



Supergeotemple

Le succès est un mauvais professeur : il pousse les gens intelligents à croire qu'ils sont infaillibles. (Bill Gates)

J'ai rencontré le puzzle Geotemple il y a vingt ans alors que je commençais à m'intéresser au Tangram. Comme l'indiquent ses inventeurs, de l'équipe du laboratoire d'Archimèdes¹ à Lausanne ce puzzle est « composé de dix pièces géométriques : deux petits triangles rectangles, six polygones convexes ayant la même aire, deux grands triangles rectangles ».

En réalité toutes les pièces de ce jeu sont constituées à partir de petits triangles rectangles qui sont des Demi-Triangles Équilatéraux (DTE dans la suite du texte) : les petites pièces triangulaires sont formées d'un seul DTE, les six polygones convexes de même aire sont les six assemblages possibles de deux de ces DTE et les deux grands triangles rectangles sont formés chacun de trois DTE selon un assemblage qui réalise un agrandissement du DTE de base de coefficient $\sqrt{3}$ (les côtés du DTE sont multipliés par $\sqrt{3} \approx 1,732$). Cette propriété est due aux angles du DTE (30° , 60° et 90°) : deux angles de 30° juxtaposés réalisent l'angle de 60° et trois angles de 60° forment un angle plat.

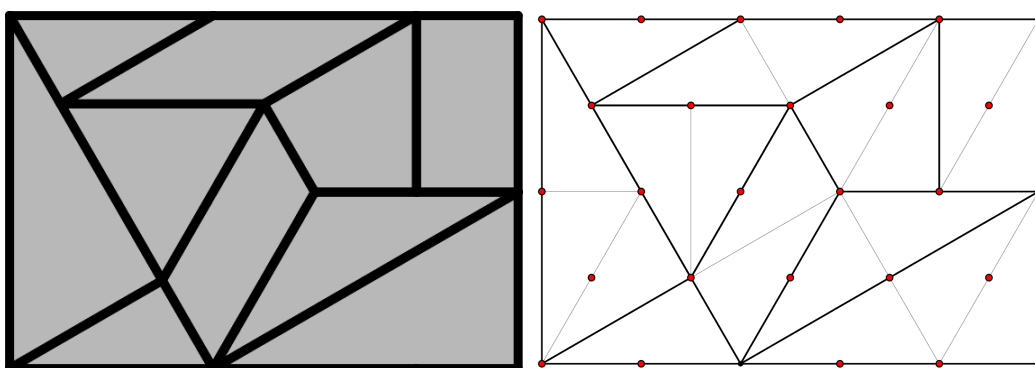


FIGURE 6.1 – Le puzzle Geotemple assemblé ici en rectangle. Avec la grille triangulaire sous-jacente (à droite), on distingue mieux que les pièces sont formées d'assemblages de DTE (Demi-Triangles Équilatéraux).

Ce puzzle original a certainement de nombreuses applications pédagogiques et récréatives comme le signalent ses auteurs. Il a déjà le mérite que je qualifierai d'« esthétique », d'offrir la collection complète des six assemblages de deux DTE. L'adjonction des deux petits et des deux grands triangles peut paraître un peu arbitraire mais elle permet de compléter le jeu de manière à réaliser une figure convexe simple – un rectangle $5 \times 2\sqrt{3}$ – totalisant vingt DTE à l'aide des dix pièces. C'est aussi un choix qui fait écho à la constitution du Tangram, ce jeu comprenant la collection complète des trois assemblages de deux demi-carrés à laquelle est ajoutée deux petits et deux grands demi-carrés qui sont, eux, composés de quatre demi-carrés.

Ma curiosité pour ces assemblages de DTE m'a amené à explorer ceux composés de trois DTE. Je connaissais déjà le grand demi-triangle équilatéral, utilisé dans le Geotemple, et je découvre rapidement neuf autres formes possibles (voir les illustrations ci-dessous) réparties en cinq formes convexes et quatre non convexes.

1. Le puzzle Geotemple (©1992-98) a été inventé par Gianni A. Sarcone et Marie-Joe Waeber, laboratoire d'Archimèdes, Lausanne. Consulter la page « The amazing geotemple puzzle » sur leur site archimedes-lab.org

1. Présentation du Supergeotemple

a. Le jeu de pièces

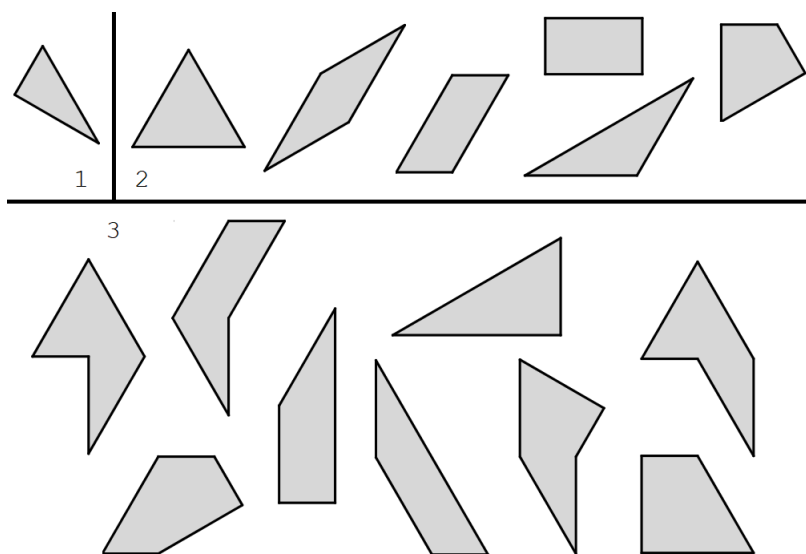


FIGURE 6.2 – Les quatorze pièces du jeu sont formées de un, deux ou trois demi-triangles équilatéraux possibles.

Ces neuf polygones totalisent une aire totale de 3×9 soit 27 DTE. Ce nombre ne permet pas de réaliser un rectangle avec ces seules pièces (car pour un rectangle il faut au moins un nombre pair de DTE), par contre il en fait de bons candidats au problème de la triplification. En effet, si l'on veut réaliser un de ces polygones agrandi trois fois, on doit utiliser 3×3^2 soit 27 DTE. Théoriquement, il est donc envisageable de réaliser un agrandissement de l'un d'eux avec ces neuf pièces. C'est un beau défi que j'ai essayé de relever. Malheureusement, après de nombreux essais, il m'apparaît décevant car impossible à réaliser. C'est mon intime conviction puisque je n'ai trouvé aucune solution mais pour avoir une réponse définitive à ce sujet, il me faudrait programmer mon ordinateur...

Si la collection de ces neuf polygones constitués de trois DTE – je les appellerai désormais triDTE, comme je dirai biDTE pour les six composés de deux DTE – ne semble pas conduire à un jeu complet², il faut envisager de la compléter avec d'autres pièces, comme l'ont fait les auteurs du Geotemple en ajoutant quatre pièces additionnelles au lot des six biDTE. Mon choix s'est porté naturellement sur ces biDTE ainsi que l'unique DTE de base.

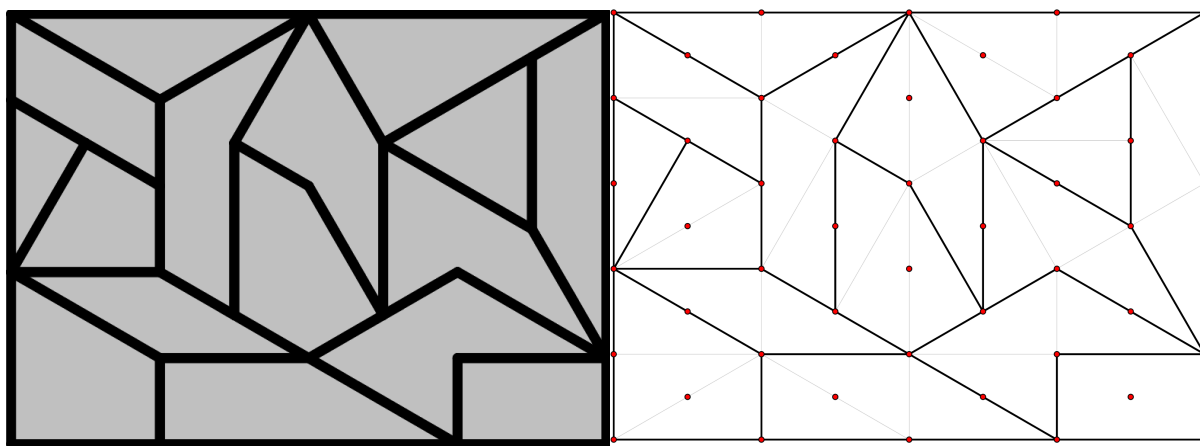


FIGURE 6.3 – Le puzzle Supergeotemple, avec ses seize pièces assemblées en rectangle. Le même avec la grille triangulaire sous-jacente et des sous-divisions montrant les 40 DTE du jeu (à droite).

2. Quelques formes convexes sont réalisables avec cet ensemble de 9 pièces et quelques amusements sont possibles comme par exemple essayer de retrouver chacun des polygones agrandis 2 fois (donc avec 4 pièces uniquement)...

Lorsqu'on regroupe ces seize polygones ($9 + 6 + 1 = 16$), on obtient un ensemble totalisant une aire de $27 + 12 + 1 = 40$ DTE. Comme ce nombre est pair, il n'est pas insensé de chercher à réaliser un rectangle avec ces pièces. C'est ce que j'ai essayé de faire et, grâce à la souplesse apportée par les petits polygones, j'ai vite trouvé ce que je cherchais. La joie de cette découverte efface en partie la frustration de n'avoir pas su tripler un triDTE. L'illustration montre cette solution que j'ai baptisé Supergeotemplet en l'honneur du Geotemplet de l'équipe d'Archimèdes-lab.

Je dois faire ici une incursion dans cette histoire : après une dizaine d'années passées sans avancée notable sur ce jeu, j'ai repris ce travail avec la ferme intention d'obtenir de plus amples résultats. Je m'aperçois alors que j'ai oublié ou mis de côté une douzaine de triDTE (les voir en annexe 1.4.2). La collection ainsi complétée en contient vingt-et-un et, avec les six biDTE et le monoDTE, cela constitue un jeu de 28 pièces totalisant 76 DTE que j'appellerai l'HyperGeotemplet. Cette parenthèse étant refermée, revenons au SuperGeotemplet.

b. Formes à reconstituer

Autres rectangles

Le rectangle obtenu n'est pas le seul qu'il soit possible de réaliser avec le Supergeotemplet. En considérant qu'on regroupe les DTE deux par deux pour former de petits rectangles, on a vingt petits rectangles à assembler pour former un grand rectangle. En omettant les rectangles de une unité de côté qui sont trop étroits pour contenir certaines pièces, on peut répartir les petits rectangles en deux rangées de dix, quatre rangées de cinq, cinq rangées de quatre ou dix rangées de deux. Cela conduit à quatre types de grands rectangles potentiellement payables avec ce jeu de pièces. Celui que j'ai découvert en premier est le plus compact, ses dimensions étant les plus proches de celles d'un carré.

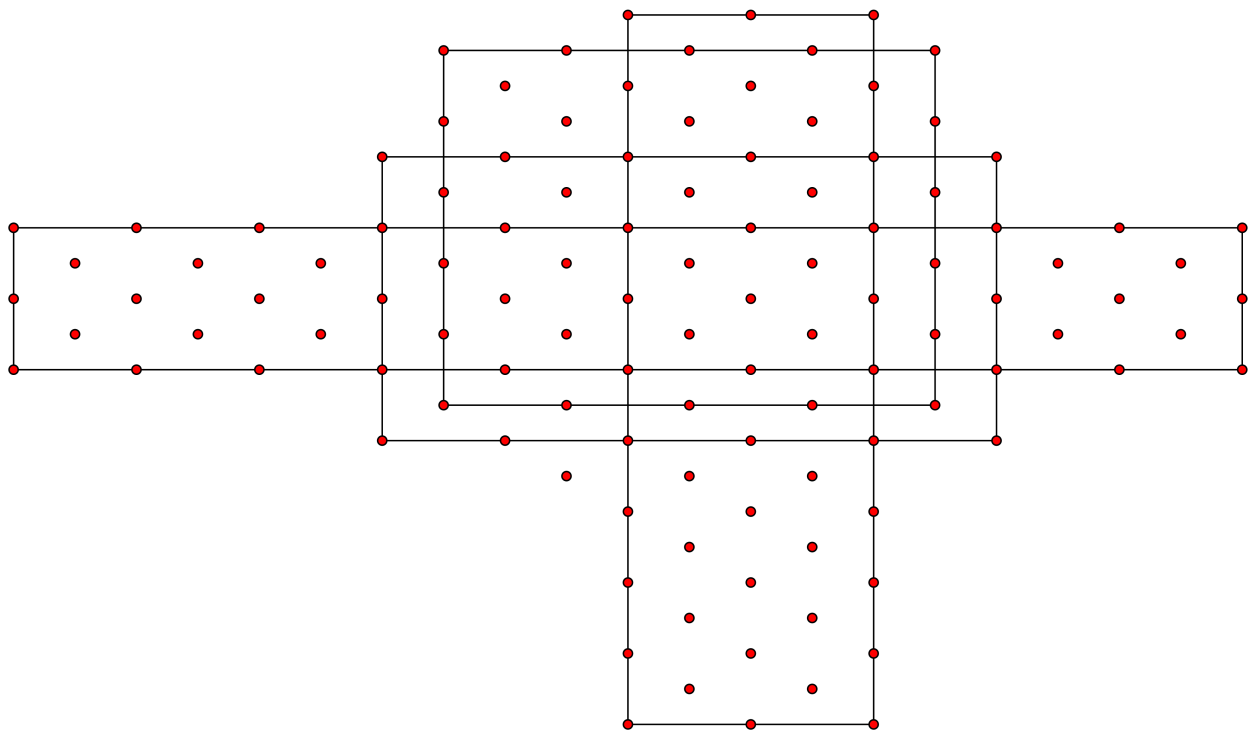
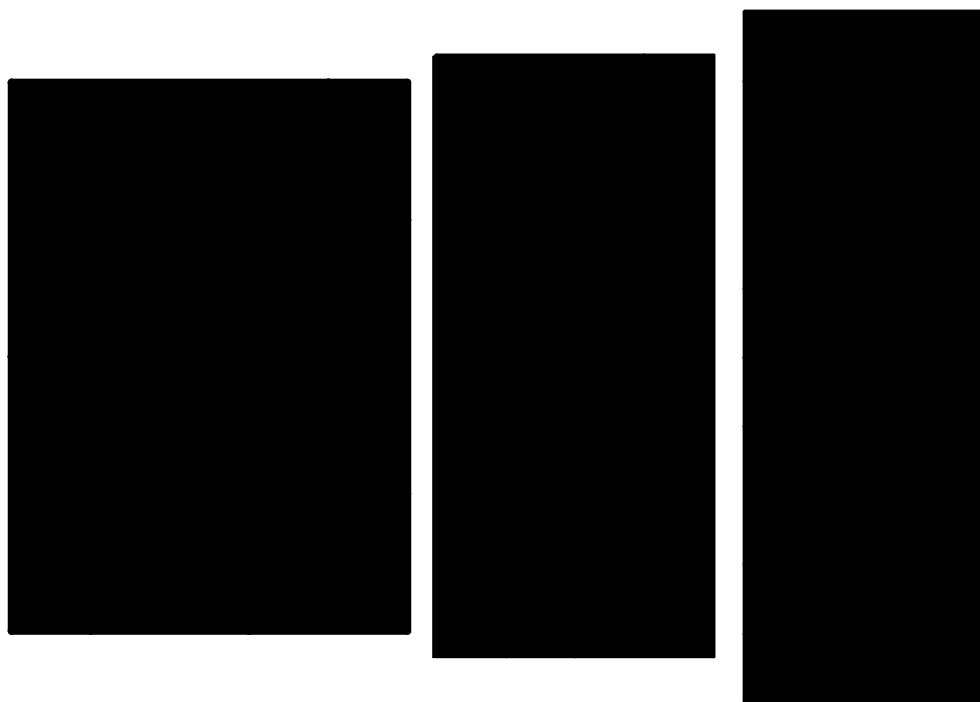


FIGURE 6.4 – Les quatre différents rectangles de 40 DTE d'aire dont les sommets sont sur la grille triangulaire.

