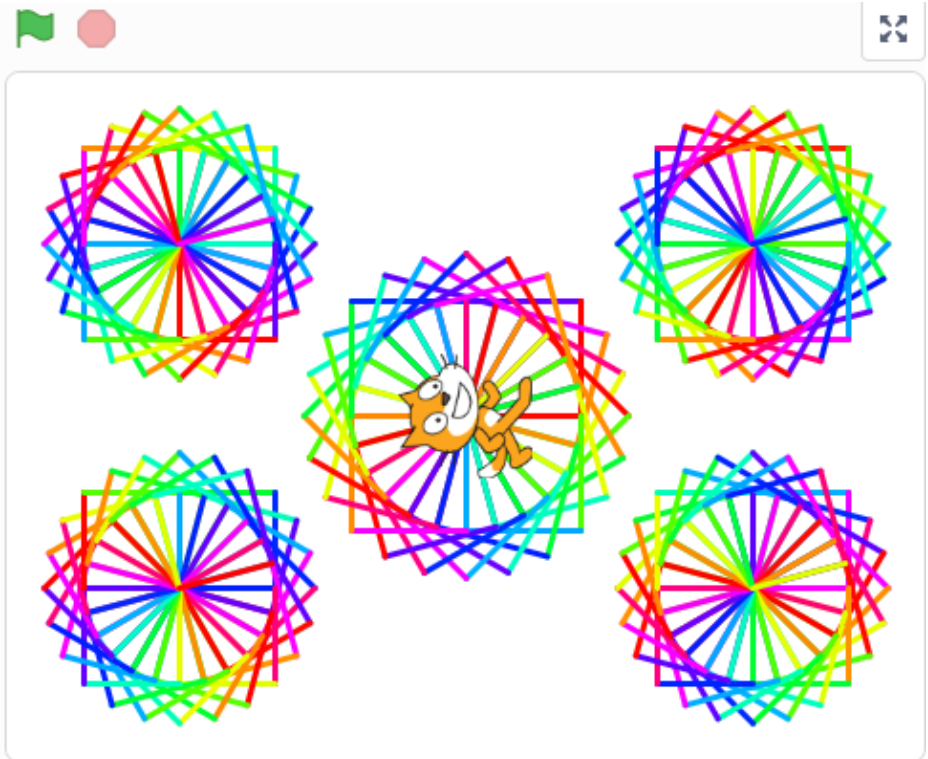


Des polygones de tous les côtés !



<https://scratch.mit.edu/projects/180940681>

Avec Scratch, nous allons demander à l'ordinateur de dessiner des figures. On commencera par des polygones simples pour arriver au final à de belles rosaces que tu pourras embellir et multiplier..

Au verso, on explique comment se servir de ces cartes.

Comment se servir de ces cartes ?

Chaque carte a un recto et un verso.

Au recto, tu trouveras ce symbole :



C'est la face Défi

Cette face te présente :

- un défi à réaliser,
- si nécessaire, des blocs qui pourront t'être utiles,
- si nécessaire, quelques conseils.

Si tu peux réaliser le défi sans aucune aide, si tu t'en sens capable, fais-le. Compare ensuite ta solution avec celle proposée au verso et passe à la carte suivante.

Si tu n'es pas trop sûr, fais-toi aider et retourne la carte

Au verso, un autre symbole qui représente « une solution ».

C'est la face Solution



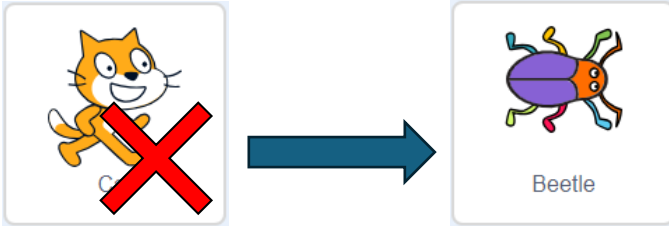
Cette face présente :

- un exemple de programme « solution » ,
- des trucs et astuces pour relever le défi,
- et quelques explications.

