

UQÀM

Explorations mathématiques à l'aide de l'informatique
MAT 4812

*Les solides de Platon :
La dualité des solides et les
développements*

Le manuel du concepteur

Travail présenté à : Monsieur André Boileau
Fait par : **Amélie Blanchard**

Hiver 2006

TABLE DES MATIÈRES

REMERCIEMENTS.....	3
INTRODUCTION.....	4
1. LE LOGICIEL POV-RAY	5
2. LES MATHÉMATIQUES DERRIÈRE LA DUALITÉ DES SOLIDES DE PLATON.....	6
2.1 Les mathématiques derrière la dualité entre le cube et l’octaèdre.....	6
2.2 Les mathématiques derrière la dualité entre l’icosaèdre et le dodécaèdre.....	9
2.3 Les mathématiques derrière la dualité du tétraèdre.....	13
3. LES MATHÉMATIQUES DERRIÈRE LES DÉVELOPPEMENTS.....	18
3.2 Le développement du cube	20
3.3 Le développement de l’octaèdre.....	21
3.4 Le développement du dodécaèdre	22
3.5 Le développement de l’icosaèdre	24
4. LES CODES	26
4.1 Le code commenté pour la dualité du cube et de l’octaèdre	27
4.2 Le code de la dualité entre l’icosaèdre et le dodécaèdre	36
4.3 Le code de la dualité entre les tétraèdres.....	44
4.4 Le code du développement du tétraèdre.....	44
4.5 Le code du développement du cube.	50
4.6 Le code du développement de l’octaèdre.....	53
4.7 Le code du développement du dodécaèdre.....	56
4.8 Le code du développement de l’icosaèdre	59
BIBLIOGRAPHIE	63

REMERCIEMENTS

Je tiens à remercier monsieur André Boileau, professeur à la faculté d'éducation de l'Université du Québec à Montréal, qui a dirigé ce travail. Ses précieux conseils m'ont permis de mettre à terme ce projet.

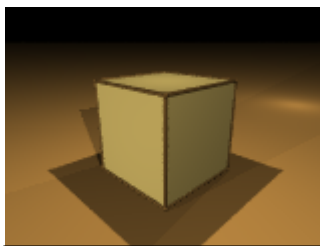
Merci !

INTRODUCTION

Ce manuel aborde les étapes de la conception de films portant sur les solides de Platon dans l'environnement *Pov-Ray*. Ces films peuvent être classés en deux catégories : la dualité des solides de Platon et le développement de ces derniers. Les 5 solides de Platon, aussi appelés polyèdres convexes réguliers sont le tétraèdre, l'hexaèdre (communément appelé le cube), l'octaèdre, le dodécaèdre et l'icosaèdre. En voici une description sommaire :



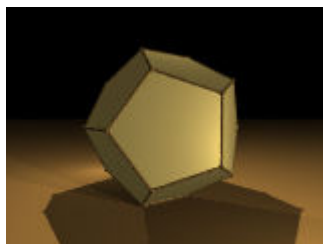
Le tétraèdre
4 triangles équilatéraux
4 sommets
6 arêtes



L'hexaèdre (le cube)
6 carrés
8 sommets
12 arêtes



L'octaèdre
8 triangles équilatéraux
6 sommets
12 arêtes



Le dodécaèdre
12 pentagones réguliers
20 sommets
30 arêtes



L'icosaèdre
20 triangles équilatéraux
12 sommets
30 arêtes

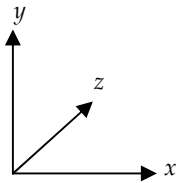
Au cours des pages qui suivent, vous saurez comment ces films ont été produits à l'aide du logiciel *Pov-Ray*.

Bonne lecture !

1. LE LOGICIEL POV-RAY

Le logiciel *Pov-Ray*, « *Persistence Of Vision – Ray Tracing* », permet de créer des images qui peuvent être utilisées afin d’engendrer des films en utilisant un logiciel de montage; on peut penser à *Quick-Time player* ou tout autre programme disponible sur le web. Le logiciel *Pov-Ray* balaye la scène décrite dans un fichier par des rayons. Pour chacun des rayons, il calcule la couleur du pixel correspondant en fonction de ce qu’il rencontre. En discernant un objet, il prendra en compte sa couleur ainsi que la réflexion et la réfraction et il remontera vers les sources lumineuses pour connaître l’éclairage.¹ Suite à ces calculs, l’image sera produite.

Trois types d’éléments sont pris en compte dans *Pov-Ray* : la caméra, les sources lumineuses ainsi que les objets. Ces éléments sont présents dans un espace tridimensionnel où les coordonnées sont écrites sous la forme $\langle x, y, z \rangle$. Il est à noter que le système d’axes tridimensionnels utilisé par *Pov-Ray* n’est pas usuel. En voici la représentation :



Pour connaître les paramètres des différents objets à insérer, il suffit d’utiliser le menu *insert* et d’y inscrire les différentes informations demandées. Il est à noter que des commentaires sont présents suite à l’insertion d’un objet et qu’il suffit d’y indiquer les paramètres pour afficher l’objet en cliquant sur le bouton *run*. C’est ainsi que l’image est créée. L’utilisation de *clock* permet d’engendrer une suite d’images via la variation de la variable *clock*.

Suite à la production des images, il suffit d’utiliser le programme *QuickTime Player Pro* pour rassembler les images et de créer un film. Le logiciel *Pov-Ray* permet de faire une panoplie d’images... il suffit de laisser aller notre imagination.

¹ Povray.free.fr

2. LES MATHÉMATIQUES DERRIÈRE LA DUALITÉ DES SOLIDES DE PLATON

Il serait important de définir ce qu'est la dualité d'un polyèdre. En fait, le mot dual vient du latin *dualis* qui veut dire deux : « Se dit de propriétés qui sont par deux et qui présentent un caractère de réciprocité. »² En fait, le polyèdre dual d'un polyèdre convexe régulier s'obtient en reliant le centre de ses faces qui ont un sommet commun. Ce qui est fascinant est que les polyèdres duals des solides de Platon sont des solides de Platon. En fait, pour qu'il y ait dualité, il faut que le nombre de faces d'un solide soit égal au nombre de sommets de son polyèdre dual. Il y a donc dualité entre le cube et l'octaèdre, entre l'icosaèdre et le dodécaèdre tandis que le tétraèdre est son propre dual.

Les films produits sont en fait une série de dualités. Pour créer ces films, j'ai décidé de faire une suite de polyèdres duals. Cependant, j'ai fait en sorte de déplacer la caméra, le plan ainsi que les sources lumineuses afin que le second polyèdre dual soit à la même position que le premier solide, ce qui permet de faire une boucle avec les images. En fait, seulement trois différents solides ont été créés pour chacun des films mais ces derniers feront en sorte que l'on croit qu'il y a une grande quantité de solides créés.

2.1 Les mathématiques derrière la dualité entre le cube et l'octaèdre

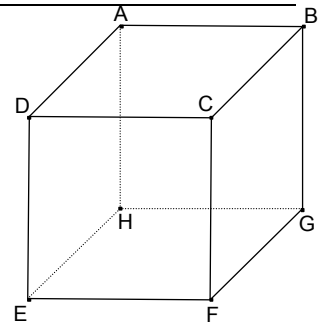
Au départ, il faut déterminer les coordonnées du premier polyèdre convexe régulier. Pour ma part, j'ai commencé par déterminer les coordonnées des différents sommets du cube. Lorsque le cube est inscrit dans une sphère dont le rayon est r , les coordonnées en x , y et z du cube seront identiques au signe près.

$$\begin{aligned}r^2 &= x^2 + y^2 + z^2 \\r^2 &= 3x^2 \\x^2 &= r^2/3 \\x &= r/\sqrt{3} \\x &= \sqrt{3}r/3\end{aligned}$$

Étant donné que j'ai décidé que la sphère circonscrite au cube aura un rayon égal à 1, les coordonnées des sommets du cube seront :

² ROBERT, Paul. *Le petit Robert I*, Montréal, Les dictionnaires ROBERT-CANADA S.C.C., 1989, 2172 pages.

A $(-\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3)$	E $(-\sqrt{3}/3, -\sqrt{3}/3, -\sqrt{3}/3)$
B $(\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3)$	F $(\sqrt{3}/3, -\sqrt{3}/3, -\sqrt{3}/3)$
C $(\sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/3)$	G $(\sqrt{3}/3, -\sqrt{3}/3, \sqrt{3}/3)$
D $(-\sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/3)$	H $(-\sqrt{3}/3, -\sqrt{3}/3, \sqrt{3}/3)$



Le cube possède 6 faces : ABCD, ABGH, ADEH, BCFG, CDEF et EFGH

Pour calculer les sommets du polyèdre dual du cube, qui est l'octaèdre, il suffit de calculer la moyenne des coordonnées des sommets de chacune des faces. En voici un exemple :

$(-\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3); (\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3); (\sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/3); (-\sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/3);$

Pour la coordonnée en x $\frac{-\sqrt{3}/3 + \sqrt{3}/3 + \sqrt{3}/3 + -\sqrt{3}/3}{4} = 0$

Pour la coordonnée en y $\frac{\sqrt{3}/3 + \sqrt{3}/3 + \sqrt{3}/3 + \sqrt{3}/3}{4} = \sqrt{3}/3$

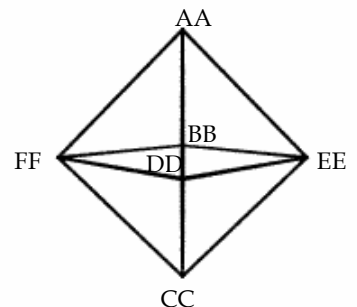
Pour la coordonnée en z $\frac{\sqrt{3}/3 + \sqrt{3}/3 + -\sqrt{3}/3 + -\sqrt{3}/3}{4} = 0$

Les coordonnées de ce sommet, qui est situé au centre de la face ABCD, seront $(0, \sqrt{3}/3, 0)$.

Le sommet AA de l'octaèdre est situé au centre de la face ABCD du cube.
 Le sommet BB de l'octaèdre est situé au centre de la face ABGH du cube.
 Le sommet CC de l'octaèdre est situé au centre de la face HGFE du cube.
 Le sommet DD de l'octaèdre est situé au centre de la face CDEF du cube.
 Le sommet EE de l'octaèdre est situé au centre de la face BCFG du cube.
 Le sommet FF de l'octaèdre est situé au centre de la face ADEH du cube.

Voici donc les coordonnées des sommets de l'octaèdre :

AA $(0, \sqrt{3}/3, 0)$	DD $(\sqrt{3}/3, 0, 0)$
BB $(0, 0, \sqrt{3}/3)$	EE $(\sqrt{3}/3, 0, 0)$
CC $(0, 0, -\sqrt{3}/3)$	FF $(-\sqrt{3}/3, 0, 0)$



L'octaèdre possède 8 faces : AABBEF, AADDEE, AADDFD, AABBEF, CCBBEF, CCDDEE, CCDDFF et CCBBEF.

Par la suite, pour déterminer les coordonnées des sommets du second cube, qui est le polyèdre dual de l'octaèdre, il suffit de procéder de la même manière.

Le sommet AAA du cube #2 est situé au centre de la face AABBBF de l'octaèdre.
 Le sommet BBB du cube #2 est situé au centre de la face AABBBE de l'octaèdre.
 Le sommet CCC du cube #2 est situé au centre de la face AADDEE de l'octaèdre.
 Le sommet DDD du cube #2 est situé au centre de la face AADDDF de l'octaèdre.
 Le sommet EEE du cube #2 est situé au centre de la face CCDDFF de l'octaèdre.
 Le sommet FFF du cube #2 est situé au centre de la face CCDDDE de l'octaèdre.
 Le sommet GGG du cube #2 est situé au centre de la face CCBDEE de l'octaèdre.
 Le sommet HHH du cube #2 est situé au centre de la face CCBDDF de l'octaèdre.

Voici les coordonnées des sommets du second cube, qui est le polyèdre dual de l'octaèdre décrit précédemment :

AAA $(-\sqrt{3}/9, \sqrt{3}/9, \sqrt{3}/9)$	EEE $(-\sqrt{3}/9, -\sqrt{3}/9, -\sqrt{3}/9)$
BBB $(\sqrt{3}/9, \sqrt{3}/9, \sqrt{3}/9)$	FFF $(\sqrt{3}/9, -\sqrt{3}/9, -\sqrt{3}/9)$
CCC $(\sqrt{3}/9, \sqrt{3}/9, -\sqrt{3}/9)$	GGG $(\sqrt{3}/9, -\sqrt{3}/9, \sqrt{3}/9)$
DDD $(-\sqrt{3}/9, \sqrt{3}/9, -\sqrt{3}/9)$	HHH $(-\sqrt{3}/9, -\sqrt{3}/9, \sqrt{3}/9)$

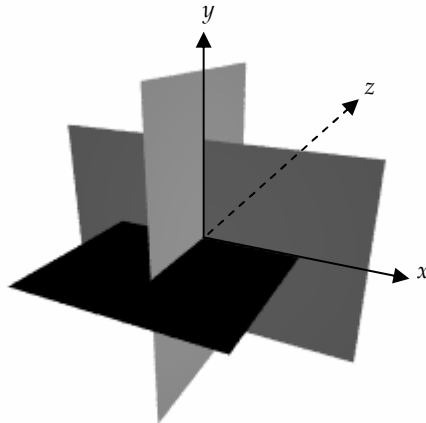
Enfin, voici les différents rapports qui seront utilisés lors de l'écriture du code :

$$\text{Rapport} = \frac{\text{Octaèdre}}{\text{Cube}} = \frac{\sqrt{0^2 + \left(\frac{\sqrt{3}}{3}\right)^2 + 0^2}}{\sqrt{\left(\frac{\sqrt{3}}{3}\right)^2 + \left(\frac{\sqrt{3}}{3}\right)^2 + \left(\frac{\sqrt{3}}{3}\right)^2}} = \sqrt{\frac{1}{3}} = \frac{1}{\sqrt{3}}$$

$$\text{Rapport 2} = \frac{\text{Cube\#2}}{\text{Cube}} = \frac{\sqrt{\left(\frac{\sqrt{3}}{9}\right)^2 + \left(\frac{\sqrt{3}}{9}\right)^2 + \left(\frac{\sqrt{3}}{9}\right)^2}}{\sqrt{\left(\frac{\sqrt{3}}{3}\right)^2 + \left(\frac{\sqrt{3}}{3}\right)^2 + \left(\frac{\sqrt{3}}{3}\right)^2}} = \sqrt{\frac{1}{9}} = \frac{1}{3}$$

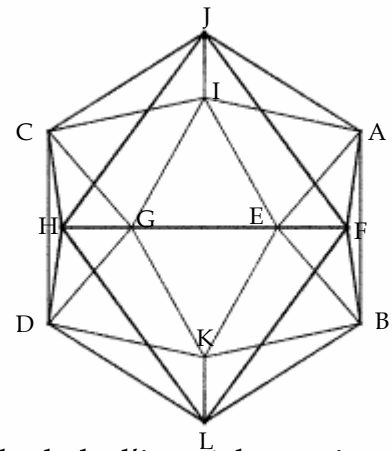
2.2 Les mathématiques derrière la dualité entre l'icosaèdre et le dodécaèdre

La construction de l'icosaèdre peut se faire à partir de trois rectangles dont le rapport longueur/largeur est le nombre d'or $\left(\varphi = \frac{1+\sqrt{5}}{2}\right)$ et qui sont disposés de manière orthogonale deux à deux. De ce fait, si la largeur du rectangle est de 2 unités et la longueur est de 2 fois le nombre d'or unités, alors ce rectangle respecte le rapport décrit ci-dessus et permet de déterminer les coordonnées des sommets de l'icosaèdre.



Les coordonnées des sommets de l'icosaèdre seront :

A ($\varphi, 1, 0$)	E ($1, 0, \varphi$)	I ($0, \varphi, 1$)
B ($\varphi, -1, 0$)	F ($1, 0, -\varphi$)	J ($0, \varphi, -1$)
C ($-\varphi, 1, 0$)	G ($-1, 0, \varphi$)	K ($0, -\varphi, 1$)
D ($-\varphi, -1, 0$)	H ($-1, 0, -\varphi$)	L ($0, -\varphi, -1$)



Pour calculer les coordonnées des sommets du polyèdre dual de l'icosaèdre, qui est le dodécaèdre, il suffit de calculer la moyenne des coordonnées des sommets de chacune des faces. En voici un exemple : $(\varphi, 1, 0) ; (0, \varphi, 1) ; (0, \varphi, -1)$

Pour la coordonnée en x : $\frac{\varphi + 0 + 0}{3} = \frac{\varphi}{3}$

Pour la coordonnée en y : $\frac{1 + \varphi + \varphi}{3} = \frac{2\varphi + 1}{3}$

Pour la coordonnée en z : $\frac{0 + 1 + -1}{3} = 0$

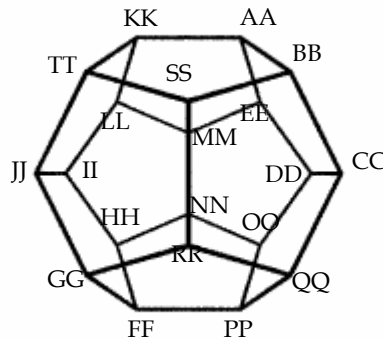
Les coordonnées du sommet, situé au centre de la face AIJ, sont $(\varphi/3, (2\varphi+1)/3, 0)$.

L'icosaèdre possède 20 faces : AIJ, AFJ, ABF, ABE, AEI, BKL, BFL, CDG, CJH, CIJ, CGI, CDH, DKL, DHL, DGK, EGI, EGK, EKL, FHL et FHJ.

