ÉDUCATION NATIONALE
Luxembourg

5G Informatique

Logique Informatique

Programmation avec

SCRATCH

Crée des histoires, des jeux et des animations.

Version 3.0 du 11 août 2020

Introduction

Scratch est un langage de programmation qui facilite l'apprentissage de la programmation de l'ordinateur. Il te permet de créer des histoires interactives et de développer des jeux et des animations.

Tu peux télécharger gratuitement le programme Scratch depuis le site web scratch.mit.edu à travers l'url https://scratch.mit.edu/download. Tu trouves également sur ce site une multitude d'informations sur Scratch (des modes d'emploi, des tutoriels vidéo, des cartes Scratch, la foire aux questions (faq), etc.), ainsi qu'un éditeur en ligne avec lequel tu peux créer des projets Scratch. Et finalement tu peux t'inscrire sur ce site officiel et publier tes propres projets Scratch.

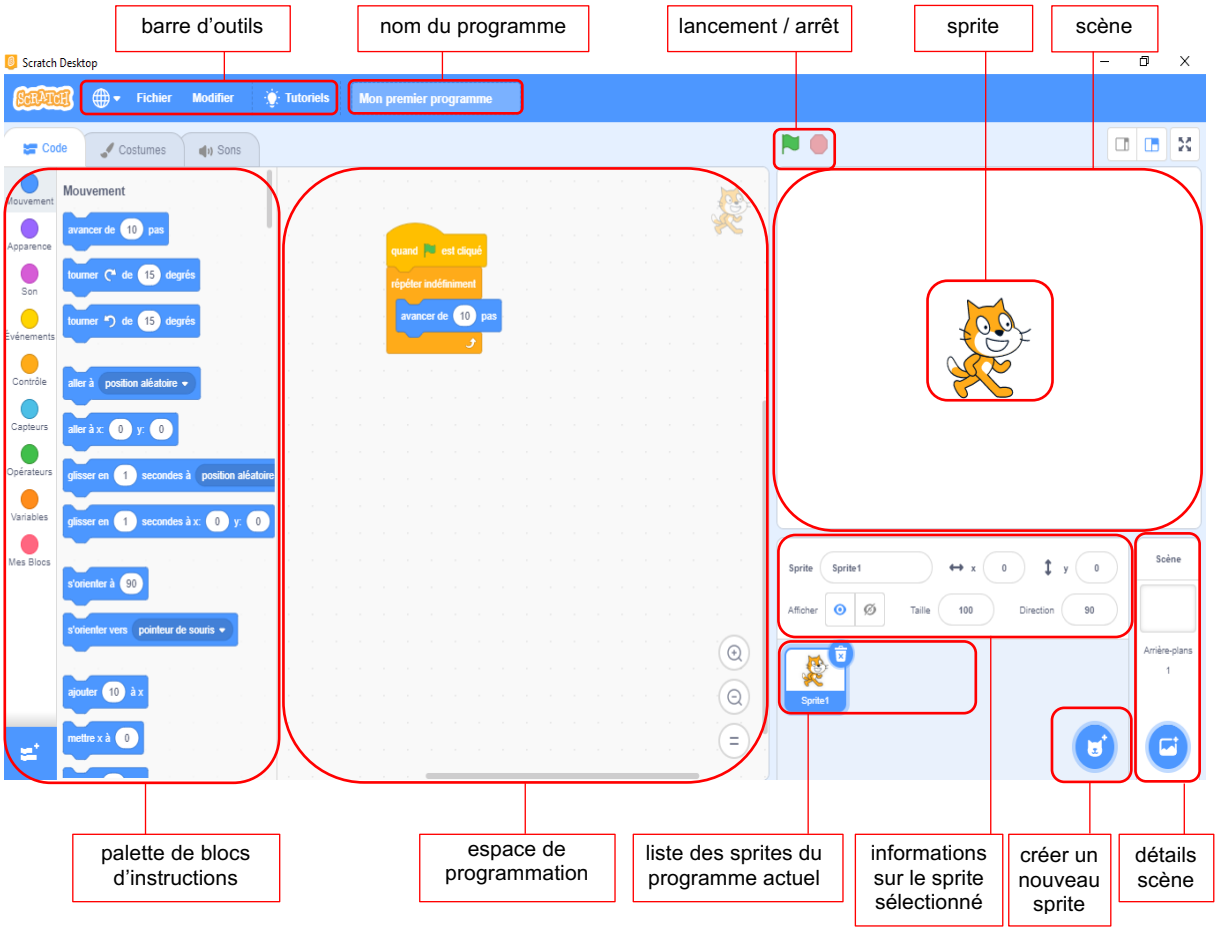
Ce manuel va t'introduire étape par étape dans la programmation avec Scratch.

Les captures d'écran s'appuient sur la version 3.10.2 de Scratch sous Windows 10.

A. L'environnement Scratch

Ce chapitre te permet de découvrir l'environnement dans lequel tu développeras tes projets Scratch.

A.1. L'interface du programme



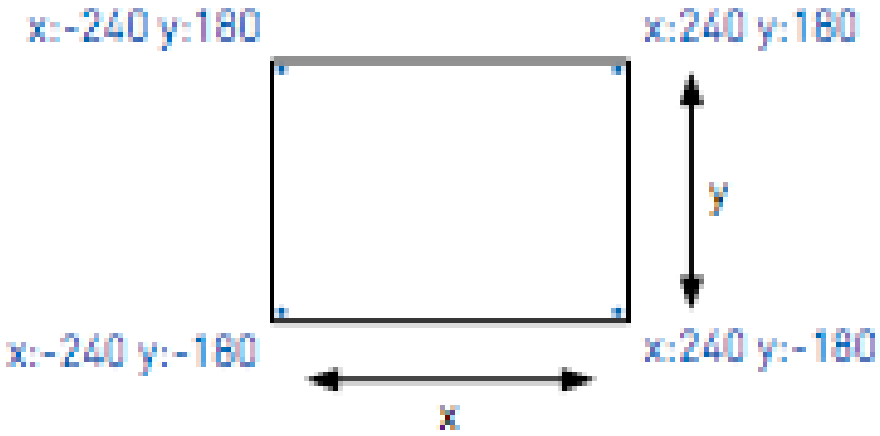
A.2. Eléments principaux d'un projet Scratch

A.2.1 La scène

La scène est le lieu où se déroulent les histoires, jeux et animations.

Les sprites (objets, voir plus loin) se déplacent sur une scène et agissent les uns avec les autres.

La scène a une largeur de 480 unités et une hauteur de 360 unités. Elle possède un système de coordonnées (repère) XY. Le centre de la scène a pour abscisse 0 et pour ordonnée 0.

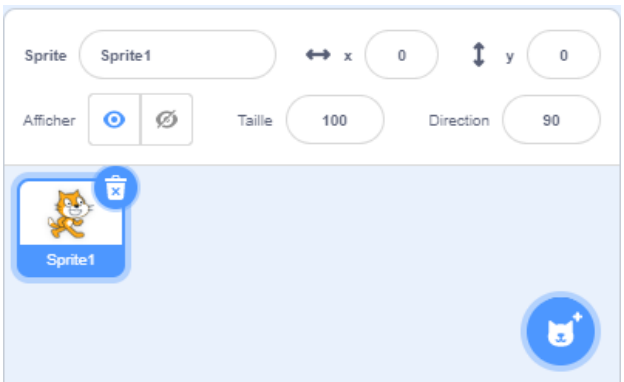


A.2.2 Les sprites

Un projet Scratch est composé d'un certain nombre d'**objets** appelés **sprites**. Sprite veut dire littéralement ogre mais désigne ici un graphique numérique. Dans la version française du programme on parle de **lutin**.

Tu peux modifier l'aspect d'un sprite en lui donnant une apparence selon ton choix, p.ex. celle d'une personne, d'un train, d'un papillon, etc.

Un sprite possède un nom. Scratch en donne automatiquement un (Sprite1, Sprite2…), mais pour que tu puisses mieux te repérer dans ton projet, il t'est conseillé de leur donner des **noms significatifs**.



A.2.3 Les instructions

Tu peux faire exécuter des ordres à un sprite, lui dire de bouger, de jouer de la musique ou de le faire interagir avec d'autres sprites. Pour communiquer une **instruction** à un sprite, tu dois faire glisser les **blocs graphiques** vers l'espace de programmation.

Ces blocs d'instructions sont regroupés dans des palettes d'instructions. Chaque instruction possède une couleur selon son type et se trouve dans la palette correspondante à son type.

Ainsi toutes les instructions servant au déplacement d'un sprite ont la couleur bleu foncé et sont regroupées dans la palette ***Mouvement***.

En plus les différents blocs d'instructions se distinguent également par leur forme.

A.2.4 Les programmes

Un programme est une suite d'instructions exécutées l'une après l'autre. Pour développer un programme Scratch tu dois assembler les blocs d'instruction nécessaires comme les pièces d'un puzzle.

Un sprite peut avoir un ou plusieurs programmes. Un autre mot pour un programme Scratch est **Script**.

Un double clic sur ton programme permet d'exécuter les blocs l'un après l'autre, du haut vers le bas.

Si tu n'as plus besoin d'un bloc d'instruction, il te suffit de le faire glisser de nouveau dans une palette d'instructions.

Exercice A-01 Mon premier programme

Objectif

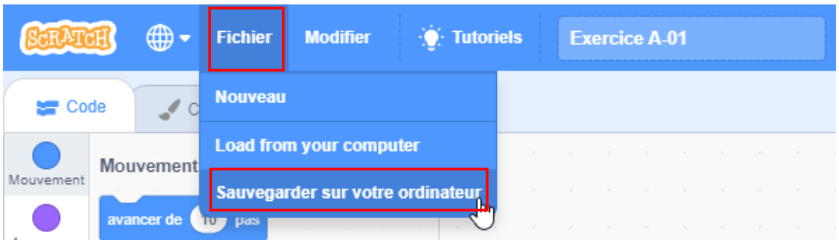
Développer un programme Scratch.

Préparation

a) Lance Scratch.

b) Renomme le projet de ***Projet Scratch*** vers ***Exercice A-01***.

c) Sauvegarde le programme sur ton ordinateur.



Description des tâches

d) Renomme le sprite en lui donnant le nom ***Chat***.

e) Fais glisser le bloc 

vers l'espace de programmation. Effectue plusieurs clics sur ce bloc. Qu'est-ce qui se passe ? Observe la position du sprite sur la scène ! Explique !

f) Développe le programme suivant :

g) Lance le programme par un double clic.

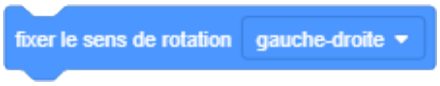
h) Arrête l'exécution du programme à l'aide du bouton suivant :  .

i) Y a-t-il un autre moyen de lancer le programme?

_____

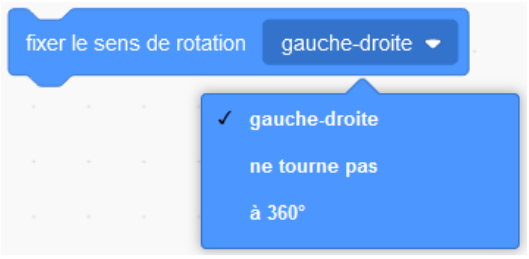
_____

j) Modifie le style de rotation du sprite et expérimente avec les différentes possibilités.

Ajoute à cet effet le bloc 

comme premier bloc après 

. Explique les différents styles de rotation :



_____

Lequel des styles de rotation est le plus adapté pour ce projet ? _____

_____

_____

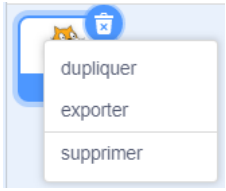
Sauvegarder le projet

k) Sauvegarde à nouveau le projet sur ton ordinateur (***Exercice A-01)***. Confirme si le fichier existant doit être remplacé pour prendre en compte d'éventuelles modifications ou pas.

A.3. Modifier les sprites

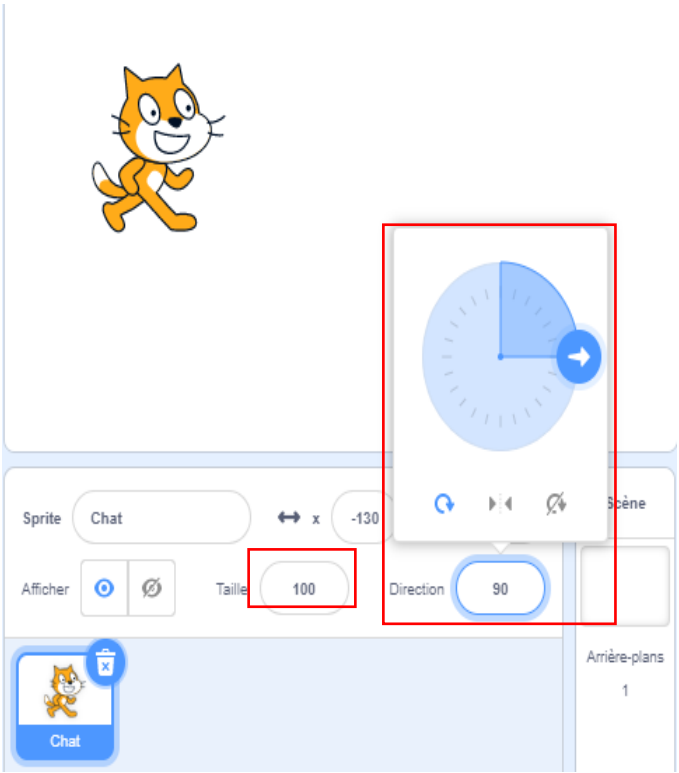
Pour **déplacer** un sprite à un autre endroit de la scène, il suffit de le glisser avec la souris en tenant le bouton gauche de la souris enfoncé.

Tu peux **dupliquer** ou **supprimer** un sprite en cliquant avec le bouton droit de la souris sur le sprite souhaité dans la liste des sprites.



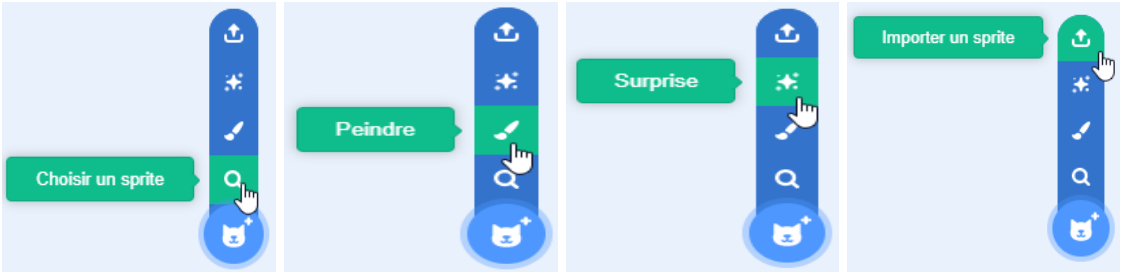
Tu peux également **agrandir** et **réduire** la taille d'un sprite. La taille est donnée en pourcentage. 100 représente la taille réelle d'un sprite. Une valeur inférieure à 100 permet de réduire le sprite et une valeur supérieure à 100 permet de l'agrandir.

Pour faire pivoter la figure, tu peux cliquer sur le champ de direction et changer la direction du sprite en bougeant la flèche.



Direction	Valeur de la direction (en dégrés)
vers le haut	0
à droite	90
vers le bas	180
à gauche	-90

Il existe plusieurs possibilités pour **créer** des sprites supplémentaires dans ton projet :



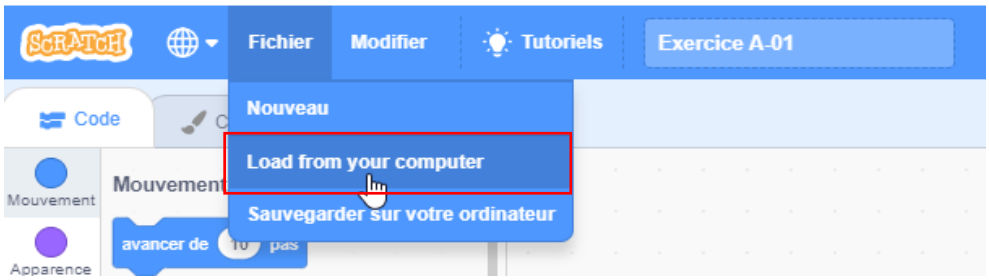
Exercice A-02 Créer d'autres sprites

Objectif

Créer de nouveaux sprites dans un projet et les modifier.

Préparation

a) Ouvre le projet de l'***Exercice A-01*** (*Load from your computer*)



b) Renomme le projet ***Exercice A-02***.

c) Sauvegarde le projet sur ton ordinateur sous le nom ***Exercice A-02***.

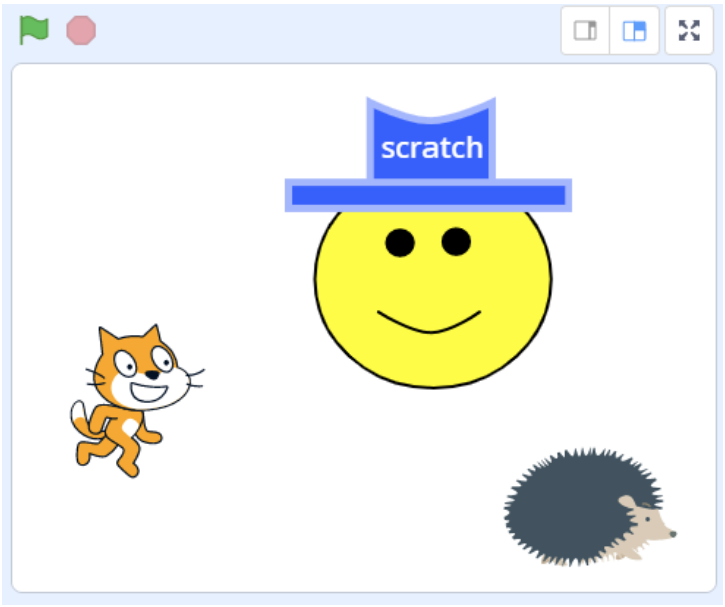
Description des tâches

d) Crée de nouveaux sprites et place-les sur la scène comme indiqué sur le schéma ci-dessous. Le hérisson (*Ang. : Hedgehog)* peut être choisi de la bibliothèque, le smiley doit être dessiné dans l'éditeur graphique.

e) Nomme les nouveaux sprites de manière appropriée.