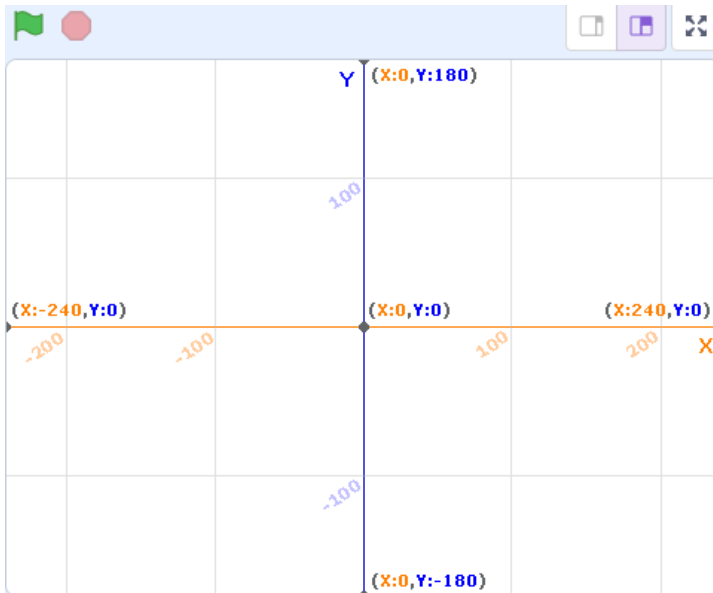
Prépare un sprite et un arrière-plan



Prépare ton projet en sélectionnant un nouveau sprite : **Beetle**. On préfère **Beetle** au sprite **Cat** car il sera plus facile à orienter. Supprime le **Cat**.



Ajoute un nouvel arrière-plan : **XY-grid**



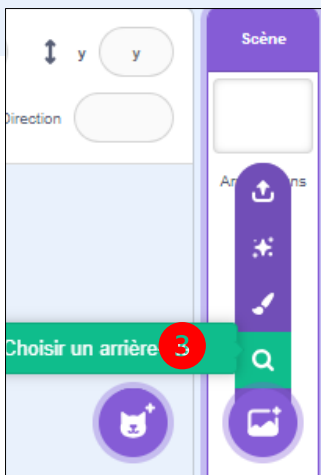
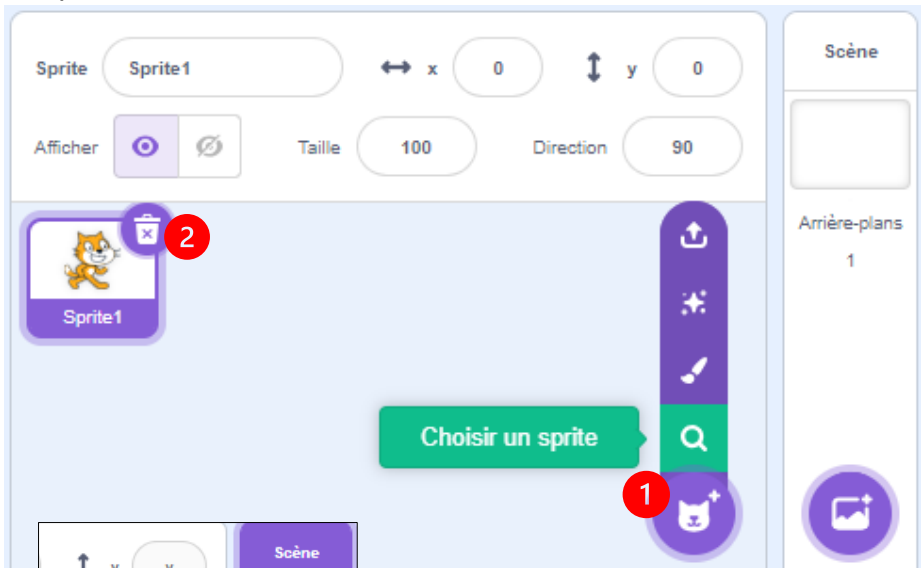
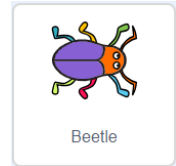
Ce n'est pas un bel arrière-plan mais tu en auras besoin pour te repérer sur la scène (le plan). Tu pourras le changer ultérieurement.