En prenant le petit côté du DTE comme unité de longueur, l'hypoténuse mesure $\sqrt{3}$ et l'autre côté mesure 2. Les quatre rectangles tracés sur la grille mesurent donc $2 \times 10\sqrt{3}$, $4 \times 5\sqrt{3}$, $5 \times 4\sqrt{3}$ et $10 \times 2\sqrt{3}$. J'ai assez rapidement réussi à reconstituer deux autres rectangles mais celui qui mesure $2 \times 10\sqrt{3}$ résiste à mes différents essais. Cela me fait un nouveau problème à résoudre avec un programme informatique.

FIGURE 6.5 – *Les trois rectangles (solution en annexe).*

Formes convexes ou symétriques

En cherchant les rectangles, je déniche deux autres formes convexes : un trapèze et un pentagone, tous deux bien symétriques. La solution trouvée pour le pentagone se divise en deux parties superposables, excepté le triangle isocèle qui la chapotte. Du coup, cela conduit à une autre solution pour le rectangle – en inversant une des parties et en complétant l'ensemble avec le triangle isocèle retourné – qui est plus simple à mémoriser (en annexe).

Contrairement aux assemblages de pièces avec un jeu de la famille du Tangram basé sur le carré, ici les assemblages ne sont pas du tout évidents. Reconnaître dans quelle orientation doit se poser une pièce est plus difficile en raison de la diversité des angles et des longueurs. La proximité de certaines longueurs pourtant différentes n'arrange rien. Je vais faire, pour voir si cela peut m'aider à comprendre, un petit inventaire des longueurs des côtés des pièces de ce jeu.

FIGURE 6.6 – *Deux formes convexes réalisables avec Supergeotemplet (solution en annexe).*

Un coup d'œil à ce tableau de répartition des longueurs de côtés me confirme ce que l'expérience m'avait fait ressentir : Les longueurs rationnelles 2 et 3 ressemblent un peu aux longueurs irrationnelles $\sqrt{3} \approx 1,732$ et $2\sqrt{3} \approx 3.464$. Cela est d'autant plus vrai que les pièces en carton que j'utilise n'ont pas forcément les longueurs exactes et que les assemblages que je réalise avec elles sont loin d'être parfaits.

Aire	Pièce	Id.	1	2	3	4	$\sqrt{3}\approx 1,7$	$2\sqrt{3}\approx 3,5$	Total
1DTE	Triangle	a	1	1			1		3
2DTE	Cerf-volant	b	2				2		4
	Tr.équilatéral	c		3					3
	Rectangle	d	2				2		4
	Parallélogr.1	e	2	2					4
	Parallélogr.2	f		2			2		4
	Tr.isocèle	g		2				1	3
3DTE	Tr.rectangle	h			2		1	1	4
	Trapèze1	i	1	1			1	1	4
	Trapèze2	j	1	2			1		4
	Trapèze3	k	1	1		1	1		4
	Pentag. cvx	l	3	1			1		5
	Pentag.2	m	1	3			1		5
	Pentag.3	n	1	3			1		5
	Pentag.4	o	1	1			3		5
	Pentag.5	p	1	3			1		5
Total			17	25	2	1	18	3	66

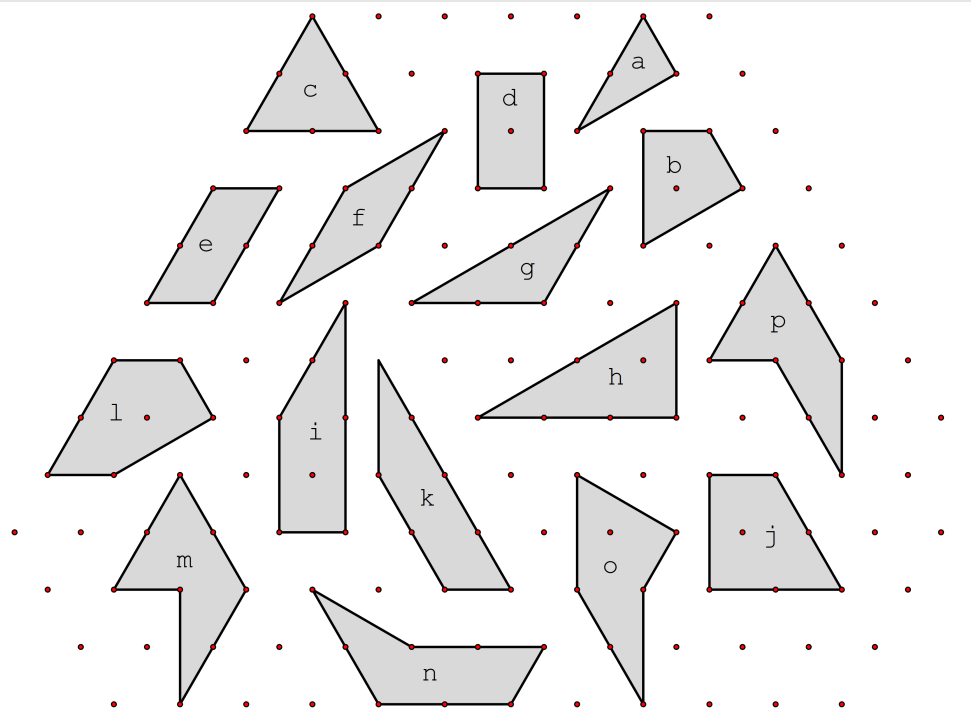


FIGURE 6.7 – Longueurs des côtés des pièces du Supergeotemplett.

Quoiqu'il en soit, certaines autres formes remarquables apparaissent progressivement lors de mes laborieux essais manuels. À côté des formes convexes, les plus remarquables à mon avis sont les formes symétriques car simples, belles et limitées en nombre. Certaines d'entre elles seulement me tombent dans la main. Le reste est à découvrir avec des moyens informatiques adaptés. Je découvre au passage deux formes ayant un bord extérieur convexe et un trou ; la première de celles-ci seulement possède un axe de symétrie.

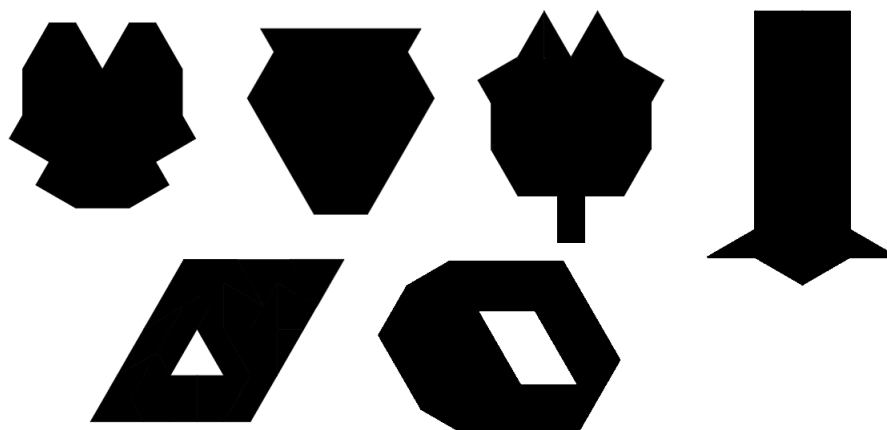


FIGURE 6.8 – Formes non-convexes et symétriques, sauf la dernière (solutions en annexe).

2. Numérisation du Supergeotemple

La programmation du jeu est un challenge que j'ai déjà abordé il y a longtemps, fort de la méthode employée avec succès pour le Tangram. Cependant mes différentes tentatives se sont jusqu'à présent soldées par un échec car, malgré les similitudes, ce jeu de pièces est très différent du Tangram. Je repars une nouvelle fois à l'assaut de ce sommet, fort d'un optimisme fondé sur l'adage qu'il n'y a pas de problème sans solution. Ce qui est bien avec l'informatique c'est qu'on finit toujours par trouver une solution satisfaisante.

a. Coordonnées des sommets

La grille sous-jacente étant triangulaire, les coordonnées des sommets doivent, pour rester entières, se lire sur un repère non-orthogonal qui se déduit de cette grille. Le système de coordonnées adopté doit également permettre le calcul des coordonnées après une rotation de 60° , 120° ou 180° dans un sens ou dans l'autre, et après un retournement. L'objectif ici est de simuler les mouvements d'une pièce manipulée par un joueur averti qui ne la place qu'à des endroits autorisés (les sommets sur les nœuds de la grille).

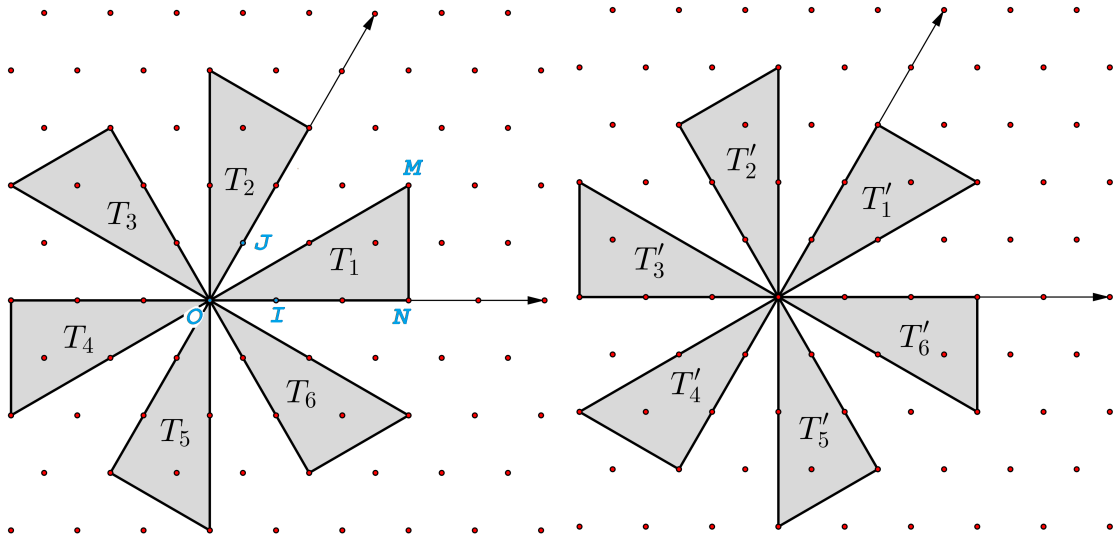


FIGURE 6.9 – Douze alias d'une forme après rotation d'un multiple de 60° ou symétrie par rapport à un côté.

Dans le cas général, lorsque la pièce n'a pas de symétrie, elle peut se présenter de douze façons différentes dans une forme. Ces douze alias ont été représentés ci-dessus pour le grand triangle (la pièce h). J'ai nommé ces alias $T_1, T_2, \dots, T_6, T'_1, T'_2, \dots, T'_6$ en prenant comme point de départ l'alias T_1 (dont les sommets sont, sur la figure, notés NOM) et en suivant le schéma suivant : en partant de T_i , j'ai obtenu T_{i+1} par rotation de 60° dans le sens trigonométrique (contrahoraire) autour du sommet O ($\hat{O} = 30^\circ$) et j'ai obtenu T'_i par symétrie d'axe l'hypoténuse $[MO]$. Les coordonnées des sommets de T_1 dans le repère (O, I, J) sont $N(3, 0)$, $O(0, 0)$ et $M(2, 2)$ (rappel : on lit une coordonnée sur un axe en projetant le point sur cet axe *parallèlement* à l'autre axe).

O(0,0)	N	M	O(0,0)	N	M
T_1	(3,0)	(2,2)	T'_1	(0,3)	(2,2)
T_2	(0,3)	(-2,4)	T'_2	(-3,3)	(-2,4)
T_3	(-3,3)	(-4,2)	T'_3	(-3,0)	(-4,2)
T_4	(-3,0)	(-2,-2)	T'_4	(0,-3)	(-2,-2)
T_5	(0,-3)	(2,-4)	T'_5	(3,-3)	(2,-4)
T_6	(3,-3)	(4,-2)	T'_6	(3,0)	(4,-2)

FIGURE 6.10 – Coordonnées des alias de NOM après rotation autour de O et/ou symétrie d'axe $[MO]$.

Déterminons les coordonnées dans le repère triangulaire (O, I, J) de $M'(x', y')$ image de $M(x, y)$ par la rotation r d'angle $\frac{\pi}{3}$ et de centre l'origine $O(0, 0)$:

$$\overrightarrow{OM'} = r(\overrightarrow{OM}) = r(x\overrightarrow{OI} + y\overrightarrow{OJ}) = xr(\overrightarrow{OI}) + yr(\overrightarrow{OJ}).$$

$$\text{Or } r(\overrightarrow{OI}) = \overrightarrow{OJ} \text{ et } r(\overrightarrow{OJ}) = \overrightarrow{OI} = -\overrightarrow{OI} + \overrightarrow{OJ}.$$

$$\text{J'en déduis } \overrightarrow{OM'} = x\overrightarrow{OJ} + yr(-\overrightarrow{OI} + \overrightarrow{OJ}) = -y\overrightarrow{OI} + (x + y)\overrightarrow{OJ}.$$

Finalement $M'(-y, x + y)$.

Je vérifie ma formule avec les points $M_1(2, 2)$ et $N_1(3, 0)$:

- ♦ $M_2(-2, 2 + 2)$ et $N_2(0, 3 + 0)$, soit $M_2(-2, 4)$ et $N_2(0, 3)$
- ♦ $M_3(-4, -2 + 4)$ et $N_3(-3, 0 + 3)$, soit $M_3(-4, 2)$ et $N_3(-3, 3)$
- ♦ $M_4(-2, -4 + 2)$ et $N_4(-3, -3 + 3)$, soit $M_4(-2, -2)$ et $N_4(-3, 0)$
- ♦ $M_5(2, -2 - 2)$ et $N_5(0, -3 + 0)$, soit $M_5(2, -4)$ et $N_5(0, -3)$
- ♦ $M_6(4, 2 - 4)$ et $N_6(3, 0 - 3)$, soit $M_6(4, -2)$ et $N_6(3, -3)$

Ces valeurs correspondent bien aux valeurs observées sur mon illustration.

Pour déterminer les coordonnées des six autres alias, obtenus par retournement des six alias précédent, je vais déterminer les coordonnées de $M'(x', y')$ symétrique de $M(x, y)$ par rapport à la droite (OI) :

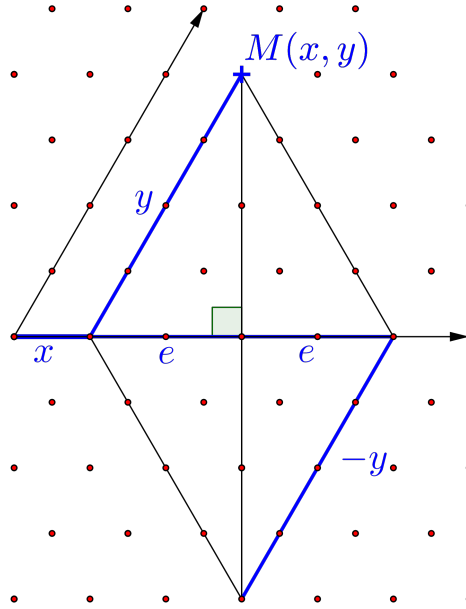


FIGURE 6.11 – Effet de la symétrie d'axe (OI) sur les coordonnées.

En me fondant sur ma figure, je remarque que $x' = x + 2e$ et $y' = -y$.

Comme $e = y \cos \frac{\pi}{3} = \frac{y}{2}$, j'en déduis que $x' = x + y$ (on aurait pu remarquer que $2e = y$).

Finalement $M'(x + y, -y)$. On remarque que cette symétrie correspond à la rotation d'angle $\frac{\pi}{3}$ suivie de la symétrie par rapport à la 1^{re} bissectrice qui, elle, a pour effet d'échanger les deux coordonnées.

Cette formule reste valable lorsque e est négatif comme par exemple pour le point $N_6(3, -3)$.

