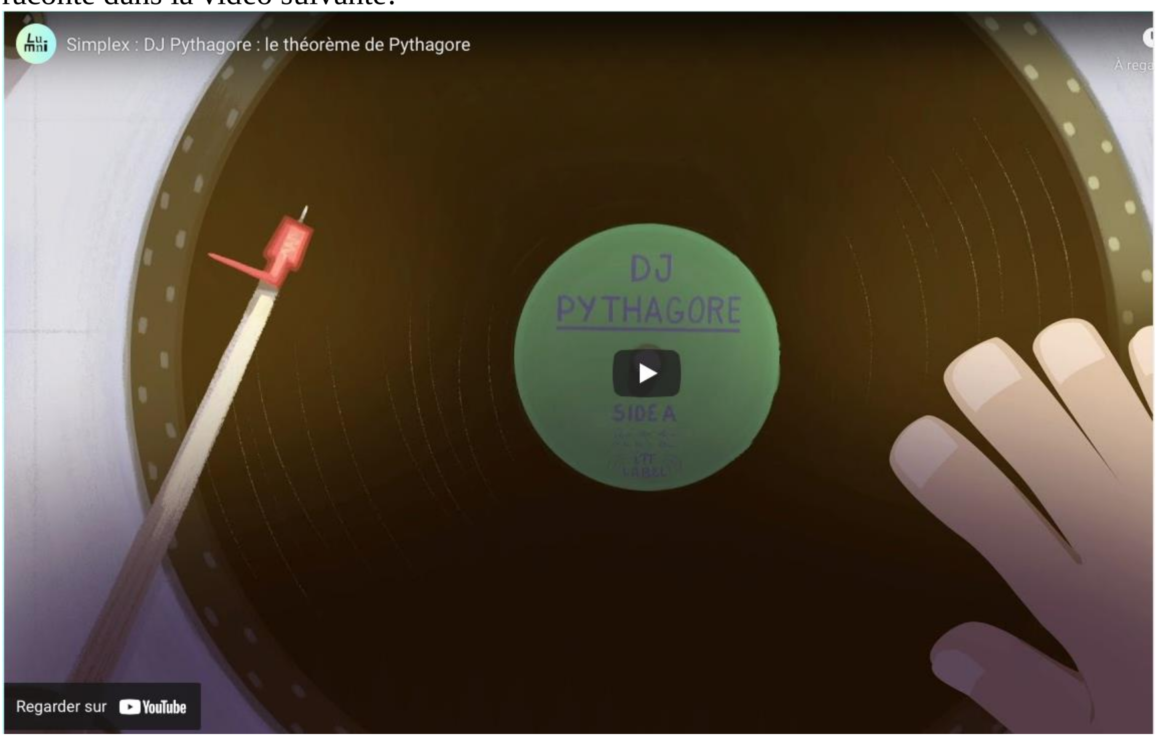
Programmer avec ***p5Visuel***[1] …
… pour **appliquer** les mathématiques

Quand on enseigne les mathématiques, la plupart de nos élèves se posent périodiquement la question suivante : à quoi ça sert? Et on essaie tant bien que mal de répondre à cette question. Mais il faut avouer que ce n'est pas toujours facile de trouver des exemples à la fois réalistes et assez simples. Nous allons voir que la programmation peut être source de tels exemples.

Considérons par exemple le théorème de Pythagore. Pour convaincre les élèves de l'utilité de ce résultat, on leur propose souvent d'utiliser ce théorème pour résoudre des problèmes comme celui-ci:

Si les cathètes d'un triangle rectangle mesurent 1 m et 2 m,
quelle est la mesure de l'hypoténuse de ce triangle?

Pour essayer de rendre la résolution de ce type de problème plus concrète et plus pertinente, on tente parfois d'ajouter un contexte plus *réaliste*, comme dans l'exemple raconté dans la vidéo suivante:



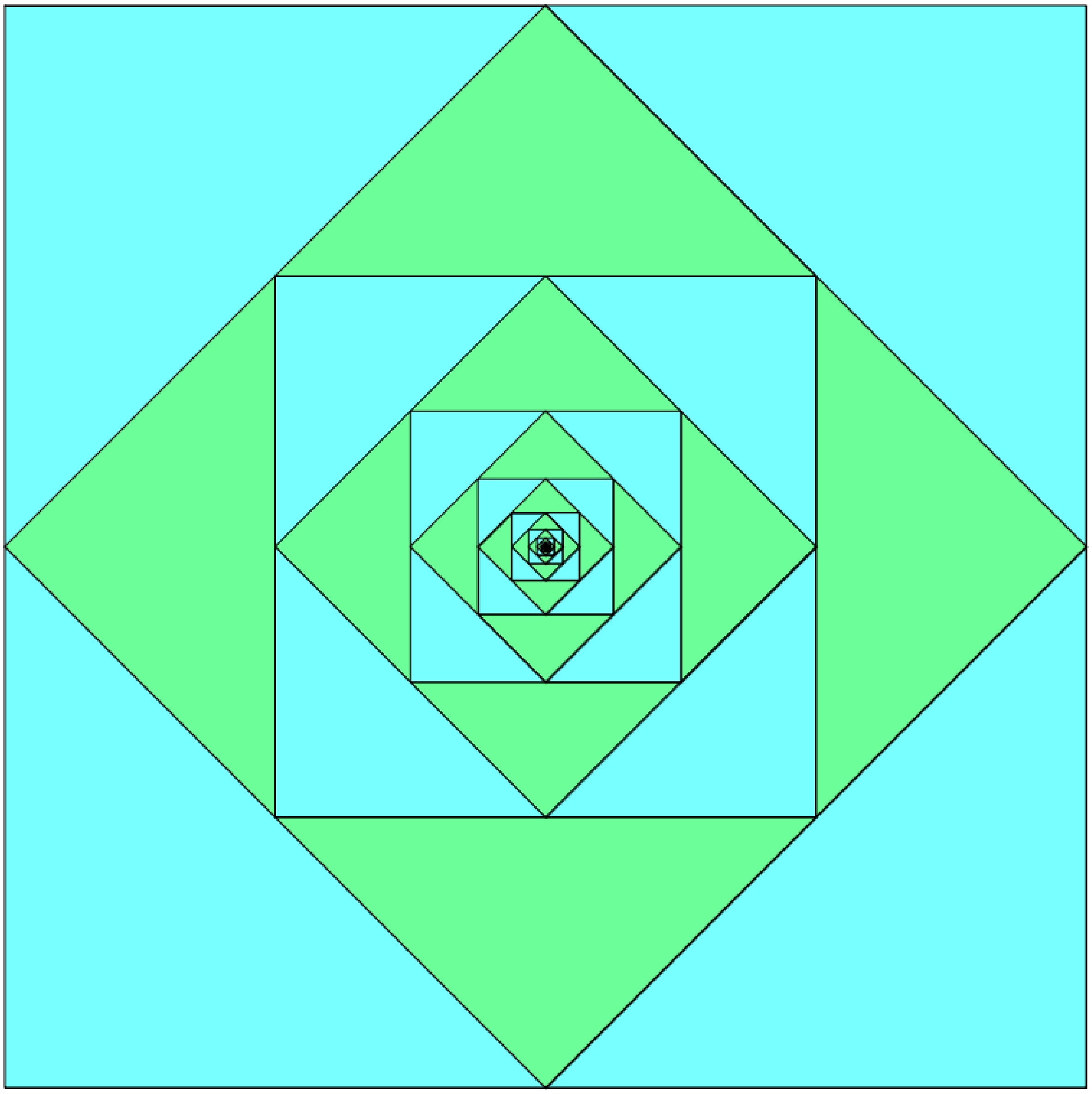
https://youtu.be/z5gMO8GTD-k

---

[1] ***p5Visuel*** est un logiciel libre et gratuit disponible à l'adresse suivante
http://profmath.uqam.ca/~boileau/p5VisuelWEB/index.html
Vous trouverez une version électronique du présent article à l'adresse suivante :
http://profmath.uqam.ca/~boileau/Nouvelles/Fichiers/Envol2021/article.pdf
Dans cette version web de l'article, tous les liens web de l'article seront actifs. Faire contrôle+clic (sur Windows ou Linux) ou commande+clic (sur Macintosh) pour ouvrir le lien dans une nouvelle fenêtre.

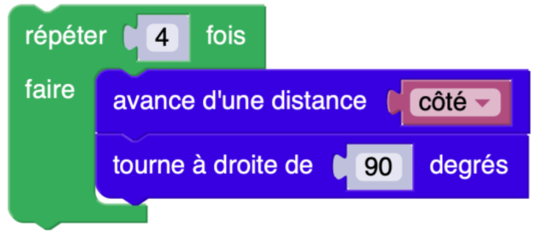
Il n'en reste pas moins que, malgré un habillage inspiré de la réalité, il s'agit quand même d'un problème qui reste assez artificiel : au lieu de mesurer la hauteur et la largeur du cadre de la porte, pourquoi ne pas mesurer directement sa diagonale?