Prépare un sprite et un arrière-plan



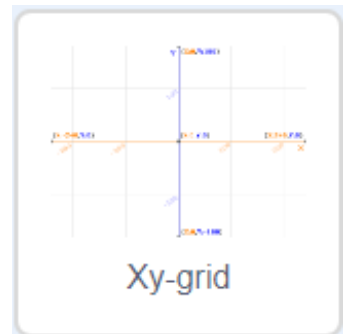
Sélectionner un sprite

- Clique sur le bouton **1 Choisir un sprite** et la petite loupe, sélectionne le sprite : **Beetle**
-
- **2** Supprime **Sprite1** (le chat) en cliquant sur la petite corbeille.



Sélectionne un arrière-plan

Clique sur **3 Choisir un arrière-plan** et sélectionne **XY-grid**,

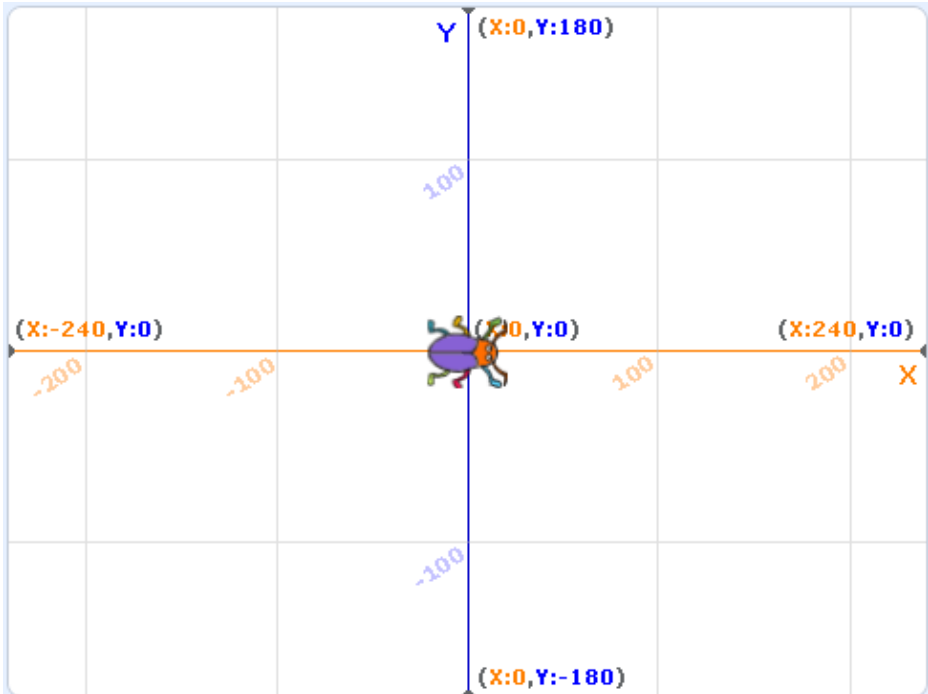




Quelques réglages

Placer *Beetle* au centre de la scène

S'il est bien au centre, ses coordonnées doivent être $x : 0, y : 0$



Vérifie l'orientation de Beetle

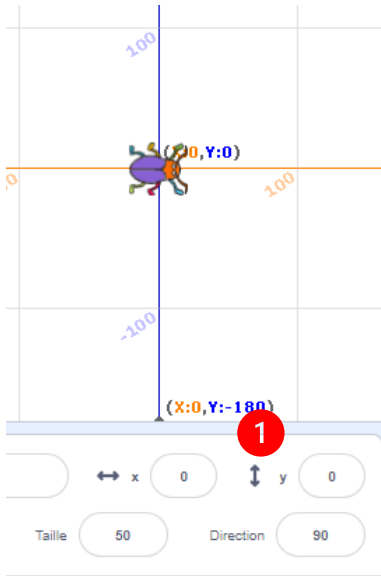
Beetle doit être orienté vers la droite, à 90°

Quelques réglages

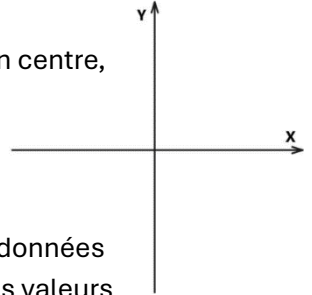


Placer **Beetle** au centre de la scène

Tu peux le déplacer avec ta souris, par cliquer-glisser. Le centre de la scène est au croisement des axes X et Y.



