LES TABLETTES DE CHOCOLAT

année 2012

PARIS Nicolas et PENGAM Matthieu, 1ère S

Cité Scolaire Môquet-Lenoir

1 Rue de l'Europe 44146 **Chateaubriant**

Enseignant : Mr GREAU

Chercheur : Mr DUCROT

Introduction : règles du jeu

Les règles du jeu sont simples : une tablette, deux joueurs. Chacun son tour, un joueur mange une partie de la tablette en entamant par le coin supérieur droit. Le dernier carreau en bas à gauche est empoisonné. On mange un rectangle composé d'un nombre *n* de carreaux comprenant le coin supérieur droit. Le but du jeu est de contraindre l'adversaire à manger le dernier carreau. (fig. 1)

On ne peut manger que des ensembles de carreaux de forme rectangulaire dans le coin supérieur droit uniquement.

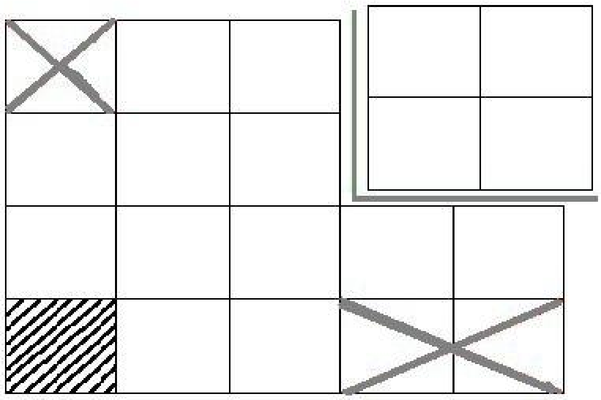

Fig. 1

Résolution de cas simples

Durant nos quelques mois de travail, nous nous sommes d'abord orientés vers la résolution de cas simples tels que des tablettes carrées de 4 cases (2 x 2) ou des tablettes rectangulaires de petite taille. (fig. 2).

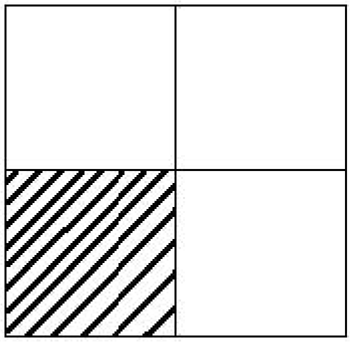

Fig. 2

Nous en sommes arrivés à la conclusion qu' afin de gagner, nous devions faire en sorte qu'à la fin de notre tour, il reste une équerre de 3 carreaux (fig. 3): un carreau au dessus et un carreau à droite du carreau empoisonné, le joueur adverse ne peut prendre qu'un carreau, nous prenons l'autre et ainsi, il se trouve obligé de manger le dernier carreau.

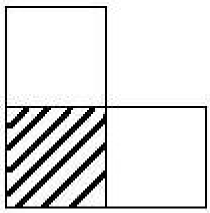

Fig. 3

Par extension, nous pouvons supposer les coups gagnants sur des cas particuliers : toutes les tablettes carrées et rectangulaires de type 2 x *n* (fig. 3)

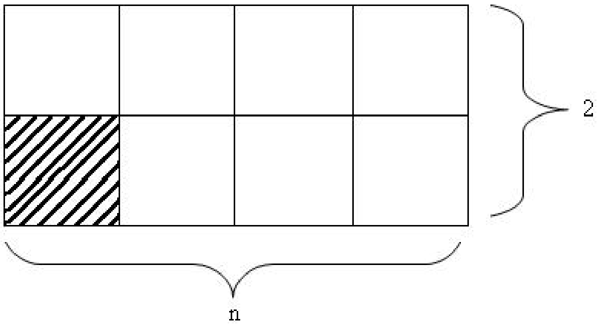

Fig. 3

Pour les tablettes carrées, on commence par prendre le carreau de coordonnées cartésiennes (2;2) (2^{eme} colonne;2^{eme} ligne). Il reste alors une équerre de cotés égaux à notre adversaire. Pour gagner, il suffit alors de jouer symétriquement par rapport à lui. Par exemple, s'il mange le carreau de coordonnées cartésiennes (1;3), on mange le carreau de coordonnées cartésiennes (3;1). On se retrouve alors rapidement dans la situation de la figure 3 et celui-ci finit par devoir manger le dernière carreau.

Pour les tablettes rectangulaires 2 x *n*, il suffit de manger le carreau du coin supérieur de façon à ce que la ligne du bas ait un carreau de plus que la ligne du haut (Fig.5).