Par contre, considérons le problème suivant: tracer la figure ci-dessous, comportant une suite de carrés emboîtés les uns dans les autres en leurs milieux:

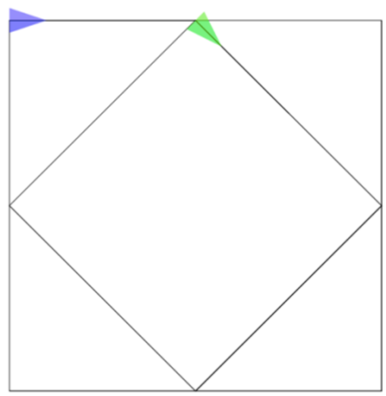


On peut imaginer plusieurs façons de procéder, dont plusieurs utilisant divers logiciels. Mais si on veut obtenir un résultat rapide et précis, on pourra choisir d'écrire un petit programme. Pour ce faire, nous allons utiliser ***p5Visuel***, un environnement de programmation visuelle (par blocs) orienté vers les mathématiques. Dans ce contexte, on peut tracer ladite figure de deux façons : en se référant aux coordonnées des sommets des carrés, ou en faisant appel à une *tortue* pour suivre le contour des divers carrés. Nous choisissons d'utiliser la *tortue*, ce qui nous permettra d'utiliser le théorème de Pythagore.

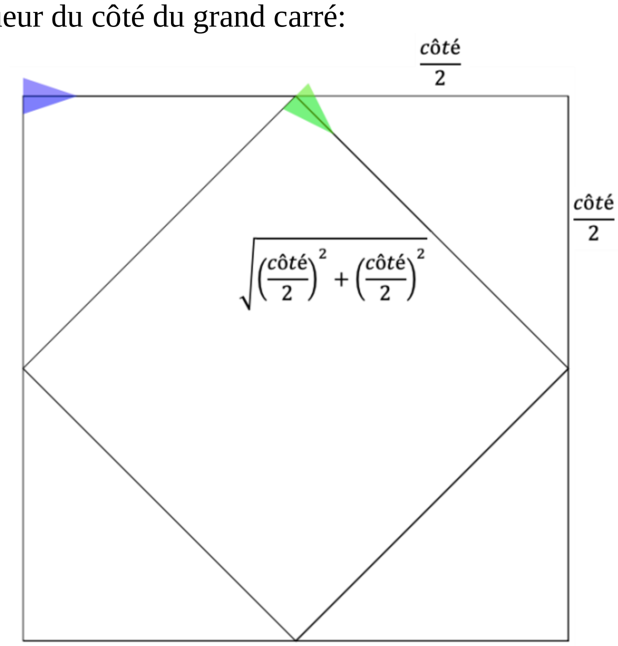
Pour tracer un carré à l'aide d'une *tortue*, il suffit de recourir aux instructions suivantes



où la variable *côté* contient la longueur du côté du carré. Puis il faut repositionner la tortue pour se préparer à tracer le carré suivant : la tortue doit passer de l'état *bleu* à l'état *vert* (voir la figure ci-dessous).



Ensuite, pour tracer le carré suivant, il faut calculer sa longueur. On peut le faire en appliquant le théorème de Pythagore aux triangles ci-dessous, dont les cathètes mesurent la moitié de la longueur du côté du grand carré:

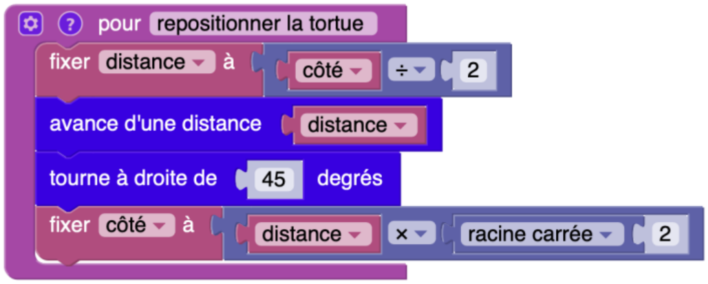


La longueur du carré suivant est donc de

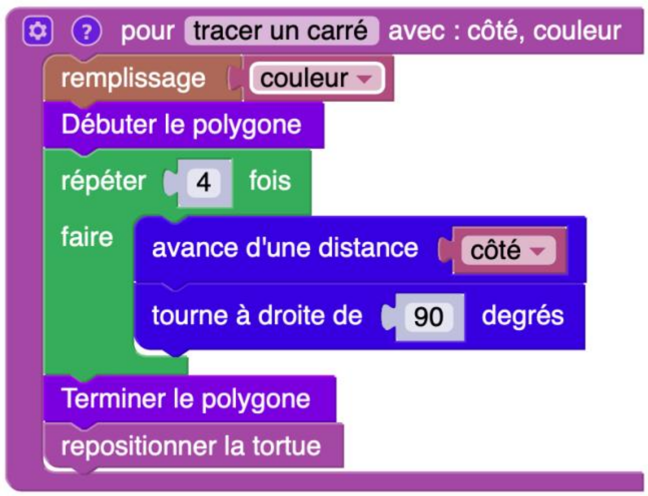
$$\sqrt{\left(\frac{côté}{2}\right)^2 + \left(\frac{côté}{2}\right)^2} = \left(\frac{côté}{2}\right)\sqrt{2}$$

.

