



UNE INITIATIVE DE



Avec la participation
de l'UNESCO



AFRICA CODE WEEK

Une introduction au langage de programmation de Scratch

par

Brendan Smith, Camden Education Trust, Irlande.

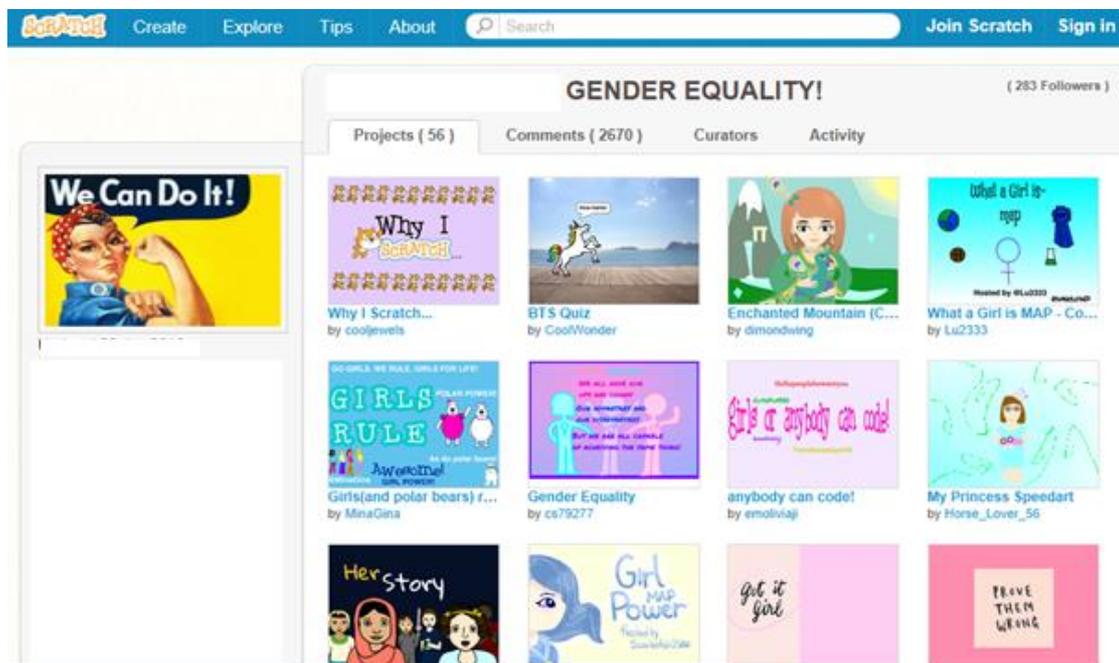


Table des matières

| | |
|--|-----|
| Introduction à Scratch et à l'art du codage | 3 |
| Leçon 1 - Introduction à l'interface Scratch..... | 6 |
| Leçon 2 - Codage : les premières étapes..... | 8 |
| Leçon 3 - Insérer des sons dans un script..... | 17 |
| Leçon 4 - Améliorer la marche du sprite..... | 21 |
| Leçon 5 - Sprites multicolores..... | 23 |
| Leçon 6 - Changer le décor (Scène) | 25 |
| Leçon 7 - Un chien qui aboie et poursuit le chat ! | 33 |
| Leçon 8 - Créer un sprite..... | 39 |
| Leçon 9 - Les sprites contrôlés par curseur | 43 |
| Leçon 10 - Le sprite psychédélique | 44 |
| Leçon 11 - Interaction du sprite..... | 46 |
| Leçon 12 - Deux sprites ayant une conversation..... | 49 |
| Leçon 13 - Créer un récif de corail | 51 |
| Leçon 14 – La balle cible | 57 |
| Leçon 15 - Les sprites dansants | 60 |
| Leçon 16 - Dessiner des formes..... | 63 |
| Leçon 17 – La balle qui rebondit | 69 |
| Leçon 18 - Dessiner à main levée..... | 71 |
| Leçon 19 – Promener le chien..... | 72 |
| Leçon 20 – Planification et conception d'un jeu..... | 80 |
| Leçon 21 - Jeu - Attaque de requin ! | 82 |
| Leçon 22 - Attaque de requin avancée..... | 88 |
| Leçon 23 – Tennis solitaire..... | 97 |
| Leçon 24 – Jeux d'aventure : l'incroyable labyrinthe ! | 101 |
| Leçon 25 – Chasseur de démons..... | 109 |
| Leçon 26 - Extension du jeu Chasseur de démons | 112 |
| Leçon 27 – Jeu de tir : les astéroïdes | 117 |
| Leçon 28 – Jeux à deux joueurs..... | 125 |
| Leçon 29 - Jeux à deux joueurs - tennis pour deux | 132 |
| Leçon 30 – Quizz de géographie : voyager à travers un continent..... | 136 |

Introduction à Scratch et à l'art du codage

Comment le codage informatique peut fonctionner en tant qu'environnement d'apprentissage interdisciplinaire



Dans cette section, nous vous présenterons Scratch, un langage de programmation développé au Media Lab du MIT aux États-Unis, qui a capté l'imagination des enfants partout dans le monde.

Scratch permet aux utilisateurs de créer facilement leurs propres histoires, animations, jeux, musique, œuvres d'art interactifs et de partager ces créations sur le Web. Il est donc idéal pour les enfants âgés de huit à onze ans.

Qu'est-ce qu'un programme ?

Un programme est un ensemble d'instructions indiquant à un ordinateur ou à un autre appareil électronique ce qu'il doit faire. Ces instructions ou commandes sont écrites en langage artificiel (c'est-à-dire non parlant). Le script utilisé est souvent appelé code ou code informatique.

La programmation informatique ou le codage est le processus d'écriture du code.

D'où vient le terme Scratch ?

Scratch, développé par le Media Lab du MIT aux États-Unis, est le **nom** donné à ce langage de programmation très puissant mais très adapté aux enfants.

Le terme Scratch a été choisi par ses inventeurs pour ses points communs avec la méthode utilisée par un DJ de hip-hop pour mélanger différentes pistes de musique afin de créer de nouveaux sons.



Car un élément clé du site Scratch est que les membres peuvent télécharger les jeux informatiques d'autres personnes et modifier les détails s'ils le souhaitent (par exemple: « accélérer » ou « ralentir » les personnages animés dans un jeu). Le résultat est qu'un grand nombre de ces projets en ligne sont des remixes d'autres projets sur le site <http://scratch.mit.edu>

Une façon amusante d'apprendre

Le langage Scratch a des similitudes avec les jeux de construction de briques des enfants. Il utilise une structure simple de **briques** ou de **blocs** graphiques de code informatique qui s'emboîtent pour créer et contrôler le son, la musique et les images. Il convient donc parfaitement au plaisir et à l'apprentissage des enfants, car il utilise leur penchant naturel pour le bricolage avec des matériaux de construction afin de créer différentes formes, jeux et histoires au sein d'une **nouvelle** dimension numérique **interactive** **excitante**. Finie l'époque de la syntaxe de texte difficile et ennuyeuse qui constituait traditionnellement un programme d'ordinateur.



Avantages éducatifs : calcul, alphabétisation et au-delà

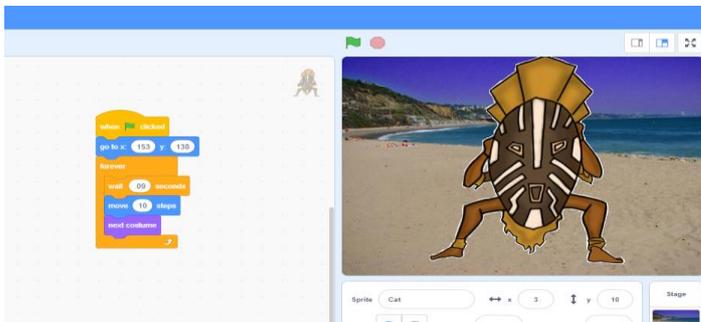
Scratch fournit aux enfants et aux jeunes un environnement unique pour développer et utiliser leurs talents artistiques et créatifs grâce à la création d'animations attrayantes. Pendant les sessions de formation, les enfants travaillent ensemble pour effectuer certains projets, apprendre

à planifier, concevoir et partager des tâches. Ils sont encouragés à montrer et expliquer leurs travaux achevés aux autres participants, où l'analyse critique et les compliments de leurs camarades de classe font partie intégrante du processus d'apprentissage de présentation et de communication.

Scratch permet aux enfants de développer leurs compétences artistiques et créatives dans un monde numérique à la fois stimulant, valorisant et aventureux.

L'enseignement de Scratch utilise beaucoup d'aspects différents des programmes scolaires d'école élémentaires, comme l'art, les langues, et plus particulièrement le calcul et l'alphabétisation.

Comme les enfants apprécient de créer et de partager les projets Scratch, ils apprennent d'importants concepts mathématiques et informatiques, tels que l'arithmétique (l'addition, la soustraction, la multiplication, la division des nombres), la géométrie (une branche de mathématiques qui traite de la mesure, des relations



de points, des lignes, des angles, etc. par des termes tels que coordonnées, formes, taille, position relative des figures), l'algèbre (variables, symboles représentant des quantités et relations

numériques) ainsi que des notions supplémentaires telles que des algorithmes, tout en apprenant à raisonner de manière systématique, à résoudre des problèmes, à travailler ensemble et à développer des compétences de présentation..



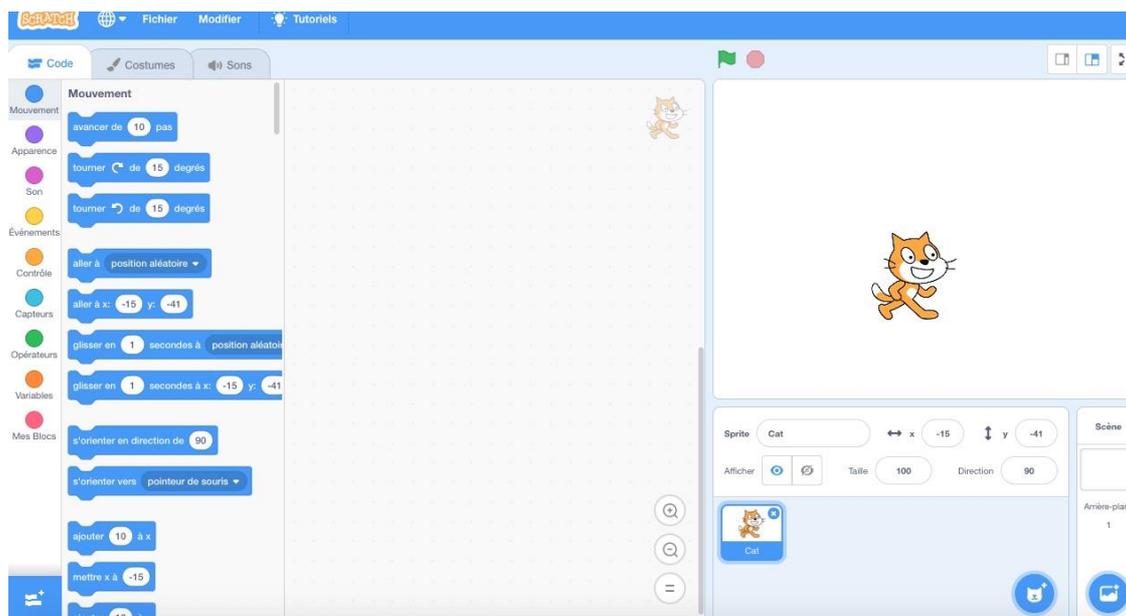
La motivation est très rarement un problème pour les élèves de Scratch

Scratch permet aux utilisateurs de créer une grande variété de projets en ligne qui peuvent refléter leurs intérêts personnels et qui peuvent également être utilisés comme ressources dans de nombreux sujets du cursus scolaire, par exemple : créer des jeux interactifs, cartographier

une visite de différents pays pour un cours de géographie, produire un concert pour un cours de musique, construire une poupée avec plusieurs vêtements dans un cours d'art ou créer un récit numérique sur un match sportif, un conte folklorique ou une question scientifique importante telle que les causes du réchauffement climatique.

Avec des élèves de primaire, l'enseignant peut élargir les éléments artistiques de Scratch en leur demandant de faire des modèles en argile de leurs sprites lors de la planification de leurs projets.

Leçon 1 - Introduction à l'interface Scratch



Partage et stockage de projets Scratch en ligne

Plusieurs projets utilisés dans ces notes d'apprentissage sont stockés en ligne sur le site Web de Scratch.

Dans ce cas, l'adresse Web spécifique du projet individuel est fournie.

Ce site devrait aussi être utilisé pour stocker les meilleurs exemples de projets créés par vos étudiants participants, afin que d'autres mentors puissent les utiliser comme ressources pédagogiques dans leur classe et comme exemples à suivre par d'autres participants.

Pour accéder au site Scratch :

Tapez dans la barre d'adresse : <http://scratch.mit.edu>

Si vous n'avez pas encore de compte, cliquez sur **Rejoindre Scratch**

Rejoindre Scratch **Se connecter**

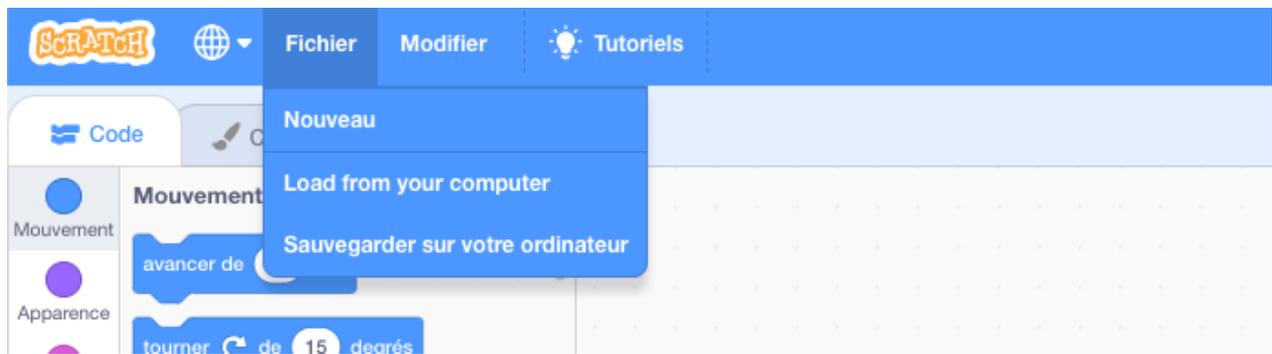
Néanmoins, vous pouvez rejoindre la communauté existante « Code for Africa » en allant à l'option de connexion et

Sous le **Nom d'utilisateur**, tapez « africacodeweek »

Sous le **Mot de Passe**, tapez « fionnfionn ».

Pour télécharger un projet terminé depuis votre ordinateur ou un autre dispositif sur le compte Africa Code Week sur le site Scratch, accédez d'abord à l'option **Créer** dans la barre de menus.

Puis sous l'option **Fichier** sélectionnez **Load from your computer**



En tant qu'enseignant, vous souhaitez peut-être créer un compte sur le site Scratch pour votre classe, votre école ou votre groupe, qui pourra être utilisé pour héberger des exemples pour chaque travail de votre étudiant.

Ce cours utilisera toutefois l'**éditeur Scratch hors ligne**

Accédez à un navigateur Web tel que Google Chrome, Safari ou Firefox.

Tapez dans la barre d'adresse :

<http://scratch.mit.edu/scratch3download/>

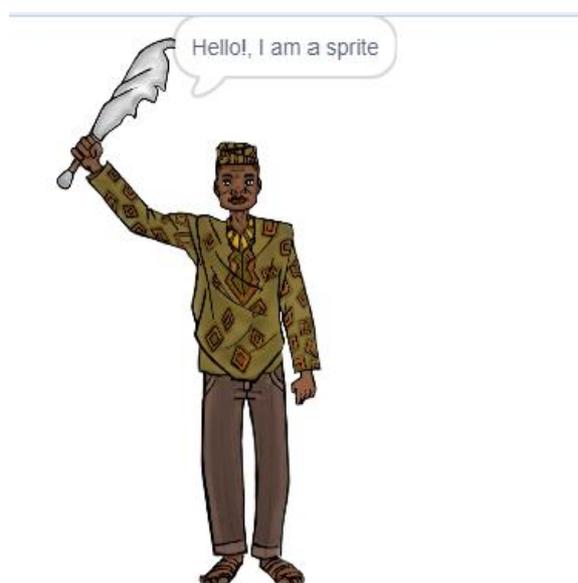
et suivez les instructions.

Pour vous familiariser avec le programme, cliquez sur **Tutoriels**.

Leçon 2 - Codage : les premières étapes

Un lutin parlant et en mouvement

Un **Sprite** est un personnage ou objet animé dans votre programme.



Dans Scratch, les Sprites peuvent se déplacer, être actifs ou être des objets immobiles. Nous allons choisir un personnage sprite que nous allons animer.

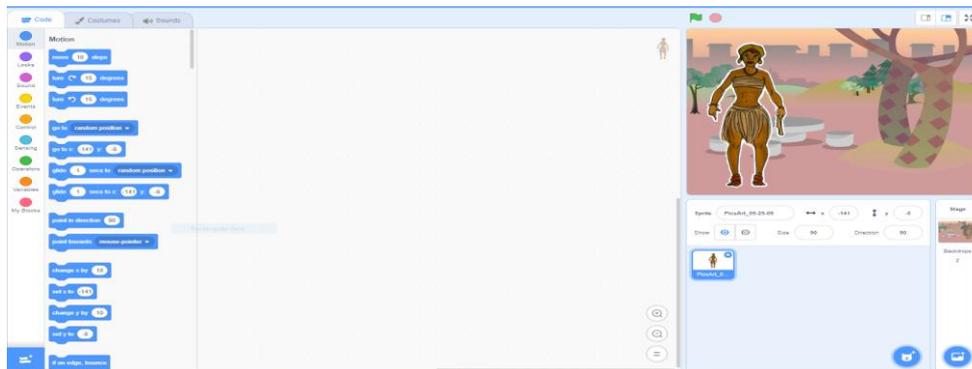
Cependant, il est important de savoir qu'un sprite ne peut rien faire de lui-même.

L'action d'un sprite provient d'une réaction à des scripts saisis par l'utilisateur dans la **Zone de script** (scripts ou code de programme). Ces scripts sont les instructions ou commandes qui indiquent au sprite ce qu'il doit faire et sont écrites selon une séquence. L'utilisateur fait glisser des morceaux de code individuels de la Palette de Blocs dans la zone de script. Ces blocs s'emboîtent ensuite comme des morceaux de briques des jeux de constructions d'enfants afin de créer les instructions.

Alors, entrons dans Scratch et programmons le sprite pour qu'il parle et se déplace sur l'écran !

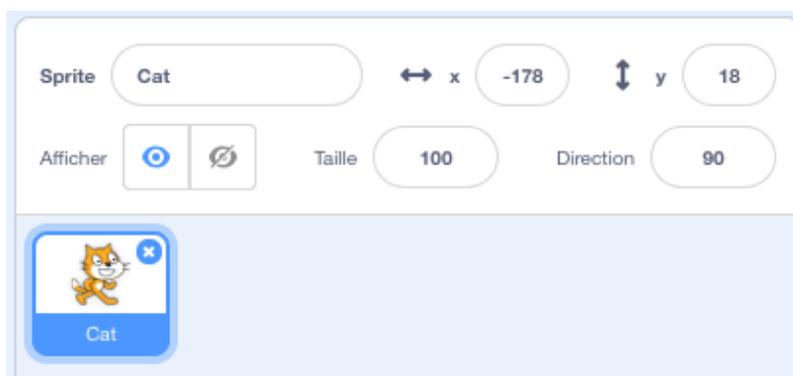
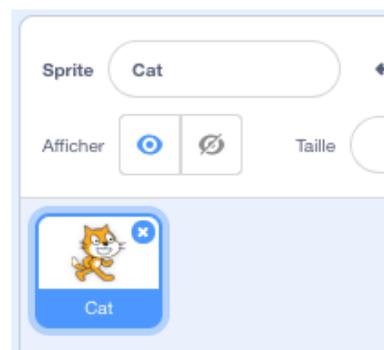
Pour ouvrir **Scratch** sur votre appareil, double cliquez sur l'icône Scratch sur votre ordinateur.

Vous verrez l'écran d'ouverture.



Prenez le temps de vous familiariser avec les principales fonctionnalités de cet écran ou avec ce que nous appellerons l'interface Scratch (page d'accueil).

Notez que la version réduite du chat est en surbrillance bleue, ce qui signifie qu'il s'agit du composant actif.



Mise en place du texte

En plaçant des blocs dans la Zone de Script, le sprite nous dira son nom.

Tout d'abord allez dans la section **Catégorie des blocs** située à gauche de l'interface Scratch, qui contient des dossiers thématiques de blocs de code tels que Mouvement, Apparence, Son, etc.

À la droite de cette section de l'interface utilisateur graphique, se trouve la **Palette des blocs** qui contient tous les blocs de code individuels.



Allez dans le dossier **Événements**.

Placez le bloc suivant dans la Zone de Script :



Ce bloc est une commande « Quand ».

Cela signifie qu'une fois que le **Drapeau vert** au-dessus de la Scène est cliqué, le Sprite suivra les commandes placées dans la zone de script.

Allez dans la section **Apparences** située dans le panneau Dossiers.

Sélectionnez deux fois le bloc **dire Bonjour pendant deux secondes**.

Tapez le texte **Bonjour !** et **Mon nom est Fionn** (ou le nom que vous voulez) avant de les mettre dans la zone de Script.



Assurez-vous que les blocs s'emboîtent

Ce serait sympa de personnaliser le sprite en lui donnant un nom.



Pour ce faire, cliquez sur l'icône  en haut à gauche de l'image-objet du chat et tapez le texte Fionn ou votre choix de nom dans la case correspondante.

Cliquez sur le Drapeau **vert** en haut à gauche de l'écran de l'ordinateur et voyez ce qu'il se produit.

Maintenant, nous devons faire bouger le chat.
Allez dans le dossier Mouvement

Sélectionnez le bloc Avancer et changez le nombre de pas à 50.

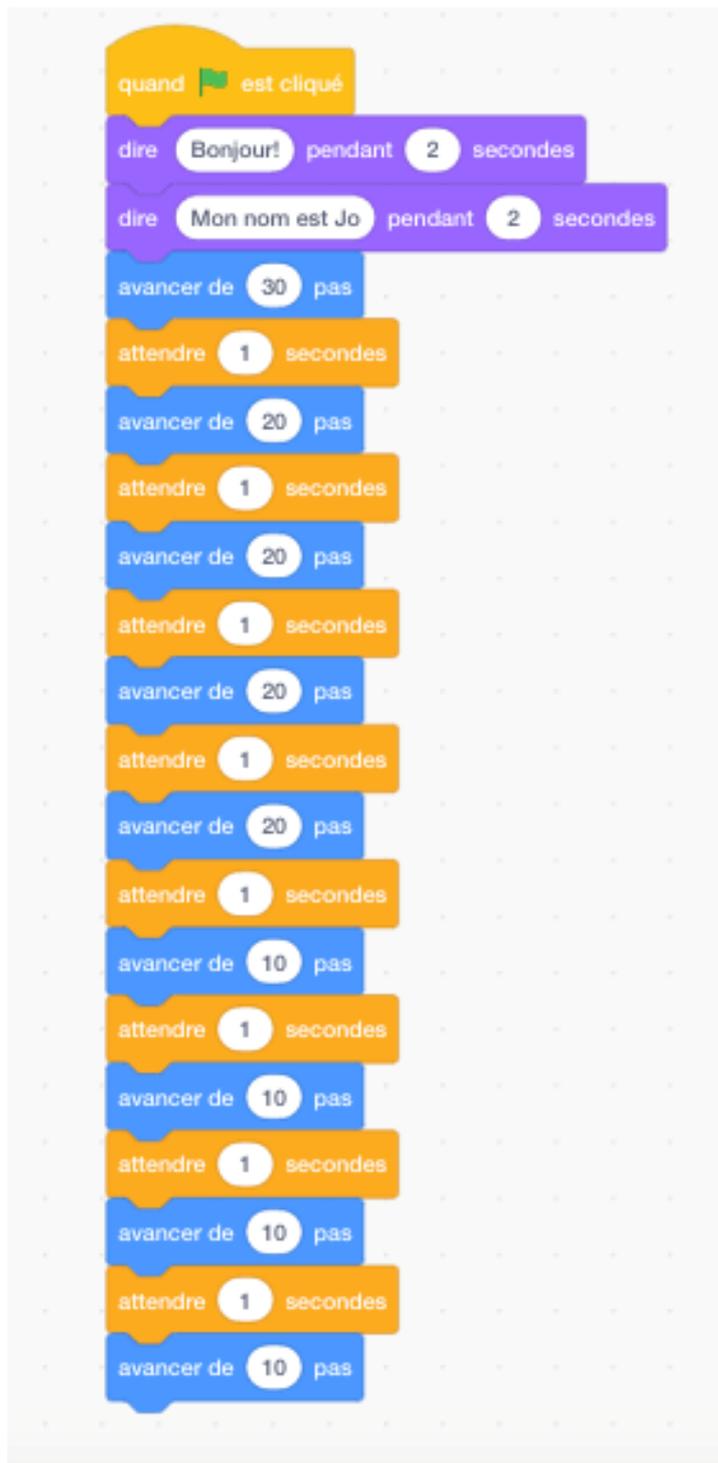


Attachez ce bloc aux autres blocs dans la zone de script et lancez le programme.



Pour augmenter le mouvement du sprite et pour lui donner une impression de marcher, tout d'abord allez dans le dossier de Contrôle et sélectionnez le bloc Attendre

Mettez ce bloc dans la zone de script en ajoutant quelques blocs Avancer :



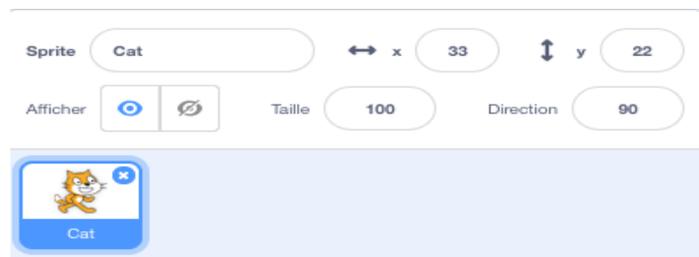
Cependant, nous avons un problème avec le sprite.

Comme vous l'avez peut-être remarqué, si nous continuons à utiliser ce script, le chat continuera à avancer jusqu'à ce qu'il disparaisse presque complètement de l'écran.

Nous devons donc insérer une commande supplémentaire qui le fera revenir au centre de l'écran chaque fois que nous utilisons cet ensemble d'instructions.

L'écran est divisé en **coordonnées X** (horizontal) et **Y** (vertical) basées sur le centre de l'écran afin qu'il soit **(X)0 (Y)0** et que les nombres soient positifs ou négatifs suivant de leur positionnement.

Demandez à vos élèves de déplacer le sprite sur l'écran et de regarder comme les valeurs des coordonnées X et Y changent juste au-dessus de l'icône du sprite située à droite de l'interface du sprite à mesure que le sprite change de position.

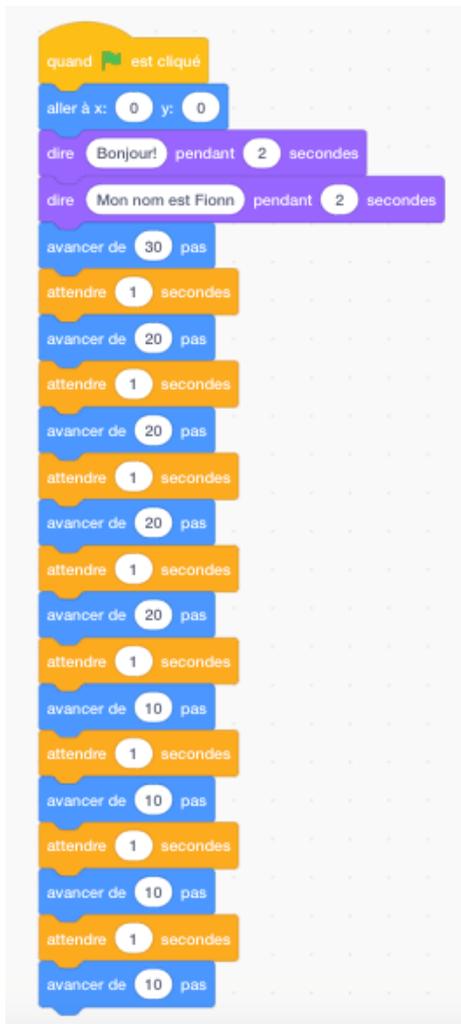


Cette fonction permet à l'utilisateur de positionner différents sprites à différents endroits.

(Voir plus loin l'encadré explicatif sur la Géométrie).

Par conséquent, nous pouvons mettre un morceau de code ou bloc au début de l'ensemble des commandes qui indiquera au Chat de revenir au centre de l'écran chaque fois que nous sélectionnons le drapeau vert.

Le texte apparaîtra comme ceci :



Testez l'efficacité de ce nouveau morceau de code en utilisant la souris pour positionner le sprite vers le bas ou le haut de l'écran avant de cliquer sur l'icône du Drapeau vert.

D'autres méthodes que le Drapeau vert peuvent être utilisées pour lancer le script utilisant la commande « Quand » dans le dossier de Contrôle.

Par exemple, la touche Espace ou les touches Flèches.

Alors, remplacez

par



(depuis la catégorie Événements) dans la Palette.

Maintenant cliquez Espace sur le clavier l'ordinateur pour programme.



la Touche de lancer le

Remplacez

par



Puis lancer le programme en cliquant sur le Chat

Revenez au bloc Drapeau

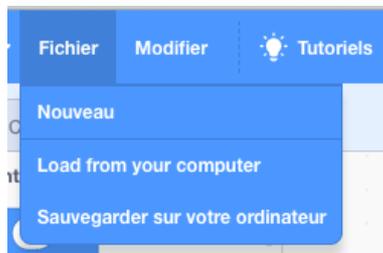


vert.

Sauvegarder le projet

Allez dans le dossier Fichier Sélectionnez « Sauvegarder sur votre ordinateur »





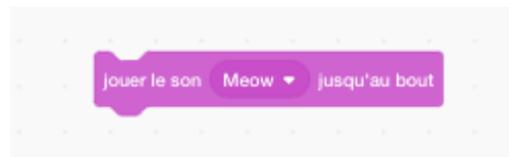
Tapez Fionn, ou votre propre nom pour votre chat dans la section **Enregistrer sous...**
Puis choisissez où vous voulez l'enregistrer, tel que sur le Bureau ou dans Mes Projets.