J'obtiens alors $(3 - 3, 3)$ soit $(0, 3)$ ce qui correspond bien aux coordonnées du symétrique de N_6 (le sommet de l'angle droit dans T_6) par rapport à (OI) .

En utilisant cette symétrie sur les sommets $M_1(2, 2)$ et $N_1(3, 0)$ de T_1 , j'obtiens les coordonnées des sommets de T'_6 . Il ne reste plus qu'à appliquer successivement cinq fois la rotation d'angle $\frac{\pi}{3}$ et de centre $O(0, 0)$ pour obtenir les coordonnées des cinq autres alias :

- ♦ par application de la symétrie sur M_1 et N_1 : $M'_6(4, -2)$ et $N'_6(3, 0)$
- ♦ par application de la rotation sur M'_6 et N'_6 : $M'_1(2, 2)$ et $N'_1(0, 3)$
- ♦ par application de la rotation sur M'_1 et N'_1 : $M'_2(-2, 4)$ et $N'_2(-3, 3)$
- ♦ idem pour les trois derniers points

Ces valeurs correspondent bien aux valeurs observées sur mon illustration.

b. Portions de losange

Le système des coordonnées pourra éventuellement être utile pour construire l'interface graphique du programme. Il n'apparaît pas très pratique pour étudier les assemblages des pièces entre elles. J'ai utilisé pour le Tangram une méthode qui codifie les pièces en les découpant selon une grille carrée. Pour adapter cette méthode au Supergeotemple, la grille doit avoir des mailles en forme de losange. Une forme de ce jeu peut être codée par une matrice dont les éléments sont des numéros correspondants à l'occupation du losange de base par la portion de pièce qui y est présente.

Les numéros ne sont pas choisis au hasard : le code d'un losange plein est 21 tandis qu'un losange vide est codé 0. Pour que deux pièces s'assemblent en un losange, la somme des deux codes doit être égale à 21. Deux pièces peuvent s'assembler sur un losange sans le remplir : par exemple les triangles 1 et 9 s'assemblent pour former le triangle $9 + 1 = 10$ ou bien les triangles 9 et 11 s'assemblent pour former le pentagone $9 + 11 = 20$. Certains assemblages doivent être déclarés interdits car provoquant des superpositions : 1 et 10 par exemple ne fait pas 11, 1 et 17 ne font pas 18, etc.

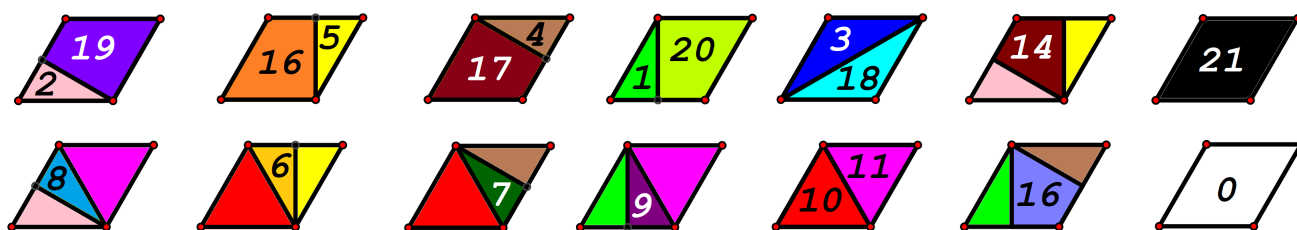


FIGURE 6.12 – La numérotation des portions du losange de base permet de reconnaître les assemblages possibles, le losange plein correspond à une somme de ses codes toujours égale à 21.

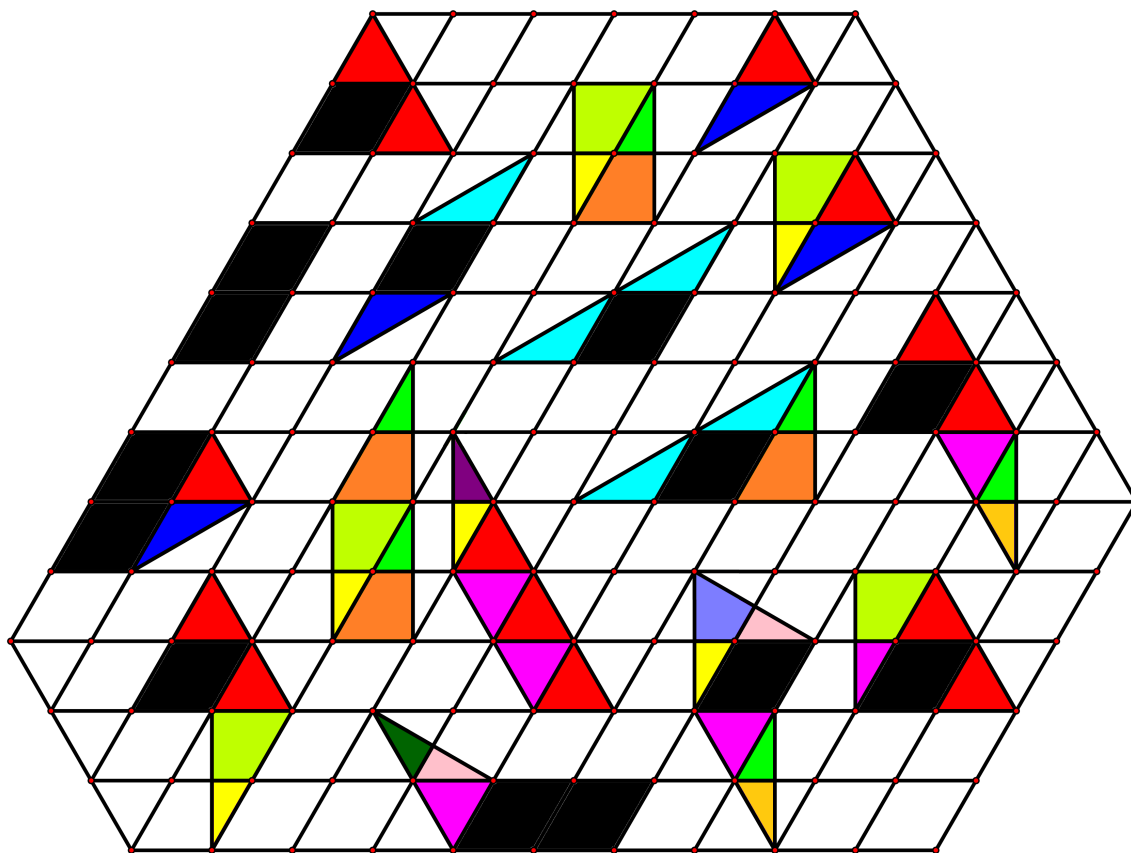


FIGURE 6.13 – Les différentes pièces du jeu découpées selon les losanges de la grille et montrant les codes (ici représentés par les couleurs, voir image précédente) utilisés dans le processus d'assemblage.

Ce système est complexe mais il devrait pouvoir fonctionner. Il utilise vingt types de portions de losange (à titre de comparaison, la codification du Tangram ne nécessitait que six types de portions

de carré) et une liste assez longue d'assemblages interdits (pas d'assemblages interdits avec le Tangram). Par ailleurs, ce système a encore besoin d'une routine pour convertir la matrice des codes en coordonnées et une routine pour la conversion inverse.

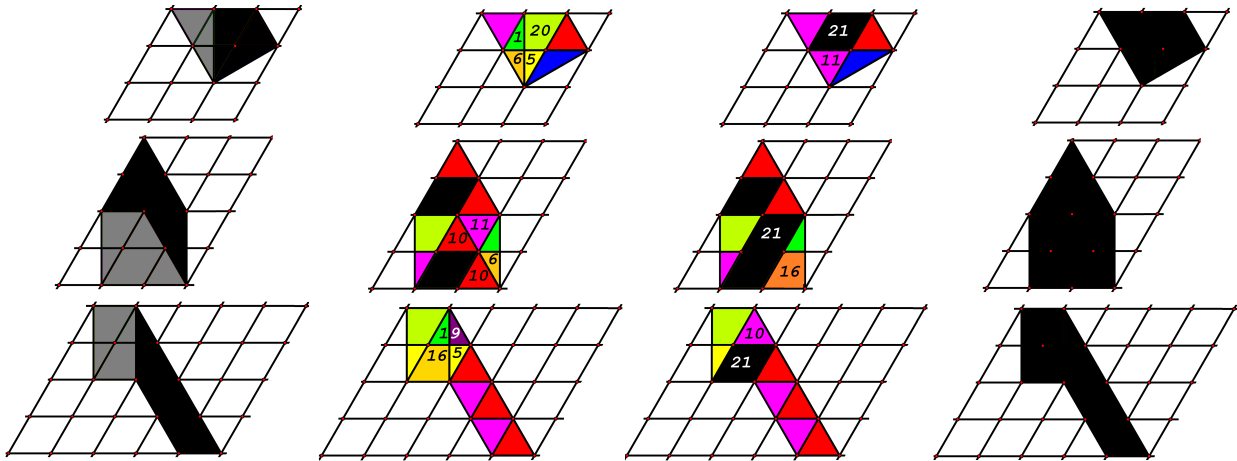


FIGURE 6.14 – Trois assemblages représentés avec les codes couleurs des portions de pièces ; les codes entrant dans la liaison entre pièces sont écrits en chiffres (au centre) ; assemblages résultants à droite.

La codification d'une pièce consiste alors en une liste de codes. Cela pourrait être une double liste pour donner le code selon la ligne et la colonne, mais pour des raisons de stockage dans un fichier je vais aussi traduire ces matrices sous la forme de mots : puisqu'il y a vingt codes à utiliser je vais utiliser les lettres majuscules correspondant au code, de A pour 1 à U pour 21. Le 0 qui code les losanges vides vont être codés par le caractère qui vient juste avant le A dans le système ASCII, soit le caractère @. Ainsi la conversion d'une lettre en code est obtenue simplement en Python, par l'instruction `ord(caractère)-64` et la conversion inverse se fait avec `chr(code+64)` : par exemple `ord('A')-64` donne 1 et `chr(1+64)` donne 'A'. Pour indiquer dans le mot comment il se découpe en lignes de n colonnes, je dois ajouter le nombre n comme suffixe, séparé du reste du mot par un point.

- Le triangle équilatéral en haut de la figure 6.13 est représenté par la matrice de 2 lignes et 2 colonnes `[[10,0],[21,10]]` mais aussi par le mot 'J@UJ.2'
- Le triangle rectangle en haut à droite de la même figure est représenté par la matrice de 2 lignes et 1 colonne `[[10],[3]]` et le mot 'JC.1'
- Le pentagone en bas à droite de la même figure est représenté par la matrice de 4 lignes et 3 colonnes `[[15,2,0],[5,21,0],[0,11,1],[0,0,6]]` et le mot 'OB@EU@KA@@F.3'

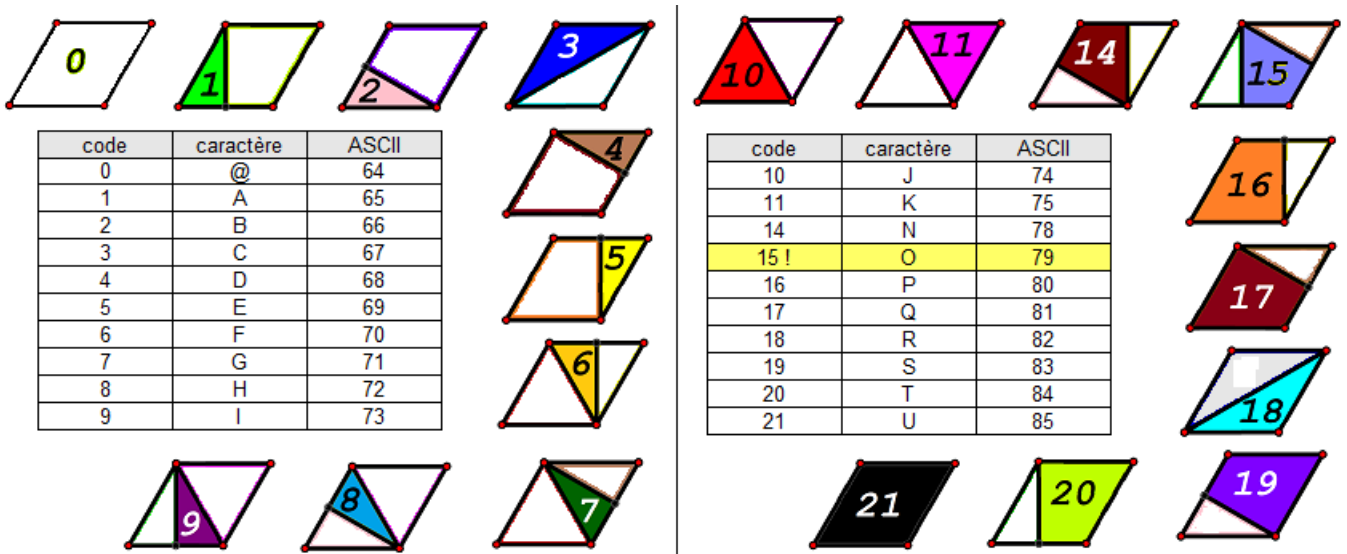


FIGURE 6.15 – Les vingt différentes portions du losange de base avec leurs codes (couleur, nombre et lettre). Le code 15 est le seul problématique qui n'assure pas parfaitement la règle d'assemblage des codes numériques.

Mon système a l'air de fonctionner correctement sauf qu'il y a une erreur dissimulée que je ne peux éviter. Je m'épargne la tâche rébarbative d'écrire ici le système d'équations que mes vingt lettres doivent vérifier pour rendre compte de tous les assemblages possibles. Ce système contient quinze égalités et n'a pas de solution où chaque lettre désigne un code unique et réciproquement. J'ai choisi la moins mauvaise pseudo-solution qui contient l'erreur suivante : la portion notée 15 assemblée à la portion 4 ne constitue pas le pentagone $15 + 4 = 19$ mais le pentagone 20. Cette erreur est minime et je vais pouvoir la gérer spécifiquement lors du processus d'assemblage.

Pour récapituler les règles du processus d'assemblage, certaines n'ayant pas encore été explicitement formulées : la superposition des losanges des deux pièces à assembler est valide si un des deux codes est 0 ou si la somme des codes est égale à 10, 11, 14, 15, 16, 17, 19 (on considère que $15 + 4 = 20$), 20 ou 21. Certaines superpositions ne constituent pas de liaison mais ne sont pas interdites (dans ce cas les pièces s'assemblent si elles constituent une liaison ailleurs) : 1 et 4, 1 et 5, 1 et 6, 1 et 7, 2 et 4, etc. Certaines superpositions sont pas interdites car elles constituent des recouvrements : 1 et 2, 1 et 3, 1 et 8, 1 et 10, etc.

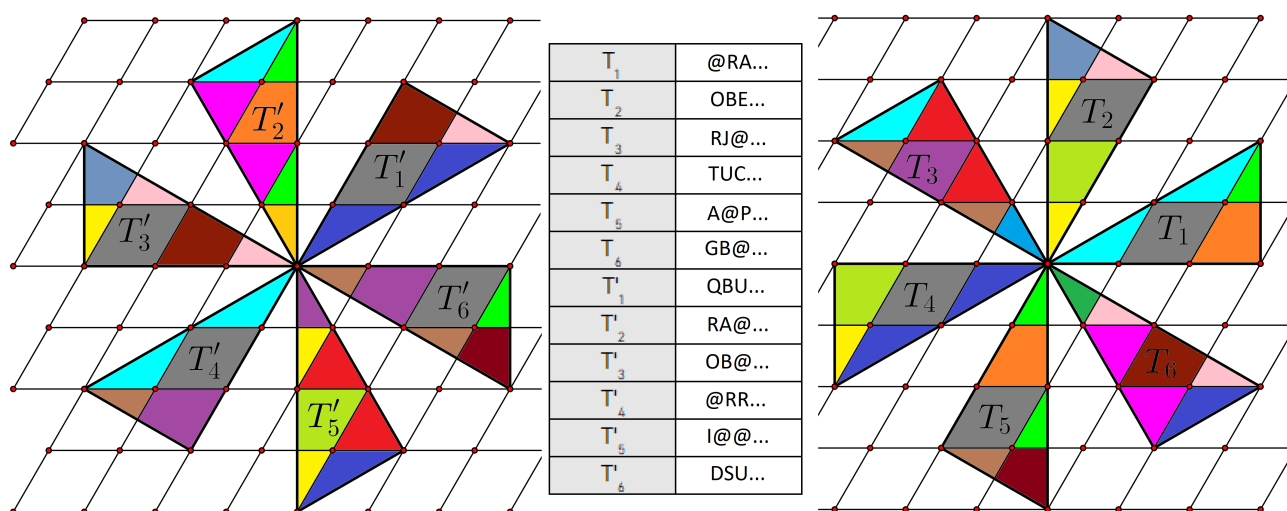


FIGURE 6.16 – Les seize alias du grand triangle découpés selon la grille de losanges afin de les nommer. Les trois premiers caractères du nom seulement sont écrits afin d'identifier l'écriture canonique de cette forme.