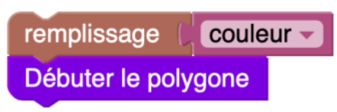
On peut donc déplacer la tortue pour tracer le carré suivant et recalculer la longueur requise en utilisant les instructions suivantes :



Pour simplifier, on peut intégrer ce repositionnement de la tortue dans la fonction qui trace un carré:



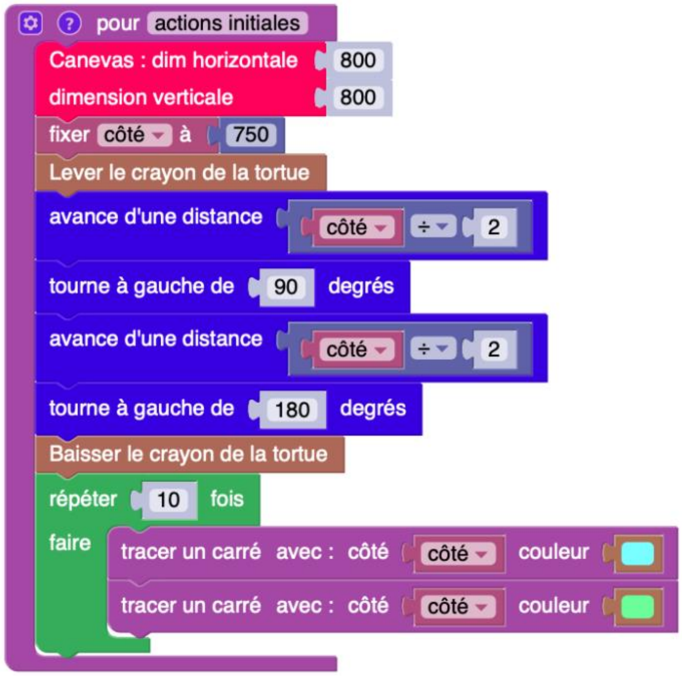
Veuillez noter l'ajout des blocs suivants



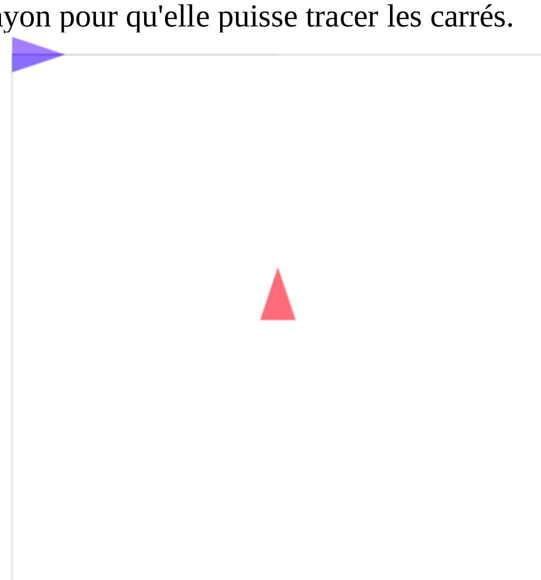
qui servent à colorer l'intérieur des carrés (ou, plus généralement, de tout polygone tracé par notre tortue). Les instructions pour tracer nos 20 carrés emboîtés seront donc:

Notez que, pour varier le nombre de carrés tracés, il suffira de changer le nombre utilisé dans le bloc *répéter*.

Il ne nous reste plus qu'à ajouter les instructions suivantes dans la fonction *actions initiales*, qui est lancée automatiquement au départ de ***p5Visuel***:



Après avoir fixé les dimensions du canevas (une zone où les graphiques seront tracés) à 800x800 et spécifié que le côté du plus grand côté sera de 750 (pixels), nous conduisons la tortue (sans tracer, c'est-à-dire en levant son crayon) à son point de départ. Au départ, la tortue est au centre du canevas (position rouge), et nous l'amenons en position bleue avant de baisser son crayon pour qu'elle puisse tracer les carrés.