Sauvegarder le projet

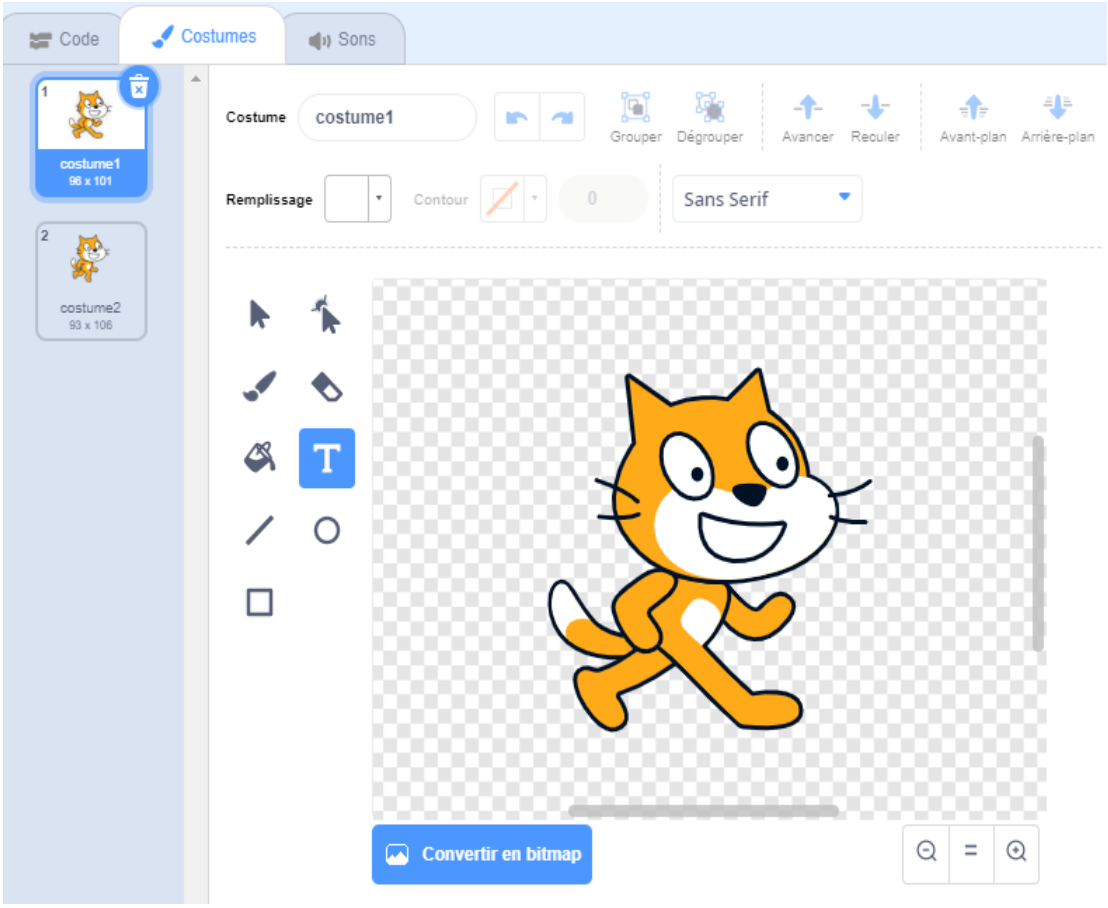
f) Sauvegarde de nouveau le projet sur l'ordinateur sous le nom ***Exercice A-02***.



A.4. Les sprites ont des costumes

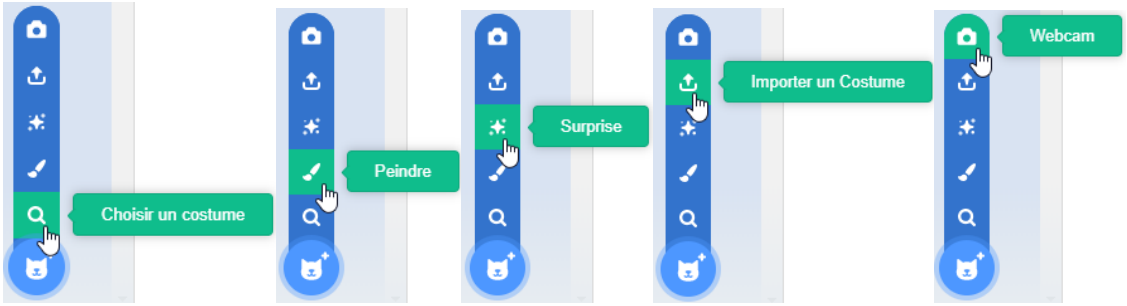
Les sprites peuvent avoir un ou plusieurs costumes (habits) et ainsi changer leur apparence.

Tu peux voir les costumes d'un sprite quand tu sélectionnes d'abord le sprite et tu cliques ensuite sur l'onglet intitulé ***Costumes***.



Le chat a p. ex. deux costumes. L'habit actuel est mis en évidence. Pour basculer vers un autre costume, il suffit de cliquer sur l'image d'aperçu du costume à gauche de la page.

Il y a cinq possibilités pour créer un nouveau costume :



Scratch est capable de traiter de nombreux formats graphiques : JPG, GIF, BMP et PNG.

Tu peux modifier l'ordre des costumes en les déplaçant à l'aide de la souris.

Les graphiques vectoriels

À gauche dans l'image se trouve une barre d'outils qui sert à éditer le costume actuel en mode vectoriel.

Les **graphiques vectoriels** sont composés de formes géométriques. Leur affichage est plus net que celui des graphiques bitmaps et ils peuvent être agrandis sans perte de qualité.

Les graphiques bitmap

Si tu transformes le costume en graphique bitmap, alors tu trouves une autre barre d'outils à droite, celle qui sert à retravailler les graphiques bitmap.

Les **graphiques bitmap** sont composés de pixels (points de couleur). Ils sont bien plus adapté à l'affichage d'images complexes comme les photos.

Exercice A-03 Les costumes

Objectif

Modifier des costumes dans un projet.

Préparation

a) Ouvre le projet de l'***Exercice A-01***.

b) Renomme le projet en ***Exercice A-03***.

c) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice A-03***.

Description des tâches

d) Charge le sprite *Bear-walking* et place-le sur la scène comme indiqué sur le schéma ci-dessous.

e) Nomme le sprite ***Ours***.

f) Combien de costumes l'ours possède-t-il ?

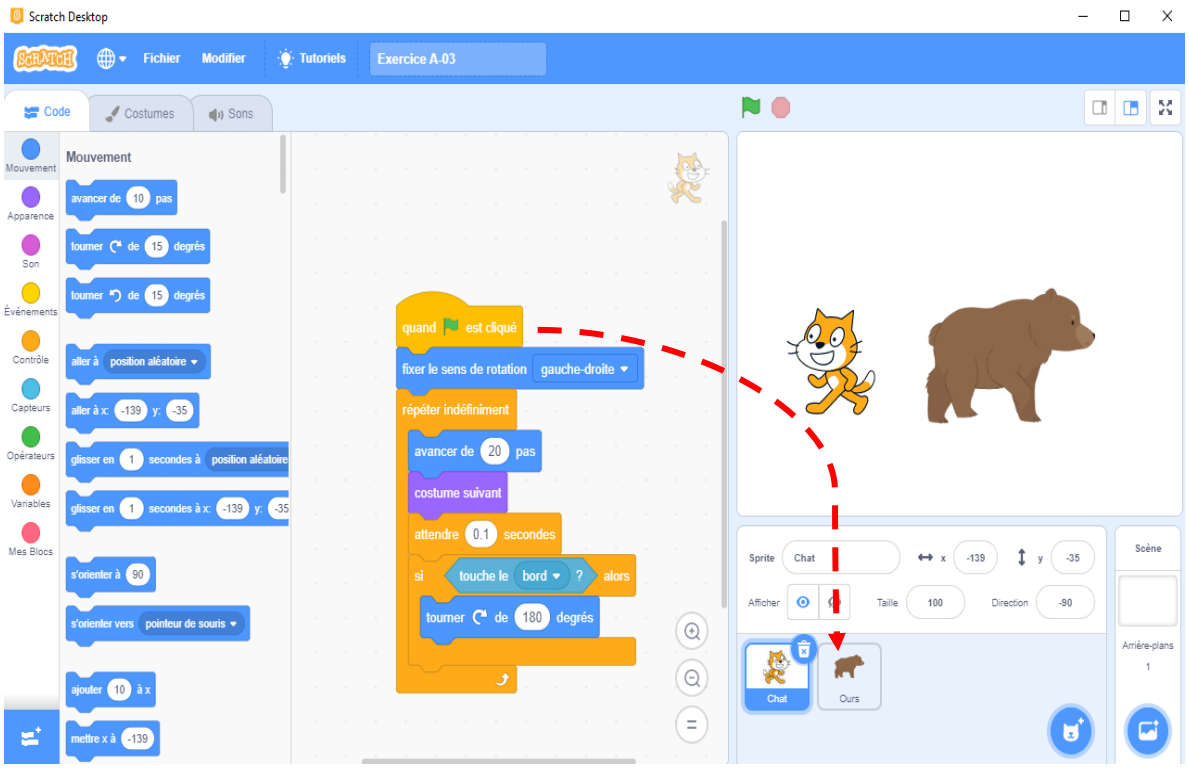
---

g) Copie le programme du sprite ***Chat*** dans le sprite ***Ours*** (tire le programme de l'espace de programmation vers le sprite ***Ours*** dans la liste des sprites comme indiqué sur le schéma).

h) Lance maintenant les deux programmes (clic sur drapeau vert).

Sauvegarder le projet

i) Sauvegarde à nouveau le projet sur ton ordinateur sous le nom ***Exercice A-03***.



A.5. La scène

De la même manière que les sprites changent d'apparence en basculant vers un autre costume, la scène peut changer d'apparence en basculant vers un autre **arrière-plan**.

Il est en plus possible de programmer la scène.

Pour modifier l'arrière-plan de la scène, clique sur l'icône scène qui se trouve à droite de la liste des sprites.

Exercice A-04 La scène – les arrière-plans

Objectif

Modifier l'arrière-plan de la scène.

Préparation

a) Ouvre le projet de l'***Exercice A-03***.

b) Nomme-le ***Exercice A-04***.

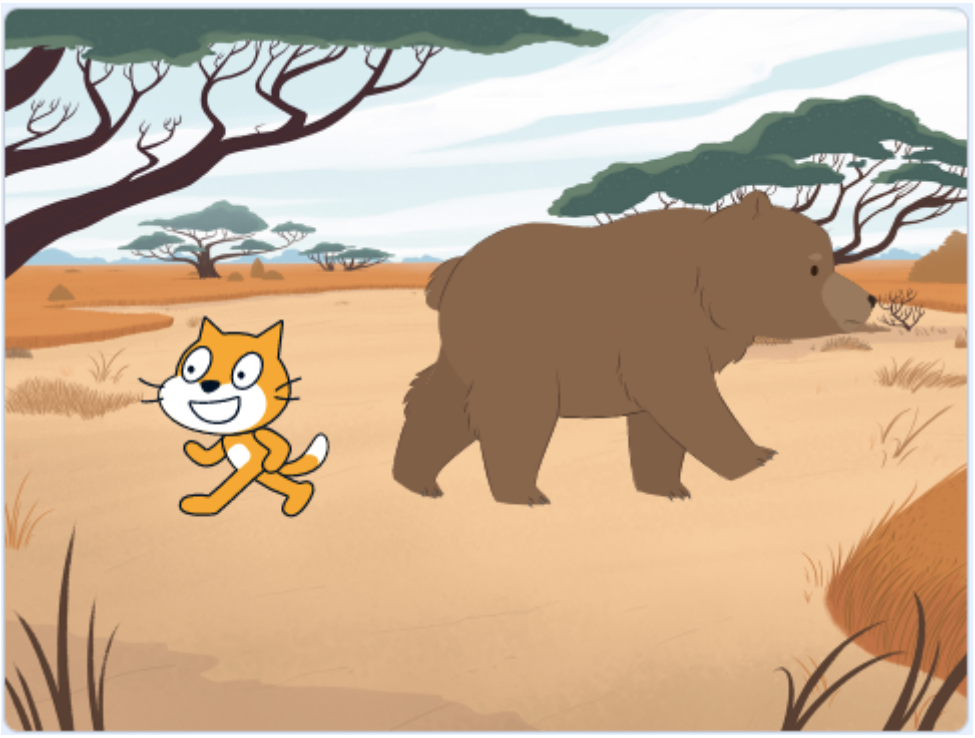
c) Sauvegarde-le sur ton ordinateur sous le nom ***Exercice A-04***.

Description des tâches

d) Modifie l'arrière-plan comme affiché ci-dessous.

Sauvegarder le projet

e) Sauvegarde à nouveau le projet sur ton ordinateur sous le nom ***Exercice A-04***.



A.6. Documentation du projet

Il est toujours conseillé de documenter son projet de manière suffisante. Ainsi quand tu voudras le modifier plus tard, une bonne documentation t'aidera à mieux t'y retrouver.

Les **Commentaires** te permettent d'annoter certaines parties de tes programmes. Cette documentation s'impose surtout pour des programmes complexes dans lesquels il est plus difficile de se retrouver.

Clique avec le bouton droit de la souris dans l'espace de programmation, choisis **Ajouter un commentaire** dans le menu contextuel, puis saisis le commentaire.

Exercice A-05 La documentation du projet

Objectif

Documenter un projet

Préparation

a) Ouvre le projet de l'***Exercice A-04*** et nomme-le ***Exercice A-05***.

b) Sauvegarde-le sur ton ordinateur sous le nom ***Exercice A-05***.

Description des tâches

c) Essaie de comprendre comment fonctionne le programme du sprite ***Chat*** et saisis un commentaire pour chaque bloc d'instruction.

Sauvegarder le projet

d) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice A-05***.

B. Développer un projet à l'aide de séquences

La séquence est la forme la plus simple pour la structure d'un programme. Plusieurs instructions sont exécutées l'une après l'autre.

Voici un exemple d'une séquence Scratch :

Exercice B-01

Objectif

Réaliser un projet à l'aide de séquences.

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice B-01***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice B-01***.

Description des tâches

c) Le chat doit exécuter les instructions suivantes quand le drapeau vert est cliqué:

- se rendre à la position x=0, y= -100,
- fixer le style de rotation à „gauche-droite“,
- regarder à gauche,
- dire "Bonjour" pendant 2 secondes.

Sauvegarder le projet

d) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice B-01***.

Exercice B-02 "Le corbeau et le renard !"

Objectif

Créer un projet à l'aide d'une séquence d'instructions.

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice B-02***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice B-02***.

c) Ajoute les sprites suivants au projet : *Fromage*, *Corbeau* et *Renard*.

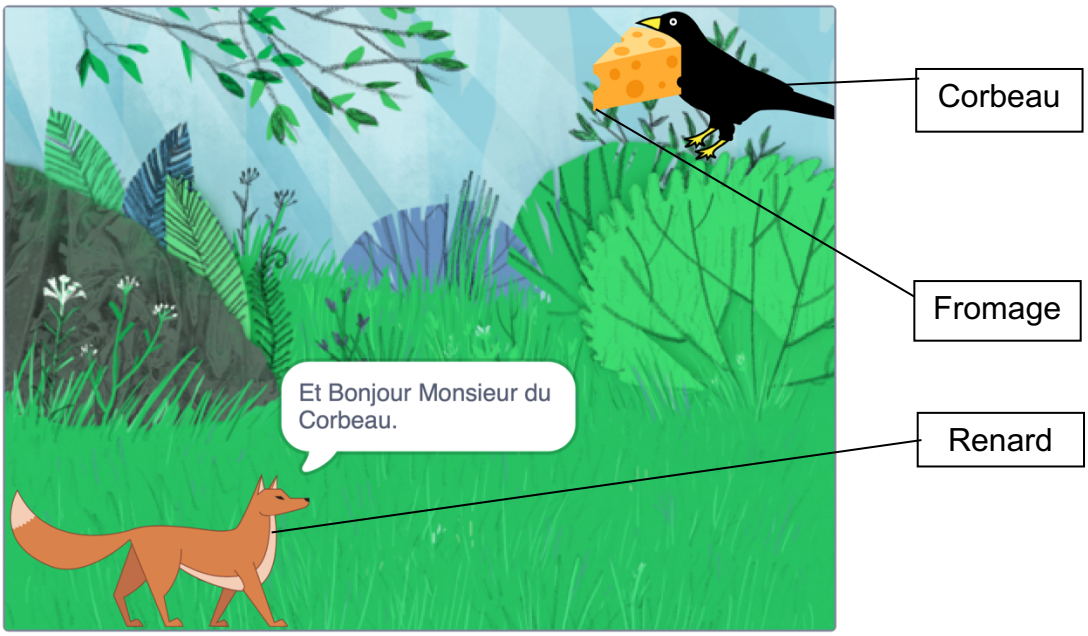
Le fromage et le corbeau ne sont pas présents dans la bibliothèque. Utilise donc les sprites que votre professeur a mis à ta disposition (*corbeau.png* et *fromage.png*) ou bien télécharge-les d'Internet.

