

Construction – Patrons du cube

Sommaire

Dessiner un patron et connaître les carrés choisis sur la grille	1
Reconnaître un patron de cube dans une grille $n \times n$	3
Application à une grille 4×4	6
Application à une grille 6×6	7
Astuces Cabri Elem.....	7
Annexes :.....	13
Légende :.....	9

Dessiner un patron et connaître les carrés choisis sur la grille

On cherche à savoir si le patron dessiné par l'élève est oui ou non un des 11 patrons du cube. On part donc d'un problème géométrique. Pourtant, Cabri Elem ne possède pas tous les outils nécessaires pour analyser et travailler sur ces objets. De fait, actuellement dans Cabri Elem, les objets géométriques construits par l'élève en cours d'utilisation du cahier ne sont pas préexistants au travail de l'élève. Le concepteur du cahier ne peut donc pas créer de feedback sur ces objets. Il faut donc concevoir un cahier où les éléments utilisés par l'élève pour construire le patron existent effectivement avant le début du travail de l'élève. L'idée est donc de proposer à l'élève un pavage de carrés parmi lesquels il sélectionne ceux qu'il veut utiliser pour fabriquer un patron et de déterminer si la sélection des carrés faite par l'élève est un patron du cube ou non. On transforme alors le problème géométrique en un problème algébrique en codant par des chiffres 0 ou 1 le fait qu'un carré de la grille est utilisé ou pas.

Pour cela, on part d'une grille carrée constituée de petits carrés.

@@@Pour cela, créer une grille de la taille voulue ($n \times n$). Plus la taille de la grille est grande, plus le travail de construction du cahier est long ($n \times n \times 2$ carrés à paramétrer, car il y aura deux grilles de carrés $n \times n$ superposées).

On fait l'explication de la construction de la grille avec les carrés sélectionnables ou pas sur 1 seul carré. Il suffit de reproduire la même procédure pour chaque carré de la grille.

Changer la couleur du carré lorsque l'on clique dessus, c'est à dire que l'élève l'a choisi pour son patron en cours de construction

On veut qu'au début l'élève dispose d'un pavage de carrés d'une seule couleur et lorsque l'élève sélectionne un carré pour le patron, le carré change de couleur. Si l'élève clique à nouveau sur le carré, il redevient de la couleur du pavage initial. Pour cela on crée deux grilles qui seront ensuite

superposées. La première est constituée d'images de carrés¹, la seconde de pièces de tangram carrées.

- Créer un carré (de taille pas trop grande pour pouvoir en mettre $n*n$, ni trop petite pour pouvoir cliquer dessus et obtenir un patron manipulable). On préconise que la grille finale prenne la moitié de la page pour laisser de la place pour insérer des commentaires, images et boutons sur la page.
- Choisir la couleur de ce carré qui sera la couleur du pavage initial puis faire une image de ce carré
- Copier le carré et le coller à côté
- Le « Changer en pièce de tangram »
- Changer la couleur de ce nouveau carré, ce sera la couleur des carrés constituant le patron de l'élève
- On a, à ce stade, un carré (puis une grille de carrés), une image de carré et une pièce de tangram carrée.



- Définir l'action sur les 2 « carrés », l'image rouge et la pièce de tangram orange :
 - o Clic sur l'image rouge entraîne : Masquer l'image et montrer la pièce de tangram orange
 - o Clic sur la pièce de tangram entraîne : Masquer la pièce orange et montrer l'image rouge.
- On peut maintenant superposer les 2 « carrés » puis masquer la pièce de tangram orange qui n'apparaît qu'en 2^{ème}.

On a maintenant un carré qui « change » de couleur lorsque l'on clique dessus (du point de vue de l'utilisateur).

Affecter une valeur au carré qui a été sélectionné

On veut savoir si l'élève a cliqué sur le carré ou non (ce qui signifie qu'il l'a choisi pour faire son patron). Pour cela, on utilise un nombre, initialisé à 0, qui passe à 1 lorsque l'élève a cliqué dessus (sélection du carré). Ce nombre repasse à 0 lorsque que l'élève clique à nouveau sur le carré (désélection du carré). C'est cette procédure qui permet de transformer le problème géométrique du patron en un problème numérique.