Pour la forme représentée ci-dessus sous ses différents aspects, j'ai effectué manuellement le début du nommage de chaque alias selon la règle énoncée. L'instruction Python `['@RA', 'OBE', 'RJ@', 'TUC', 'A@P', 'GB@', 'QBU', 'RA@', 'OB@', '@RR', 'I@@', 'DSU'].sort` ordonne les seize noms et renvoie la liste triée `['@RA', '@RR', 'A@P', 'DSU', 'GB@', 'I@@', 'OB@', 'OBE', 'QBU', 'RA@', 'RJ@', 'TUC']` ce qui permet de désigner '@RA...' comme l'écriture canonique de cette forme.

c. Encodage du bord

Le système d'encodage par portion me paraît un peu trop compliqué et ne se prête pas facilement aux rotations et symétrie qui sont pourtant indispensables. J'envisage alors un autre système que j'espère plus prometteur. Je l'ai employé à l'époque de mon travail sur le Tangram mais il faut l'adapter à la géométrie de ce jeu : il s'agit de décrire le bord extérieur d'une forme en donnant la direction d'un segment et sa longueur. L'unité de longueur sera différente selon la direction.

Je note les douze directions possibles par un chiffre hexadécimal de 0 à b (code le 11), le code 0 étant la direction Est et les autres codes sont obtenus en tournant de 30° dans le sens horaire : le Sud est codé 3, l'Ouest 6 et le Nord 9. La grille de losanges étant orientée Est-Ouest, les longueurs dirigées dans des directions paires sont des multiples du côté d'un losange. Les longueurs dirigées dans des directions impaires sont des multiples de $\sqrt{3}$ fois le côté d'un losange mais elles seront codées par des entiers, la multiplication par $\sqrt{3}$ ne s'effectuant qu'au dernier moment.

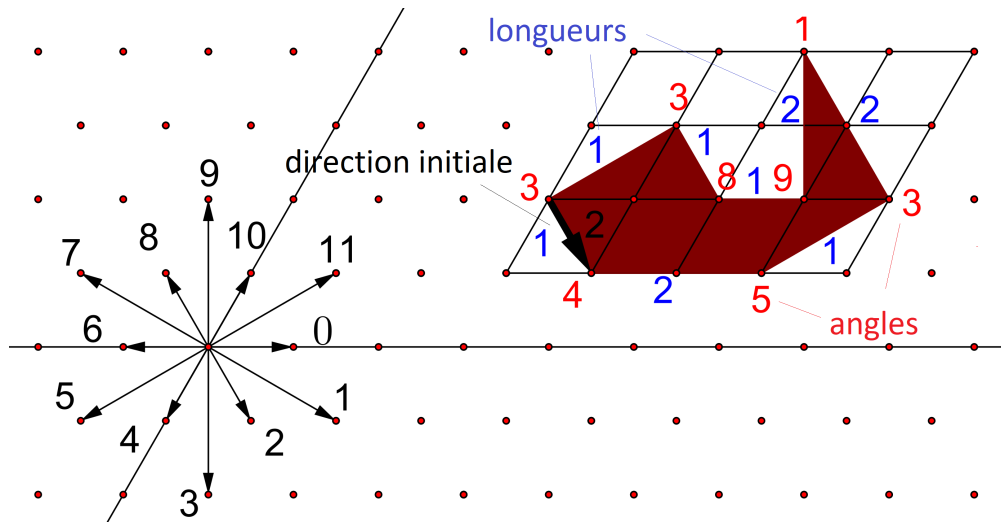


FIGURE 6.17 – Encodage du bord d'un octogone : le point de départ étant sur le bord gauche de la grille, la direction initiale est 2, les angles sont, dans l'ordre, 3-4-5-3-1-9-8-3 (marqués en rouge) et les longueurs sont, dans l'ordre, 1-2-1-2-2-1-1-1 (marqués en bleu). Cette forme est alors encodée 34531983(2)12122111.

À partir des coordonnées des pièces, je peux facilement déterminer cet encodage du bord. Il suffit de déterminer les distances entre un sommet et le suivant en unités entières de longueur (les longueurs réelles dépendent de la direction). Comme je sais déjà tourner et retourner une forme avec les coordonnées, je peux obtenir les différents encodages du bord de tous les alias et choisir parmi ceux-là celui qui est le premier dans l'ordre lexicographique. Je dispose ainsi rapidement du code canonique de chacune des seize pièces.

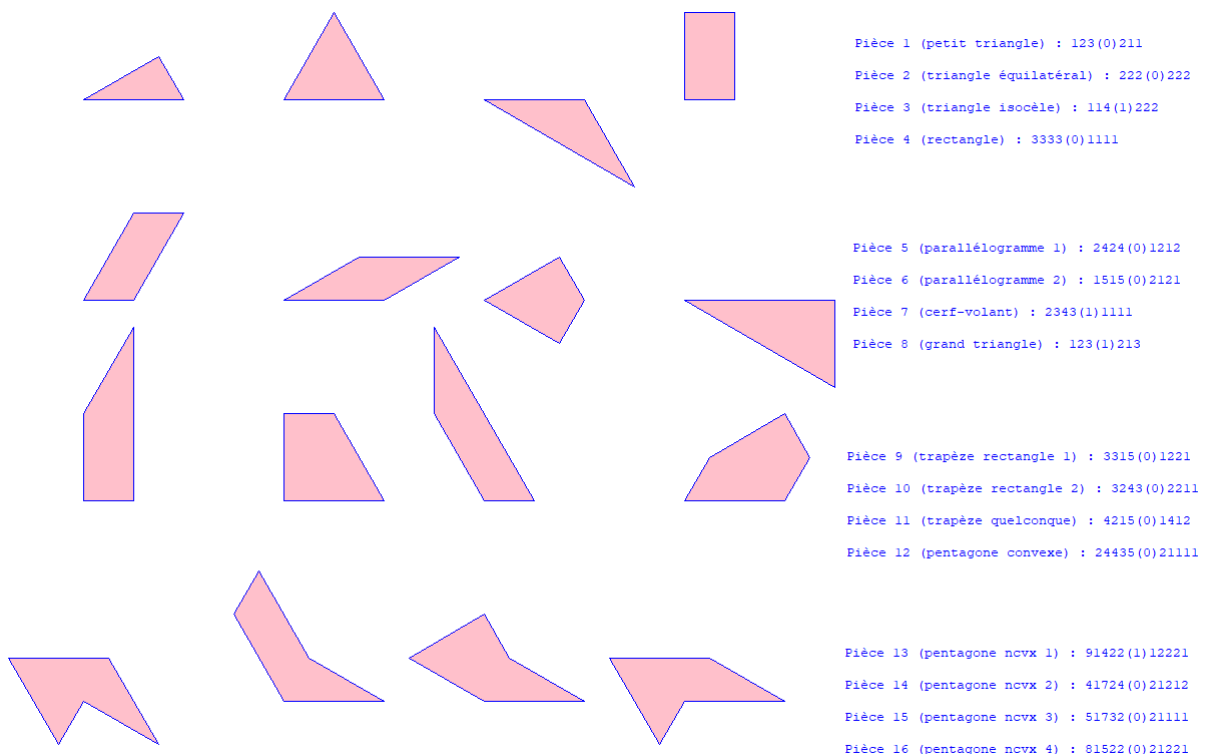


FIGURE 6.18 – Les seize pièces du jeu représentées dans la disposition décrite par les codes de droite.

Ce type d'encodage amène deux avancées : il est très facile d'obtenir la représentation visuelle d'une pièce et d'éliminer, parmi les alias, ceux qui sont des encodages différents d'une forme déjà encodée dans une autre position. En effet, pour faire les images de polygones, la bibliothèque `turtle` de Python se révèle ici particulièrement adaptée. Et pour reconnaître les doublons (par exemple 114(7)222 et 141(2)222 sont deux encodages différents d'une même forme), il suffit de mettre au point une routine d'identification de l'encodage canonique.

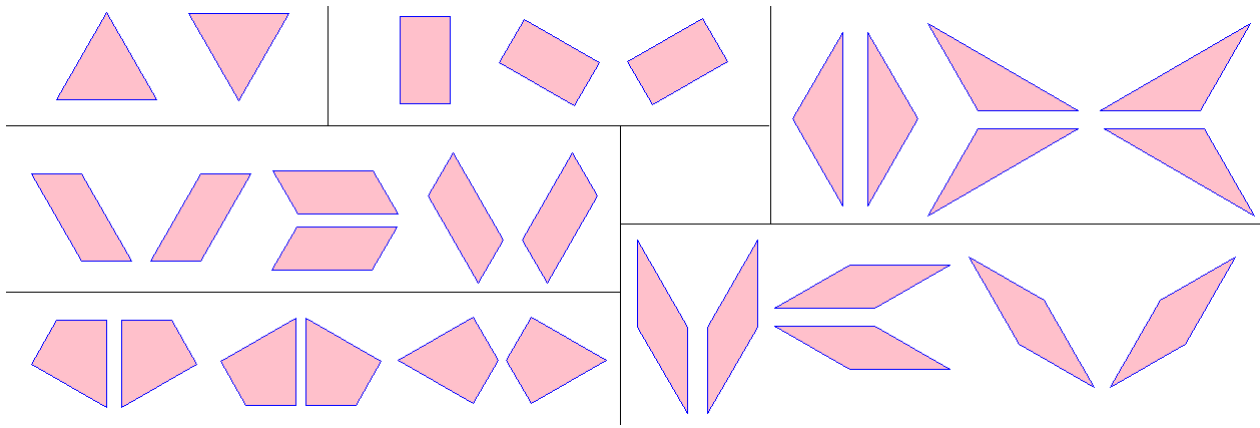


FIGURE 6.19 – Les six pièces symétriques ayant moins de seize alias.

Pour assembler deux pièces entre elles, et ultérieurement une pièce avec une forme déjà agrégée, ce système a besoin d'identifier des bords orientés dans des directions opposées. Pour cela, il suffit de soustraire le plus petit des codes de direction au plus grand, cette différence doit être égale à 6. Les longueurs des segments qui s'assemblent n'ont pas nécessairement besoin d'être identiques. La forme assemblée a un code qui se déduit des deux parties qui fusionnent : les angles s'ajoutent et les segments qui entrent dans la liaison sont éliminés.

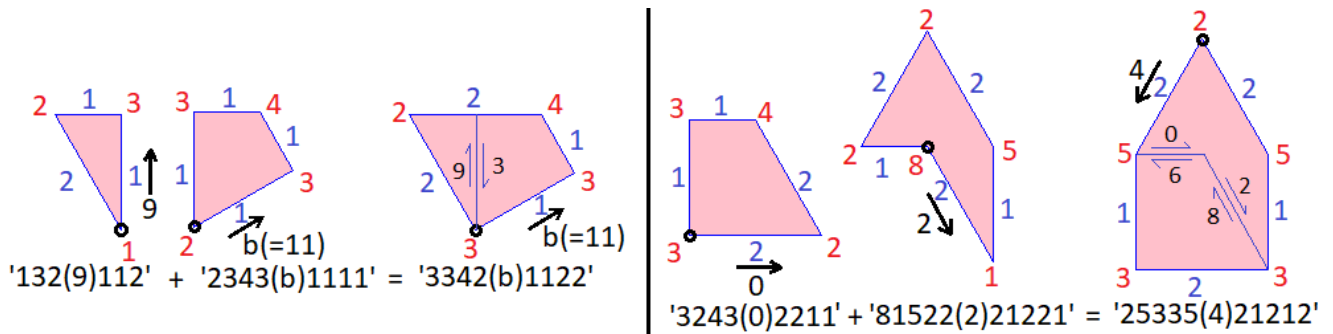


FIGURE 6.20 – Assemblages de deux pièces entre elles avec ce système d'encodage des bords.

d. Génération des formes convexes

Finalement, le dernier système d'encodage exposé semblant cumuler tous les avantages, je vais l'utiliser pour générer les formes réalisables avec le Supergeotemple. Une inquiétude s'est cependant insinuée en moi concernant la faisabilité de ce projet : le nombre de formes attendu est considérable car il y a seize pièces à assembler et pour dix d'entre elles on a douze alias, ce qui est énorme (au total, il n'y a pas moins de 149 alias à combiner de toutes les façons). Le temps de traitement et la quantité de mémoire sollicitée vont être prohibitifs. J'ai peur que le projet n'aboutisse pas.

Au temps de mon travail sur le Tangram, j'avais exploré une autre piste qui s'était révélée très efficace pour le but restreint d'identifier les formes convexes réalisables : je génèrais tous les polygones pouvant être tracés sur un quadrillage de carrés dont l'aire égalait celle des pièces du jeu (8 carrés unitaires). Ensuite, au lieu d'agréger les pièces entre elles, j'avais utilisé un algorithme de retour sur trace pour tenter de paver ces polygones avec les sept pièces du jeu. Cet algorithme n'est pas simple à programmer mais, une fois réalisé, comme le nombre de polygones convexes candidats est faible (20 pour le Tangram), le programme aboutissait très vite à la conclusion (13 polygones convexes pavables avec le Tangram). Ce *modus operandi* évite un processus d'agrégation fastidieux et excessif en termes de durée et de mémoire.

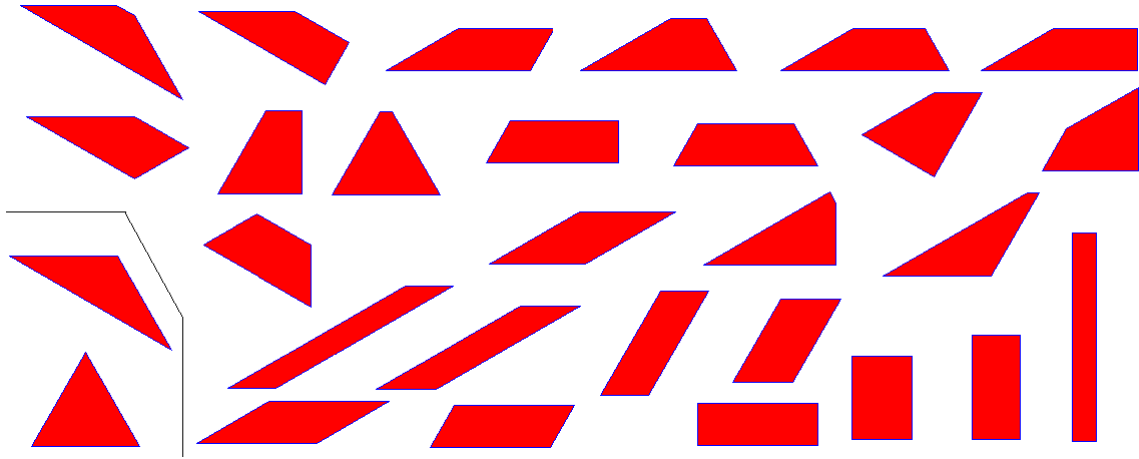


FIGURE 6.21 – Triangles et quadrilatères candidats pour être reconstitués par le Supergeotemplet.