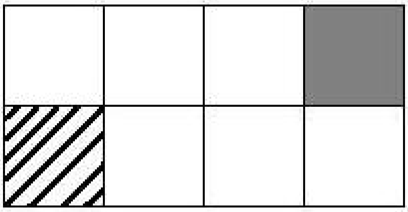

Fig.5

Encore une fois, il faut suivre le jeu de son adversaire afin de conserver cette configuration (fig. 5), il sera rabattu sur cette équerre de 3 carreaux.

Résolution de cas avancés

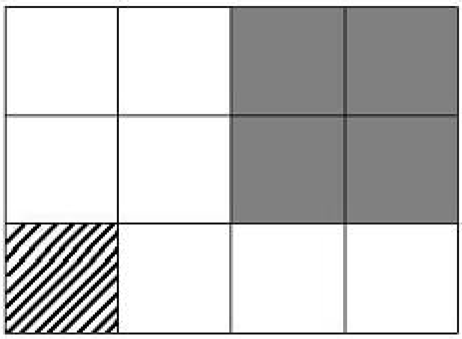

Fig. 6

Il s'agit cependant de cas rares et bien trop simples, nous avons donc décidé de chercher des solutions pour d'autres cas, principalement des tablettes rectangulaires 3 x *n* en commençant par le plus évident : une tablette rectangulaire 3 x 4 (fig. 6).

Après maintes réflexions et parties, nous avons finalement compris comment, dès le premier coup de la partie, nous pouvons prendre l'ascendant sur l'adversaire. En effet, peu importe quel carreau il mange après notre tour, nous arrivons soit sur un carré, soit une équerre 3 x 4, soit un rectangle 3 x 2 (ou 2 x 3). Dans ces cas là, nous sommes assurés de pouvoir gagner en commençant la partie.

Nous avons commencé la résolution d'une tablette rectangulaire 3 x 5 (fig. 7), le nombre de possibilités ayant considérablement augmenté, nous avons procédé par élimination : après avoir éprouvé toutes les entames possible, nous avons conclu qu'il nous faut manger les deux carreaux de la ligne du haut à droite (coordonnées (4 ; 3) et (5;3)). Peu importe la réponse de l'adversaire, nous pouvons rediriger le jeu vers une autre disposition dans laquelle nous sommes gagnants.

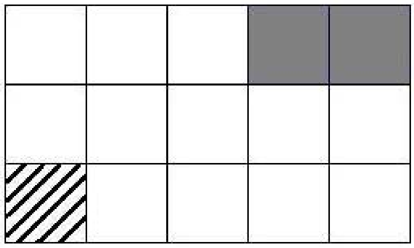

Fig. 7

(1)

Travail informatique

Partis ainsi, nous aurions pu chercher des solutions pour tous les cas possibles, nous avons cependant décidé de créer un programme reproduisant ce jeu à l'aide du logiciel de programmation Algobox. La tache ne fut pas aisée pour autant : nous devions mettre en place un jeu comprenant un affichage de la grille, un système de coordonnées, des interdictions selon les règles et d'autres options facilitant l'utilisation du jeu, comme un compteur noté z qui permet de gérer le tour de chaque joueur (voir l'annexe).

Nous avons commencé par matérialiser le jeu en traçant une grille, il nous fallait donc donner deux nombres *x* et *y* correspondant aux coordonnées d'un point (AlgoBox utilisant un repère orthonormé).

Autour de chaque point sont dessinés 4 segments rouges équidistants du point, ceci donne donc des carreaux qui juxtaposés donnent une tablette.

Afin de savoir si un carreau est mangé, nous avons ajouté un troisième nombre servant d'indicateur : 1 pour intact et 0 pour mangé. Lorsque le carreau est mangé, ses cotés ainsi que ses diagonales apparaissent en vert (Fig. 8).