Le sommet AA du dodécaèdre est situé au centre de la face AIJ de l'icosaèdre.
 Le sommet BB du dodécaèdre est situé au centre de la face AFJ de l'icosaèdre.
 Le sommet CC du dodécaèdre est situé au centre de la face ABF de l'icosaèdre.
 Le sommet DD du dodécaèdre est situé au centre de la face ABE de l'icosaèdre.
 Le sommet EE du dodécaèdre est situé au centre de la face AEI de l'icosaèdre.
 Le sommet FF du dodécaèdre est situé au centre de la face DKL de l'icosaèdre.
 Le sommet GG du dodécaèdre est situé au centre de la face DLH de l'icosaèdre.
 Le sommet HH du dodécaèdre est situé au centre de la face DKG de l'icosaèdre.
 Le sommet II du dodécaèdre est situé au centre de la face DGC de l'icosaèdre.
 Le sommet JJ du dodécaèdre est situé au centre de la face DCH de l'icosaèdre.
 Le sommet KK du dodécaèdre est situé au centre de la face JCI de l'icosaèdre.
 Le sommet LL du dodécaèdre est situé au centre de la face CIG de l'icosaèdre.
 Le sommet MM du dodécaèdre est situé au centre de la face IGE de l'icosaèdre.
 Le sommet NN du dodécaèdre est situé au centre de la face GEK de l'icosaèdre.
 Le sommet OO du dodécaèdre est situé au centre de la face EKB de l'icosaèdre.
 Le sommet PP du dodécaèdre est situé au centre de la face KBL de l'icosaèdre.
 Le sommet QQ du dodécaèdre est situé au centre de la face BLF de l'icosaèdre.
 Le sommet RR du dodécaèdre est situé au centre de la face LFH de l'icosaèdre.
 Le sommet SS du dodécaèdre est situé au centre de la face FHJ de l'icosaèdre.
 Le sommet TT du dodécaèdre est situé au centre de la face HJC de l'icosaèdre.

AA ($\varphi/3, (2\varphi+1)/3, 0$)	KK ($-\varphi/3, (2\varphi+1)/3, 0$)
BB ($((\varphi+1)/3, (\varphi+1)/3, (-\varphi-1)/3$)	LL ($(-\varphi-1)/3, (\varphi+1)/3, (\varphi+1)/3$)
CC ($((2\varphi+1)/3, 0, -\varphi/3$)	MM ($0, \varphi/3, (2\varphi+1)/3$)
DD ($((2\varphi+1)/3, 0, \varphi/3$)	NN ($0, -\varphi/3, (2\varphi+1)/3$)
EE ($((\varphi+1)/3, (\varphi+1)/3, (\varphi+1)/3$)	OO ($((\varphi+1)/3, (-\varphi-1)/3, (\varphi+1)/3$)
FF ($(-\varphi/3, (-2\varphi-1)/3, 0$)	PP ($(\varphi/3, (-2\varphi-1)/3, 0$)
GG ($((-\varphi-1)/3, (-\varphi-1)/3, (-\varphi-1)/3$)	QQ ($((\varphi+1)/3, (-\varphi-1)/3, (-\varphi-1)/3$)
HH ($((-\varphi-1)/3, (-\varphi-1)/3, (\varphi+1)/3$)	RR ($0, -\varphi/3, (-2\varphi-1)/3$)
II ($((-2\varphi-1)/3, 0, \varphi/3$)	SS ($0, \varphi/3, (-2\varphi-1)/3$)
JJ ($((-2\varphi-1)/3, 0, -\varphi/3$)	TT ($((-\varphi-1)/3, (\varphi+1)/3, (-\varphi-1)/3$)

Le dodécaèdre possède 12 faces : AABBCCDDEE, CCQQPPOODD, IILLKKTJJ, FFGGJJIIHH, DDOOMMNNEE, BBSSRRQQCC, HHNNMMLLII, GGRRSSTTJJ, AAKKLLMMEE, AAKKTTSSBB, FFRRQQPPFF et FFPPOONNHH.



Par la suite, il suffit de calculer les coordonnées des sommets du second icosaèdre, qui est le polyèdre dual du dodécaèdre, de la même manière.

Le sommet AAA de l'icosaèdre #2 est situé au centre de la face AABBCCDDEE du dodécaèdre.

Le sommet BBB de l'icosaèdre #2 est situé au centre de la face CCQQPPOODD du dodécaèdre.

Le sommet CCC de l'icosaèdre #2 est situé au centre de la face IILLKKTJJ du dodécaèdre.

Le sommet DDD de l'icosaèdre #2 est situé au centre de la face FFGGJJIIHH du dodécaèdre.

Le sommet EEE de l'icosaèdre #2 est situé au centre de la face DDOOMMNNEE du dodécaèdre..

Le sommet FFF de l'icosaèdre #2 est situé au centre de la face BBSSRRQQCC du dodécaèdre.

Le sommet GGG de l'icosaèdre #2 est situé au centre de la face HHNNMMLLII du dodécaèdre.

Le sommet HHH de l'icosaèdre #2 est situé au centre de la face GGRRSSTTJJ du dodécaèdre.

Le sommet III de l'icosaèdre #2 est situé au centre de la face AAKKLLMMEE du dodécaèdre.

Le sommet JJJ de l'icosaèdre #2 est situé au centre de la face AAKKTTSSBB du dodécaèdre.

Le sommet KKK de l'icosaèdre #2 est situé au centre de la face FFRRQQPPFF du dodécaèdre.

Le sommet LLL de l'icosaèdre #2 est situé au centre de la face FFPPOONNHH du dodécaèdre.

AAA ($-\sqrt{3}/9, \sqrt{3}/9, \sqrt{3}/9$)

EEE ($-\sqrt{3}/9, -\sqrt{3}/9, -\sqrt{3}/9$)

BBB ($\sqrt{3}/9, \sqrt{3}/9, \sqrt{3}/9$)

FFF ($\sqrt{3}/9, -\sqrt{3}/9, -\sqrt{3}/9$)

CCC ($\sqrt{3}/9, \sqrt{3}/9, -\sqrt{3}/9$)

GGG ($\sqrt{3}/9, -\sqrt{3}/9, \sqrt{3}/9$)

DDD ($-\sqrt{3}/9, \sqrt{3}/9, -\sqrt{3}/9$)

HHH ($-\sqrt{3}/9, -\sqrt{3}/9, \sqrt{3}/9$)

Enfin, voici les différents rapports qui seront utilisés lors de l'écriture du code :

$$\text{Rapport2} = \frac{\text{Dodécaèdre}}{\text{Icosaèdre}} = \sqrt{\frac{\left(\frac{\left(\frac{\sqrt{5}+1}{2}\right)^2}{3}\right) + \left(\frac{2\left(\frac{\sqrt{5}+1}{2}\right)+1}{3}\right)^2}{\left(\frac{\sqrt{5}+1}{2}\right)^2 + 1^2 + 0^2}} = \sqrt{\frac{5+2\sqrt{5}}{15}}$$

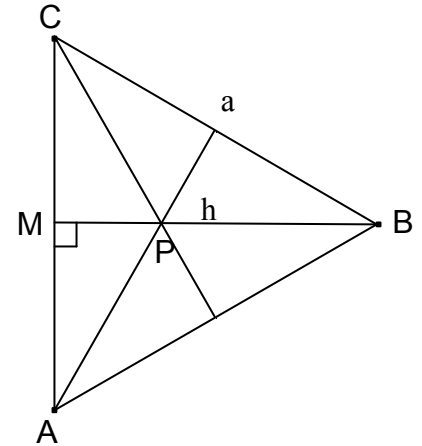
$$\text{Rapport} = \frac{\text{Icosaèdre2}}{\text{Icosaèdre}} = \sqrt{\frac{\left(\frac{7\left(\frac{\sqrt{5}+1}{2}\right)+4}{15}\right)^2 + \left(\frac{4\left(\frac{\sqrt{5}+1}{2}\right)+3}{15}\right)^2}{\left(\frac{\sqrt{5}+1}{2}\right)^2 + 1^2 + 0^2}} = \sqrt{\frac{9+4\sqrt{5}}{45}}$$

2.3 Les mathématiques derrière la dualité du tétraèdre

Voici la démonstration qui permet de déterminer les coordonnées du tétraèdre :

Soit le triangle ABC qui est équilatéral.
Soit a , la mesure du côté du triangle ABC.
Soit h , la mesure de la hauteur du triangle ABC.
Soit P , le point d'intersection des 3 médiatrices du triangle ABC.

Les trois médiatrices ont été tracées et se coupent en P . Un théorème précise que les 3 médiatrices d'un triangle équilatéral se coupent en un seul point et ce point se situe au $\frac{2}{3}$ de la médiatrice partant d'un sommet du triangle.



Commençons par déterminer la mesure de h .

Par Pythagore, nous obtenons :

$$(\overline{BM})^2 = (\overline{AB})^2 - (\overline{MA})^2$$

$$h^2 = a^2 - \left(\frac{a}{2}\right)^2$$

$$h = \frac{\sqrt{3}}{2}a$$

Pour obtenir un tétraèdre, il faut ajouter un autre sommet, soit le sommet D . Calculons la hauteur H du tétraèdre.

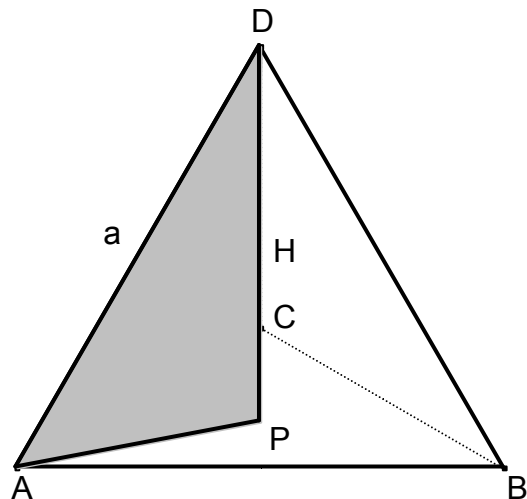
$$\overline{AP} = \frac{2}{3} \times \frac{\sqrt{3}}{2}a = \frac{\sqrt{3}}{3}a$$

Par Pythagore, nous obtenons :

$$(\overline{DP})^2 = (\overline{AD})^2 - (\overline{AP})^2$$

$$H^2 = a^2 - \left(\frac{\sqrt{3}}{3}a\right)^2$$

$$H = \frac{\sqrt{6}}{3}a$$

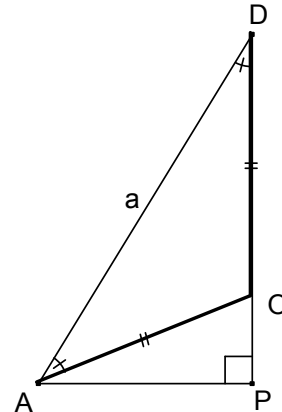


Soit O , le centre du tétraèdre. Il faut que \overline{AO} soit égal à \overline{OD} car il s'agit des rayons de la sphère qui sera circonscrite au tétraèdre.

Commençons par déterminer la mesure de l'angle ADP .

$$\sin(m\angle ADP) = \frac{\frac{\sqrt{3}}{3}a}{a}$$

$$m\angle ADP \approx 35,26^\circ$$



Étant donné qu'il faut que la mesure de \overline{OD} soit égale à la mesure de \overline{AO} , le triangle AOD sera un triangle isocèle. De ce fait, la mesure de l'angle DAO sera égale à la mesure de l'angle ADO . Mais il est possible d'affirmer que l'angle ADP est congru à l'angle ADO car il s'agit d'un angle commun aux deux triangles.

De ce fait, $m\angle ADP = m\angle ADO = m\angle DAO \approx 35,26^\circ$

Par le fait même, nous pouvons affirmer que $m\angle AOD \approx 109,47^\circ$

Pour connaître la mesure du segment \overline{AO} et, par le fait même, la mesure du segment \overline{OD} , il suffit d'utiliser la loi des sinus.

$$\frac{\sin(109,47)}{a} = \frac{\sin(35,36)}{m\overline{AO}}$$

$$m\overline{AO} = \frac{\sqrt{6}}{4}a$$

$\frac{\sqrt{6}}{4}a$ est exactement la mesure du rayon de la sphère circonscrite au tétraèdre. Cependant, comme j'ai décidé que le tétraèdre sera inscrit dans une sphère de rayon 1, qui est la mesure de \overline{AO} , je peux conclure que a vaut $\frac{2\sqrt{6}}{3}$.

Il serait utile de calculer le rapport $\frac{m\overline{OA}}{m\overline{DP}}$. Cela donne $\frac{\frac{\sqrt{6}}{4}}{\frac{\sqrt{6}}{3}} = \frac{3}{4}$. En d'autres termes, le point O

se situe au $\frac{3}{4}$ de la hauteur issue d'un sommet en partant de ce sommet.

Calculons maintenant les coordonnées des 4 sommets du tétraèdre.

Le sommet D sera placé en $(0,0,1)$.

Comme la coordonnée en z du sommet D est 1, il est possible de déterminer la coordonnée en z des trois autres sommets qui seront placés sur le même plan par une simple proportion.

$$\frac{\frac{3}{4}}{\frac{1}{4}} = \frac{1}{z}$$

Ici, z vaut donc $1/3$. Il faut cependant que la coordonnée soit négative pour obtenir un tétraèdre. z vaut donc $-1/3$.

La coordonnée en x du sommet B est située au $2/3$ de la hauteur issue du sommet B du triangle équilatéral :

$$x_1 = \frac{2}{3} \times \frac{\sqrt{3}}{2} \times \frac{2\sqrt{6}}{3} = \frac{2\sqrt{2}}{3}$$

Pour déterminer les coordonnées en x des sommets A et C , il faut se rappeler que ces derniers sont placés à $1/3$ de la hauteur issue du sommet B du triangle équilatéral.

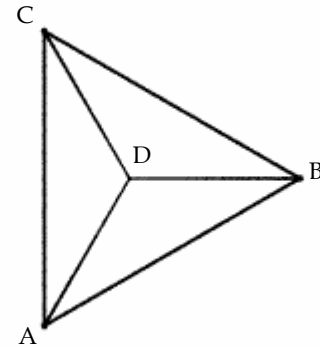
$x_2 = x_3 = \frac{1}{3} \times \frac{\sqrt{3}}{2} \times \frac{2\sqrt{6}}{3} = \frac{\sqrt{2}}{3}$. Cependant, la valeur de la coordonnée doit être négative pour respecter les distances. Donc, $x_2 = x_3 = -\frac{\sqrt{2}}{3}$.

Il suffit maintenant de déterminer les valeurs des coordonnées en y des trois sommets. Nous avons placé le sommet B de telle sorte que la valeur en y soit 0. Nous savons également que la valeur de a est $\frac{2\sqrt{6}}{3}$. Nous n'avons qu'à diviser cette valeur par 2 étant donné que les sommets A et C sont placés de part et d'autre de l'axe des x de manière symétrique. De ce fait, les valeurs des coordonnées des sommets en y sont $\pm \frac{\sqrt{6}}{3}$.

Les coordonnées des 4 sommets du tétraèdre régulier, inscrit dans une sphère de rayon 1, sont donc

$$A \left(-\frac{\sqrt{2}}{3}, -\frac{\sqrt{6}}{3}, -\frac{1}{3} \right) \quad C \left(-\frac{\sqrt{2}}{3}, \frac{\sqrt{6}}{3}, -\frac{1}{3} \right)$$

$$B \left(\frac{2\sqrt{2}}{3}, 0, -\frac{1}{3} \right) \quad D (0, 0, 1)$$



Pour calculer les coordonnées des sommets du polyèdre dual du tétraèdre, qui est le tétraèdre, il suffit de calculer la moyenne des coordonnées des sommets de chacune des faces.

En voici un exemple : $\left(-\frac{\sqrt{2}}{3}, -\frac{\sqrt{6}}{3}, -\frac{1}{3} \right), \left(-\frac{\sqrt{2}}{3}, \frac{\sqrt{6}}{3}, -\frac{1}{3} \right), (0, 0, 1)$

$$\text{Pour la coordonnée en } x \quad \frac{-\frac{\sqrt{2}}{3} + -\frac{\sqrt{2}}{3} + 0}{3} = -\frac{2\sqrt{2}}{9}$$

$$\text{Pour la coordonnée en } y \quad \frac{-\frac{\sqrt{6}}{3} + \frac{\sqrt{6}}{3} + 0}{3} = 0$$

$$\text{Pour la coordonnée en } z \quad \frac{-\frac{1}{3} + -\frac{1}{3} + 1}{3} = -\frac{1}{9}$$

Les coordonnées du sommet, situé au centre de la face AII, sont $\left(-\frac{2\sqrt{2}}{9}, 0, -\frac{1}{9} \right)$

Le tétraèdre possède 4 faces : ABC, ABD, ACD et BCD.

Le sommet AA du tétraèdre #2 est situé au centre de la face ABC du tétraèdre #1.

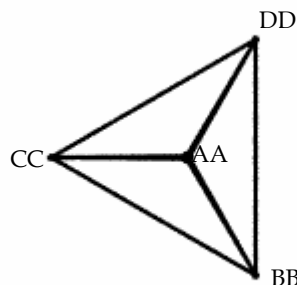
Le sommet BB du tétraèdre #2 est situé au centre de la face ABD du tétraèdre #1.

Le sommet CC du tétraèdre #2 est situé au centre de la face ACD du tétraèdre #1.

Le sommet DD du tétraèdre #2 est situé au centre de la face BCD du tétraèdre #1.

$$AA (0, 0, -1/3) \quad CC \left(-\frac{2\sqrt{2}}{9}, 0, \frac{1}{9} \right)$$

$$BB \left(\frac{\sqrt{2}}{9}, -\frac{\sqrt{6}}{9}, \frac{1}{9} \right) \quad DD \left(\frac{\sqrt{2}}{9}, \frac{\sqrt{6}}{9}, \frac{1}{9} \right)$$



Par la suite, il suffit de calculer les coordonnées des sommets du troisième tétraèdre, qui est le polyèdre dual du tétraèdre, de la même manière.

Le sommet AAA du tétraèdre #3 est situé au centre de la face AABBC du tétraèdre #2.

Le sommet BBB du tétraèdre #3 est situé au centre de la face AABBD du tétraèdre #2.

Le sommet CCC du tétraèdre #3 est situé au centre de la face AACCD du tétraèdre #2.