Je commence par générer la liste des angles que peuvent avoir les polygones convexes candidats sachant que le nombre n de côtés nous donne la somme des angles : $180 \times (n - 2)^\circ$, soit avec mon unité d'angle $6 \times (n - 2)$. Comme je ne cherche que les polygones convexes, les angles qui interviennent appartiennent tous à la liste 1, 2, 3, 4, 5. Pour éviter les doublons, je choisis, parmi les permutations et inversions d'une suite d'angles, celle qui arrive en premier dans l'ordre lexicographique. En procédant ainsi pour les triangles, j'obtiens les trois valeurs déjà connues. Mais la diversité des formes explose ensuite jusqu'aux dodécagones, avec un maximum de possibilités pour les hexagones. Voici la liste complète de ces listes d'angles qui contient 165 éléments.

- ♦ 3 valeurs pour les triangles : '114', '123', '222'
- ♦ 16 valeurs pour les quadrilatères : '1155', '1245', '1254', '1335', '1344', '1353', '1425', '1434', '1515', '2235', '2244', '2325', '2334', '2343', '2424', '3333'
- ♦ 28 valeurs pour les pentagones : '12555', '13455', '13545', '13554', '14355', '14445', '14454', '14535', '15255', '15345', '22455', '22545', '23355', '23445', '23454', '23535', '23544', '23553', '24255', '24345', '24354', '24435', '24444', '24525', '25335', '33345', '33435', '33444'
- ♦ 46 valeurs pour les hexagones : '135555', '144555', '145455', '145545', '145554', '153555', '154455', '154545', '155355', '225555', '234555', '235455', '235545', '235554', '243555', '244455', '244545', '244554', '245355', '245445', '245454', '245535', '252555', '253455', '253545', '254355', '254445', '255255', '333555', '334455', '334545', '334554', '335355', '335445', '343455', '343545', '344355', '344445', '344454', '344535', '344544', '345345', '345354', '345435', '353535', '444444'
- ♦ 35 valeurs pour les heptagones : '1455555', '1545555', '1554555', '2355555', '2445555', '2454555', '2455455', '2455545', '2455554', '2535555', '2544555', '2545455', '2545545', '2553555', '2554455', '3345555', '3354555', '3355455', '3435555', '3444555', '3445455', '3445545', '3445554', '3453555', '3454455', '3454545', '3454554', '3455355', '3455445', '3455535', '3535455', '3544455', '3544545', '3545355', '4444455'
- ♦ 22 valeurs pour les octogones : '15555555', '24555555', '25455555', '25545555', '25554555', '33555555', '34455555', '34545555', '34554555', '34555455', '34555545', '34555554', '35355555', '35445555', '35454555', '35455455', '35455545', '35535555', '35544555', '35545455', '35553555', '44445555'
- ♦ 6 valeurs pour les nonagones : '255555555', '345555555', '354555555', '355455555', '355545555', '444555555'
- ♦ 7 valeurs pour les décagones : '3555555555', '4455555555', '4545555555', '4554555555', '4555455555', '4555545555'
- ♦ 1 valeur pour les hendécagones : '45555555555'
- ♦ 1 pour les dodécagones : '555555555555'

J'utilise ensuite cette liste pour générer les formes convexes candidates à une résolution par le Supergeotemplet. J'essaie pour cela toutes les combinaisons de longueurs jusqu'à une certaine valeur raisonnable que je fixe empiriquement à 15 pour les triangles et quadrilatères, 10 pour les pentagones (à cause de ces valeurs 10 et 15 j'utilise des caractères hexadécimaux pour écrire les angles), 6 pour les hexagones, 5 pour les heptagones, 4 pour octogones et nonagones, 3 pour décagones à dodécagones. C'est arbitraire mais cela évite des traitements trop longs. À ce stade de ma recherche, négliger l'exhaustivité ne me paraît pas si grave, l'idée étant d'aboutir à des résultats tangibles.

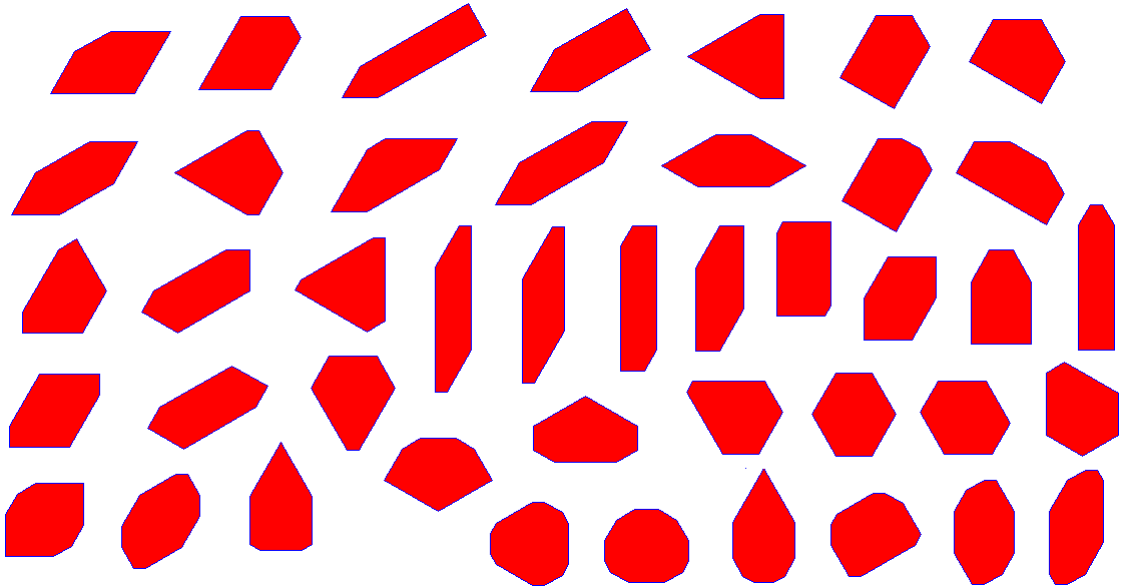


FIGURE 6.22 – Florilège symétrique des pentagones aux dodécagones candidats pour le Supergeotemplett.

Malgré cette limitation que je me suis imposée pour les longueurs, j'obtiens quelques centaines de formes candidates, 708 pour être exact. Je les visualise pour en extraire quelques images dont ce florilège de formes symétriques, ou apparemment symétriques car je n'ai pas rigoureusement effectué cette sélection. Au passage je m'aperçois que ma procédure d'expurgation des doublons ne fonctionne pas parfaitement car j'ai quelques doublons dans ce résultat finalement partiel et légèrement redondant.

3. Formes convexes reconstituables avec le jeu

a. Agrégation versus désagrégation

Mes règles d'assemblages conduisent à des formes que je qualifierai d'*homogènes* : les pièces s'assemblent en mettant en commun des segments entiers de même nature : les liaisons commencent et s'achèvent toujours sur un des nœuds de la grille de losanges. Les segments assemblés ont des longueurs entières (1, 2, 3 ou 4) ou irrationnelles ($\frac{\sqrt{3}}{2}$, $\sqrt{3}$ ou $2\sqrt{3}$).

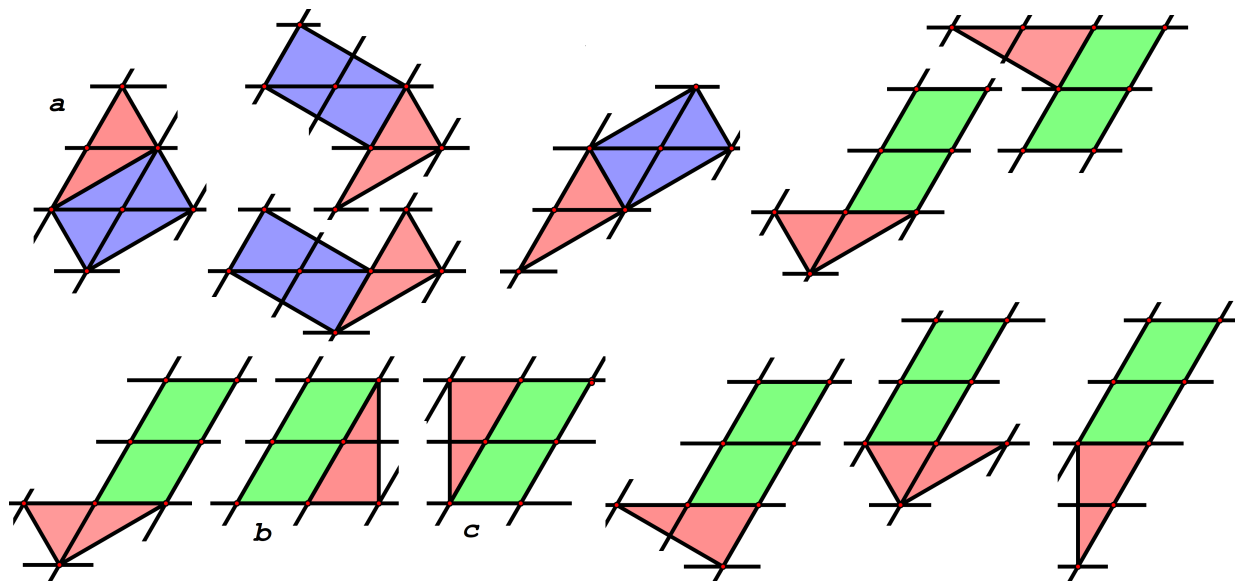


FIGURE 6.23 – En bleu, assemblages homogènes du rectangle sur un noyau constitué par le petit triangle. En vert, assemblages homogènes du petit triangle sur un noyau constitué par le parallélogramme.

Pour établir la liste de toutes les formes homogènes constructibles avec le Supergeotemplet, on doit savoir agréger une pièce à un noyau constitué par une ou plusieurs pièces préalablement agrégées entre elles : on fixe la position du noyau et on essaie de lui agréger la pièce de toutes les manières possibles. Successivement sur chacun des nœuds du bord extérieur du noyau, on essaie de placer la pièce dans chacune des positions de chacun de ses alias. On retient les formes valides et, avant de les enregistrer, on les compare à toutes les formes déjà obtenues, afin de ne pas enregistrer de doublon. La figure ci-dessus montre les différents assemblages du rectangle avec le petit triangle ainsi qu'une partie des assemblages du parallélogramme avec le petit triangle. Sur cette figure, on remarque que les formes **b** et **c** sont des doublons de la forme **a** ce qui conduit à un seul enregistrement.

Comme je l'ai expliqué plus haut, j'ai renoncé à la génération de toutes les formes homogènes pour me consacrer uniquement aux formes convexes. Disposant d'une liste conséquente sinon exhaustive des formes convexes candidates, j'envisage de passer chacune d'entre elles à la moulinette de la désagrégation : j'enlève à la forme candidate une 1^{re} pièce, positionnée dans l'alias et la position qui conviennent. J'obtiens une forme résiduelle à laquelle j'applique ce même traitement pour une 2^e pièce, etc. jusqu'à obtenir une impossibilité de placement de la n^e pièce. Je rétrograde alors en reprenant une position antérieure moins désagrégée pour laquelle je fais évoluer la place ou l'alias de la dernière pièce. Si aucune possibilité nouvelle n'est possible, je rétrograde encore davantage. L'issue de ce processus peut être une impossibilité (on a essayé toutes les possibilités) ou un placement valide de toutes les pièces.

Ce programme part de la liste des formes candidates établies précédemment dans le système d'encodage du bord extérieur. Mais ce système d'encodage ne peut être conservé que si la soustraction d'une pièce ne disloque pas la forme candidate en deux formes distinctes. Cela poserait de sérieux problèmes pour le traitement et devrait toujours pouvoir être évité car toutes les pièces étant successivement employées, il y en a forcément une qui peut être soustraite sans disloquer la forme résiduelle.

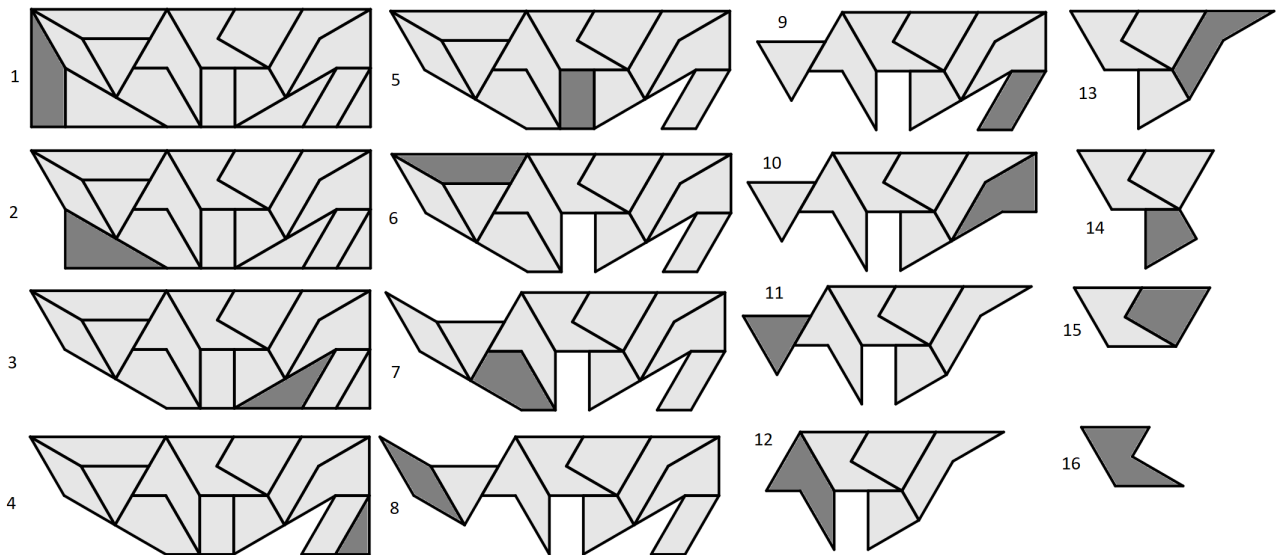


FIGURE 6.24 – Désagrégation du rectangle $10 \times 2\sqrt{3}$ selon un ordre des pièces enlevées qui respecte l'intégrité de la forme. La pièce enlevée à chaque étape est indiquée en gris foncé (un autre ordre serait plus probable).

D'un point de vue pratique, pour identifier une pièce pouvant être éliminée d'une forme résiduelle (ou d'une forme entière), il faut procéder à l'envers de ce qui est décrit pour l'agrégation, en particulier dans la figure 6.20 : Si un des alias de la pièce possède un de ses côtés allant dans la direction d'un des segments de la forme résiduelle, on peut tenter d'enlever cette pièce à la forme. Cette tentative ne pouvant aboutir que si tous les sommets de la pièce sont à l'intérieur, ou éventuellement sur le bord de la forme et si, comme on l'a dit, la forme résultante est d'un seul tenant (s'accoler par un sommet seulement peut être considéré comme inacceptable puisque, dans tous les cas, un autre ordre des pièces pourrait convenir).

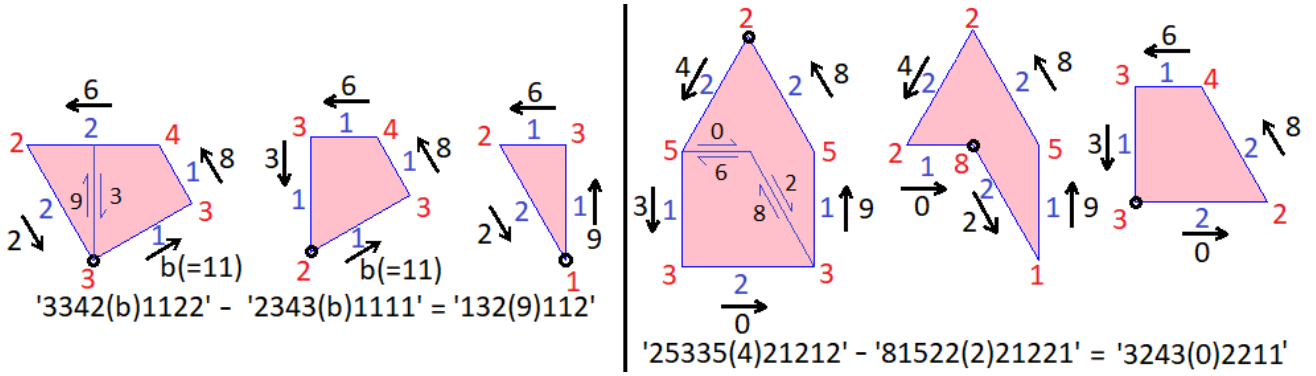


FIGURE 6.25 – Désagrégation d'une forme résiduelle lui enlevant une pièce orientée de façon acceptable.