Si tu le places en plein centre, il devrait avoir ❶ les coordonnées **$x:0, y:0$** .



Tu peux modifier ces données en tapant de nouvelles valeurs dans ❶ les deux champs x et y.

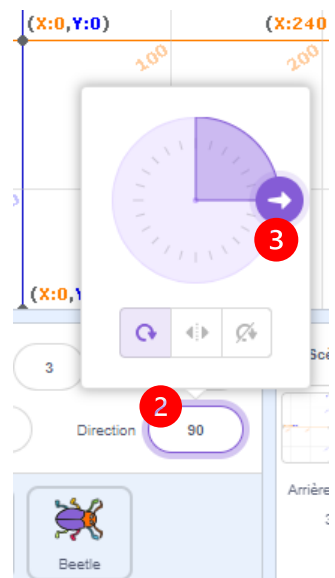
Retiens bien ceci : si tu modifie la valeur de Y, ton sprite va se déplacer verticalement. Si tu changes la valeur de X, il se déplacera horizontalement.

Vérifie la direction de Beetle

Clique sur le champ ❷ **direction** pour vérifier qu'il est bien tourné vers la droite. La valeur indiquée devrait être 90°.

Fais tourner la poignée ❸ et observe ce qu'il se passe.

Remet-là bien à 90° avant de passer à la suite.

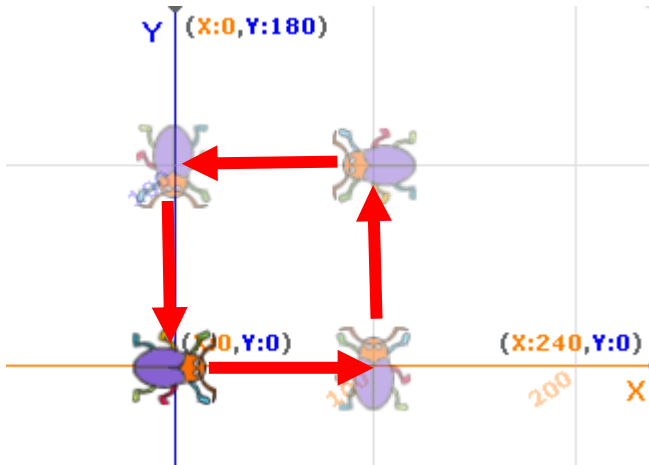


Beetle doit effectuer une trajectoire carrée



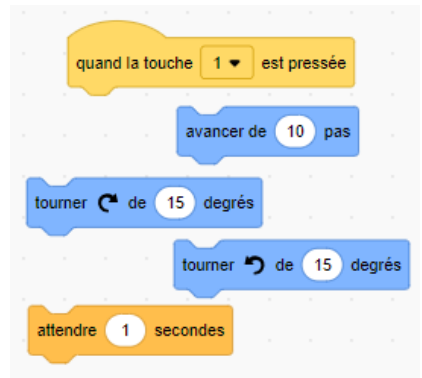
C'est parti pour un premier défi « programmation »

Beetle doit effectuer une trajectoire carrée en changeant de direction à chaque sommet du carré. Le carré doit avoir un coté de 100 pas (ou 100 pixels).



Tu peux utiliser ces blocs autant de fois que nécessaire. Duplique-les pour gagner du temps.

Clique sur un ou plusieurs blocs pour tester ton programme.





Beetle effectue une trajectoire carrée

Voici ton premier programme

Le bloc *quand la touche 1 est pressée* est un *bloc événement*. On peut aussi l'appeler *bloc de tête*. Il est toujours en tête d'un script. C'est lui qui va servir de déclencheur.

Les autres blocs sont simples. Comme presque tous les blocs de Scratch, il suffit de les lire pour comprendre.