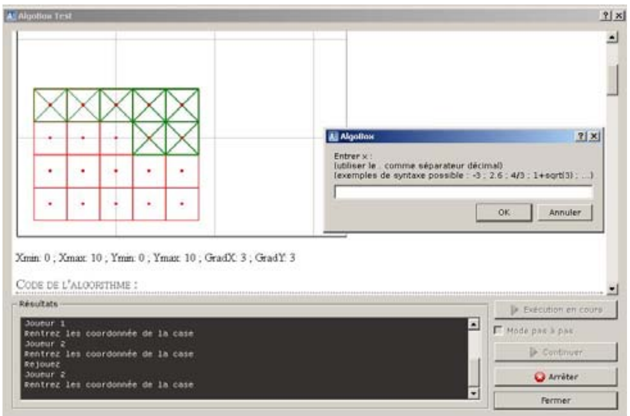

Fig. 8

Un carreau est distingué par une série de 3 nombres et une tablette d'une liste de 3*n* nombres pour *n* carreaux. Pour une tablette rectangulaire 3 x 2, on obtient la liste de 18 nombres suivante: 1,1,1,2,1,1,3,1,1,1,2,1, 2,2,1,2,3,1. Nous avons combiné cette liste avec le premier algorithme de dessin de grille, puis nous avons mis en place le système permettant de manger plusieurs carreaux : l'indicateur passe de 1 à 0 pour le carreau choisi (on entre ses coordonnées) plus les carreaux dont les coordonnées sont toutes les deux supérieures ou égales aux coordonnées entrées.

Bien sûr, ce système présente des inconvénients, il faut à chaque fois calculer soi-même la liste et ajuster le programme en fonction. Nous avons donc mis au point un algorithme créant cette liste selon le nombre de lignes et de colonnes choisi par le joueur.

Combiné avec l'algorithme dernièrement cité, nous avons mis au point un programme dessinant la grille automatiquement et permettant de jouer virtuellement . Les séances suivantes, nous avons amélioré le programme : interdictions, désignation des joueurs grâce à un compteur

Raisonnement type d'une résolution de cas

Pour tous les cas que nous avons résolus mis à part les tablettes carrées, nous avons inversé le problème et cherché la solution à partir de la fin. Ici (Fig. 9) l'illustration de cette méthode de réflexion sur le cas d'une tablette rectangulaire 2 x 4, le joueur 1 commençant et étant gagnant à la fin de la partie.

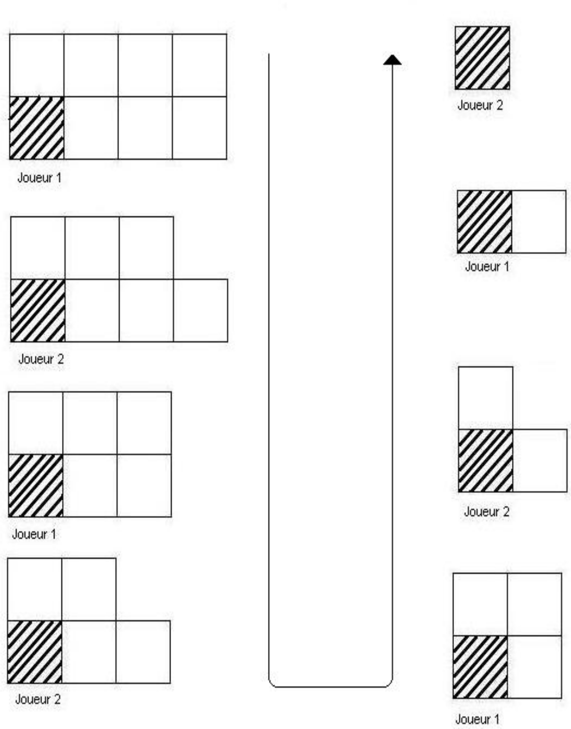

Fig. 9

Nous voyons ici que le joueur 1 est en situation gagnante tout au long de la partie : peut importe le jeu de l'adversaire, il a toujours un moyen de gagner. A l'inverse, le joueur 2 est toujours en situation perdante : quoi qu'il joue, son adversaire a toujours la possibilité de le battre.

Ce type de raisonnement s'appuyant sur des situations perdantes ou gagnantes nous permet de simplifier la problématique de notre sujet : il s'agit de démontrer que la situation de base, la tablette intacte, est une situation gagnante.

Conclusion

Finalement, nous n'avons pas les moyens de démontrer que le joueur commençant une partie de ce jeu est assuré de gagner, tant le nombre de tablettes possibles est important. Nous ne nous sommes penchés que sur un nombre infime de cas possibles et imaginables.

Cependant, M^{r} Ducrot nous a expliqué que celui qui commence a les moyens de gagner, sans savoir à l'avance comment il doit faire. A l'heure actuelle, nous n'avons pas les moyens de prouver l'affirmation de M^{r} Ducrot.