Le sommet DDD du tétraèdre #3 est situé au centre de la face BBCCD du tétraèdre #2.

$$AAA \left(-\frac{\sqrt{2}}{27}, -\frac{\sqrt{6}}{27}, -\frac{1}{27} \right) \qquad CCC \left(-\frac{\sqrt{2}}{27}, \frac{\sqrt{6}}{27}, -\frac{1}{27} \right)$$

$$BBB \left(\frac{2\sqrt{2}}{27}, 0, -\frac{1}{27} \right) \qquad DDD (0, 0, 1/9)$$

Enfin, voici les différents rapports qui seront utilisés lors de l'écriture du code :

$$\text{Rapport} = \frac{\text{Tétraèdre \#2}}{\text{Tétraèdre}} = \sqrt{\frac{0^2 + 0^2 + \left(\frac{1}{3}\right)^2}{0^2 + 0^2 + 1^2}} = \sqrt{\frac{1}{9}} = \frac{1}{3}$$

$$\text{Rapport 2} = \frac{\text{Tétraèdre \#3}}{\text{Tétraèdre}} = \sqrt{\frac{0^2 + 0^2 + \left(\frac{1}{9}\right)^2}{0^2 + 0^2 + 1^2}} = \sqrt{\frac{1}{81}} = \frac{1}{9}$$

3. LES MATHÉMATIQUES DERRIÈRE LES DÉVELOPPEMENTS

Cinq films seront produits concernant le développement des polyèdres convexes réguliers. Les coordonnées des sommets, calculées précédemment, seront utiles pour les développements. En fait, il suffit de connaître les coordonnées des sommets de départ ainsi que les coordonnées des sommets d'arrivée pour écrire le code. Pour produire les séquences d'images, chacun des sommets du développement se déplacera de manière linéaire pour arriver à former le polyèdre voulu. Il suffit donc de déterminer la fonction linéaire qui décrit le déplacement des coordonnées des sommets du polyèdre convexe régulier. Trois fonctions linéaires seront définies pour chacun des sommets. Voici un exemple :

D1 $(-4\sqrt{2}/3, 0, -1/3)$

D1' $(0, 0, 1)$

variable clock	Coordonnée en x	Coordonnée en y	Coordonnée en z
0	$-4\sqrt{2}/3$	0	$-1/3$
1	0	0	1

$$x = \frac{0 + \frac{4\sqrt{2}}{3}}{1 - 0} \text{clock} - \frac{4\sqrt{2}}{3} = \frac{4\sqrt{2}}{3} \text{clock} - \frac{4\sqrt{2}}{3}$$

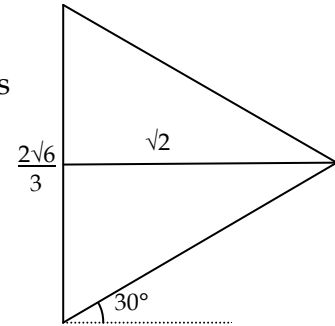
$$y = 0\text{clock} - 0 = 0$$

$$z = \frac{1 + \frac{1}{3}}{1 - 0} \text{clock} - \frac{1}{3} = \frac{4}{3} \text{clock} - \frac{1}{3}$$

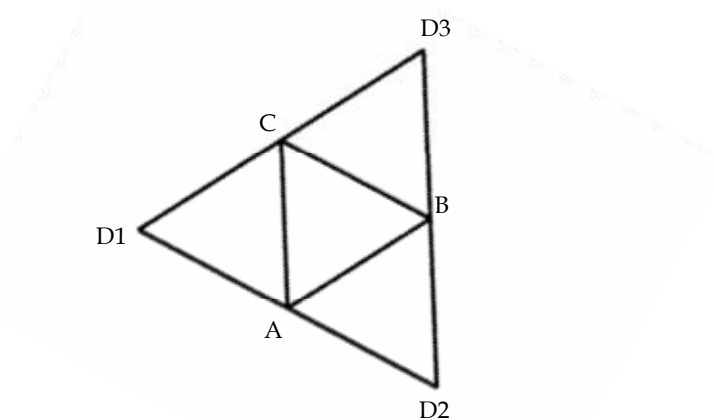
Suite à la détermination des équations, il suffit de les intégrer au code pour obtenir les déplacements des sommets.

3.1 Le développement du tétraèdre

Voici les mesures utiles pour déterminer les coordonnées des sommets du tétraèdre.

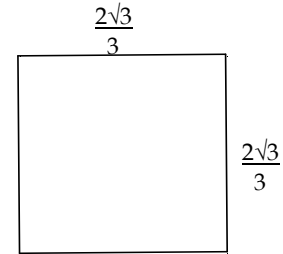


Coordonnées des sommets de départ	Coordonnées des sommets d'arrivée.
A $(-\sqrt{2}/3, -\sqrt{6}/3, -1/3)$	A' $(-\sqrt{2}/3, -\sqrt{6}/3, -1/3)$
B $(2\sqrt{2}/3, 0, -1/3)$	B' $(2\sqrt{2}/3, 0, -1/3)$
C $(-\sqrt{2}/3, \sqrt{6}/3, -1/3)$	C' $(-\sqrt{2}/3, \sqrt{6}/3, -1/3)$
D1 $(-4\sqrt{2}/3, 0, -1/3)$	D1' $(0, 0, 1)$
D2 $(\sqrt{2}/3, -2\sqrt{6}/3, -1/3)$	D2' $(0, 0, 1)$
D3 $(\sqrt{2}/3, 2\sqrt{6}/3, -1/3)$	D3' $(0, 0, 1)$

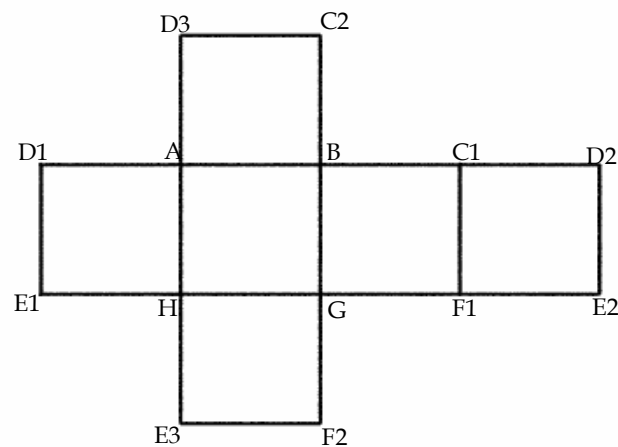


3.2 Le développement du cube

Voici les mesures utiles pour déterminer les coordonnées des sommets du cube.

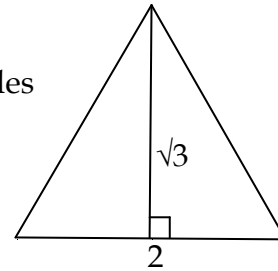


Coordonnées des sommets de départ	Coordonnées des sommets d'arrivée.
A ($-\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3$)	A' ($-\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3$)
B ($\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3$)	B' ($\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3$)
C1 ($\sqrt{3}, \sqrt{3}/3, \sqrt{3}/3$)	C1' ($\sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/3$)
C2 ($\sqrt{3}/3, \sqrt{3}, \sqrt{3}/3$)	C2' ($\sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/3$)
D1 ($-\sqrt{3}, \sqrt{3}/3, \sqrt{3}/3$)	D1' ($-\sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/3$)
D2 ($5\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3$)	D2' ($-\sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/3$)
D3 ($-\sqrt{3}/3, \sqrt{3}, \sqrt{3}/3$)	D3' ($-\sqrt{3}/3, \sqrt{3}/3, -\sqrt{3}/3$)
E1 ($-\sqrt{3}, -\sqrt{3}/3, \sqrt{3}/3$)	E1' ($-\sqrt{3}/3, -\sqrt{3}/3, -\sqrt{3}/3$)
E2 ($5\sqrt{3}/3, -\sqrt{3}/3, \sqrt{3}/3$)	E2' ($-\sqrt{3}/3, -\sqrt{3}/3, -\sqrt{3}/3$)
E3 ($-\sqrt{3}/3, -\sqrt{3}, \sqrt{3}/3$)	E3' ($-\sqrt{3}/3, -\sqrt{3}/3, -\sqrt{3}/3$)
F1 ($\sqrt{3}, -\sqrt{3}/3, \sqrt{3}/3$)	F1' ($\sqrt{3}/3, -\sqrt{3}/3, -\sqrt{3}/3$)
F2 ($\sqrt{3}/3, -\sqrt{3}, \sqrt{3}/3$)	F2' ($\sqrt{3}/3, -\sqrt{3}/3, -\sqrt{3}/3$)
G ($\sqrt{3}/3, -\sqrt{3}/3, \sqrt{3}/3$)	G' ($\sqrt{3}/3, -\sqrt{3}/3, \sqrt{3}/3$)
H ($-\sqrt{3}/3, -\sqrt{3}/3, \sqrt{3}/3$)	H' ($-\sqrt{3}/3, -\sqrt{3}/3, \sqrt{3}/3$)

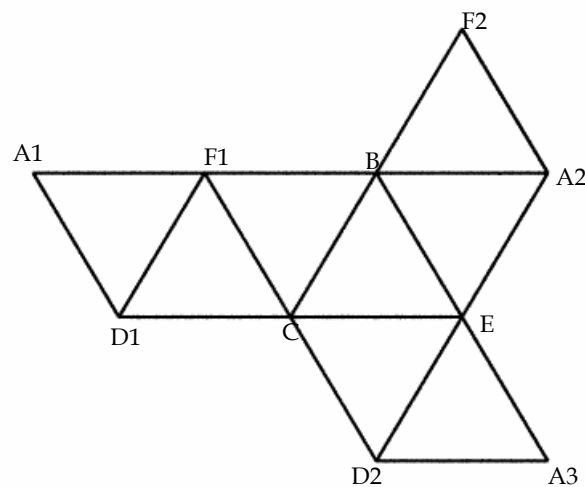


3.3 Le développement de l'octaèdre

Voici les mesures utiles pour déterminer les coordonnées des sommets de l'octaèdre.

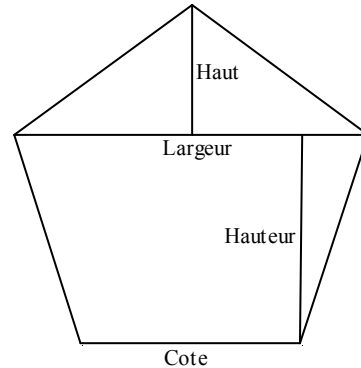


Coordonnées des sommets de départ	Coordonnées des sommets d'arrivée.
A1 (-3, $\sqrt{3}$, 0)	A1' (0, 1, 0)
A2 (3, $\sqrt{3}$, 0)	A2' (0, 1, 0)
A3 (3, $-\sqrt{3}$, 0)	A3' (0, 1, 0)
B (1, $\sqrt{3}$, 0)	B' (0, 0, 1)
C (0, 0, 0)	C' (0, -1, 0)
D1 (-2, 0, 0)	D1' (0, 0, -1)
D2 (1, $-\sqrt{3}$, 0)	D2' (0, 0, -1)
E (2, 0, 0)	E' (1, 0, 0)
F1 (-1, $\sqrt{3}$, 0)	F1' (-1, 0, 0)
F2 (2, $2\sqrt{3}$, 0)	F2' (-1, 0, 0)



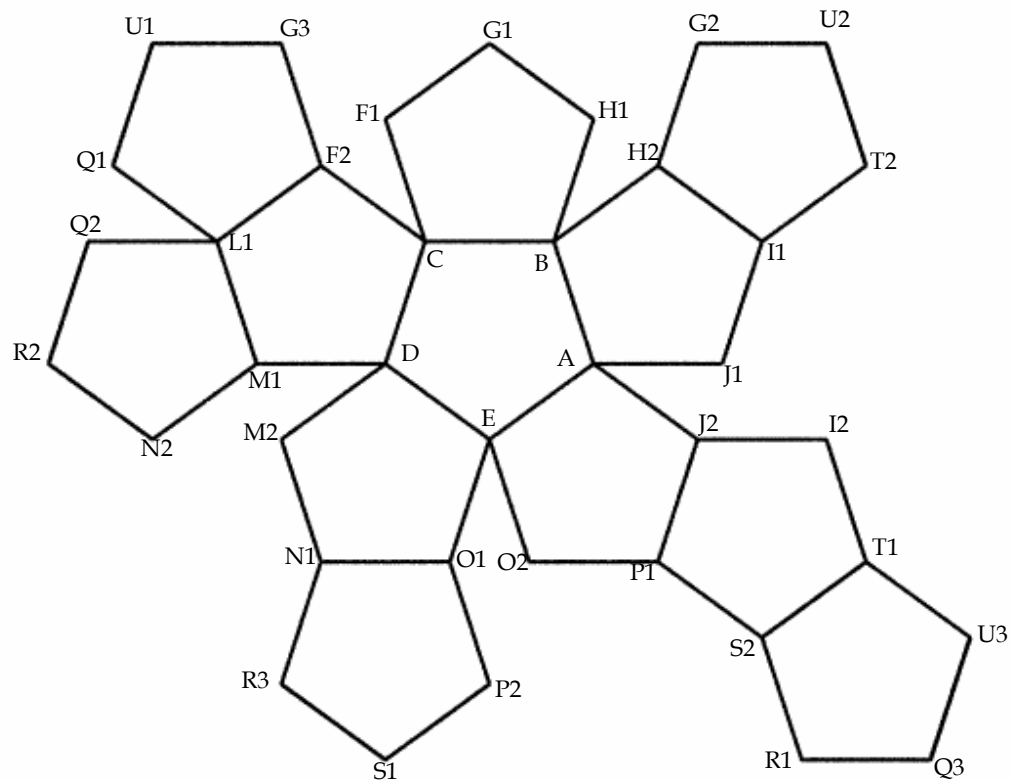
3.4 Le développement du dodécaèdre

$$\begin{aligned} \text{cote} &= 2 \varphi \\ \text{largeur} &= 4\varphi \cos(36) \\ \text{hauteur} &= 2 \varphi \sin(72) \\ \text{haut} &= 2 \varphi \sin(36) \\ \text{OR} &= \frac{\sqrt{5} + 1}{2} \end{aligned}$$



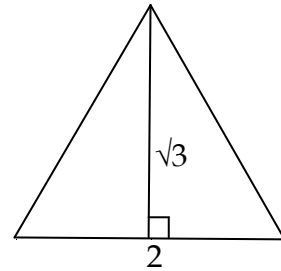
Coordonnées des sommets de départ	Coordonnées des sommets d'arrivée.
A (largeur/2, -hauteur, OR)	A' (OR+1, OR+1, OR+1)
B (cote/2, 0, OR)	B' (OR, 2OR+1, 0)
C (-cote/2, 0, OR)	C' (-OR, 2OR+1, 0)
D (-largeur/2, -hauteur, OR)	D' (-OR-1, OR+1, OR+1)
E (0, -haut-hauteur, OR)	E' (0, OR, 2OR+1)
F1 (-largeur/2, hauteur, OR)	F1' (-OR-1, OR+1, -OR-1)
F2 (-cote/2-largeur/2, haut, OR)	F2' (-OR-1, OR+1, -OR-1)
G1 (0, haut+hauteur, OR)	G1' (0, OR, -2OR-1)
G2 (largeur, haut+hauteur, OR)	G2' (0, OR, -2OR-1)
G3 (-largeur, haut+hauteur, OR)	G3' (0, OR, -2OR-1)
H1 (largeur/2, hauteur, OR)	H1' (OR+1, OR+1, -OR-1)
H2 (cote/2+largeur/2, haut, OR)	H2' (OR+1, OR+1, -OR-1)
I1 (cote/2+largeur, 0, OR)	I1' (2OR+1, 0, -OR)
I2 (largeur+cote, -hauteur-haut, OR)	I2' (2OR+1, 0, -OR)
J1 (largeur/2+cote, -hauteur, OR)	J1' (2OR+1, 0, OR)
J2 (largeur, -hauteur-haut, OR)	J2' (2OR+1, 0, OR)
L1 (-cote/2-largeur, 0, OR)	L1' (-2OR-1, 0, -OR)
M1 (-largeur/2-cote, -hauteur, OR)	M1' (-2OR-1, 0, OR)
M2 (-largeur, -hauteur-haut, OR)	M2' (-2OR-1, 0, OR)
N1 (-cote/2-largeur/2, -2hauteur-haut, OR)	N1' (-OR-1, -OR-1, OR+1)
N2 (-largeur-cote, -hauteur-haut, OR)	N2' (-OR-1, -OR-1, OR+1)
O1 (cote/2-largeur/2, -2hauteur-haut, OR)	O1' (0, -OR, 2OR+1)
O2 (-cote/2+largeur/2, -2hauteur-haut, OR)	O2' (0, -OR, 2OR+1)
P1 (cote/2+largeur/2, -2hauteur-haut, OR)	P1' (OR+1, -OR-1, OR+1)
P2 (0, -3hauteur-haut, OR)	P2' (OR+1, -OR-1, OR+1)
Q1 (-3largeur/2-cote/2, haut, OR)	Q1' (-OR-1, -OR-1, -OR-1)

Q2 (-3cote/2-largeur, 0, OR)	Q2' (-OR-1, -OR-1, -OR-1)
Q3 (3largeur/2+cote, -3hauteur-2haut, OR)	Q3' (-OR-1, -OR-1, -OR-1)
R1 (3largeur/2, -3hauteur-2haut, OR)	R1' (-OR, -2OR-1, 0)
R2 (-3largeur/2-cote, -hauteur, OR)	R2' (-OR, -2OR-1, 0)
R3(-largeur, -3hauteur-haut, OR)	R3' (-OR, -2OR-1, 0)
S1 (-largeur/2, -3hauteur-2haut, OR)	S1' (OR, -2OR-1, 0)
S2 (cote/2+largeur,-2hauteur-2haut, OR)	S2' (OR, -2OR-1, 0)
T1 (cote/2+3largeur/2, -2hauteur-haut, OR)	T1' (OR+1, -OR-1, -OR-1)
T2 (cote/2+3largeur/2, haut, OR)	T2' (OR+1, -OR-1, -OR-1)
U1 (-largeur-cote, hauteur+haut, OR)	U1' (0, -OR, -2OR-1)
U2 (largeur+cote, haut+hauteur, OR)	U2' (0, -OR, -2OR-1)
U3 (cote/2+2largeur, -2hauteur-2haut, OR)	U3' (0, -OR, -2OR-1)

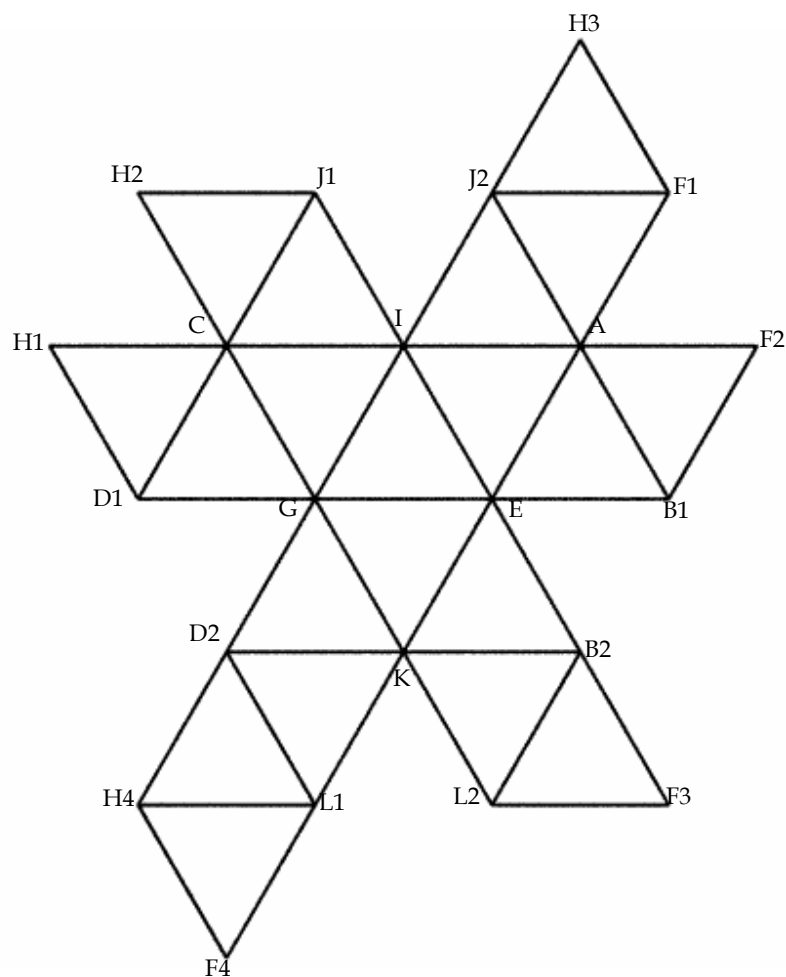


3.5 Le développement de l'icosaèdre

Voici les mesures utiles pour déterminer les coordonnées des sommets de l'icosaèdre.