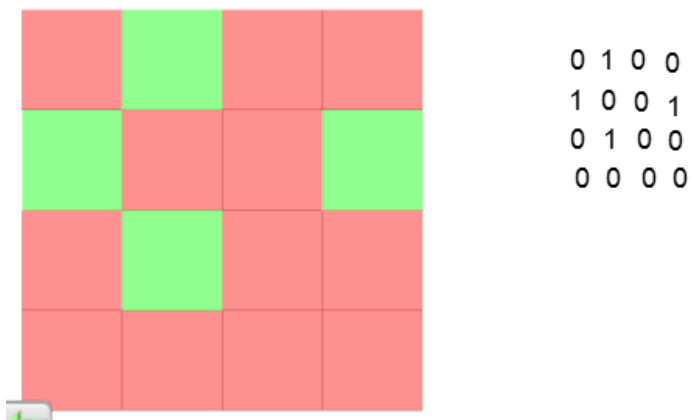
L'explication de la construction se fait toujours pour un seul carré de la grille :

¹ On choisit des images pour les carrés initiaux car cela permet ensuite, lors de l'utilisation de l'outil patron, que les carrés initiaux présents à l'écran ne soient pas pris en compte pour la création du patron.

- Créer un nombre 0 (en faire n^2 pour une grille $n*n$, les ranger en ligne et colonne de façon analogue à la grille proposée à l'élève)
- Ajouter une action sur le « carré » visible au début (image de carré rouge) : « incrémenter la valeur » et sélectionner le nombre qui correspond à ce carré.
- Ajouter une action sur le carré visible après avoir cliqué (pièce de tangram orange) : « décrémenter la valeur » et sélectionner le nombre.

On a donc un nombre qui nous indique suivant sa valeur, 0 ou 1, si le carré a été sélectionné ou non pour dessiner le patron de cube.

Il faut refaire ceci sur toute la grille des $n*n$ carrés. On obtient par exemple cette grille remplie (les pièces de tangram oranges sont vertes dans l'image ci-dessous) avec sa matrice associée remplie de 0 ou 1 selon que les carrés ont été sélectionnés ou pas.

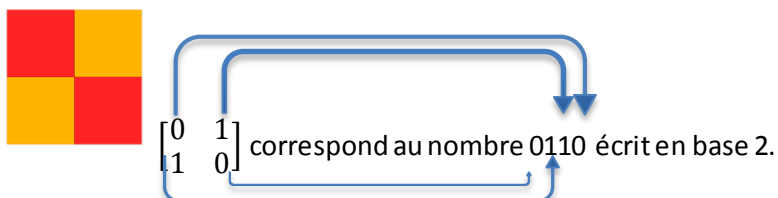


Reconnaître un patron de cube dans une grille $n \times n$

Maintenant que l'on a une grille, une matrice associée avec des 0 ou 1 suivant si les carrés sont sélectionnés, il faut pouvoir analyser le résultat : savoir s'il s'agit d'un patron de cube ou non.

Association d'un nombre unique à chaque combinaison de carrés sélectionnés

Pour cela, on choisit d'associer à chaque valeur de la matrice, une puissance de 2. En fait, on fait correspondre à la matrice la liste des valeurs 0 et 1, c'est à dire un nombre écrit en base 2, ce nombre étant unique pour chaque matrice.



A la 1^{ère} ligne, 1^{ère} colonne, le chiffre des unités 2^0 , écrit tout à droite dans l'écriture linéaire.

