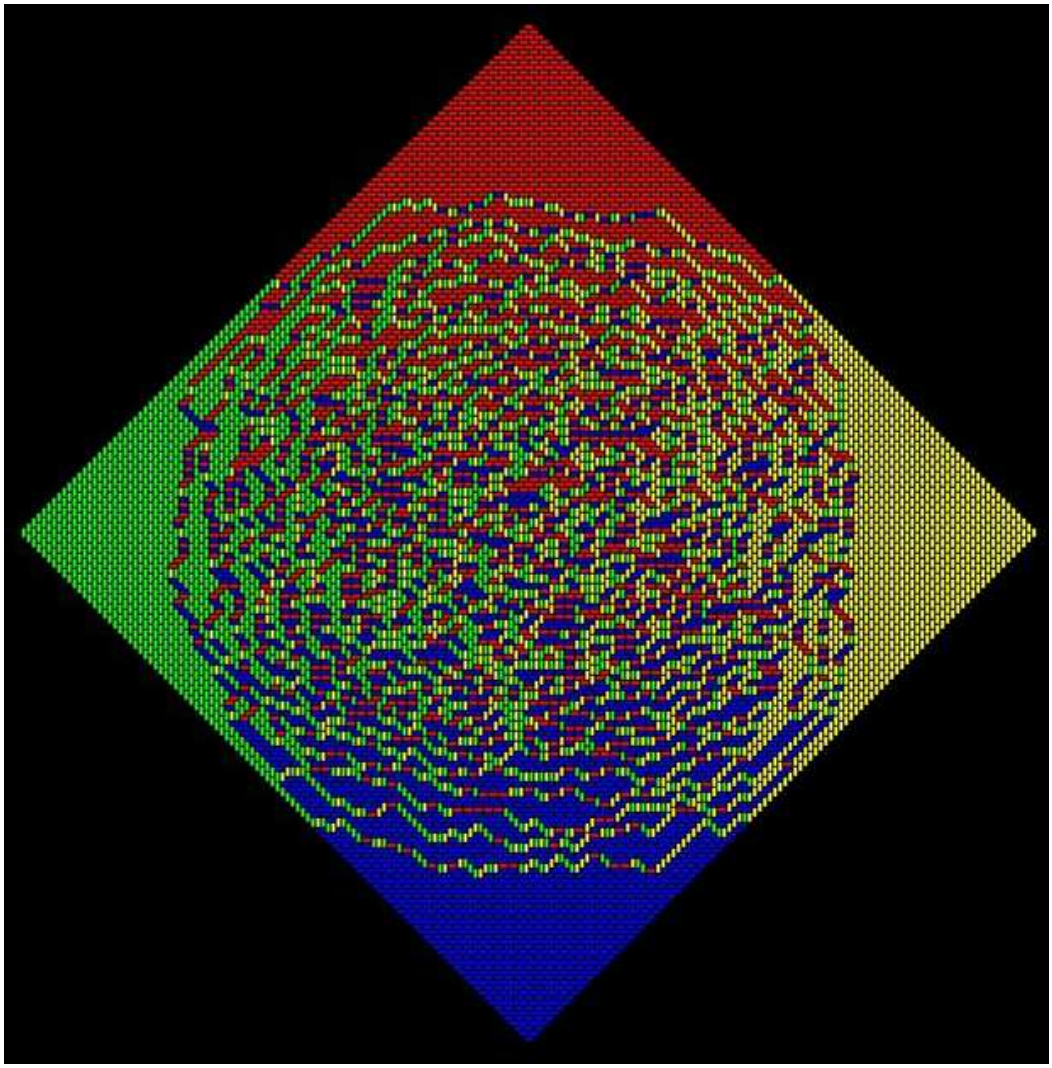
```
#include <SDL/SDL.h>
#include <SDL/SDL_ttf.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define zoom 30
```

```
void pause(void);
void putpixel(int xe, int ye, Uint32 couleur);
Uint32 getpixel(int xe, int ye);
void cercle( int xo, int yo, int R, Uint32 couleur);
void ligne(int x0,int y0, int x1,int y1, Uint32 c);
SDL_Surface * ecran ,*dominoH,*dominoV, *carre;
SDL_Rect position;
Uint32 noir,color[5],c0,c1,c2,c3;
int c[300][300];
int xorig,yorig;
int main(int argc, char ** argv)
{   int i,j,k, L,xo,yo,xg,yh,xeg[10000],yeh[10000],x[10000],y[10000],nbblocs;
    int xx1,yy1, xx2,yy2, xx3,yy3,xe1,ye1, xe2,ye2, xe3,ye3,h;
 SDL_Init(SDL_INIT_VIDEO);
 ecran=SDL_SetVideoMode(800,800,32, SDL_HWSURFACE|SDL_DOUBLEBUF);
 color[4]=SDL_MapRGB(ecran->format,0,0,0);
 color[0]=SDL_MapRGB(ecran->format,255,0,0);
 color[1]=SDL_MapRGB(ecran->format,0,0,255);
 color[2]=SDL_MapRGB(ecran->format,0,255,0);
 color[3]=SDL_MapRGB(ecran->format,255,255,0);
 dominoH=SDL_CreateRGBSurface(SDL_HWSURFACE,2*zoom-1,zoom-1,32,0,0,0,0);
 dominoV=SDL_CreateRGBSurface(SDL_HWSURFACE,zoom-1,2*zoom-1,32,0,0,0,0);
 carre=SDL_CreateRGBSurface(SDL_HWSURFACE,2*zoom-1,2*zoom-1,32,0,0,0,0);
 SDL_FillRect(ecran,0,color[4]);   srand(time(NULL));
 xorig=350; yorig=350;

 for(L=1;L<=10;L++)
  { for(i=-10;i<=10;i++) for(j=-10; j<=10; j++)  /* enregistrement des couleurs */
     if (getpixel(xorig+zoom*i, yorig+zoom*j)==color[4] ) c[i+10][j+10]=4;
     else if (getpixel(xorig+zoom*i, yorig+zoom*j)==color[0]) c[i+10][j+10]=0 ;
     else if (getpixel(xorig+zoom*i, yorig+ zoom*j)==color[1]) c[i+10][j+10]=1 ;
     else if (getpixel(xorig+zoom*i, yorig+ zoom*j)==color[2]) c[i+10][j+10]=2 ;
    else if (getpixel(xorig+zoom*i, yorig+ zoom*j)==color[3]) c[i+10][j+10]=3 ;
    SDL_FillRect(ecran,0,color[4]);
    k=0;   /* points supérieurs gauches des paves carrés */
   for(xo=-L+1;xo<=0;xo++)
    { yo=-xo-L+1;
      for(xg=xo;xg<xo+L;xg++)
       { yh=xg+yo-xo;    x[k]=xg; y[k]=yh;
         xeg[k]=xorig+zoom*xg ; yeh[k]=yorig+zoom*yh;
         k++;
       }
    }
  nbblocs=k;
 for(k=0;k<nbblocs;k++)  /* on prend les paves carrés un par un */
    { xx1=x[k]+1;  yy1= y[k]; xe1=xeg[k]+zoom;  ye1= yeb[k];
      xx2=x[k]  ; yy2=y[k]+1;   xe2=xeg[k]  ; ye2=yeb[k]+zoom;   /* couleurs des  points */
      xx3= x[k]+1; yy3=y[k]+1;  xe3=xeg[k]+zoom  ; ye3=yeb[k]+zoom;          /* 0,1,2,3 */
      c0=color[c[x[k]+10][y[k]+10]];        c1=color[c[xx1+10][yy1+10]]; points
      c2=color[c[xx2+10][yy2+10]];        c3=color[c[xx3+10][yy3+10]];

      if ((c0==color[L%2] && c1==color[L%2]  /* pavé rempli avec 2 dominos */
          && c2==color[(L+1)%2] && c3==color[(L+1)%2])
         || (c0==color[2+L%2] && c1==color[2+(L+1)%2]
           && c2==color[2+L%2]&& c3==color[2+(L+1)%2] ))
```

```
            {continue; }

          else if (c0==color[L%2] && c1==color[L%2]   /* pavé avec un seul domino */