Conditions de départ

d) Quand on clique sur le drapeau vert, le renard se trouve en bas à gauche de l'écran, le corbeau se trouve dans le coin supérieur droit de l'écran et le fromage dans sa propre position de départ, c'est-à-dire dans le bec du corbeau.

Utilise le bloc **aller à** pour tous les sprites au début afin de les placer dans leurs positions respectives. On parle dans ce cas d'**initialisation**. Cette initialisation sera désormais spécifiée par défaut dans tous les exercices suivants.

Description des tâches



e) Le renard doit réciter <u>phrase par phrase</u> (l'une après l'autre) le texte suivant (un bloc par ligne de phrase) :

Et Bonjour Monsieur du Corbeau,
Que vous êtes joli ! Que vous me semblez beau !
Sans mentir, si votre ramage se rapporte à votre plumage
Vous êtes le phénix des hôtes de ces bois

f) Le **Corbeau** dit ensuite le texte « Merci », après quoi le **Fromage** tombe par terre.

Remarques :

- Réfléchis à l'avance sur la façon de synchroniser l'exécution des actions des 3 sprites dans un ordre chronologique.
- Le fromage tombe mieux avec un mouvement utilisant le bloc « ***glisser en***… ».

Sauvegarder le projet

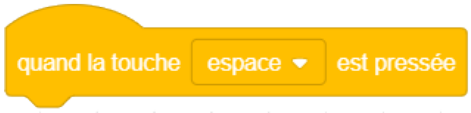
g) Sauvegarde à nouveau le projet sur ton ordinateur sous le nom ***Exercice B-02***.

C. Réagir à des événements

Les événements sont symbolisés par des blocs appelés « *chapeaux* ». Ces blocs possèdent une bordure supérieure arrondie comme p.ex. :

Le programme accolé est exécuté,
quand on clique sur le drapeau vert

Le programme accolé est exécuté,
quand on clique sur le sprite sélectionné.



Le programme accolé est exécuté
quand on appuie sur la touche indiquée.

Un tel bloc forme le sommet d'une pile de blocs assemblés. **Il attend qu'un certain événement ait lieu** (p.ex. on appuie sur une touche indiquée). Lors de cet événement les blocs en-dessous du bloc chapeau sont exécutés.

Exercice C-01

Objectif

Réagir au clic sur le drapeau vert et au clic de la souris.

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice C-01***.

b) Sauvegarde-le sous sur ton ordinateur sous le nom ***Exercice C-01***.

Description des tâches

c) Ajoute au projet un ballon. Crée au moins deux autres costumes pour représenter l'éclatement du ballon.

d) Quand le drapeau vert est cliqué, le ballon gonflé se trouve au milieu de la scène.

Tu peux choisir toi-même l'arrière-plan de la scène.

e) Quand tu cliques sur le ballon gonflé avec la souris, il éclate.

Sauvegarder le projet

f) Sauvegarde à nouveau le projet sur ton ordinateur sous le nom ***Exercice C-01***.

Exercice C-02 Extension pour avancés

Objectif

Réagir aux événements de clavier.

Préparation

a) Ouvre le projet ***Exercice C-01*** et renomme-le ***Exercice C-02***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice C-02***.

Description des tâches

c) Lorsque la touche „k" est pressée, le ballon est cloné (copié). Utilise le bloc

d) Après le clonage, le ballon doit glisser vers une position aléatoire sur la scène et changer de couleur.

Quel <u>événement</u> te permet de réaliser cela ?

Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice C-02***.

Exercice C-03

Objectif

Réagir aux événements de clavier

Préparation

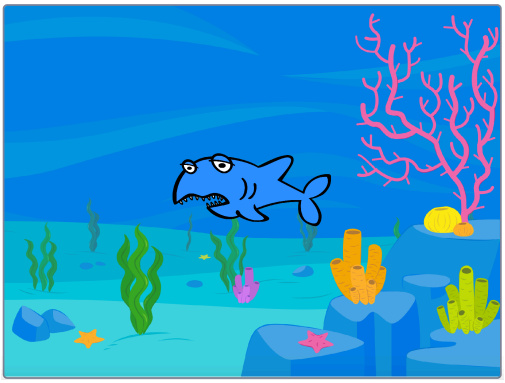
a) Crée un nouveau projet et nomme ***Exercice C-03***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice C-03***.

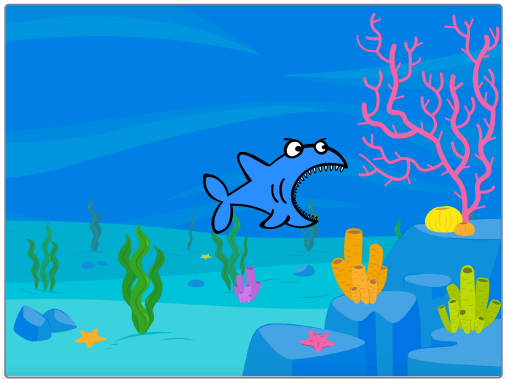
Description des tâches

c) Crée un projet avec le requin en haute mer. Quand le drapeau vert est cliqué, le requin se trouve au milieu de la scène et sa taille est de 100%.

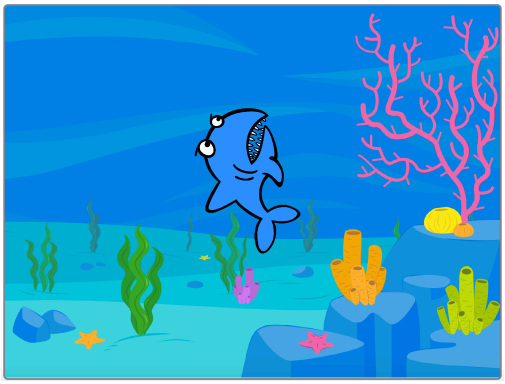
d) À l'aide des quatre touches flèche, tu peux déplacer le requin dans l'eau dans une des quatre directions. Chaque fois qu'une des flèches est pressée, le requin effectue un mouvement de nage (pas) dans la direction de la flèche appuyée et change de costume. Sélectionne le style de rotation pour les différentes touches fléchées afin que le requin tourne dans le sens correspondant sans nager sur le dos (voir illustration).



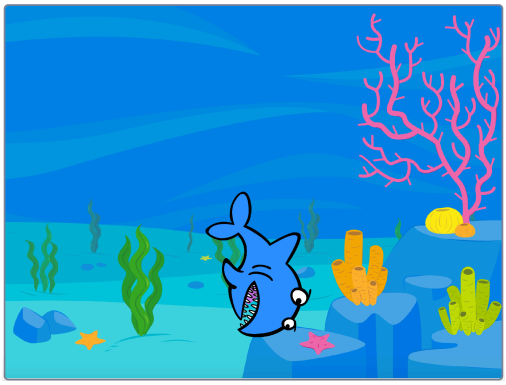
Touche ←



Touche →



Touche ↑



Touche ↓

Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice C-03***.

Exercice C-04 Extension pour avancés

Objectif

Réagir aux événements de clavier

Préparation

a) Ouvre le projet de l'***Exercice C-03*** et nomme-le ***Exercice C-04***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice C-04***.

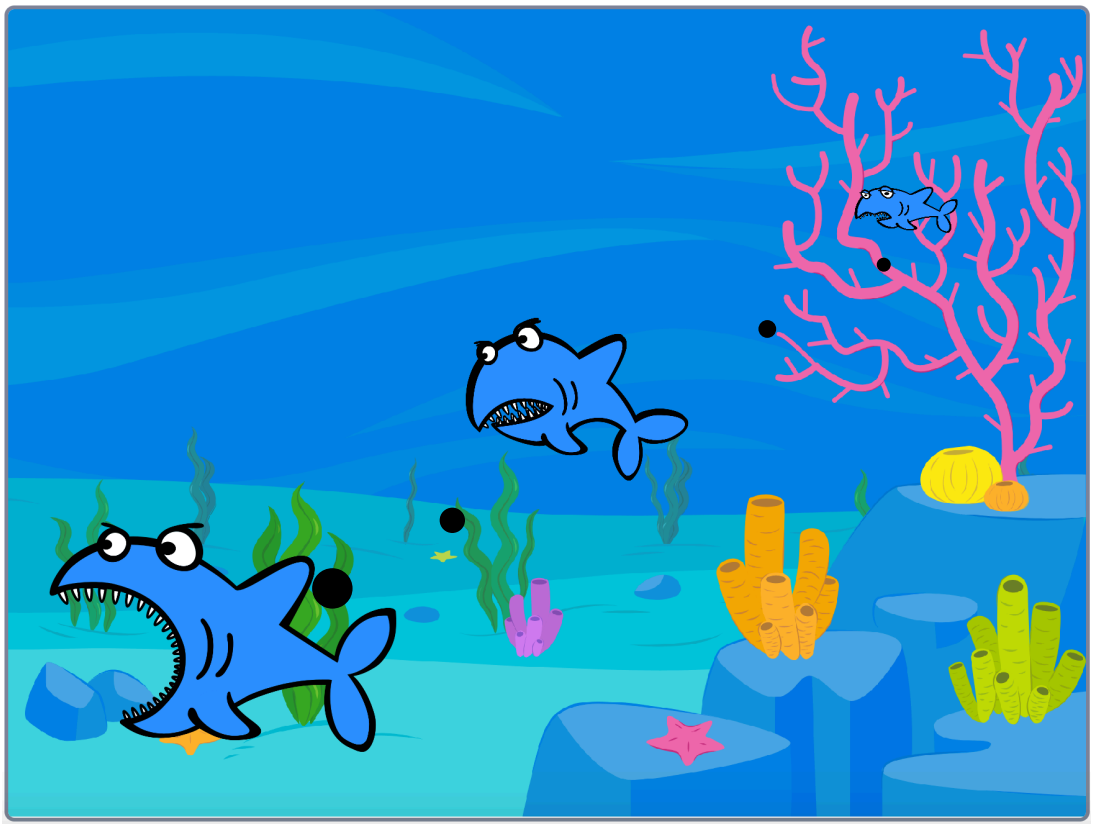
Description des tâches

c) Programme les touches „H“ et „V“

Touche „H“ : Chaque fois que la touche est appuyée, le requin nage (pas) dans la direction correspondante (en diagonale) vers le haut à droite et devient de plus en plus petit. On crée ainsi un effet de profondeur.

Touche „V“ : Chaque fois que la touche est appuyée, le requin nage (pas) dans la direction correspondante (en diagonale) vers le bas à gauche tout en devenant de plus en plus grand.

Utilise un bloc pour modifier automatiquement la taille du requin.



Sauvegarder le projet

d) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice C-04***.

D. La boucle infinie

Dans cette unité, nous aborderons le bloc de contrôle intitulé "**répéter indéfiniment**". Il nous permet de répéter des actions indéfiniment. En termes de programmation, nous parlons de "boucle infinie".

Si nous insérons les blocs d'instructions des exercices précédents dans cette boucle infinie, nous allons voir qu'ils seront exécutés jusqu'à ce qu'on clique sur l'icône stop rouge.

Exercice D-01 L'infatigable papillon

Objectif

Utiliser la boucle infinie.

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice D-01***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice D-01***.

Description des tâches

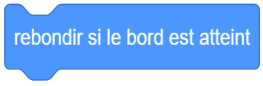
c) Crée un projet avec le sprite et la scène ci-dessous.

d) Conditions de départ :

Le papillon est placé au milieu de l'écran et regarde vers la droite.

Le style de rotation doit être réglé à "gauche-droite" à l'aide d'un bloc d'instruction.

e) Le papillon dispose de deux costumes qui permettent de le voir voler. Quand nous cliquons sur le drapeau vert, nous pouvons observer l'observer voler sur la scène en faisant des aller-retours.

Utiliser le bloc d'instruction suivant pour faire tourner le papillon lorsqu'il arrive sur le bord de la scène : 

Sauvegarder le projet

f) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice D-01***.

Exercice D-02 Une montre basique

Objectif

Utiliser la boucle infinie.

Préparation

a) Ouvre le projet ***Exercice D-02***, que ton enseignant met à ta disposition.

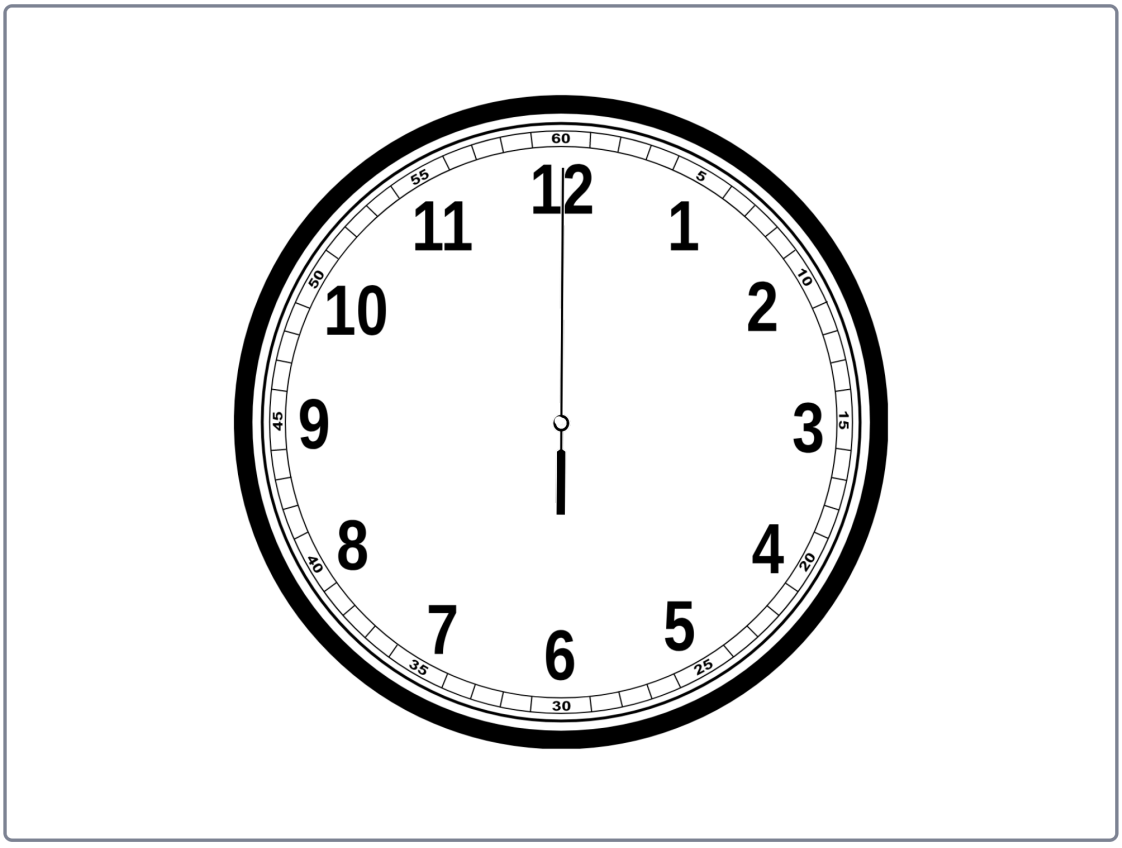
b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice D-02***.

Description des tâches :

c) Conditions de départ :

Le cadran et l'aiguille des secondes sont placés au milieu de l'écran et la taille est réglée à 200%.

L'aiguille des secondes est orientée vers le haut et pointe sur 12h.



d) Au démarrage du programme, l'aiguille des secondes est animée. Elle effectue une rotation complète (360°) en 60 secondes. L'angle de rotation est logiquement de 360 : 60 = 6°. L'aiguille n'arrête jamais de tourner dès que le drapeau vert est cliqué.

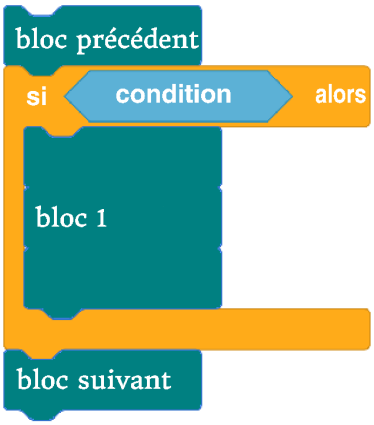
Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice D-02***.

E. Prendre des décisions

Tu as certainement remarqué qu'il est souvent nécessaire, lors de la programmation de prendre une décision (comme dans la vie réelle). Ainsi ton sprite devra faire une chose ou une autre selon les circonstances qu'il rencontre.

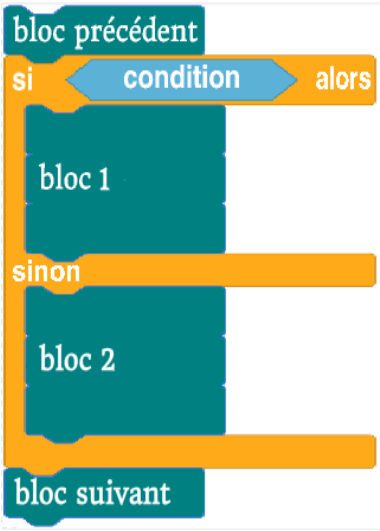
Voici les blocs utilisés pour réaliser ce choix :



si *la condition est vérifiée,*

alors *le bloc d'instructions* ***bloc 1*** *est exécuté.*

Ensuite, le **bloc suivant** le bloc **si** est exécuté.



si *la condition est vérifiée,*

alors *le bloc d'instructions* **bloc 1** *est exécuté,*

sinon *le bloc d'instructions* **bloc 2** *est exécuté.*

Ensuite, le **bloc suivant** le bloc **si** est exécuté.

Important : Le bloc d'instruction **si** est exécuté une seule fois. Afin de vérifier de manière continue une condition, tu dois intégrer cette condition dans une boucle.

La condition doit être "correcte" ou bien "vraie" pour faire exécuter les instructions de la partie "si".