1^{ère} ligne, 2^{ème} colonne le chiffre de l'ordre juste après : 2^1

Et ainsi de suite sur toute la ligne. Puis on passe à la ligne suivante, on multiplie par 2^n (n la taille de la grille). Ici le principe sur une grille 2×2 .

$$\begin{bmatrix} 2^0 & 2^1 \\ 2^2 & 2^3 \end{bmatrix}$$

On peut alors calculer ce nombre en base 10 avec l'expression :

$$a + 2 \times (b + 2 \times (c + \dots))$$

Cependant, dans le cas de grilles de taille importante, on obtient des formules avec par exemple 36 variables (pour une grille 6*6). Du coup, pour Cabri Elem, on fait le calcul par ligne de la matrice et on calculera le résultat final en appliquant une autre formule analogue aux résultats obtenus pour chaque ligne (voir plus bas). Exemple de la formule permettant de calculer du nombre associé à une ligne pour une grille 6*6 :

$$a + 2 \times \left(b + 2 \times \left(c + 2 \times \left(d + 2 \times \left(e + 2 \times f \right) \right) \right) \right)$$

Plutôt que d'avoir la formule en somme, on factorise pour avoir moins de calculs et on lit la ligne de droite à gauche. Cette opération est répétée sur chaque ligne.

Enfin, pour calculer le résultat final, on sait qu'il y a entre chaque ligne, un facteur 2^n . On calcule donc l'expression :

$$a + 2^n \times (b + 2^n \times (c + \dots))$$

$$a + 2^6 \times \left(b + 2^6 \times \left(c + 2^6 \times \left(d + 2^6 \times \left(e + 2^6 \times f \right) \right) \right) \right) \text{ (exemple en grille } 6 \times 6)$$

On obtient enfin un résultat en base 10, qui caractérise de façon unique le patron qui est construit par l'élève. On va alors le comparer avec toutes les valeurs de chaque patron pour savoir d'abord s'il s'agit d'un patron, et ensuite savoir de quel patron il s'agit.

[Trouver tous les nombres associés à un patron et ses différentes positions dans une grille n x n](#)

Il existe 11 patrons de cube : 10 qui entrent dans une grille 4*3 (longueur 4 carrés et largeur 3) et un 11^{ème} qui entre dans une grille 5*2. Il est donc impossible de construire ce 11^{ème} patron sur une grille 4 x 4. Pourtant, avant d'arriver à configurer une plus grande grille, il a fallu s'assurer de la validité de la solution choisie et donc prendre une grille 4 x 4 : la grille carrée la plus petite contenant au moins 1 patron du cube.

Néanmoins, pour effectuer les calculs, on utilisera le script où l'on peut choisir n (>=4).

On se base sur le principe qu'une grille carrée est invariante pour 8 transformations : l'identité, la symétrie d'axe horizontale, symétrie d'axe verticale, deux symétries d'axe oblique, une symétrie centrale, deux rotations de 90° en direct et indirect. Ces transformations permettent d'obtenir plusieurs positions d'un même patron dans la grille. Il ne faut néanmoins pas oublier que certaines translations, bien que ne laissant pas invariante la grille, permettent également de déplacer le patron dans la grille. Il faut donc également les prendre en compte. Le nombre de translations dépend de l'occupation du patron dans la grille (par exemple un patron de dimension 4*3 dans une grille n*n

peut être translaté par (n-4) translations horizontalement et (n-3) translations verticalement. On a donc :

$$8 \times \text{nombre de translations} = \text{nombre de positions possibles du patron}$$

Dans le cas concret, on obtient moins de positions différentes car certaines sont obtenues plusieurs fois par des compositions de transformations différentes.

Il a fallu ensuite caractériser les différentes transformations par leur action sur la matrice déterminant les carrés sélectionnés ou pas. Pour cela, il nous faut juste la matrice M, une matrice diagonale n,n :

$$M = \begin{bmatrix} 0 & \dots & 1 \\ \vdots & 1 & \vdots \\ 1 & \dots & 0 \end{bmatrix}$$

En effet, avec une matrice A n,n quelconque, on retrouve les 8 transformations par :

'Symétrie horizontale'

$$MA$$

'Symétrie verticale'

$$AM$$

'Rotation 180 ° = symétrie centrale'

$$MAM$$

'Symétrie axiale \'

tA

'Symétrie axiale /'

$$M {}^t(MA)$$

'Rotation +90°'

$$M {}^tA$$

'Rotation -90°'

$${}^tA M$$

On a alors élaboré un script matlab permettant de produire la liste des nombres en base 10 correspondants à un patron donné dans une position donnée dans une grille n*n.