Nous avons ajouté le bloc *attendre 0.5 secondes*. Pourquoi ? Retire-les pour voir ce qu'il se passe.



Explication . Si on empile des blocs de

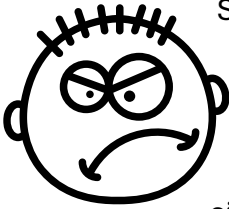
mouvement, Scratch les exécute tellement vite qu'on a l'impression qu'ils sont instantanés. En ajoutant une demi-seconde, on permet à notre cerveau de percevoir les 4 mouvements. Attention, on écrit **0.5** et non **0,5** !

Initialisation

❶ Ces deux blocs permettent de s'assurer que *Beetle* sera bien en place avant de commencer les déplacements.



Défi 1 : Moins d'instructions SVP !



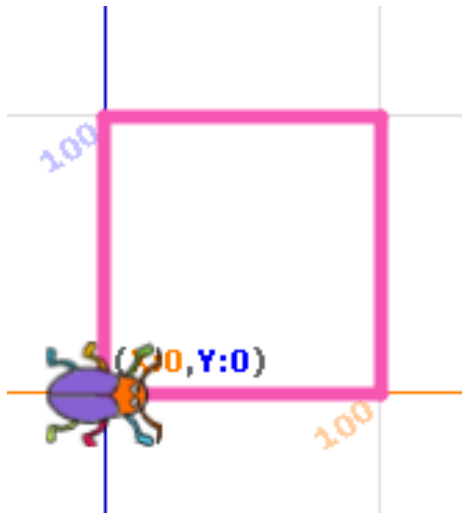
Si tu montres à un programmeur professionnel le code qui est proposé à la page 8, il risque fort de faire la grimace.

Les programmeurs détestent les redondances, c'est-à-dire les instructions identiques qui se répètent.

Essaie de trouver un bloc grâce auquel tu devrais pouvoir réduire de moitié le nombre de blocs utilisés et pour le même résultat.

Défi 2 : laisse une trace

Scratch propose des instructions (des blocs) grâce auxquelles ton sprite va dessiner un trait en se déplaçant.



Cette catégorie de blocs se cache bien mais tu peux la trouver...

Une fois que tu l'as trouvée et ajoutée aux blocs de Scratch, modifie le code pour arriver à un résultat comme celui-ci...

Tu peux essayer d'autres couleurs...

Moins d'instructions SVP !



Modifie le code

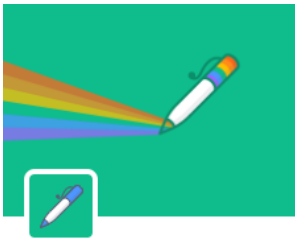
C'est si simple... Voici les avantages du **bloc répéter ... fois** :

- Si on décide de changer le nombre de pas, il ne faut le faire qu'une fois !
- Le programme occupe moins de mémoire dans l'ordinateur.
- Le programme est plus lisible.

Laisse une trace

Clique sur le bouton **Extensions** et sélectionne l'extension **Stylo**. Il te reste à ajouter quelques blocs...

Le bloc **effacer tout** est souvent placé en début de script pour effacer



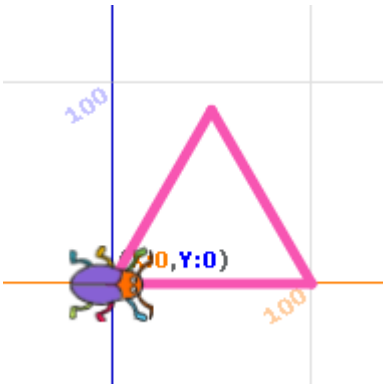
Stylo

Dessiner avec vos sprites.

les traces précédentes .



Et si on dessinait un triangle !



Duplique le code qui produit un carré et effectue les modifications utiles pour que Beetle dessine :

- Un triangle équilatéral
- Dont le côté est 100 pas
- Au départ du point $x : 0$; $y : 0$
- Quand la touche 2 est pressée

En principe, tu ne dois déplacer ni ajouter aucun bloc. Il suffit de modifier certaines valeurs.