Il existe plusieurs types de conditions, donc deux sont illustrés ci-dessous :

Ce bloc te permet de tester p. ex. si le bord ou un sprite est touché.

Ce bloc te permet de tester si une couleur est touchée.

Tu trouves ces deux blocs ci-dessus dans la palette « **Capteurs** ».

Exercice E-01 Les danseuses

Objectif

Prendre des décisions.

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice E-01***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice E-01***.

Description des tâches

Les deux danseuses Jouvi (à gauche) et LB (à droite) attendent patiemment d'exécuter leurs pas de danse. Dès qu'on déplace le pointeur de la souris sur leur sprite, ils commencent à danser et disent avec joie : « Maintenant, je danse ! ». Sinon, ils s'arrêtent et disent : « J'attends ».

Sabi a développé le programme ci-dessus pour la danseuse Jouvi.

c) Crée un nouveau projet avec deux sprites et un arrière-plan comme sur l'image ci-dessus.

d) Construis le programme de Sabi pour le sprite Jouvi et teste-le. Que constates-tu ?

e) Améliore le programme de Sabi afin qu'il fonctionne.

f) Construis le programme de la danseuse LB.

Sauvegarder le projet

g) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice E-01***.

Exercice E-02 Robot dans l'espace

Objectif

Prendre des décisions.

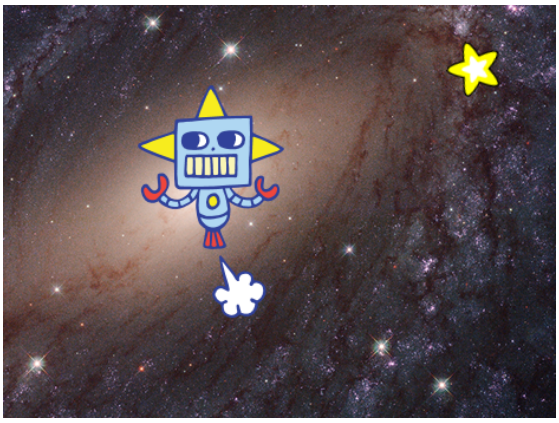
Préparation

a) Crée un nouveau projet et nomme-le ***Exercice E-02***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice E-02***.

Description des tâches

Il s'agit dans cet exercice d'un robot spatial qui veut attraper une étoile.



c) Crée un nouveau projet avec deux sprites et un arrière-plan comme sur la figure ci-dessus.

d) Conditions de départ :

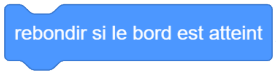
Robot (c'est aussi le nom du sprite) :

- se trouve en bas à gauche de l'écran,
- son style de rotation est « gauche-droite »,
- il se trouve en avant-plan sur l'écran.

Etoile (c'est aussi le nom du sprite) :

- il se trouve en haut à droite de l'écran.

e) Quand le drapeau vert est cliqué, le robot essaie sans arrêt d'atteindre l'étoile en glissant vers une position aléatoire tout en restant complètement sur scène.

Utilise à cet effet le bloc d'instruction 

. S'il attrape l'étoile, il dit « Mon étoile » pendant une seconde.

f) L'étoile se déplace vers une nouvelle position aléatoire toutes les 2 secondes tout en restant toujours visible à l'écran.

Sauvegarder le projet

g) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice E-02***.

Exercice E-03 Récolte de pommes

Objectif

Prendre des décisions.

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice E-03***.

b) Sauvegarde-le sur ton ordinateur sous le nom de *Exercice E-03*.

Description des tâches

Il s'agit dans cet exercice d'attraper avec un bol des pommes tombant d'un arbre.

c) Crée un nouveau projet avec deux sprites et un arrière-plan comme sur la figure ci-dessus.

d) Conditions de départ :

Bol (c'est aussi le nom du sprite) :

- se trouve en bas au milieu de l'écran,
- son style de rotation est « ne tourne pas »,

Pomme (c'est aussi le nom du sprite) :

- il se trouve en haut à une position aléatoire comprise entre les abscisses x=-200 et x=+200. Tu trouves dans la palette « Opérateurs » le bloc d'instruction 

qui te permet de générer un nombre aléatoire.
- son style de rotation est « ne tourne pas ».

e) Le bol réagit aux touches flèche de droite et de gauche en restant toujours visible à l'écran.

f) La pomme tombe par pas de 5. Si elle touche le bord inférieur, elle se rend en haut selon les conditions de départ et dit « Manqué ! » pendant 0.5 seconde. Si elle touche le bol, alors elle dit « Attrapé ! » pendant 0.5 seconde et se rend à nouveau en haut selon les conditions de départ.

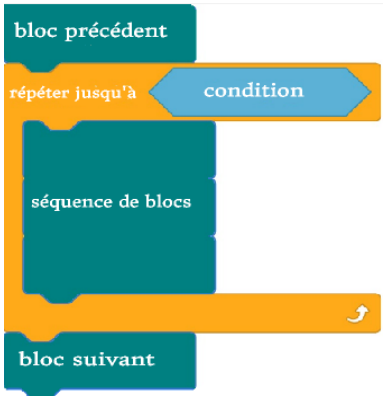
Sauvegarder le projet

g) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de *Exercice E-03*.

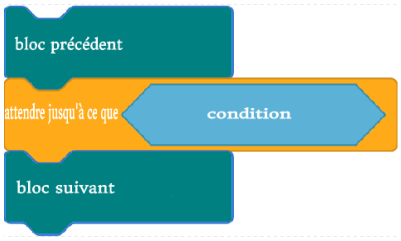
F. Boucles à condition

Jusqu'à maintenant, tu t'es déjà beaucoup exercé en Scratch et tu as appris beaucoup de choses. Tu sais programmer des séquences d'instructions, de même qu'utiliser les boucles infinies et les conditions.

Savais-tu qu'on pouvait parfois combiner les deux : boucles infinies et conditions ? En le faisant tu obtiens ceci :



Dans ce cas, le programme regarde d'abord si la condition est vérifiée. Si oui, le contenu de la boucle est ignoré et le programme continue avec l'instruction qui suit la boucle. Sinon, elle exécute les instructions de la séquence intérieure de la boucle (séquence de blocs), puis contrôle à nouveau si la condition est vérifiée. la séquence de blocs est exécutée jusqu'à ce que la condition soit vérifiée. Alors l'exécution se poursuit avec l'instruction qui suit la boucle (bloc suivant).



Avec le bloc d'instruction **attendre jusqu'à**, la condition est testée. Si la condition est vraie, alors le bloc suivant est exécuté. Dans le cas contraire, la condition est testée de manière répétée jusqu'à ce qu'elle soit vraie. Ce n'est qu'à ce moment que l'instruction suivant le bloc attendre (bloc suivant) est exécuté.

Exercice F-01 Jogging sur la plage

Objectif

Utiliser des boucles à condition.

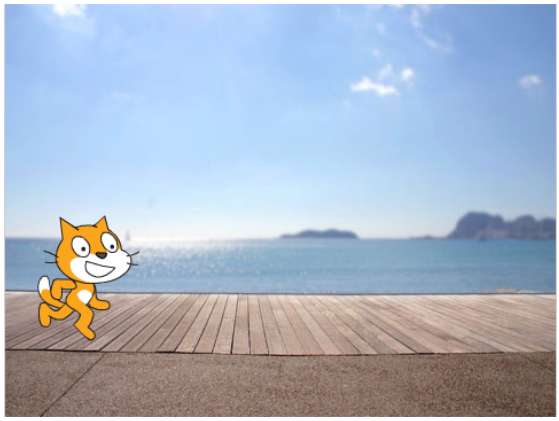
Préparation

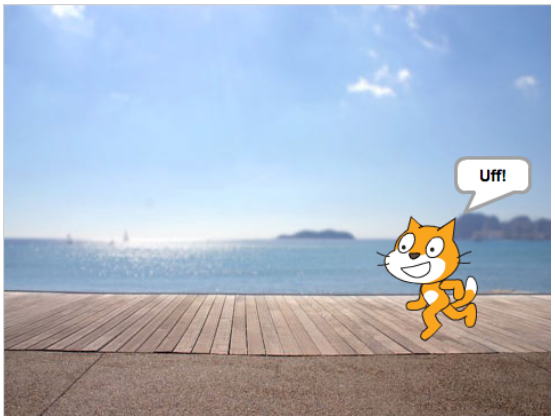
a) Crée un nouveau projet et nomme-le ***Exercice F-01***.

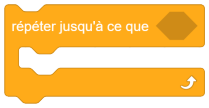
b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice F-01***.

Description des tâches

Le chat fait son jogging par des allers-retours jusqu'à ce que la touche espace soit appuyée.





Bloc autorisé :	
Bloc interdit :	

c) Crée un projet avec le chat et l'arrière-plan de la scène ci-dessus.

d) Conditions de départ : le style de rotation du chat est « gauche-droite ».

e) Le chat fait son jogging par des allers-retours jusqu'à ce que la touche espace soit appuyée. Elle s'arrête ensuite et dit "Ouf !". Utilise la boucle « **répéter jusqu'à ce que** » !

Sauvegarder le projet

f) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice F-01***.

Exercice F-02 Jogging sur la plage

Objectif

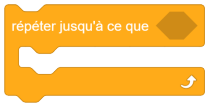
Utiliser des boucles à condition.

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice F-02***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice F-02***.

Description des tâches :

Bloc autorisé :	
Bloc interdit :	

c) Essaye de programmer l'exercice précédent (Exercice F-01) en n'utilisant que la **boucle infinie.**

d) Quelle solution est la meilleure ? Justifie ta réponse !

Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice F-02***.

Exercice F-03 Libérer le crabe !

Objectif

Utiliser des boucles à condition.

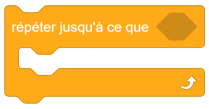
Préparation

a) Crée un nouveau projet et nomme-le ***Exercice F-03***.

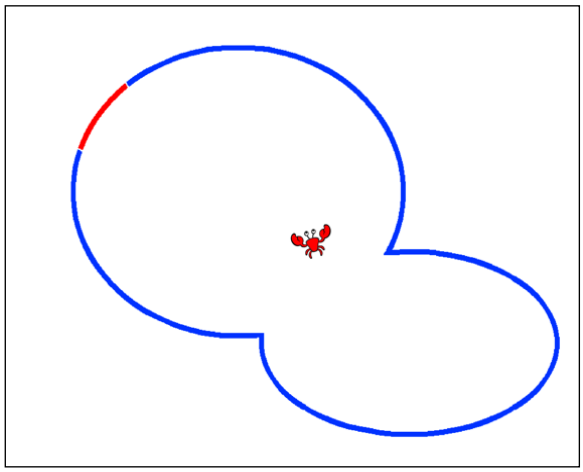
b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice F-03***.

Description des tâches

Le crabe doit se libérer de sa cage.

Blocs autorisés :	
Blocs interdits :	

c) Utilise le sprite ***Crabe*** de la collection Scratch. Dessine pour la scène un arrière-plan ("cage") qui possède un bord de couleur bleue. La sortie est représentée par une deuxième couleur comme sur l'image l'exemple ci-après :



d) Conditions de départ :

Crabe (c'est aussi le nom du sprite) :

- Il se trouve au milieu de l'écran.
- Il pointe vers une direction fixée à l'aide d'un nombre aléatoire.

e) Le crabe essaie de s'échapper de sa cage. Quand il rencontre la couleur (bleue) du bord il pense « Hmmm… », se tourne un peu (l'ange de rotation à utiliser dépend du dessin de ta cage) et essaie à nouveau. Quand il rencontre la couleur (rouge) de la sortie, il s'écrie « enfin libre » et le programme s'arrête.

Sauvegarder le projet

f) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice F-03***.

Exercice F-04 Libérer le crabe – Extension (pour avancés)

Objectif

Utiliser des boucles à condition.

Préparation

a) Ouvre le projet de l'***Exercice F-03*** et nomme-le ***Exercice F-04***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice F-04***.

Description des tâches

c) Les 2 costumes du crabe ne se distinguent que peu. Dessine un costume plus rigolo, par exemple avec des yeux qui sortent.

d) Tu veux tricher et influencer la direction.

Tu peux faire tourner le crabe vers sa droite en appuyant sur la « **flèche droite »**.

La « **flèche gauche »** te permet de le faire tourner vers sa gauche.

Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice F-04***.

Exercice F-05 Cherche le repas !

Objectif

Utiliser des boucles à condition.

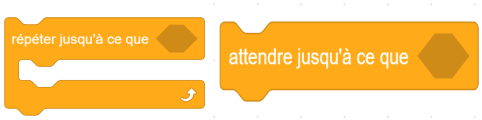
Préparation

a) Crée un nouveau projet et nomme-le ***Exercice F-05***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice F-05***.

Description des tâches

Le chat cherche le repas.

Blocs autorisés :	
Blocs interdits :	

c) Crée un nouveau projet avec deux sprites et un arrière-plan comme sur l'image ci-dessus.

d) Condition de départ :

Chat (c'est aussi le nom du sprite) :

- Il se trouve en bas à gauche de l'écran,
- Il est orienté vers la droite,
- son style de rotation est « gauche-droite ».

Repas (c'est aussi le nom du sprite) :

- il se trouve en bas à droite de l'écran,
- sa taille est fixée à l'aide d'un bloc à la moitié de sa taille d'origine,
- le repas est caché (invisible),
- le style de rotation est « ne tourne pas ».

e) Le chat a très faim, (il est vorace). S'il touche le bord, il pense « Mais où est passé mon repas ? » pendant 0,5 seconde, se tourne et avance de 10 pas. Sinon, il fait des aller-retours en faisant des pas de 10.

f) Le repas apparaît quand la touche espace est pressée et glisse vers une position qui se trouve 100 pixels plus à gauche de sa position de départ.

g) Quand le chat trouve son repas le chat dit « Le voilà enfin ! » et le programme s'arrête.

h) Lorsque le chat a trouvé le repas, son costume (***Repas-plein***) sera remplacé par un deuxième costume (***Repas-vide***), que tu auras dessiné au préalable.

Sauvegarder le projet

i) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice F-05***.

Exercice F-06 Liberté pour le chat

Objectif

Utiliser des boucles à condition.

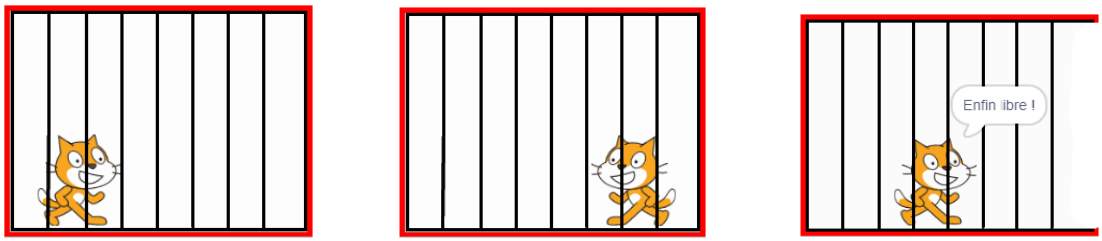
Préparation

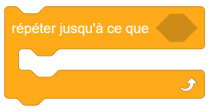
a) Crée un nouveau projet et nomme-le ***Exercice F-06***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice F-06***.

Description des tâches

Le chat veut être libéré de sa cage.



Blocs autorisés :	
Blocs interdits :	

c) Crée un nouveau projet avec deux sprites. Tu dois toi-même dessiner le sprite de la cage en utilisant 2 costumes. **La cage ne doit pas être dessiné comme arrière-plan** et ne doit pas occuper toute la largeur de l'écran.

d) Condition de départ :

Chat (c'est aussi le nom du sprite) :

- Il se trouve sur le côté gauche de la cage,
- il est orienté vers la droite,
- son style de rotation est « gauche-droite ».

Cage (c'est aussi le nom du sprite) :

- il se trouve en avant-plan sur l'écran,
- il bascule sur le costume avec la cage fermée.

e) Le chat court de gauche à droite et de droite à gauche (c.-à-d. il court tant qu'il touche la prison). Il change de direction en touchant le bord de couleur rouge de la cage.

f) Quand on appuie sur la touche 'espace', la cage s'ouvre et le chat te remercie pour sa liberté.

Sauvegarder le projet

g) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice F-06***.

Exercice F-07 Liberté pour le chat – Extension (avancés)

Objectif

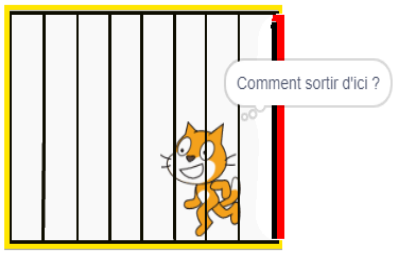
Utiliser des boucles à condition.

Préparation

a) Ouvre le projet de l'***Exercice F-06*** et nomme-le ***Exercice F-07***.

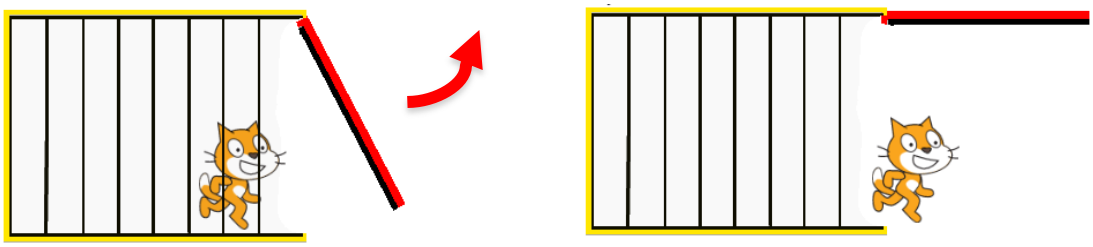
b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice F-07***.