Le script matlab se divise alors en plusieurs parties :

- 1) Choisir la dimension de la grille

- 2) Créer une matrice nous permettant de stocker les 8 différentes positions initiales de chaque patron (celles obtenues pas les transformations qui laissent la grille globalement invariante).
- 3) Entrer les 11 patrons dans cette matrice
- 4) Trouver toutes les translations possibles et stocker ces positions dans cette grande matrice
- 5) Travailler sur chaque patron 1 par 1
 - a. Définir le patron utilisé
 - b. Utiliser une matrice pour stocker toutes les positions possibles d'un patron
 - c. Extraire des positions, le nombre en base 10 associé
 - d. Enlever les doublons et les ranger dans l'ordre (décroissant ici)
 - e. Afficher pour chaque patron, la liste de tous les numéros possibles
 - f. Remarque : le script MathLab propose, en plus de la liste des n° de patrons possibles, une formule qui utilise ces n° et qui permettra de calculer directement le n° d'un des 11 patrons utilisé. Cela permet d'accélérer le travail en automatisant la création de la formule (à copier directement dans Cabri Elem Creator)

Remarque sur le transfert du résultat de matlab dans Cabri Elem :

- A partir de n=6, les nombres en base 10 deviennent grand (>10^10). Il faut donc choisir d'afficher des flottants (nombres à virgules) qui prennent en considérations plus de chiffres afin d'obtenir le nombre exacte et non une approximation.

Créer la formule qui permet de déterminé lequel des 11 patrons est construit par l'élève

Lorsque l'on dispose de la liste des n° de positions de patrons (dans une grille 6*6 cela peut faire jusqu'à 528 numéros différents), on ne peut pas comparer le numéro du patron construit par l'élève à chacun de ces 528 numéros, en utilisant un booléen « = ».

En revanche, on peut calculer la différence entre N le nombre du patron crée par l'élève avec chaque numéro et si cette différence est nulle, alors c'est que qu'on a identifié le patron construit par l'élève. De plus, on peut faire le produit de ces différences, qui sera nul dès qu'un patron correct est construit par l'élève.

Pour le patron p_i ($i=1$ à 11), il y a 48 numéros p_{ij} ($j=1$ à 48). La formule suivante est nulle dès que l'élève a construit le patron dans une des 48 positions possibles :

$$\prod_{j=1}^{48} (N - p_{ij})$$

On peut alors produire 11 formules, une par patron qui produit 0 dès qu'un des patrons est construit par l'élève.

Remarque sur le transfert des 11 formules de matlab dans Cabri Elem :

- Cabri Elem Creator a parfois du mal à évaluer une expression très longue. On propose de « découper en plusieurs morceaux » la formule produite par le script matlab (exemple : $n=6$, les expressions sont découpées en soit 2 ou 3 morceaux. La multiplication de ces nombres est faite ensuite).

Application à une grille 4 x 4

Fichiers concernés :

- Patrons-cube-v9.clmc
- Page 1 : Patrons-cube-v10.clmc

Comme pour l'explication du script, c'est sur la grille 6 x 6 que se feront les explications.

- Voir aussi l'annexe de ce document : Script matlab grille 4 x 4

Application à une grille 6 x 6

Fichier concernés :

- Patrons-cube-v10.clmc les pages 2 – 3
- Patrons-cube-v12.clmc
- L'annexe de ce document : Script matlab grille n x n

Conception de la page Cabri Elem, avec possibilité de construction de patrons et feedback (identification des patrons construits et nombre de patrons différents trouvés)

Dans cette page l'élève peut :