Vous trouverez le programme complet à l'adresse

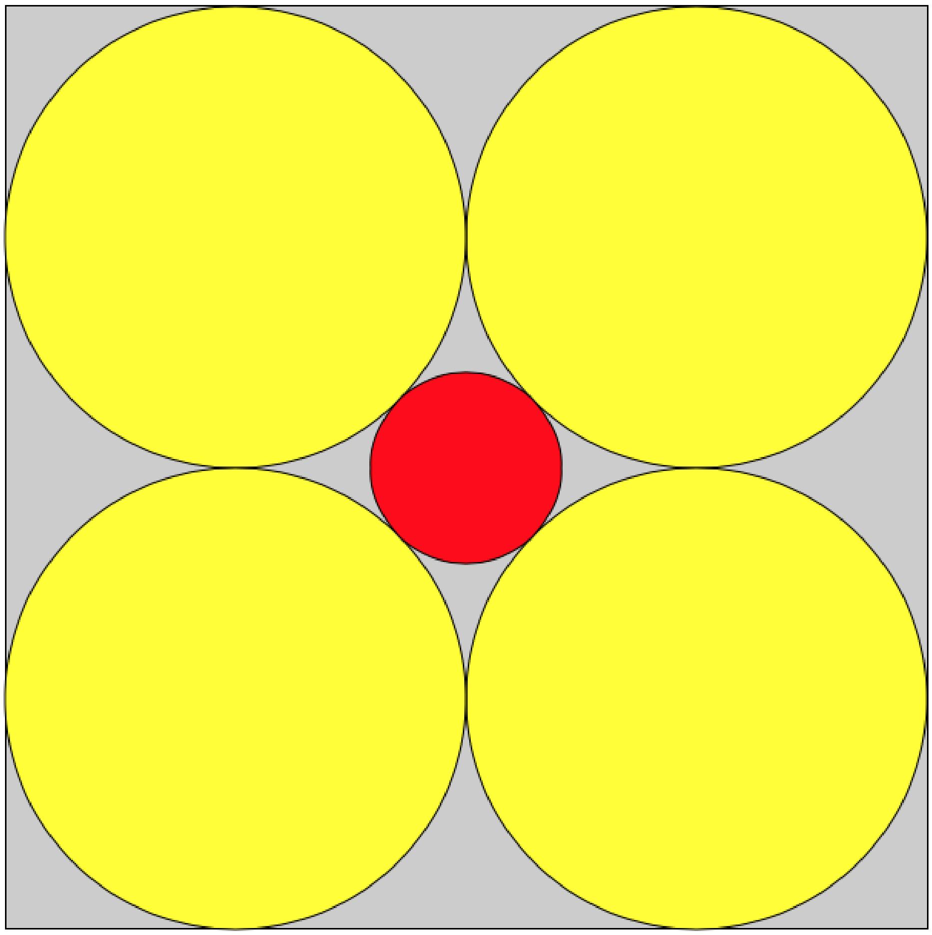
http://profmath.uqam.ca/~boileau/Nouvelles/Fichiers/Envol2021/carres_tortue

Comme je l'ai mentionné précédemment, on peut aussi tracer la même figure en utilisant les coordonnées des sommets des carrés plutôt que les déplacements d'une tortue

http://profmath.uqam.ca/~boileau/Nouvelles/Fichiers/Envol2021/carres_coords

Cette façon de procéder ne fait cependant pas appel au théorème de Pythagore...

Cependant, il y a des situations où on doit faire appel au théorème de Pythagore, même quand on utilise les coordonnées des divers points. Supposons que l'on veut réaliser la figure suivante, où chaque élément (carré ou cercle) est tangent à tous les éléments avec qui il est en contact.