Un petit coup de main ?

Pour dessiner un carré, Beetle tourne 4 fois de 90°.

Quand il revient à sa position de départ, il a pivoté de 360°.

$$4 \times 90^\circ = 360^\circ$$

Si tu tiens le même raisonnement pour un triangle équilatéral, tu devrais trouver sans problème de combien de degrés il faut tourner aux sommets.

Pour dessiner un triangle, Beetle tourne fois de°.

Quand il revient à sa position de départ, il a pivoté de 360°.

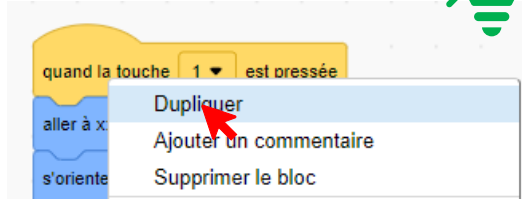
$$3 \times \text{.....}^\circ = 360^\circ$$

Si tu as encore besoin d'aide, au verso, il y a une figure...

Et si on dessinait un triangle !



Duplique le code
« dessine un carré »



Modifie le code dupliqué

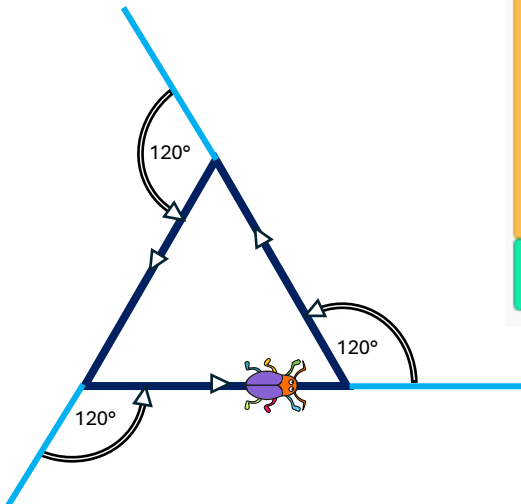
Il y a juste trois petits changements.

On modifie :

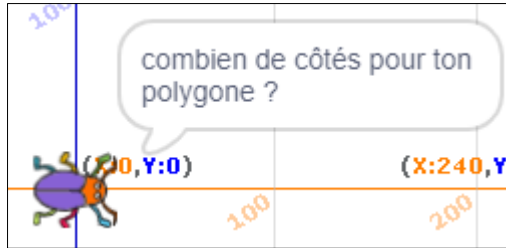
- le bloc événement : **quand la touche 2 est pressée**,
- de 4 côtés, on passe à 3,
- et l'angle de rotation vaut 120°

Explication

Beetle tourne trois fois de 120° ,
c'est-à-dire 360° au total



Choisir un polygone

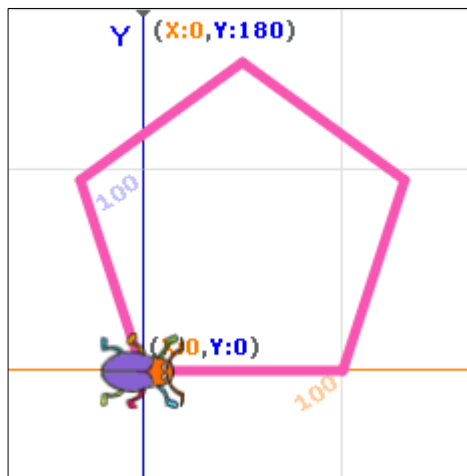


Un peu plus compliqué...

Voici le défi :

- Beetle doit interroger l'utilisateur et lui demander le nombre de côtés que le polygone doit compter.
- Le programme doit mémoriser ce nombre et recalculer les différents paramètres afin d'obtenir un dessin qui correspond au polygone demandé.
-

Le but est par exemple d'obtenir ceci si on introduit le nombre 5 :



Bon codage !



Choisir un polygone

Le code

Duplique le code précédent et modifie-le.

Les blocs **demander...** et **attendre** et **réponse** se trouvent dans la catégorie **capteurs**.

Son rôle ?

