

Compléments aux activités SCRATCH

Trigonométrie

Étant donné un polygone régulier convexe à n côtés, dont la longueur des côtés est notée l , on peut déterminer la longueur $L(p)$ des côtés du polygone étoilé inscrit de pas p , ainsi que l'angle de rotation du lutin à chaque sommet lorsqu'il parcourt ce polygone étoilé.

Notons α la mesure en degrés de l'angle de rotation du lutin à chaque sommet du polygone régulier convexe ; Comme le lutin fait un tour complet en tournant n fois de α , on a :

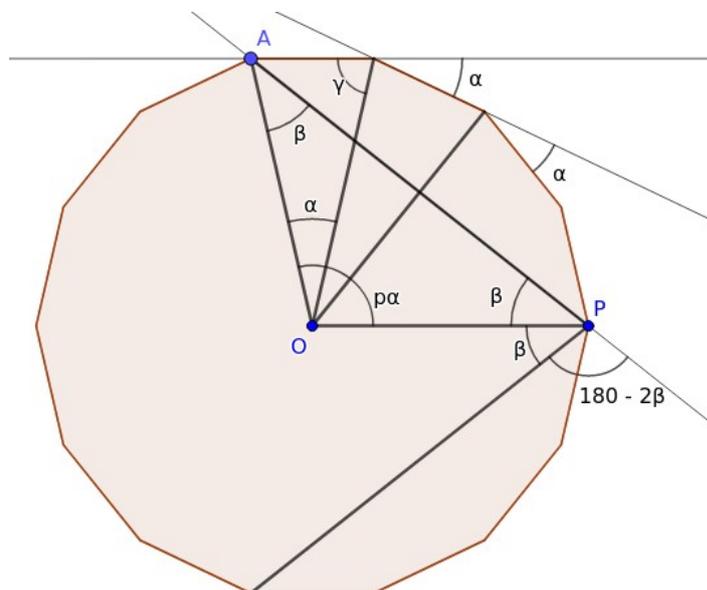
$$n\alpha = 360$$

donc α est aussi la mesure de l'angle au centre de chacun des triangles isocèles joignant le centre O du polygone convexe à deux sommets consécutifs (notons γ la mesure de chacun des deux autres angles de ces triangles, on a bien sûr $\alpha + 2\gamma = 180$). Il s'ensuit que $p\alpha$ est la mesure de l'angle au centre de chacun des triangles isocèles joignant O à deux sommets consécutifs du polygone étoilé de pas p . Notons β la mesure des deux autres angles de ces triangles, on a $p\alpha + 2\beta = 180$ et

$$180 - 2\beta = p\alpha$$

est la mesure de l'angle de rotation du lutin aux sommets du polygone étoilé à partir du 2^e sommet. Au départ, après avoir tracé le polygone convexe, le lutin est revenu en A , orienté à 90° (vers la droite), il doit donc tourner de $\gamma - \beta$ pour tracer le premier côté. On vérifie facilement que :

$$\gamma - \beta = (p - 1)\alpha / 2$$



Reste à calculer la longueur $L(p)$ des côtés du polygone étoilé. Notons R la distance OA et coupons le triangle isocèle OAP en deux triangles rectangles « d'angle au centre » de mesure $p\alpha / 2$, d'hypoténuse de longueur R et dont le côté opposé à l'angle au centre mesure $L(p) / 2$. On obtient

$$L(p) = 2R \sin(p\alpha / 2) = 2R \sin(180 p / n)$$

En particulier, pour $p = 1$, on retrouve le polygone convexe donc $l = L(1) = 2R \sin(180 / n)$.

Avec ces informations, on peut écrire une variante du script de l'activité 2 qui ne nécessite pas l'usage des listes de variables : après avoir tracé le polygone convexe, on commence par tourner de $(p - 1)\alpha / 2 = 180 (p - 1) / n$ puis on répète n fois « avancer de $L(p)$ et tourner de $p\alpha$ ».

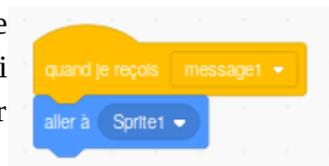
Remarque. On peut utiliser la formule $l = 2R \sin(180 / n)$ pour ajuster la longueur l des côtés du polygone convexe à la taille de la fenêtre graphique de Scratch (un peu plus de 300 pixels) : on choisit par exemple $R = 150$ donc $l = 300 \sin(180 / n)$; il faut alors placer le lutin au point de coordonnées $(0, 150)$ au départ du programme.