                  && !(c2==color[(L+1)%2] && c3==color[(L+1)%2]))
             { SDL_FillRect(dominoH,NULL,color[(L+1)%2]);
               position.x= xe2; position.y=ye2 ;
               SDL_BlitSurface(dominoH,NULL,ecran,&position);
             }
          else if ( !(c0==color[L%2] && c1==color[L%2])
                  && c2==color[(L+1)%2] && c3==color[(L+1)%2])
             { SDL_FillRect(dominoH,NULL,color[L%2]);
               position.x= xeg[k]; position.y=yeb[k];
              SDL_BlitSurface(dominoH,NULL,ecran,&position);
             }
          else if   (c0==color[2+L%2]  && c2==color[2+L%2]
                   && !(c1==color[2+(L+1)%2]  && c3==color[2+(L+1)%2]))
             { SDL_FillRect(dominoV,NULL,color[2+(L+1)%2]);
               position.x= xe1; position.y=ye1;
              SDL_BlitSurface(dominoV,NULL,ecran,&position);
             }
          else if (!(c0==color[2+L%2] && c2==color[2+L%2])
                   && c1==color[2+(L+1)%2] && c3==color[2+(L+1)%2])
            { SDL_FillRect(dominoV,NULL,color[2+L%2] );
              position.x= xeg[k]; position.y=yeb[k] ;
             SDL_BlitSurface(dominoV,NULL,ecran,&position);
            }
          else                                  /* pavé sans aucun domino entier */
            { h=rand()%2;
              if (h==0)
                { SDL_FillRect(dominoH,NULL,color[L%2] );
                  position.x= xeg[k]; position.y=yeb[k];
                  SDL_BlitSurface(dominoH,NULL,ecran,&position);
                  SDL_FillRect(dominoH,NULL,color[(L+1)%2]);
                  position.x= xe2; position.y=ye2 ;
                  SDL_BlitSurface(dominoH,NULL,ecran,&position);
                }
              else
                { SDL_FillRect(dominoV,NULL,color[2+L%2]);
                  position.x= xeg[k]; position.y=yeb[k] ;
                  SDL_BlitSurface(dominoV,NULL,ecran,&position);
                  SDL_FillRect(dominoV,NULL,color[2+(L+1)%2]);
                  position.x= xe1; position.y=ye1;
                  SDL_BlitSurface(dominoV,NULL,ecran,&position);
                }
            }
        }
      SDL_Flip(ecran);pause();
    }
  pause();return 0;
  }
```

Par une légère modification du programme, on peut aller jusqu'à L=100 (il faudra faire des décalages de 100 au lieu de 10), et le pavage obtenu présente le phénomène dit

du cercle arctique, avec des zones uniformes de dominos identiques aux quatre coins du diamant aztèque, comme si ces zones étaient gelées.



Références bibliographiques :

J. Propp, *Generalized domino-shuffling*, Theoretical Computer Science 303, 2003.

T. de la Rue, E. Janvresse, pavages aléatoires par touillage de dominos, sur le site Images des Mathématiques, CNRS, 2009.

A. Fathi, *Pavages du diamant aztèque par des dominos*, mémoire de maîtrise, département informatique, Université Paris 8, 2001.