Leçon 3 - Insérer des sons dans un script

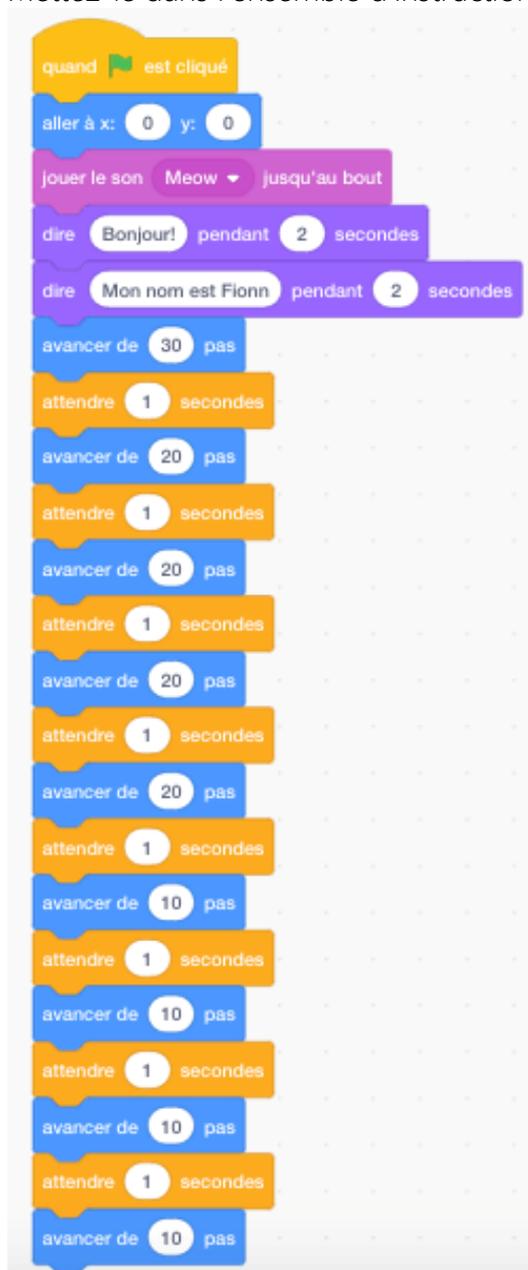
Faire ronronner le chat !

Aller dans la catégorie Son.

Sélectionnez le bloc suivant :



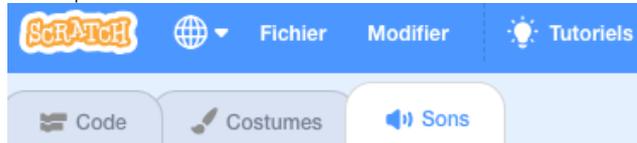
Mettez-le dans l'ensemble d'instructions dans la zone de Script



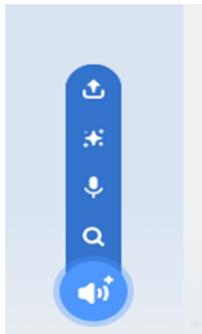
Lancez le programme.

Maintenant remplacez le Miaou du chat par un autre son d'animal dans le script.

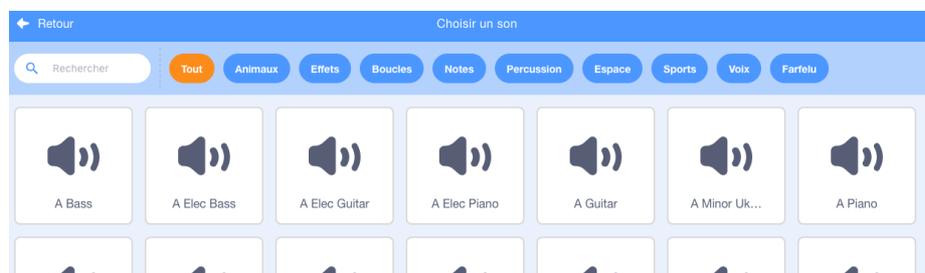
Ceci peut être fait en allant dans la librairie des Sons



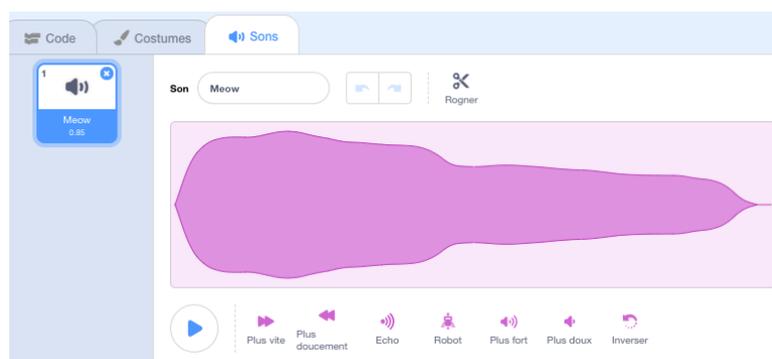
et en cliquant sur l'icône haut-parleur dans le coin en bas à gauche de la page.



Tout un répertoire de sons apparaît.



Cliquez sur le son de votre choix qui apparaîtra ensuite dans la liste de sons sur la colonne de gauche (sous le Code entête).



Vous pouvez maintenant insérer le nouveau son dans votre script en choisissant le bloc concerné (et option de son) sous la librairie Code.

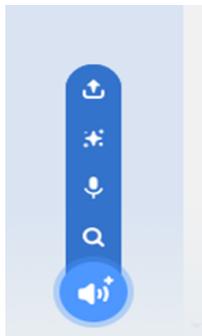


Exercice

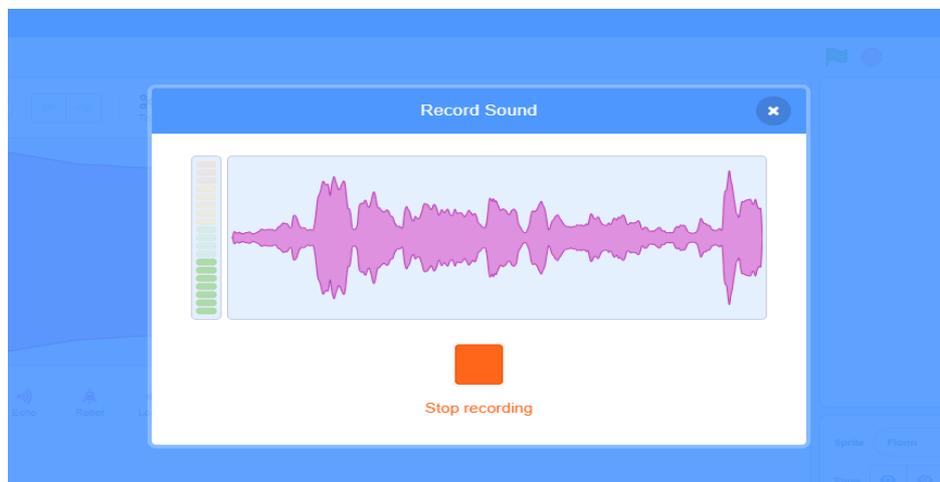
Waouh ! Un chat qui parle !

Maintenant, demandons au chat de dire les mots que nous avons écrits dans le programme, à savoir Bonjour et Mon nom est Fionn.

Cliquez sur le l'icône Micro dans la sélection des icônes en bas à gauche de l'écran, comme illustré ci-dessous :



Puis choisissez d'enregistrer votre voix en cliquant sur le bouton **ROUGE** juste en dessous :



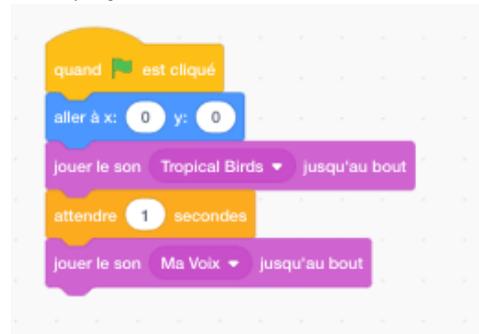
Dites « Bonjour » bien haut et fort.
Vous pouvez ajuster et éditez l'audio selon vos besoins.



Changez le titre de l'Enregistrement1 en Ma voix.

Revenez à la Zone de Script et sélectionnez la catégorie Son (rose).
Mettez le bloc Jouer le son Ma Voix dans le script juste au-dessus du bloc Dites Bonjour pendant 2 secondes

Répétez ce processus d'enregistrement pour Mon Nom est Fionn.



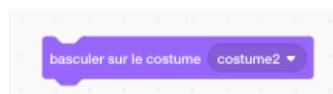
Leçon 4 - Améliorer la marche du sprite

Maintenant rendons les mouvements du chat plus réalistes en lui faisant bouger les jambes lorsqu'il marche.

Ceci peut être obtenu en changeant l'aspect physique d'un sprite individuel sous la section Costumes.

Allez dans Costumes et vous remarquerez qu'il y a une deuxième image ou Costume du même sprite avec les jambes et bras dans une autre position que celle de la première image.

Après les premiers blocs « Avancer » et « Attendre » dans le script, mettez un bloc « Basculer sur le Costume » avec Costume2 sélectionné depuis la librairie des Apparences.



L'option Costume 2 est choisie en cliquant sur la petite flèche blanche à droite de ce bloc

Puis, pour les blocs « Avancer » et « Attendre » suivants, mettez « Basculer sur le Costume » avec Costume 1.



Répétez ce processus tout au long du script, alternant donc entre Costume 1 et Costume 2.

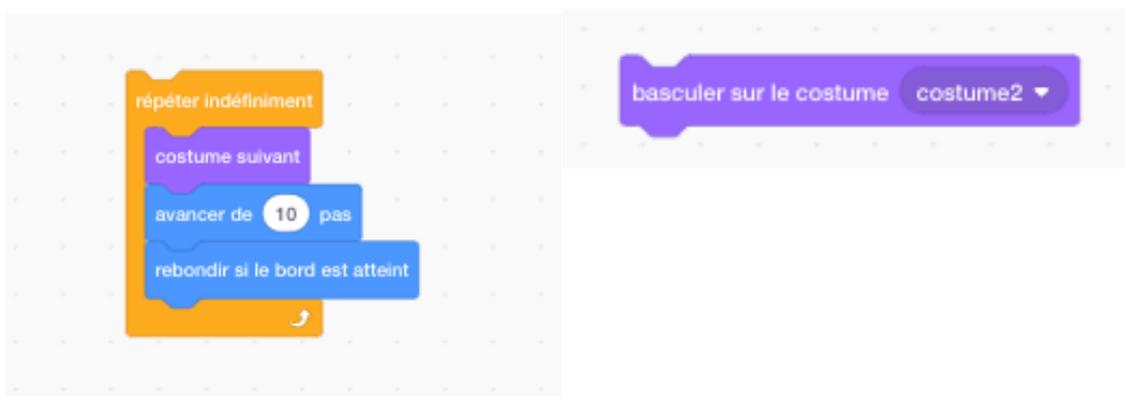
Lancez le Script en cliquant sur l'icône Drapeau vert.

Un chat très rapide !

Dans cette leçon, les étudiants sont initiés au très important bloc **Répéter indéfiniment**, qui montre comment le chat peut se déplacer rapidement à travers l'écran sans s'arrêter !

Commencez d'abord par séparer tout le texte du code placé sous le premier évènement du bloc suivant :

A la place des blocs retirés, mettez les nouvelles commandes suivantes :



Expliquez aux étudiants l'importance du « Répéter indéfiniment » dans la programmation informatique.

Ce morceau de code indique que le sprite changera de costume et avancera une série de pas, non seulement une fois, mais indéfiniment. De plus, l'ajout de la commande **rebondir si le bord est atteint** indique que le sprite ne disparaîtra pas de l'écran mais se retournera chaque fois qu'il atteint le bord de l'écran afin de continuer à marcher.



Par contre, vous constaterez que le sprite marche maintenant peut être à l'envers !

Pour s'assurer que le chat se tient bien droit et qu'il se déplace avec la tête dans le bon sens, allez dans la catégorie Mouvement et mettez sous le bloc suivant :



Leçon 5 - Sprites multicolores

Un chat caméléon ! Changer la couleur d'un sprite

Changeons la couleur du chat pendant qu'il marche.

Pour ce faire, allez dans l'option Costumes, sélectionnez l'icône du chat Costume 1 puis avec le clic droit de la souris sélectionnez Dupliquer.

Faites la même chose pour l'icône Costume 2.

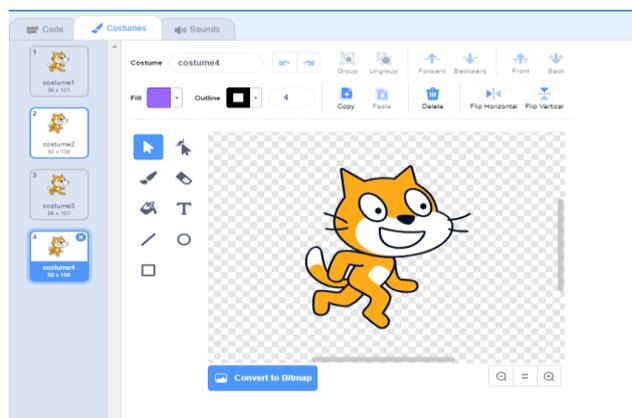
Répétez ce procédé pour chaque Costume 1 et 2 jusqu'à ce que vous ayez neuf costumes.

Modifiez les numéros de Costume dans le script afin qu'ils apparaissent par ordre chronologique.

Puis cliquez sur l'icône Costume 2



À droite des icônes costumes se trouve l'**Éditeur de Dessin**, qui permet de colorer le sprite existant ainsi que d'en dessiner et colorer de nouveaux.



Choisissez une couleur différente dans la palette nommée **Remplissage** située en haut à gauche de l'écran.

Choisissez une couleur et utilisez l'icône pot de peinture pour intégrer cette nouvelle caractéristique dans le Sprite existant sur l'écran.

Assurez-vous que l'Éditeur de dessin soit en **Mode Bitmap** lors de l'utilisation de l'outil pot de peinture.





Remarque : Présentez aux étudiants certaines des autres fonctionnalités de l'Éditeur de Dessin, notamment les outils de duplication, de texte et d'effacement.

Lorsque vous avez terminé appuyez sur ok.

Répétez ce procédé pour intégrer une nouvelle couleur dans chaque costume restant.

Lancez le Script en cliquant sur l'icône Drapeau vert et observez le chat à multiples couleurs marcher !

Leçon 6 - Changer le décor (Scène)

Pour le moment, nous utilisons un fond blanc pour le chat. Apportons donc un peu d'excitation dans sa vie en l'animant dans un nouveau paysage.

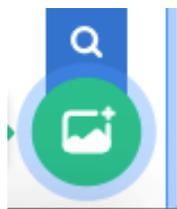
En observant la partie en bas à gauche de l'écran, on remarque que Fionn le chat est actuellement en surbrillance bleue, ce qui signifie, comme indiqué précédemment, que le Sprite est l'élément d'action dans le Script.

Cliquez sur Scène.

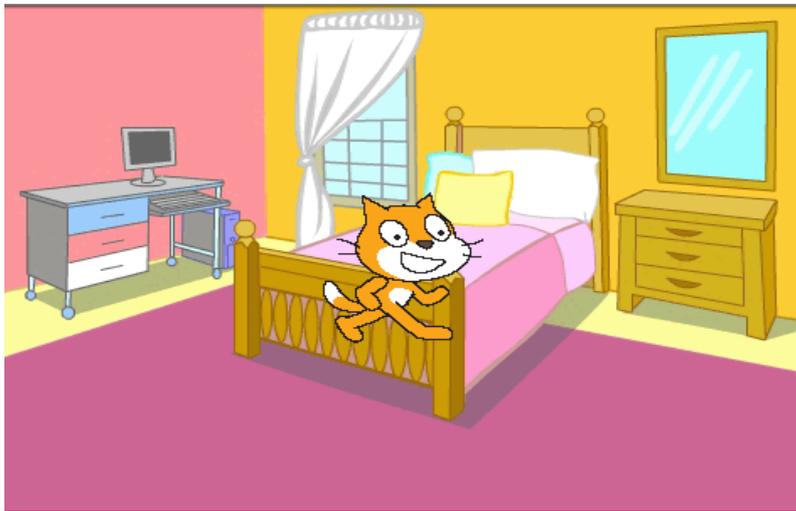
L'icône Scène est maintenant en surbrillance bleue indiquant que c'est devenu l'élément actif que l'utilisateur peut maintenant changer.



Allez sur l'icône Choisir un arrière-plan situé sous l'icône Scène et sélectionnez une des cinq options dans la rangée des icônes (voir ci-dessous).



Allez dans le dossier nommé Intérieur et sélectionnez Bedroom 2.



Cliquez sur l'icône Drapeau vert pour lancer le programme.

Vous remarquerez qu'il y a un problème, on dirait que le chat Fionn marche en l'air !

Commencez donc par demander aux élèves de proposer une solution sur la manière de coder le script afin de faire bouger le chat sur le sol de la chambre à coucher.

Nous devons bien sûr ajouter quelques codes de commande dans le script du chat afin de s'assurer qu'il marche sur le sol.

Alors double-cliquez sur l'icône Arrière-plan pour revenir au Script du sprite (chat).

Déplacez le Sprite dans le coin inférieur gauche de l'écran Scène.



GÉOMETRIE

Les images d'ordinateur sont composées de pixels. Ils peuvent être identifiés à l'aide des coordonnées X et Y.

Un écran d'ordinateur ou une image est constitué d'unités de base appelées **Pixels**. C'est une abréviation de deux mots, image et élément.

Les pixels sont les plus petites unités de couleur sur un écran d'ordinateur ou sur une image d'ordinateur pouvant être contrôlées ou programmées.

Les coordonnées X, Y sont respectivement les adresses **horizontale** (X) et **verticale** (Y) de tout pixel ou point adressable sur un écran d'affichage d'ordinateur.

La coordonnée x est un nombre donné de pixels le long de l'axe horizontal d'un affichage à partir du pixel (pixel 0) au centre de l'écran. La coordonnée y est un nombre donné de pixels le long de l'axe vertical d'un affichage à partir du pixel (pixel 0) de l'écran central. Ensemble, les coordonnées x et y localisent un emplacement de pixel spécifique sur l'écran.

Les coordonnées X et Y font partie de la branche des **mathématiques** connue sous le nom de **géométrie**, qui concerne les questions de position relative des figures, des formes, des tailles et des propriétés de l'espace.

Pour familiariser les utilisateurs avec les quadrillages XY, accédez à la nouvelle section Arrière-plan, comme précédemment entrepris, sélectionnez la dernière ou avant-dernière icône dans la rangée des cinq icônes. Puis choisissez la grille XY, l'avant dernière options d'écran de la bibliothèque de l'arrière-plan.

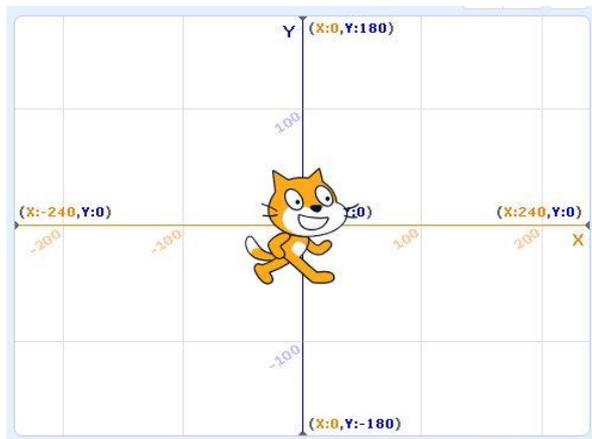
L'écran suivant s'affiche :

Permettez aux étudiants de voir l'écran et de déplacer le sprite afin de remarquer les changements de coordonnées.

Expliquez leur importance, notamment dans les jeux informatiques (par exemple déplacer des objets dans un jeu vers une position de départ).

Revenez dans le décor de la chambre 2.

Déplacez le chat dans le coin supérieur droit de la scène. Les coordonnées X et Y du chat apparaîtront sous la scène comme suit :



Une fois que le chat est en position, les coordonnées X et Y actuelles du chat apparaîtront dans les cases X et Y (comme illustré dans l'image ci-dessus).

Allez dans **Mouvement**.

Trouver le bloc **Aller à X: Y:**

Déplacer ce bloc dans la zone de script et placez-le directement sous le premier bloc (Drapeau vert) de l'ensemble des instructions du programme.

Comme mentionné ci-dessus, les coordonnées X et Y du chat seront indiquées dans le bloc.

Cliquez sur l'icône Drapeau vert pour lancer le programme

Question : que devons-nous faire pour que le chat saute sur le lit ?

Bien entendu, l'opérateur doit demander au sprite de se mettre sur le lit en plaçant, au bon endroit dans le programme, un bloc de mouvement comprenant les coordonnées X et Y correctes.

Alors tout d'abord lancez le programme.

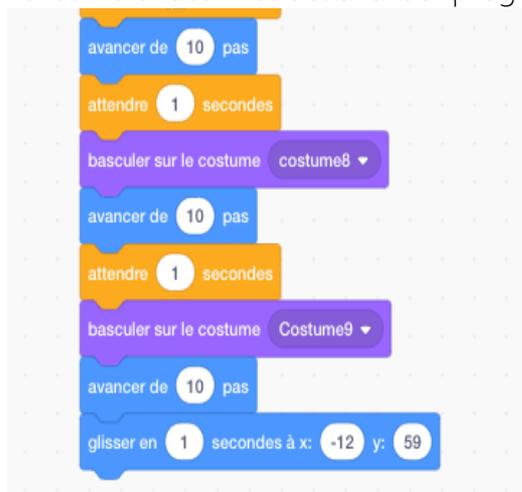
Regardez l'endroit où le chat s'arrête et le programme se termine.

Déplacez le chat sur le lit.

Allez dans Mouvement.

Trouvez le bloc **Glissez en 1 secondes à X: Y:**

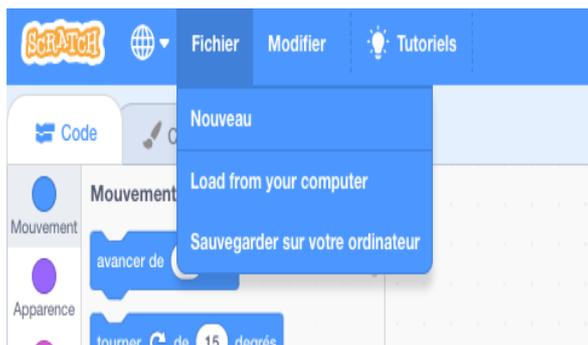
Les nouvelles coordonnées X et Y (du chat sur le lit) seront affichées dans le bloc de code. Déplacez ce bloc dans la Zone de script et mettez-le à la bonne place dans l'ensemble des instructions du programme (il est probablement préférable de l'insérer à la fin du programme).



Cliquez sur l'icône Drapeau vert pour lancer le programme.



Sauvegarder votre fichier de programme avec le nom ChatSaute en sélectionnant **Sauvegarder sur votre ordinateur** dans le menu déroulant **Fichier** situé en haut de l'écran Scratch.



Exercice

Placez quelques blocs de code supplémentaires dans le programme pour permettre au chat de sauter du lit et de marcher quelques pas.

De plus, assurez-vous que le chat change de couleur à chaque pas lors de sa marche.

Exercice

Remplacez l'arrière-plan par une nouvelle image de la bibliothèque d'arrière-plans.

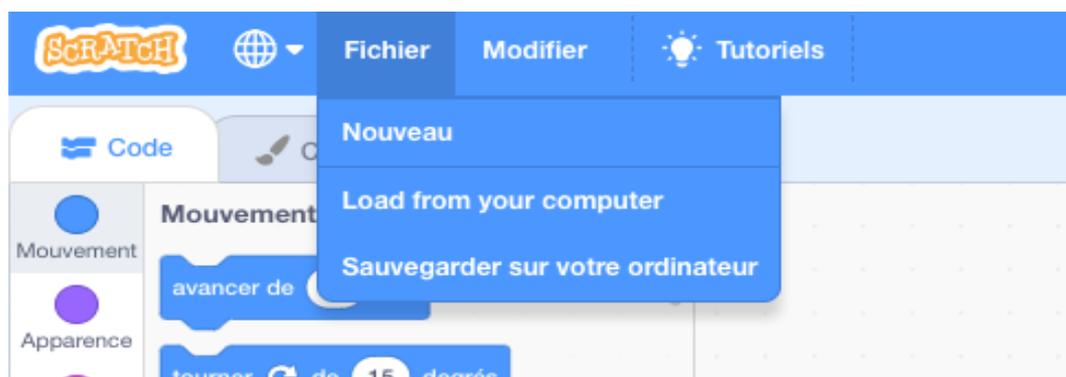
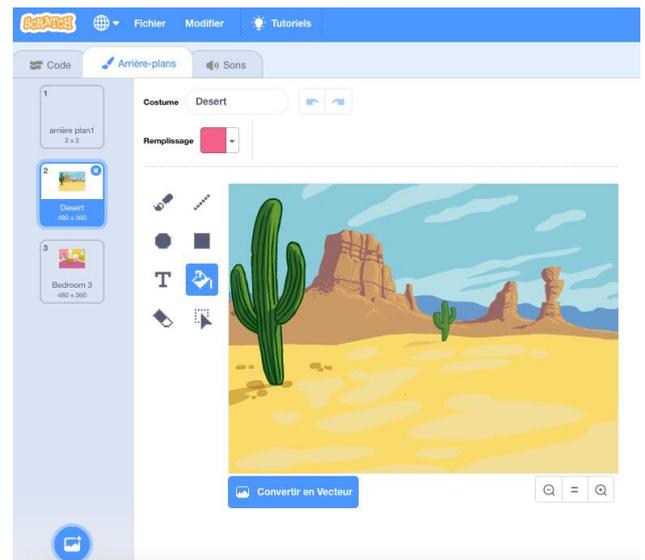
Notez que le nouvel arrière-plan apparaît dans une liste d'arrière-plan pour votre projet sur le côté gauche de l'écran.

Pour revenir à un arrière-plan précédent ou un nouveau, amenez simplement votre curseur sur l'image d'arrière-plan appropriée et cliquez.

La nouvelle image est maintenant en surbrillance, c'est l'élément actif et il devient l'arrière-plan qui apparaît sur l'écran Scène.

Répétez ensuite les instructions pour que le sprite saute sur et depuis un objet figurant dans l'image sélectionnée.

Remarque : l'utilisateur doit cliquer sur le sprite du chat avant qu'elle ne soit mise en surbrillance (en bleu) lorsque le code peut être ajouté.

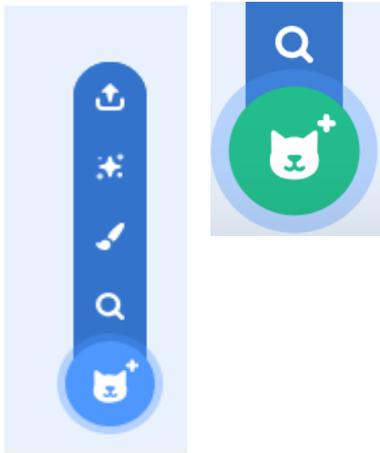


Sauvegardez votre nouveau fichier programme avec le nom ChatSaute2 en sélectionnant **Sauvegardez sur votre ordinateur** dans le menu déroulant **Fichier** situé en haut à gauche de l'écran Scratch.

Ajout d'un oiseau volant

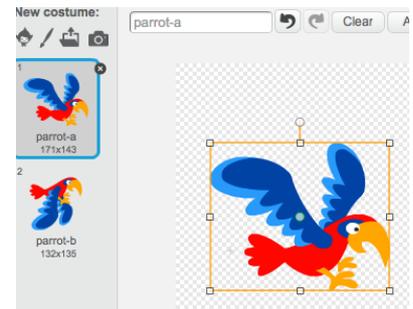
Pour cette fonction, nous plaçons un oiseau qui vole continuellement à travers la pièce, près du plafond, de manière à être hors de portée du chat.

Sélectionnez un nouveau sprite en cliquant sur n'importe quelle icône illustrée ci-dessous



Sélectionnez un oiseau avec deux changements de costume (afin de donner une impression de mouvement d'aile).

Insérez le code suivant :



Pour l'ajout de son, insérez le script supplémentaire suivant :



Exercice

Marcher le long d'une route.

Dessinez un nouveau sprite, par exemple un garçon ou une fille.

Dessinez différents costumes (versions).

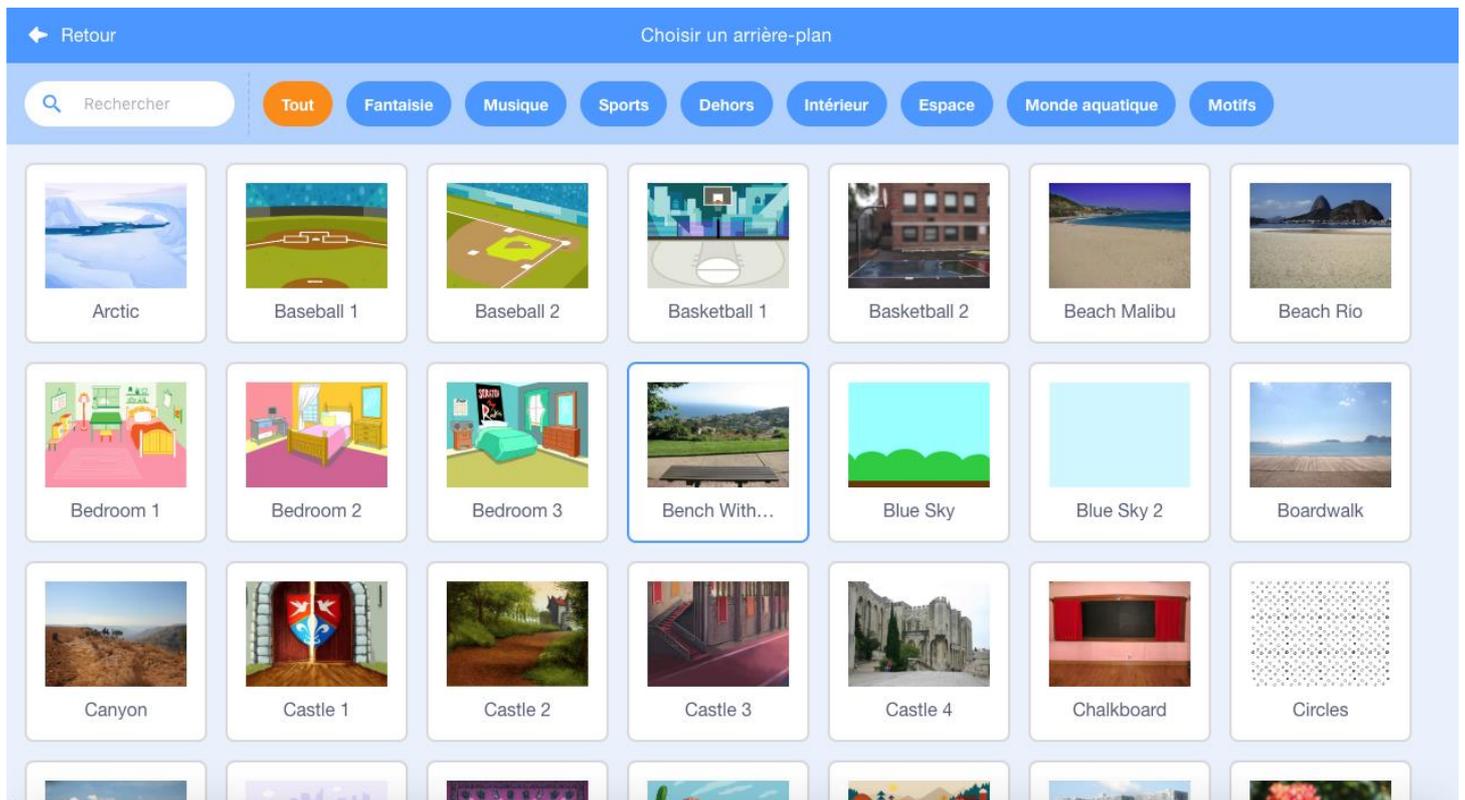
Dessinez une scène de rue (arrière-plan).