Description des tâches



c) Pendant que le chat court dans sa cage il pense toutes les 10 secondes "Comment sortir d'ici ?"

d) La bordure de la cage change constamment du rouge au jaune (cage électrifiée). Le chat doit encore se retourner sur la paroi de la cage.



e) Quand la touche espace est appuyée, la porte de la cage s'ouvre lentement.

f) Quand le chat a quitté sa cage il te remercie pour sa liberté.

Aide : Pour résoudre cette extension tu peux utiliser les conditions complexes (voir chapitre I.2.2).

Sauvegarder le projet

g) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice F-07***.

G. Envoyer et recevoir des messages

Dans certaines situations, un sprite aimerait faire exécuter une action (programme) à un autre sprite.

Scratch le rend possible par l'envoi et la réception de messages.

Il existe deux instructions qui servent à envoyer un message :

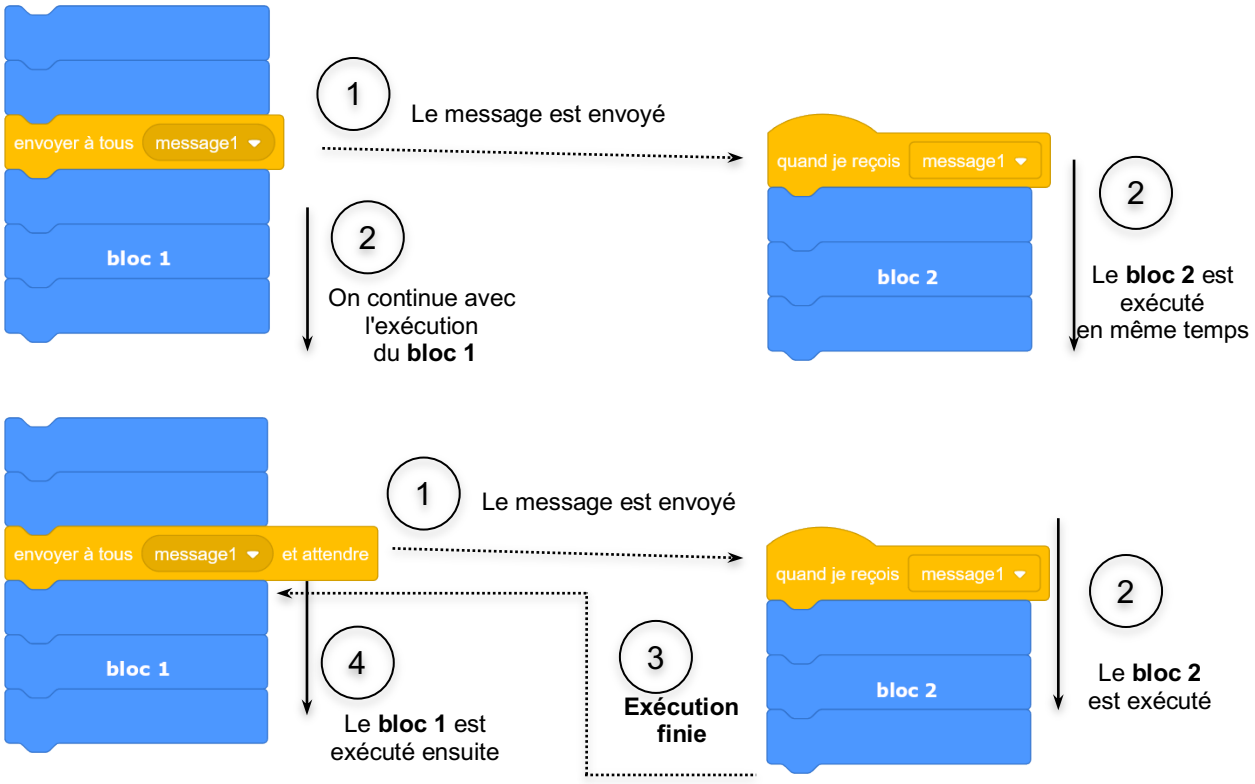
Envoie un message à tous les sprites et la scène, ce qui peut les amener à exécuter une action. Le programme continue tout de suite son exécution sans attendre que les autres sprites aient fini leurs actions.

Envoie un message à tous les sprites et la scène, ce qui peut les amener à exécuter une action. Le programme attend de continuer son exécution jusqu'à ce que les autres sprites aient fini leurs actions.

Il existe une instruction pour la réception d'un message :

Lorsqu'un un sprite ou la scène reçoit le message alors la séquence d'instructions qui se trouve sous le bloc chapeau est exécutée.

Les schémas suivants illustrent la différence entre les deux instructions d'envoi :



Exercice G-01 « Vite sur le tapis ! »

Objectif

Envoyer et recevoir des messages.

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice G-01***.

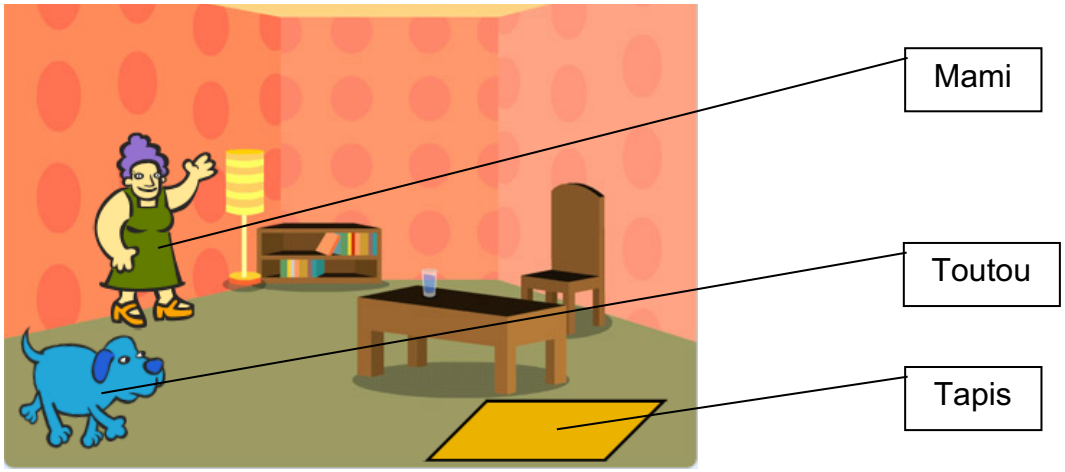
b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice G-01***.

c) Ajoute les *sprites* suivants à ton projet : *Mamie, Toutou* et *Tapis*. Le tapis n'étant pas inclus dans la bibliothèque, tu dois le dessiner toi-même.

Conditions de départ

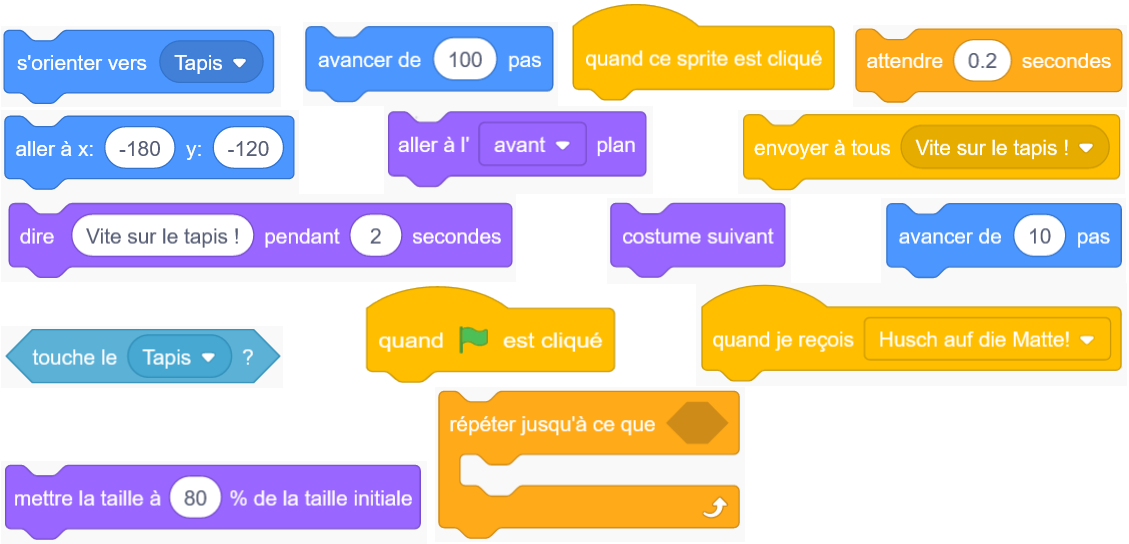
d) Toutou est positionné en bas à gauche sur l'écran, au premier plan et tourné vers le tapis.

Description des tâches



e) Quand on clique sur la Mamie, alors elle dit « Vite sur le tapis ! » et Toutou doit se rendre à sur le tapis.

Programme la ***Mamie*** et ***Toutou*** en utilisant les instructions suivantes :



f) Ajoute l'action suivante :
Quand Toutou arrive sur son tapis alors la mamie dit « Bon Toutou ! »

Sauvegarder le projet

g) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice G-01***

Exercice G-02 Une aiguille pour les minutes

Objectif

Envoyer et recevoir des messages.

Préparation

a) Ouvre le projet ***Exercice G-02*** mis à disposition par ton enseignant.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice G-02***.

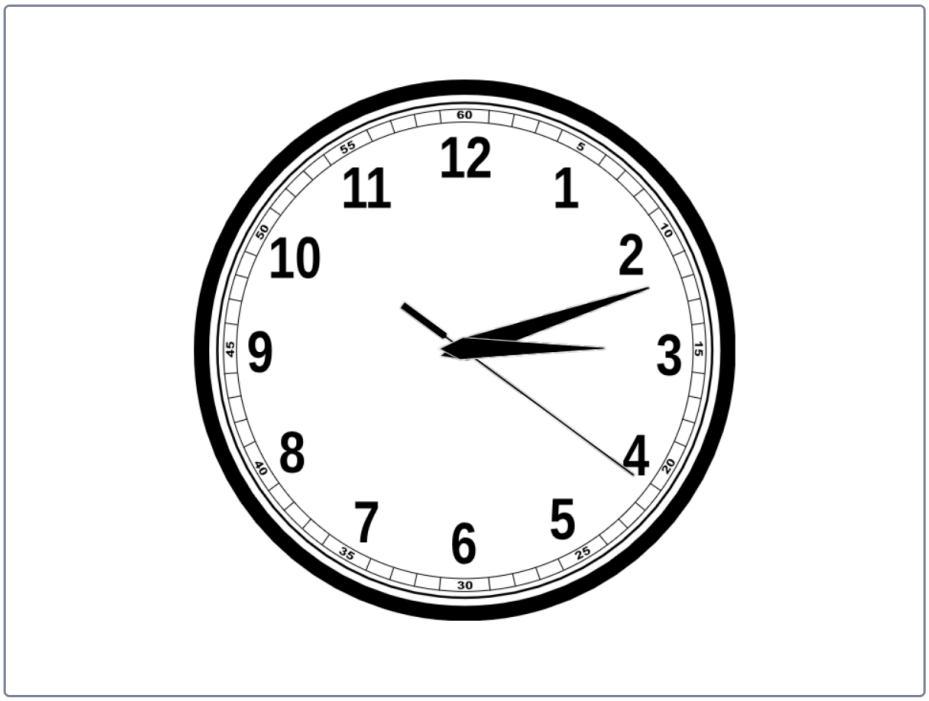
Conditions de départ

c) Les aiguilles pour les minutes et pour les heures pointent tous les deux vers le haut à 12 heures.

Description des tâches

d) Chaque fois que l'aiguille des secondes pointe sur le 12 elle envoie un message à l'aiguille des minutes pour que celle-ci avance d'un cran.

Chaque fois que l'aiguille des minutes pointe sur le 12 elle envoie un message à l'aiguille des heures pour que celle-ci avance d'un cran.



Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice G-02***.

H. Contrôle à l'aide de la boucle

Dans ce chapitre, tu apprendras comment tu peux contrôler l'exécution répétée d'instructions à l'aide d'une boucle.

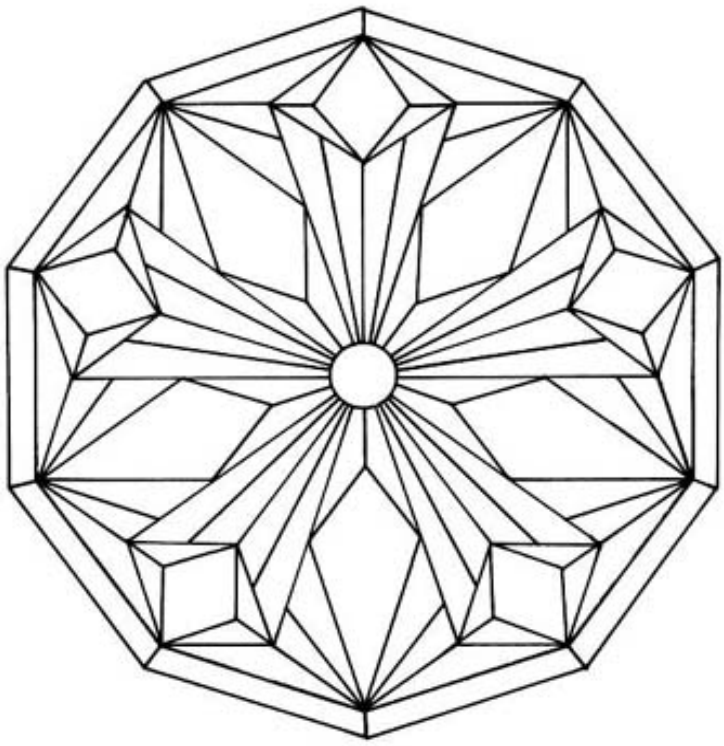
Pour réaliser cette unité, il est important de rafraîchir nos notions de géométrie (*calcul d'angles ; symétrie centrale ; rotation ; système de coordonnées, quatre quadrants*).

Mandala

Le mot **Mandala** (du sanscrit) veut dire 'cercle' et désigne une forme symbolique circulaire ou carrée avec un centre. À l'origine il a été utilisé uniquement dans un contexte religieux (Wikipédia).

Aujourd'hui le mandala sert plutôt pour les feuilles de coloriage pour enfants et adolescents.

Dans cette unité tu programmeras un mandala… plus simple que l'exemple ci-contre.

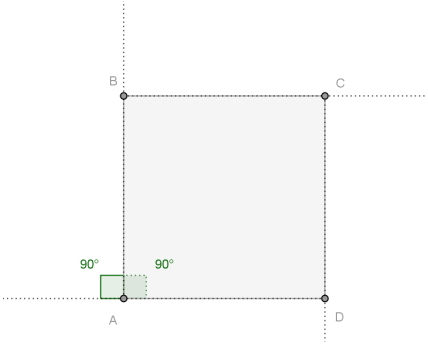


Le mandala ci-dessus est composé de parties égales (ici 5 ou plutôt 2x5) autour du centre.

Pour dessiner, vous avez besoin de la palette **Stylo**. Comme elle n'apparaît pas parmi les palettes par défaut, il faut l'activer avec la commande **Ajouter une extension**.

Tous les sprites peuvent utiliser le stylo et laisser des traces de peinture sur la scène. Le sprite doit allumer le stylo et se déplacer. La trace de la peinture se trouve toujours au point de pivot du sprite.

H.1. Dessiner un carré



Le carré est dessiné autour du centre de la scène (point 0;0).

Un carré possède …… côtés de …… …….. et …… angles de …… ……... La somme des angles d'un carré est de …… Un angle mesure exactement ……, il s'agit d'un angle ……

Exercice H-01 Un carré

Objectif

Dessiner avec le stylo

Préparation

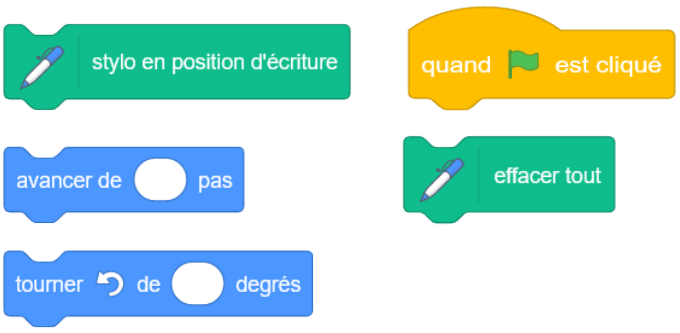
a) Crée un nouveau projet et nomme-le ***Exercice H-01***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-01***.

Description des tâches

c) Dessine un carré d'une longueur de 100. À la fin du script le sprite doit de nouveau pointer vers la direction de départ.

Blocs d'instruction utilisés :



d) Que remarques-tu ?

Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-01***.

Exercice H-02 Un carré 2

Objectif

Dessiner un carré à l'aide d'une boucle

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice H-02***.

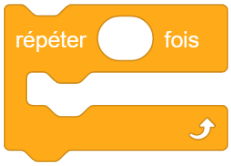
b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-02***.

Conditions de départ

c) Toutes traces d'un dessin précédent sont effacées.

d) Le stylo est positionné à la position (0 ; 0).

Description des tâches

e) Dessine à nouveau un carré d'une longueur de 100 en utilisant cette fois-ci la boucle suivante : 

.

f) Le carré est dessiné quand on appuie sur la touche '4'.

Sauvegarder le projet

g) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-02***.

H.2. Polygones réguliers

« Un polygone d'ordre n est dit régulier s'il est équilatéral (côtés égaux) et équiangle (angles égaux) »[1]

Dessine des polygones réguliers après avoir analysé et compris leurs caractéristiques.