L'orientation de chaque segment du bord n'est pas explicite dans l'encodage d'une forme, excepté pour le premier côté dont l'orientation est écrite entre parenthèses. Pour les autres segments, il suffit de calculer l'orientation du côté i à partir de l'orientation du côté $i-1$ et de l'angle i donné dans l'encodage avec la formule $\text{direction}[i+1] = \text{direction}[i] + 6 + \text{int}(\text{angles}[i+1], 16)$ [12]. Dans cette formule, $\text{int}(\text{angles}[i+1], 16)$ convertit en entier l'encodage hexadécimal de l'angle $i+1$ (NB : l'angle i est celui qui précède le côté i).

Le cerf-volant ci-dessus à gauche ayant pour code $2343(b)1111$, l'orientation du côté 1 est 'b', soit 11 ; celle du côté 2 est 8 car $11+6+3$ [12]=20. La suivante est 6 car $8+6+4=18$ [12] et la dernière 3 car $6+6+3=15$ [12]. Au passage, on peut vérifier que l'orientation qui se déduit ensuite est bien l'orientation initiale ($3+6+2=11$) ce qui démontre que le 1^{er} caractère de mon encodage est superflu.

Quand on constate qu'une pièce peut être enlevée à une forme sans la disloquer, il suffit de reconstruire l'encodage de la forme résiduelle en prenant, d'une part ce qui reste du bord de la forme auquel on a enlevé la partie commune avec la pièce, et d'autre part la partie du bord de la pièce qui n'est pas commune avec la forme, en inversant le sens de parcours. On retient alors ce code de la forme résiduelle ainsi que le rang de la pièce, son numéro d'alias et ses coordonnées. Ce dernier point doit permettre de localiser la pièce pour un éventuel déplacement si cette position ne permet pas d'aboutir sur une solution. Dans le cas où une solution est trouvée, les coordonnées de chacune des pièces permettent de représenter graphiquement la solution.

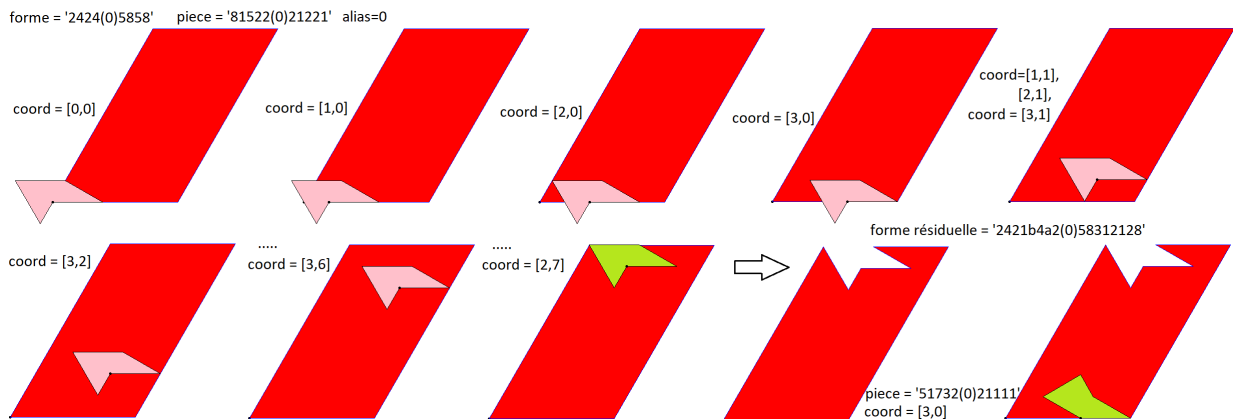


FIGURE 6.26 – Pérégrination de la pièce n°16, alias n°1, jusqu'à une position où elle peut être soustraite à la forme choisie. La forme résiduelle est ensuite balayée par la pièce n°15, alias n°1, où elle trouve sa place.

b. L'embourbement

J'ai énoncé plus haut le principe : « si un des alias de la pièce possède un de ses côtés allant dans la direction d'un des segments de la forme résiduelle, on peut tenter d'enlever cette pièce à la forme ». Il reste à préciser dans quel ordre doit se dérouler l'essai des différents alias et les différents positionnements de cet alias au-dessus de la forme.

L'ordre de l'alias n'est pas important puisqu'on veut les essayer tous. Par contre, l'ordre des pièces paraît intuitivement plus important : il vaut sans doute mieux placer en premier les formes non-convexes, les plus difficiles à placer. Je privilégierai l'ordre inverse de celui de la figure 6.18. Mais si, dans cet ordre, la désintégration complète d'une forme n'aboutit pas, est-on garanti que celle-ci n'est pas reconstituable avec les pièces du jeu ? Ce n'est pas certain, il me faudra alors procéder à un autre essai en choisissant un autre ordre des pièces. Je pourrai recommencer avec un ordre aléatoire, mais un échec ne me garantirait pas davantage qu'une solution ne peut être trouvée avec un autre ordre. Une exploration systématique de tous les ordres de pièces possibles me paraît inenvisageable car il y en a 16 soit 20 922 789 888 000. Je ne vais pas m'aventurer davantage sur ce terrain sans savoir si un ordre quelconque peut ou pas venir à bout d'une forme reconstituable.

L'exploration des déplacements d'une pièce au-dessus de la forme résiduelle me paraît adaptée au système des coordonnées tel qu'il a été exposé plus haut (partie 2.a). En prenant pour origine le premier sommet de la forme initiale, on devrait toujours pouvoir localiser la pièce, en lui attribuant les coordonnées du premier sommet de l'alias utilisé. La pièce va alors balayer la surface de la forme résiduelle jusqu'à en trouver une qui soit acceptable.

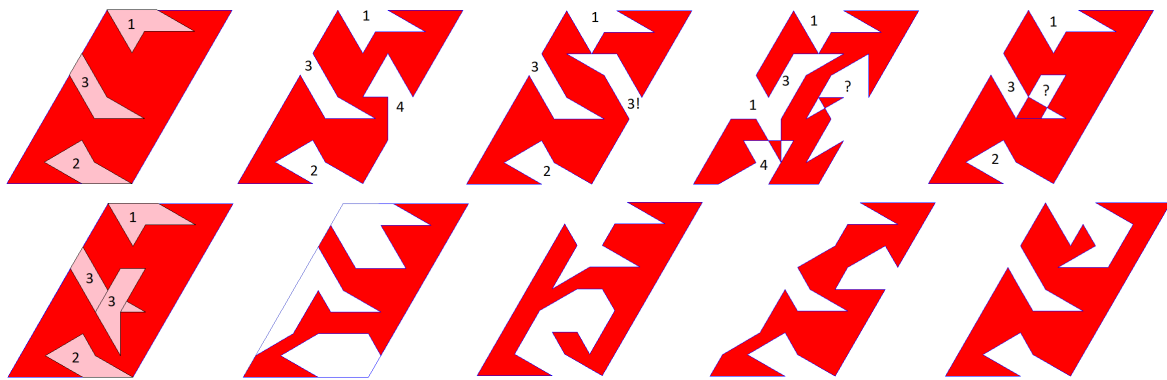


FIGURE 6.27 – Les premiers résultats obtenus exhibent les premiers défauts de mon algorithme de désintégration d'une forme convexe : pièces qui se chevauchent ou en doublons, positions valides non retenues, etc.

La première difficulté est d'identifier une position valide : le bord de la pièce examinée doit être en contact avec le bord de la forme résiduelle à désagréger et, surtout, l'intérieur de la pièce doit être intégralement à l'intérieur de la forme. C'est facile à énoncer mais, dans la pratique, cela nécessite de nombreux calculs et de nombreux tests : en supposant que l'on a montré que tous les sommets de la pièce sont, soit à l'intérieur, soit sur le bord de la forme, est-on certain que la pièce peut être soustraite de la forme ? Mes premiers essais me rappellent que non. Le bord de la pièce peut traverser le bord de la forme ce qui le conduit à avoir une de ses parties à l'extérieur. On doit identifier et rejeter ces positions non-valides.

La deuxième difficulté est de reconstituer l'ordre correct des sommets de la forme résiduelle. Quand je disais : « il suffit de reconstruire l'encodage de la forme résiduelle en prenant, d'une part ce qui reste du bord de la forme auquel on a enlevé la partie commune avec la pièce, et d'autre part la partie du bord de la pièce qui n'est pas commune avec la forme, en inversant le sens de parcours », je pensais travailler avec les encodages de forme. Mais j'ai commencé à travailler avec les coordonnées (pour déplacer une pièce au-dessus de la forme, pour identifier si un sommet de la pièce est sur un sommet de la forme résiduelle ou strictement sur un bord ou à l'intérieur) et c'est sans doute préférable de continuer avec les coordonnées.

Je constitue donc, pour chaque position de la pièce, la liste S des sommets qui sont sur le bord, en distinguant les placements sur un sommet ou sur un bord strict (à l'exclusion des extrémités). Dans cette liste, je dois inclure le rang du côté de la forme sur lequel se situe le sommet de la pièce. Car il me faut ensuite vérifier que la liste des sommets de S convient : la position de la pièce est invalidée si S contient moins de deux éléments, ou si deux sommets successifs de S sont sur des bords trop éloignés. Il y a d'autres cas à exclure mais déjà je vais avoir du mal à définir ce que signifie *trop* éloigné.

Cette phase du travail est, de loin, la plus difficile. Je regrette un peu mon système de notation complexe des portions de losange (partie 2.b) qui aurait eu le bon goût de faciliter grandement cette

phase de désintégration. Je l'avais abandonné parce que la traduction automatique d'une forme donnée par son encodage de forme dans cet encodage me paraissait trop compliqué. Mais après avoir exploré cette autre voie, je me retrouve embourbé dans un vaste marécage, enveloppé dans le brouillard des indices des rangs, des bords, des pièces... Il me manque la lumière et l'altitude pour prendre la bonne direction et une boussole pour arrêter de tourner en rond.

Je gère cette phase de léger découragement en relâchant mon effort. Je laisse ainsi venir à moi les idées éventuelles. Je reprends mon jeu en carton et cherche des assemblages convexes manuellement, sans en trouver de nouveau. J'essaie d'autres idées de découpage des formes en portions plus simples que les portions d'un losange de base, mais cette idée s'avère inutilisable car elle conduit à des agrégats qui ne s'écrivent ni de façon unique ni sous forme matricielle. Je me penche sur une autre forme d'encodage qu'avais mis décrit Ronald Read pour le Tangram : une description de la forme avec juste des codes de directions (8 pour le Tangram, 12 pour les puzzles de la famille du Geotemplet) répétés autant de fois qu'il y a de *segments unitaires*. En reprenant mes codes donnés figure 6.17, j'écrirais la forme 34531983(2)12122111 de cette figure 200b883685, ou plutôt 00b8836852 si je veux privilégier une permutation canonique. L'idée est sans doute excellente mais m'apporterait-elle quelque chose pour la désintégration d'une forme convexe ? Il ne me semble pas possible de traiter cette question sans utiliser les coordonnées donc, finalement, cette notation ingénieuse ne pourrait pas m'aider.

Ces tergiversations ne m'ayant pas amené sur un meilleur terrain, je reprends mon code initial avec sa liste **S** et je cherche à l'améliorer, plus ou moins convaincu que je n'ai que cette façon de progresser.

c. Le couronnement

Les modifications de mon algorithme s'étalant sur près d'un mois, je ne vais pas donner tous les détails de ce long chemin où alternent l'optimisme, la lassitude et le découragement. J'ai pourtant l'impression d'avancer, corrigeant petit-à-petit les erreurs rencontrées dans l'exécution. Pour cela j'affiche énormément d'informations pour identifier ce qui ne va pas. Progressivement, mon algorithme arrive à traiter correctement des milliers de placements avant de tomber sur une erreur, puis ce sont des dizaines de milliers de placements, puis des centaines de milliers...

Pour simplifier le travail qui semble très long à aboutir, je lance mon algorithme sur le puzzle Geotemplet qui ne contient que dix pièces. J'ai reconstitué rapidement la listes des formes convexes candidates qui contient pas moins de 365 formes ayant entre 4 et 10 côtés. Je poursuis mes multiples essais sur la forme princeps du rectangle de la figure 6.1 dont le code est 2424(0)4545. Je m'aperçois que pour ces dix pièces, le temps de traiter tous les placements est prohibitif : en un jour et demi, les deux plus grosses pièces n'ont toujours pas été changées de position...

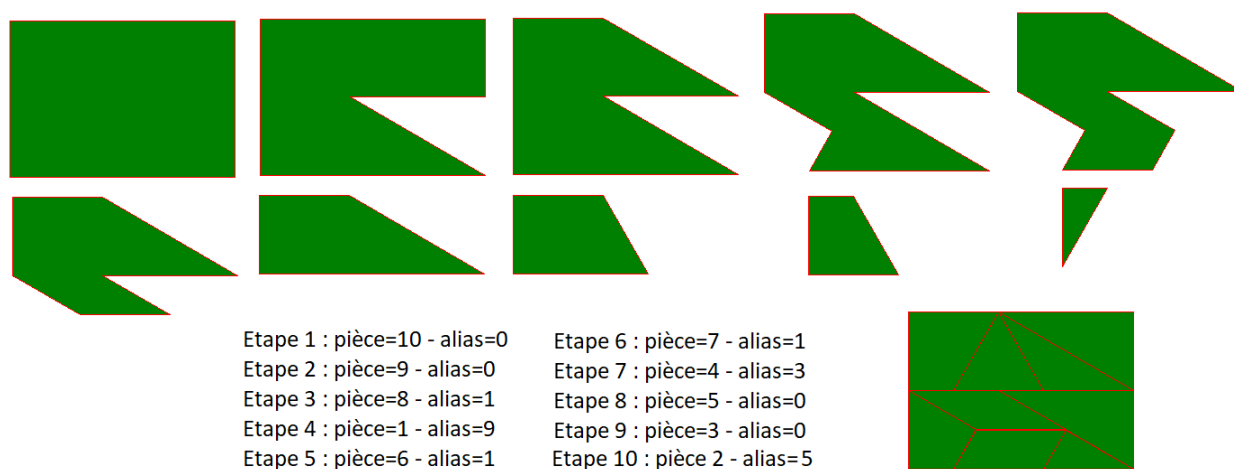


FIGURE 6.28 – La première solution trouvée par mon algorithme concerne le Geotemplet. Pour cette solution, les deux grands triangles sont encore utilisés dans leur premier alias.

Lors de cet ultime essai mon programme affiche des millions de lignes d'informations et, comme je lui demande de dessiner les placements où neuf pièces au moins sont correctement assemblées je

m'aperçois qu'il y en a de nombreux qui sont dessinés avec neuf pièces et, parmi ceux-là, certains présentent un trou d'un seul DTE! Ce trou ne pouvant être comblé que par une des deux petits triangles du jeu, l'algorithme devrait détecter une solution. J'arrête le traitement et vérifie sur les informations affichées que les neuf pièces sont correctement assemblées et qu'il y a bien un trou qui peut être comblé par un petit triangle et que cette pièce est effectivement disponible. Comme c'est bien le cas, je comprends que ma condition d'arrêt du programme n'est pas bien au point. Au lieu de détecter une solution, il identifiait une erreur!

J'enlève alors l'affichage de toutes les informations ainsi que les dessins de placements de neuf pièces trouvés en chemin pour ne conserver que le dessin de la première solution à dix pièces. Mon programme est enfin prêt à fournir des solutions pour les différentes formes convexes candidates. Sans les affichages et dessins intermédiaires, les solutions éventuelles apparaissent aussi plus rapidement mais ce n'est pas pour cela que je vais réussir engranger les solutions convoitées de ces formes convexes. Le temps d'exécution moyen pour déterminer une solution est très variable mais généralement excessivement long. Le programme peut tourner une journée sur une forme en ne réussissant qu'à y caser neuf pièces, la dixième n'ayant jamais la forme du trou à combler.