Code du programme Algobox

```
VARIABLES
  i EST_DU_TYPE NOMBRE
  lg EST_DU_TYPE NOMBRE
  a EST_DU_TYPE NOMBRE
  b EST_DU_TYPE NOMBRE
  L EST_DU_TYPE LISTE
  j EST_DU_TYPE NOMBRE
  x EST_DU_TYPE NOMBRE
  y EST_DU_TYPE NOMBRE
  Z EST_DU_TYPE NOMBRE
  v EST_DU_TYPE NOMBRE
  w EST_DU_TYPE NOMBRE
  n EST_DU_TYPE NOMBRE
  m EST_DU_TYPE NOMBRE
  k EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  LIRE n
  LIRE m
  POUR w ALLANT_DE 1 A m
    DEBUT_POUR
    POUR v ALLANT_DE 1 A n
      DEBUT_POUR
      L[(w-1)*3*n+1+3*(v-1)] PREND_LA_VALEUR v
      L[(w-1)*3*n+2+3*(v-1)] PREND_LA_VALEUR w
      L[(w-1)*3*n+3+3*(v-1)] PREND_LA_VALEUR 1
      FIN_POUR
    FIN_POUR
  POUR k ALLANT_DE 1 A 3*n*m
    DEBUT_POUR
    AFFICHER L[k]
    FIN_POUR
  Z PREND_LA_VALEUR 1
  POUR j ALLANT_DE 1 A n*m
    DEBUT_POUR
    SI ((L[3*(j-1)+3])%2==1) ALORS
      DEBUT_SI
      a PREND_LA_VALEUR L[3*(j-1)+1]
      b PREND_LA_VALEUR L[3*(j-1)+2]
      TRACER_SEGMENT (a-1/2,b-1/2)->(a+1/2,b-1/2)
      TRACER_SEGMENT (a-1/2,b+1/2)->(a-1/2,b-1/2)
      TRACER_SEGMENT (a+1/2,b-1/2)->(a+1/2,b+1/2)
      TRACER_SEGMENT (a-
```

```
1/2,b+1/2)->(a+1/2,b+1/2)
          TRACER_POINT (a,b)
          FIN_SI
        FIN_POUR
      PAUSE
      Z PREND_LA_VALEUR 1
      TANT_QUE (L[3]==1) FAIRE
        DEBUT_TANT_QUE
        SI (Z%2==1) ALORS
          DEBUT_SI
          AFFICHER "Joueur 1"
          FIN_SI
        SI (Z%2==0) ALORS
          DEBUT_SI
          AFFICHER "Joueur 2"
          FIN_SI
        AFFICHER "Rentrez les coordonnée de la case"
        LIRE x
        LIRE y
        SI (L[3*n*(y-1)+3*x]==1) ALORS
          DEBUT_SI
          POUR j ALLANT_DE 1 A n*m
            DEBUT_POUR
            SI ((L[3*(j-1)+1])>=x ET (L[3*(j-1)+2])>=y) ALORS
              DEBUT_SI
              L[3*(j-1)+3] PREND_LA_VALEUR 0
              FIN_SI
            FIN_POUR
          POUR j ALLANT_DE 1 A n*m
            DEBUT_POUR
            SI ((L[3*(j-1)+3])%2==0) ALORS
              DEBUT_SI
              a PREND_LA_VALEUR L[3*(j-1)+1]
              b PREND_LA_VALEUR L[3*(j-1)+2]
              TRACER_SEGMENT (a-1/2,b-1/2)->(a+1/2,b-1/2)
              TRACER_SEGMENT (a-1/2,b+1/2)->(a-1/2,b-1/2)
              TRACER_SEGMENT (a+1/2,b-1/2)->(a+1/2,b+1/2)
              TRACER_SEGMENT (a-1/2,b+1/2)->(a+1/2,b+1/2)
              TRACER_SEGMENT (a-1/2,b-1/2)->(a+1/2,b+1/2)
              TRACER_SEGMENT (a-1/2,b+1/2)->(a+1/2,b-1/2)
              TRACER_POINT (a,b)
              FIN_SI
            FIN_POUR
          Z PREND_LA_VALEUR (Z+1)%2
          FIN_SI
        SINON
          DEBUT_SINON
          AFFICHER "Rejouez"
          FIN_SINON
        FIN_TANT_QUE
      AFFICHER "   Game Over !"
      AFFICHER "Victoire du "
      SI (Z%2==1) ALORS
        DEBUT_SI
        AFFICHER "joueur 1"
        FIN_SI
        SINON
          DEBUT_SINON
          AFFICHER "joueur 2"
          FIN_SINON
FIN_ALGORITHME
```

Note de l'édition

(1) En déterminant quelles positions sont gagnantes on peut prouver qu'il existe nécessairement une stratégie gagnante