Écrivez un programme pour avoir un garçon ou une fille marchant le long d'une rue.

Dessinez d'autres sprites ayant également des scripts leur permettant de marcher, courir et faire du vélo.

Mettez un oiseau qui vole constamment en va-et-vient dans le ciel.

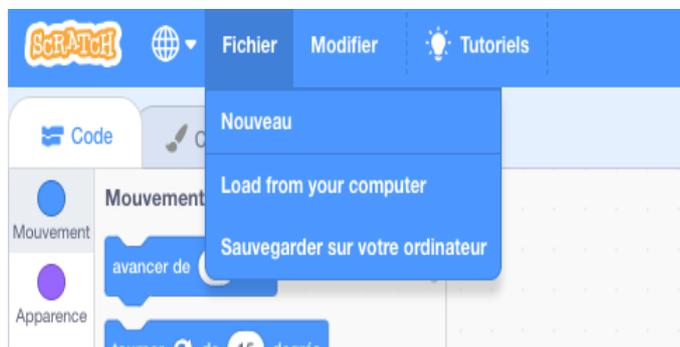
Remarque : avant l'exercice ci-dessus, montrez aux élèves comment une photo peut être importée en tant « qu'arrière-plan » à partir d'Internet ou de l'ordinateur, et comment elle peut être modifiée en utilisant les outils de **l'éditeur de dessin**.



Leçon 7 - Un chien qui aboie et poursuit le chat !

Programmons un chien qui aboie chassant Fionn, notre adorable chat !

Allez dans le menu Fichier



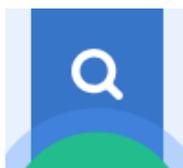
Et choisissez **Load from your computer**

Puis cliquez sur **ChatSaute2** pour ouvrir votre fichier Scratch précédent

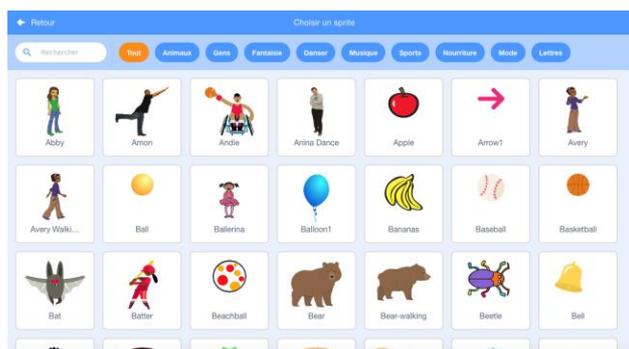
Allez sur



Suivi de

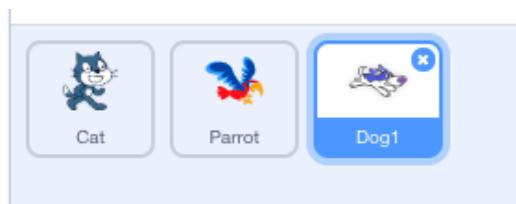


L'écran suivant apparaît



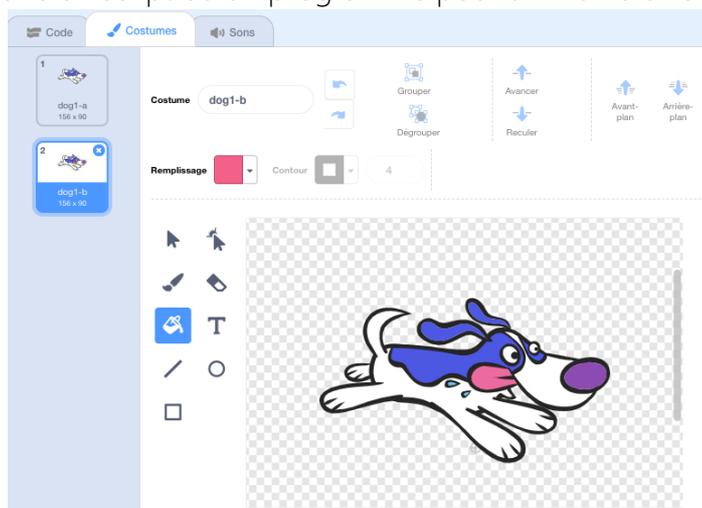
Puis allez dans le dossier Animaux et choisissez l'image du premier chien qui cour (Dog1).

Remarquez que l'icône du Chien est maintenant en surbrillance bleue sur le contour indiquant que c'est le sprite qui est actif.



Nous devons à présent construire un script ou un programme pour animer le chien lui permettant de courir et de parler.

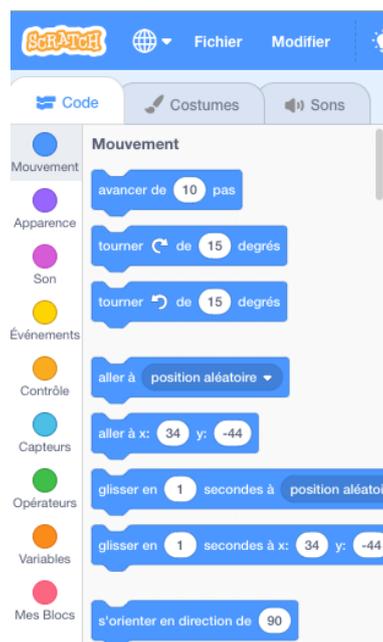
Donc, pour ce faire, nous avons besoin d'au moins une autre version du chien qui, avec le sprite de chien actuel, donnera une impression de mouvement.



En sélectionnant les costumes on voit qu'il existe déjà deux versions utilisables dans le programme.

Construire un script pour animer le Chien

Allez sur les catégories des blocs sur la gauche de l'écran Scratch.



Cliquez sur le dossier Événements

Insérez le bloc suivant dans la zone de script :



Cela indiquera cliquera sur le dessus, le sprite commandes insérées dans la zone de script.

que lorsque l'on Drapeau vert au-suivra les

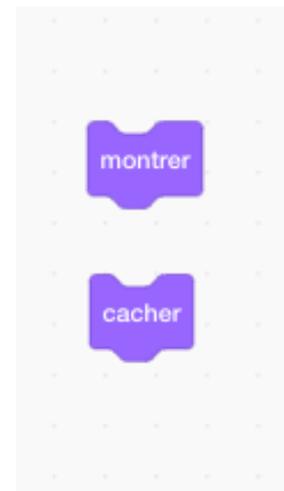
Cependant, souvenez-vous que nous avons maintenant deux sprites, à savoir le chat et le chien.

Nous voulons que le chien apparaisse sur la scène (l'écran) seulement après que le chat ait sauté sur le lit.

Par conséquent, nous devons le cacher lorsque le programme commence et le faire apparaître (être visible) dans la scène qu'à un certain endroit (X et Y) et qu'après un certain temps (attente).

Les blocs suivants, dans une séquence de combinaisons, permettront à l'opérateur d'effectuer cette tâche :

Les blocs Montrer et Cacher dans la catégorie Apparences

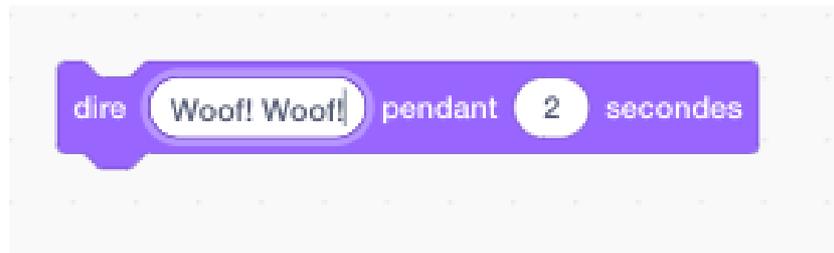


Et le bloc Attendre dans la catégorie Contrôle

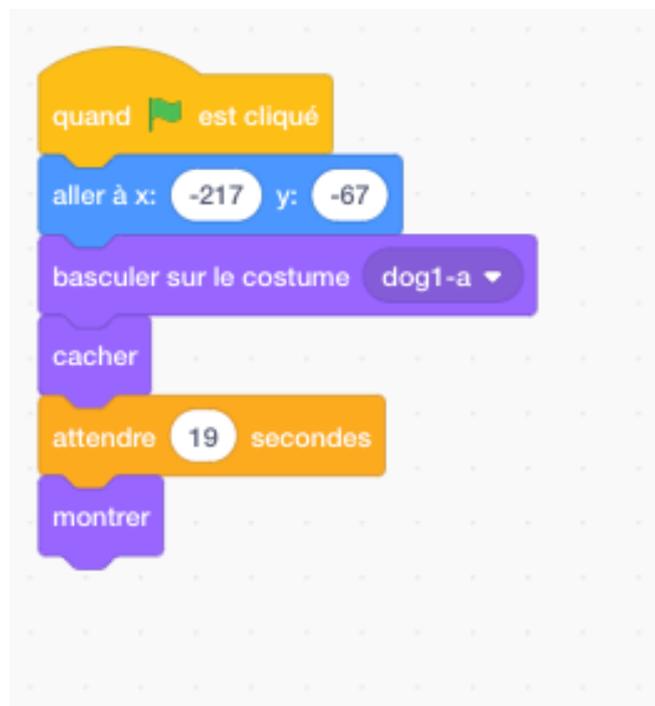


Ainsi que le bloc des coordonnées X et Y dans la catégorie Mouvement. Les coordonnées doivent correspondre à une position à l'extrême gauche du chat (sur le même axe X) et à proximité du bord de l'écran.

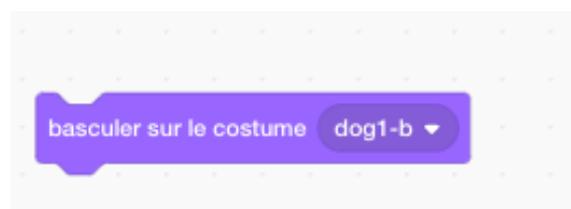
Faites aboyer le chien en sélectionnant le bloc Dire _____ dans la catégorie Apparences



Pour donner un mouvement au sprite, nous devons d'abord insérer un bloc « Basculer sur le costume » vers le début du script afin que la première version du chien, à savoir dog1-a, apparaisse systématiquement lorsque le programme démarre.



Après ce premier Mouvement (dans ce cas-ci le bloc Aller à x ___ et y___) et le bloc Attendre, situé dans le script, mettez un bloc « Basculer sur le costume » avec dog1-b pris dans la catégorie Apparences.

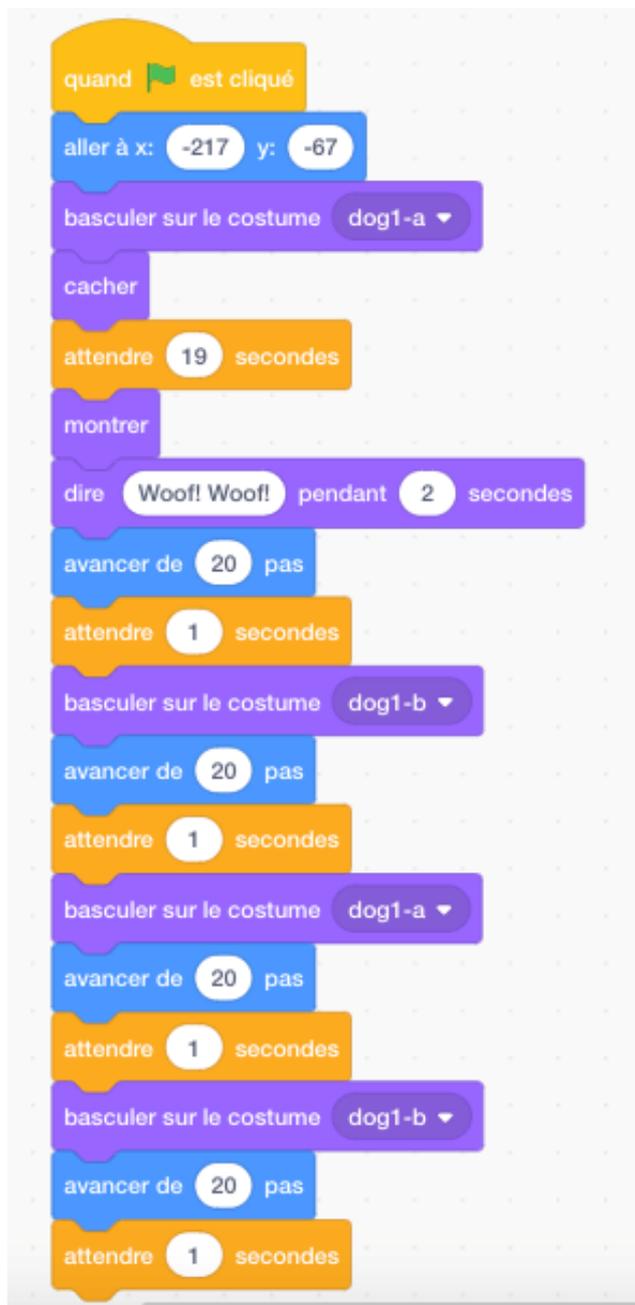


L'option dog1-b est choisie en cliquant sur l'icône de triangle blanc située à droite de ce bloc.

Puis, pour les prochains blocs Avancer et Attendre, mettez un « Basculer sur le costume » pour activer le bloc costume avec dog1-a.

Répétez ce procédé dans tout le script, en alternant ainsi successivement entre dog1-a et dog1-b.

Le résultat devrait être comme dans l'illustration ci-dessous :



Un chat effrayé !

Afin de donner plus d'authenticité au projet, le chat Fionn doit crier «Au secours!» et afficher un visage triste lorsque le chien apparaît et  avant qu'il ne saute sur le lit.

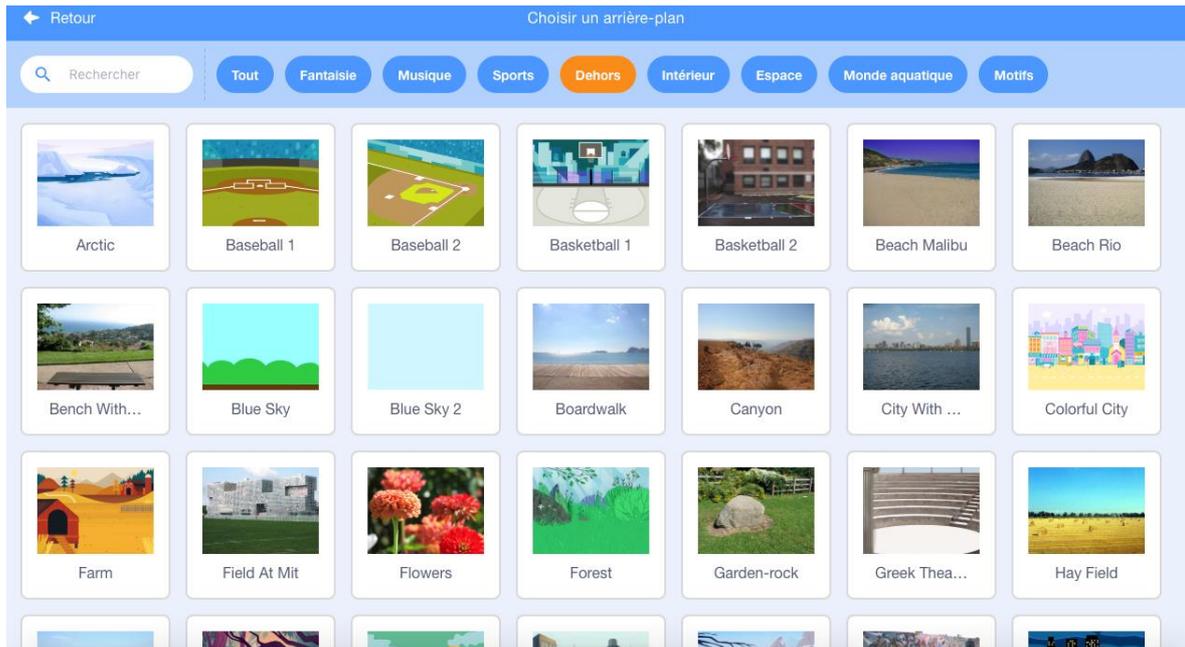
Alors qu'il est sur le lit, le visage du chat pourrait avoir plein de larmes !

Par conséquent, demandez aux élèves de créer un changement de costume supplémentaire pour le chat et d'insérer le code approprié dans la Zone de Script.

Leçon 8 - Créer un sprite

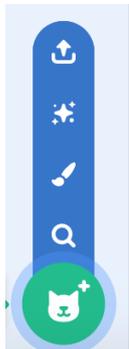
Dans cette leçon, les élèves se familiariseront avec le puissant éditeur de dessin de Scratch, qui permet aux utilisateurs de créer leurs propres sprites.

Cliquez sur l'icône Arrière-plan dans la section Scène en bas à droite de l'écran et sélectionnez la scène Beach Malibu dans la section Dehors de la bibliothèque d'arrière-plan.



Vous pouvez insérer le sprite chat dans la scène de la plage. Mais déplacez-le dans le coin inférieur gauche de l'écran, en saisissant un script qui le fera défiler sur la plage.

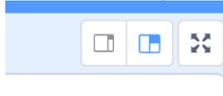
Maintenant allez dans la section des nouveaux sprites située à droite sous la scène.



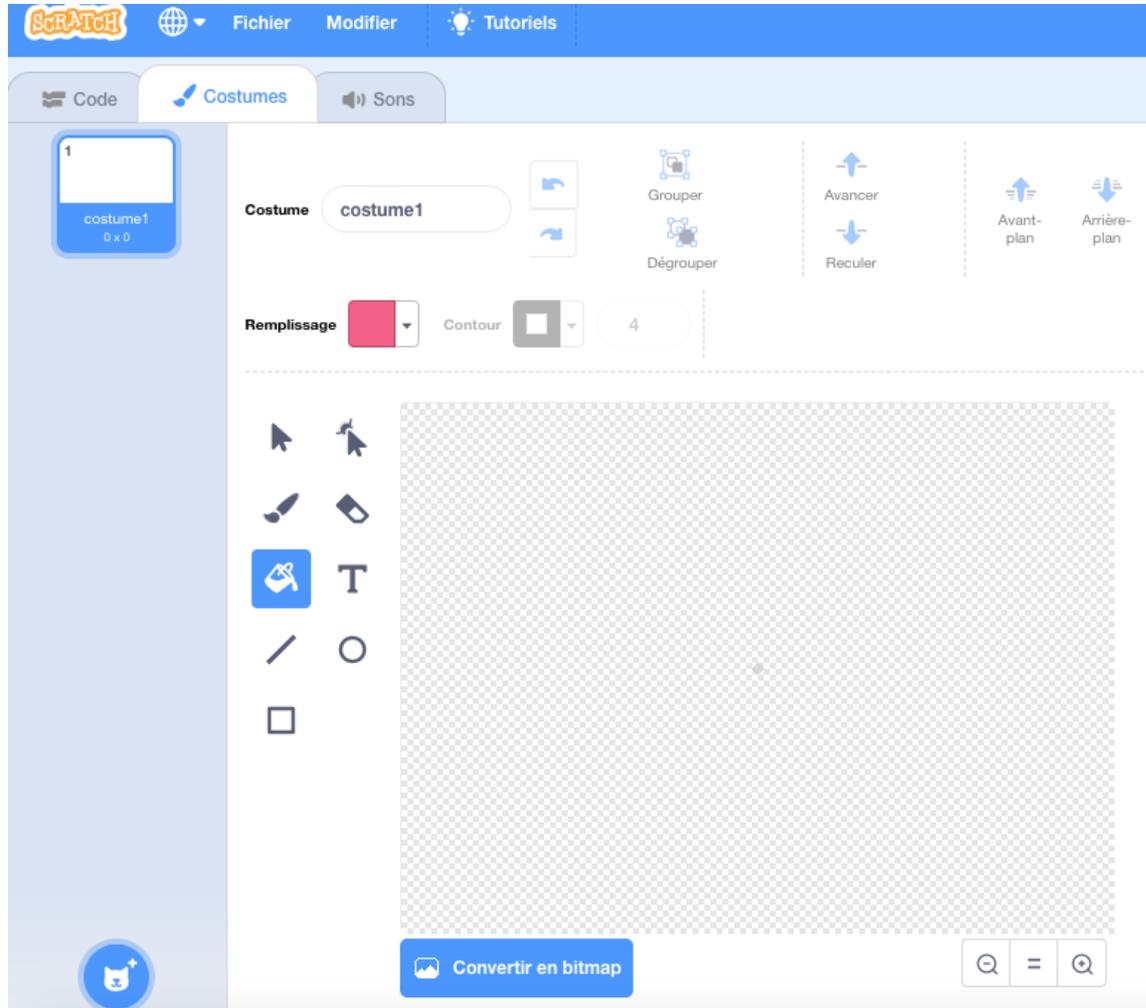
Sélectionnez l'icône pinceau.



Réduisez la taille de la Scène en cliquant sur l'icône du milieu en haut à droite de la scène afin que l'Éditeur de dessin soit prioritaire.



L'Éditeur de dessin est illustré ci-dessous



Assurez-vous que l'Éditeur est en mode Bitmap (en bas à gauche).

Ce programme de dessin intégré est un système de dessin puissant et facile à utiliser.

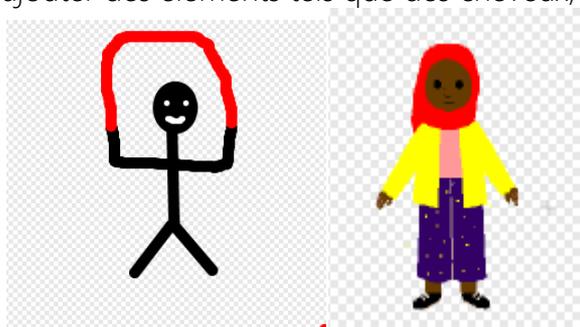
Familiarisez-vous avec toutes ses fonctionnalités clés : palette de couleurs, effacement, pinceau, etc.

Maintenant dessinez un sprite bonhomme allumette qui saute à l'aide de la palette de couleurs (**remplissage**), des options de cercle, de trait et de pinceau.

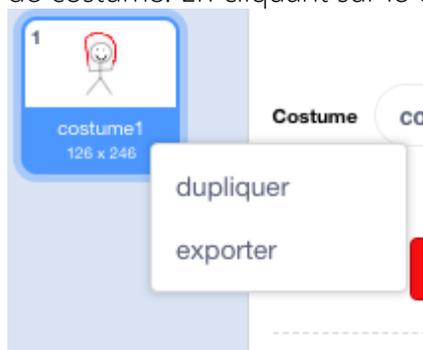


Assurez-vous que le dessin est au centre de l'écran pixelisé.

Le sprite peut être aussi simple ou détaillé que vous le souhaitez. Ainsi, vous pouvez ajouter des éléments tels que des cheveux, des yeux, une bouche, etc.



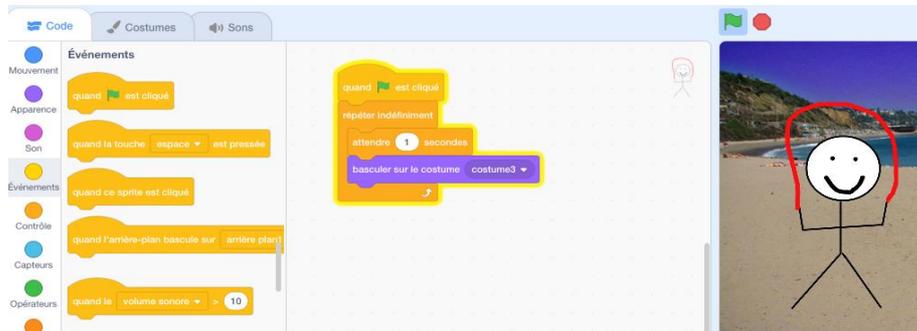
Une fois le bonhomme allumette terminé, allez sur sa petite représentation iconique de costume. En cliquant sur le clavier ou avec le clic droit de la souris dupliquez-le.



Modifiez cette seconde version du costume de manière à ce que les bras, la corde et les autres caractéristiques du sprite soient positionnés différemment de la première version.



Une fois terminé, utilisez un script simple pour animer le sprite à sautant.



Leçon 9 - Les sprites contrôlés par curseur

Dans cette leçon, les élèves se familiariseront avec la programmation des sprites pour suivre les mouvements du curseur.

Oiseau tropical sur la plage

Sélectionnez la scène Beach Malibu et un sprite à deux costumes depuis les bibliothèques arrière-plan/sprite



Utilisez l'option **Pointeur de souris** situé dans le bloc **s'orienter vers _____** (Mouvement) et construisez le script suivant :



Changez les chiffres dans le bloc **attendre _____secondes**.

Remarquez l'effet sur le mouvement du perroquet.

Pour empêcher l'oiseau de voler à l'envers, utilisez l'option du milieu dans la section **Rotation** :



Exercice :

1. Dessinez votre propre sprite oiseau avec deux changements de costume utilisant une scène dans la section Dehors de la librairie Arrière-plan.
2. Créer un programme avec le sprite contrôlé par le code Pointeur de souris.

Leçon 10 - Le sprite psychédélique

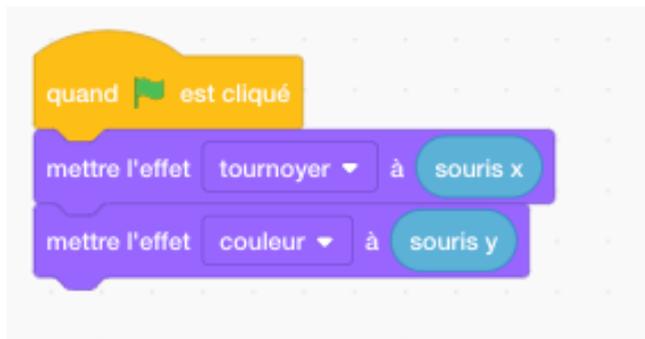
Cette leçon présente aux élèves la catégorie **Capteurs** qui nous permettra dans ce cas-ci de modifier l'apparence physique du Sprite par des mouvements de la souris (ou du clavier).

Sélectionnez un Sprite, de préférence un corps multicolore tel que le Butterfly 2.

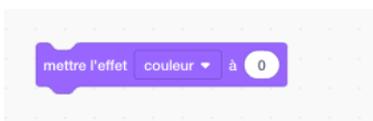


Allez à la section Code du sprite

Insérez dans la Zone de script les blocs code comme illustré ci-dessous :



Le bloc ci-contre, comme illustré dans le script ci-dessus, est situé dans la catégorie Apparences.



Allez dans la catégorie Capteurs pour trouver les blocs Souris X et Souris Y que vous déposerez dans le bloc **Mettre l'effet _____ à _____** comme illustré dans le script ci-dessus.

Lancez le programme en cliquant sur le Drapeau vert en haut à droite de l'écran.

En raison des instructions que vous avez saisies, qui déplacent le pointeur de la souris le long de l'axe X, le sprite changera de forme.

Si vous déplacez la souris le long de l'axe Y, le sprite changera de couleur.

Exercice

Testez d'autres effets en cliquant sur la flèche blanche dans le 'mettre l'effet... à' et choisissez d'autres options que Tournoyer et Couleur.

Leçon 11 - Interaction du sprite

Cette leçon s'appuie sur vos connaissances des commandes de **Capteurs** acquises à la leçon 9 pour montrer comment coder une interaction automatique entre deux sprites à l'aide du bloc de code **Touche**.

Tout d'abord, supprimez le chat.

Choisissez Max depuis la catégorie Gens de la librairie des sprites.



Puis choisissez le trampoline comme second sprite.

Sous la section Costumes de Max, supprimez toutes les versions du sprite excepté max-b et max-c.

Ces deux versions devraient suffirent pour animer Max sautant de haut en bas sur le trampoline.

Cependant, vous pouvez bien sûr utiliser l'Éditeur de dessin pour changer un ou plusieurs costumes afin de visualiser les bras encore plus haut ou plus bas. Ceci peut être réalisé en utilisant l'outil en surbrillance de l'illustration ci-dessous.

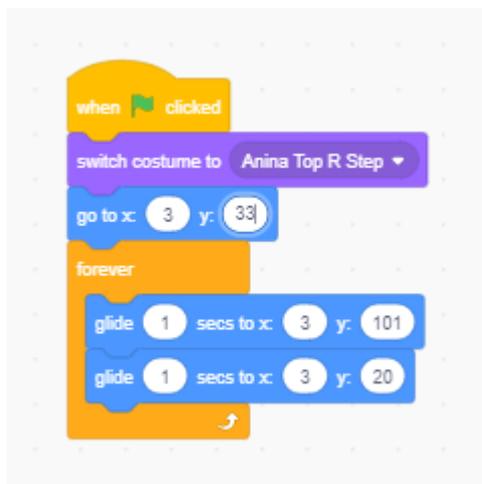


Nous voulons bien sûr que Max saute sur et en bas du trampoline.

Comme nous l'avons fait dans une leçon précédente, nous donnons l'impression de sauter en utilisant le bloc glisser depuis la catégorie **Mouvement**.

Mais cette fois-ci nous utilisons deux blocs Glisser car nous voulons qu'elle monte à un certain point (axe Y) avant qu'elle ne descende toucher le trampoline.

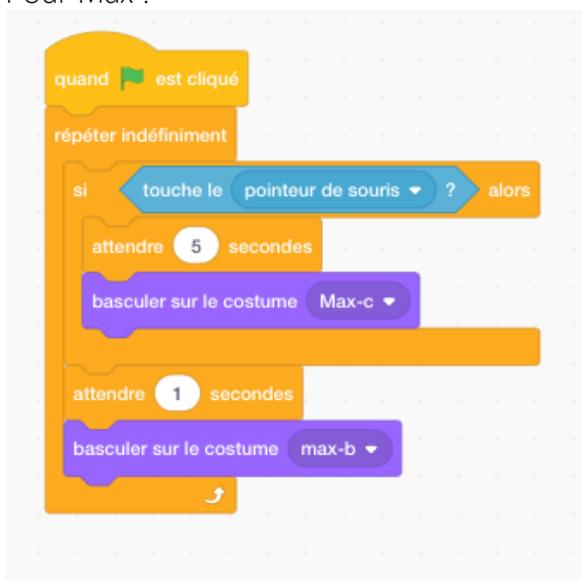
Ceci est réalisé par le deuxième bloc Glisser ayant une coordonnée Y inférieure (nombre).



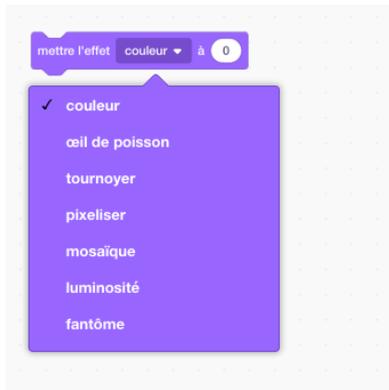
Quand Max touchera le trampoline, ses bras se lèveront. En même temps, la surface du trampoline réagira au poids de Max en s'étirant vers le bas jusqu'à ce qu'elle saute vers le haut à nouveau.