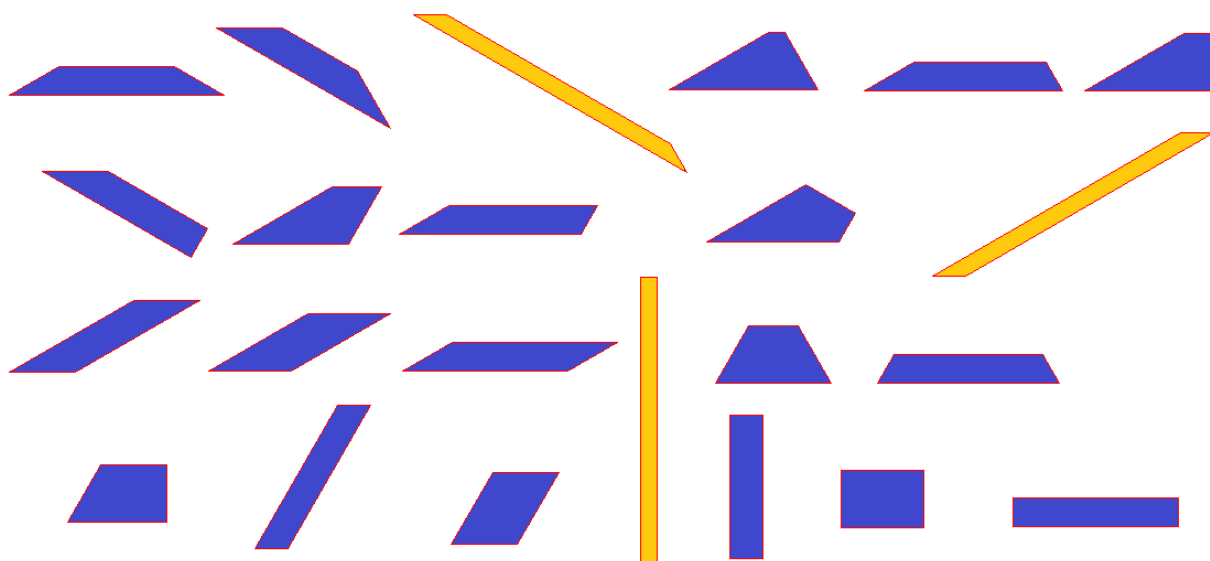


FIGURE 6.29 – En bleu, les vingt quadrilatères convexes réalisables avec le Geotemplet (solutions en annexe). En orange, les trois qui sont impossibles, une des pièces au moins ne pouvant y être insérée.

Face à cette nouvelle déception, et pour occuper le temps que l'ordinateur utilise pour passer en revue tous les placements, je me lance dans la recherche manuelle, à l'aide de mes pièces en carton. J'obtiens rapidement un premier succès, en moins de cinq minutes alors que l'ordinateur lambine sur cette forme depuis des heures. Je me prends alors au jeu de m'essayer sur les quadrilatères candidats. Au bout d'une journée, j'ai la quasi certitude que l'on peut en reconstituer dix-huit. Parmi les formes candidates restantes, trois sont impossibles car une pièce au moins ne peut être placée. Il reste deux formes étroites dans lesquelles entrent toutes les pièces mais je peine à les résoudre et les pense impossibles. Quelques jours plus tard cependant j'en trouve une solution, toujours manuellement.

L'ensemble des formes candidates peut être ainsi testé manuellement mais obtenir la liste définitive des formes convexes reconstituables avec le Geotemplet semble tout de même hors de portée. Aurais-je la patience de tester toutes les formes candidates? Si je le faisais, il me serait encore impossible de garantir qu'une de ces formes non résolue manuellement n'est pas reconstituable. Avec mon programme, le temps de traitement étant exagérément long – plusieurs jours pour épuiser toutes les possibilités – il ne serait pas raisonnable d'effectuer ces vérifications. En outre, j'ai conscience que ma liste des formes candidates n'est pas complète : j'ai limité la longueur des côtés et, d'autre part, il doit subsister des erreurs dans mon programme car j'ai découvert manuellement plusieurs formes convexes reconstituables qu'il n'avait pas trouvées. Pour ces deux raisons au moins, je me bornerai à donner un minorant du nombre de formes convexes reconstituables avec le Geotemplet.

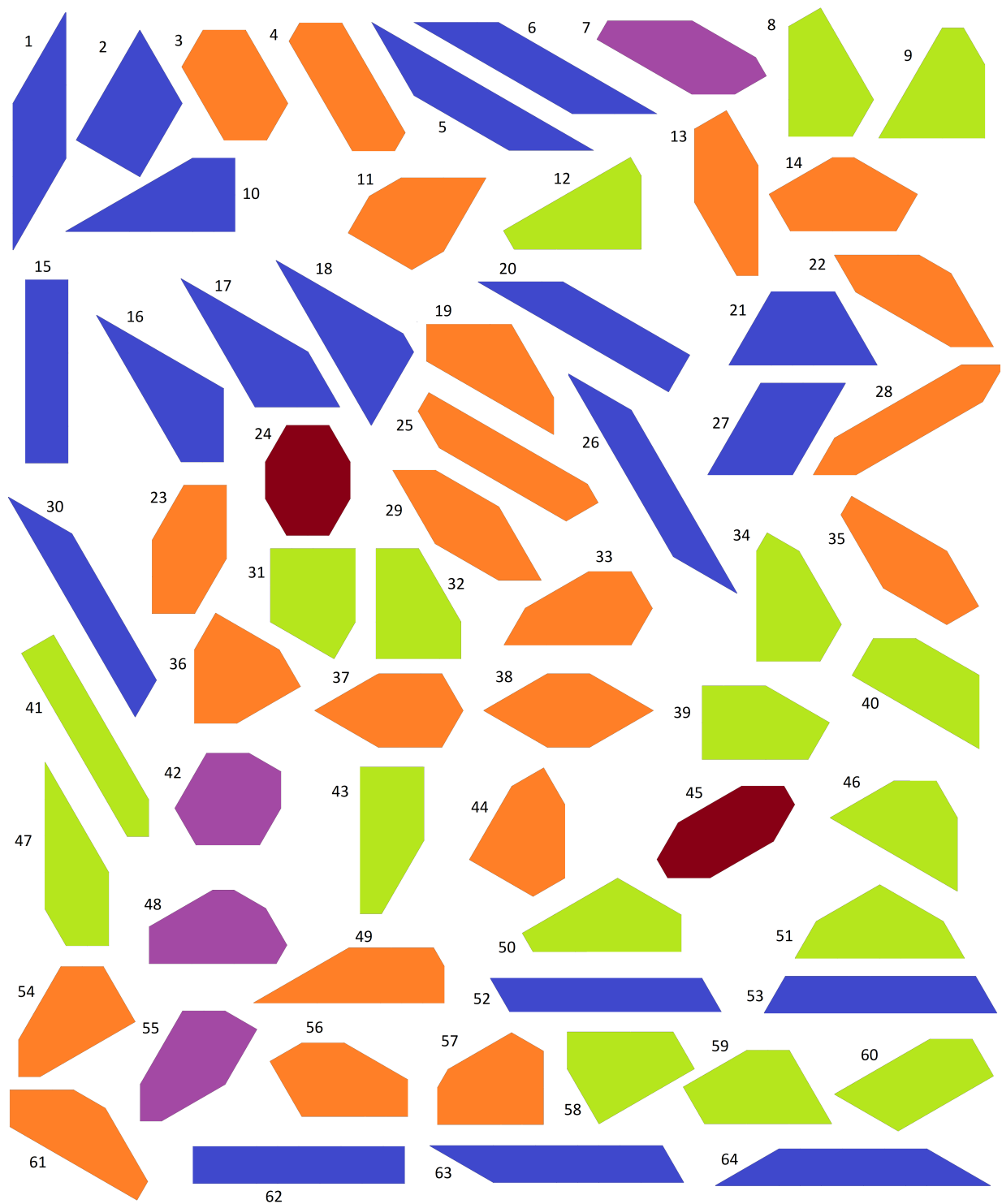


FIGURE 6.30 – Soixante-quatre formes convexes reconstituables avec les pièces du Geotemplet (solutions en annexe par couleur, c'est-à-dire par nombre de côtés).

Je savais en l'entamant ce sujet difficile mais refermer cette étude après trois mois de travail (en dilettante, certes, mais tout de même) avec juste une liste de formes trouvées manuellement pour ce jeu que je connais depuis dix ans... Je ne suis pas très fier de ce résultat qu'un enfant de sept ans aurait aussi bien pu obtenir et dois bien me résoudre à clore ce chapitre sur cette triste note, fausse humilité ou orgueil blessé et aveu de faiblesse car je persiste à croire que le but n'est pas si éloigné. Je sais bien qu'en reprenant tout depuis le début et en effectuant d'autres choix j'aurai pu mieux traiter ce sujet. Que ceux qui voudraient reprendre le flambeau de cette étude se mettent au travail !



FIGURE 6.31 – Nouveau problème avec le Geotemplet : former une figure convexe avec un trou convexe, le plus grand possible ; à gauche, le trou mesure 10 DTE tandis qu'à droite il mesure 12 DTE, solutions en annexe). NB : Le problème suppose que la forme respecte la règle d'assemblage homogène : on accole uniquement des bords de même nature c'est-à-dire qui ont, soit tous deux des longueurs entières (comme les bords du triangle équilatéral ou l'hypoténuse du DTE), soit tous deux des longueurs irrationnelles (comme l'hypoténuse du grand demi-triangle équilatéral, les grands côtés du rectangle). Les pièces ne peuvent s'accoler que si elles mettent un bord ou une partie en commun, mais pas seulement un sommet.