Exercice H-03 Triangle équilatéral

Objectif

Dessiner un triangle équilatéral

Préparation

a) Ouvre le programme ***Exercice H-02***.

b) Renomme-le ***Exercice H-03***.

c) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-03.***

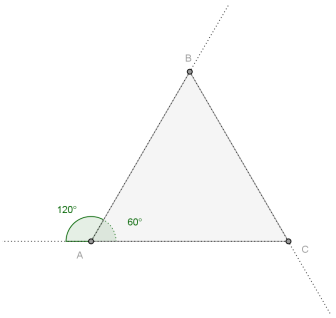
[1] http://fr.wikipedia.org/wiki/Polygone

Conditions de départ

d) Toutes traces d'un dessin précédent sont effacées.

e) Le stylo est positionné à la position (0 ; 0).

Description des tâches



f) Duplique le script présent et modifie-le afin de dessiner un triangle équilatéral (de longueur 100). Fait d'abord l'analyse du triangle pour déterminer l'angle à appliquer.

g) Le carré est dessiné quand on appuie sur la touche '3'.

Sauvegarder le projet

h) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-03***.

Exercice H-04 Polygone à n sommets

Réfléchis, en te basant sur les connaissances acquises lors des exercices précédents, comment trouver les valeurs qui manquent dans le tableau ci-dessous. Il s'agit dans chaque ligne de créer un polygone régulier. Tu n'as pas besoin de Scratch pour cet exercice.

Figure géométrique	Nombre de sommets	Angle (°)	Somme des angles (°)
Triangle			
Carré			
Pentagone			
Polygone à n sommets			

Exercice H-05 Pentagone

Objectif

Dessiner un pentagone

Préparation

a) Ouvre le projet ***Exercice H-03*** et appelle-le ***Exercice H-05***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-05***.

Conditions de départ

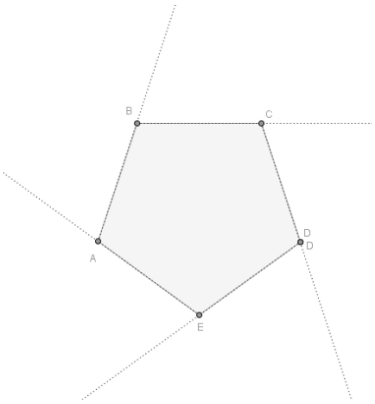
c) Toutes traces d'un dessin précédent sont effacées.

d) Le stylo est positionné à la position (0 ; 0).

Description des tâches

e) Ajoute un script pour dessiner un pentagone de longueur 100.

f) Le pentagone est dessiné quand on appuie sur la touche '5'.



Sauvegarder le projet

g) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-05***.

Exercice H-06 Hexagone régulier

Objectif

Dessiner un hexagone

Préparation

a) Ouvre le projet ***Exercice H-05*** et appelle-le ***Exercice H-06***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-06***.

Conditions de départ

c) Toutes traces d'un dessin précédent sont effacées.

d) Le stylo est positionné à la position (0 ; 0).

Description des tâches

e) Ajoute un script pour dessiner un hexagone de longueur 100.

f) Le pentagone est dessiné quand on appuie sur la touche '6'.

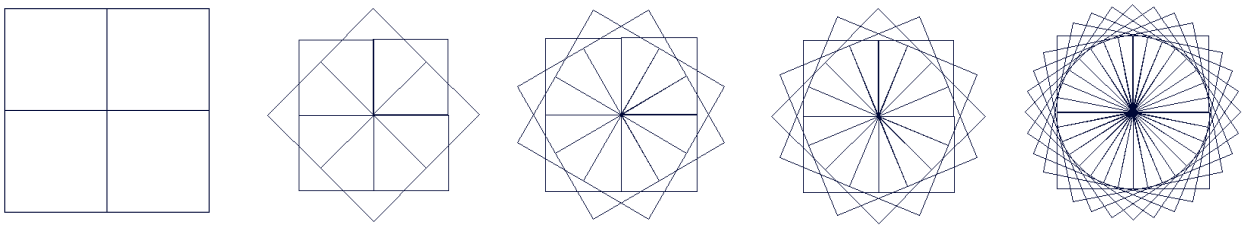
Sauvegarder le projet

g) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-06***.

H.3. Mandalas simples

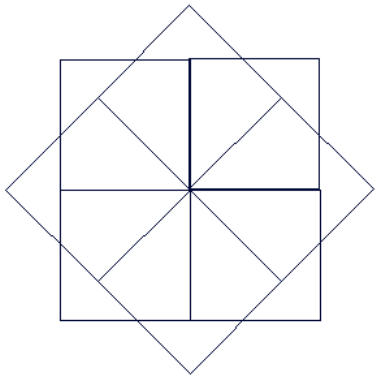
Le mandala qu'on créera maintenant est composé de plusieurs carrés qui sont dessinés en rotation autour du centre.

Exemple :



4 carrés | 8 carrés | 12 carrés | 16 carrés | 32 carrés

Exercice H-07 Mon premier mandala



Objectif

Un mandala composé de 8 carrés

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice H-07***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-07***.

Conditions de départ

c) Toutes traces d'un dessin précédent sont effacées.

d) Le stylo est positionné à la position (0 ; 0).

Description des tâches

e) Il faut d'abord calculer l'angle de rotation entre deux carrés consécutifs. Fais-en sorte que le dessin final soit symétrique.

f) Calcule maintenant l'angle utilisé pour le mandala :

Angle = …….° / ……. = …….°

g) Le mandala est dessiné quand on appuie sur la touche '8'.

h) Ajoute le script qui dessine le mandala composé de 8 carrés.

Sauvegarder le projet

i) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-07***.

Exercice H-08 Dessiner un cadre

Objectif

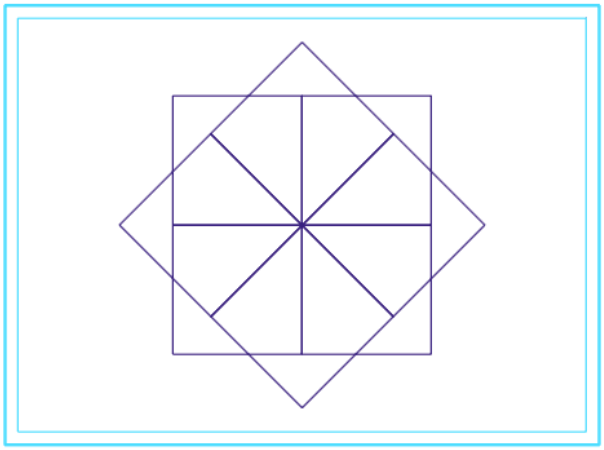
Dessiner un cadre

Préparation

a) Ouvre le projet ***Exercice H-07Exercice H-08*** et appelle-le ***Exercice H-08***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-08.***

Description des tâches



c) Le cadre est dessiné quand on appuie sur la touche ‘b’.

d) Le mandala est mis en valeur si tu l’entoures d’un cadre double.

e) Point du coin en haut à droite du cadre extérieur : 230 ; 170

f) Taille du stylo: 1

g) Point du coin en haut à droite du cadre intérieur : 220 ; 160

h) Taille du stylo: 2

Sauvegarder le projet

i) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-08***.

Exercice H-09 Mon deuxième mandala

Objectif

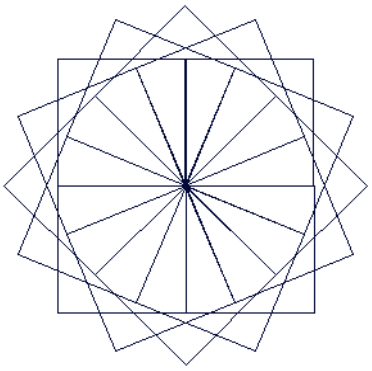
Dessiner un deuxième mandala

Préparation

a) Ouvre le projet ***Exercice H-08*** et appelle-le ***Exercice H-09***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-09.***

Description des tâches



c) Ce mandala est dessiné quand on appuie sur la touche 'm'.

d) Ajoute le script qui dessine le mandala composé de 16 carrés (voir les opérateurs du chapitre **I.2** pour le calcul de l'angle).

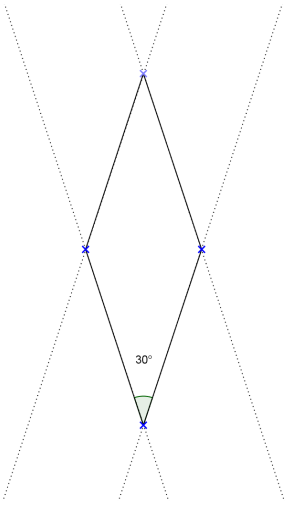
Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-09***.

Exercice H-10 Un mandala encore plus beau

Objectif

Dessiner un mandala encore plus beau avec des losanges



Préparation

a) Crée un nouveau projet et nomme-le ***Exercice H-10***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-10***.

Conditions de départ

c) Toutes traces d'un dessin précédent sont effacées.

d) Le stylo est positionné à la position (0 ; 0).

Description des tâches

e) Ce mandala est dessiné quand on appuie sur la touche 'r'.

f) Crée un mandala composé de 12 losanges identiques. Commence avec un seul losange, détermine les angles par un calcul et inscris-les sur le schéma ci-dessus.

Sauvegarder le projet

g) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-10***.

H.4. Quelques mandalas composés simples

Exercice H-11 Des mandalas composés

Objectif

Réaliser un mandala en combinant deux ou plusieurs motifs.

Préparation

a) Ouvre le projet ***Exercice H-10*** et appelle-le ***Exercice H-11***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-11.***

Description des tâches

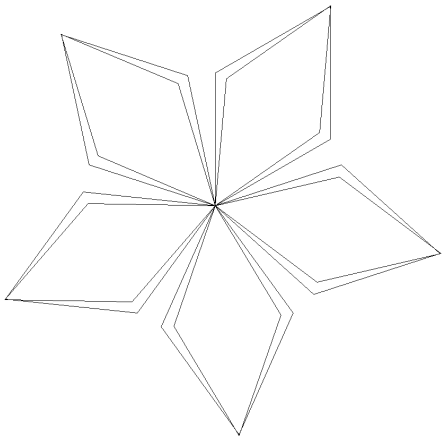
c) Crée le mandala ci-contre.

d) Ce mandala est dessiné quand on appuie sur la touche '1'.

e) Le grand losange a des côtés de longueur 82 et des angles de 130° et de 50°.

f) Le petit losange a des côtés de longueur 80 et des angles de 140° et de 40°.

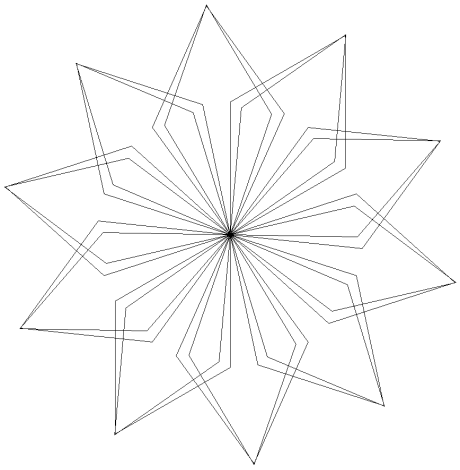
g) Le mandala est composé de 5 sous-éléments.



h) Crée ensuite le mandala ci-contre.

i) Ce mandala est dessiné quand on appuie sur la touche '2'.

j) Le mandala est composé de 10 sous-éléments, les mêmes que pour le mandala précédent.



Sauvegarder le projet

k) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-11***.

H.5. Dessiner des cercles

On peut simuler le dessin d'un cercle en dessinant un polygone régulier avec beaucoup de sommets (par exemple 360 sommets, en se tournant de 1° à chaque fois).

Exercice H-12 Mandalas composés 2

Objectif

Dessiner un mandala à l'aide de cercles

Préparation

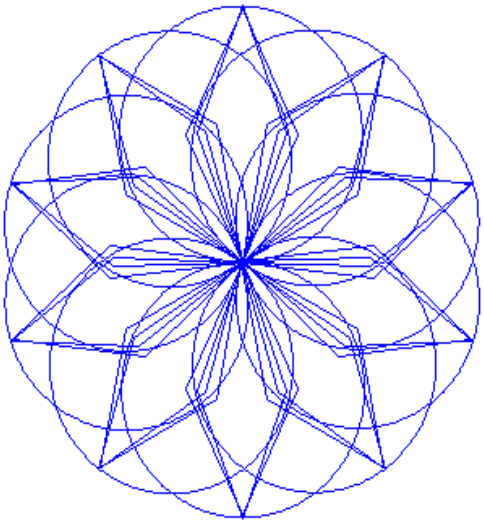
a) Ouvre le programme ***Exercice H-11*** et appelle-le ***Exercice H-12***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-12.***

Description des tâches

c) Copie le script du mandala qui est dessiné quand on appuie sur 'a'.

d) Modifie le script pour réaliser le mandala ci-contre.

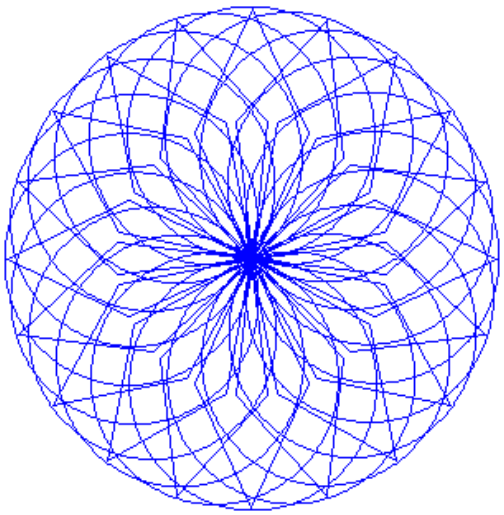


e) Ce mandala est dessiné quand on appuie sur la touche 'c'.

f) Le cercle est un polygone à 60 côtés d'une longueur de 8.

g) Le mandala est composé de 10 sous-éléments.

h) Crée ensuite le mandala ci-contre.



i) Ce mandala est dessiné quand on appuie sur la touche 'd'.

j) Le mandala est composé de 20 sous-éléments.

Sauvegarder le projet

k) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-12***.

H.6. Dessiner des cercles (pour avancés)

La difficulté pour le dessin d'un cercle est de déterminer la longueur ℓ des segments pour arriver à un radius *r* donné.

Longueur totale des 360 segments (longueur ℓ) : ℓ (1)

Circonférence du cercle (radius *r*) : (2)

Si dans une équation tu mets (1) à égalité de (2), tu trouves une formule pour la longueur ℓ :

$$\ell = \text{———}$$

Exercice H-13 Déterminer la longueur d'un côté (pour avancés)

Objectif

Calculer et vérifier la longueur d'un segment

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice H-13***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice H-13***.

Description des tâches

c) Calcule la longueur d'un segment en sachant que le cercle doit avoir un radius de 50.

d) Vérifie sur ton dessin si ton radius est correct.

Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice H-13***.

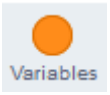
I. Variables et expressions mathématiques

I.1. Variables

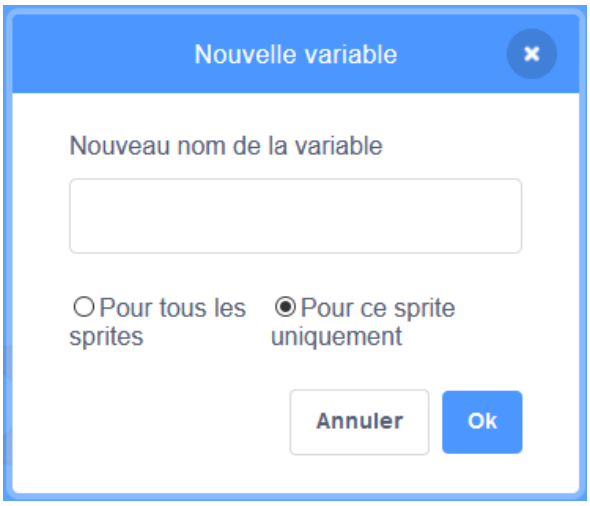
Nous sommes très souvent amenés à devoir nous souvenir de certaines choses : un numéro de maison, le code PIN d'un téléphone, un mot de passe, le nom d'un interlocuteur au téléphone, combien d'argent il nous reste en poche, ce qu'il faut encore faire l'après-midi, etc. En programmation, nous utilisons à cette fin des 'variables' au lieu d'un bout de papier ou d'un nœud dans un mouchoir.

Les variables nous permettent de mémoriser des valeurs, de les relire et de les changer. En Scratch, des variables de types différents sont utilisées : les nombres entiers, les nombres décimaux, ou des textes.

I.1.1 Créer et afficher des variables

La palette 

nous permet de créer de nouvelles variables, de les modifier et de les supprimer. Lors de la création d'une variable nous devons d'abord lui donner un nom significatif :



Nous définissons également quels objets ont le droit d'utiliser les variables, voire de les modifier.

Principe:

En général, une variable devrait toujours rester 'privée' (option "Pour ce lutin uniquement"). Aucun autre objet n'a le droit de modifier volontairement ou involontairement notre variable. (Nous ne laissons pas non plus trainer notre carnet d'adresses au milieu de la salle de classe.)

Dans quelques cas d'exception, nous mettons notre variable à disposition des autres objets.

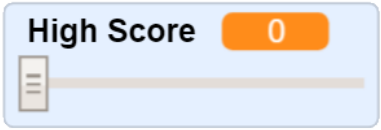
Dès sa création, la variable apparaît dans la palette sous forme d'un bloc :

(Le crochet indique que la variable est visible sur la scène).