Pour réaliser cet effet nous insérons la commande **Touche** depuis la catégorie **Capteurs** dans les scripts de Max et du trampoline.

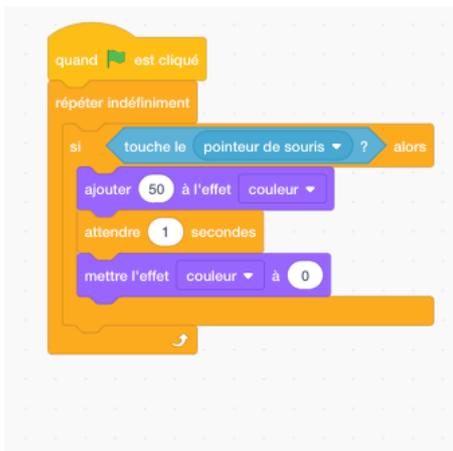
Pour Max :



Pour le trampoline, l'utilisation de l'effet œil de poisson de la catégorie **Apparences** donne l'impression d'un renflement.



Le script est comme ci-dessous



Veillez noter que l'effet œil de poisson doit être ramené à zéro sinon le trampoline continuera à se bomber chaque fois que Max le touchera.

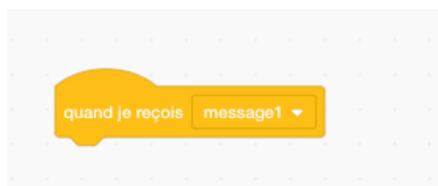
Exercice

1. Créer une scène complète d'aire de jeux avec une balançoire classique et une autre à bascule.
2. Dessinez deux enfants placés à chaque extrémité de la balançoire à bascule
3. Donnez une impression de mouvement à la balançoire à bascule.
4. Dessinez un enfant sur la balançoire classique.
5. Donnez une impression de mouvement à la balançoire.
6. Dessinez un garçon ou une fille sautant. Ces personnages peuvent être de simples bonhommes allumettes.

Leçon 12 - Deux sprites ayant une conversation

Dans cette leçon, les élèves se familiariseront avec le code de commande **Envoyer** qui envoie un message à une autre partie du programme (par exemple un autre sprite ou arrière-plan) lui demandant d'appliquer un changement.

« Envoyer » est utilisé pour permettre la communication entre sprites et autres éléments de Scratch. La réponse du sprite destinataire sera déclenchée par une insertion dans un morceau de code connu sous le nom de



Sélectionnez deux fois le sprite Butterfly 2 depuis la librairie de Scratch. Choisissez un arrière-plan approprié.



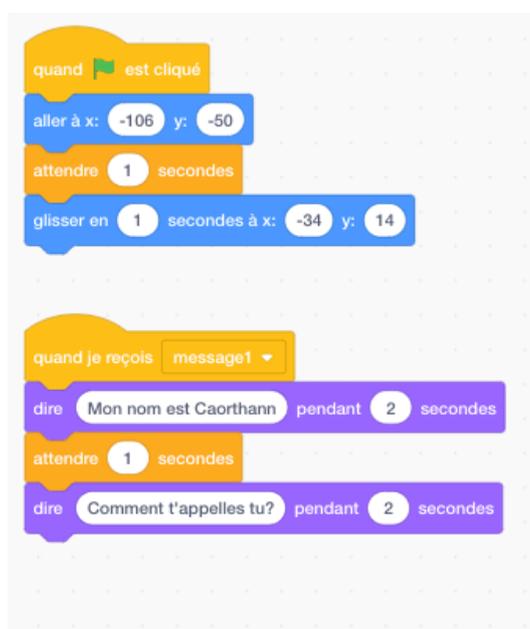
Veillez noter que si votre sprite est tourné dans le mauvais sens il faut cliquer sur l'onglet costume.

Dans l'Éditeur de dessin, cliquez sur l'icône Flip horizontal (gauche-droite)

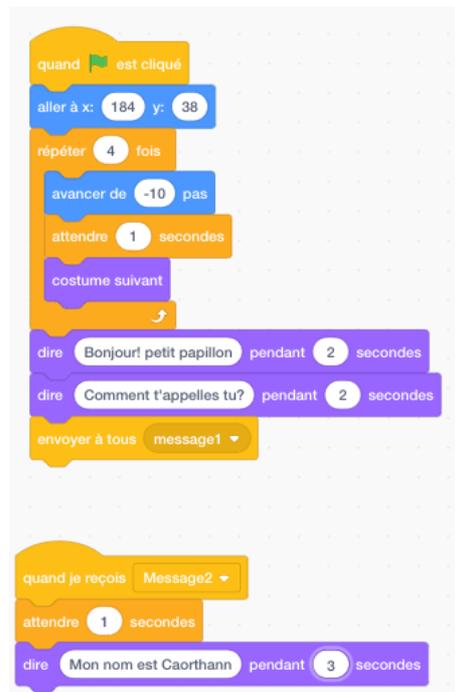


en mode bitmap pour inverser la direction du sprite.

Insérez le code suivant dans le premier sprite Butterfly 2



Insérez le code suivant dans le deuxième sprite Butterfly 2



Continuez ce genre de codage jusqu'à ce que la conversation soit terminée.

Exercice

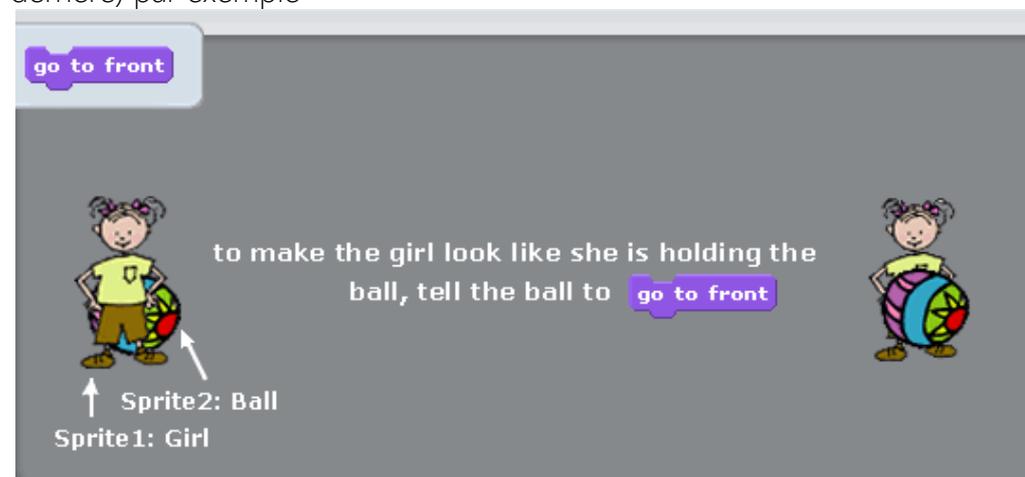
1. Créez un programme où deux personnes discutent après une réunion, tout en marchant.

Leçon 13 - Créer un récif de corail

Objectifs de la leçon :

- De comprendre la signification de la commande **Si** (avec son impact de cause à effet) qui se trouve dans la catégorie **Contrôle**.
- D'utiliser le bloc de code **Nombre Aléatoire** de la catégorie **Opérateurs** pour changer la position du sprite
- D'utiliser la commande **Capteurs** pour donner différents effets lorsque les sprite entre en contact
- D'ajouter une partition de **musique**.

Expliquez que le '**aller à l'avant plan**' est très utile lorsqu'il y a deux sprites placés l'un au-dessus de l'autre et que vous voulez que l'un apparaisse toujours devant (ou derrière) par exemple



Les blocs de code '**aller à l'arrière-plan**' peuvent être appliqués lorsque vous avez plusieurs objets à l'écran tel qu'un aquarium, peuplé de plusieurs poissons et plantes et où vous devez placer certains objets à l'arrière alors que d'autre seraient à l'avant lorsqu'ils se croisent.

2^{ème} Partie : Créer un monde marin

Créons donc maintenant notre propre monde marin tropical coloré.

Dans le menu Fichier aller sur Nouveau

Supprimez le sprite Chat

Sélectionnez le décor Underwater dans le dossier Monde aquatique

Allez au menu des sprites



et cliquez sur



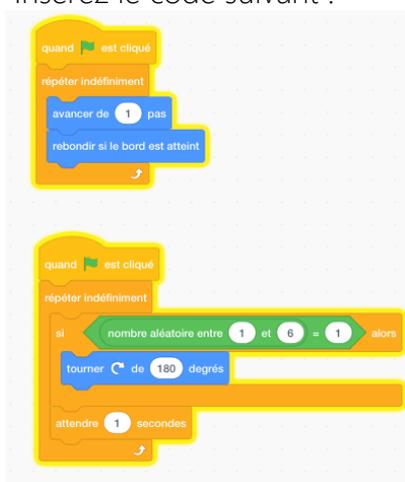
pour accéder à la librairie des sprites.

Choisissez le sprite Shark depuis le dossier Animaux.

Choisissez le sprite Fish depuis le dossier Animaux et dupliquez-le.

Supprimer tous les costumes sauf un pour chaque poisson. Assurez-vous que chacun des deux sprites de poisson utilise un costume différent afin que chaque poisson soit une espèce différente.

Insérez le code suivant :



Plutôt que de faire changer de direction les sprites (par exemple seulement se retourner) lorsqu'ils « rebondissent » si le bord est atteint, l'unique code que nous avons utilisé jusqu'à présent, ce dernier morceau de code permet au sprite de tourner (par exemple de 180 degrés ou de gauche à droite) de façon aléatoire dans une certaine fourchette (par exemple de 1 à 26).

L'utilisation de

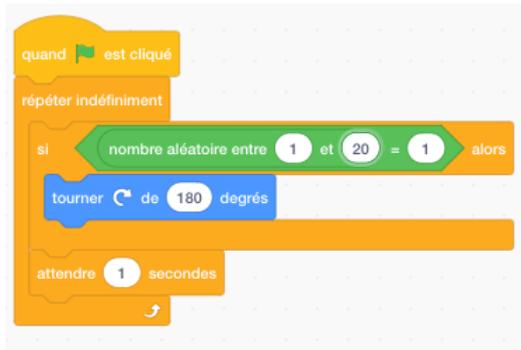


combiné avec



situé dans le dossier **Opérateurs** signifie que ce changement de direction aura lieu une fois tous les vingt. Donc, si la fourchette est plus petite (par exemple de 1 à 6), le sprite tournera plus souvent.

L'utilisation de la puissante commande **SI ___ ALORS** dans le script signifie que si une certaine inscription saisie est générée, la réaction indiquée correspondante contenue dans le code aura automatiquement lieu.



L'utilisation du bloc **Répété indéfiniment** avec le bloc de **Si ___ Alors** assure que ce changement de sens du sprite sera une action continue.

Exercice

Demandez aux étudiants de tester des modifications de « Nombres aléatoires » et du bloc « Tourner de ___ degrés ».

Utilisez le même code que pour le sprite précédent (ci-dessus)

Ajout d'algues et d'anémones

Pour donner au programme une ambiance plus authentique de récif de corail vivant, des anémones, des algues ou des coraux doivent être ajoutés à la scène.



Dessinez un sprite d'algue en cliquant sur l'icône

Puis utilisez l'outil pinceau depuis le menu sur la gauche pour créer une algue.

Créer deux ou trois d'algue, chaque version en position différente afin de donner un mouvement de balancement lorsqu'elle sera animée par le codage.



versions de costumes pour le sprite

Insérez le codage

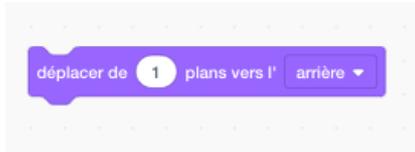
suivant :

Dessinez un ou deux sprites d'anémones.

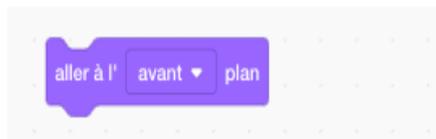
Créer un effet 3D : Utiliser 'Plan'

Cependant, vous remarquerez peut-être que les algues sont toutes placées à l'avant, ce qui bloque la vue des poissons en mouvement.

Pour repositionner certaines algues vers l'arrière, donnant ainsi l'effet que le poisson se déplace devant et entre des plantes individuelles, allez à Apparences et sélectionnez le bloc suivant :

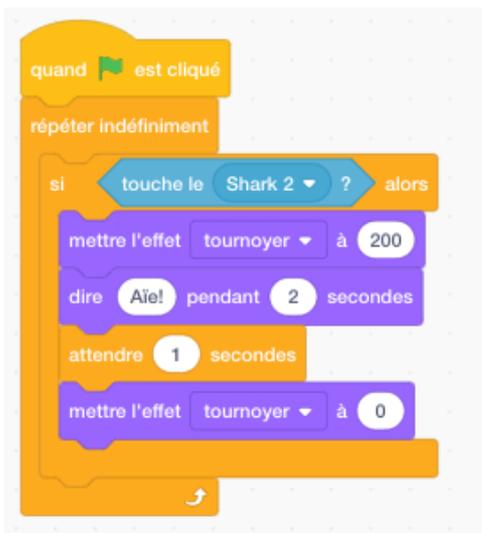


Chaque plan spécifique de Plus le nombre est marins nagent devant les algues.



représente une zone mouvement du sprite. élevé, plus les sprites

Pour qu'une algue soit située à l'avant de l'écran, utilisez le bloc de mouvement suivant :



Lorsque les créatures marines entrent en collision

La catégorie Capteurs contient des blocs de code qui permettent aux sprites d'interagir entre eux.

Sélectionnez un sprite dont le code lui permet de croiser le chemin d'un autre sprite.

Ajoutez le code suivant à ce sprite :

« Tournoyer », l'effet spécial susmentionné et utilisé dans l'exemple ci-dessus, modifiera le nombre

Mais il est après un bloc

tournoyer

Sinon, le « Le » effets situés dans



la forme physique du sprite. Plus le est élevé, plus la distorsion du sprite est importante.

important d'ajouter au bloc de code, une attente d'une ou deux secondes, le supplémentaire

mettre l'effet à 0 ou annuler les effets graphiques.

sprite restera indéfiniment distordu.

tournoyer » est l'un des nombreux spéciaux de ce bloc de commande

Apparences, comme illustré ci-dessous :



Demandez aux étudiants de tester ces options et de changer le nombre dans le mettre l'effet tournoyer à

Un son ambiant pour l'aquarium

Enfin, sélectionnez une musique adéquate pour aller avec l'ambiance des poissons qui se déplacent lentement.

Exercice

Demandez aux élèves de dessiner :

- Une scène de forêt ou de jungle peuplée de nombreux animaux en mouvement tels que des oiseaux, des mammifères et des insectes ou
- Une scène de rue peuplée de marcheurs, de gens faisant les magasins, de cyclistes, d'artistes de rue, etc.

Sauvegarder les sprites que vous avez créés

Les sprites créés dans un projet peuvent être sauvegardés afin d'être utilisés dans d'autres projets.

Amener le curseur sur le sprite que vous voulez sauvegarder.



Sélectionnez exporter ce sprite.

Puis choisissez l'emplacement où vous voulez sauvegarder le sprite (par exemple : bureau, lutin Scratch, dossier « Objets »), suivi d'un nom puis sauvegardez.

Leçon 14 – La balle cible

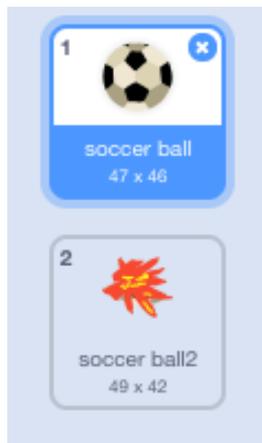
Objectifs de la leçon :

- D'apprendre l'utilisation de **variables**
- D'utiliser le bloc de code **Nombre aléatoire (Opérateurs)** pour changer la position du sprite.
- De montrer comment une action peut être programmée pour se produire lorsque l'on clique sur un sprite.

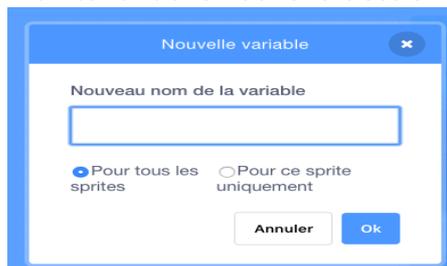
Balle cible - résumé du plan de codage

La balle se déplace à une vitesse aléatoire à travers la scène. L'utilisateur essaie de cibler la balle en enregistrant un score pour chaque frappe.

Faites un deuxième costume du sprite balle qui donne l'impression qu'elle explose.



Maintenant allez dans le dossier **Variables**, sélectionnez l'option Créer une variable.

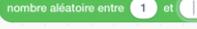


Tapez Score sous le nom de la variable et choisissez Pour tous les sprites. Puis placez le bloc suivant dans la zone de script réglant le score à zéro.

Maintenant nous voulons que le sprite se déplace à travers la scène selon un modèle défini. Donc, nous utilisons le bloc dans Opérateurs



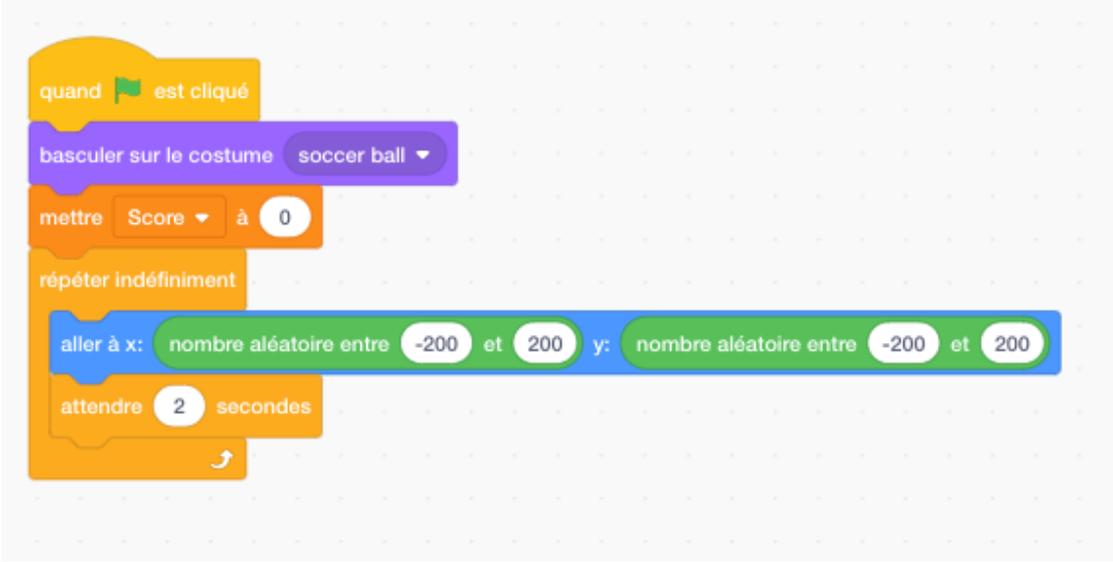
 pour les coordonnées X (horizontal) et Y (vertical). Les nombres aléatoires sont des nombres qui ne constituent aucun modèle et sont imprévisibles.

Mais les deux cases blanches dans  permettent à l'utilisateur de définir une fourchette pour X et Y. Donc, dans ce cas-ci le sprite se déplacera à travers un maximum d'espace sur la scène en utilisant le bloc suivant



Pour s'assurer que le mouvement aléatoire soit continu utilisez le bloc Répéter indéfiniment.

Cependant, il doit être utilisé conjointement avec . Sinon, la vitesse du chat rendrait très difficile l'enregistrement d'un score.



Maintenant, nous entrons dans le script un morceau de code distinct qui donne un score à chaque fois que la balle est frappée.

L'utilisation de Montrer et Cacher ci-contre signifie que la balle disparaît chaque fois qu'elle est frappée. Bien sûr, cela signifie que le jeu est très court et se termine lorsque le sprite est touché une fois.

Donc, afin de rendre le jeu plus long et plus amusant, copiez le sprite cinq ou six fois. Puis, donnez à chacun de ces sprites de balles un ensemble différent de coordonnées X et Y afin qu'elles se trouvent toutes à des emplacements différents au début du jeu.

Leçon 15 - Les sprites dansants

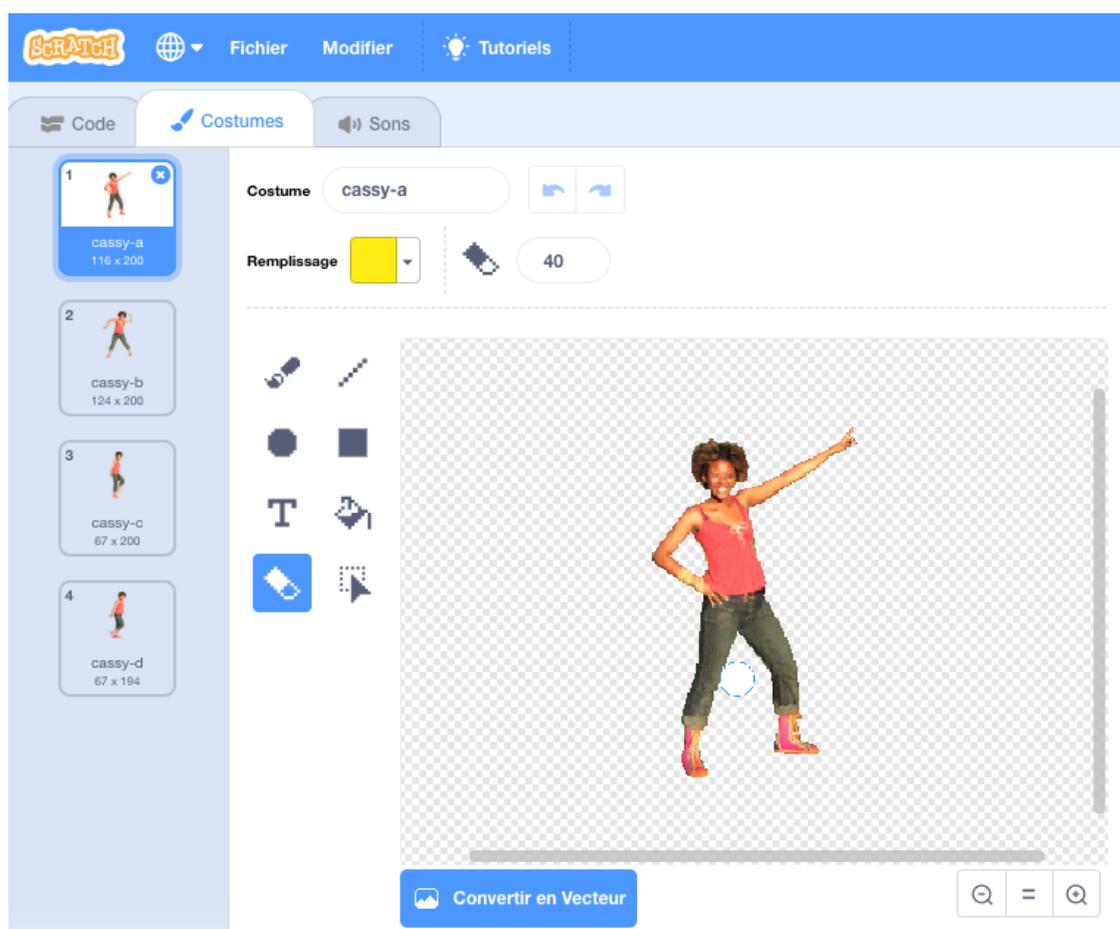
Dans cette leçon, les étudiants apprennent à créer une animation de danse à travers un ensemble de commandes depuis les dossiers Apparences, Contrôle et Son.

Premièrement, cliquez sur l'icône Scène située sous la section Scène de l'écran Scratch.



Allez sur  et choisissez un arrière-plan approprié pour la séquence de danse que vous allez créer.

Puis choisissez un nouveau sprite qui a un choix de différentes postures. Ensuite allez dans la section costume et importez un nombre de différentes versions (ostumes) du même sprite.



Et maintenant animons !

Sélectionnez d'abord



Suivi du bloc Répéter indéfiniment.

Cette commande laissera le programme de danse s'exécuter de manière continue jusqu'à ce que l'utilisateur sélectionne l'icône **Stop** du cercle rouge située en haut à droite de l'écran Scène.

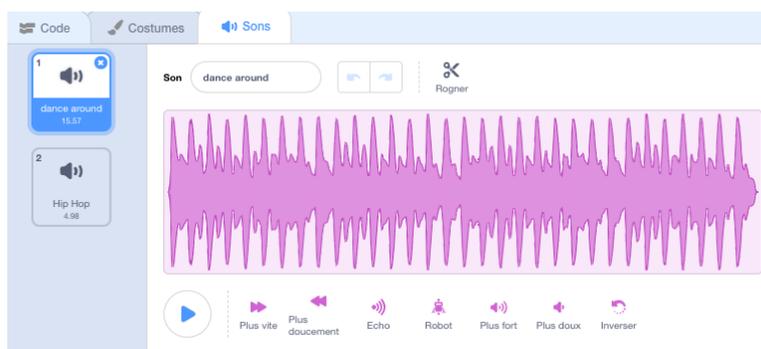
Ensuite sélectionnez la commande Attendre et réduire la durée de 1 seconde à 0.5 secondes sinon le sprite dansera trop lentement.

Allez dans Apparences, sélectionnez Costume suivant et insérez-le dans le bloc Répétez indéfiniment.



Lancez le programme.

Afin d'ajouter un son, allez à , cliquez sur  et sélectionnez un son approprié pour une danse tel que le Hip-Hop dans Boucles.



Dans la section Sprite, ajouter ce qui suit :



Lancez le programme.

Maintenant ajoutez le script suivant pour introduire un effet de lumière disco sur la scène.



Exercice

Demandez aux étudiants de démarrer un nouveau programme avec un arrière-plan différent, une musique différente et trois sprites dansants.

Encouragez-les à dessiner, par exemple, quelques danseurs traditionnels dans différentes postures et de décaler (utiliser le bloc Attendre) leur apparence sur scène.

Recherchez une musique appropriée et créez votre propre comédie musicale !

Leçon 16 - Dessiner des formes

Dans cette leçon, les étudiants apprennent à comprendre comment dessiner des formes géométriques en utilisant une série de scripts. Ils apprennent aussi le concept de Chance en utilisant un bloc de sélection Aléatoire depuis Opérateurs.

Créer un script pour dessiner un carré

Cette fois nous n'avons pas besoin d'un sprite. Mais comme toutes les commandes de Scratch ne peuvent pas fonctionner sans sprite, nous devons le cacher.

Utilisez les commandes suivantes pour faire disparaître le sprite :



Afin de dessiner dans Scratch, nous avons amené le curseur dans le coin inférieur gauche de l'interface et sélectionné l'option Ajouter une extension représentée par l'icône.

Cliquez ensuite sur la section Stylo qui ajoutera automatiquement cette option à vos catégories thématiques de Bloc.

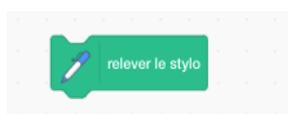
Depuis Stylo, sélectionnez une taille et couleur.

Nous donnons également des instructions au stylo pour

a) commencer



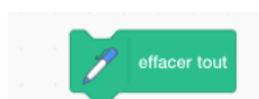
et b) finir



de dessiner.

Il est également important d'effacer les dessins précédents de la scène et de recommencer le processus de dessin sur une toile vierge une fois que le Drapeau vert est sélectionné pour relancer le projet.

Donc, utilisez la commande



Ajouter le bloc « Attendre 1 seconde » permettra au spectateur de mieux apprécier visuellement à la fois le début du dessin et le processus réel de formation d'un nouvel objet.

Pour nous assurer que l'objet géométrique que nous sommes sur le point de dessiner, à savoir un carré, a des dimensions suffisantes pour faciliter la visualisation, utilisez un nombre important d'étapes dans le dossier Mouvement, par exemple : 100 pas.

Pour entrer le code dans la commande qui fera que les quatre lignes forment une boîte, nous utilisons d'abord la commande Répéter. Dans la construction d'un carré c'est **Répéter 4 fois**.

Notez également que pour chaque forme géométrique, l'angle est proportionnel au nombre de côtés, c'est-à-dire 360 degrés divisé par le nombre de côtés. Par exemple, un carré est 360 divisé par quatre = 90 ; Un triangle est 360 divisé par trois = 120 et Un cercle est 360 degrés divisé par 360 = 1



Donc, nous devons tourner les lignes en utilisant un bloc de degré dans la commande de Mouvement

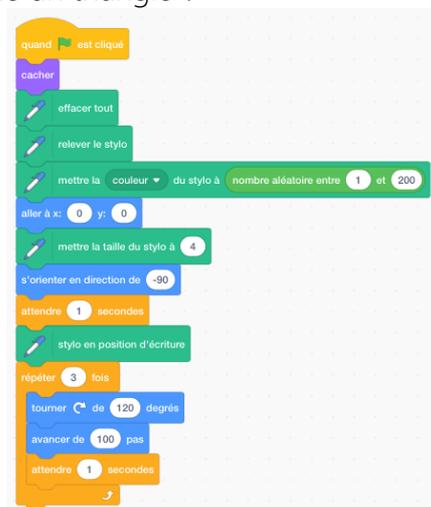
Nous pouvons aussi modifier les couleurs de chaque dessin en choisissant l'option aléatoire dans la catégorie Opérateurs. Le spectre de couleurs va de 1 à 200. Par conséquent, choisir une plage aléatoire élevée, par exemple 1-200 (voir image ci-dessous) pour la variation de couleur permettra au programme de sélectionner de manière aléatoire parmi la gamme complète de couleurs à chaque exécution.



Maintenant, comment faisons-nous un triangle ?

Combien de côté possède un triangle ?

Quels seraient les (degrés) ?



réglages d'angle

Dessignons un **cercle**.

Combien de tours (degrés) dans un cercle ?

Donc, tournez un degré à la fois. N'utilisez pas le bloc  dans le script. En effet, son utilisation signifiera que le cercle aura besoin de 360 secondes pour se terminer.



Ajoutez le code suivant dans la boucle :



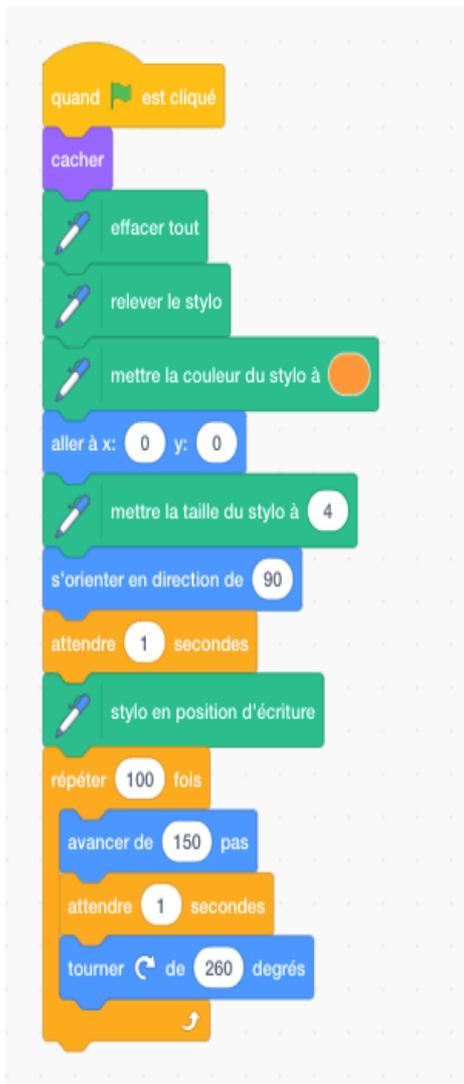
pour donner



et appréciez l'effet coloré.