Parallélisme

On fait fonctionner deux lutins en parallèle (ajouter un lutin en cliquant sur l'icône « chat + »  dans le cadre des lutins), avec un script pour chacun, l'un appelant l'autre grâce à la commande (Menu Événements) :

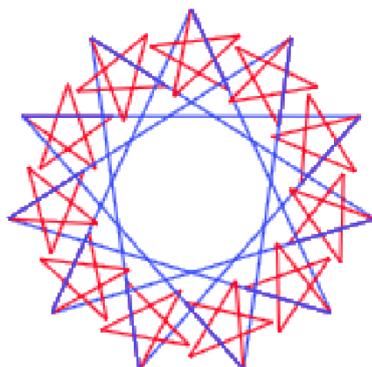


Les deux lutins se placent au même point au départ. Celui qui part en premier trace le polygone régulier convexe, il envoie un message pour que le second le rejoigne lorsqu'il arrive au premier sommet par lequel celui-ci doit passer, et ainsi de suite. Le premier lutin devra décrire plusieurs fois le polygone régulier convexe pour que le second trace entièrement le polygone étoilé inscrit.

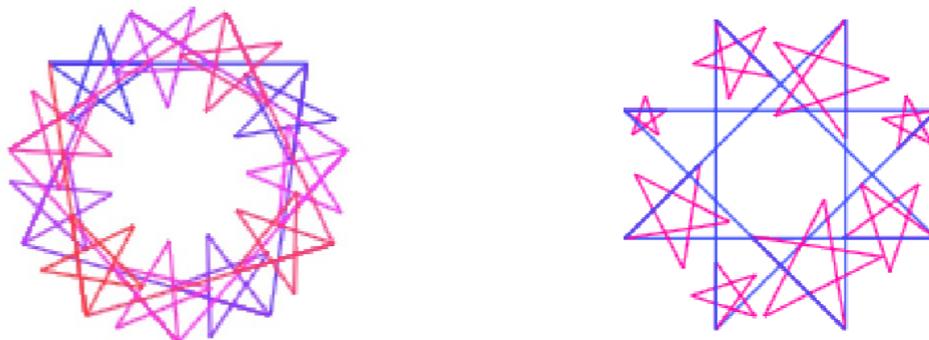


Blocs

Une fois au point le script de l'activité 1 qui trace des polygones étoilés, on peut en faire un bloc de programme (Menu « Mes blocs », *Créer un bloc*) qu'on pourra insérer dans d'autres scripts, en fixant le nombre de côtés et le pas, par exemple pour que ce bloc trace des étoiles à 5 branches (5 côtés, pas égal à 2) de taille suffisamment petite pour qu'on puisse en faire plusieurs. On peut alors faire appel à ce bloc dans le script de départ pour ajouter une étoile à 5 branches (de couleur différente !) à chacun des sommets du polygone étoilé, pour obtenir par exemple :



Noter la possibilité de faire dépendre le bloc d'un paramètre (quand on crée le bloc, cliquer sur « Ajouter une entrée *nombre ou texte* »), par exemple pour modifier la couleur en lui ajoutant un nombre lorsqu'on utilise le bloc. Une bonne pratique est d'utiliser alors un paramètre *local* pour décrire le fonctionnement du bloc, différent des variables utilisées dans le script principal ; on peut par exemple inscrire « couleur » dans la case *nombre ou texte* et l'utiliser (en faisant glisser l'ovale contenant « couleur » dans l'entête du bloc) pour ajouter « couleur » à la couleur du stylo. Quand on appelle ce bloc dans le script principal, on lui affecte alors une variable globale (par exemple le pas du polygone du script principal, même si cela n'a pas tellement d'intérêt), ce qui peut donner le dégradé suivant (à gauche) :



Autre exemple peut-être plus intéressant (illustré ci-dessus à droite) : faire varier la taille de l'étoile à 5 branches du bloc. On peut appeler « taille » le paramètre local du bloc et créer une variable globale « cote », initialisée à 20 dans l'exemple ci-dessus et à laquelle on ajoute 50 divisé par le nombre de côtés à chaque tour de boucle.

