

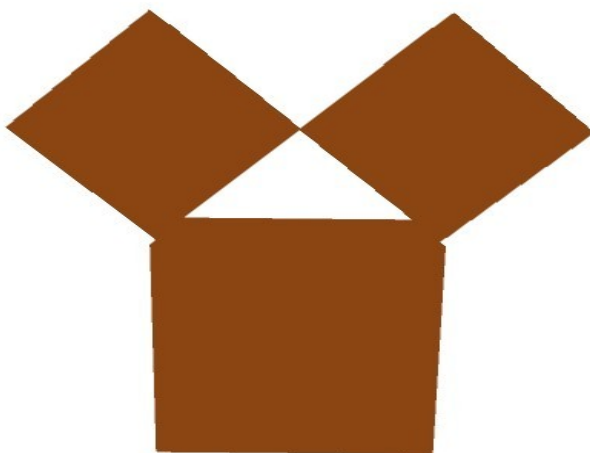
Rapport de Projet (Création Numérique)

Notre projet consiste à construire un arbre de Pythagore. Une fractale formée à partir de carrés dont chaque triplet constitue un triangle rectangle (figure permettant l'application du théorème de Pythagore). Cette fractale porte le nom d'«arbre» de Pythagore en raison de sa forme arborescente (le premier carré étant son «tronc» et la succession des autres carrés représentant l'axe principal avec ses «branches» repliées de part et d'autre).

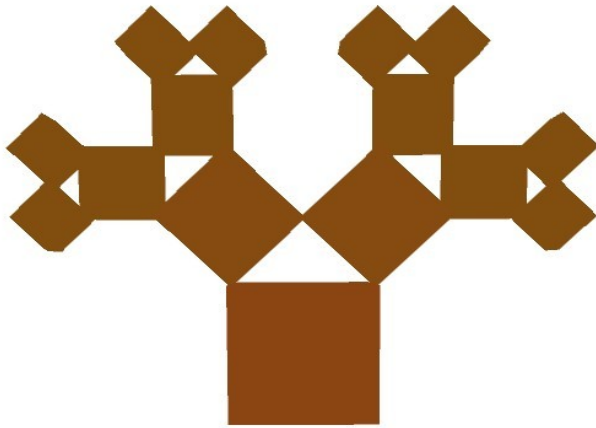
Pour cela, nous nous servons de 3 algorithmes :

- tracerSixFaces() pour tracer un cube.
- polToCart() créant un PVector de coordonnées cartésiennes à partir de coordonnées polaires.
- arbreDePythagore() étant l'algorithme récursif qui crée l'arbre à partir des procédures ci-dessus.

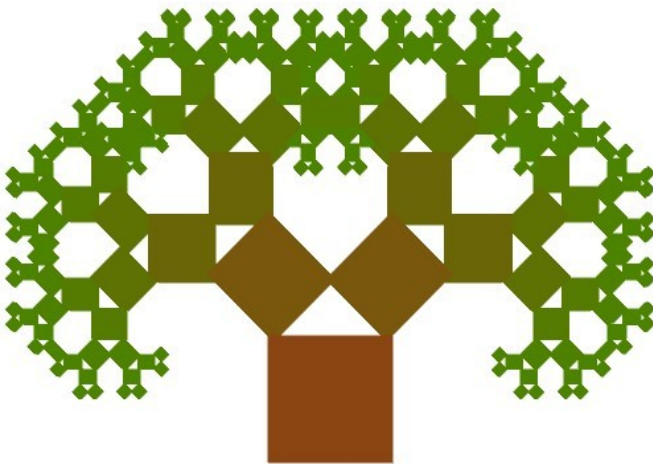
Tout d'abord, on commence par tracer un cube (comme la réunion de deux carrés), d'une couleur choisie par l'utilisateur, en fonction de ses 8 sommets grâce à la procédure tracerSixFaces(). Ensuite, on utilise les deux sommets sur lesquels tracer le niveau supérieur ainsi que les composantes RGB initialisées à 139, 69 et 19 respectivement afin d'obtenir du marron (rappelant la couleur du tronc d'un arbre classique). On calcule le côté des deux cubes créés grâce à la formule : $\cosinus * \text{hypoténuse} = \text{côté adjacent}$. Puis, on recrée les deux points de base avec la nouvelle altitude (correspondant à la moitié du côté calculé précédemment) pour la 3D. Par la suite, on construit successivement les 12 autres points formant les deux cubes (soit 16 points au total moins 2 puisqu'ils partagent une arête formée par 2 points) en se servant des formules trigonométriques de base, de la relation entre les degrés et les radians, et de polToCart() pour obtenir les coordonnées cartésiennes de chaque point.



Par ailleurs, à chaque niveau de récursion, l'angle varie de 45° (-45° pour les carrés de la partie de gauche et $+45^\circ$ pour ceux de la partie de droite) donc chaque couple de cubes posé sur le cube formé au niveau de récursion inférieur est décalé de 45° par rapport à celui-ci et de 135° par rapport au carré originel si on prend le dernier couple formé dans l'exemple ci-dessous. C'est cela qui rend possible cet effet d'enroulement caractéristique de l'arbre.



De plus, nous avons ajouté 3 variables pour les composantes RGB de la couleur des cubes dont les valeurs varient en fonction du niveau actuel de récursion. En effet, le triplet RGB se rapproche de $(0,255,0)$, correspondant à un vert pur, à mesure que l'on augmente le niveau de récursion. Ceci s'explique par le fait qu'on diminue la composante bleue et rouge et on augmente la composante verte à chaque appel récursif. Cependant, cela se produit à une vitesse suffisamment lente (cf code) pour que l'on voit les différentes nuances entre la couleur de départ et celle de fin. Ainsi, nous obtenons un dégradé du marron vers le vert clair pour être raccord avec l'apparence d'arbre que l'on voulait donner au projet.



L'effet d'arbre est particulièrement visible au dernier niveau de récursion puisqu'on voit clairement l'effet d'enroulement des deux côtés (par la divergence liée au décalage de 45° à chaque appel récursif) et le dégradé du marron jusqu'au vert car il y a eu suffisamment d'appels récursifs (niveau = 10 pour la figure ci-dessous) pour que la composante verte se détache des deux autres.