L'affichage de la variable sur la scène peut se faire de plusieurs manières :

lecture normal

lecture large



barre de défilement

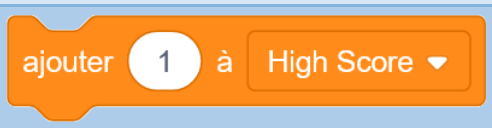
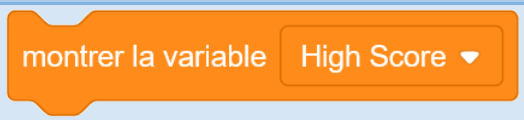
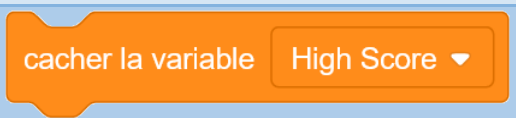
Le mode d'affichage (la 'lecture') de la variable peut être modifié par un double clic ou par un clic droit de la souris.

Cas spécial de la barre de défilement :

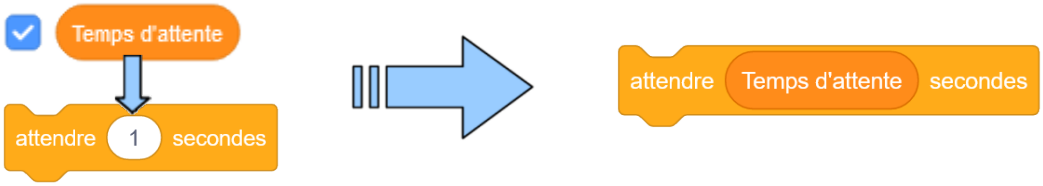
- Ce mode ne fonctionne évidemment que pour les nombres entiers.
- Il est possible de définir également par un clic droit son intervalle de validité, c.-à-d. son minimum et son maximum.

I.1.2 Travailler avec des variables

Après avoir créé la première variable, nous voyons apparaître quelques blocs qui nous permettent de modifier les variables dans nos programmes :

	Sélectionner une variable puis entrer une valeur (il est possible d'entrer des nombres négatifs, des décimales ou du texte)
	Sélectionner une variable puis entrer une valeur (il est possible d'entrer des nombres négatifs ou des décimales – mais **pas de texte**)
	Sélectionner une variable : Ces deux blocs nous permettent de rendre une variable visible sur la scène – ou de la cacher.
	

Dans un programme, tu peux utiliser les variables partout où jusqu'à présent tu as saisi directement des nombres ou des textes. Il suffit de les tirer avec la souris au bon endroit du bloc d'instruction.



Attention : il n'est pas possible de mélanger les textes et les nombres !

- Les variables numériques peuvent être utilisées partout.
- Les variables alphanumériques (de type texte) devraient être utilisées aux endroits prévus : (dit [ ] … , pense [ ] … , demande [ ] …)

Exercice I-01 Free the crab – reloaded!

Objectif

Le crabe doit tenter de se libérer de la cage et compter combien de fois il heurte le mur de la cage.

Préparation

a) Ouvre le projet ***Exercice F-03*** („Free the crab") et appelle-le ***Exercice I-01***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice I-01***.

Description des tâches

c) Ajoute une variable '*Collisions*', qui sera incrémentée (+1) à chaque fois que le crabe touche le mur.

d) Relance le programme quand le crabe a trouvé la sortie et a été replacé au milieu de la cage. Qu'est-ce que tu observes ? Résous ce problème !

Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice I-01***.

Exercice I-02 Free the crab – revisited!

Objectif

Le crabe doit tenter de se libérer de la cage et compter combien de fois il touche le mur de la cage. La vitesse du crabe peut être saisie.

Préparation

a) Ouvre le projet ***Exercice I-01*** („Free the crab – reloaded!") et appelle-le ***Exercice I-02***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice I-02***.

Description des tâches

c) Ajoute une variable '*Temps d'attente*' représentée comme barre de défilement. Règle la barre de défilement à un intervalle entre 0 et 5. Incorpore maintenant la variable dans ton programme afin de régler avec elle la vitesse du crabe.

Sauvegarder le projet

d) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice I-02***.

Exercice I-03 FishChomp – II

Objectif

Ajouter à un jeu existant un score et une limitation de temps.

Préparation

a) Ouvre le projet ***FishChomp***, que ton enseignant met à ta disposition et appelle-le ***Exercice I-03***.

Principe : le gros poisson suit la position de la souris et tente de manger les petits poissons.

Analyse les scripts des différents objets et essaie de comprendre comment le programme fonctionne.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice I-03***.

Description des tâches

c) Crée une nouvelle variable '*Score*' qui est incrémentée de 10 points pour chaque petit poisson que le gros poisson attrape. Le score doit bien-entendu être remis à zéro au lancement du programme.

d) Le jeu doit être limité à 30 secondes. Crée une variable '*Countdown*', qui est mise à 30 au début du programme, puis décrémentée (diminuée) de 1 à chaque seconde. Après l'écoulement de 30 secondes, le jeu s'arrête.

e) N'affiche la règle du jeu (le texte affiché en haut à droite) que pendant les 5 premières secondes du jeu.

Sauvegarder le projet

f) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice I-03***.

I.1.3 Les variables prédéfinies

Il existe pour chaque objet quelques variables prédéfinies qu'on peut utiliser de la même manière que les variables créées par nous-mêmes :

- palette **Mouvement** : position x,y, direction (angle)
- palette **Apparence :** n° de costume, n° d'arrière-plan, taille
- palette **Sons :** volume
- palette **Capteurs :** réponse, chronomètre, volume sonore, … actuelle

I.2. Les expressions

Comme tout autre langage de programmation, Scratch permet de comparer des valeurs (les nombres et les textes) et de calculer avec elles. On n'a pas besoin de beaucoup de blocs d'opération pour obtenir finalement un puissant langage de programmation.

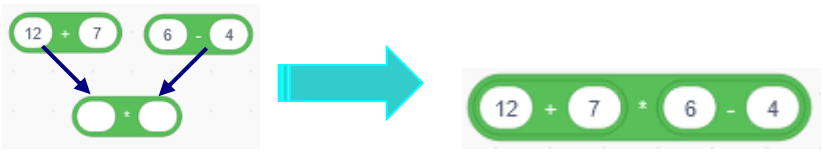
La plupart de ces blocs se situent dans la palette '**Opérateurs**' et traitent des nombres :

	addition, soustraction multiplication, division	pour les nombres entiers et décimaux
	générer un nombre aléatoire	réservé aux nombres entiers
	le reste de la division euclidienne	réservé aux nombres entiers
	arrondir un nombre vers l'entier le plus proche	pour les nombres décimaux, le résultat est un nombre entier
	plusieurs fonctions mathématiques	La fonction la plus importante : *racine carrée, valeur absolue* Autres fonctions : *sin, cos, tan, asin, acos, atan, ln, log, e^, 10^*

Il est possible d'imbriquer les blocs les uns dans les autres afin de réaliser des calculs plus complexes. On peut comparer cela avec les parenthèses en mathématiques :

Exemple :

(12+7)*(6-4)
correspond à



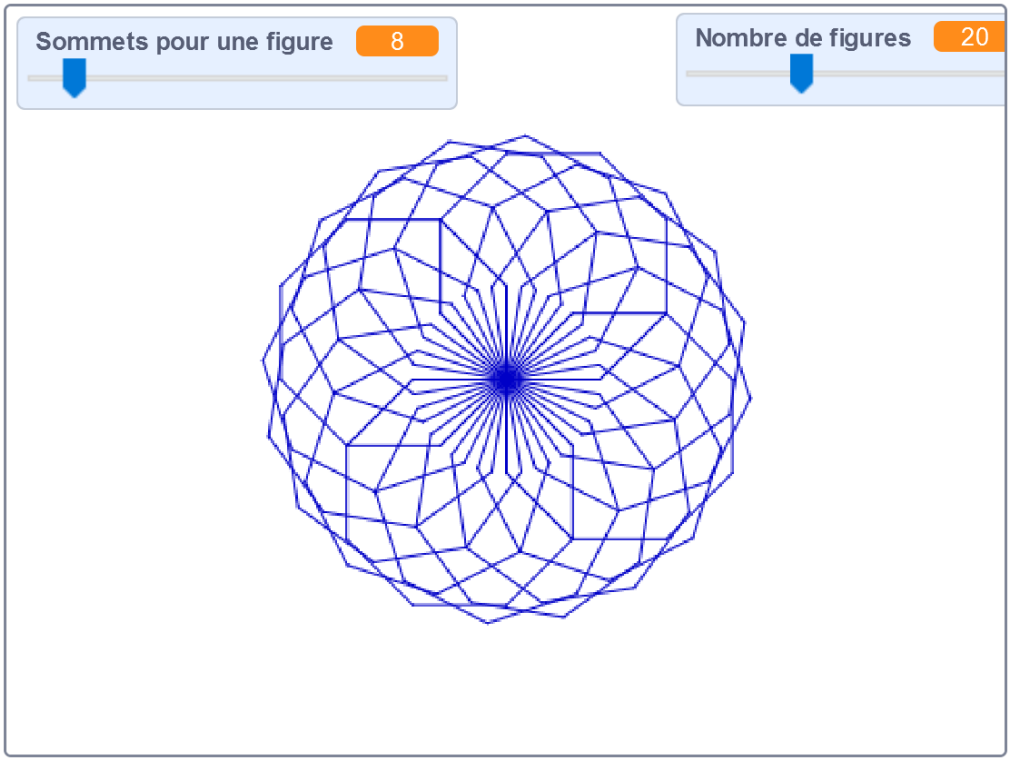
Il faut bien sûr veiller à imbriquer les blocs dans le bon ordre.

Principe: Le système procède en commençant avec le bloc le plus haut et en finissant avec le bloc le plus bas.

Exercice I-04 Générateur de mandalas

Objectif

Calculer automatiquement des mandalas



Préparation

a) Crée un nouveau projet et nomme-le ***Exercice I-04***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice I-04***.

Description des tâches

c) Définis 2 variables **„Sommets pour une figure“** [3…60]. et **„Nombre de figures“** [0…60], sous forme de barres de défilement.

d) Quand on appuie sur la touche ‘espace’, le programme dessine un mandala composé d’autant de figures que la valeur de la variable **‘Nombre de figures’**, chaque figure possédant autant de coins que la valeur de la variable **‘Sommets pour une figure’**.
La longueur d’un côté doit être calculée automatiquement de manière à ce que la figure ne dépasse pas la scène.

e) Affine les mandalas selon ton goût (couleur, taille et intensité du stylo, repeindre…).

Sauvegarder le projet

f) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice I-04***.

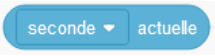
Exercice I-05 Montre analogue avec des « *variables de temps* » (pour avancés)

Préparation

a) Ouvre le projet ***Exercice D-02*** et appelle-le ***Exercice I-05***.

b) Sauvegarde-le sur ton ordinateur sous le nom de ***Exercice I-05***.

Description des tâches

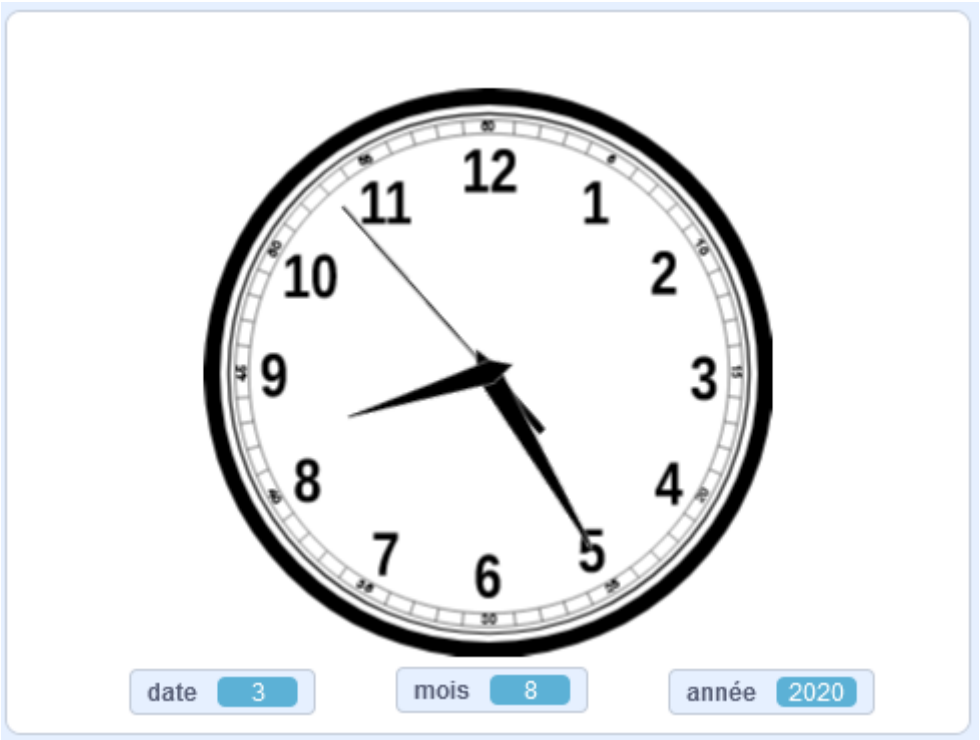
c) Anime l'aiguille existante des secondes en utilisant la *variable de temps* suivante : 

d) Ajoute les aiguilles pour les minutes et les heures.

e) Affiche aussi la date actuelle.

Sauvegarder le projet

f) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice I-05***.



I.2.1 Vrai ou faux

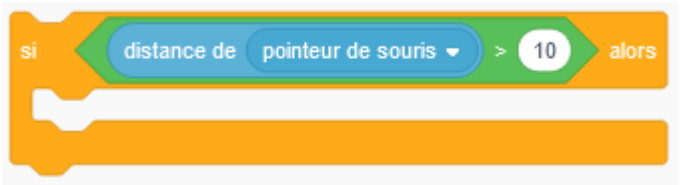
Comme nous avons vu dans le chapitre E (Prendre des décisions), les blocs aux extrémités pointues servent à formuler des **conditions**.

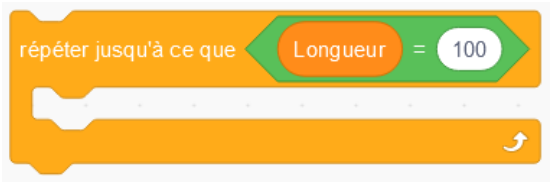
Plus précisément, un bloc aux extrémités pointues donne ou bien le résultat '**vrai**' (condition vérifiée) ou bien le résultat '**faux**' (condition non vérifiée). De tels blocs peuvent être utilisés comme sous-blocs p.ex. dans les blocs "si <…> … sinon …", "attendre jusqu'à <…>" ou "répéter jusqu'à <…>".

Les blocs opérateurs suivants nous permettent de comparer des valeurs (des nombres ou des textes) et d'utiliser leur résultat dans le contrôle de nos scripts.

Opérateurs de comparaison :

Exemples





Exercice I-06 FishChomp – III

Objectif

Retenir et afficher le score maximal (Highscore)

Préparation

a) Ouvre le projet *FishChomp* – II (***Exercice I-03***) et appelle-le ***Exercice I-06***.

b) Sauvegarde le projet sur ton ordinateur sous le nom de ***Exercice I-06***.

Description des tâches

c) Modifie le compte à rebours afin de pouvoir utiliser la boucle '*répéter jusqu'à*'. (Ceci donne une solution plus flexible puisqu'il ne faut changer qu'à un seul endroit la valeur de départ de la variable '**Countdown**' pour modifier la durée du jeu.).

d) Crée une variable '**Highscore**' qui vérifie à la fin du jeu si 'Score' dépasse le Highscore actuel. Dans ce cas, le score actuel devient le nouveau Highscore et un son (de ton choix) se fait entendre.

Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice I-06***.