Exercice

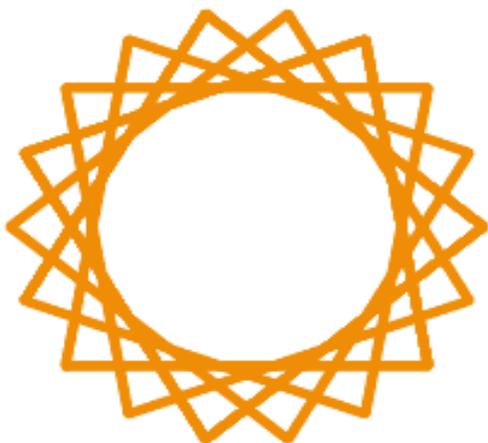
1. Dessinez un pentagone
2. Écrivez un programme qui dessine trois formes différentes qui apparaissent sur la scène à différents moments et à différents endroits.



Formes - autres

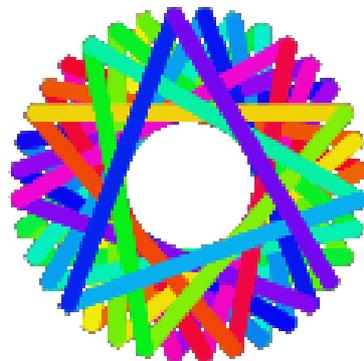
Testez avec différents angles et "Répéter" dans le programme.

Par exemple, sélectionnez le code ci-dessous et admirez les résultats



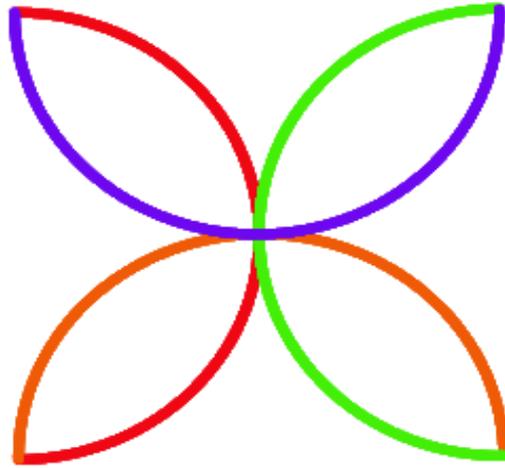


Insérez le code suivant :



pour donner

Créez une fleur à quatre pétales



```
quand est cliqué
  cacher

quand est cliqué
  aller à x: -90 y: 90
  s'orienter en direction de 90
  effacer tout
  répéter 4 fois
    répéter 180 fois
      avancer de 2 pas
      tourner de 1 degrés
    mettre la taille du stylo à 5
    stylo en position d'écriture
    mettre la couleur du stylo à nombre aléatoire entre 1 et 200
    tourner de 90 degrés
```

L'effet est obtenu en faisant quatre demi-cercles (180 degrés) et en tournant à droite (90 degrés) à la fin de chacun.

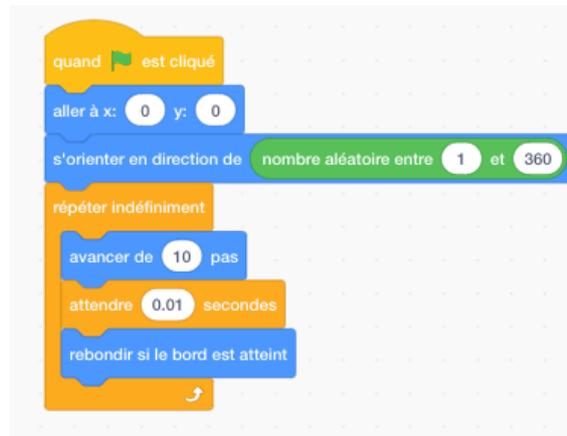
Leçon 17 – La balle qui rebondit

Dans cette leçon, les étudiants se familiariseront avec les *Degrés* et les *Variables de l'espace*.

D'abord dessinez une Balle en utilisant l'option Dessiner un nouveau sprite



Puis écrivez le script suivant :



Testez la vitesse de la balle en augmentant et en diminuant le nombre dans la zone du bloc Avancer de _ pas.

Nous pouvons aussi contrôler la vitesse de la balle durant le jeu en utilisant la catégorie Variables.



variable

Sélectionnez Créer une variable (pour tous les sprites) et tapez le mot Vitesse dans l'option du Nom de

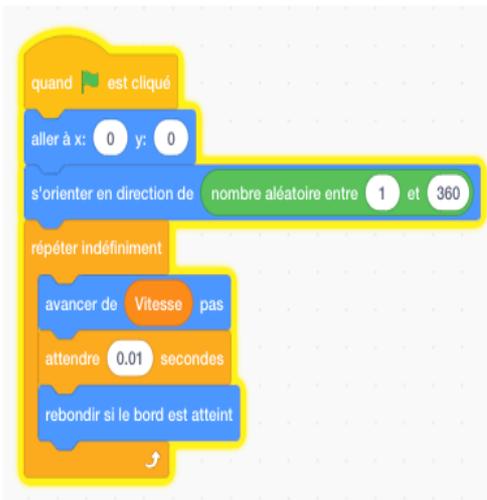
Un bloc Vitesse apparaît sur la scène



Cliquez avec le bouton droit sur l'icône (si vous utilisez la souris) et sélectionnez l'option Curseur, qui permet à l'utilisateur de régler la vitesse de la balle en mouvement.



Tirez le bloc Vitesse de la catégorie Variables et mettez-le dans la boîte blanche (numéro) du bloc **avancer de _____ pas**.



Le code de script se lit maintenant comme suit :



Générer une queue de comète sur la balle qui rebondit

Exercice

1. Écrivez un programme avec plusieurs balles qui rebondissent
2. Écrivez un programme sur le thème de la mer qui présente un certain nombre de créatures marines différentes qui rebondissent, telles que des méduses et des requins.

Leçon 18 - Dessiner à main levée

Dans cette leçon, les étudiants apprennent à créer un programme permettant à un utilisateur de dessiner à main levée.



Tout d'abord, comme pour les programmes pour dessiner des formes, l'utilisateur doit faire disparaître le chat (masquer). En effet, il serait très difficile de dessiner correctement avec un sprite tel qu'un chat utilisé comme outil de dessin.

L'utilisateur utilise ensuite le bloc vide suivi du bloc de taille de stylo défini afin d'avoir un stylo de dessin de taille suffisante.



L'utilisateur peut devenir le crayon à dessiner en codant le mouvement de la souris, comme nous l'avons fait dans une leçon précédente lorsque le mouvement d'un sprite animal était contrôlé par le suivi de la souris.

Mais nous devons maintenant nous assurer que le programme tienne compte du fait que le mouvement de l'utilisateur n'est pas continu. Sinon, l'écran se remplirait d'une ligne sans fin.

Par conséquent, nous devons coder dans un ensemble d'instructions afin que l'artiste puisse lever son stylo à tout moment pour cesser de dessiner et pour continuer à dessiner à un autre endroit de l'écran.

Nous y parvenons en codant dans un script principalement basé sur le bloc



du dossier Apparences et le



du dossier Contrôles.

Cela donnera

Le script final lira comme suit :



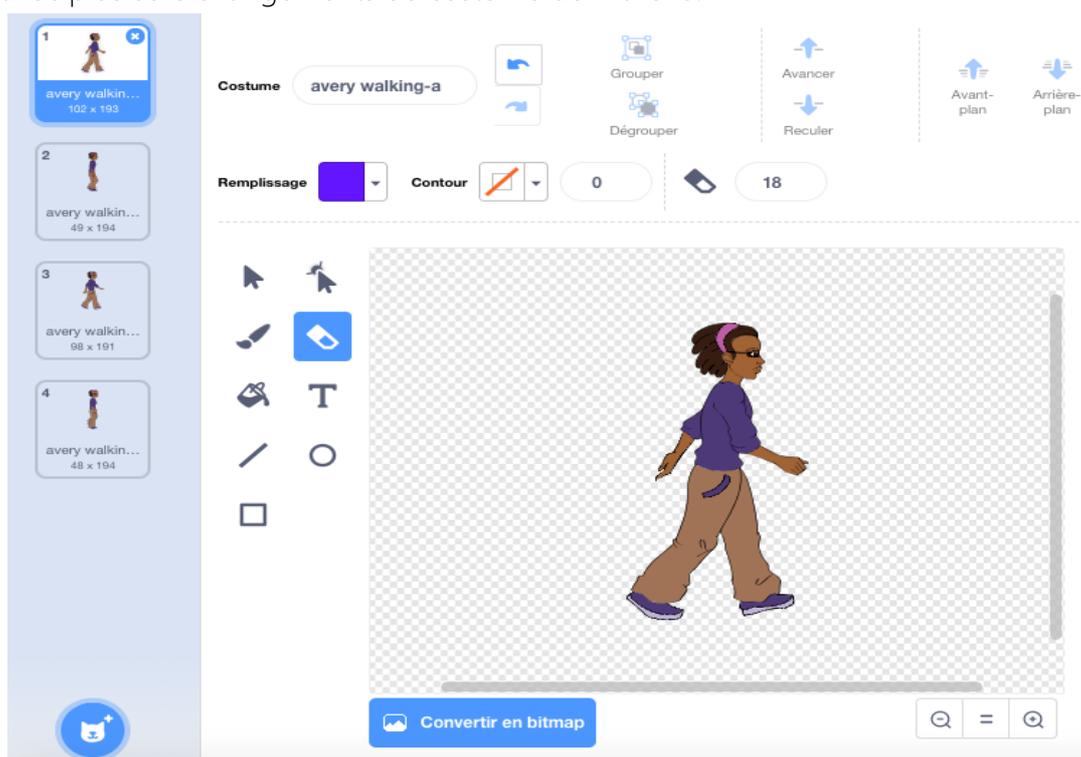
Leçon 19 – Promener le chien

Dans cette leçon, la fonctionnalité de Minuterie est présentée aux élèves. Elle permet de modifier les arrière-plans et les sprites à certains moments spécifiques.

Les élèves utiliseront également le bloc « Point en direction », qui contrôle la direction dans laquelle le sprite regarde et se déplace.

La leçon est basée sur une fille ou un garçon promenant un chien dans différents paysages.

Sélectionnez dans la section Gens de la bibliothèque des sprites un sprite approprié avec plusieurs changements de costume de marche.



Allez dans la bibliothèque d'arrière-plans de l'option Scène et sélectionnez trois scènes différentes dans le dossier Dehors.

Sous la commande



nous ajoutons deux blocs supplémentaires par-dessus et au-dessus de ceux que l'utilisateur connaît déjà des leçons précédentes.

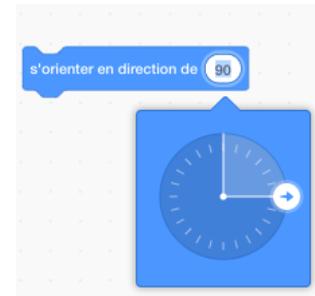
Choisissez d'abord le bloc



depuis le dossier **Capteurs**, ce qui ramène le temps à zéro lorsque l'on clique sur le Drapeau vert.



La deuxième caractéristique supplémentaire est le bloc



qui code la direction que suivra le sprite. Ceci est important car le scénario inclura le demi-tour de la fille à la fin de sa promenade.

Ainsi, le script initial serait :



Nous plaçons maintenant sous les blocs ci-dessus un repère supplémentaire qui repositionnera le sprite à l'extrême gauche (fin de l'écran) le long de l'axe X lorsqu'il atteindra l'extrême droite de l'écran. Car nous voulons que la fille soit vue en train de se promener dans différents arrière-plans (d'une scène à l'autre) comme s'il s'agissait d'une longue promenade en continu.

Pour ce faire, nous utilisons un bloc de code qui permet que quelque chose se passe lorsque la fin de l'écran est atteinte. C'est le bloc



Allez dans le dossier **Opérateurs** et sélectionnez le bloc **supérieur** à



qui est placé dans l'espace du bloc dans le script ci-dessus.

Tapez les chiffres 226 dans la case de droite qui représente le nombre de pixels (axe des X) à l'extrême droite de l'écran.

Donc, mettez

depuis



Mouvement dans la boîte de gauche

Nous utilisons le



Comme le sprite qui bouge  seulement sur l'axe horizontal (X).

Depuis le même dossier prenez

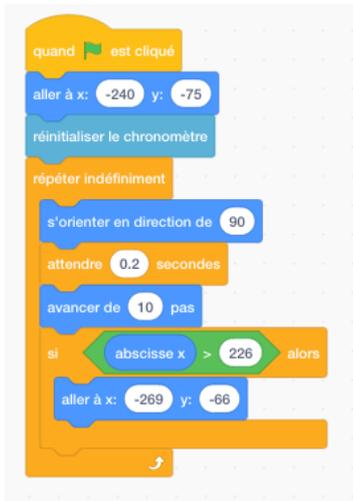


Et insérez dans le bloc



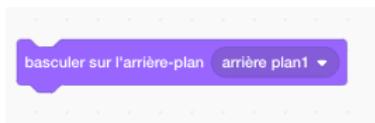
Notez que la coordonnée X a un nombre négatif élevé qui, lorsque le programme est lancé, donne l'impression que la fille marche sur la scène (écran).

Le code va maintenant apparaître comme suit :



Nous passons maintenant au script Scène où nous allons coder dans trois arrière-plans qui changeront en fonction d'un contrôle de paramètre de temps.

Les commandes principales pour ces blocs sont :



(Apparences)



(Capteurs)

et en associant



(Opérateurs)

Pour donner :

```
quand est cliqué
  basculer sur l'arrière-plan Desert
  répéter indéfiniment
    si chronomètre > 11 et chronomètre < 22 alors
      basculer sur l'arrière-plan Winter
    si chronomètre > 23 et chronomètre < 34 alors
      basculer sur l'arrière-plan Savanna
```

Veillez noter que les chiffres saisis sont basés sur le temps nécessaire à la fille pour se déplacer le long de l'axe des X d'un bout à l'autre de l'écran. De toute évidence, il est essentiel que le codeur synchronise le chronomètre avec le repositionnement du sprite à gauche de l'écran chaque fois qu'elle atteint la fin de l'écran. Cela impliquera probablement un peu « d'essais et d'erreurs » de la part de l'étudiant.

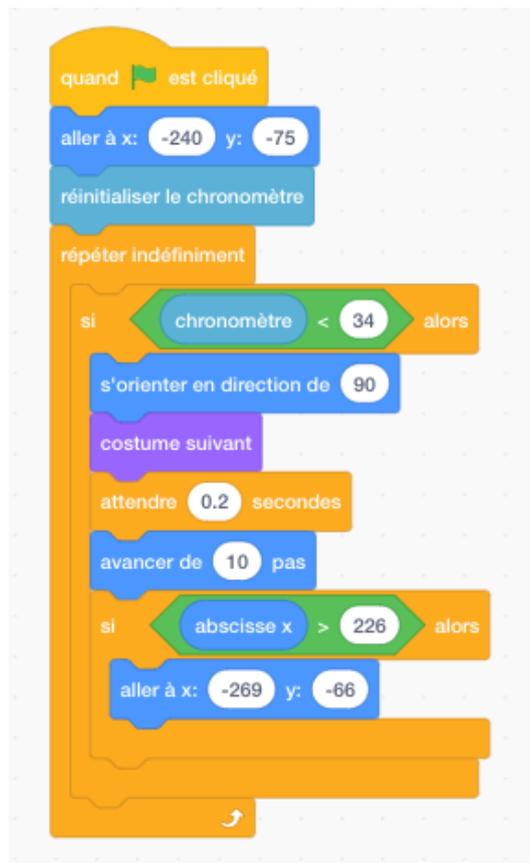
Donc, afin de compléter cette partie du programme, nous allons ajouter un morceau de code supplémentaire qui limitera le temps que la fille avance à droite pour lui permettre de faire demi-tour quand elle arrive à la fin du dernier arrière-plan et reculer de quelques pas (à gauche).



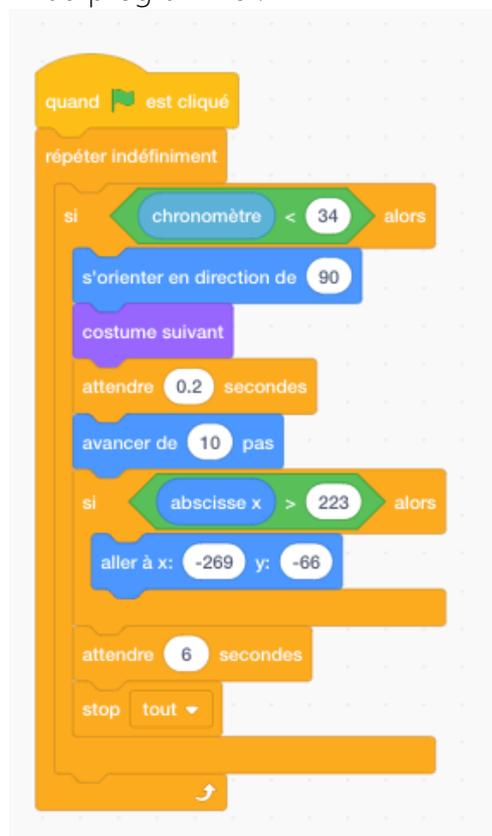
D'abord, nous mesurons le temps qu'il faut au sprite pour marcher du début de la première scène à la fin de la dernière scène. Dans ce projet spécifique, 34 secondes ont été insérées dans le Chronomètre dans le bloc **Si ____alors**

```
si chronomètre < 34 alors
```

et nous l'avons placé dans le code comme suit



Il faut maintenant un bloc supplémentaire qui inversera la direction du sprite qui marche et mettra fin au programme :



Maintenant, nous introduisons un deuxième sprite, à savoir un chien qui aboie.

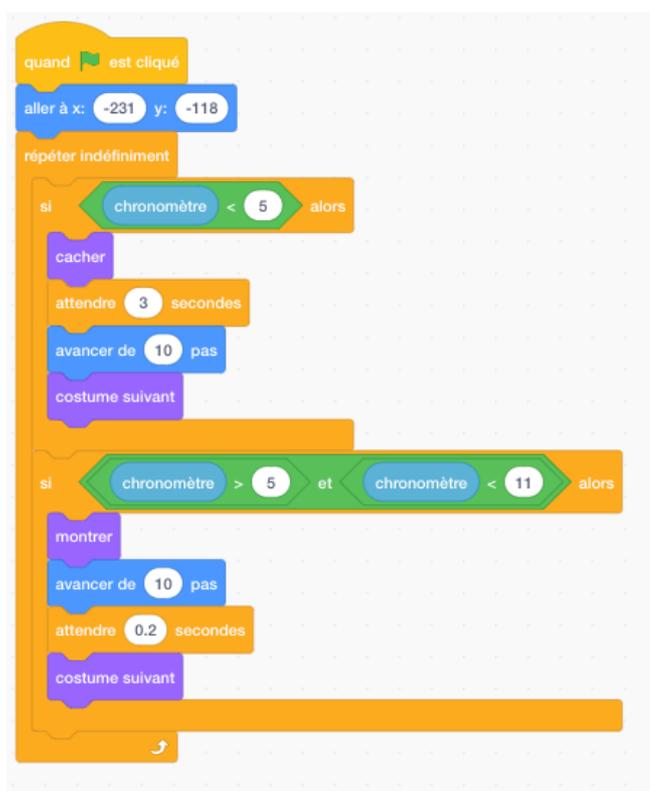


Mettez un code similaire à celui utilisé dans le script de la fille.

Par contre, nous ferons en sorte que le chien apparaisse après que la fille ait appelé son animal de compagnie pour la rejoindre pendant la promenade.

Ainsi, comme l'animal ne se montre que quelques secondes après que la fille ait commencé sa promenade et en réponse à son invitation à la rejoindre, tout en marchant toujours quelques pas derrière sa propriétaire, nous devons coder dans le script une/des commande(s) qui cachera le chien avant qu'il n'atteigne la fin de l'écran (veuillez noter que les arrière-plans changent au moment même où la fille atteint la fin de l'écran). Le chien ne réapparaît également à l'écran suivant seulement lorsque la fille a avancé de quelques pas.

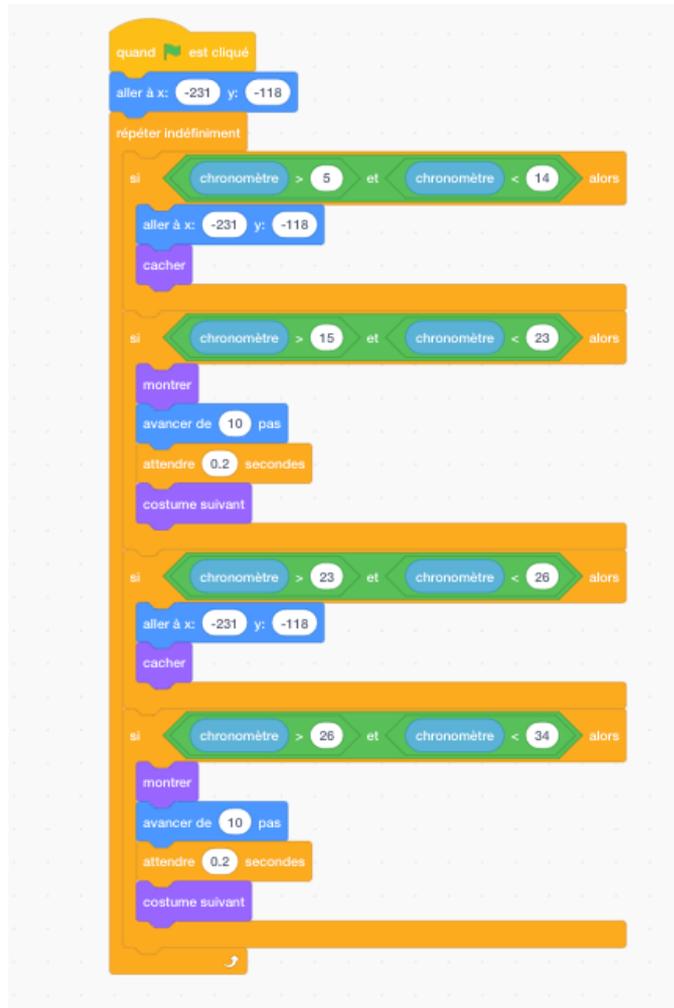
Ainsi, nous utilisons l'option **Montrer** et **Cacher** du dossier **Apparences**.



```
quand le drapeau est cliqué
  aller à x: -231 y: -118
  répéter indéfiniment
    si chronomètre < 5 alors
      cacher
      attendre 3 secondes
      avancer de 10 pas
      costume suivant
    si chronomètre > 5 et chronomètre < 11 alors
      montrer
      avancer de 10 pas
      attendre 0.2 secondes
      costume suivant
```

Comme pour le script du sprite de la fille, une fois la fin du dernier écran atteinte, saisissez un code qui inverse la direction du sprite pour qu'il fasse quelques pas en arrière.

Le codage des paramètres du chronomètre pour le chien doit être synchronisé avec celui du script fille.



Enfin, nous devrions également insérer un code pour permettre un peu de

conversation entre la fille et le chien.

Leçon 20 – Planification et conception d'un jeuⁱ

*Les élèves doivent être informés de l'importance de **planifier** d'abord le jeu proposé avec un crayon, un stylo et du papier plutôt que de se précipiter à écrire des scripts. Ils doivent comprendre la nécessité de définir le but du jeu, comment y parvenir et le rôle de chaque élément du jeu tel que les différents sprites et comment ils interagissent (effet).*

Le plan écrit (ou Algorithme) est ensuite utilisé pour construire le jeu.

Cependant, le tuteur devrait d'abord fournir un exemple des différents types de jeux pouvant être utilisés comme points de départ ou modèles pour les créations des élèves.

Exemple de **directives** de jeu « Attaque de requin ! »

- *Combien de personnages (deux sprites pour commencer)*
- *Rôle des personnages (sprite Petit Poisson, sprite Méchant Requin)*
- *Opérations des personnages (le mouvement d'un sprite est contrôlé par les quatre « touches flèches » ; le second personnage « suivra » le premier sprite en utilisant le bloc de code « s'orienter vers »)*
- *Quel arrière-plan conviendrait le mieux au type de jeu que je suis en train de créer ?*

Télécharger un jeu sur le site Scratch

Des exemples de projets de tous les élèves devraient être téléchargés sur le compte en ligne Scratch de l'école, de la classe ou du groupe que vous avez aidé à configurer sur le site de Scratch, comprenant les meilleurs projets stockés également sur votre propre compte de projet Scratch.

Il est donc important qu'une note explicative, brève et bien écrite, expliquant le thème et le fonctionnement du projet, soit fournie pour chaque projet téléchargé afin que les utilisateurs en ligne en bénéficient.

Une fois votre projet téléchargé, allez sur la page du projet et complétez les sections à droite de l'écran intitulées *Instructions*, *Notes et crédits* et *Ajouter des balises de projet*.



Frog in the Forest

See inside



Instructions

Tell people how to use your project (such as which keys to press).

Notes and Credits

How did you make this project? Did you use ideas, scripts or artwork from other people? Thank them here.



Leçon 21 - Jeu - Attaque de requin !

Cette leçon présente aux élèves a) la programmation des **Touches flèches** pour contrôler le mouvement des sprites (caractéristique courante de nombreux jeux informatiques) et le b) Code **Stop tout**.

Cette leçon présente également une puissante commande **Touche le** et la commande **si** (avec son impact de cause à effet).

Jouabilité - Résumé du plan de codage

Le requin poursuit un petit poisson, qui tente désespérément de lui échapper. Lorsque le requin le touche, le petit poisson disparaît comme s'il avait été mangé par le requin qui lui change de costume pour simuler un processus de déglutition. Le requin dit joyeusement « Miam ! » et le jeu se termine.

Sélectionnez depuis le dossier Animal a) le sprite Requin avec deux changements de costumes (bouche « ouverte » et « fermée ») e b) le sprite d'un petit poisson. Donnez des noms au requin et au poisson. Dans cet exemple j'ai nommé le requin Kraken et le poisson Pandora.



Entrez les morceaux de code suivants dans le script de Kraken (Requin) :



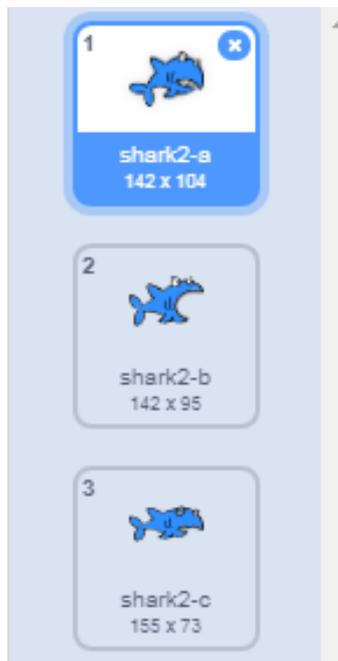
Le bloc de code de la boucle *répéter indéfiniment* et le contenu qu'il contient signifieront que, où que soit Pandora le requin le *suivra toujours*.

Le « *avancer de 2 pas* » devrait être une vitesse suffisante pour la première version de ce jeu Attaque de requin car il permettra au petit poisson d'échapper à Kraken.



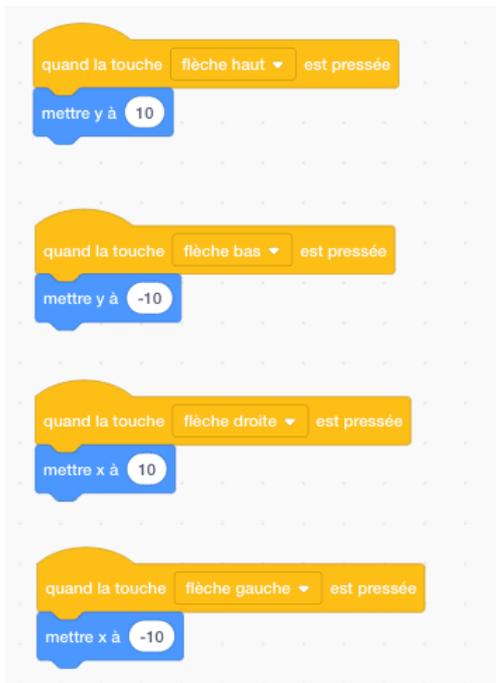
Dans le morceau de code ci-dessus, l'option « touche le » se trouve dans le dossier **Capteurs**.

Les morceaux de code additionnels du dossier **Apparences** donneront l'impression que le Requin mange le petit poisson en alternant de costume avec les options de bouche a) fermée b) ouverte (costumes).

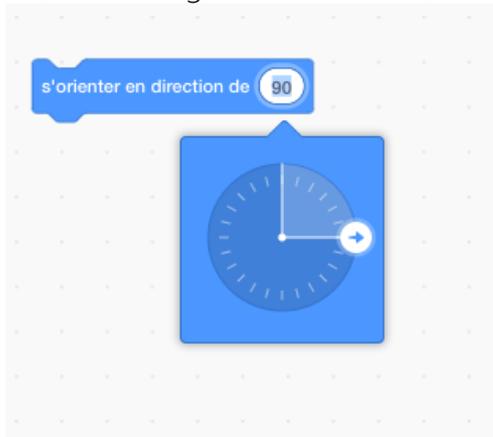


Le script de Pandora (petit poisson)

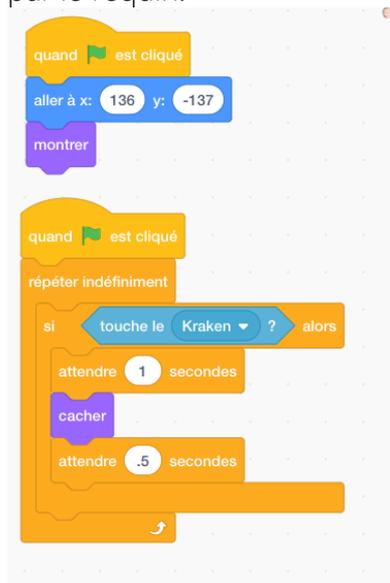
Tapez les morceaux de code suivant dans le script de Pandora (poisson) :



Pour s'assurer que le poisson va toujours dans la bonne direction lorsqu'il est déplacé à l'aide des touches flèches gauche et droite, le bloc **s'orienter en direction de**, de la catégorie Mouvement, doit être ajouté au script comme suit :

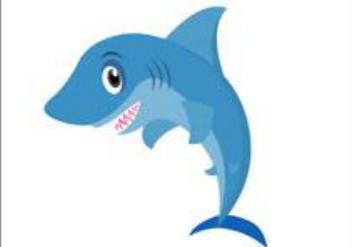


Enfin, l'insertion des blocs *Montrer* et *Cacher*, ainsi que deux options *Attendre* (illustrées ci-dessous), signifiera que Pandora disparaîtra après avoir été « mangé » par le requin.