- choisir des carrés dans la grille en cliquant dessus
- fabriquer le patron avec les carrés sélectionnés et le replier
- demander l'évaluation de son patron
 - o s'il s'agit d'un patron du cube, alors une image de ce patron apparaît
 - o s'il ne s'agit pas d'un patron du cube, des messages indiquant s'il manque ou s'il y a trop de carré apparaissent
 - o s'il s'agit d'un patron déjà trouvé, alors un message le lui dit
 - o affichage du nombre de patrons déjà trouvés
 - o affichage d'un smiley si 11 patrons distincts sont trouvés
- demander de l'aide pour manipuler les outils

Légende :

1) Boutons de construction : Masquer / Afficher les objets

Il s'agit d'une zone permettant de faciliter la création de la page. Ils permettent de masquer ou montrer les objets donc le nom se trouve juste à droite. Ils n'ont pas de fonctionnalité lorsque l'élève travaille.

2) Zone de calcul du patron en cours de construction

Les valeurs de la matrice 6 x 6 sont initialisées à 0. Elles valent 1 dès que le carré est sélectionné.

La première case correspond à 2^0 , la deuxième 2^1 ... la dernière $2^{(6 \times 6 - 1)}$. On fait la somme pour obtenir le résultat unique caractérisant l'état de la matrice suivant les cases cochées, c'est à dire déterminant le numéro du patron construit par l'élève.

3) Zone de formules

On trouve les formules pour les calculs intermédiaires permettant de paramétrer les booléens et faire des tests avec les nombres de la page.

4) Paramètre taille image patrons

Les images des 11 patrons qui s'affichent sur la partie droite de la page du cahier au fur et à mesure que l'élève les construit. Il est possible de faire varier leur taille en modifiant le nombre devant : taille patron. Le nombre suivant correspond à la taille du patron 11 qui est plus long d'un carré et dont la longueur est calculé par $5/4 * (\text{taille patron})$.

Pour obtenir l'affichage de ces images avec une taille modifiable, les images-patrons sont collées sur des segments dont la longueur varie. 10 patrons utilisent la même longueur de segment, le 11^{ème} doit être sur un segment de longueur $5/4$ du segment des précédents.

5) Booléens pour reconnaître les patrons de cube et afficher les messages correspondants

Il s'agit du nerf central de la page du cahier.

Il faut distinguer 2 parties : la partie dynamique à gauche des pointillés, c'est à dire qui s'actualise à chaque action de l'élève et la partie « statique » à droite, qui n'est modifiée que lorsqu'une demande de vérification est faite par l'utilisateur, elle garde en mémoire les résultats déjà obtenus.

- Décrémenter le booléen « vrai si demande de vérification » afin « d’informer » l’ordinateur que toutes les actions ont été effectuées.

d. Vrai si trouvé

	Vrai si trouvé
1	Faux
2	Faux
3	Faux
4	Faux
5	Faux
6	Faux
7	Faux
8	Faux
9	Faux
10	Faux
11	Faux

Cette colonne correspond à l’état des recherches sur les patrons du cube par l’utilisateur. Le passage à vrai de l’un de ces booléens déclenche l’affichage de l’image du patron correspondant. Cela permet de garder un visuel des divers patrons trouvés par l’utilisateur.

e. Vrai si trouvé, reconnu et demande vérification

Il s’agit du dernier cas possible. Le patron est reconnu par l’ordinateur, l’utilisateur a demandé de vérifier s’il s’agissait d’un patron mais le patron en question a déjà été trouvé précédemment. Ce booléen est donc créé comme un ET logique des booléens :

- Vrai si demande vérification
- Vrai si reconnu
- Vrai si trouvé

Le passage à vrai de l’un de ces booléen déclenche l’affichage du message « Ce patron a déjà été dessiné ».

Remarque : le booléen « Vrai si 1 des 11 patrons trouvé, reconnu et demande vérification » a été créé afin de faciliter la mise en place de l’action. En effet, c’est dès qu’un seul des patrons a été de nouveau validé qu’il faut afficher le message et décrémenter le booléen « Vrai si demande vérification ». Ainsi, ce booléen créé comme un OU logique sur les 11 booléens permet de n’avoir qu’une seule action à créer et non 11.