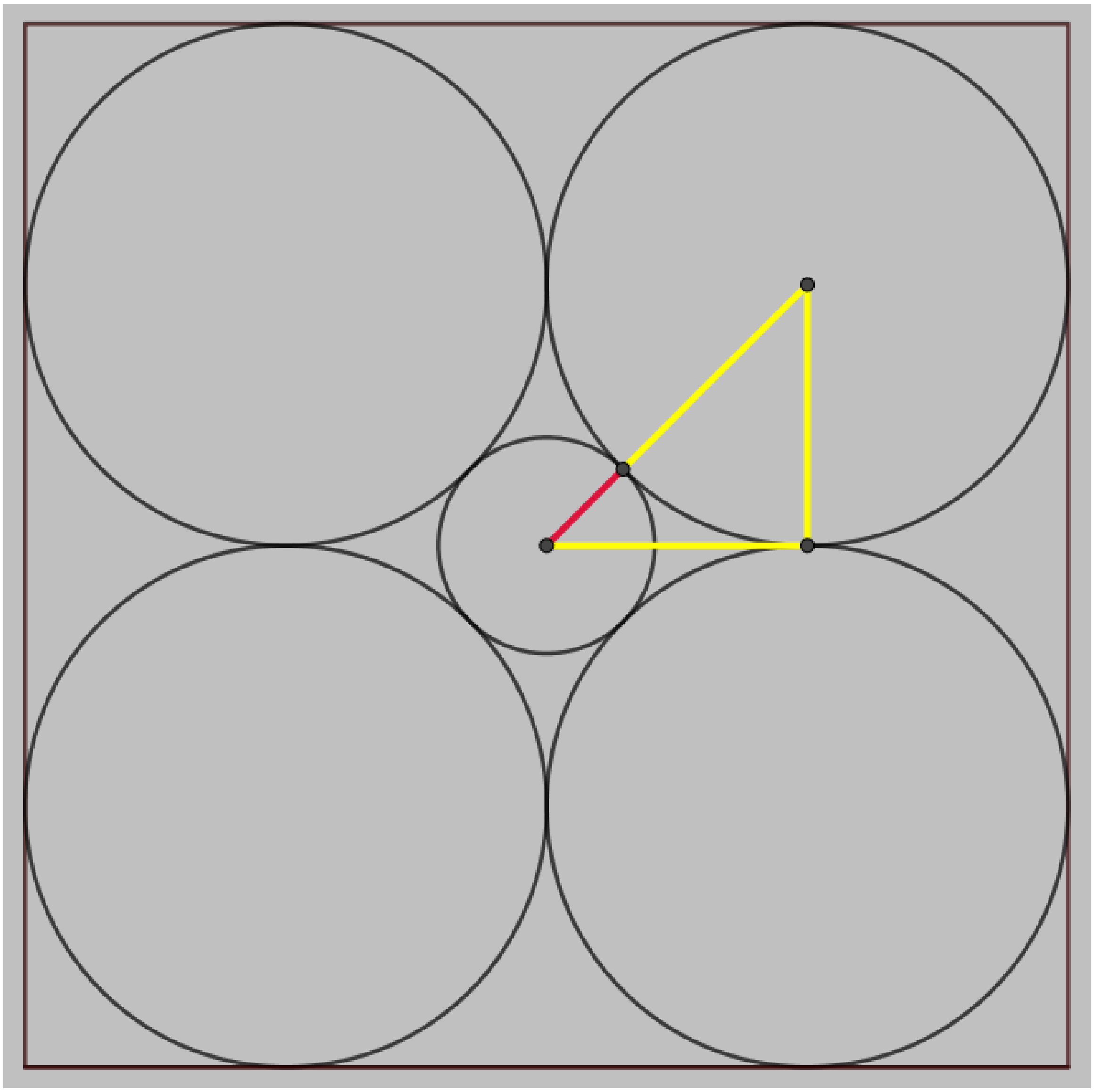
Si c est la longueur du carré de départ (en gris), et si l'origine du système d'axes est au centre de la figure, alors les cercles jaunes seront centrés aux points $\left(\pm\frac{c}{4}, \pm\frac{c}{4}\right)$ et auront $\frac{c}{4}$ pour rayons. Le cercle rouge sera centré à l'origine, et on devra calculer son rayon r. C'est là qu'intervient le théorème de Pythagore.

Considérons la figure ci-dessous, obtenue en prenant le « squelette » de la figure ci-dessus et en traçant quelques segments qui nous seront utiles. On connaît déjà la longueur des segments jaunes, soit $\frac{c}{4}$, et on veut calculer la longueur r du segment rouge. En appliquant le théorème de Pythagore, on obtient successivement:

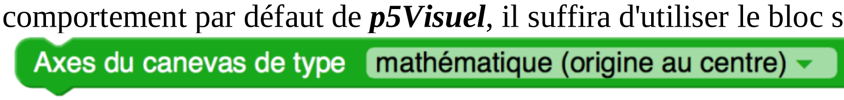
$$\left(r+\frac{c}{4}\right)^2 = \left(\frac{c}{4}\right)^2 + \left(\frac{c}{4}\right)^2 = 2\left(\frac{c}{4}\right)^2$$

$$r+\frac{c}{4} = \sqrt{2}\frac{c}{4}$$

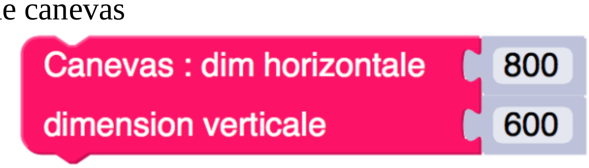
$$r = \sqrt{2}\frac{c}{4} - \frac{c}{4} = \left(\sqrt{2}-1\right)\frac{c}{4}$$