- Poser une question,
- Attendre que l'utilisateur **1** tape une réponse et la **2** valide
- Stocker la valeur dans une variable **réponse**. Cette variable existe, il ne faut pas la créer.

```
quand la touche 3 est pressée
  aller à x: 0 y: 0
  s'orienter à 90
  effacer tout
  mettre la couleur du stylo à [rose]
  mettre la taille du stylo à 5
  demander combien de côtés pour ton polygone ? et attendre
  stylo en position d'écriture
  répéter réponse fois
    attendre 0.5 secondes
    avancer de 100 pas
    tourner de 360 / réponse degrés
  relever le stylo
```

On utilise ensuite la variable **réponse** pour :

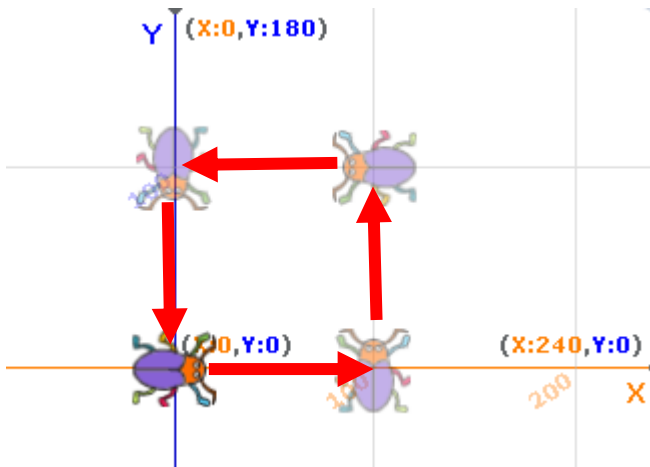
- La boucle qui devient **répéter réponse fois**, c'est-à-dire, pour un pentagone, **répéter 5 fois**
- Obtenir la valeur de l'angle externe en faisant ce calcul :
 $360^\circ / \text{nombre de côtés} = \text{angle externe}$
Pour un pentagone : $360^\circ / 5 = 72^\circ$

Un mouvement plus fluide SVP !



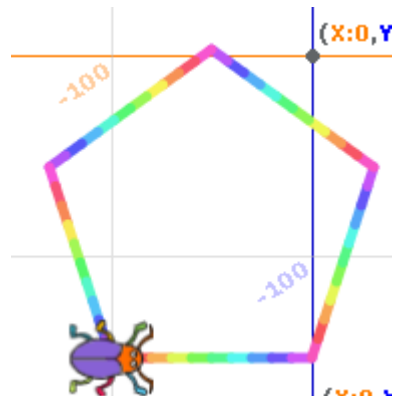
Lors de son déplacement, le scarabée *Beetle* se déplace très vite d'un sommet à l'autre. Et les rotations aux angles sont très brutales. Trop brutales !

Trouve un moyen pour que le déplacement et les rotations soient plus fluides.



Tu pourras solutionner ce problème avec des boucles. Après, tu pourras embellir un peu en ajoutant des couleurs.

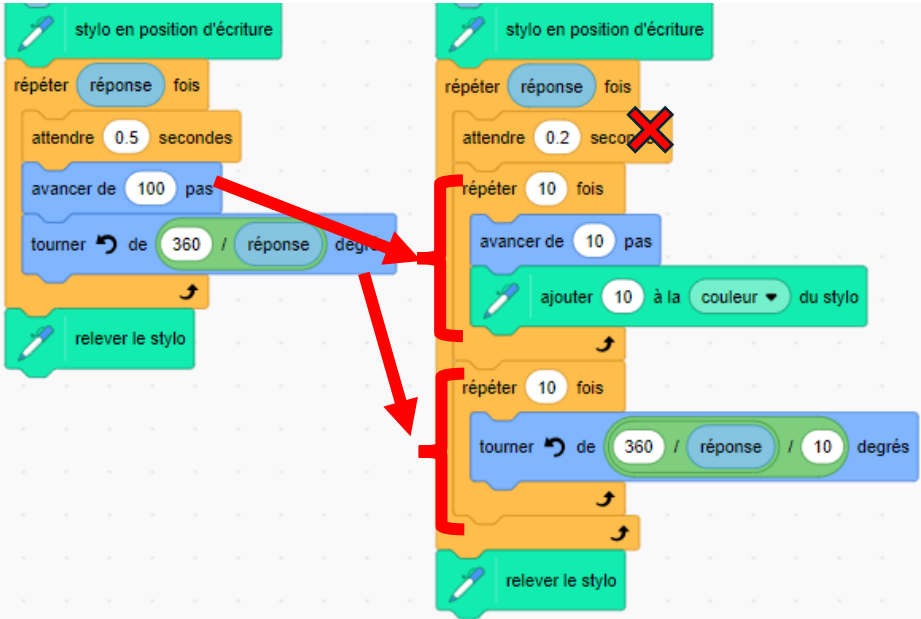
Bon amusement !