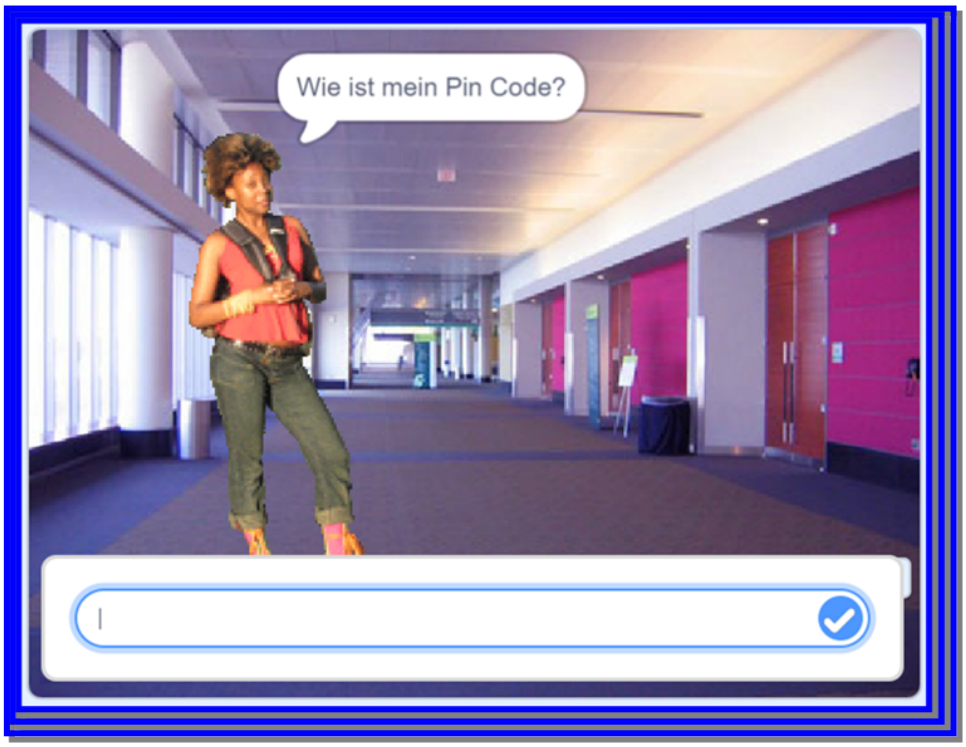
Exercice I-07 Devinette

Objectif

Deviner un nombre secret aléatoire calculé par l'ordinateur

Contexte

Il faut absolument que tu puisses téléphoner, mais les piles de ton portable sont vides. Cassy a son portable sur elle mais elle ne veut pas que tu l'utilises. Elle te lance le défi suivant : *« Si tu arrives à deviner mon code PIN à 4 chiffres en moins de 15 essais alors tu peux utiliser mon portable autant que tu veux. À chaque essai, je ne te dis que si mon code PIN est plus grand ou plus petit que le nombre de ton essai. »*



Préparation

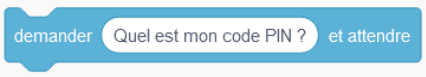
a) Crée un nouveau projet et nomme-le ***Exercice I-07***.

b) Sauvegarde le projet sur ton ordinateur sous le nom de ***Exercice I-07***

Description des tâches

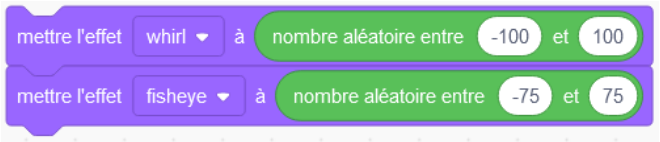
c) Développe un programme qui simule Cassy et sa devinette :

- Au début du programme, il faut enregistrer dans une variable (invisible) ***Pin*** un code aléatoire entre 1000 et 9999. C'est le nombre que le joueur doit deviner.
- Tu peux également présenter le contexte de l'histoire. (P.ex. Cassy téléphone au début et dit *« Ah bon, tu veux utiliser mon téléphone. Eh bien, je te le laisse si tu arrives à deviner mon code à 4 chiffres en moins de 15 essais ! »*)

- Utilise le bloc 

, pour attendre la saisie du nombre suivant. Tu trouves ce bloc dans la palette **Capteurs**. La valeur saisie par le joueur se trouve alors automatiquement dans la variable prédéfinie 

.
- Une deuxième Variable (visible) ***Essai*** mémorise le nombre d'essais du joueur.
- Si le joueur arrive à deviner le code PIN en moins de 15 essais, Cassy se fâche et donne le portable ☺
- Si, par contre, il n'y est pas arrivé après 14 essais, elle danse de joie pendant quelques secondes…
- Tu trouves des images pour Cassy dans la bibliothèque de Scratch. Pour la danse tu peux importer des sons de la bibliothèque ou utiliser tes propres sons. Pour rendre la danse plus rigolote encore, jette un coup d'œil sur les effets de la palette **Apparence** :



Amuse-toi bien pour la programmation et la devinette !

Sauvegarder le projet

d) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice I-07***.

I.2.2 Pour avancés: conditions combinées

Nous avons souvent besoin de plusieurs conditions pour vérifier un état. Ces conditions doivent alors être combinées.

En mathématiques p.ex. il faut souvent vérifier si un nombre se trouve à l'intérieur d'un intervalle : $0 < X < 1000$

Une condition combinée (ou complexe) peut être réalisée en utilisant les blocs Scratch suivants :

<et> veut dire que les deux conditions doivent être vérifiées

<ou> veut dire qu'au moins une des deux conditions doit être vérifiée

<non> veut dire que la condition ne doit pas être vérifiée

Pour vérifier si $0 < X < 1000$ on peut écrire :

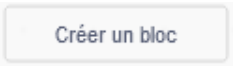
Pour vérifier si $X \leq 0$ ou $X \geq 1000$ on peut écrire en Scratch :

Mais on peut également écrire :

J. Procédures

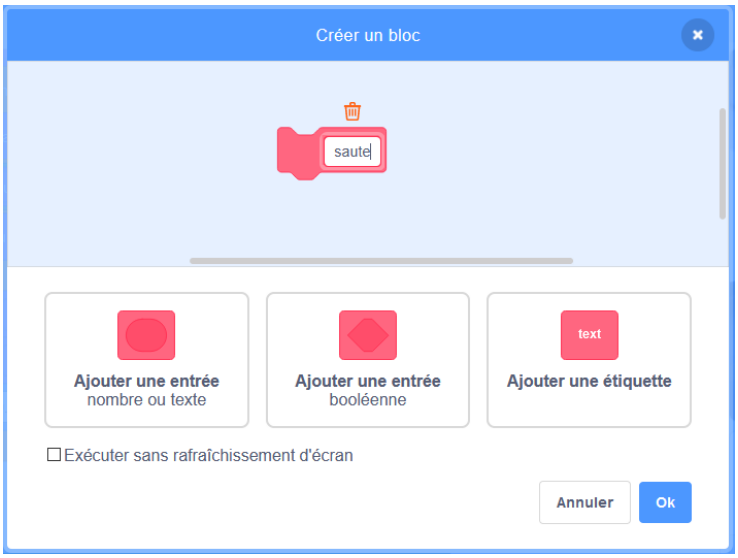
Il est possible de créer de nouveaux blocs d'instructions à partir des blocs existants. Ces nouveaux blocs sont appelés **procédures**. On peut les utiliser dans un script de la même manière qu'on utilise les instructions.

J.1. Définir une procédure

On crée une nouvelle procédure par un clic sur le bouton 

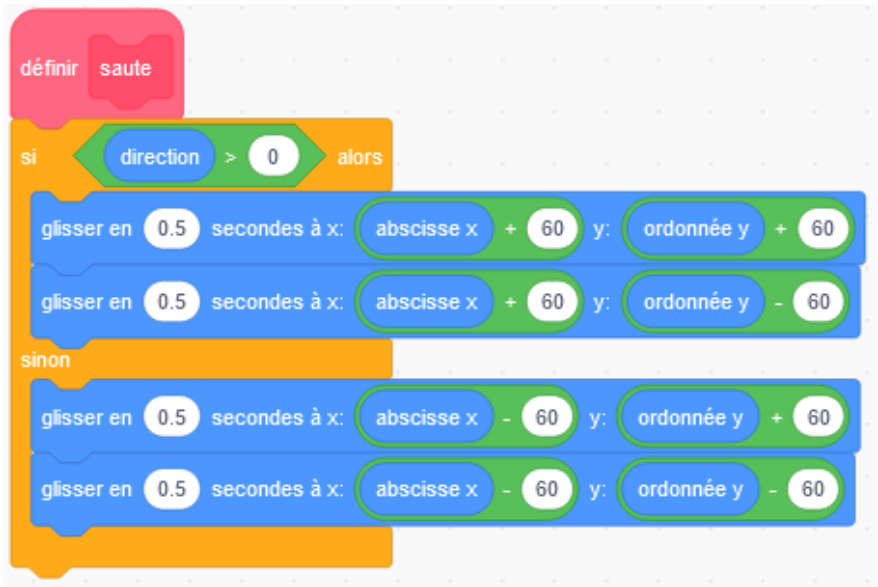
de la palette 

. Il faut saisir ensuite le nom de la procédure dans la fenêtre de dialogue.



Après avoir cliqué sur **OK**, le chapeau "**définir**" apparaît dans l'espace de programmation. Nous pouvons maintenant assembler sous ce chapeau les instructions qui doivent être exécutées lors d'un appel de la procédure.

L'exemple suivant montre la définition de la procédure "**saute**". Elle permet au sprite de sauter 120 pixels en longueur et 60 pixels en hauteur.



J.2. Appeler une procédure

Après avoir défini une procédure, celle-ci apparait sous forme de bloc dans la palette et peut être insérée dans un script, comme les autres instructions.

Notons :

Dans un sprite on peut utiliser les procédures définis dans celle-ci, mais pas les procédures définis dans d'autres sprites.

Exercice J-01 Le chat saute

Objectif

Le chat saute par-dessus des obstacles.

Préparation

a) Crée un nouveau projet et nomme-le ***Exercice J-01***.

b) Sauvegarde le projet sur ton ordinateur sous le nom de ***Exercice J-01***.

Description des tâches

c) Définis la procédure **saute** comme indiqué plus haut.

d) Quand on appuie sur la touche espace, le chat saute. Utilise la procédure **saute** !

e) Quand on clique sur le drapeau vert, le chat se promène sur la plage en faisant des allées et venues. Quand il touche la pierre, il doit sauter par-dessus automatiquement. Utilise la procédure **saute** !

Sauvegarder le projet

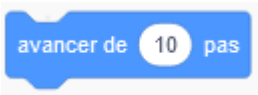
f) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice J-01***.

g) Imagine-toi avoir réalisé le programme sans la procédure **saute**. Quelles conclusions peux-tu en tirer ? Quels sont les avantages des procédures ?

J.3. Arguments et paramètres

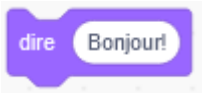
Beaucoup d'instructions que tu connais déjà possèdent des **arguments**.

Exemples :



L'instruction **avancer** a un argument qui indique la grandeur du pas (10 dans l'exemple).

Le champ d'édition est arrondi puisqu'il attend une valeur numérique.

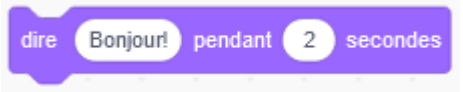


L'instruction **dire** a un argument qui indique le texte que le sprite doit dire ("Coucou !" dans l'exemple).

Le champ d'édition est rectangulaire puisqu'il attend une valeur de texte.

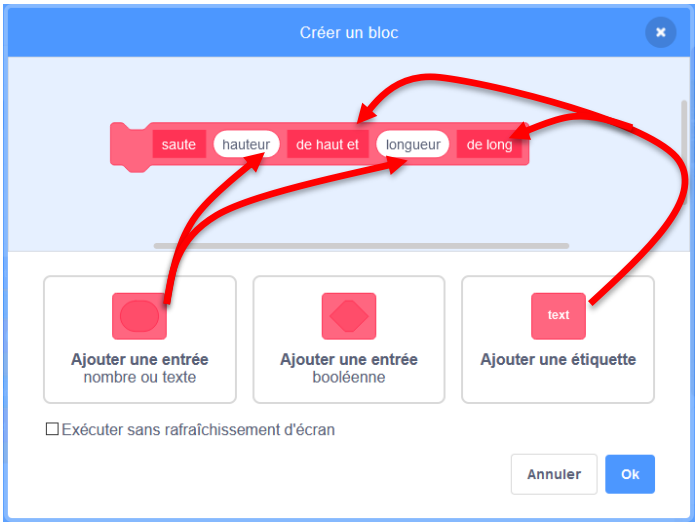
L'instruction **attendre** a un argument qui indique une condition qu'il faut attendre avant de continuer l'exécution.

Le champ d'édition a des bouts pointus puisqu'il attend une condition (dont la valeur est ***vrai*** ou ***faux***).



Cette commande a deux arguments : Le texte à afficher (champ d'édition rectangulaire) et la durée d'affichage du texte (champ d'édition arrondi).

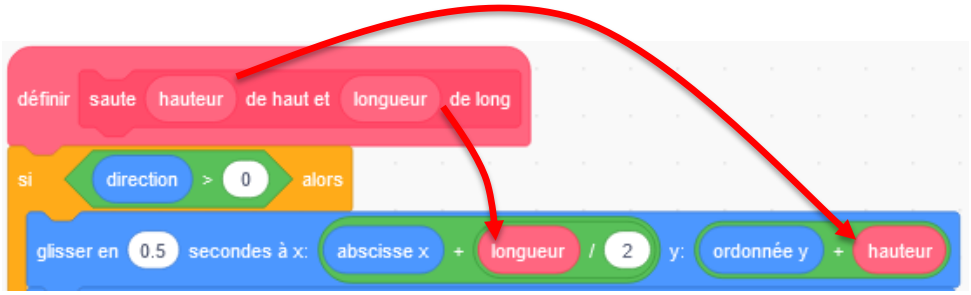
Une procédure peut également posséder des arguments. Lors de la définition de la procédure nous pouvons décrire les **paramètres**. C'est ainsi qu'on appelle les variables qui vont recevoir les arguments à l'aide de champs d'édition.



Dans cet exemple nous venons d'ajouter deux paramètres à notre procédure **saute**, un pour la hauteur et un pour la longueur du saut. Nous pouvons donc maintenant indiquer la hauteur et la longueur du saut lors de l'appel de la procédure.

Comme nos paramètres attendent des valeurs numériques, nous avons cliqué sur les boutons arrondis. Nous avons ensuite nommé les paramètres et ajouté des textes descriptifs pour rendre le bloc plus compréhensible.

Après un clic sur **OK**, le chapeau **définir** apparaît, muni des deux nouveaux paramètres avec leurs indications textuelles.



Il est maintenant facile de glisser ces paramètres aux endroits souhaités.

Notons :

Pour modifier la définition d'une procédure, il faut cliquer avec la touche droite de la souris sur son bloc et choisir ensuite **modifier**.

Exercice J-02 Le chat sait sauter encore plus haut

Objectif

Le chat saute par-dessus d'obstacles plus hauts encore.

Préparation

a) Ouvre le projet ***Exercice J-01*** et appelle-le ***Exercice J-02***.

b) Sauvegarde le projet sur ton ordinateur sous le nom de ***Exercice J-02***.

Description des tâches

c) Définis la procédure **saute** comme indiqué plus haut avec les paramètres **hauteur** et **longueur** et complète la programmation des scripts.

d) Place un palmier (anglais : *palmtree*) sur la plage et modifie le script pour que le chat saute par un petit saut au-dessus de la pierre mais avec un grand saut au-dessus du palmier !

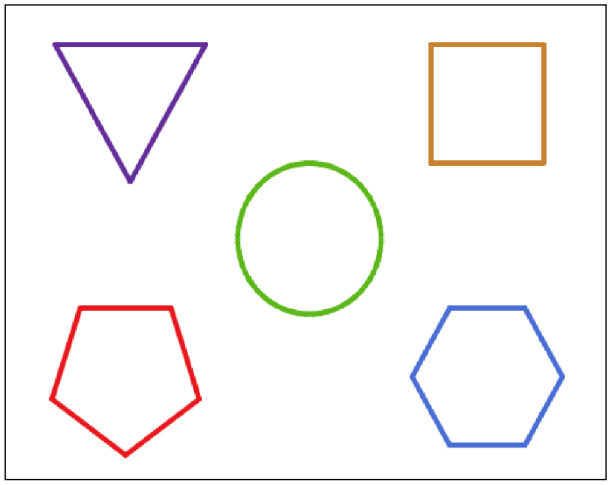
Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice J-02***.

Exercice J-03 Polygones réguliers

Objectif

Dessiner des polygones réguliers différents à l'aide d'une seule procédure.



Préparation

a) Crée un nouveau projet et nomme-le ***Exercice J-03***.

b) Sauvegarde le projet sur ton ordinateur sous le nom de ***Exercice J-03***.

Description des tâches

c) Crée une procédure **polygone** qui dessine un polygone régulier. Le nombre de sommets ainsi que la longueur d'un côté sont donnés sous forme d'arguments.

d) Crée un programme qui dessine des polygones de différentes formes et couleurs sur la scène.

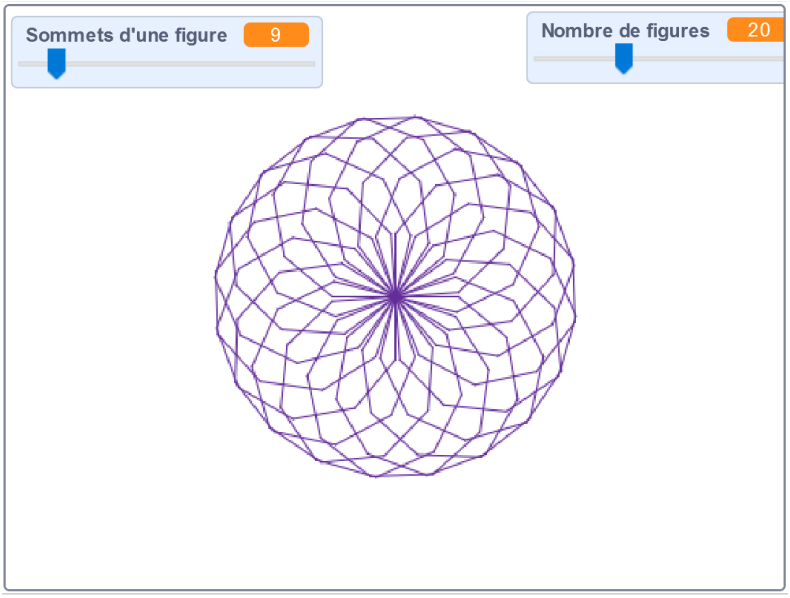
Sauvegarder le projet

e) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice J-03***.

Exercice J-04 Générateur de mandalas

Objectif

Générateur de mandalas avec procédures



Préparation

a) Ouvre le projet ***Exercice J-03*** et appelle-le ***Exercice J-04***.

b) Sauvegarde le projet sur ton ordinateur sous le nom de ***Exercice J-04***.

Description des tâches

c) Crée une procédure **mandala** qui dessine un mandala composé d'un nombre donné de polygones. Utilise une 2e procédure **polygone**. Essaie de trouver toi-même les paramètres nécessaires pour les procédures.

d) Utilise la procédure **mandala** pour créer un générateur de mandalas (voir ***Exercice I-04*** du chapitre I).

e) Crée une procédure **rectangle** qui dessine un rectangle. La longueur et la largeur du rectangle sont les arguments de la procédure.

f) Utilise la procédure **rectangle** pour dessiner un cadre double autour du mandala (lancé par la touche 'r').

g) Modifie la définition de la procédure en cochant la case à cocher "Exécuter sans rafraîchissement de l'écran" et relance le générateur de mandalas. Qu'est-ce que tu remarques ?

Sauvegarder le projet

h) Sauvegarde de nouveau le projet sur ton ordinateur sous le nom de ***Exercice J-04***.