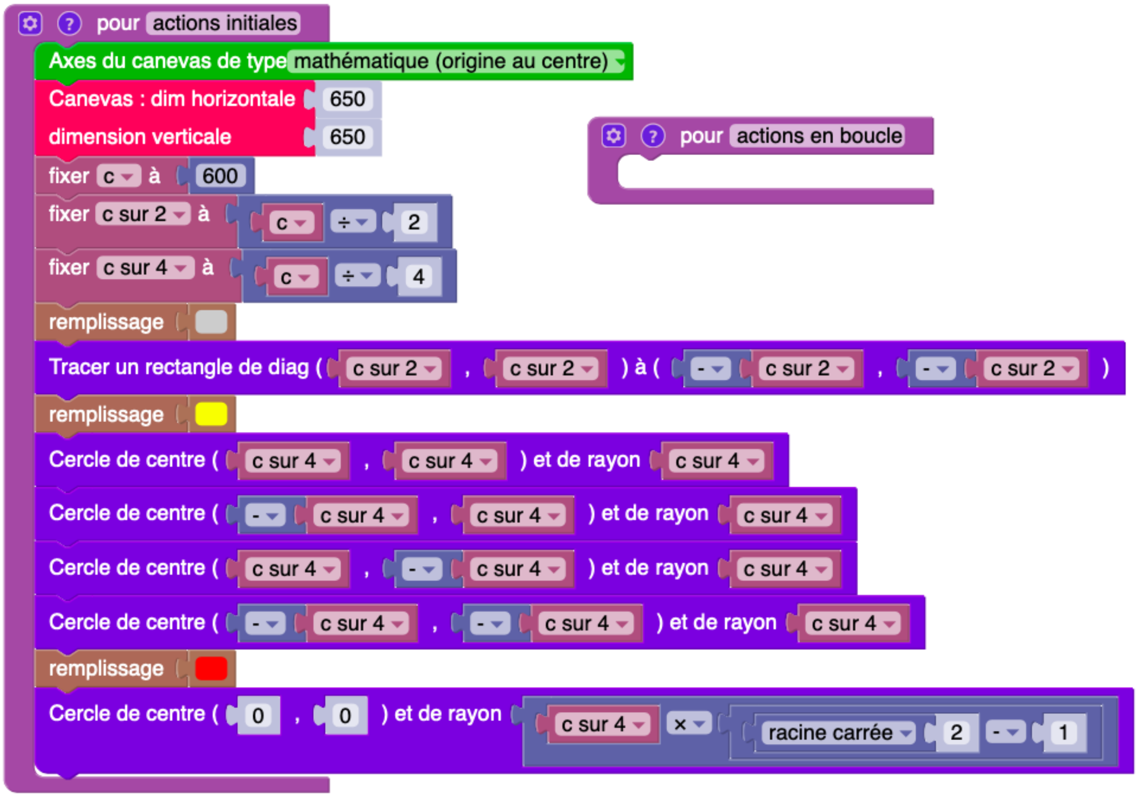
Avant de coder le tout dans ***p5Visuel***, il reste encore une petite difficulté à résoudre : dans notre analyse, nous avons utilisé un système d'axes mathématique (origine au centre et axe des *y* pointant vers le haut) alors que ***p5Visuel*** utilise par défaut un système d'axes informatique (origine au coin supérieur gauche et axe des *y* pointant vers le bas). Pour changer ce comportement par défaut de ***p5Visuel***, il suffira d'utiliser le bloc suivant