Comme vous pouvez le constater avec le code ci-dessus, l'utilisation de *Stop Tout* terminera le jeu « Attaque de Requin ! ».

Requins menacés



Il existe des preuves fossiles de requins remontant à 400 millions d'années. Ceci est remarquable, surtout lorsqu'on se rend compte que les premiers dinosaures ne sont apparus sur la planète qu'il y a 243 millions d'années.

Nous connaissons aujourd'hui plus de 1000 espèces de requins et de nouvelles espèces sont découvertes chaque année.

Mais l'existence même de ces remarquables poissons est menacée par les activités de l'homme. Avec 70 millions de requins tués chaque année, de nombreuses espèces sont maintenant menacées et pourraient disparaître dans un proche avenir.

Sauver ces animaux est essentiel pour restaurer la vie des océans sur la Terre.

Exercice

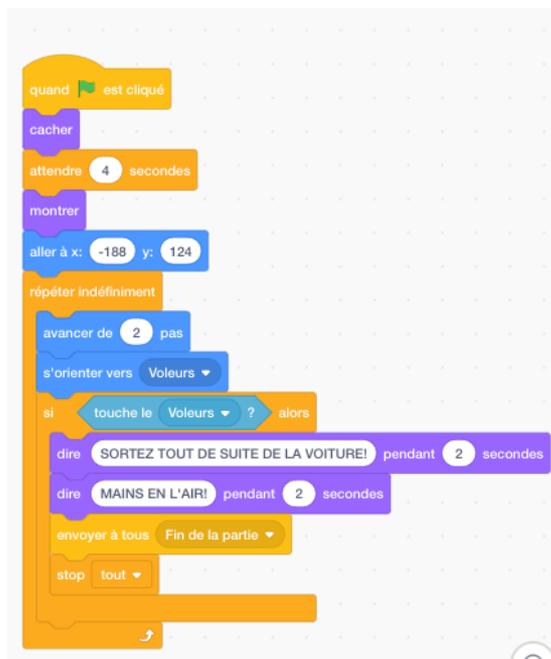
Poursuite de voitures

Demandez aux élèves d'utiliser les commandes et les processus de codage enseignés dans « Attaque de Requin ! ». Leçon pour créer un projet de poursuite de voitures.

Code pour un jeu de poursuite de voitures

Demandez d'abord aux élèves d'utiliser l'Éditeur de dessin pour dessiner une voiture de police et une voiture de voleur, ainsi qu'une série d'arrière-plans de route adaptés.

Demandez aux élèves d'utiliser un code similaire à celui ci-dessous pour la voiture de police :

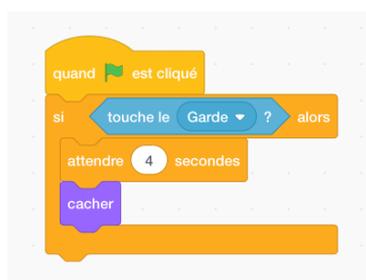


```
quand est cliqué
  cacher
  attendre 4 secondes
  montrer
  aller à x: -188 y: 124
  répéter indéfiniment
    avancer de 2 pas
    s'orienter vers Voleurs
    si touche le Voleurs ? alors
      dire SORTEZ TOUT DE SUITE DE LA VOITURE! pendant 2 secondes
      dire MAIS EN L'AIR! pendant 2 secondes
      envoyer à tous Fin de la partie
      stop tout
```

Demandez aux élèves d'utiliser un code similaire à celui ci-dessous pour la voiture du voleur :



```
quand est cliqué
  aller à x: -188 y: 124
  cacher
  attendre 4 secondes
  montrer
```



```
quand est cliqué
  si touche le Garde ? alors
    attendre 4 secondes
    cacher
```

Remarque : programmer les quatre touches flèches pour contrôler la direction et le mouvement de la voiture

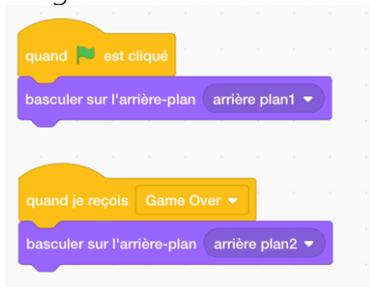
Dans « Attaque de requin ! » des blocs de codes spécifiques étaient utilisés pour que le petit poisson chassé *soit tourné sur la gauche* lorsqu'il se déplaçait *vers la gauche* et *sur la droite* lorsqu'il se déplaçait *vers la droite*.

Cette fois, nous devons également avoir la voiture poursuivie aller vers le haut (nord) et vers le bas (sud) lorsque nous allons dans ces directions.

Voir ci-dessous :



Programmer dans différents Arrière-plans



Leçon 22 - Attaque de requin avancée

Cette leçon augmente la complexité du jeu "Attaque de requin !" en introduisant dans le projet un élément concurrentiel sous la forme d'un **chronomètre** (commande *Capteurs*), de la **sélection aléatoire** (notion de hasard), de l'utilisation de **variables** simples sous la forme de "vies" et de la création de différents **niveaux**.

Remarque : en mathématiques, une **variable** est une valeur qui peut changer dans le cadre d'un problème ou d'un ensemble d'opérations donné. Dans cet exemple, nous utilisons des variables pour représenter le nombre de vies d'un des sprites, qui change à la suite d'une interaction avec un autre sprite.

Jouabilité - résumé du plan de codage

Le requin poursuit un petit poisson. Ce dernier a trois vies et doit rester en vie durant 30 secondes puis le jeu se termine automatiquement. Chaque fois que le requin touche le poisson, ce dernier perd une vie.

Après dix secondes, le jeu passe à un niveau différent, ce qui implique quelques modifications à l'arrière-plan, une vitesse augmentée pour le requin et l'apparition d'un crabe qui, s'il touche le poisson, entraînera une réduction de vie de ce dernier. Si le poisson est encore en vie après 30 secondes, il a gagné.

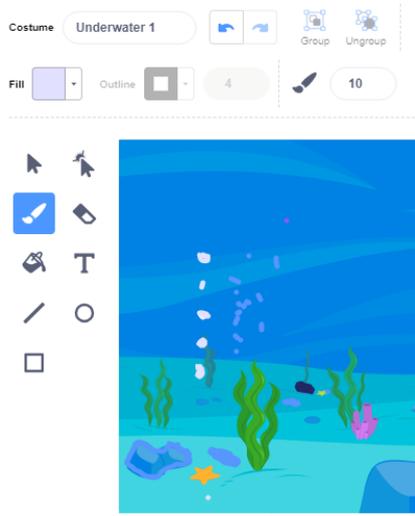
Comme pour la leçon précédente (Attaque de requin !), nous utiliserons un Requin (Kraken), un petit poisson (Pandora) et un arrière-plan (Underwater) du dossier *Dehors de Scratch*.

Donc, commencez par télécharger le projet précédent *Attaque de requin !*

Par contre cette fois nous allons considérablement modifier les scripts du requin et du petit poisson ; apportez différentes versions de l'arrière-plan *Underwater* qui représenteront des niveaux de difficulté croissante et inscrivez les messages *Vous avez gagné !* et *Vous avez perdu !* pour indiquer la fin du jeu.

Démarrez

Cliquez sur l'icône *Arrière-plan*, puis dupliquez une fois l'arrière-plan *Underwater*. Modifiez ce nouveau niveau (niveau 2) en y intégrant des fonctionnalités supplémentaires telles que des brins / tiges supplémentaires ou plus grandes sur les algues, les coraux et les rochers, ainsi qu'un changement de couleur de l'arrière-plan en un bleu plus foncé.



Lorsque la modification est terminée, dupliquez le niveau 2 deux fois de plus. Avec le deuxième nouveau niveau (niveau 3), utilisez l'option d'édition pour entrer dans l'Éditeur de dessin. Puis tapez les phrases appropriées en gros caractères gras tels que ***Vous avez perdu !*** En utilisant l'icône T(exte). Avec le troisième nouveau niveau (niveau 4), tapez le texte ***Vous avez gagné !*** en gros caractères gras. Pour repositionner le texte à un emplacement approprié, cliquez sur l'icône T(exte). Placez le curseur sur le petit rectangle noir qui apparaît dans le coin supérieur gauche du texte à l'écran (voir ci-dessous). Une icône "main" apparaît alors, permettant à l'opérateur de faire glisser le texte vers un nouvel emplacement.



Codage des *niveaux* supplémentaires dans le script

Pour coder les trois niveaux supplémentaires dans le programme, nous passons d'abord au script du sprite de petit poisson. Puis placez un *chronomètre* qui permettra de changer les arrière-plans en fonction d'un certain laps de temps.

Allez au dossier *Contrôle* et choisissez :



Puis allez au dossier « *Opérateurs* » et au bloc de code « *plus grand que* »



Dans la première case blanche vide ci-dessus, placez le **Chronomètre** à partir du dossier *Capteurs*.

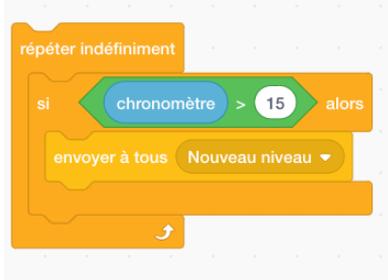
Dans la deuxième case blanche vide, insérez un numéro qui représentera une durée de temps mesurée en secondes :



Dans les blocs *Répéter indéfiniment si*, ci-dessus, placez l'option *envoyer à tous* depuis le dossier *Événements*.

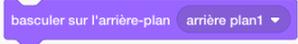


Pour s'assurer que *l'arrière-plan* change après 15 secondes, sélectionnez *Nouveau message* dans le bloc « Envoyer à tous » et écrivez « Nouveau niveau ».



Envoyer à tous est un code de commande très important dans Scratch car il envoie un message à une autre partie du programme lui demandant d'appliquer une modification.

Mais pour qu'il fonctionne il doit avoir une commande correspondante **Je reçois** qui, dans ce cas-ci, sera positionnée dans le script de **l'arrière-plan**.

Remarque : Il y a une alternative et façon plus simple de changer les arrière-plans, à savoir l'utilisation directe de  dans le sprite plutôt que dans le script de l'arrière-plan. Toutefois, les élèves sont invités à se familiariser avec l'importance des commandes « Envoyer à tous ».

Donc, allez au script de l'arrière-plan. Entrez la commande **Quand je reçois** _____ suivie du par le morceau de code **basculer sur l'arrière-plan** ___ présent dans le dossier *Apparences*, comme indiqué ci-dessous :



Pour s'assurer que le jeu se termine et que tous les éléments s'arrêtent quand le message *Vous avez gagné !* ou *Vous avez perdu !* apparaît, placez le code suivant dans le script Pandora (ou Kraken) :



L'utilisateur doit également s'assurer que le bon écran apparaît chaque fois que le jeu est joué ou redémarré en saisissant le code suivant dans le script de l'*Arrière-plan* :



Une caractéristique du nouveau niveau sera le fait que le requin augmentera en vitesse, ce qui rendra le jeu plus difficile pour Pandora (utilisateur). Cette fonctionnalité est obtenue en entrant le code suivant dans le script du Requin :



Vous pouvez bien sûr augmenter encore plus la vitesse en changeant les chiffres dans le bloc *avancer de __ pas*.

Le sprite de petit poisson doit rester en vie pour un temps défini que l'utilisateur a choisi. S'il parvient à rester en vie durant ce temps imparti, alors il est déclaré vainqueur.

Ceci est réalisé en entrant le code suivant dans le petit script de poisson basé sur une durée de jeu de 30 secondes :



Le *Chronomètre* doit être remis à 0 à chaque nouvelle partie.

Ceci est réalisé en plaçant l'option Réinitialiser le chronomètre dans le code du petit poisson comme suit :



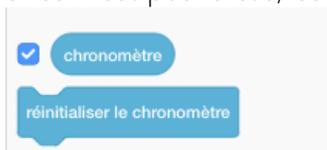
Cliquez sur le Drapeau vert pour vérifier que le jeu fonctionne correctement.

Notez qu'une boîte d'indicateur du chronomètre apparaît en haut à gauche de l'écran :



Cependant, pour que la minuterie soit affichée sur la scène, la petite *case* à gauche du code chronomètre dans le dossier Capteurs doit être cochée.

Si ce n'est pas le cas, cochez-la.



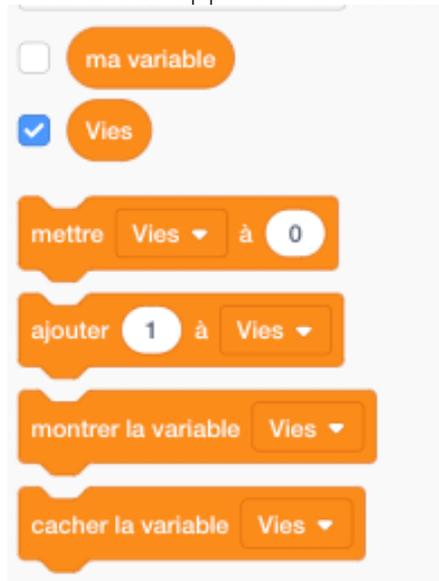
L'utilisation de variables

Pour programmer des Vies supplémentaires pour le sprite du petit poisson, allez dans le dossier Variables et sélectionnez l'option *Créer une variable*. Vous devrez entrer un nom.

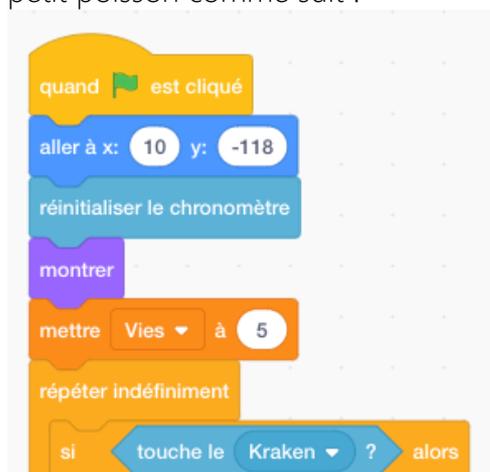


Entrez le mot *vies*

L'icône **Vies** apparaît maintenant dans le dossier *Variables*



Entrez le chiffre 5 dans le bloc **Mettre vies à** __ avant de le placer dans le script du petit poisson comme suit :



Le petit poisson possède maintenant 5 vies.

S'il perd toutes ses vies *avant* d'atteindre 30 secondes, la partie est terminée.

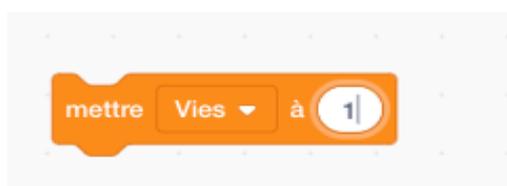
Alors entrez le code suivant :



Qui est constitué des blocs dans les dossiers *Contrôle*, *Variables* et *Opérateurs*.

Le petit poisson perd une vie chaque fois qu'il est touché par le requin.

Ce résultat est obtenu en insérant dans la section des blocs *touche* du script du requin.

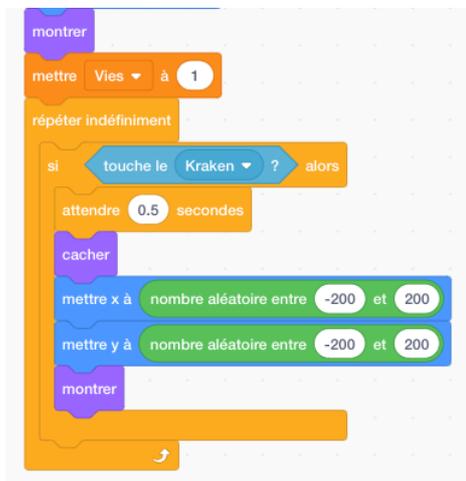


En outre, enlevez le bloc  du script et supprimez-le.

En effet, contrairement à l'ancienne version d'*Attaque de requin*, ce jeu ne se termine pas la première fois que le petit poisson est touché par le requin car ce dernier dispose désormais de 5 vies.



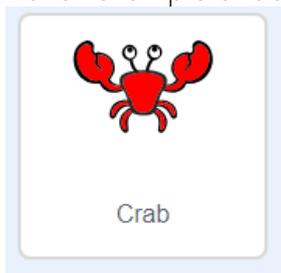
Pour donner une chance au petit poisson de tenter d'échapper aux dents du requin, nous devons ajouter un morceau de code qui lui permettra d'apparaître aléatoirement à différents endroits chaque fois qu'il est attrapé (touché) par le requin.

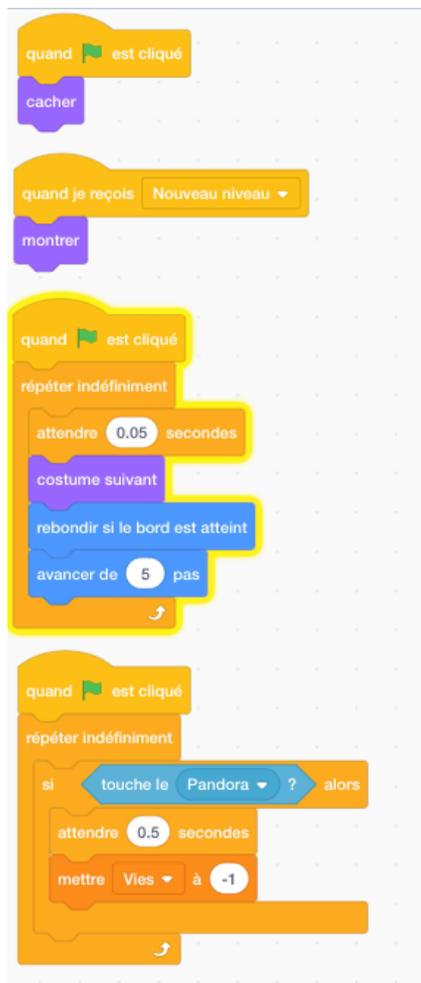


Plus la plage de nombres est large dans le bloc nombre aléatoire (vert), plus grande sera la zone de réapparition du petit poisson (c.-à-d. *montrer*) après chaque attaque réussie du requin.

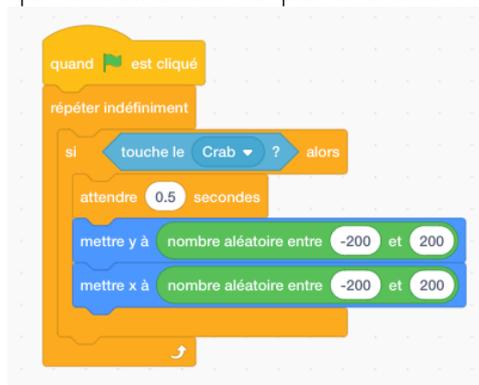
Pour ajouter encore plus de difficulté au jeu, créez un autre sprite qui apparait lorsque le deuxième niveau est atteint. Ce sprite sera codé de sorte que, s'il touche le petit poisson, il ôtera également une vie.

Dans l'exemple ci-dessous, nous utilisons un sprite de *crabe*.





Revenons au script du petit poisson (Pandora), ou nous allons ajouter les instructions qui entraîneront la perte d'une vie à chaque fois qu'il est touché par le crabe.



Exercice

Poursuivre les dinosaures

Demandez aux participants de créer le jeu d'un dinosaure poursuivant un autre dinosaure en utilisant *Vies* et *Niveaux*. Cela impliquera un sprite allant d'une forêt vers un désert puis vers une scène d'eau, de champ ou de montagne. À partir de la deuxième scène, demandez-leur d'insérer un deuxième dinosaure (ou une autre créature exotique) qui commencera à poursuivre le personnage principal.

Leçon 23 – Tennis solitaire

Cette leçon présente un jeu simple inspiré d'un des grands jeux classiques connus sous le nom de **Pong**. Il présentera aux élèves une autre commande de **Capteurs** basée sur le **toucher des couleurs**. Ils apprendront également comment **changer** la direction d'un Sprite à l'aide d'une combinaison de commandes de **Mouvement** et d'**Opérateurs**.

Après avoir ouvert Scratch, supprimez le sprite du chat.

Le script Raquette



Allez sur l'icône nouveau sprite et dessinez une raquette noire en utilisant l'outil de dessin *ligne* ainsi qu'une *taille de pinceau* adéquate.



Renommez le sprite « *raquette* ».

Allez sur le script pour la raquette.
Insérez les morceaux de code suivants :



Le bloc **Mettre x à 0** est dans le dossier *Mouvement*.

Sélectionnez  depuis le dossier *Capteurs* et placez le dans le 0 de .

Cela garantit que le curseur de la souris ne se déplacera toujours que vers la gauche ou vers la droite sur l'écran le long de l'axe X.

Les Buts – Préparer le terrain

Pour créer une ligne de but, allez à l'icône *Scène* et choisissez l'option *Arrière-plan*. Puis sélectionnez l'outil de dessin *Ligne*, une grande *taille de Pinceau* et la couleur **ROUGE** depuis la *palette*.

Amenez le curseur au bas de l'écran et tracez une épaisse ligne rouge de droite à gauche sur toute sa longueur.

Placez la raquette légèrement au-dessus de la ligne de but ROUGE.

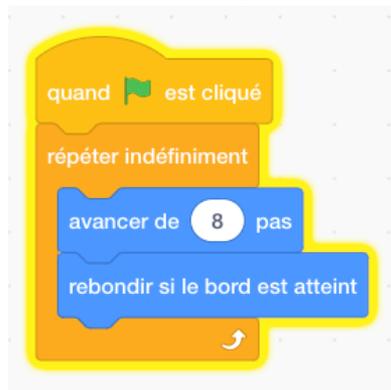
Le script de la balle



Allez à la colonne des sprites et sélectionnez un sprite de balle dans la galerie. Sinon, vous pouvez dessiner votre propre balle. Si vous décidez de le faire, ne choisissez pas le rouge comme couleur car nous utilisons le rouge comme couleur de la ligne de but.

Il est cependant recommandé aux élèves d'utiliser un sprite existant dans la galerie pour ce projet.

Entrons maintenant un script de mouvement pour la balle



suivi d'un bloc de commande séparé qui permettra à la balle de rebondir dans une direction généralement ascendante avec un bel effet sonore si elle touche la raquette.

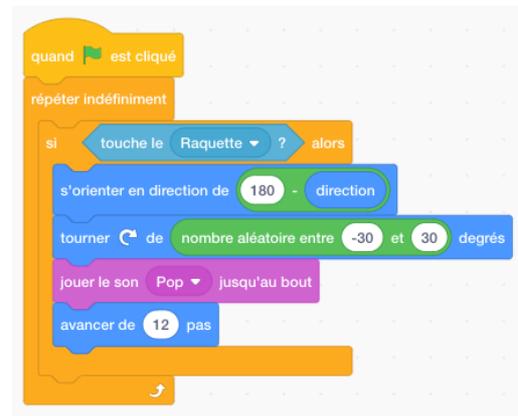
Pour ce faire, entrez d'abord le script ci-dessous :



Toutefois, pour que la balle rebondisse vers le haut lorsqu'elle touche la raquette plutôt que vers le bas, nous devons utiliser un bloc *s'orienter dans la direction de*, où le mouvement vers le bas (c.-à-d. 180)



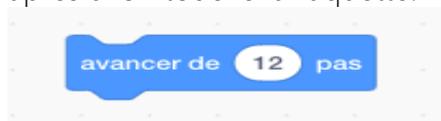
est annulé en le soustrayant de sa trajectoire actuelle (direction), en utilisant le code approprié dans le dossier Operateurs.



La variation de direction dans son mouvement ascendant est obtenue par l'utilisation d'une *sélection de nombre aléatoire* qui, grâce à la plage de paramètres entrée, garantit un changement de direction à chaque fois que la balle touche la raquette



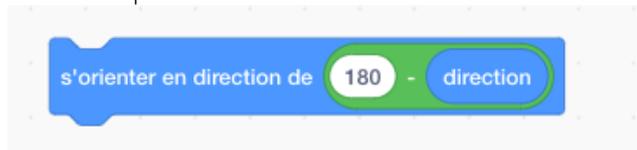
Sa poussée vers le haut est facilitée par l'augmentation du nombre de pas qu'il fait après avoir touché la raquette.



La séquence de code finale pour cette commande devrait être la suivante :

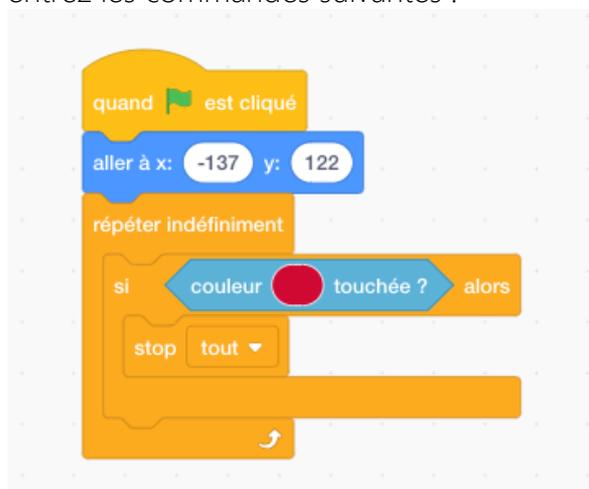
La disposition des blocs de codage, illustrée dans l'exemple ci-dessus, est très importante et essentielle au bon déroulement du jeu.

Par exemple si



apparaît *après* le bloc avancer de **12 pas**, le sprite de balle continuera à aller vers le bas ou vers le côté plutôt que vers le haut.

Pour garantir que le jeu se termine correctement lorsque la ligne rouge est touchée, entrez les commandes suivantes :



REMARQUE : pour que le bloc de code ci-dessus fonctionne correctement, la couleur rouge choisie pour la **couleur touchée** devrait avoir la même valeur de couleur (*couleur, saturation et luminosité*) que celle de la valeur de couleur rouge pour la ligne rouge.

Leçon 24 – Jeux d’aventure : l’incroyable labyrinthe !

Les jeux **d’aventure** sont basés sur un personnage principal entreprenant un voyage à travers de nombreux paysages différents où il / elle rencontre des obstacles ou des dangers qui doivent être surmontés ou contournés pour atteindre la destination finale et réclamer un trésor ou une récompense.

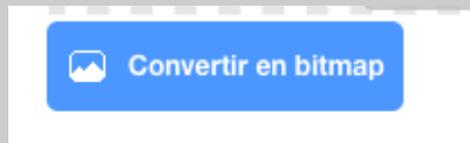
Le labyrinthe est une variante populaire de ce type de jeu.

Cette leçon renforce plusieurs des commandes importantes apprises dans les chapitres précédents, telles que **l’envoi de messages** et le **toucher de couleur**.

Bitmaps et vecteurs – un bref aperçu

Les Bitmaps et les Vecteurs sont différentes sortes de graphiques en deux dimensions qui sont utilisés dans l’Éditeur de dessin de Scratch.

Ce qui les différencie c’est la façon dont ils stockent leurs informations.



Les images Bitmap stockent les informations de couleur. Elles stockent les couleurs de chaque pixel qui les constituent.

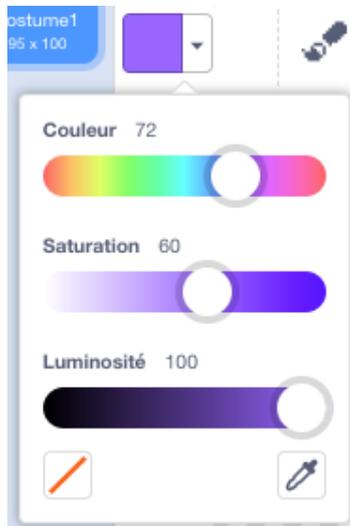
Les images bitmap stockent les informations de couleur de chaque pixel composant l'image ci-dessus.



Les images vectorielles stockent les formules mathématiques qui dessinent les lignes et les courbes.

Sélection des couleurs dans l’Éditeur de dessin

Lors de la sélection des couleurs dans l’éditeur de dessin, il est essentiel que les utilisateurs notent les numéros qui apparaissent sous *Couleur*, *Saturation* et *Luminosité* (voir illustration ci-dessous). Sinon, des problèmes risquent de se produire lors de la sélection des couleurs dans les blocs de Capteurs pour la création d’un script.



Créer des niveaux

D'abord sélectionnez l'icône Scène, cliquez sur Éditer et colorez l'arrière-plan en noir.

Copiez cet arrière-plan noir trois fois.

Dans l'un des arrière-plans, utilisez l'outil TEXTE pour écrire **Vous avez gagné !**

Dans un deuxième arrière-plan, utilisez l'outil TEXTE pour écrire **Vous avez perdu !**

Allez dans l'un des derniers arrière-plans (**Niveau basique**)

Utilisez les outils Rectangle, couleur blanche et l'icône de contour pour dessiner un certain nombre de boîtes à différentes places de l'écran. Dans le coin en haut mettez un rectangle jaune.

Allez dans le dernier arrière-plan et renommez-le **Niveau avancé**

Coloré l'espace principal en **Vert**.

Utilisez les outils Rectangle, couleur blanche et l'icône plein pour dessiner deux nuages et deux formes de type château.

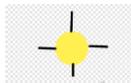
Les sprites

Commencez par sélectionner un sprite de personnage principal dont le mouvement sera contrôlé par les quatre touches flèches.



Allez sur l'icône de nouveau sprite . Dessinez une balle colorée en utilisant l'outil *cercle*.

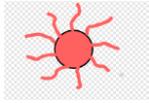
Utilisez l'icône pinceau pour dessiner un certain nombre de lignes irrégulières venant de la zone du cercle.



Nommez ce sprite *Héros*.



À nouveau, utilisez  pour dessiner une autre balle colorée (ROUGE) en utilisant l'outil de dessin *cercle*. Puis avec le pinceau dessinez quelques lignes inégales BLEUES venant du cercle.



Vous pouvez également sélectionner les sprites appropriées dans la galerie. Ce sera plus facile et moins problématique pour vous en tant qu'enseignant, car cela évitera aux élèves d'oublier les nombres spécifiques de leur sélection de couleur (Couleur, Saturation et Luminosité), pour les sprites qu'ils ont créés.

Allez au script pour ce sprite et entrez les commandes suivantes :

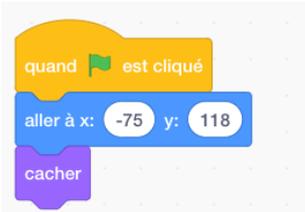


Ce sprite représente un obstacle pour le sprite Héros qui, s'il le touche, lui fera perdre la partie.

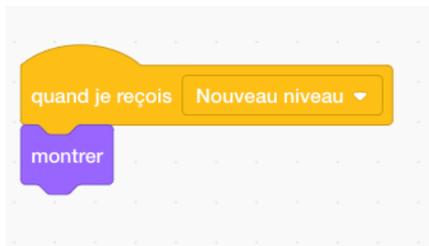
Voilà pourquoi, pour augmenter la difficulté de Héros, nous aurons le deuxième sprite en rotation constante, obtenu par l'utilisation d'un bloc *Tourner de degrés* dans le dossier *Mouvement* en association avec un bloc *Répéter indéfiniment*.