Coordonnées des sommets de départ	Coordonnées des sommets d'arrivée.
A (2, $\sqrt{3}$, OR)	A' (OR, 1, 0)
B1 (3, 0, OR)	B1' (OR, -1, 0)
B2 (2, $-\sqrt{3}$, OR)	B2' (OR, -1, 0)
C (-2, $\sqrt{3}$, OR)	C' (-OR, 1, 0)
D1 (-3, 0, OR)	D1' (-OR, -1, 0)
D2 (-2, $-\sqrt{3}$, OR)	D2' (-OR, -1, 0)
E (1, 0, OR)	E' (1, 0, OR)
F1 (3, $2\sqrt{3}$, OR)	F1' (1, 0, -OR)
F2 (4, $\sqrt{3}$, OR)	F2' (1, 0, -OR)
F3 (3, $-2\sqrt{3}$, OR)	F3' (1, 0, -OR)
F4 (-2, $-3\sqrt{3}$, OR)	F4' (1, 0, -OR)
G (-1, 0, OR)	G' (-1, 0, OR)
H1 (-4, $\sqrt{3}$, OR)	H1' (-1, 0, -OR)
H2 (-3, $2\sqrt{3}$, OR)	H2' (-1, 0, -OR)
H3 (2, $3\sqrt{3}$, OR)	H3' (-1, 0, -OR)
H4 (-3, $-2\sqrt{3}$, OR)	H4' (-1, 0, -OR)
I (0, $\sqrt{3}$, OR)	I' (0, OR, 1)
J1 (-1, $2\sqrt{3}$, OR)	J1' (0, OR, -1)
J2 (1, $2\sqrt{3}$, OR)	J2' (0, Or, -1)
K (0, $-\sqrt{3}$, OR)	K' (0, -OR, 1)
L1 (-1, $-2\sqrt{3}$, OR)	L1' (0, -OR, -1)
L2 (1, $-2\sqrt{3}$, OR)	L2' (0, -OR, -1)



4. LES CODES

Les codes concernant les dualités permettent de faire 11 séquences d'images (il suffit de modifier la valeur de la variable *numerofilm* du code):

1. La disparition de la sphère et l'apparition du cube.
2. L'apparition des sommets de l'octaèdre.
3. La disparition des faces du cube en ayant un effet de miroir.
4. La disparition du reflet des faces du cube qui donnait l'effet de miroir.
5. La disparition des arêtes et des sommets du cube.
6. Le déplacement de la caméra.
7. L'apparition des sommets du cube #2.
8. La disparition des faces de l'icosaèdre en ayant un effet de miroir.
9. La disparition du reflet des faces de l'icosaèdre qui donnait l'effet de miroir.
10. La disparition des arêtes et des sommets de l'icosaèdre.
11. Le déplacement des lumières et du plan.

Les codes concernant les développements permettent de faire 3 séquences d'images (il suffit de modifier la valeur de la variable *numerofilm* du code):

1. Déplacement des sommets.
2. Apparition des sommets.
3. Rotation de la caméra.

Pour le développement du cube, il y a cependant 4 séquences d'images :

1. Déplacement des sommets.
2. Apparition des sommets.
3. Déplacement de la caméra.
4. Rotation de la caméra.

4.1 Le code commenté pour la dualité du cube et de l'octaèdre

```
#declare numerofilm = 11;           // Déclaration de la variable numerofilm (séquences d'images).
#declare rapport = sqrt(3)/3;       // Déclaration de la variable rapport.
#declare rapport2 = 1/3;           // Déclaration de la variable rapport2.
global_settings {max_trace_level 25} // Limite du nombre maximum de réflexions récursives que le code
                                     // calculera.

background { color rgb <0,0,0> }    // Couleur de fond : Noire
#if (numerofilm <=10)               // Lumières pour les séquences d'images 1 à 10.
    light_source {
        0*x                         // Position de la lumière.
        color rgb <1,1,1>           // Couleur de la lumière : blanche.
        spotlight                   // Type de lumière.
        translate <-5,5,-5>         // Translation de la lumière.
        point_at <0, 0, 0>          // Direction du « spotlight ».
        radius 60                  // Point névralgique intérieur en degrés.
        tightness 50               // Étroitesse du rayon lumineux.
        falloff 8}                 // Intensité du rayon lumineux extérieur en degrés.
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <0,5,-5>
        point_at <0, 0, 0>
        radius 60
        tightness 1
        falloff 20}
#else                                //Lumières pour le film 11.
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <(-5*rapport2+5)*clock-5,(5*rapport2-5)*clock+5,(-5*rapport2+5)*clock-5>
        point_at <0, 0, 0>
        radius 60
        tightness 50
        falloff 8}
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <0,(5*rapport2-5)*clock+5,(-5*rapport2+5)*clock-5>
        point_at <0, 0, 0>
        radius 60
        tightness 1
        falloff 20}
#end
```

```
#if (numerosfilm =1)                                // Sphère pour la séquence d'images 1.
  #declare goldsphere = texture {pigment {color rgbt <0.65, 0.5, 0.2,clock>}
                                     // Définition de la couleur goldsphere.
  finish {                                           //Finition.
    ambient 0.7                                     //Intensité de la couleur.
    brilliance 15                                   // Brilliance.
    diffuse 0.9                                     // Éclairage.
    metallic                                       // Donne un aspect métallique à la couleur.
    specular -0.2*clock + 0.20                    // Effet miroir.
    roughness 1/60                                 // Rugosité du métal.
    reflection -0.05*clock+0.05                   // Réflexion du métal.
  };
  sphere { <0, 0, 0>, 1 texture {goldsphere}} // Sphère de rayon 1 et de couleur goldsphere.
#end

#if (numerosfilm <=2 | 7<=numerosfilm)              // La couleur bronze pour les séquences d'images 2 à 7.
  #declare bronze = texture {pigment {color rgbt<0.58, 0.42, 0.20,0>}} // Définition de la couleur bronze.
#else
  #if (numerosfilm =3)                              // La couleur bronze pour la séquence d'images 3.
    #declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20,clock>}} // La couleur devient
    #else                                           transparente.
    #declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20,1>}} //Aucune couleur présente.
  #end
#end
finish {ambient 0.6
  brilliance 3
  diffuse 0.2
  metallic
  roughness 1/90
  #if (numerosfilm =4)                              // La réflexion et l'effet miroir pour la séquence d'images 4.
    reflection 0.5-0.5*clock                        // La réflexion disparaît.
    specular -0.2*clock+0.2                          // L'effet miroir disparaît.
  #else
    #if (numerosfilm =5)                            // La réflexion et l'effet miroir pour la séquence d'images 5.
      specular 0                                     // Aucun effet miroir.
      reflection 0                                  // Aucune réflexion.
    #else                                           // La réflexion et l'effet miroir pour les autres séquences d'images.
      specular 0.20
      reflection 0.5
    #end
  #end}};
```

```
#if (numérofilm < 5 | 7<=numérofilm)      //La couleur gold pour les séquences d'images 1 à 4 et 7 à 11.
    #declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>} // Définition de la couleur gold.
#else
    #if (numérofilm =5)                    //La couleur gold pour la séquence d'images 5.
        #declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,clock>} // La couleur devient
#else                                     transparente.
        #declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,1>} //Aucune couleur présente.
    #end
#end
    finish { ambient 0.25
        brilliance 3
        diffuse 0.4
        metallic
        roughness 1/60
        #if (numérofilm =5)                // L'effet miroir et la réflexion pour la séquence d'images 5.
            specular -0.2*clock+0.2        // L'effet miroir disparaît.
            reflection -0.4*clock+0.4      // La réflexion disparaît.
        #else                             // L'effet miroir et la réflexion pour les autres séquences d'images.
            specular 0.20
            reflection 0.4
        #end}};
#if (numérofilm <=9)                      // La couleur bronze pour les séquences d'images 1 à 9.
    #declare bronze1 = texture {pigment {color rgbt <0.58, 0.42, 0.20,0>} // Définition de la couleur bronze1.
#else
    #if (numérofilm=10)                   // La couleur bronze pour la séquence d'images 10.
        #declare bronze1 = texture {pigment {color rgbt <0.58, 0.42, 0.20,clock>} // La couleur disparaît.
    #else
        #declare bronze1 = texture {pigment {color rgbt <0.58, 0.42, 0.20,1>} // Aucune couleur présente.
    #end
#end
    finish {
        ambient 0.25
        brilliance 3
        diffuse 0.4
        metallic
        roughness 1/60
        #if (numérofilm =10)              // L'effet miroir et la réflexion pour la séquence d'images 10.
            specular -0.2*clock+0.2        // L'effet miroir disparaît.
            reflection 0.5-0.5*clock      // La réflexion disparaît.
        #else
            specular 0.20
            reflection 0.5
        #end}};
```

```
#if (numérofilm <=7)                                // La couleur gold1 pour les séquences d'images 7 à 11.
    #declare gold1 = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>} //Définition de la couleur gold1.
#else
    #if (numérofilm =8)                              // Pour la séquence d'images 8.
        #declare gold1 = texture {pigment {color rgbt <1.00, 0.87, 0.475,clock>} // La couleur disparaît.
    #else
        #declare gold1 = texture {pigment {color rgbt <1.00, 0.87, 0.475,1>} // Aucune couleur présente.
    #end
#end
finish {
    ambient 0.25
    brilliance 3
    diffuse 0.4
    metallic
    specular 0.20
    roughness 1/60
    #if (numérofilm =9)                              // La réflexion et l'effet miroir pour la séquence d'images 9.
        reflection 0.4-0.4*clock                    // La réflexion disparaît.
        specular -0.2*clock+0.2                      // L'effet miroir disparaît.
    #else
        #if (numérofilm =10)                          // La réflexion et l'effet miroir pour la séquence d'images 10.
            reflection 0                              // Aucune réflexion.
            specular 0                                // Aucun effet miroir.
        #else
            reflection 0.4
            specular 0.2
        #end
    #end}};

#if (numérofilm<=5)                                // Positionnement de la caméra pour les séquences d'images 1 à 5.
    camera{ location <-3,2,-3>                        // Position de la caméra.
        look_at <0.0, 0.0, 0.0>}                    // Direction de la caméra.
#else
    #if (numérofilm =6)                              // Positionnement de la caméra pour la séquence d'images 6.
        camera{location <2*clock-3, (-4/3)*clock+2, 2*clock-3> // La caméra se déplace.
            look_at <0.0, 0.0, 0.0>}
    #else
        camera{location <-1, 2/3, -1>                // Positionnement de la caméra pour les films 7 à 11.
            look_at <0.0, 0.0, 0.0>}
    #end
#end
```

```

#declare rayon = 0.04;           // Déclaration de la variable rayon.
#declare rayon2 = 0.02;          // Déclaration de la variable rayon2.
#declare rayon3 = 0.04*rapport;   // Déclaration de la variable rayon3.
#declare rayon4 = 0.02*rapport;   // Déclaration de la variable rayon4.
#declare rayon5 = 0.04*rapport2;  // Déclaration de la variable rayon5.
#declare rayon6 = 0.02*rapport2;  // Déclaration de la variable rayon6.

#if (numérofilm =1)               // Les rayons pour la séquence d'images 1.
    #declare rayon = 0.04*clock;  // La valeur du rayon augmente de façon constante de 0 à 0,04.
    #declare rayon2 = 0.02*clock; // La valeur du rayon2 augmente de façon constante de 0 à 0,02.
#else
    #if (numérofilm=2)             // Les rayons pour la séquence d'images 2.
        #declare rayon3 =0.04*rapport*clock; // La valeur du rayon3 diminue de façon constante de 0 à
                                                0,04* $\sqrt{3}/3$ .
        #declare rayon4 = 0.02*rapport*clock; // La valeur du rayon4 diminue de façon constante de 0 à
                                                0,02* $\sqrt{3}/3$ .
    #else
        #if (numérofilm =7)        // Les rayons pour la séquence d'images 7.
            #declare rayon5 =0.04*rapport2*clock; // La valeur du rayon5 diminue de façon constante de 0,04
                                                    à 0.04/3.
            #declare rayon6 = 0.02*rapport2*clock; // La valeur du rayon6 diminue de façon constante de
                                                    0,02 à 0.02/3.
        #end
    #end
#end

#declare Coin=(sqrt(3)/3);        // Déclaration de la variable Coin.

#if (numérofilm <=10)             // Le plan pour les séquences d'images 1 à 10.
    plane {y, -Coin-0.08 pigment {color rgbt <1, 0.87, 0.475>}} // Emplacement du plan.
#else
    plane {y, ((-Coin-0.08)*rapport2+Coin+0.08)*clock-Coin-0.08 pigment {color rgbt <1.00, 0.87, 0.475>}}
#end

#declare A = <-Coin, Coin, Coin>; // Déclaration des coordonnées du sommet A du cube.
#declare B = <Coin, Coin, Coin>;  // Déclaration des coordonnées du sommet B du cube.
#declare C = <Coin, Coin, -Coin>; // Déclaration des coordonnées du sommet C du cube.
#declare D = <-Coin, Coin, -Coin>; // Déclaration des coordonnées du sommet D du cube.
#declare E = <-Coin, -Coin, -Coin>; // Déclaration des coordonnées du sommet E du cube.
#declare F = <Coin, -Coin, -Coin>; // Déclaration des coordonnées du sommet F du cube.
#declare G = <Coin, -Coin, Coin>;  // Déclaration des coordonnées du sommet G du cube.
#declare H = <-Coin, -Coin, Coin>; // Déclaration des coordonnées du sommet H du cube.

```

<pre>#declare CUBE = object {union { polygon { 5, A, B, C, D, A} polygon { 5, A, B, G, H, A} polygon { 5, H, G, F, E, H} polygon { 5, D, C, F, E, D} polygon { 5, C, B, G, F, C} polygon { 5, A, D, E, H, A} texture {bronze}}}</pre>	<pre>// Déclaration du CUBE en tant qu'objet. (Les faces du cube) // Union des polygones suivants : // Les polygones sont les 4 faces carrées du cube. // Couleur bronze.</pre>
<pre>#declare SCUBE = object {union { sphere { A, rayon} sphere { B, rayon} sphere { C, rayon} sphere { D, rayon} sphere { E, rayon} sphere { F, rayon} sphere { G, rayon} sphere { H, rayon} texture {gold}}}</pre>	<pre>// Déclaration de SCUBE en tant qu'objet. (Les sommets du cube) // Union des sphères suivantes: // Couleur gold.</pre>
<pre>#declare ACUBE = object {union { cylinder { A, B, rayon2} cylinder { B, C, rayon2} cylinder { C, D, rayon2} cylinder { D, A, rayon2} cylinder { H, G, rayon2} cylinder { G, F, rayon2} cylinder { F, E, rayon2} cylinder { E, H, rayon2} cylinder { A, H, rayon2} cylinder { B, G, rayon2} cylinder { C, F, rayon2} cylinder { D, E, rayon2} texture {gold}}}</pre>	<pre>// Déclaration de ACUBE en tant qu'objet. (Les arêtes du cube) // Union des cylindres suivants : // Couleur gold.</pre>
<pre>#declare AA = <0, Coin, 0>; #declare BB = <0, 0, Coin>; #declare CC = <0, -Coin, 0>; #declare DD = <0, 0, -Coin>; #declare EE = <Coin, 0, 0>; #declare FF = <-Coin, 0, 0>;</pre>	<pre>// Déclaration des coordonnées du sommet AA de l'octaèdre. // Déclaration des coordonnées du sommet BB de l'octaèdre. // Déclaration des coordonnées du sommet CC de l'octaèdre. // Déclaration des coordonnées du sommet DD de l'octaèdre. // Déclaration des coordonnées du sommet EE de l'octaèdre. // Déclaration des coordonnées du sommet FF de l'octaèdre.</pre>


```

#declare OCTA = object                                // Déclaration de OCTA en tant qu'objet. (Les faces de l'octaèdre)
{union {                                              // Union des polygones suivants :
    polygon { 4, AA, BB, EE, AA}
    polygon { 4, AA, EE, DD, AA}
    polygon { 4, AA, DD, FF, AA}
    polygon { 4, AA, FF, BB, AA}
    polygon { 4, CC, BB, EE, CC}
    polygon { 4, CC, EE, DD, CC}
    polygon { 4, CC, DD, FF, CC}
    polygon { 4, CC, FF, BB, CC}
    texture { gold1 }}}                               // Couleur gold1.
#declare SOCTA = object                             // Déclaration de SOCTA en tant qu'objet. (Les sommets de
{union {                                              // Union des sphères suivantes :                               l'octaèdre )
    sphere { AA, rayon3}
    sphere { BB, rayon3}
    sphere { CC, rayon3}
    sphere { DD, rayon3}
    sphere { EE, rayon3}
    sphere { FF, rayon3}
    texture {bronze1}}}                               // Couleur bronze1.
#declare AOCTA = object                             // Déclaration de AOCTA en tant qu'objet. (Les arêtes de
{union {                                              // Union des cylindres suivants :                               l'octaèdre).
    cylinder { AA, BB, rayon4}
    cylinder { AA, EE, rayon4}
    cylinder { AA, DD, rayon4}
    cylinder { AA, FF, rayon4}
    cylinder { CC, BB, rayon4}
    cylinder { CC, EE, rayon4}
    cylinder { CC, DD, rayon4}
    cylinder { CC, FF, rayon4}
    cylinder { BB, EE, rayon4}
    cylinder { EE, DD, rayon4}
    cylinder { DD, FF, rayon4}
    cylinder { FF, BB, rayon4}
    texture {bronze1}}}                               // Couleur bronze1.
#declare Coin2=(sqrt(3)/9);                          // Déclaration de la variable Coin2.
#declare AAA = <-Coin2, Coin2, Coin2>;               // Déclaration des coordonnées du sommet AAA du cube#2.
#declare BBB = <Coin2, Coin2, Coin2>;                // Déclaration des coordonnées du sommet BBB du cube#2.
#declare CCC = <Coin2, Coin2, -Coin2>;               // Déclaration des coordonnées du sommet CCC du cube#2.
#declare DDD = <-Coin2, Coin2, -Coin2>;               // Déclaration des coordonnées du sommet DDD du cube#2.
#declare EEE = <-Coin2, -Coin2, -Coin2>;             // Déclaration des coordonnées du sommet EEE du cube#2.
#declare FFF = <Coin2, -Coin2, -Coin2>;              // Déclaration des coordonnées du sommet FFF du cube#2.
#declare GGG = <Coin2, -Coin2, Coin2>;                // Déclaration des coordonnées du sommet GGG du cube#2.
#declare HHH = <-Coin2, -Coin2, Coin2>;              // Déclaration des coordonnées du sommet HHH du cube#2.