Primalité

On peut tester la co-primalité de deux entiers n et p en utilisant le script de l'activité 2, qui trace le polygone étoilé de pas p inscrit dans le polygone régulier convexe à n côtés. Il suffit pour cela d'insérer dans la boucle qui trace le polygone étoilé un test de retour au sommet de départ et une variable « compteur » qui compte le nombre de sommets parcourus avant le premier retour au sommet de départ. En effet, on a la propriété suivante :

Théorème. Deux entiers n et p sont premiers entre eux si et seulement si, lorsqu'on va de p en p dans un polygone à n côtés, on décrit tous les sommets avant de revenir au départ.

Preuve. On peut commencer par remarquer que, si d est un diviseur de n et qu'on écrit $n = dm$ avec m entier alors, en allant de d en d dans un polygone à n côtés (convexe ou étoilé), on revient au sommet de départ au bout de m déplacements. Si l'on note d le pgcd de n et p et qu'on écrit $p = dq$ et $n = dm$ avec q et m entiers, alors quand on va de p en p dans un polygone à n côtés, on ne passe que par les m sommets qui sont atteints lorsqu'on se déplace de d en d (car p est multiple de d). Il s'ensuit que si l'on passe par tous les sommets avant de revenir au départ, alors $m = n$ et $d = 1$, c'est-à-dire que p et n sont premiers entre eux.

Pour montrer la réciproque, numérotions les sommets de 0 (départ) à $n-1$ dans l'ordre dans lequel ils apparaissent lorsqu'on décrit le polygone à n côtés (convexe ou étoilé). Aller de p en p dans ce polygone signifie passer par les sommets dont les numéros sont congrus à un multiple de p modulo

n . Or si p et n sont premiers entre eux, alors tout entier est congru à un multiple de p modulo n : tout entier est multiple de 1 et, par le théorème de Bézout, 1 est congru à un multiple de p modulo n (autrement dit p est inversible modulo n). Il s'ensuit alors que tous les sommets sont atteints en allant de p en p dans le polygone à n côtés.

Remarque. Le raisonnement de la deuxième partie de la preuve peut facilement être transformé pour montrer directement l'équivalence énoncée dans le Théorème ci-dessus. Cependant de la première partie de la preuve, on peut déduire un peu plus : avec les mêmes notations ($p = dq$ et $n = dm$), aller de p en p dans un polygone à n côtés, c'est aller de q en q dans le « sous-polygone » obtenu en allant de d en d dans le polygone à n côtés ; or puisque d est le pgcd de n et p , m et q sont premiers entre eux, donc le parcours dans le sous-polygone est exhaustif (on passe par tous ses sommets). Il s'ensuit que le nombre de sommets parcourus est égal à m , c'est-à-dire à

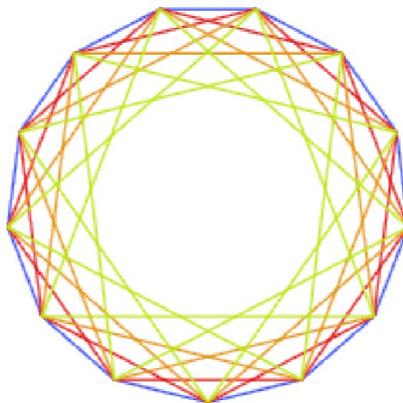
$$n / \text{pgcd}(n,p)$$

Le script de l'activité 2 modifié comme indiqué ci-dessus peut donc fournir le pgcd des entiers n et p , puisque le compteur donne le nombre m de sommets parcourus et que, par ce qui précède :

$$\text{pgcd}(n,p) = n / m$$

Enfin, on peut isoler dans ce dernier script le test de co-primauté et en faire un bloc permettant de tester la primalité d'un entier n , en testant s'il est premier à tous les entiers inférieurs à sa racine carrée à l'aide d'une boucle.

On obtiendra alors des figures avec des polygones étoilés de pas différents inscrits dans le même polygone régulier convexe, le pas variant de 2 jusqu'à la racine carrée de l'entier testé ou s'arrêtant lorsqu'on trouve un diviseur. Par exemple pour $n = 13$ (couleur modifiée à chaque incrémentation du pas) :



On remarque qu'on est allé jusqu'au pas 4 (en vert) alors que $4^2 = 16 > 13$, le script n'est donc pas optimisé (du moins pour les entiers non carrés). D'un autre côté on aurait pu poursuivre le test de co-primauté jusqu'à la partie entière de la moitié de l'entier n testé pour avoir sur la figure toutes les « espèces » de polygones étoilés à n côtés.