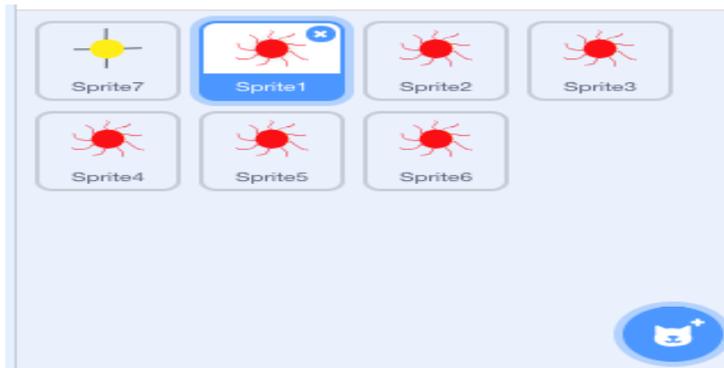
Par ailleurs, ce sprite apparaîtra *uniquement* dans le niveau avancé. D'où l'utilisation de CACHER dans le code d'ouverture



Et en entrant le bloc suivant :



Dupliquez ce sprite six fois.



Bien sûr, la position de chacun de ces sprites dupliqués doit être différente et doit être située de manière à donner un effet d'obstacle maximum au sprite Héros. Donc changez l'entrée dans chaque bloc de code individuel



Nous allons maintenant mettre un autre sprite obstacle qui sera programmé pour se déplacer à travers l'écran, pour augmenter sa vitesse lorsque le Niveau Avancé est atteint et lorsqu'il est touché par Héros, le jeu **se termine** et le message **Vous avez perdu** apparaît sur un écran noir. Ceci est réalisé en utilisant les trois morceaux de codes suivants pour constituer son script :

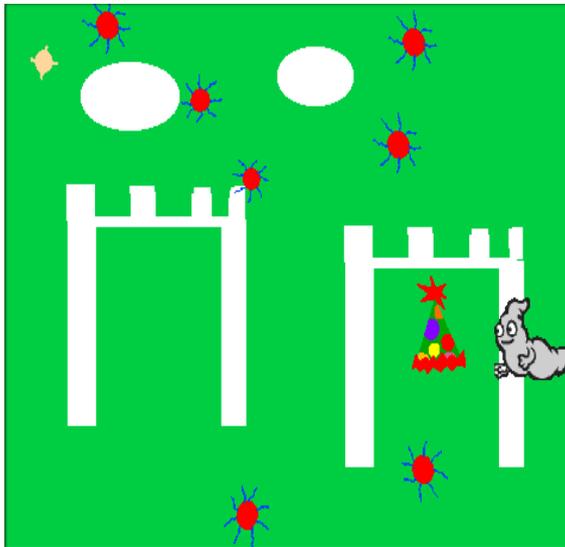


Pour augmenter le niveau de difficulté pour le sprite Héros, le sprite Fantôme peut être dupliqué plusieurs fois, ayant, pour des raisons évidentes, les coordonnées X et Y différentes pour chaque version.

L'ajout du sprite final servira de PRIX. Si le sprite Héros l'atteint le message **Vous avez gagné !** sera inscrit à l'écran et le jeu se terminera.



Le positionnement final des sprites dans le Niveau avancé doit être similaire à :



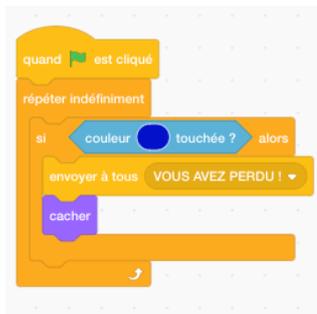
L'intensité de la difficulté pour Héros à obtenir le PRIX peut être augmentée, par exemple, en ajoutant deux blocs de code au script du sprite de cercle rouge positionné devant le PRIX, ce qui lui permettra de se déplacer dans un mouvement circulaire.



Le script pour Héros devrait inclure :
un ensemble d'instructions indiquant que s'il touche les murs *blancs* du labyrinthe, Héros disparaîtra.



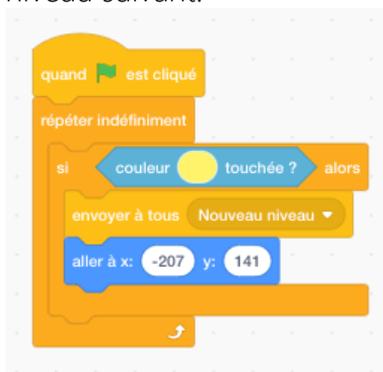
un ensemble d'instructions indiquant que s'il touche les pics *bleus* des *mines*, le message « Vous avez Perdu » apparaîtra et Héros disparaîtra.



un ensemble d'instructions indiquant que s'il touche le sprite *Fantôme*, le message « Vous avez Perdu » apparaîtra et Héros disparaîtra.



un ensemble d'instructions indiquant que s'il touche la boîte *jaune*, le jeu passe au niveau suivant.



un ensemble d'instructions indiquant que s'il touche le prix dans le deuxième niveau, le message « Vous avez gagné » apparaîtra et le jeu se terminera.



Enfin, le code de la scène doit être écrit pour répondre (via l'intégration d'un bloc **Quand je reçois**) aux « Envoyer à » contenus dans le sprite Héros :

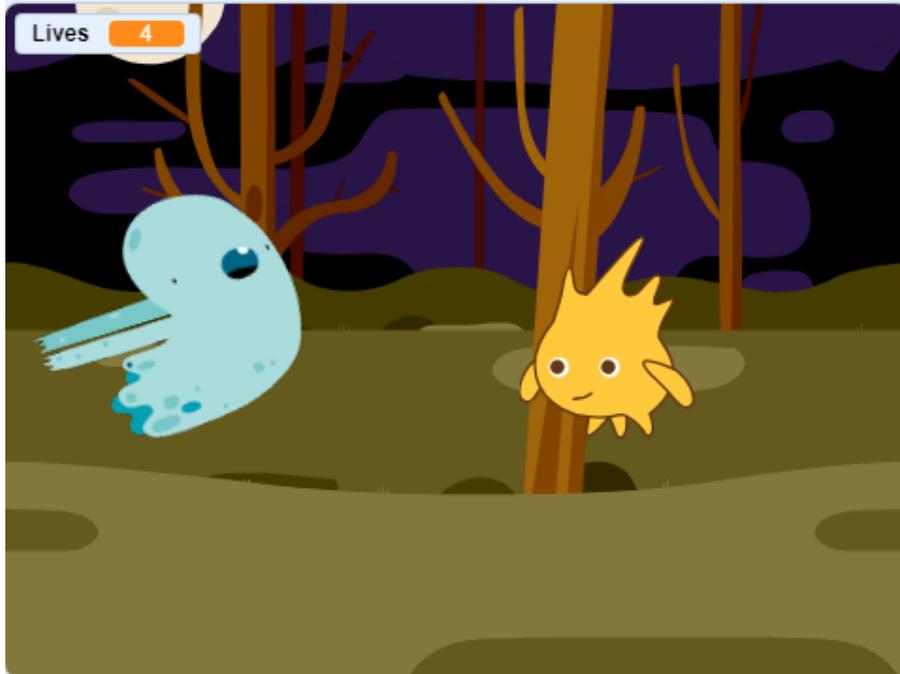


Exercice :

Dites « Cheese » ! Demandez aux élèves de construire un projet Labyrinthe dans lequel une souris doit éviter de se faire prendre par des pièges à souris et d'être attrapée par des chats afin d'atteindre la ligne d'arrivée et de réclamer une belle grosse tranche de fromage.

Leçon 25 – Chasseur de démons

Chasseur de démons



Jouabilité - Résumé du plan de codage

- Le « Bon Sprite » se déplace avec les touches flèches mais doit toujours avancer
- Le démon se déplace aléatoirement pour que le bon sprite ne sache pas où il va apparaître
- Le bon sprite commence avec 5 vies
- Si le démon touche le bon sprite, ce dernier perd une vie
- Si les vies du bon sprite sont égales à zéro, alors le jeu se termine et un arrière-plan « Game Over » apparaît.
- Lorsque le jeu redémarre, l'arrière-plan est réinitialisé et les vies sont également réinitialisées à 5.

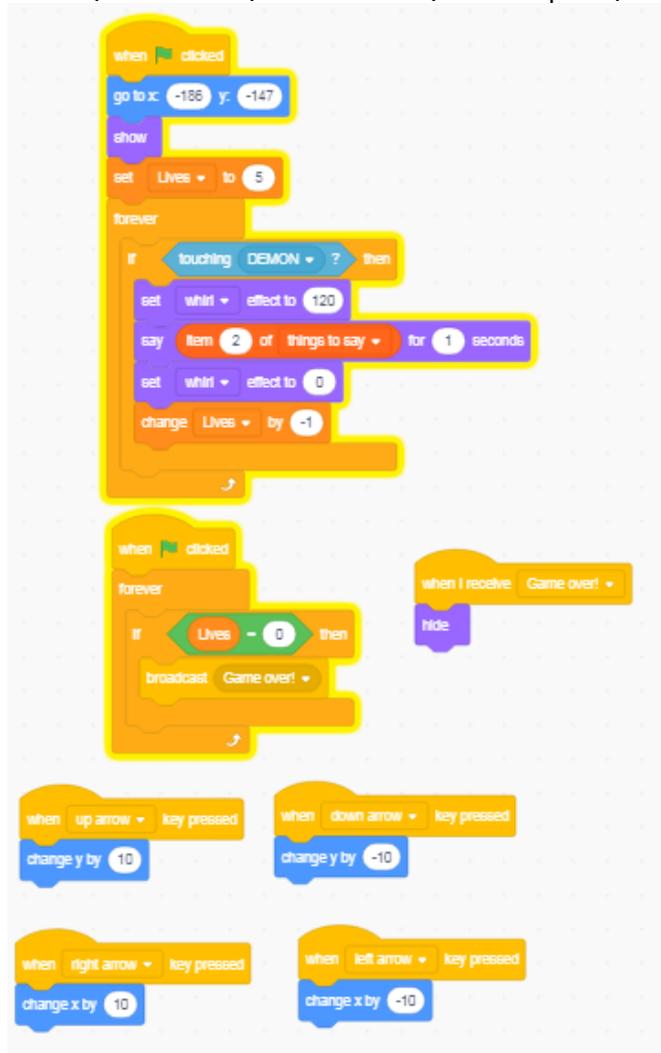
Blocs de commandes clés: « Répéter indéfiniment si _____ », « Quand je reçois », « Envoyer à » (dossier Contrôle), « Dire » (Apparences), « Touche le (un sprite) » (Capteurs), « Nombre aléatoire entre ___ et ___ », « Créer une Variable », « Créer une Liste » (Donnée), « ___or___ » (Opérateurs), « Tourner de ___ degrés », « Rebondir si le bord est atteint » (Mouvement), « Jouer le son _____ » (Son).

L'option *Créer une Liste* dans le dossier *Donnée* permet au développeur de compiler, par exemple, une liste de commentaires ou de mots qu'un sprite dirait quand quelque chose de spécifique se produit, comme être touché par un autre sprite pendant un jeu.

Le jeu nécessite deux sprites et deux versions d'arrière-plan. Ce dernier aurait un écran affichant le texte **Game Over**.

Améliorations : Après avoir suivi les instructions ci-dessus, demandez aux élèves d'effectuer certains ajustements qu'ils jugeraient bon pour améliorer le jeu.

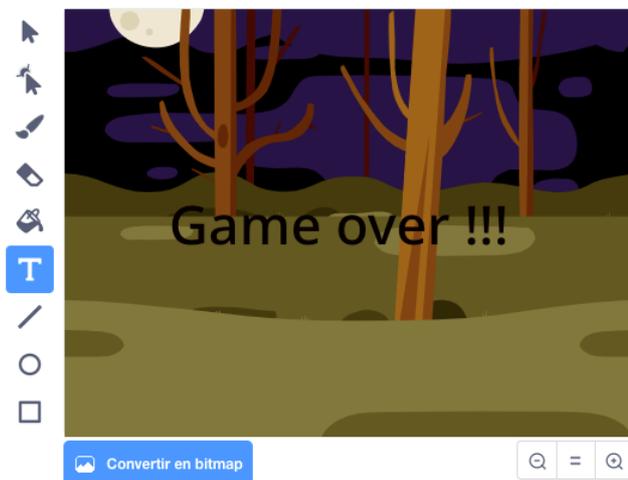
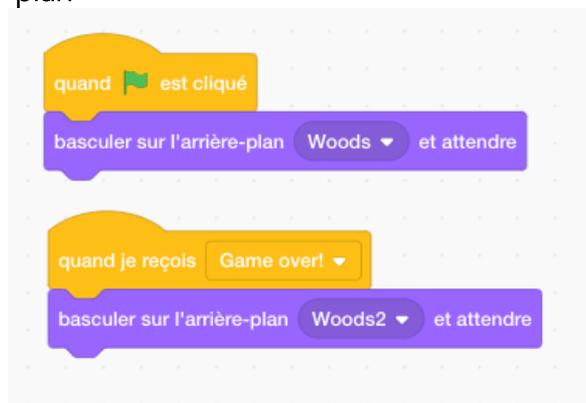
Code (Instructions) Pour Gobo (le 'bon sprite')



Code pour le démon



Code pour l'arrière-plan



Exercice

Demandez aux élèves de créer leur propre jeu, en se basant sur ce qu'ils ont déjà appris dans cette leçon ainsi que dans d'autres leçons, pour la prochaine classe, en utilisant des sprites qu'ils ont dessinés eux-mêmes.

Encouragez chaque personne ou groupe à expliquer et à démontrer ses propres créations de projets à toute la classe.

Leçon 26 - Extension du jeu Chasseur de démons

Jouabilité - résumé du plan de codage

Dans cette leçon, les élèves prolongeront le jeu Chasseur de démons en ajoutant des trésors à collectionner pour le bon sprite (Gobo), en maintenant un score et en modifiant les niveaux lorsque le score atteint certaines valeurs. Un joueur perdra la partie si le nombre de vies atteint zéro et gagnera la partie si le niveau 3 est atteint.



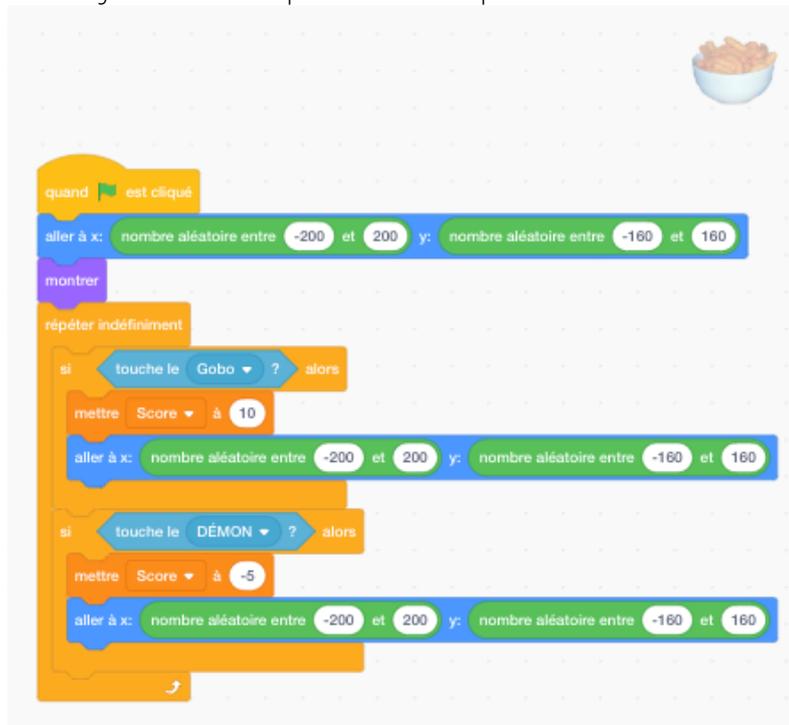
Nous ajoutons d'abord un nouveau sprite pour le trésor (ici un bol de « Cheesy Puffs »).



Puis nous ajoutons deux nouvelles variables, une nommée « Score » et l'autre nommée « Niveau »

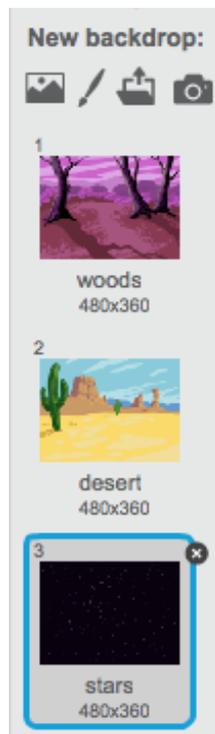


Nous ajoutons le script suivant au sprite « trésor » :

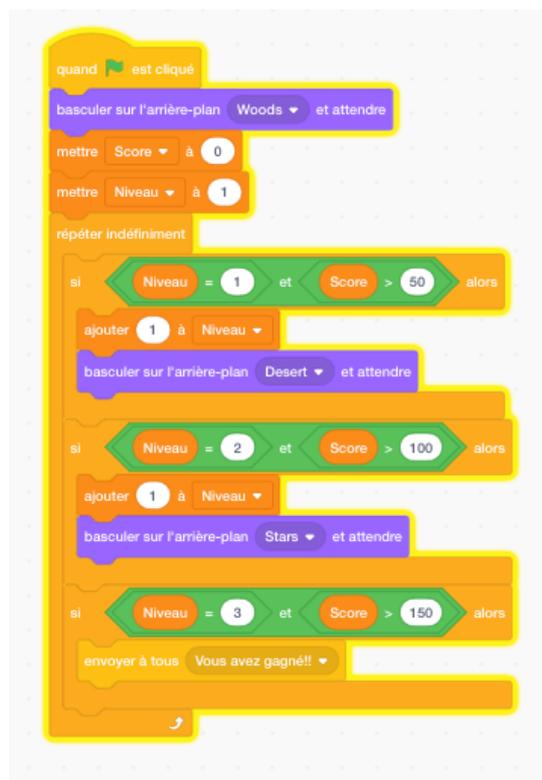


Ce script déplace le sprite trésor vers une partie aléatoire de l'écran lorsque le jeu commence. Chaque fois que Gobo touche le trésor, le score augmente de 10 points et le trésor est déplacé vers une nouvelle position aléatoire. Si le Démon touche le trésor alors le score est réduit de 5 points et le trésor est déplacé vers une nouvelle position aléatoire.

Maintenant, nous supprimons l'arrière-plan « woods1 » d'origine contenant le message « Game Over ». Nous ajoutons 2 nouveaux arrière-plans à la scène. Lorsque le niveau changera, nous changerons l'arrière-plan.



Nous ajoutons le script suivant à la scène pour contrôler les niveaux :



Lorsque le drapeau vert est cliqué, nous fixons le score à 0, le niveau à 1 et l'arrière-plan à « woods » (le premier arrière-plan). Nous ajoutons une boucle *répéter indéfiniment* pour continuer à vérifier le score et pour passer au niveau suivant si nécessaire. Si nous sommes au niveau 1 et que le score est plus de 50, alors nous passons au niveau 2 et nous basculons sur l'arrière-plan suivant. Pareillement, si on est au niveau 2 et que le score est plus de 100, nous passons au niveau 3 et basculons

sur l'arrière-plan 3. Si on est au niveau 3 et que le score est plus de 150, alors le joueur gagne et nous diffusons un message « Vous avez gagné ».

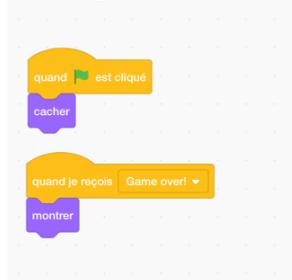
Maintenant nous ajoutons 2 sprites pour indiquer si un joueur a gagné ou perdu le jeu. Utiliser des sprites pour cela nous permet simplement de montrer un message « gagné » ou « perdu » sur l'arrière-plan actuel :

Nous ajoutons le script suivant au sprite « Vous avez gagné » :



Lorsque le jeu commence, nous cachons le sprite « Vous avez gagné ». Si nous recevons le message « Vous avez gagné » diffusé pour la scène, alors nous le montrons.

Nous ajoutons le script suivant au sprite « Vous avez perdu » :



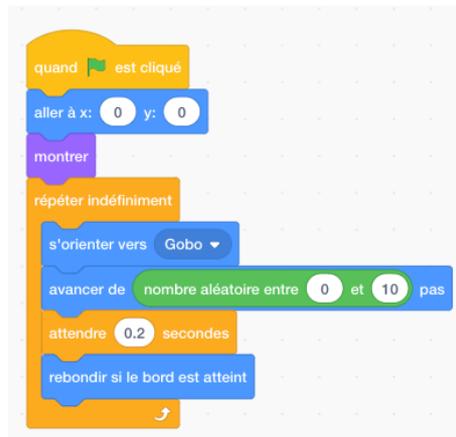
Lorsque le jeu commence, nous cachons le sprite « Vous avez perdu ». Si nous recevons le message « Game Over » diffusé pour le bon sprite, alors nous le montrons.

Nous ajoutons les scripts suivants aux sprites « bon sprite », « méchant sprite » et « trésor », pour les masquer lorsque le jeu est fini :



Pour rendre le jeu un peu plus difficile, nous pouvons faire en sorte que le Méchant Sprite suive activement le bon sprite au lieu de simplement se déplacer de façon aléatoire. Nous pouvons aussi le faire bouger plus rapidement à mesure que les

niveaux augmentent. Pour ce faire, nous modifions le script du méchant sprite comme suit :



Dans ce script, nous dirigeons continuellement le méchant sprite vers le bon sprite et le faisons avancer un nombre aléatoire de pas vers le bon sprite. Le nombre maximum de pas que le méchant sprite peut effectuer est 10 fois supérieur au niveau actuel. Cela signifie qu'il a tendance à accélérer au fur et à mesure que les niveaux augmentent, rendant le jeu plus difficile à chaque nouveau niveau.

Exercice

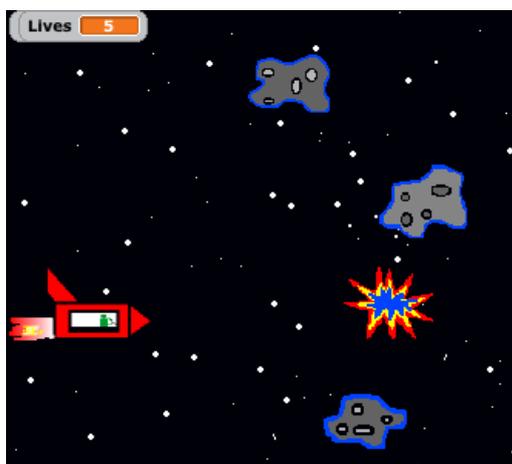
Demandez aux élèves de créer un jeu d'aventure basé sur un héros mythologique légendaire de leur propre pays / culture, qui doit combattre des démons et des monstres dans les montagnes, dans les mers et dans les grottes afin de gagner un trésor.

Leçon 27 – Jeu de tir : les astéroïdes

Les jeux de **tir** sont basés sur un environnement dans lequel un personnage principal obtient des points pour toucher un autre personnage ou un ensemble de personnages. Ce genre est souvent assez difficile et peut tester la vitesse et le temps de réaction du joueur. Souvent, le personnage-joueur, s'il survit à des obstacles, ne perd pas de vie et revendique un certain score, il peut ensuite avancer vers un autre niveau ou une autre mission.

Astéroïdes

Dans cette version simplifiée du jeu classique *Astéroïdes*, les élèves découvriront pour la première fois comment créer un sprite imitant les mouvements d'action rapides d'un laser (ou d'une balle).



Jouabilité - Résumé du plan de codage

Le vaisseau spatial entre dans un champ d'astéroïdes. Le pilote doit détruire les roches astrales pour sauver le vaisseau.

Le vaisseau ne peut survivre que trois fois avant d'être détruit.

Mais pour chaque coup direct, le pilote gagne un point.

Tout d'abord, sélectionnez l'arrière-plan « *Stars* » depuis le dossier *Espace*.

Le sprite vaisseau spatial

Dessinez (ou sélectionnez depuis la galerie des sprites) un vaisseau spatial avec trois costumes, dont un qui représente une explosion.

Les deux premiers costumes, une fois codés, donneront l'impression que le vaisseau spatial se déplace rapidement dans l'espace en raison de la faible différence de taille et de forme du vaisseau, en particulier de la largeur ou de la longueur des flammes du booster de fusée.



Configurez les *Vies* à l'aide de l'option *Ma variable* du dossier *Variables*.
Sélectionnez un maximum de 5 vies.
Créez un moniteur de *Score* en utilisant la même procédure en utilisant zéro (0) comme chiffre de départ.



Contrôlez le *mouvement* du *vaisseau spatial* uniquement à l'aide des touches flèches *Haut* et *Bas*.



Placez un code d'effet de son adéquat



Dessinez un sprite d'**astéroïde** avec deux costumes.



Dans le premier costume, l'astéroïde devrait être gris.

Sélectionnez les numéros suivants lors du dessin :

Couleur : 25 ; Saturation : 0 ; Luminosité : 65.

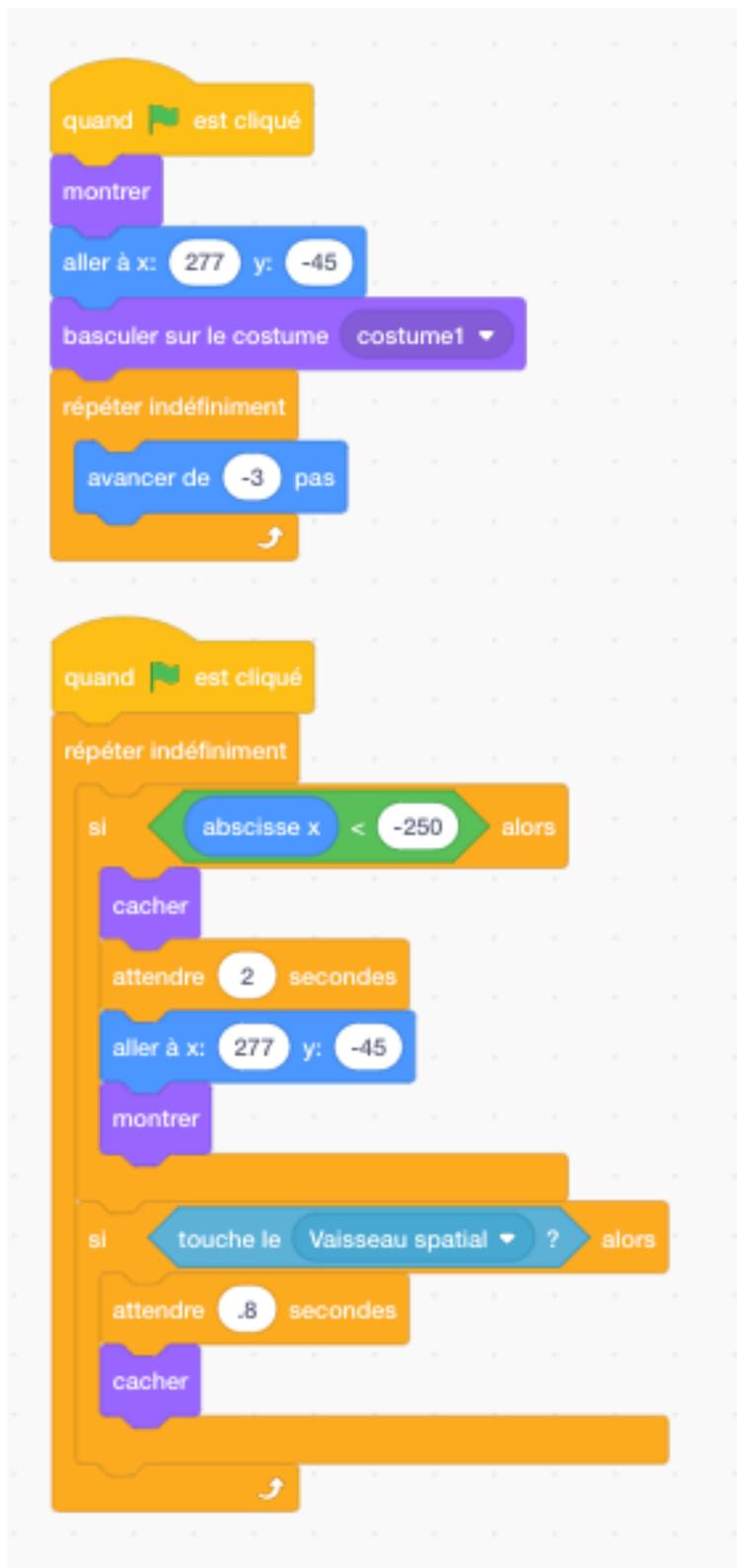
Le deuxième costume devrait représenter une explosion.

Maintenant dupliquez l'astéroïde au moins quatre fois.

Chaque script d'astéroïde devrait avoir un axe X et Y différent afin d'assurer que les sprites ne démarrent pas du même endroit ni ne suivent le même trajet de vol.

Par ailleurs, il y aura plus d'ambiance si ces roches en mouvement disparaissent juste avant d'atteindre le bord gauche de l'écran et de réapparaître une ou deux secondes plus tard en sortant du côté droit.

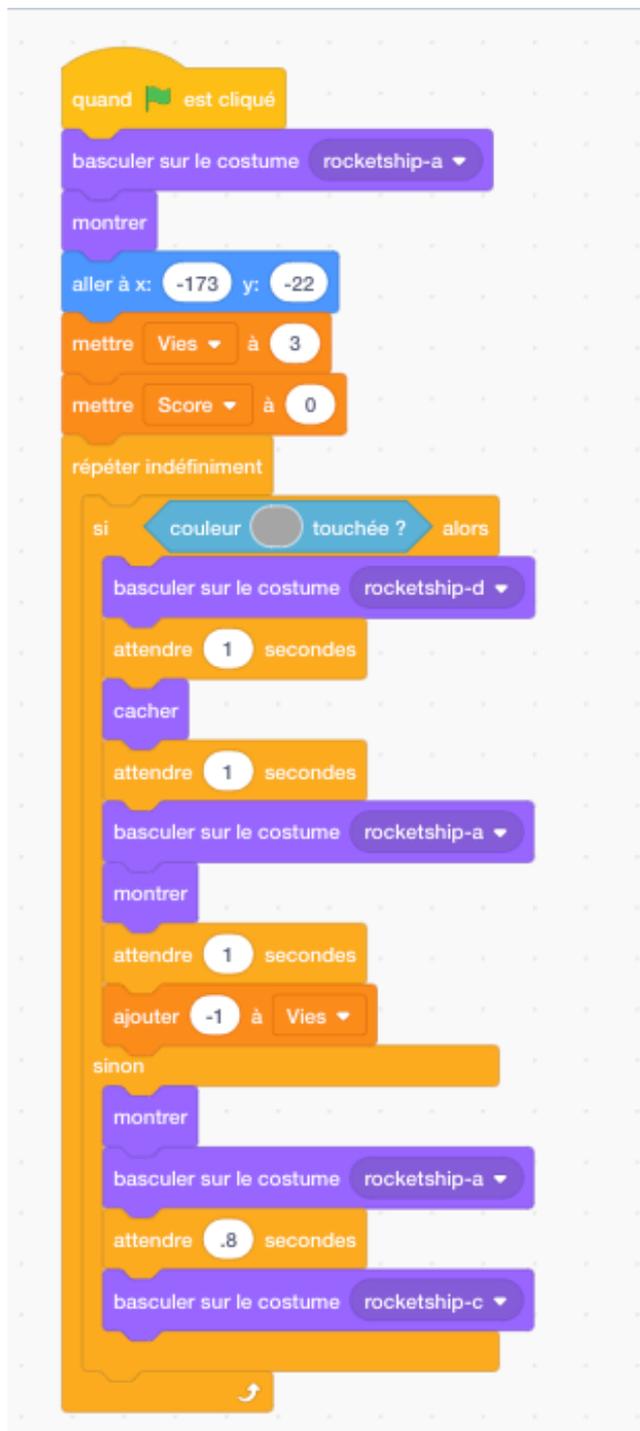
Ceci est fait en utilisant le code suivant (avec chaque sprite de roche ayant différentes coordonnées X et Y) :



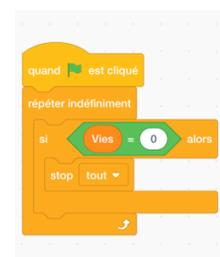
Revenons maintenant au script du Vaisseau spatial, car nous devons programmer des commandes qui indiqueront une collision avec un astéroïde.