```

```
#declare CUBE2 = object                                     // Déclaration de CUBE2 en tant qu'objet. (Les faces)
{union {                                                    // Union des polygones suivants:
    polygon { 5, AAA, BBB, CCC, DDD, AAA}
    polygon { 5, AAA, BBB, GGG, HHH, AAA}
    polygon { 5, HHH, GGG, FFF, EEE, HHH}
    polygon { 5, DDD, CCC, FFF, EEE, DDD}
    polygon { 5, CCC, BBB, GGG, FFF, CCC}
    polygon { 5, AAA, DDD, EEE, HHH, AAA}
    texture {bronze }}}                                     //Couleur bronze.

#declare SCUBE2 = object                                    // Déclaration de SCUBE2 en tant qu'objet. (Les sommets)
{union {                                                    // Union des sphères suivantes:
    sphere { AAA, rayon5}
    sphere { BBB, rayon5}
    sphere { CCC, rayon5}
    sphere { DDD, rayon5}
    sphere { EEE, rayon5}
    sphere { FFF, rayon5}
    sphere { GGG, rayon5}
    sphere { HHH, rayon5}
    texture {gold}}}                                       // Couleur gold.

#declare ACUBE2 = object                                    // Déclaration de ACUBE en tant qu'objet. (Les arêtes)
{union {                                                    // Union des cylindres suivants:
    cylinder { AAA, BBB, rayon6}
    cylinder { BBB, CCC, rayon6}
    cylinder { CCC, DDD, rayon6}
    cylinder { DDD, AAA, rayon6}
    cylinder { HHH, GGG, rayon6}
    cylinder { GGG, FFF, rayon6}
    cylinder { FFF, EEE, rayon6}
    cylinder { EEE, HHH, rayon6}
    cylinder { AAA, HHH, rayon6}
    cylinder { BBB, GGG, rayon6}
    cylinder { CCC, FFF, rayon6}
    cylinder { DDD, EEE, rayon6}
    texture {gold}}}                                       // Couleur gold.
```

```
#if (1<=numérofilm & numérofilm <=5 ) // Apparition des objets pour les séquences d'images 1 à 5.
    CUBE // Tracer CUBE.
    SCUBE // Tracer SCUBE.
    ACUBE // Tracer ACUBE.
#end

#if ( 2<=numérofilm & numérofilm <=10) // Apparition des objets pour les séquences d'images 2 à 10.
    OCTA // Tracer OCTA.
    AOCTA // Tracer AOCTA.
    SOCTA // Tracer SOCTA.
#end

#if (7<=numérofilm ) // Apparition des objets pour les séquences d'images 7 à 11.
    CUBE2 // Tracer CUBE2.
    ACUBE2 // Tracer ACUBE2.
    SCUBE2 // Tracer SCUBE2.
#end
```

4.2 Le code de la dualité entre l'icosaèdre et le dodécaèdre

```
#declare numerofilm = 11;           // 11 films.
global_settings {max_trace_level 25}
#declare OR = ((sqrt(5)+1)/2);
#declare rapport = sqrt((9+4*sqrt(5))/45);
#declare rapport2 = sqrt((5+2*sqrt(5))/15);

#if (numerofilm <=10)
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <-5,5,-5>
        point_at <0, 0, 0>
        radius 60
        tightness 50
        falloff 8}
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <0,5,-5>
        point_at <0, 0, 0>
        radius 60
        tightness 1
        falloff 20}
#else
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <(-5*rapport+5)*clock-5,(5*rapport-5)*clock+5,(-5*rapport+5)*clock-5>
        point_at <0, 0, 0>
        radius 60
        tightness 50
        falloff 8}
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <0,(5*rapport-5)*clock+5,(-5*rapport+5)*clock-5>
        point_at <0, 0, 0>
        radius 60
        tightness 1
        falloff 20}
#end
```

```
#if (numérofilm =1)
  #declare goldsphere = texture {pigment {color rgbt <0.65, 0.5, 0.2,clock>}
  finish {
    ambient 0.7
    brilliance 15
    diffuse 0.9
    metallic
    specular -0.2*clock + 0.20
    roughness 1/60
    reflection -0.05*clock+0.05
  };
  sphere { <0, 0, 0>, pow( pow(((sqrt(5)+1)/2),2)+1,0.5) texture {goldsphere}}
#end

#if (numérofilm <=5)
  camera{ location <2,2,-5>
    look_at <0.0, 0.0, 0.0>}
#else
  #if (numérofilm =6)
    camera{location <(rapport*2-2)*clock+2,(rapport*2-2)*clock+2,(rapport*-5+5)*clock-5>
      look_at <0.0, 0.0, 0.0>}
  #else
    camera{location <(rapport*2-2)*1+2,(rapport*2-2)*1+2,(rapport*-5+5)*1-5>
      look_at <0.0, 0.0, 0.0>}
  #end
#end

#if (numérofilm <=2 | 7<=numérofilm)
  #declare bronze = texture {pigment {color rgbt<0.58, 0.42, 0.20,0>}}
#else
  #if (numérofilm =3)
    #declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20,clock>}}
  #else
    #declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20,1>}}
  #end
#end

finish {ambient 0.6
  brilliance 3
  diffuse 0.2
  metallic
  roughness 1/90
  #if (numérofilm =4)
    reflection 0.5-0.5*clock
    specular -0.2*clock+0.2
```

```
#else
    #if (numérofilm =5)
        specular 0
        reflection 0
    #else
        specular 0.20
        reflection 0.5
    #end
#end}};

#if (numérofilm < 5 | 7<=numérofilm)
    #declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>}}
#else
    #if (numérofilm =5)
        #declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,clock>}}
    #else
        #declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,1>}}
    #end
#end
finish { ambient 0.25
    brilliance 3
    diffuse 0.4
    metallic
    roughness 1/60
    #if (numérofilm =5)
        specular -0.2*clock+0.2
        reflection -0.4*clock+0.4
    #else
        specular 0.20
        reflection 0.4
    #end}};

#if (numérofilm <=9)
    #declare bronze1 = texture {pigment {color rgbt <0.58, 0.42, 0.20,0>}}
#else
    #if (numérofilm=10)
        #declare bronze1 = texture {pigment {color rgbt <0.58, 0.42, 0.20,clock>}}
    #else
        #declare bronze1 = texture {pigment {color rgbt <0.58, 0.42, 0.20,1>}}
    #end
#end
finish {
    ambient 0.25
    brilliance 3
    diffuse 0.4
    metallic
    roughness 1/60
```

```
#if (numérofilm =10)
    specular -0.2*clock+0.2
    reflection 0.5-0.5*clock
#else
    specular 0.20
    reflection 0.5
#endif}};

#if (numérofilm <=7)
    #declare gold1 = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>}}
#else
    #if (numérofilm =8)
        #declare gold1 = texture {pigment {color rgbt <1.00, 0.87, 0.475,clock>}}
    #else
        #declare gold1 = texture {pigment {color rgbt <1.00, 0.87, 0.475,1>}}
    #end
#end
finish {
    ambient 0.25
    brilliance 3
    diffuse 0.4
    metallic
    specular 0.20
    roughness 1/60
    #if (numérofilm =9)
        reflection 0.4-0.4*clock
        specular -0.2*clock+0.2
    #else
        #if (numérofilm =10)
            reflection 0
            specular 0
        #else
            reflection 0.4
            specular 0.2
        #end
    #end
#end}};

#declare rayon = 0.05;
#declare rayon2 = 0.03;
#declare rayon3 =0.05*rapport2;
#declare rayon4 = 0.03*rapport2;
#declare rayon5 =0.05*rapport;
#declare rayon6 = 0.03*rapport;
```

```

#if (numérofilm =1)
  #declare rayon = 0.05*clock;
  #declare rayon2 = 0.03*clock;
#else
  #if (numérofilm=2)
    #declare rayon3 =0.05*rapport2*clock;
    #declare rayon4 = 0.03*rapport2*clock;
  #else
    #if (numérofilm =7)
      #declare rayon5 =0.05*rapport*clock;
      #declare rayon6 = 0.03*rapport*clock;
    #end
  #end
#end
background { color rgb <0,0,0> }
#if (numérofilm<=10)
  plane {y, -sqrt(OR*OR+1)-0.05 pigment {color rgbt <1, 0.87, 0.475>}}
#else
  plane {y, (rapport*(-sqrt(OR*OR+1)-0.05)-(-sqrt(OR*OR+1)-0.05))*clock+(-sqrt(OR*OR+1)-0.05)pigment
{color rgbt <1, 0.87, 0.475>}}
#end
#declare A = <OR, 1, 0>;
#declare B = <OR, -1, 0>;
#declare C = <-OR,1, 0>;
#declare D = <-OR, -1, 0>;
#declare E = <1, 0, OR>;
#declare F = <1, 0, -OR>;
#declare G = <-1, 0, OR>;
#declare H = <-1, 0, -OR>;
#declare I = <0, OR, 1>;
#declare J = <0, OR, -1>;
#declare K = <0, -OR, 1>;
#declare L = <0, -OR, -1>;
#declare ICO= object
{union {
  polygon { 4, A, I, J, A}
  polygon { 4, A, F, J, A}
  polygon { 4, A, B, F, A}
  polygon { 4, A, B, E, A}
  polygon { 4, A, E, I, A}
  polygon { 4, D, K, L, D}
  polygon { 4, D, L, H, D}
  polygon { 4, D, K, G, D}
  polygon { 4, D, G, C, D}
  polygon { 4, D, C, H, D}
  polygon { 4, J, C, I, J}

```



```

polygon { 4, C, I, G, C}
  polygon { 4, I, G, E, I}
  polygon { 4, G, E, K, G}
  polygon { 4, E, K, B, E}
  polygon { 4, K, B, L, K}
  polygon { 4, B, L, F, B}
  polygon { 4, L, F, H, L}
  polygon { 4, F, H, J, F}
  polygon { 4, H, J, C, H}
  texture {bronze }
}

```

#declare SICO = object

```

{union {
  sphere { A, rayon}
  sphere { B, rayon}
  sphere { C, rayon}
  sphere { D, rayon}
  sphere { E, rayon}
  sphere { F, rayon}
  sphere { G, rayon}
  sphere { H, rayon}
  sphere { I, rayon}
  sphere { J, rayon}
  sphere { K, rayon}
  sphere { L, rayon}
  texture {gold}}}

```

#declare AICO = object

```

{union {
  cylinder { A, I, rayon2}
  cylinder { A, F, rayon2}
  cylinder { A, J, rayon2}
  cylinder { A, B, rayon2}
  cylinder { A, E, rayon2}
  cylinder { B, F, rayon2}
  cylinder { B, E, rayon2}
  cylinder { B, K, rayon2}
  cylinder { B, L, rayon2}
  cylinder { C, G, rayon2}
  cylinder { C, D, rayon2}
  cylinder { C, J, rayon2}
  cylinder { C, I, rayon2}
  cylinder { C, H, rayon2}
  cylinder { D, K, rayon2}
  cylinder { D, L, rayon2}
  cylinder { D, G, rayon2}
  cylinder { D, H, rayon2}
}

```

```

cylinder { E, I, rayon2}
cylinder { E, G, rayon2}
cylinder { E, K, rayon2}
cylinder { F, J, rayon2}
cylinder { F, L, rayon2}
cylinder { F, H, rayon2}
cylinder { G, K, rayon2}
cylinder { G, I, rayon2}
cylinder { H, J, rayon2}
cylinder { I, J, rayon2}
cylinder { K, L, rayon2}
cylinder { L, H, rayon2}
texture {gold}}}

```

#declare AA = <OR/3, (2*OR+1)/3, 0>;

#declare BB = <(OR+1)/3, (OR+1)/3, (-1*OR-1)/3>;

#declare CC = <(2*OR+1)/3, 0, -1*OR/3>;

#declare DD = <(2*OR+1)/3, 0, OR/3>;

#declare EE = <(OR+1)/3, (OR+1)/3, (OR+1)/3>;

#declare FF = <-OR/3, (-2*OR-1)/3, 0>;

#declare GG = <(-1*OR-1)/3, (-1*OR-1)/3, (-1*OR-1)/3>;

#declare HH = <(-1*OR-1)/3, (-1*OR-1)/3, (OR+1)/3>;

#declare II = <(-2*OR-1)/3, 0, OR/3>;

#declare JJ = <(-2*OR-1)/3, 0, -OR/3>;

#declare KK = <-OR/3, (2*OR+1)/3, 0>;

#declare LL = <(-1*OR-1)/3, (OR+1)/3, (OR+1)/3>;

#declare MM = <0, OR/3, (2*OR+1)/3>;

#declare NN = <0, -OR/3, (2*OR+1)/3>;

#declare OO = <(OR+1)/3, (-1*OR-1)/3, (OR+1)/3>;

#declare PP = <OR/3, (-2*OR-1)/3, 0>;

#declare QQ = <(OR+1)/3, (-1*OR-1)/3, (-1*OR-1)/3>;

#declare RR = <0, -OR/3, (-2*OR-1)/3>;

#declare SS = <0, OR/3, (-2*OR-1)/3>;

#declare TT = <(-1*OR-1)/3, (OR+1)/3, (-1*OR-1)/3>;

#declare DODE= object

{ union {

polygon {6, AA, BB, CC, DD, EE, AA}

polygon {6, CC, QQ, PP, OO, DD, CC}

polygon {6, II, LL, KK, TT, JJ, II}

polygon {6, FF, GG, JJ, II, HH, FF}

polygon {6, DD, OO, NN, MM, EE, DD}

polygon {6, BB, SS, RR, QQ, CC, BB}

polygon {6, HH, NN, MM, LL, II, HH}

polygon {6, GG, RR, SS, TT, JJ, GG}

polygon {6, AA, KK, LL, MM, EE, AA}

polygon {6, AA, KK, TT, SS, BB, AA}

polygon {6, FF, PP, OO, NN, HH, FF}

```

    polygon {6, GG, RR, QQ, PP, FF, GG}
    texture { gold1 }
#declare SDODE = object
{union {
    sphere { AA, rayon3}
    sphere { BB, rayon3}
    sphere { CC, rayon3}
    sphere { DD, rayon3}
    sphere { EE, rayon3}
    sphere { FF, rayon3}
    sphere { GG, rayon3}
    sphere { HH, rayon3}
    sphere { II, rayon3}
    sphere { JJ, rayon3}
    sphere { KK, rayon3}
    sphere { LL, rayon3}
    sphere { MM, rayon3}
    sphere { NN, rayon3}
    sphere { OO, rayon3}
    sphere { PP, rayon3}
    sphere { QQ, rayon3}
    sphere { RR, rayon3}
    sphere { SS, rayon3}
    sphere { TT, rayon3}
    texture { bronze1}}}
#declare ADODE = object
{union {
    cylinder { AA, BB, rayon4}
    cylinder { AA, EE, rayon4}
    cylinder { AA, KK, rayon4}
    cylinder { BB, CC, rayon4}
    cylinder { BB, SS, rayon4}
    cylinder { CC, DD, rayon4}
    cylinder { CC, QQ, rayon4}
    cylinder { DD, EE, rayon4}
    cylinder { DD, OO, rayon4}
    cylinder { EE, MM, rayon4}
    cylinder { FF, GG, rayon4}
    cylinder { FF, HH, rayon4}
    cylinder { FF, PP, rayon4}
    cylinder { GG, JJ, rayon4}
    cylinder { GG, RR, rayon4}
    cylinder { HH, II, rayon4}
    cylinder { HH, NN, rayon4}
    cylinder { II, LL, rayon4}
    cylinder { II, JJ, rayon4}