4. Annexes

a. Solutions

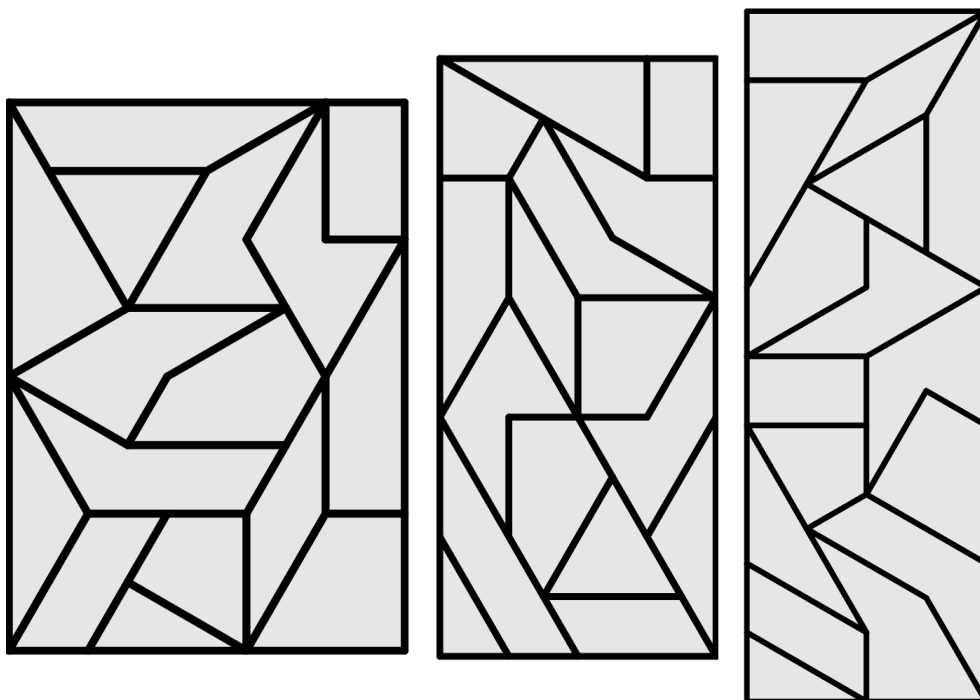


FIGURE 6.32 – *Solution pour les trois rectangles.*

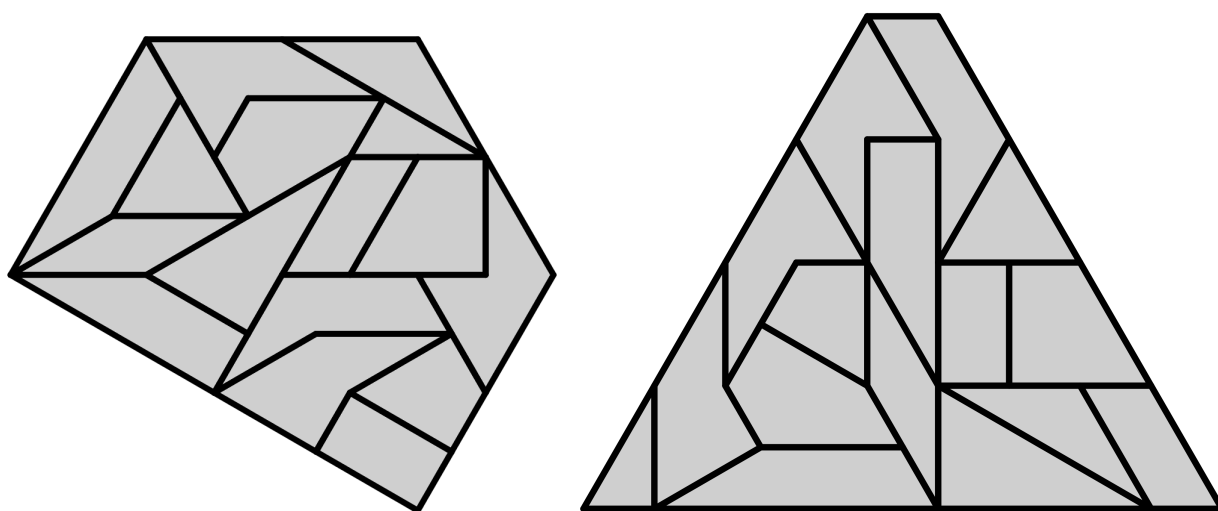


FIGURE 6.33 – *Solution pour les deux formes convexes.*

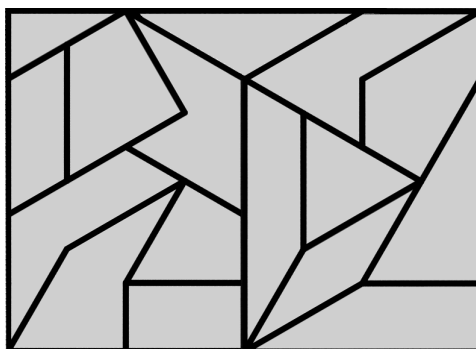
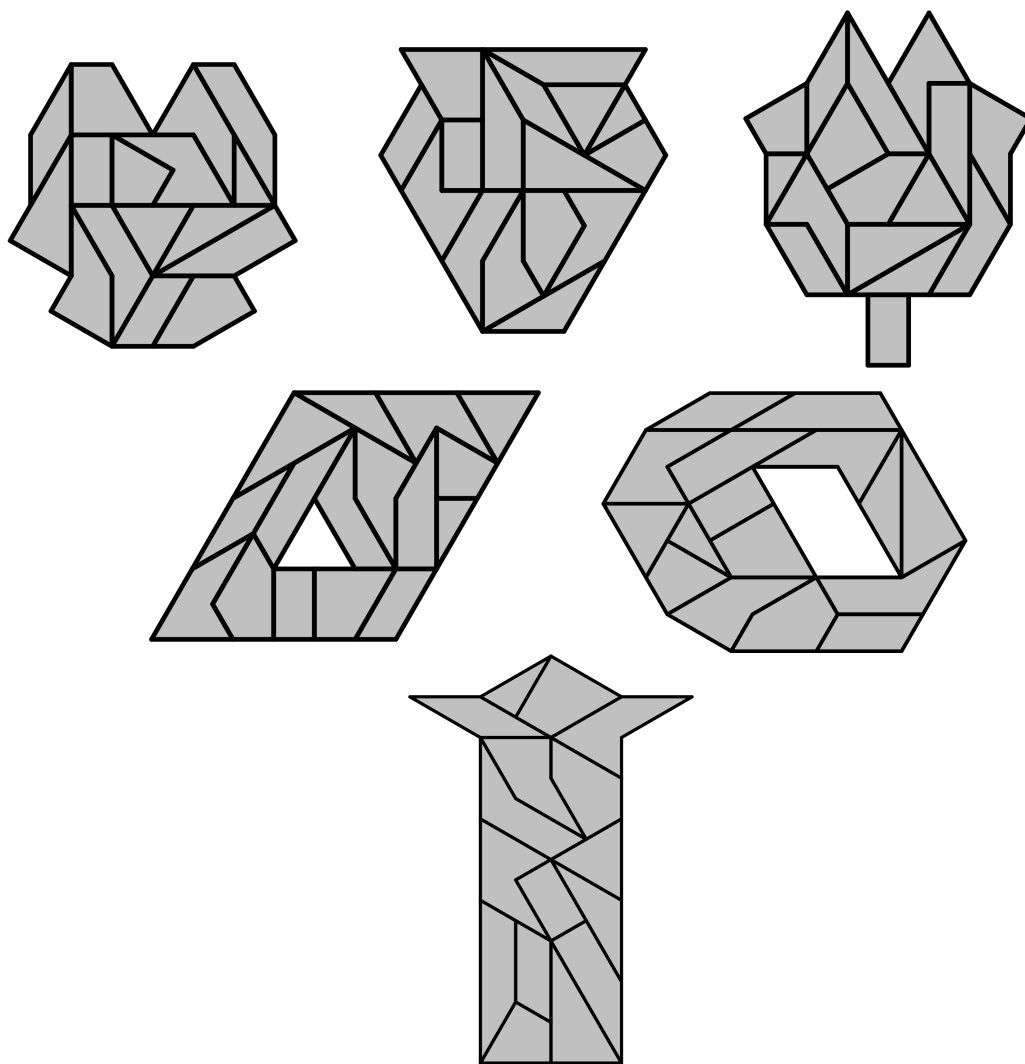
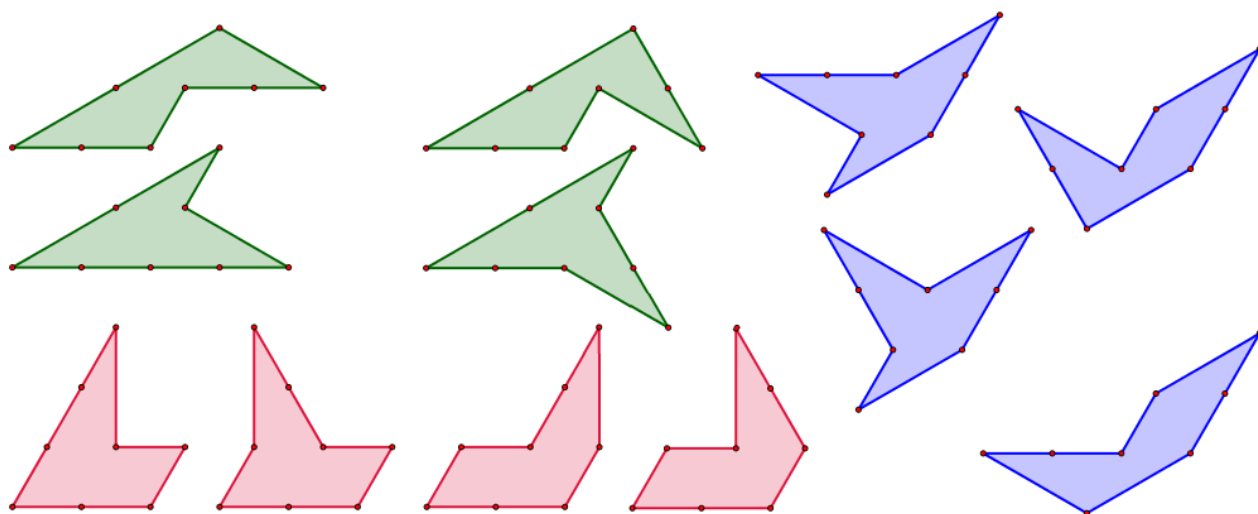


FIGURE 6.34 – *Autre solution pour le rectangle initial.*

FIGURE 6.35 – *Solution pour les formes non-convexes.*

b. Compléments sur les triDTE

FIGURE 6.36 – *Les 12 triDTE qui ont été oubliés dans le SuperGeotemplet. En les y incluant, on obtient le jeu de 28 pièces appelé HyperGeotemplet qui totalise 76 DTE et dont je ne connais pas les capacités combinatoires.*

c. Solutions des formes convexes du Geotemplet

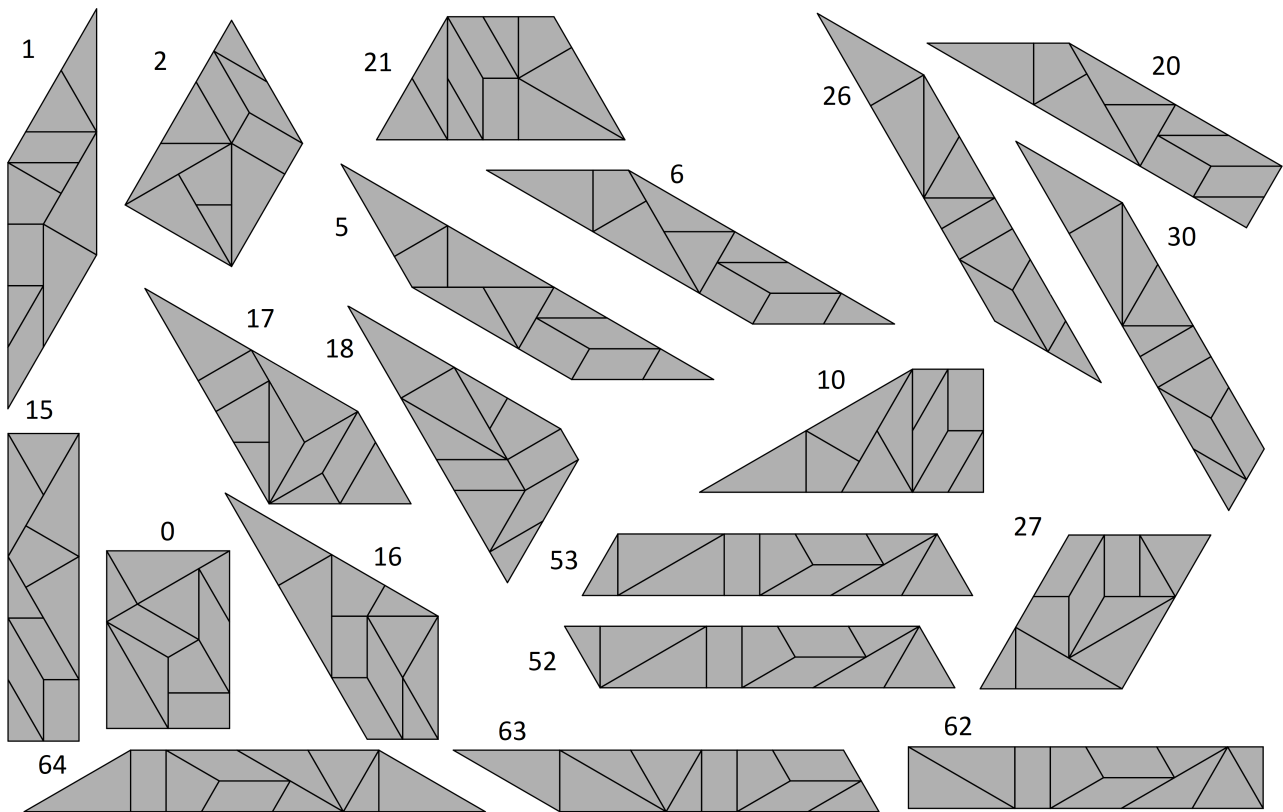


FIGURE 6.37 – Les 20 quadrilatères convexes réalisables avec le Geotemplet (j'ai ajouté le rectangle original, celui de la figure 6.1).

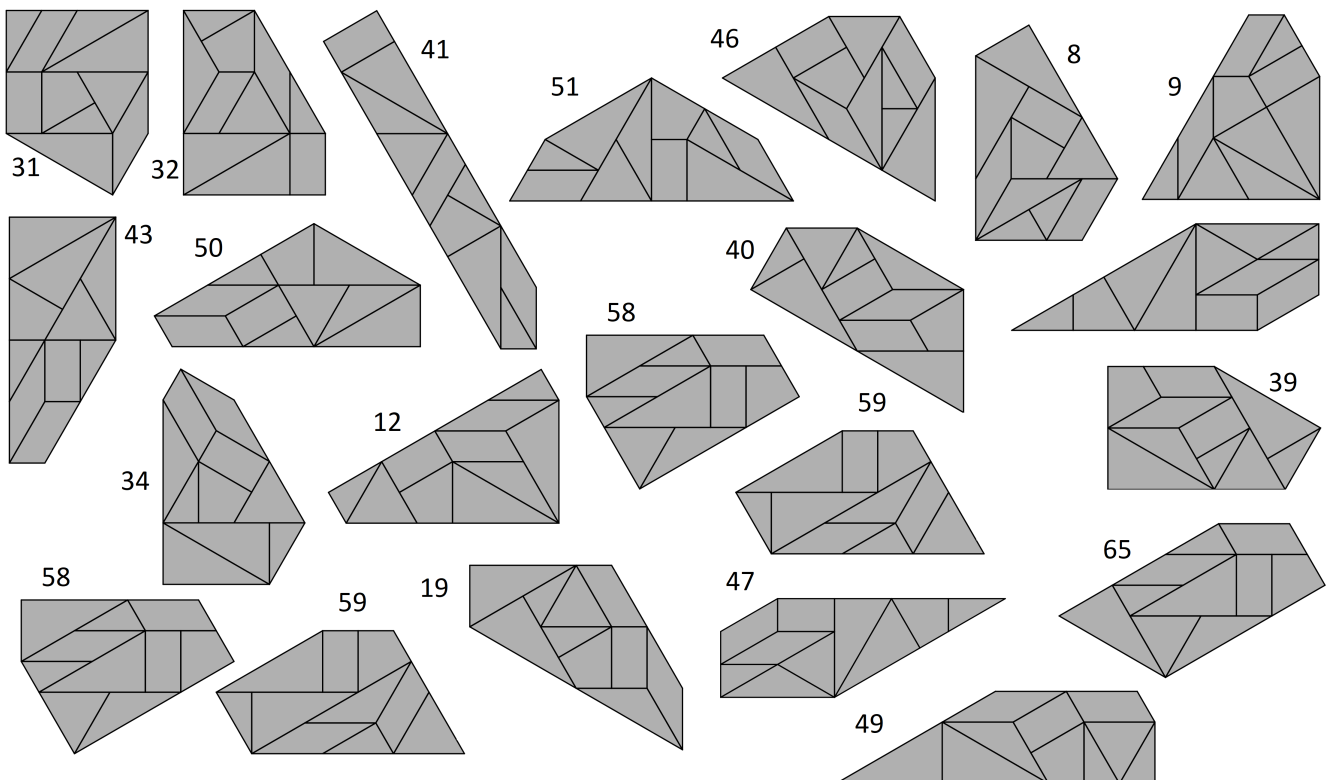


FIGURE 6.38 – Pentagones convexes réalisables avec le Geotemplet.

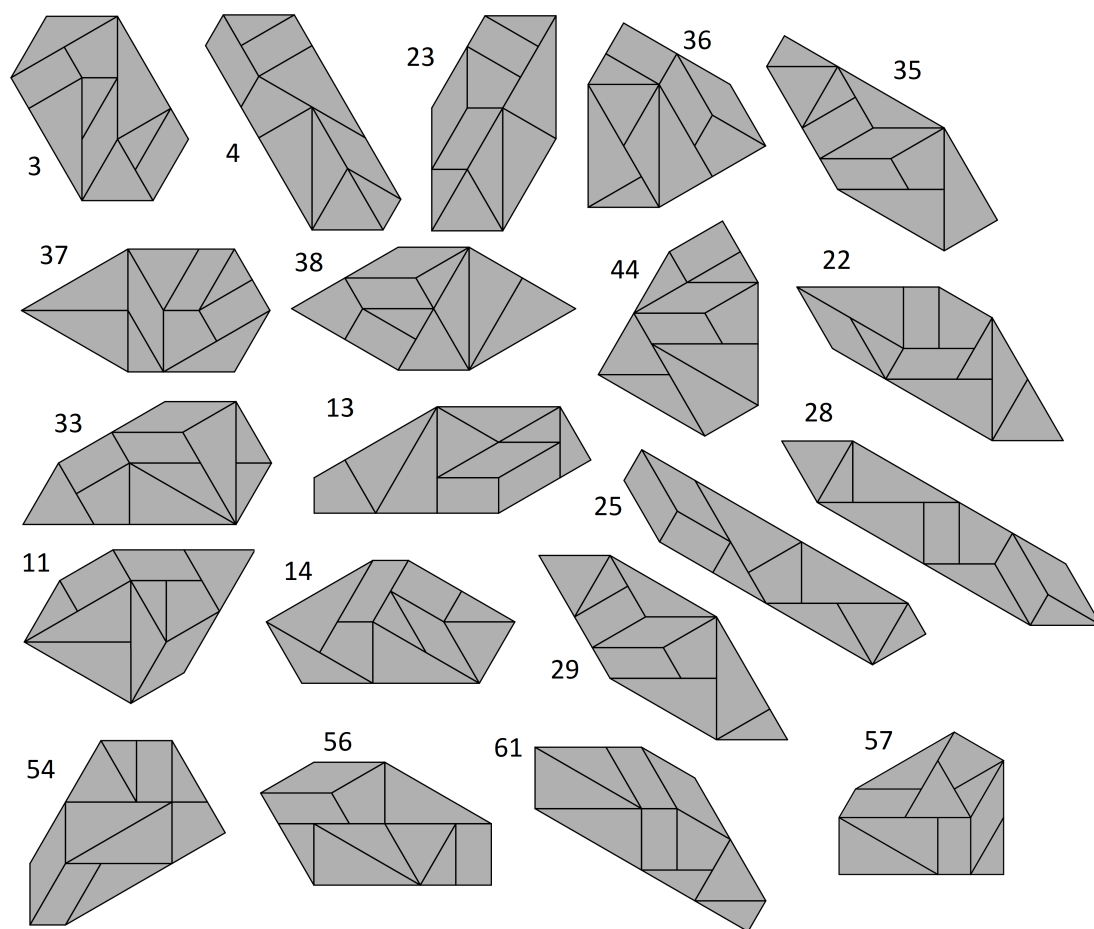


FIGURE 6.39 – Hexagones convexes réalisables avec le Geotemplet.

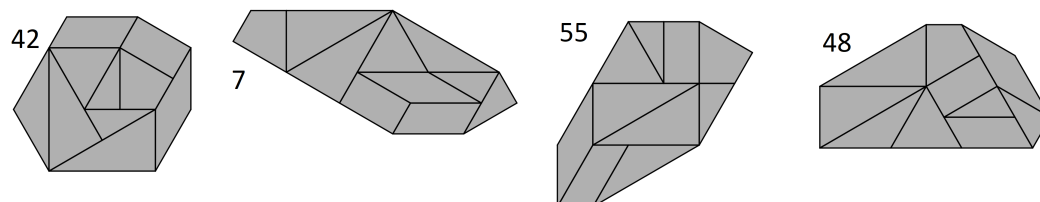


FIGURE 6.40 – Heptagones convexes réalisables avec le Geotemplet.

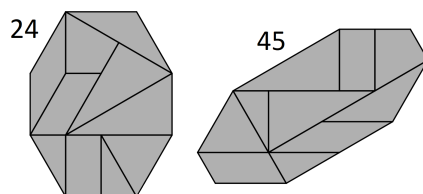


FIGURE 6.41 – Octogones convexes réalisable avec le Geotemplet.

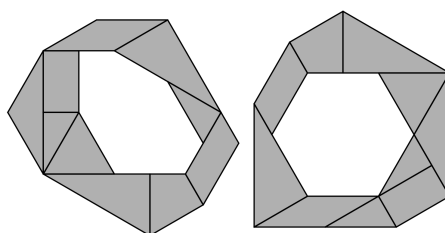


FIGURE 6.42 – Octogone convexe percé d'un heptagone convexe (à gauche); heptagone convexe percé d'un hexagone convexe (à droite)