Un mouvement plus fluide SVP !



Duplique et modifie le code :



Avancer de 100 pas ou avancer 10 fois de 10 pas, pour ton prof de Math, c'est pareil.

$10 \times 10 = 100$. Pas de discussion !



Pour Scratch, il y a une différence importante.

Les boucles s'exécutent à une vitesse fixe : **30 x par seconde**. Donc, avancer 10 x de 10 pas prend un peu de temps. $10/30^{\text{ièmes}}$ de seconde = $1/3$ de seconde.

Alors que avancer de 100 pas prend... zéro secondes...

Et nous effectuons la même opération pour la rotation qui désormais deviendra **répéter 10 fois tourner de ...° / 10**. Ne pas oublier de diviser par 10 !

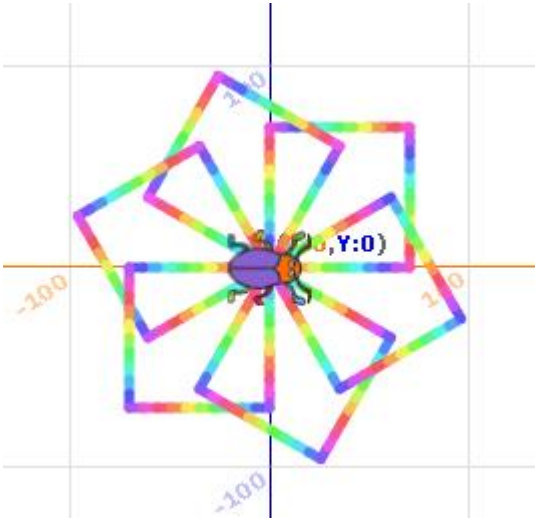


Des rosaces

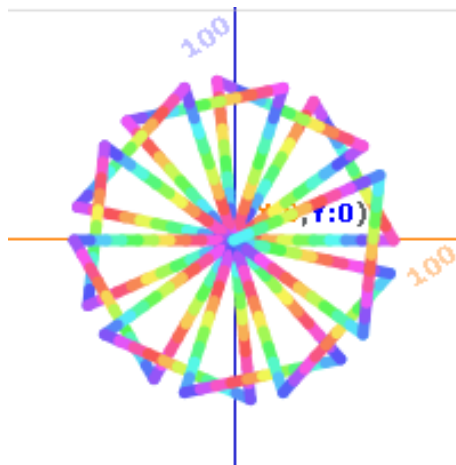
Demande à Scratch de dessiner plusieurs polygones pour obtenir de belles rosaces.

Il faudra juste ajouter une boucle et une rotation.

Dans cas-ci : répéter 6 fois et tourner de 60 degrés...



Essaie aussi :



Exemple de
code :



```
quand [drapeau] est cliqué
  aller à x: 0 y: 0
  s'orienter à 90
  effacer tout
  mettre la couleur du stylo à [rose]
  mettre la taille du stylo à 5
  demander [combien de côtés pour ton polygone ?] et attendre
  stylo en position d'écriture
  répéter 6 fois
    répéter [réponse] fois
      répéter 10 fois
        avancer de 5 pas
        ajouter 10 à la couleur du stylo
      répéter 10 fois
        tourner de (360 / réponse / 10) degrés
    tourner de 60 degrés
  relever le stylo
```