```

```

cylinder { JJ, TT, rayon4}
cylinder { KK, LL, rayon4}
cylinder { KK, TT, rayon4}
cylinder { LL, MM, rayon4}
cylinder { MM, NN, rayon4}
cylinder { NN, OO, rayon4}
cylinder { OO, PP, rayon4}
cylinder { PP, QQ, rayon4}
cylinder { RR, SS, rayon4}
cylinder { RR, QQ, rayon4}
cylinder { SS, TT, rayon4}
    texture {bronze1}}}
#declare AAA = <(7*OR+4)/15, (4*OR+3)/15, 0>;
#declare BBB = <(7*OR+4)/15, (-4*OR-3)/15, 0>;
#declare CCC = <(-7*OR-4)/15, (4*OR+3)/15, 0>;
#declare DDD = <(-7*OR-4)/15, (-4*OR-3)/15, 0>;
#declare EEE = <(4*OR+3)/15, 0, (7*OR+4)/15>;
#declare FFF = <(4*OR+3)/15, 0, (-7*OR-4)/15>;
#declare GGG = <(-4*OR-3)/15, 0, (7*OR+4)/15>;
#declare HHH = <(-4*OR-3)/15, 0, (-7*OR-4)/15>;
#declare III = <0,(7*OR+4)/15, (4*OR+3)/15>;
#declare JJJ = <0,(7*OR+4)/15, (-4*OR-3)/15>;
#declare KKK = <0,(-7*OR-4)/15, (4*OR+3)/15>;
#declare LLL = <0,(-7*OR-4)/15, (-4*OR-3)/15>;

#declare ICO2= object
{union {
    polygon { 4, AAA, III, JJJ, AAA}
    polygon { 4, AAA, FFF, JJJ, AAA}
    polygon { 4, AAA, BBB, FFF, AAA}
    polygon { 4, AAA, BBB, EEE, AAA}
    polygon { 4, AAA, EEE, III, AAA}
    polygon { 4, DDD, KKK, LLL, DDD}
    polygon { 4, DDD, LLL, HHH, DDD}
    polygon { 4, DDD, KKK, GGG, DDD}
    polygon { 4, DDD, GGG, CCC, DDD}
    polygon { 4, DDD, CCC, HHH, DDD}
    polygon { 4, JJJ, CCC, III, JJJ}
    polygon { 4, CCC, III, GGG, CCC}
    polygon { 4, III, GGG, EEE, III}
    polygon { 4, GGG, EEE, KKK, GGG}
    polygon { 4, EEE, KKK, BBB, EEE}
    polygon { 4, KKK, BBB, LLL, KKK}
    polygon { 4, BBB, LLL, FFF, BBB}
    polygon { 4, LLL, FFF, HHH, LLL}
    polygon { 4, FFF, HHH, JJJ, FFF}

```

```

polygon { 4, HHH, JJJ, CCC, HHH}
    texture { bronze }}}
#declare SICO2 = object
{union {
    sphere { AAA, rayon5}
    sphere { BBB, rayon5}
    sphere { CCC, rayon5}
    sphere { DDD, rayon5}
    sphere { EEE, rayon5}
    sphere { FFF, rayon5}
    sphere { GGG, rayon5}
    sphere { HHH, rayon5}
    sphere { III, rayon5}
    sphere { JJJ, rayon5}
    sphere { KKK, rayon5}
    sphere { LLL, rayon5}
    texture {gold}}}}
#declare AICO2 = object
{union {
    cylinder { AAA, III, rayon6}
    cylinder { AAA, FFF, rayon6}
    cylinder { AAA, JJJ, rayon6}
    cylinder { AAA, BBB, rayon6}
    cylinder { AAA, EEE, rayon6}
    cylinder { BBB, FFF, rayon6}
    cylinder { BBB, EEE, rayon6}
    cylinder { BBB, KKK, rayon6}
    cylinder { BBB, LLL, rayon6}
    cylinder { CCC, GGG, rayon6}
    cylinder { CCC, DDD, rayon6}
    cylinder { CCC, JJJ, rayon6}
    cylinder { CCC, III, rayon6}
    cylinder { CCC, HHH, rayon6}
    cylinder { DDD, KKK, rayon6}
    cylinder { DDD, LLL, rayon6}
    cylinder { DDD, GGG, rayon6}
    cylinder { DDD, HHH, rayon6}
    cylinder { EEE, III, rayon6}
    cylinder { EEE, GGG, rayon6}
    cylinder { EEE, KKK, rayon6}
    cylinder { FFF, JJJ, rayon6}
    cylinder { FFF, LLL, rayon6}
    cylinder { FFF, HHH, rayon6}
    cylinder { GGG, KKK, rayon6}
    cylinder { GGG, III, rayon6}
    cylinder { HHH, JJJ, rayon6}

```

```

cylinder { III, JJJ, rayon6}
    cylinder { KKK, LLL, rayon6}
    cylinder { LLL, HHH, rayon6}
    texture {gold}}}}

#if (1<=numérofilm & numérofilm <=5 )
    ICO
    SICO
    AICO
#end

#if ( 2<=numérofilm & numérofilm <=10)
    DODE
    ADODE
    SDODE
#end

#if (7<=numérofilm )
    ICO2
    AICO2
    SICO2
#end

```

4.3 Le code de la dualité entre les tétraèdres

```
#declare numerofilm = 11;           // 11 films.
#declare rapport = 1/3;
#declare rapport2 = 1/9;

global_settings {max_trace_level 25}
background { color rgb <0,0,0> }

#if (numerofilm <=10)
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <-5,5,-5>
        point_at <0, 0, 0>
        radius 60
        tightness 50
        falloff 8}
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <0,5,-5>
        point_at <0, 0, 0>
        radius 60
        tightness 1
        falloff 20}
#else
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <(-5*rapport2+5)*clock-5,(5*rapport2-5)*clock+5,(-5*rapport2+5)*clock-5>
        point_at <0, 0, 0>
        radius 60
        tightness 50
        falloff 8}
    light_source {
        0*x
        color rgb <1,1,1>
        spotlight
        translate <0,(5*rapport2-5)*clock+5,(-5*rapport2+5)*clock-5>
        point_at <0, 0, 0>
        radius 60
        tightness 1
        falloff 20}
#end
```

```
#if (numérofilm =1)
  #declare goldsphere = texture {pigment {color rgbt <0.65, 0.5, 0.2,clock>}
  finish {
    ambient 0.7
    brilliance 15
    diffuse 0.9
    metallic
    specular -0.2*clock + 0.20
    roughness 1/60
    reflection -0.05*clock+0.05
  };

  sphere { <0, 0, 0>, 1 texture {goldsphere}}
#end

#if (numérofilm <=5)
  camera{ location <sqrt(2)/3,0,-3>
    look_at <0.0, 0.0, 0.0>}
#else
  #if (numérofilm =6)
    camera{location <(-2*sqrt(2)/9)*clock+sqrt(2)/3,0,2*clock-3> // film 6
      look_at <0.0, 0.0, 0.0>}
  #else
    #if (numérofilm =11)
      camera{location <(-2*sqrt(2)/27)*clock+sqrt(2)/9,0,2/3*clock-1> // film 11
        look_at <0.0, 0.0, 0.0>}
    #else
      camera{location <(-2*sqrt(2)/9)*1+sqrt(2)/3,0,2*1-3> // film 7
        look_at <0.0, 0.0, 0.0>}
    #end
  #end
#end

#if (numérofilm <=2 | 7<=numérofilm)
  #declare bronze = texture {pigment {color rgbt<0.58, 0.42, 0.20,0.3>}}
#else
  #if (numérofilm =3)
    #declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20,0.7*clock+0.3>}}
  #else
    #declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20,1>}}
  #end
#end

finish {ambient 0.6
  brilliance 3
  diffuse 0.2
  metallic
  roughness 1/90}
```

```
#if (numerofilm =4)
    reflection 0.5-0.5*clock
    specular -0.2*clock+0.2
#else
    #if (numerofilm =5)
        specular 0
        reflection 0
    #else
        specular 0.20
        reflection 0.5
    #end
#end}};

#if (numerofilm < 5 | 7<=numerofilm)
    #declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>}}
#else
    #if (numerofilm =5)
        #declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,clock>}}
    #else
        #declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,1>}}
    #end
#end
    finish { ambient 0.25
        brilliance 3
        diffuse 0.4
        metallic
        roughness 1/60
        #if (numerofilm =5)
            specular -0.2*clock+0.2
            reflection -0.4*clock+0.4
        #else
            specular 0.20
            reflection 0.4
        #end}};

#if (2< numerofilm & numerofilm <=9)
    #declare bronze1 = texture {pigment {color rgbt <0.58, 0.42, 0.20,0>}}
#else
    #if (numerofilm =10)
        #declare bronze1 = texture {pigment {color rgbt <0.58, 0.42, 0.20,clock>}}
    #else
        #if (numerofilm=2)
            #declare bronze1 = texture {pigment {color rgbt <0.58, 0.42, 0.20,1-clock>}}
        #else
            #declare bronze1 = texture {pigment {color rgbt <0.58, 0.42, 0.20,1>}}
        #end
    #end
#end
#end
```

```
finish {
    ambient 0.25
    brilliance 3
    diffuse 0.4
    metallic
    roughness 1/60
    #if (numérofilm =10)
        specular -0.2*clock+0.2
        reflection 0.5-0.5*clock
    #else
        specular 0.20
        reflection 0.5
    #end}};
#if (2< numérofilm & numérofilm <=7)
    #declare gold1 = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>}}
#else
    #if (numérofilm=8 )
        #declare gold1 = texture {pigment {color rgbt <1.00, 0.87, 0.475,clock>}}
    #else
        #if (numérofilm =2)
            #declare gold1 = texture {pigment {color rgbt <1.00, 0.87, 0.475,1-clock>}}
        #else
            #declare gold1 = texture {pigment {color rgbt <1.00, 0.87, 0.475,1>}}
        #end
    #end
#end
finish {
    ambient 0.25
    brilliance 3
    diffuse 0.4
    metallic
    specular 0.20
    roughness 1/60
    #if (numérofilm =2)
        reflection 0.4*clock
        specular 0.2*clock
    #else
        #if (numérofilm =9)
            reflection 0.4-0.4*clock
            specular -0.2*clock+0.2
        #else
            #if (numérofilm=1 | numérofilm =10)
                reflection 0
                specular 0
            #else
                reflection 0.4
                specular 0.2
            #end
        #end
    #end
#end}};
```

```
#declare rayon = 0.04;
#declare rayon2 = 0.02;
#declare rayon3 = 0.04*rapport;
#declare rayon4 = 0.02*rapport;
#declare rayon5 = 0.04*rapport2;
#declare rayon6 = 0.02*rapport2;
#if (numérofilm = 1)
    #declare rayon = 0.04*clock;
    #declare rayon2 = 0.02*clock;
#else
    #if (numérofilm = 2)
        #declare rayon3 = 0.04*rapport*clock;
        #declare rayon4 = 0.02*rapport*clock;
    #else
        #if (numérofilm = 7)
            #declare rayon5 = 0.04*rapport2*clock;
            #declare rayon6 = 0.02*rapport2*clock;
        #end
    #end
#end
#end
#if (numérofilm <= 5)
    plane {y, -1 pigment {color rgbt <1, 0.87, 0.475> }
        rotate <0, 0, 30>}
#else
    #if (numérofilm = 6)
        plane {y, 2/3*clock-1 pigment {color rgbt <1, 0.87, 0.475> }
            rotate <0, 0, 30>}
    #else
        #if (numérofilm = 11)
            plane {y, (2/9)*clock-1/3 pigment {color rgbt <1, 0.87, 0.475> }
                rotate <0, 0, 30>}
        #else
            plane {y, 2/3-1 pigment {color rgbt <1, 0.87, 0.475> }
                rotate <0, 0, 30>}
        #end
    #end
#end
#end

#declare A = <-sqrt(2)/3, -sqrt(6)/3, -1/3>;
#declare B = <2*sqrt(2)/3, 0, -1/3>;
#declare C = <-sqrt(2)/3, sqrt(6)/3, -1/3>;
#declare D = <0,0,1>;

#declare TETRA= object
{union {polygon { 4, A, B, C, A}
    polygon { 4, A, B, D, A}
    polygon { 4, A, C, D, A}
    polygon { 4, B, C, D, B}
    texture {bronze }}}}
```



```
#declare STETRA = object
```

```
{union {
  sphere { A, rayon}
  sphere { B, rayon}
  sphere { C, rayon}
  sphere { D, rayon}
  texture {gold}}}
```

```
#declare ATETRA = object
```

```
{union {
  cylinder { A, B, rayon2}
  cylinder { A, C, rayon2}
  cylinder { A, D, rayon2}
  cylinder { B, C, rayon2}
  cylinder { B, D, rayon2}
  cylinder { C, D, rayon2}
  texture {gold}}}
```

```
#declare AA = <0,0,-1/3>;
```

```
#declare BB = <sqrt(2)/9, -sqrt(6)/9, 1/9>;
```

```
#declare CC = <-2*sqrt(2)/9, 0, 1/9>;
```

```
#declare DD = <sqrt(2)/9, sqrt(6)/9, 1/9>;
```

```
#declare TETRA2= object
```

```
{union {
  polygon { 4, AA, BB, CC, AA}
  polygon { 4, AA, BB, DD, AA}
  polygon { 4, AA, CC, DD, AA}
  polygon { 4, BB, CC, DD, BB}
  texture { gold1 }}}}
```

```
#declare STETRA2 = object
```

```
{union {
  sphere { AA, rayon3}
  sphere { BB, rayon3}
  sphere { CC, rayon3}
  sphere { DD, rayon3}
  texture {bronze1}}}
```

```
#declare ATETRA2 = object
```

```
{union {
  cylinder { AA, BB, rayon4}
  cylinder { AA, CC, rayon4}
  cylinder { AA, DD, rayon4}
  cylinder { BB, CC, rayon4}
  cylinder { BB, DD, rayon4}
  cylinder { CC, DD, rayon4}
  texture {bronze1}}}
```

```
#declare AAA = <-sqrt(2)/27, -sqrt(6)/27, -1/27>;
```

```
#declare BBB = <2*sqrt(2)/27, 0, -1/27>;
```

```
#declare CCC = <-sqrt(2)/27, sqrt(6)/27, -1/27>;
```

```
#declare DDD = <0,0,1/9>;
```

```
#declare TETRA3= object
```

```
{union {
  polygon { 4, AAA, BBB, CCC, AAA}
  polygon { 4, AAA, BBB, DDD, AAA}
  polygon { 4, AAA, CCC, DDD, AAA}
  polygon { 4, BBB, CCC, DDD, BBB}
  texture { bronze }}}}
```

```
#declare STETRA3 = object
```

```
{union {
  sphere { AAA, rayon5}
  sphere { BBB, rayon5}
  sphere { CCC, rayon5}
  sphere { DDD, rayon5}
  texture {gold}}}
```

```
#declare ATETRA3 = object
```

```
{union {
  cylinder { AAA, BBB, rayon6}
  cylinder { AAA, CCC, rayon6}
  cylinder { AAA, DDD, rayon6}
  cylinder { BBB, CCC, rayon6}
  cylinder { BBB, DDD, rayon6}
  cylinder { CCC, DDD, rayon6}
  texture {gold}}}
```

```
#if (1<=numerofilm & numerofilm <=5 )
```

```
  TETRA
```

```
  STETRA
```

```
  ATETRA
```

```
#end
```

```
#if ( 2<=numerofilm & numerofilm <=10)
```

```
  TETRA2
```

```
  ATETRA2
```

```
  STETRA2
```

```
#end
```

```
#if (7<=numerofilm )
```

```
  TETRA3
```

```
  ATETRA3
```

```
  STETRA3
```

```
#end
```

4.4 Le code du développement du tétraèdre.

```
#declare numerofilm = 3;           // Déclaration de la variable numerofilm. (Séquences d'images)
global_settings {max_trace_level 25} // Limite du nombre maximum de réflexions récursives que le code
                                     // calculera.

light_source                       // Source lumineuse.
  {<0, 3, 0>                       // Emplacement de la lumière.
   color rgb <1, 1, 1>}            // Couleur de la lumière : Blanche.

light_source
  {<0, 0, 5>
   color rgb <1, 1, 1>}

light_source
  {<0, 0, -5>
   color rgb <1, 1, 1>}

light_source
  {<10, 10, 0>
   color rgb <1, 1, 1>}

#if (numerofilm = 1)               // Positionnement de la caméra pour la séquence d'images 1.
  camera{location <0, 0, 5>         // Position de la caméra.
    look_at <0.0, 0.0, 0.0>}       // Direction de la caméra.
#else
  #if (numerofilm = 2)             // Positionnement de la caméra pour la séquence d'images 2
    camera{location <0,0,-2.5*clock+5> // La caméra se déplace.
      look_at <0.0, 0.0, 0.0>}
  #else
    camera{ location <0,0,2.5>
      rotate <0.0, -180*clock, 0.0> // La caméra effectue une rotation de 180.°
      look_at <0.0, 0.0, 0.0>}
  #end
#end

background { color rgb <0,0,0> }    // Couleur de fond: Noire
plane {y, -0.03-2*sqrt(2)/3 pigment {color rgbt <0.58, 0.42, 0.20>}} //Emplacement du plan

#declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>}} // Définition de la couleur gold.
finish {                             // Finition.
  ambient 0.30                       // Intensité de la couleur.
  brilliance 3                       // Brillance.
  diffuse 0.4                       // Éclairage.
  Metallic                           // Donne un aspect métallique à la couleur.
  specular 0.2                      // Effet miroir.
  roughness 1/60}};                 // Rugosité du métal.
```

```
#declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20>} // Définition de la couleur bronze.
finish {
    ambient 0.25
    brilliance 3
    diffuse 0.4
    metallic
    specular 0.20
    roughness 1/60}};
#declare rayon2 = 0.02; // Déclaration de la variable rayon2.
#if (numérofilm=1) // Les variables pour la séquence d'images 1.
    #declare A = <-sqrt(2)/3, -sqrt(6)/3, -1/3>; //Déclaration des coordonnées des sommets.
    #declare B = <2*sqrt(2)/3, 0, -1/3>;
    #declare C = <-sqrt(2)/3, sqrt(6)/3, -1/3>;
    #declare D1 = <4*sqrt(2)/3*clock-4*sqrt(2)/3, 0, 4/3*clock-1/3>;
    #declare D2 = <-2*sqrt(2)/3*clock+2*sqrt(2)/3, 2*sqrt(6)/3*clock-2*sqrt(6)/3, 4/3*clock-1/3>;
    #declare D3 = <-2*sqrt(2)/3*clock+2*sqrt(2)/3, -2*sqrt(6)/3*clock+2*sqrt(6)/3, 4/3*clock-1/3>;
#else
    #declare A = <-sqrt(2)/3, -sqrt(6)/3, -1/3>; //Déclaration des coordonnées des sommets.
    #declare B = <2*sqrt(2)/3, 0, -1/3>;
    #declare C = <-sqrt(2)/3, sqrt(6)/3, -1/3>;
    #declare D1 = <0,0,1>;
    #declare D2 = <0,0,1>;
    #declare D3 = <0,0,1>;
#end
#declare DEVL = object //Déclaration de DEVL en tant qu'objet. (Les faces)
{union { // Union des polygones.
    polygon { 4, C, B, A, C}
    polygon { 4, A, C, D1, A}
    polygon { 4, A, B, D2, A}
    polygon { 4, B, C, D3, B}
    texture {gold} //Couleur gold.
    rotate <0, 0, 30>}} //Rotation de 30° autour de l'axe des z.
#declare ATETRA = object //Déclaration de ATETRA en tant qu'objet. (Les arêtes)
{union { // Union des cylindres suivants:
    cylinder { A,B, rayon2}
    cylinder { B,C, rayon2}
    cylinder { A,C, rayon2}
    cylinder { B,D3, rayon2}
    cylinder { D3,C, rayon2}
    cylinder { C,D1, rayon2}
    cylinder { A,D1, rayon2}
    cylinder { A,D2, rayon2}
    cylinder { D2,B, rayon2}
    texture {bronze} //Couleur bronze
    rotate <0, 0, 30>}} // Rotation de 30° autour de l'axe des z.
```

```
#if (numérofilm = 1)                                //Les rayons pour la séquence d'images 1.
  #declare rayon = 0;                                //La valeur du rayon est 0.
#else
  #if (numérofilm = 2)                                //Les rayons pour la séquence d'images 2.
    #declare rayon = 0.03*clock;                      //La valeur du rayon augmente de manière constante de 0 à 0,03
  #else
    #declare rayon = 0.03;                            //La valeur du rayon est 0,03.
  #end
#end

#declare STETRA = object                             //Déclaration de STETRA en tant qu'objet. (Les sommets)
{union {                                              //Union des sphères suivantes:
  sphere { A, rayon}
  sphere { B, rayon}
  sphere { C, rayon}
  sphere { D1, rayon}
  texture {bronze}                                    //Couleur bronze.
  rotate <0, 0, 30>}}                                // Rotation de 30° autour de l'axe des z.

DEVL                                                  //Trace DEVL.
ATETRA                                                //Trace ATETRA.
STETRA                                                //Trace STETRA.
```