avant le bloc créant le canevas

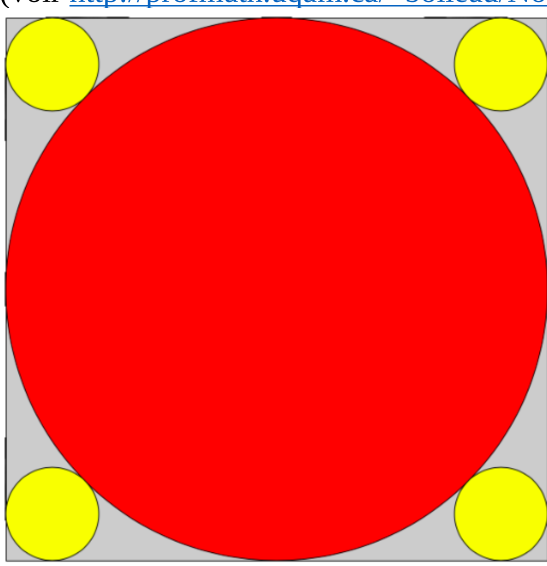


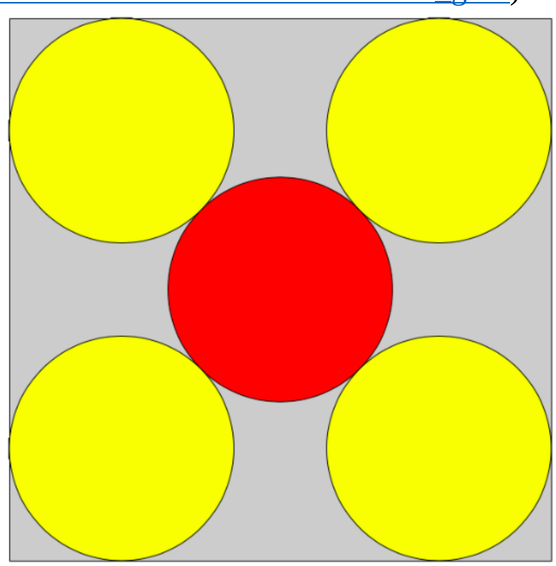
Nous obtenons alors le programme ci-dessous



http://profmath.uqam.ca/~boileau/Nouvelles/Fichiers/Envol2021/carre_cercles

On peut imaginer diverses variations sur le problème que nous venons de résoudre :
• cercle rouge le plus grand possible (à gauche ci-dessous)
• cercle rouge de même rayon que les cercles jaunes (à droite ci-dessous)
• cercle rouge prenant toutes les dimensions intermédiaires via une glissière
(voir http://profmath.uqam.ca/~boileau/Nouvelles/Fichiers/Envol2021/cercles_gliss)





Et ce n'est qu'un très bref aperçu de l'utilisation de la programmation pour donner des exemples d'applications des mathématiques. Pour plus de détails sur ***p5Visuel*** et pour d'autres exemples, vous pouvez consulter le lien suivant
http://profmath.uqam.ca/~boileau/p5VisuelWEB/p5Visuel/p5BLOCS/DOCU/YouTube/prog_polySpi/article.html

Un peu de recul

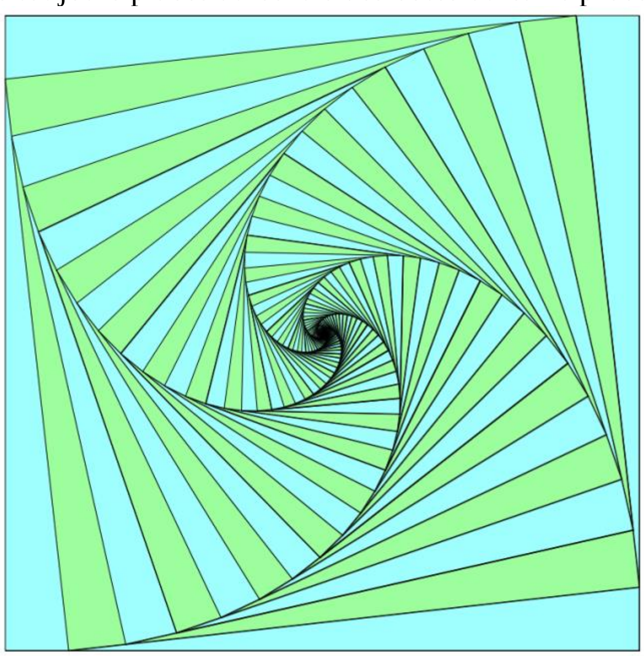
Nous venons de donner quelques exemples de problèmes issus de la programmation : dans chacun des cas évoqués ci-dessus, on voulait tracer une figure, et la programmation constituait une façon naturelle de réaliser nos objectifs. Mais ne pourrait-on pas raconter l'histoire de ces problèmes, et inciter nos élèves à les résoudre « théoriquement », sans utiliser la programmation? Sans doute, mais on perdrait beaucoup au change…

En effet, le recours à la programmation permet à l'élève de s'approprier le problème, de le transformer en un problème qu'il adopte personnellement, et non un problème que d'autres ont eu et qu'on lui demande de résoudre.

De plus, l'écriture et l'exécution d'un programme permet tout naturellement de vérifier nos réponses : si ça ne fonctionne pas comme prévu, on doit chercher et corriger la ou les erreurs commises.

Enfin, il ne faut pas négliger la sensation d'accomplissement ressentie lorsque notre programme fonctionne bien. Dans certains cas, cela pourra inciter certains élèves à aller plus loin, tant du point de vue de la programmation que du point de vue mathématique.

Imaginez alors la satisfaction du professeur voyant un élève lui demander de l'aider à réaliser plus que ce qui lui était demandé : par exemple, faire des carrés emboîtés dont les sommets ne sont pas toujours placés au centre des côtés du carré précédent.



http://profmath.uqam.ca/~boileau/Nouvelles/Fichiers/Envol2021/carres_spirales