Vrai si -trouvé -reconnu -demande vérification	
Faux	Vrai si 1 des 11 patrons :
Faux	-trouvé
Faux	-reconnu
Faux	-demande vérification
Faux	Faux
Faux	
Faux	
Faux	
Faux	
Faux	
Faux	

6) Grille de sélection

C’est ici que l’utilisateur va sélectionner les carrés qu’il souhaite utiliser afin de faire son patron de cube.

7) Images des patrons trouvés

Il s’agit d’une partie de la page réservée pour l’affichage des 11 images de patron de cube afin de garder un visuel pour l’utilisateur de son avancement dans la découverte des patrons.

8) Boutons + feedback

En lisant de gauche à droite.

On commence par le bouton de validation. Lorsque l’on clique dessus, cela incrémente le booléen : « Vrai si demande vérification ». Tous les booléens de la partie statique dépendent en amont de ce booléen. Cela va en quelque sorte lancer l’analyse des résultats.

Juste à droite se trouvent les phrases feedback qui aident l'utilisateur. Savoir que son patron n'est pas correct, qu'il l'a déjà dessiné, qu'il manque des carrés pour arriver aux 6 nécessaires à un patron du cube.

9) Connaître nombre de patrons trouvés

On initialise un nombre à 0. Celui-ci est incrémenté à chaque fois qu'un nouveau patron est trouvé. Lorsque celui-ci arrive à 11, cela commande l'affichage d'un grand smiley et l'on masque le reste des commandes.

10) Paramétrer les messages d'aides

Dans cette partie, on gère l'affichage de messages en fonction des réalisations de l'élève qui ne correspondent pas à un patron. Si la sélection de carrés faite par l'élève ne correspond pas à un patron, il faut choisir entre 3 messages :

1. il manque des faces (ou carrés ?) : ton patron contient-il le bon nombre de faces carrées ?
2. il y en a trop de faces (ou carrés ?) : ton patron contient-il le bon nombre de faces carrées ?
3. les faces sont mal placées (ou les carrés sont mal placés) : « Les carrés sont-ils bien positionnés? ».

Le booléen « Vrai si patron reconnu » est créé comme un OU logique sur tous les booléens « Vrai si patron X reconnu ».

Le booléen « Vrai si non reconnu » est créé comme son NON logique.

Le booléen « Vrai si non reconnu et demande vérification » est créé comme le ET logique entre « Vrai si demande vérification » et « Vrai si non reconnu ». Il passe donc à vrai lorsque l'utilisateur a dessiné un objet qui n'est pas un patron de cube mais qu'il a demandé de vérifier.

En parallèle, on a calculé le nombre de cases cochées comme la somme de toutes les valeurs de la matrice. 3 booléens indiquent la comparaison avec 6, le nombre de faces carrées d'un cube :

- Vrai si égale
- Vrai si inférieur
- Vrai si supérieur

Il est alors possible de créer les booléens contrôlant l'affichage des messages d'aide en créant les 3 booléens issus d'un Et logique entre « Vrai si Egal » (resp. inférieur ou supérieur) et le booléen « Vrai si non reconnu » et demande vérification pour afficher le message : « Les carrés sont-ils bien positionnés? ».

Remarque :

- L'ajout d'un chrono permet d'effacer le message au bout de quelques secondes.
- Lorsque l'on clique sur un message, celui-ci s'efface aussi dans le but de libérer de la place sur l'écran pour manipuler les objets tranquillement.

11) Formules

Il s'agit des 11 formules obtenues à partir du script matlab pour chaque patron.

12) Formules évaluées avec le résultat de la matrice

Chaque formule est évaluée avec comme variable le nombre correspondant à la sélection de carrés faite par l'élève (le « résultat » de la matrice). On notera la nécessité de séparer les formules en 2-3 parties pour soulager Cabri Elem et ainsi faire une bonne évaluation de la formule.

Si le résultat est 0, alors les carrés sélectionnés forment un patron du cube.

Annexes :

- Script matlab grille 4 x 4
- Script matlab grille n x n