4.5 Le code du développement du cube.

```
#declare numerofilm = 4;                // 4 films.
global_settings {max_trace_level 25}
light_source
  {<0, 2, 0>
   color rgb <1, 1, 1>}
light_source
  {<0, 0, -5>
   color rgb <1, 1, 1>}
light_source
  {<0, 0, 5>
   color rgb <1, 1, 1>}
light_source
  {<3, 2, -2>
   color rgb <1, 1, 1>}
#if (numerofilm <=2 )
  camera{location <0, 0, -5>
    look_at <0.0, 0.0, 0.0>}
#else
  #if(numerofilm=3)
    camera {location <2*clock,2*clock,3*clock-5>
      look_at <0.0, 0.0, 0.0>}
  #else
    camera{location <2,2,-2>
      rotate <0.0, -180*clock, 0.0>
      look_at <0.0, 0.0, 0.0>}
  #end
#end
background { color rgb <0,0,0>}
plane {y, -sqrt(3)-0.03 pigment {color rgbt <0.58, 0.42, 0.20>}}

#declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>}
finish {
  ambient 0.30
  brilliance 3
  diffuse 0.4
  metallic
  specular 0.2
  roughness 1/60}};

#declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20>}
finish {
  ambient 0.25
  brilliance 3
  diffuse 0.4
  metallic
  specular 0.20
  roughness 1/60}};
```

```
#declare rayon2 = 0.02;

#declare Coin=(sqrt(3)/3);

#if (numérofilm = 1)
  #declare rayon = 0;
#else
  #if (numérofilm = 2)
    #declare rayon = 0.03*clock;
  #else
    #declare rayon = 0.03;
  #end
#end

#if (numérofilm=1)
  #declare A = <-Coin, Coin, Coin>;
  #declare B = <Coin, Coin, Coin>;
  #declare C1 = <-2*Coin*clock+3*Coin, Coin, -2*Coin*clock+Coin>;
  #declare C2 = <Coin, -2*Coin*clock+3*Coin, -2*Coin*clock+Coin>;
  #declare D1 = <2*Coin*clock-3*Coin, Coin, -2*Coin*clock+Coin>;
  #declare D2 = <-6*Coin*clock+5*Coin, Coin, -2*Coin*clock+Coin>;
  #declare D3 = <-Coin, -2*Coin*clock+3*Coin, -2*Coin*clock+Coin>;
  #declare E1 = <2*Coin*clock-3*Coin, -Coin, -2*Coin*clock+Coin>;
  #declare E2 = <-6*Coin*clock+5*Coin, -Coin, -2*Coin*clock+Coin>;
  #declare E3 = <-Coin, 2*Coin*clock-3*Coin, -2*Coin*clock+Coin>;
  #declare F1 = <-2*Coin*clock+3*Coin, -Coin, -2*Coin*clock+Coin>;
  #declare F2 = <Coin, 2*Coin*clock-3*Coin, -2*Coin*clock+Coin>;
  #declare G = <Coin, -Coin, Coin>;
  #declare H = <-Coin, -Coin, Coin>;
#else
  #declare A = <-Coin, Coin, Coin>;
  #declare B = <Coin, Coin, Coin>;
  #declare C1 = <Coin, Coin, -Coin>;
  #declare C2 = <Coin, Coin, -Coin>;
  #declare D1 = <-Coin, Coin, -Coin>;
  #declare D2 = <-Coin, Coin, -Coin>;
  #declare D3 = <-Coin, Coin, -Coin>;
  #declare E1 = <-Coin, -Coin, -Coin>;
  #declare E2 = <-Coin, -Coin, -Coin>;
  #declare E3 = <-Coin, -Coin, -Coin>;
  #declare F1 = <Coin, -Coin, -Coin>;
  #declare F2 = <Coin, -Coin, -Coin>;
  #declare G = <Coin, -Coin, Coin>;
  #declare H = <-Coin, -Coin, Coin>;
#end
```

```
#declare DEVL = object
{union {
    polygon { 5, D1, A, H, E1, D1}
    polygon { 5, A, B, G, H, A}
    polygon { 5, B, C1, F1, F1, G}
    polygon { 5, C1, D2, E2, F1, C1}
    polygon { 5, D3,C2, B, A, D3}
    polygon { 5, H, G, F2, E3, H}
    texture {gold}}}
```

```
#declare ACUBE = object
{union {
    cylinder { D1,A, rayon2}
    cylinder { A, B, rayon2}
    cylinder { B, C1, rayon2}
    cylinder { C1,D2, rayon2}
    cylinder { D2,E2, rayon2}
    cylinder { E2,F1, rayon2}
    cylinder { F1,G, rayon2}
    cylinder { G, H, rayon2}
    cylinder { H, E1, rayon2}
    cylinder { D1,E1, rayon2}
    cylinder { A,D3, rayon2}
    cylinder { D3,C2, rayon2}
    cylinder { C2,B, rayon2}
    cylinder { H,E3, rayon2}
    cylinder { E3,F2, rayon2}
    cylinder { F2,G, rayon2}
    cylinder { A,H, rayon2}
    cylinder { B,G, rayon2}
    cylinder { C1,F1, rayon2}
    cylinder { F2,G, rayon2}
    texture {bronze}}}
```

```
#declare SCUBE = object
{union {
    sphere { A, rayon}
    sphere { B, rayon}
    sphere { C1, rayon}
    sphere { D1, rayon}
    sphere { E1, rayon}
    sphere { F1, rayon}
    sphere { G, rayon}
    sphere { H, rayon}
    texture {bronze}}}
```

```
DEVL
ACUBE
SCUBE
```

4.6 Le code du développement de l'octaèdre

```
#declare numerofilm = 3;                // 3 films.
global_settings {max_trace_level 25}
light_source
  {<0, 2, 0>
   color rgb <1, 1, 1>}
light_source
  {<0, 0, -5>
   color rgb <1, 1, 1>}
light_source
  {<0, 0, 5>
   color rgb <1, 1, 1>}
light_source
  {<3, 2, -2>
   color rgb <1, 1, 1>}
#if(numerofilm=1)
  camera{location <0.5*clock,clock,5*clock-8>
    look_at <0,0, 0>}
#else
  #if(numerofilm=2)
    camera{location <0.5,1,-3>
      look_at <0,0, 0>}
  #else
    camera{location <0.5,1,-3>
      rotate <0.0, -180*clock, 0.0>
      look_at <0,0, 0>}
  #end
#endif
background { color rgb <0,0,0> }
plane {y, -sqrt(3)-0.03 pigment {color rgbt <0.58, 0.42, 0.20>}}

#declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>}}
finish {
  ambient 0.30
  brilliance 3
  diffuse 0.4
  metallic
  specular 0.2
  roughness 1/60}};
#declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20>}}
finish {
  ambient 0.25
  brilliance 3
  diffuse 0.4
  metallic
  specular 0.20
  roughness 1/60}};
```



```
#declare rayon2 = 0.02;
#declare hauteur = sqrt(3);
#iif(numerofilm=1)
    #declare A1 = <3*clock-3, (1-hauteur)*clock+hauteur, 0>;
    #declare A2 = <-3*clock+3, (1-hauteur)*clock+hauteur, 0>;
    #declare A3 = <-3*clock+3, (1+hauteur)*clock-hauteur,0>;
    #declare B = <-clock+1,-hauteur*clock+hauteur,clock>;
    #declare C = <0,-clock,0>;
    #declare D1 = <2*clock-2,0,-clock>;
    #declare D2 = <-clock+1, hauteur*clock-hauteur, -clock>;
    #declare E = <-clock+2,0,0>;
    #declare F1 = <-1, -hauteur*clock+hauteur,0>;
    #declare F2 = <-3*clock+2, -2*hauteur*clock+2*hauteur,0>;
#else
    #declare A1 = <0, 1, 0>;
    #declare A2 = <0, 1, 0>;
    #declare A3 = <0, 1, 0>;
    #declare B = <0, 0, 1>;
    #declare C = <0, -1, 0>;
    #declare D1 = <0, 0, -1>;
    #declare D2 = <0, 0, -1>;
    #declare E = <1, 0, 0>;
    #declare F1 = <-1, 0, 0>;
    #declare F2 = <-1, 0, 0>;
#end
#declare DEVL = object
{union {
    polygon { 4, A3,D2, E, A3}
    polygon { 4, A2, B, E, A2}
    polygon { 4, A2, F2, B, A2}
    polygon { 4, F1, A1, D1, F1}
    polygon { 4, C, F1, D1, C}
    polygon { 4, E, C, D2, E}
    polygon { 4, E, B, C, E}
    polygon { 4, B, F1, C, B}
    texture {gold}}}
#declare AOCTA = object
{union {
    cylinder { A1,F1, rayon2}
    cylinder { F1,B, rayon2}
    cylinder { B,A2, rayon2}
    cylinder { B,F2, rayon2}
    cylinder { F2,A2, rayon2}
    cylinder { A1,D1, rayon2}
    cylinder { D1,C, rayon2}
    cylinder { C,E, rayon2}
    cylinder { E,A2, rayon2}
    cylinder { D1,F1, rayon2}
```

```
cylinder { C,B, rayon2}
cylinder { F1,C, rayon2}
cylinder { B,E, rayon2}
cylinder { C,D2, rayon2}
cylinder { D2,E, rayon2}
cylinder { E,A3, rayon2}
cylinder { D2,A3, rayon2}
texture {bronze}}}}

#if (numérofilm = 1)
  #declare rayon = 0;
#else
  #if (numérofilm = 2)
    #declare rayon = 0.04*clock;
  #else
    #declare rayon = 0.04;
  #end
#end

#declare SOCTA = object
{union {
  sphere { A1, rayon}
  sphere { B, rayon}
  sphere { C, rayon}
  sphere { D1, rayon}
  sphere { E, rayon}
  sphere { F1, rayon}
  texture {bronze}}}}

DEVL
AOCTA
SOCTA
```

4.7 Le code du développement du dodécaèdre

```
#declare numerofilm = 1;                // 3 films.

global_settings {max_trace_level 25}
#include "math.inc"

light_source
  {<0,-2, -10>
   color rgb <1, 1, 1>}
light_source
  {<0,-2, 10>
   color rgb <1, 1, 1>}
light_source
  {<0,10, 0>
   color rgb <1, 1, 1>}

#declare rayon2 = 0.06;
#declare OR = (sqrt(5)+1)/2;
#declare cote = 2*OR;
#declare largeur = 2*cote*cosd(36);
#declare hauteur = cote*sind(72);
#declare haut = cote*sind(36);

#if (numerofilm = 1)
  light_source
    {<0, ((-hauteur-haut)/2))*clock+((-hauteur-haut)/2), -25>
    color rgb <1, 1, 1>}
#else
  light_source
    {<0, ((-hauteur-haut)/2))+((-hauteur-haut)/2), -25>
    color rgb <1, 1, 1>}
#end

#if (numerofilm =1)
  camera{ location <0,(-hauteur-haut)/2,8*clock-22>
    look_at <0.0, ((hauteur+haut)/2)*clock+(-hauteur-haut)/2 , 0.0>}
#else
  #if (numerofilm = 3)
    camera{ location <0,(-hauteur-haut)/2,-14>
      rotate <0,180*clock,0>
      look_at <0.0, ((hauteur+haut)/2)+(-hauteur-haut)/2 , 0.0>}
  #else
    camera{ location <0,(-hauteur-haut)/2,-14>
      look_at <0.0, ((hauteur+haut)/2)+(-hauteur-haut)/2 , 0.0>}
  #end
#end

background { color rgb <0,0,0> }
```

```
plane {y, -3*hauteur-2*haut-0.05 pigment {color rgbt <0.58, 0.42, 0.20>}}
```

```
#declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>}
```

```
finish {
```

```
    ambient 0.30
```

```
    brilliance 3
```

```
    diffuse 0.4
```

```
    metallic
```

```
    specular 0.2
```

```
    roughness 1/60}};
```

```
#declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20>}
```

```
finish {
```

```
    ambient 0.25
```

```
    brilliance 3
```

```
    diffuse 0.4
```

```
    metallic
```

```
    specular 0.20
```

```
    roughness 1/60}};
```

```
#if (numerofilms = 1)
```

```
    #declare A = <((OR+1)-largeur/2)*clock+largeur/2, (OR+1+hauteur)*clock-hauteur,(OR+1-OR)*clock+OR>;
```

```
    #declare B = <(OR-cote/2)*clock+cote/2,(2*OR+1)*clock,-OR*clock+OR>;
```

```
    #declare C = <(-OR+cote/2)*clock-cote/2, (2*OR+1)*clock, -OR*clock+OR>;
```

```
    #declare D = <(-OR-1+largeur/2)*clock-largeur/2, (OR+1+hauteur)*clock-hauteur,(OR+1-OR)*clock+OR>;
```

```
    #declare E = <0,(OR+haut+hauteur)*clock-haut-hauteur,(2*OR+1-OR)*clock+OR>;
```

```
    #declare F1 = <(-OR-1+largeur/2)*clock-largeur/2, (OR+1-hauteur)*clock+hauteur,(-OR-1-OR)*clock+OR>;
```

```
    #declare F2 = <(-OR-1+cote/2+largeur/2)*clock-cote/2-largeur/2, (OR+1-haut)*clock+haut, (-OR-1-OR)*clock+OR>;
```

```
    #declare G1 = <0,(OR-haut-hauteur)*clock+haut+hauteur,(-2*OR-1-OR)*clock+OR>;
```

```
    #declare G2 = <-largeur*clock+largeur, (OR-haut-hauteur)*clock+haut+hauteur, (-2*OR-1-OR)*clock+OR>;
```

```
    #declare G3 = <largeur*clock-largeur, (OR-haut-hauteur)*clock+haut+hauteur, (-2*OR-1-OR)*clock+OR>;
```

```
    #declare H1 = <(OR+1-largeur/2)*clock+largeur/2, (OR+1-hauteur)*clock+hauteur,(-OR-1-OR)*clock+OR>;
```

```
    #declare H2 = <(OR+1-cote/2-largeur/2)*clock+cote/2+largeur/2, (OR+1-haut)*clock+haut, (-OR-1-OR)*clock+OR>;
```

```
    #declare I1 = <(2*OR+1-cote/2-largeur)*clock+cote/2+largeur, 0, (-OR-OR)*clock+OR>;
```

```
    #declare I2 = <(2*OR+1-largeur-cote)*clock+largeur+cote,(hauteur+haut)*clock-hauteur-haut, (-OR-OR)*clock+OR>;
```

```
    #declare J1 = <(2*OR+1-largeur/2-cote)*clock+largeur/2+cote, hauteur*clock-hauteur, OR>;
```

```
    #declare J2 = <(2*OR+1-largeur)*clock+largeur,(hauteur+haut)*clock-hauteur-haut, OR>;
```

```
    #declare L1 = <(-2*OR-1+cote/2+largeur)*clock-cote/2-largeur, 0, (-OR-OR)*clock+OR>;
```

```
    #declare M1 = <(-2*OR-1+largeur/2+cote)*clock-largeur/2-cote, hauteur*clock-hauteur, OR>;
```

```
    #declare M2 = <(-2*OR-1+largeur)*clock-largeur,(hauteur+haut)*clock-hauteur-haut, OR>;
```

```
    #declare N1 = <(-OR-1+cote/2+largeur/2)*clock-cote/2-largeur/2, (-OR-1+2*hauteur+haut)*clock-2*hauteur-haut, (OR+1-OR)*clock+OR>;
```

```
    #declare N2 = <(-OR-1+largeur+cote)*clock-largeur-cote, (-OR-1+hauteur+haut)*clock-hauteur-haut, (OR+1-OR)*clock+OR>;
```

```
    #declare O1 = <(-cote/2+largeur/2)*clock+cote/2-largeur/2, (-OR+2*hauteur+haut)*clock-2*hauteur-haut, (2*OR+1-OR)*clock+OR>;
```

```
    #declare O2 = <(cote/2-largeur/2)*clock-cote/2+largeur/2, (-OR+2*hauteur+haut)*clock-2*hauteur-haut, (2*OR+1-OR)*clock+OR>;
```

```
    #declare P1 = <(OR+1-cote/2-largeur/2)*clock+cote/2+largeur/2, (-OR-1+2*hauteur+haut)*clock-2*hauteur-haut, (OR+1-OR)*clock+OR>;
```

```
    #declare P2 = <(OR+1)*clock,(-OR-1+3*hauteur+haut)*clock-3*hauteur-haut, (OR+1-OR)*clock+OR>;
```

```
    #declare Q1 = <(-OR-1+(3*largeur/2)+cote/2)*clock-(3*largeur/2)-cote/2, (-OR-1-haut)*clock+haut, (-OR-1-OR)*clock+OR>;
```

```
    #declare Q2 = <(-OR-1+(3*cote/2)+largeur)*clock-(3*cote/2)-largeur, (-OR-1)*clock, (-OR-1-OR)*clock+OR>;
```

```
    #declare Q3 = <(-OR-1-(3*largeur/2+cote))*clock+(3*largeur/2)+cote, (-OR-1+3*hauteur+2*haut)*clock-3*hauteur-2*haut, (-OR-1-OR)*clock+OR>;
```

```
    #declare R1 = <(-OR-(3*largeur/2))*clock+3*largeur/2, (-2*OR-1+3*hauteur+2*haut)*clock-3*hauteur-2*haut,(-OR)*clock+OR>;
```

```

#declare R2 = <(-OR+(3*largeur/2)+cote)*clock-(3*largeur/2)-cote, (-2*OR-1+hauteur)*clock-hauteur, (-OR)*clock+OR>;
#declare R3 = <(-OR+largeur)*clock-largeur,(-2*OR-1+3*hauteur+haut)*clock-3*hauteur-haut, (-OR)*clock+OR>;
#declare S1 = <(OR+largeur/2)*clock-largeur/2, (-2*OR-1+3*hauteur+2*haut)*clock-3*hauteur-2*haut, (-OR)*clock+OR>;
#declare S2 = <(OR-cote/2-largeur)*clock+cote/2+largeur, (-2*OR-1+2*hauteur+2*haut)*clock-2*hauteur-2*haut, (-OR)*clock+OR>;
#declare T1 = <(OR+1-cote/2-3*largeur/2)*clock+cote/2+3*largeur/2, (-OR-1+2*hauteur+haut)*clock-2*hauteur-haut, (-OR-1-OR)*clock+OR>;
#declare T2 = <(OR+1-cote/2-3*largeur/2)*clock+cote/2+3*largeur/2, (-OR-1-haut)*clock+haut, (-OR-1-OR)*clock+OR>;
#declare U1 = <(largeur+cote)*clock-largeur-cote, (-OR-hauteur-haut)*clock+hauteur+haut, (-2*OR-1-OR)*clock+OR>;
#declare U2 = <(-largeur-cote)*clock+largeur+cote, (-OR-haut-hauteur)*clock+haut+hauteur, (-2*OR-1-OR)*clock+OR>;
#declare U3 = <(-cote/2-2*largeur)*clock+(cote/2)+2*largeur, (-OR+2*hauteur+2*haut)*clock-2*hauteur-2*haut, (-2*OR-1-OR)*clock+OR>;
#else
#declare A = <OR+1, OR+1, OR+1>;
#declare B = <OR, 2*OR+1, 0>;
#declare C = <-OR, 2*OR+1, 0>;
#declare D = <-OR-1, OR+1, OR+1>;
#declare E = <0, OR, 2*OR+1>;
#declare F1 = <-OR-1, OR+1, -OR-1>;
#declare F2 = <-OR-1, OR+1, -OR-1>;
#declare G1 = <0, OR, -2*OR-1>;
#declare G2 = <0, OR, -2*OR-1>;
#declare G3 = <0, OR, -2*OR-1>;
#declare H1 = <OR+1, OR+1, -OR-1>;
#declare H2 = <OR+1, OR+1, -OR-1>;
#declare I1 = <2*OR+1, 0, -OR>;
#declare I2 = <2*OR+1, 0, -OR>;
#declare J1 = <2*OR+1, 0, OR>;
#declare J2 = <2*OR+1, 0, OR>;
#declare L1 = <-2*OR-1, 0, -OR>;
#declare L2 = <-2*OR-1, 0, -OR>;
#declare M1 = <-2*OR-1, 0, OR>;
#declare M2 = <-2*OR-1, 0, OR>;
#declare N1 = <-OR-1, -OR-1, OR+1>;
#declare N2 = <-OR-1, -OR-1, OR+1>;
#declare O1 = <0, -OR, 2*OR+1>;
#declare O2 = <0, -OR, 2*OR+1>;
#declare P1 = <OR+1, -OR-1, OR+1>;
#declare P2 = <OR+1, -OR-1, OR+1>;
#declare Q1 = <-OR-1, -OR-1, -OR-1>;
#declare Q2 = <-OR-1, -OR-1, -OR-1>;
#declare Q3 = <-OR-1, -OR-1, -OR-1>;
#declare R1 = <-OR, -2*OR-1, 0>;
#declare R2 = <-OR, -2*OR-1, 0>;
#declare R3 = <-OR, -2*OR-1, 0>;
#declare S1 = <OR, -2*OR-1, 0>;
#declare S2 = <OR, -2*OR-1, 0>;
#declare T1 = <OR+1, -OR-1, -OR-1>;
#declare T2 = <OR+1, -OR-1, -OR-1>;
#declare U1 = <0, -OR, -2*OR-1>;
#declare U2 = <0, -OR, -2*OR-1>;
#declare U3 = <0, -OR, -2*OR-1>;
#end

```

```
#declare DEVL = object
{union {
  polygon { 6, A, B, C, D, E, A}
  polygon { 6, H1, G1, F1, C, B, H1}
  polygon { 6, B, H2, I1, J1, A, B}
  polygon { 6, C, F2, L1, M1, D, C}
  polygon { 6, D, E, O1, N1, M2, D}
  polygon { 6, E, A, J2, P1, O2, E}
  polygon { 6, L1, Q1, U1, G3, F2, L1}
  polygon { 6, S2, T1, U3, Q3, R1, S2}
  polygon { 6, L1, Q2, R2, N2, M1, L1}
  polygon { 6, N1, R3, S1, P2, O1, N1}
  polygon { 6, P1, J2, I2, T1, S2, P1}
  polygon { 6, H2, G2, U2, T2, I1, H2}
  texture {gold}}}
#declare ADODE = object
{union {
  cylinder { S2, R1, rayon2}
  cylinder { R1,Q3, rayon2}
  cylinder { Q3,U3 rayon2}
  cylinder { U3,T1, rayon2}
  cylinder { U1,Q1, rayon2}
  cylinder { U1,G3 rayon2}
  cylinder { G1,F1, rayon2}
  cylinder { L1, Q1, rayon2}
  cylinder { F2, G3, rayon2}
  cylinder { G1,H1, rayon2}
  cylinder { H1,B, rayon2}
  cylinder { B,C, rayon2}
  cylinder { C,F1, rayon2}
  cylinder { M1,N2, rayon2}
  cylinder { N2,R2, rayon2}
  cylinder { R2,Q2, rayon2}
  cylinder { Q2,L1, rayon2}
  cylinder { L1,M1, rayon2}
  cylinder { L1,F2, rayon2}
  cylinder { F2,C, rayon2}
  cylinder { C,D, rayon2}
  cylinder { D,M1, rayon2}
  cylinder { H2,G2, rayon2}
  cylinder { G2,U2, rayon2}
  cylinder { U2,T2, rayon2}
  cylinder { T2,I1, rayon2}
  cylinder { I1,H2, rayon2}
  cylinder { H2,B, rayon2}
  cylinder { B,A, rayon2}
  cylinder { A,J1, rayon2}
  cylinder { J1,I1, rayon2}
  cylinder { J2,I2, rayon2}
  cylinder { I2,T1, rayon2}
```

```
cylinder { T1,S2, rayon2}
cylinder { S2,P1, rayon2}
cylinder { P1,J2, rayon2}
cylinder { J2,A, rayon2}
cylinder { A,E, rayon2}
cylinder { E,O2, rayon2}
cylinder { O2,P1, rayon2}
cylinder { O1,P2, rayon2}
cylinder { P2,S1, rayon2}
cylinder { S1,R3, rayon2}
cylinder { R3,N1, rayon2}
cylinder { N1,O1, rayon2}
cylinder { O1,E, rayon2}
cylinder { E,D, rayon2}
cylinder { D,M2, rayon2}
cylinder { M2,N1, rayon2}
  texture {bronze}}}
#if (numerofilm =1)
  #declare rayon = 0;
#else
  #if (numerofilm=2)
    #declare rayon = 0.12*clock;
  #else
    #declare rayon = 0.12;
  #end
#end
#declare SDODE = object
{union {
  sphere { A, rayon}
  sphere { B, rayon}
  sphere { C, rayon}
  sphere { D, rayon}
  sphere { E, rayon}
  sphere { F1, rayon}
  sphere { G1, rayon}
  sphere { H1, rayon}
  sphere { I1, rayon}
  sphere { J1, rayon}
  sphere { L1, rayon}
  sphere { M1, rayon}
  sphere { N1, rayon}
  sphere { O1, rayon}
  sphere { P1, rayon}
  sphere { Q1, rayon}
  sphere { R1, rayon}
  sphere { S1, rayon}
  sphere { T1, rayon}
  sphere { U1, rayon}
  texture { bronze}}}
DEVL
ADODE
SDODE
```

4.8 Le code du développement de l'icosaèdre

```
#declare numerofilm = 3;           // 3 films.
global_settings {max_trace_level 25}

light_source
  {<0, 0, 5>
   color rgb <1, 1, 1>}
light_source
  {<0, 0, -5>
   color rgb <1, 1, 1>}
light_source
  {<0, 5, 0>
   color rgb <1, 1, 1>}
light_source
  {<5, 4, 5>
   color rgb <1, 1, 1>}
#if (numerofilm =1)
  camera{location <0,0,8*clock-13>
    look_at <0.0, 0.0, 0.0>}
#else
  #if(numerofilm=2)
    camera{ location <0,0,-5>
      look_at <0.0, 0.0, 0.0>}
  #else
    camera{ location <0,0,-5>
      rotate <0.0, -180*clock, 0.0>
      look_at <0.0, 0.0, 0.0>}
  #end
#end
background { color rgb <0,0,0> }
plane {y, -5.2 pigment {color rgbt <0.58, 0.42, 0.20>}}
#declare gold = texture {pigment {color rgbt <1.00, 0.87, 0.475,0>}}
finish {
  ambient 0.30
  brilliance 3
  diffuse 0.4
  metallic
  specular 0.2
  roughness 1/60}};

#declare bronze = texture {pigment {color rgbt <0.58, 0.42, 0.20>}}
finish {
  ambient 0.25
  brilliance 3
  diffuse 0.4
  metallic
  specular 0.20
  roughness 1/60}};
```

```
#declare rayon2 = 0.02;
#declare OR = ((sqrt(5)+1)/2);
#if(numerofilm =1)
    #declare A = <(OR-2)*clock+2,(1-sqrt(3))*clock+sqrt(3), -OR*clock+OR>;
    #declare B1 = <(OR-3)*clock+3,-clock+0,-OR*clock+OR>;
    #declare B2 = <(OR-2)*clock+2, (-1+sqrt(3))*clock-sqrt(3), -OR*clock+OR>;
    #declare C = <(-OR+2)*clock-2,(1-sqrt(3))*clock+sqrt(3),-OR*clock+OR>;
    #declare D1 = <(-OR+3)*clock-3,-clock+0,-OR*clock+OR>;
    #declare D2 = <(-OR+2)*clock-2, (-1+sqrt(3))*clock-sqrt(3),-OR*clock+OR>;
    #declare E = <1,0,OR>;
    #declare F1 = <-2*clock+3,-2*sqrt(3)*clock+2*sqrt(3), -2*OR*clock+OR>;
    #declare F2 = <-3*clock+4,-sqrt(3)*clock+sqrt(3),-2*OR*clock+OR>;
    #declare F3 = <-2*clock+3,2*sqrt(3)*clock-2*sqrt(3), -2*OR*clock+OR>;
    #declare F4 = <3*clock-2, 3*sqrt(3)*clock-3*sqrt(3), -2*OR*clock+OR>;
    #declare G = <-1,0,OR>;
    #declare H1 = <3*clock-4,-sqrt(3)*clock+sqrt(3),-2*OR*clock+OR>;
    #declare H2 = <2*clock-3,-2*sqrt(3)*clock+2*sqrt(3), -2*OR*clock+OR>;
    #declare H3 = <-3*clock+2,-3*sqrt(3)*clock+3*sqrt(3), -2*OR*clock+OR>;
    #declare H4 = <2*clock-3, 2*sqrt(3)*clock-2*sqrt(3), -2*OR*clock+OR>;
    #declare I = <0,(OR-sqrt(3))*clock+sqrt(3),(1-OR)*clock+OR>;
    #declare J1 = <clock-1, (OR-2*sqrt(3))*clock+2*sqrt(3), (-1-OR)*clock+OR>;
    #declare J2 = <-clock+1, (OR-2*sqrt(3))*clock+2*sqrt(3), (-1-OR)*clock+OR>;
    #declare K = <0,(-OR+sqrt(3))*clock-sqrt(3), (1-OR)*clock+OR>;
    #declare L1 = <clock-1, (-OR+2*sqrt(3))*clock-2*sqrt(3), (-1-OR)*clock+OR>;
    #declare L2 = <-clock+1, (-OR+2*sqrt(3))*clock-2*sqrt(3), (-1-OR)*clock+OR>;
#else
    #declare A = <OR,1,0>;
    #declare B1 = <OR,-1,0>;
    #declare B2 = <OR,-1,0>;
    #declare C = <-OR,1,0>;
    #declare D1 = <-OR,-1,0>;
    #declare D2 = <-OR,-1,0>;
    #declare E = <1,0,OR>;
    #declare F1 = <1,0,-OR>;
    #declare F2 = <1,0,-OR>;
    #declare F3 = <1,0,-OR>;
    #declare F4 = <1,0,-OR>;
    #declare G = <-1,0,OR>;
    #declare H1 = <-1,0,-OR>;
    #declare H2 = <-1,0,-OR>;
    #declare H3 = <-1,0,-OR>;
    #declare H4 = <-1,0,-OR>;
    #declare I = <0, OR, 1>;
    #declare J1 = <0, OR, -1>;
    #declare J2 = <0, OR, -1>;
    #declare K = <0, -OR, 1>;
    #declare L1 = <0, -OR, -1>;
    #declare L2 = <0, -OR, -1>;
#end
```



```
#declare DEVL = object
{union {
  polygon { 4, J2, H3, F1, J2}
  polygon { 4, C,H2, J1, C}
  polygon { 4, I, C, J1, I}
  polygon { 4, I, J2, A, I}
  polygon { 4, A, J2, F1, A}
  polygon { 4, D1, H1,C,D1}
  polygon { 4, D1, C, G, D1}
  polygon { 4, G, C, I, G}
  polygon { 4, G, I, E, G}
  polygon { 4, E, I, A, E }
  polygon { 4, E, A, B1,E}
  polygon { 4, B1, A, F2, B1}
  polygon { 4, D2, G, K, D2}
  polygon { 4, K, G, E, K}
  polygon { 4, K, E, B2, K}
  polygon { 4, H4, D2, L1, H4}
  polygon { 4, L1, D2, K, L1}
  polygon { 4, L2, K, B2, L2}
  polygon { 4, L2, B2, F3,L2}
  polygon { 4, F4, H4, L1, F4}
  texture {gold}}}
```

```
#declare AICO = object
{union {
  cylinder { H2,J1, rayon2}
  cylinder { J2, F1, rayon2}
  cylinder { H1,C, rayon2}
  cylinder { C,I, rayon2}
  cylinder { I,A, rayon2}
  cylinder { A,F2, rayon2}
  cylinder { D1,G, rayon2}
  cylinder { G, E, rayon2}
  cylinder { E,B1, rayon2}
  cylinder { D2,K, rayon2}
  cylinder { K,B2, rayon2}
  cylinder { H4,L1, rayon2}
  cylinder { L2,F3, rayon2}
  cylinder { D1,C, rayon2}
  cylinder { C,J1, rayon2}
  cylinder { H4,D2, rayon2}
  cylinder { D2,G, rayon2}
  cylinder { G,I, rayon2}
  cylinder { I,J2, rayon2}
  cylinder { J2,H3, rayon2}
  cylinder { F4,L1, rayon2}
  cylinder { L1,K, rayon2}
  cylinder { K,E, rayon2}
```

```
cylinder { E,A, rayon2}
  cylinder { A,F1, rayon2}
  cylinder { L2,B2, rayon2}
  cylinder { B1,F2, rayon2}
  cylinder { H4,F4, rayon2}
  cylinder { L1,D2, rayon2}
  cylinder { D1,H1, rayon2}
  cylinder { L2,K, rayon2}
  cylinder { K,G, rayon2}
  cylinder { G,C, rayon2}
  cylinder { C,H2, rayon2}
  cylinder { F3,B2, rayon2}
  cylinder { B2,E, rayon2}
  cylinder { E,I, rayon2}
  cylinder { I,J1, rayon2}
  cylinder { B1,A, rayon2}
  cylinder { A, J2, rayon2}
  cylinder { F1,H3, rayon2}
  texture {bronze}}}
```

```
#if (numerofilm = 1)
  #declare rayon = 0;
#else
  #if (numerofilm = 2)
    #declare rayon = 0.04*clock;
  #else
    #declare rayon = 0.04;
  #end
#end
```

```
#declare SICO = object
{union {
  sphere { A, rayon}
  sphere { B1, rayon}
  sphere { C, rayon}
  sphere { D1, rayon}
  sphere { E, rayon}
  sphere { F1, rayon}
  sphere { G, rayon}
  sphere { H1, rayon}
  sphere { I, rayon}
  sphere { J1, rayon}
  sphere { K, rayon}
  sphere { L1, rayon}
  texture {bronze}}}
```

```
DEVL
AICO
SICO
```

BIBLIOGRAPHIE

BOILEAU, André. *Présentation de Pov-Ray*. MAT 3812, Progiciels dans l'enseignement des mathématiques II. Hiver 2005.

MIAOU'S PLACE, *Didacticiel pour Pov-Ray*. <http://povray.free.fr/>.

ROBERT, Paul. *Le petit Robert*¹, Montréal, Les dictionnaires ROBERT-CANADA S.C.C, 1989, 2172 pages.

SORTAIS, Yvonne et René. *Géométrie de l'espace et du plan*. Paris. Hermann, Éditeurs des sciences et des arts. Collection : Formation des enseignants et formation continue. 1995. 395 pages.

FERRÉOL, Robert. *Encyclopédie des formes remarquables courbes, surfaces, fractals, polyèdres*. <http://www.mathcurve.com/> Site participant à l'Anneau des Mathématiques Francophones.