Comme il existe plusieurs astéroïdes, il faudrait beaucoup de codes pour reconnaître chacun des sprites individuels. Cependant, nous identifions collectivement tous les astéroïdes en utilisant la couleur d'identification **grise**.

Donc le script de codage principal du vaisseau spatial devrait donc se lire comme suit :



Enfin, entrez le code qui arrêtera le jeu une fois que toutes les vies du vaisseau spatial auront été perdues :



Le Sprite Laser

Dessinez un nouveau sprite qui est constitué d'une ligne courte et plutôt épaisse.

Copiez ce costume six ou sept fois.

Donnez une couleur différente à chaque costume.

Ceci donnera l'impression d'un *tir* lorsqu'il sera codé.



Alignez le sprite Laser avec le vaisseau spatial en utilisant  dans la catégorie Mouvement.

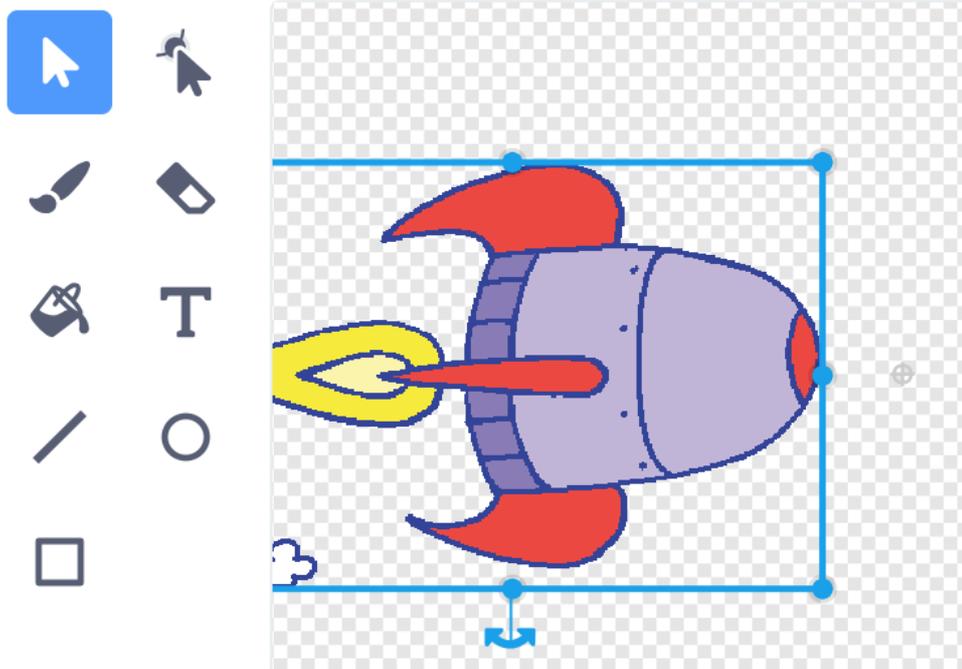
Cependant, vous devez aller dans *l'Éditeur de dessin* du costume du sprite du vaisseau pour obtenir un meilleur alignement entre les deux sprites.

Cela fait, déplacez le sprite vaisseau spatial du milieu de l'écran dans *l'Éditeur de dessin* à l'aide de l'icône de positionnement *Flèche* située dans le coin supérieur gauche du menu outil en mode *Vecteur*.

Toutefois, si le sprite comprend plusieurs éléments pouvant se déplacer indépendamment les uns des autres à l'aide de cet outil, accédez à l'outil situé en haut à droite du menu qui permet de le manipuler comme une seule entité.



Vous verrez un très petit marquage *cible*. Déplacez le sprite vaisseau spatial vers la gauche et devant la cible.

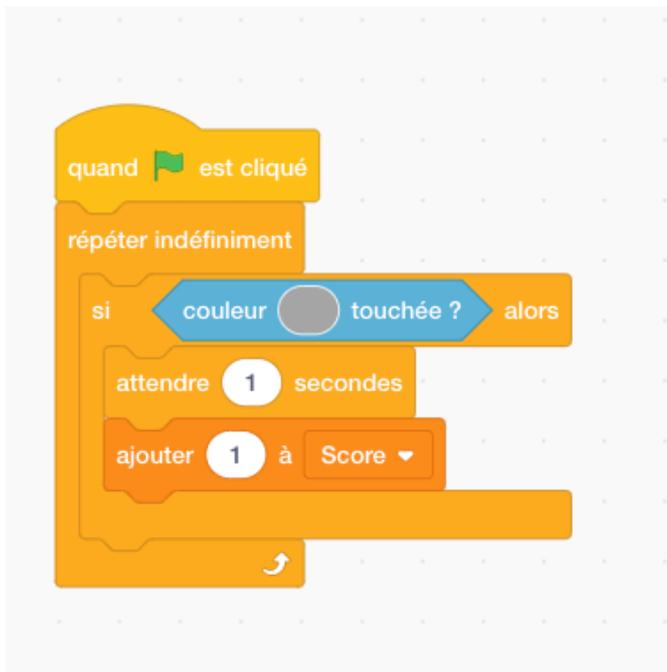


Lancez le projet de script complet et voyez si cela fonctionne. Vous devrez peut-être tester plusieurs fois le déplacement du sprite vaisseau spatial dans l'éditeur de dessin jusqu'à ce que vous sentiez que l'alignement entre les deux sprites est satisfaisant. Maintenant, nous allons construire un code qui donnera l'impression visuelle que, lorsque la barre d'espace est touchée, un laser est tiré du vaisseau spatial dans une ligne de tir droite en direction de l'astéroïde.

```

quand la touche espace est pressée
  aller à x: -200 y: -25
  cacher
  répéter jusqu'à ce que abscisse x > 200
    montrer
    avancer de 60 pas
    attendre .0001 secondes
    costume suivant
  si abscisse x > 200 alors
    cacher
  
```

Le code ci-dessous inclut des commandes qui enregistrent les touches du laser sur un astéroïde :



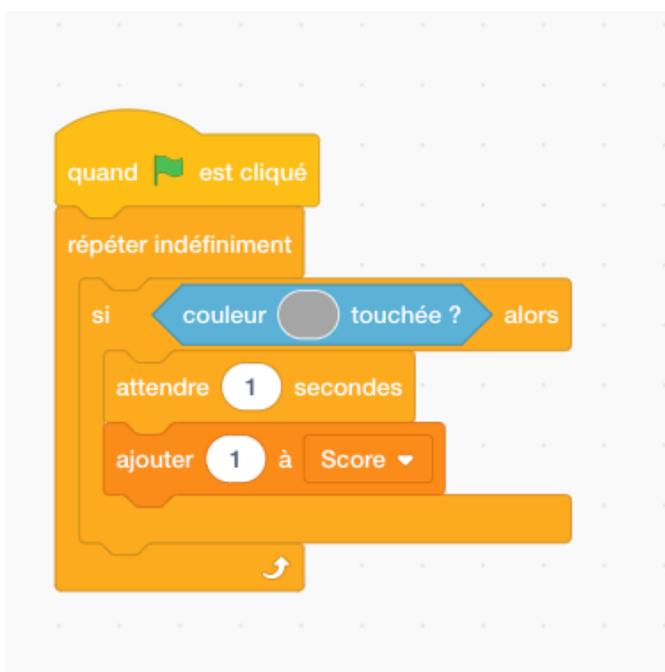
Toutefois, assurez-vous que, dans le



bloc, vous entrez les mêmes numéros dans la boîte de couleur que lors de la coloration de l'astéroïde. Sinon aucun score ne sera enregistré lorsque vous toucherez un astéroïde.

Ces chiffres sont :

Couleur : 25 ; Saturation : 0 ; Luminosité : 65.



Leçon 28 – Jeux à deux joueurs

Le travail d'équipe est un pilier essentiel de la pédagogie Scratch qui recommande des travaux pratiques réguliers impliquant deux étudiants ou plus travaillant sur un même projet.

De même, les participants sont encouragés à créer des jeux impliquant plusieurs joueurs, car les jeux informatiques modernes ont tendance à se jouer dans un environnement de réseau social interactif en ligne.

Attaquant de foot

Les jeux de type tir sont très populaires.

Jouabilité - résumé du plan de codage

Un joueur, l'attaquant, essaie de marquer 5 tirs. L'autre joueur, le gardien de but, essaie de sauver 5 tirs. Le premier qui atteint son objectif gagne le jeu.

Le gardien de but se déplace en utilisant les touches W et S. L'attaquant et le ballon se déplacent le long de l'axe Y en utilisant les touches flèches Haut et Bas. Pour tirer, l'attaquant clique sur la barre d'espace. Le positionnement de ces quatre touches de mouvement permet à deux joueurs d'utiliser le clavier en même temps.

Le programme est réinitialisé immédiatement après que les tirs soient sur la cible, hors cible ou après le sauvetage des tirs. Ceci continue jusqu'à ce que l'un des deux joueurs atteigne cinq sur l'écran de sauvegarde / score.

Ce jeu contient également notre première rencontre avec une séquence de **Questions et Réponses**.

A l'origine il a été développé par Philip, âgé de huit ans.

Blocs de commandes clés : commande de capteurs « Poser une Question et Réponse », « Couleur touchée », de la commande « Si », Variables & Opérateurs, « Quand je reçois » et « Envoyer à tous ».

Notez que la commande *Demander* (dossier de capteurs) permet à l'utilisateur d'insérer une réponse.

La véracité ou la fausseté de la réponse et des réponses résultantes sont contrôlées par les commandes insérées.



Dessinez une partie appropriée d'un terrain de foot.

Mettez des sprites qui prendront les rôles d'attaquant, de gardien de but, de ballon de foot et d'hôte officiel.

Remarque : dans les scripts ci-dessous, toutes les positions X et Y pour l'attaquant, le gardien de but, le ballon de football et l'hôte sont tous spécifiques au projet de démonstration et ne seront conformes à aucun autre projet. Ils n'existent seulement comme lignes directives.

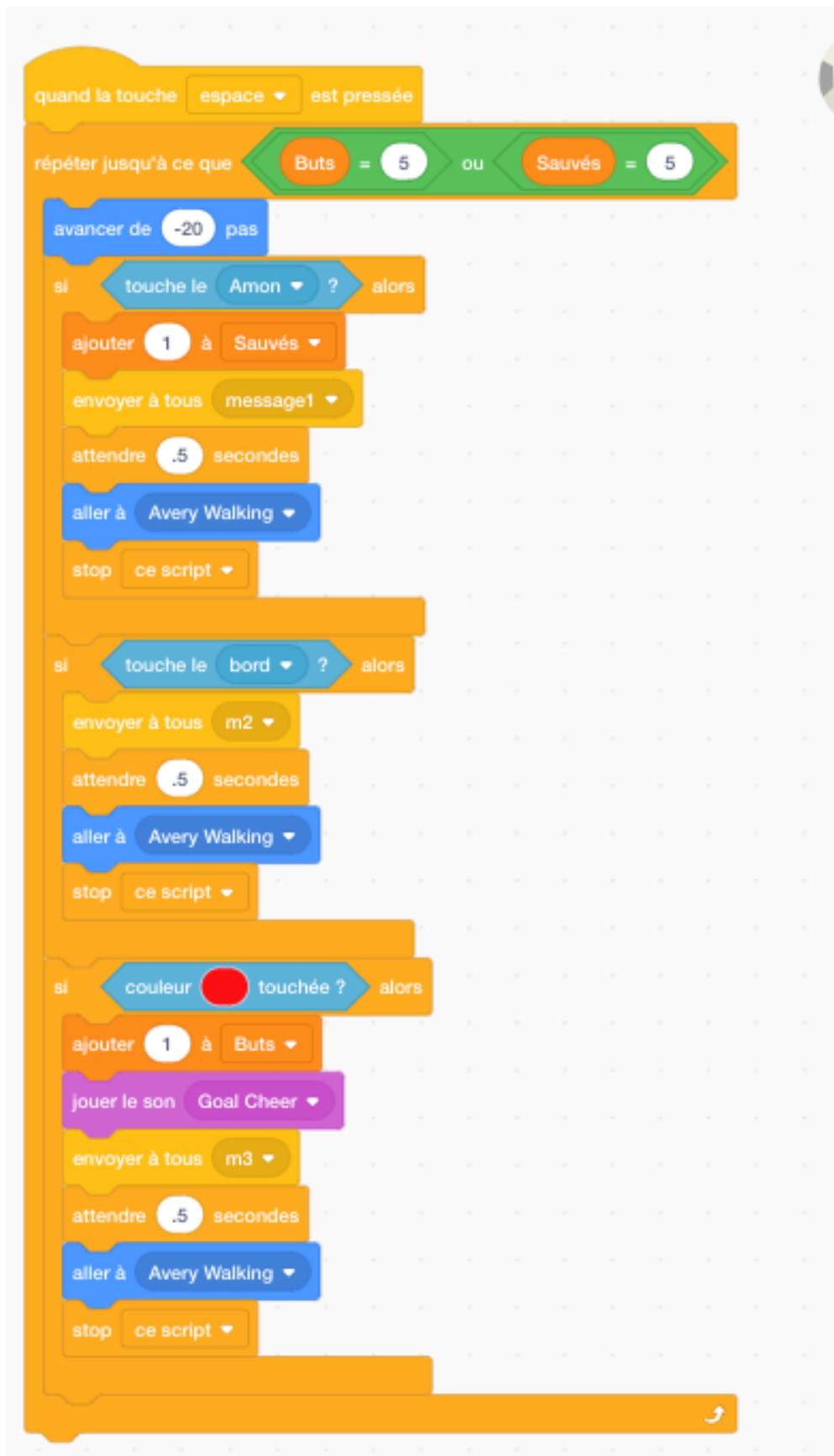
Script : Le sprite MC ou Hôte

```
quand [drapeau] est cliqué
mettre x à -96
mettre y à -114
montrer
demander "Veux-tu jouer au foot ?" et attendre
si réponse = oui alors
envoyer à tous "Oui"
dire "Utilise W et S pour déplacer le Gardien" pendant 2.5 secondes
dire "Utilise la flèche du Haut et du Bas pour déplacer l'Attaquant" pendant 2.5 secondes
dire "Utilise la touche Espace pour tirer" pendant 2 secondes
dire "Prêt ?" pendant 1 secondes
jouer le son "Gong"
cacher
sinon
dire "Le Match est annulé !" pendant 2 secondes
envoyer à tous "Annulé"
cacher
```

Scripts : sprite ballon

```
quand la touche flèche haut est pressée
mettre y à 10

quand la touche flèche bas est pressée
mettre y à -10
```



Remarque : Le  dans le script ci-dessus fait référence à la *couleur* des filets du but qui doivent être touchés par le ballon pour enregistrer un score.



Dans le texte ci-dessus, le bloc  apparaît à côté de l'attaquant.

Cependant, vous aurez probablement besoin d'entrer dans l'*Éditeur de dessin* des costumes du sprite de l'attaquant afin de placer le ballon aux pieds de l'attaquant.

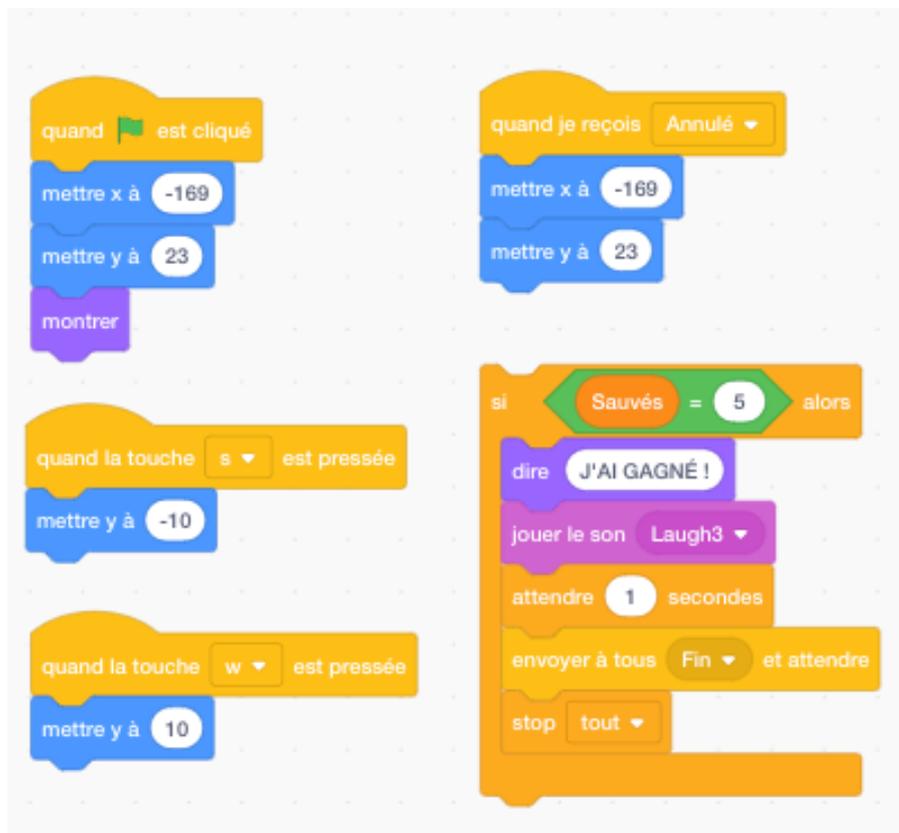
Cela fait, déplacez le sprite de l'attaquant au milieu de l'écran dans l'*Éditeur de dessin* à l'aide de l'icône de positionnement de *flèche* située dans le coin supérieur gauche du menu Outils en mode *vecteur*. Cependant, si le sprite comprend plusieurs éléments pouvant se déplacer indépendamment les uns des autres à l'aide de cet



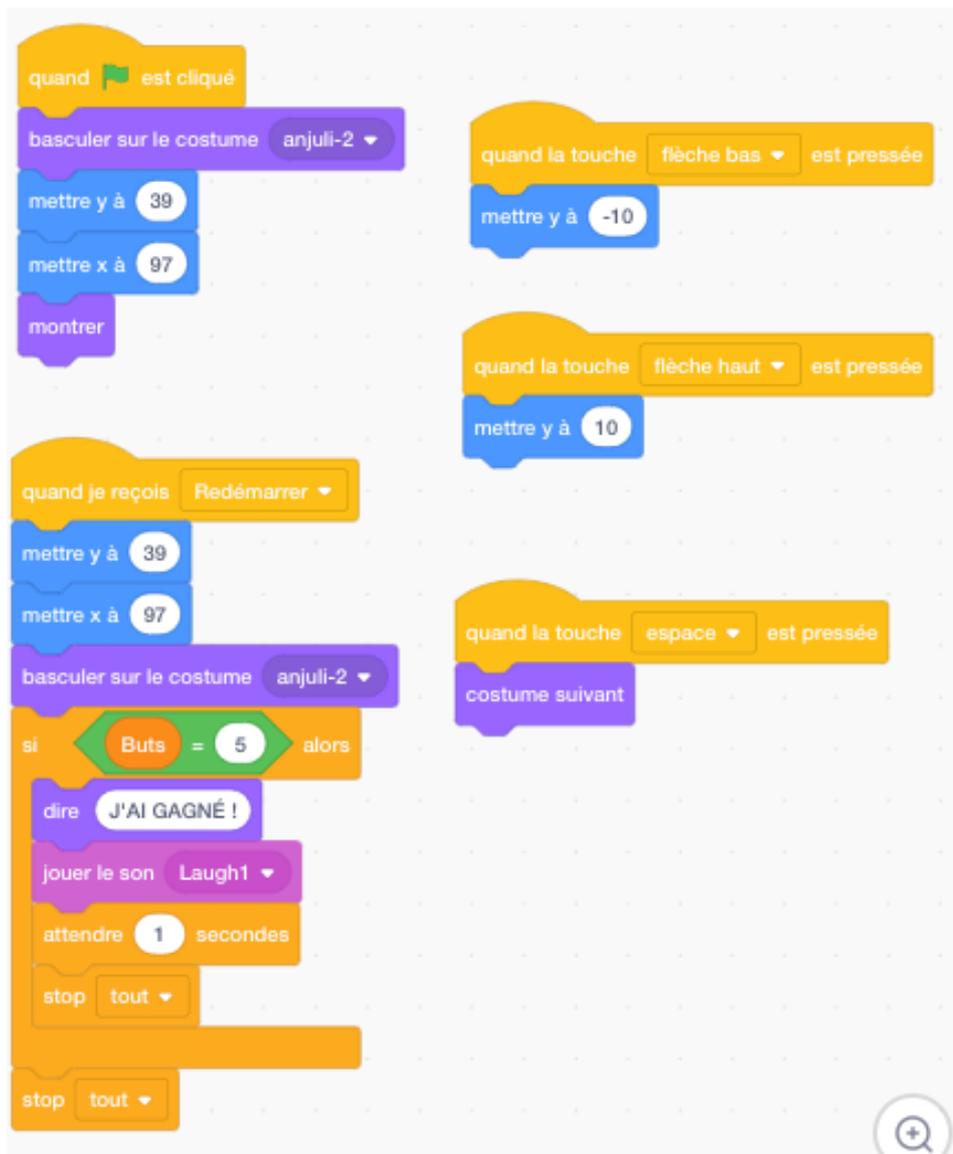
outil, accédez à l'outil  situé en haut à droite du menu qui permet de le manipuler comme une seule entité.

Une fois que vous déplacez le sprite, vous verrez un très petit marquage de *cible*. Déplacez le sprite attaquant jusqu'à ce que ses pieds soient derrière le sprite ballon. Regarder la scène vous aidera à obtenir le bon alignement.

Scripts : sprite gardien de but



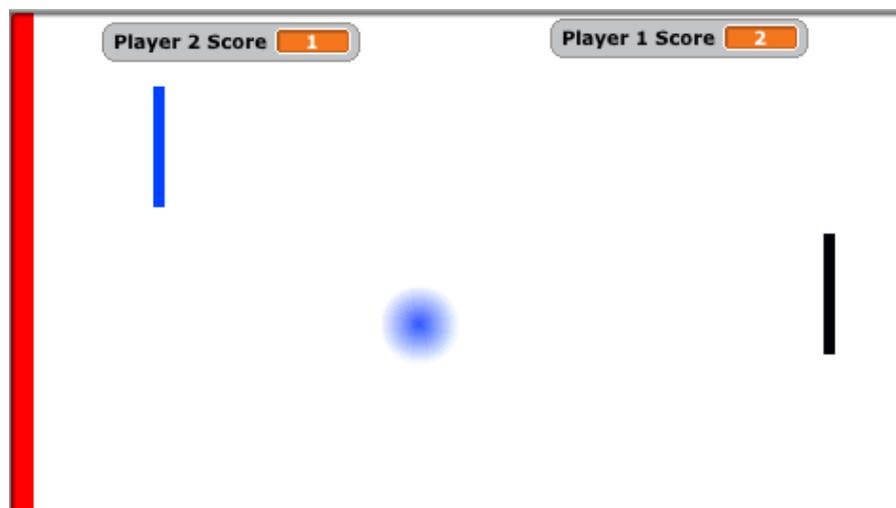
Script : sprite attaquant



Exercice

Créez un *jeu de hockey* ou un autre jeu sportif basé sur une structure identique ou similaire au jeu *Attaquant de Foot* ci-dessus.

Leçon 29 - Jeux à deux joueurs - tennis pour deux



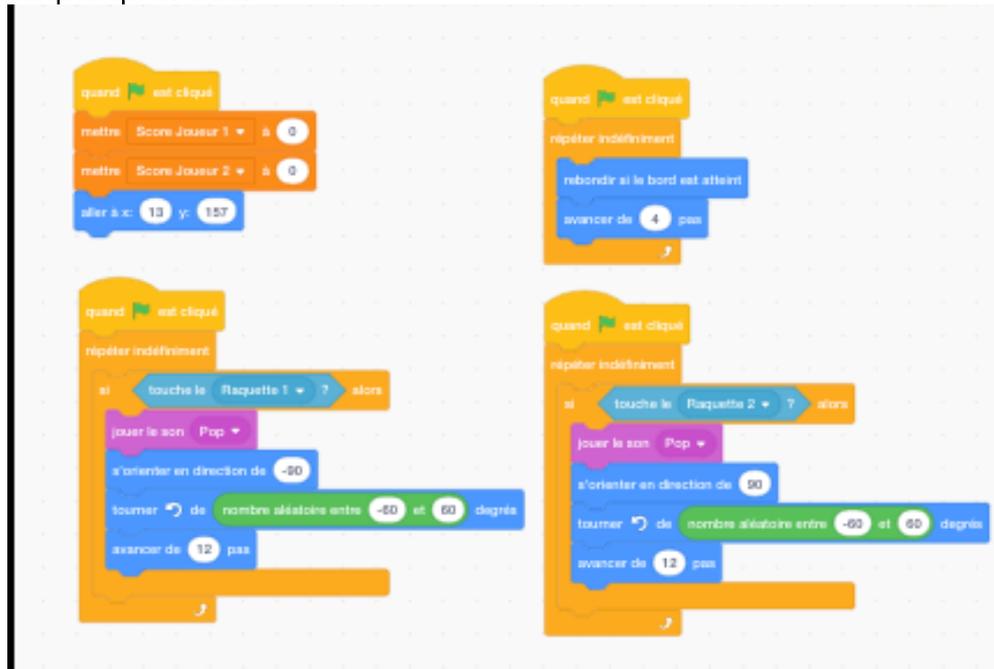
Encore plus que la version à un seul joueur utilisée dans une leçon précédente, ce jeu de tennis à deux joueurs est plus proche du jeu classique d'origine appelé **Pong**.

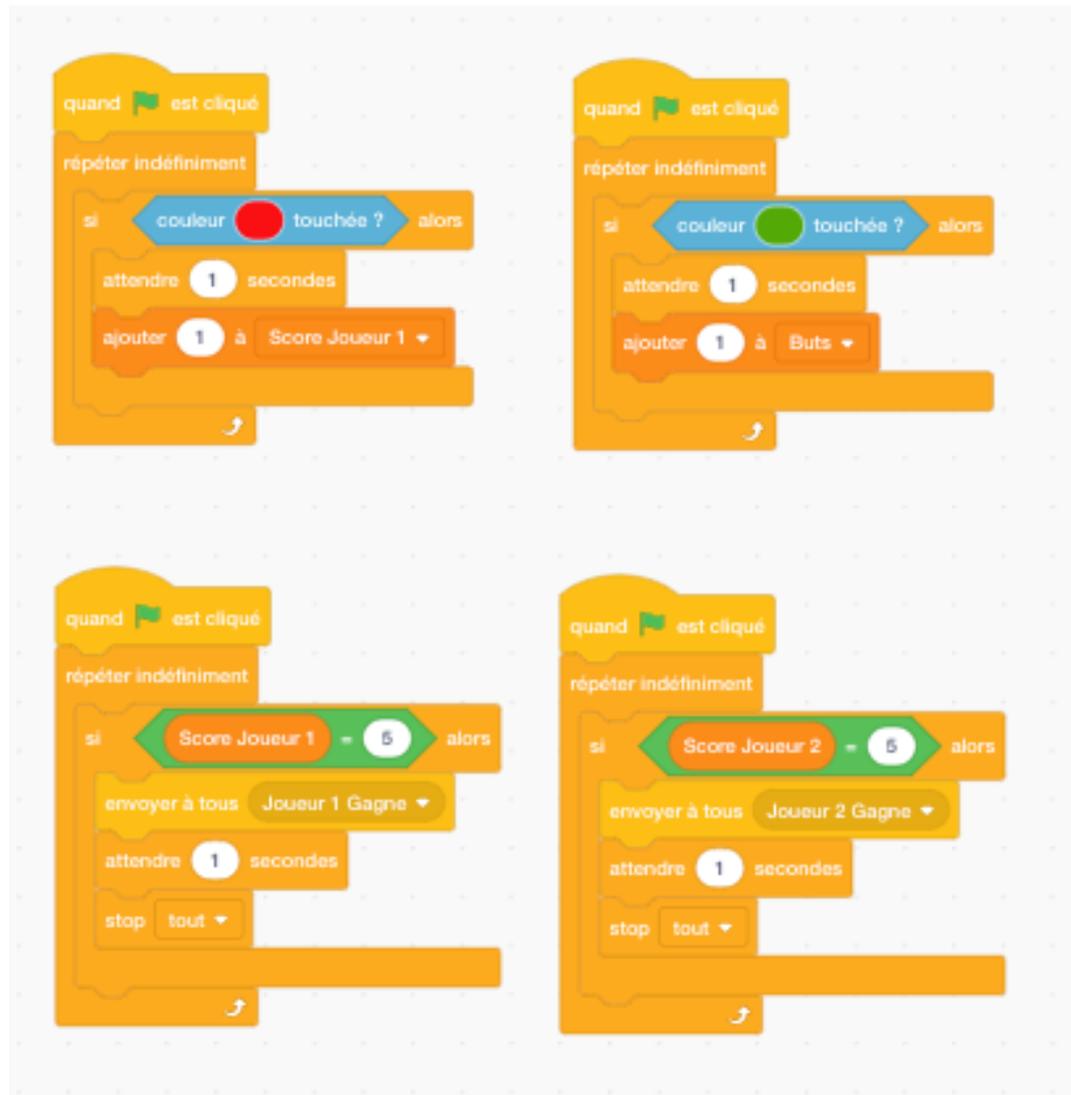
Jouabilité - Résumé du plan de codage

Le but du jeu est que l'un des deux joueurs soit déclaré vainqueur en faisant en sorte que la balle touche la ligne de couleur du côté de l'adversaire à cinq reprises. Les Raquettes agissent à la fois comme boucliers défensifs et comme tireurs offensifs. Chaque raquette est contrôlée par un ensemble séparé de deux touches adjacentes (flèches Haut / Bas et W / S).

Blocs de commandes clés : « répéter indéfiniment », « Quand je reçois et Envoyer à tous » et « Stop tout » (dossier Contrôle), « Touche le (un sprite) » (Capteurs), « Mettre ___ à » et « Ajouter ___ à » (Variables), « Nombre aléatoire entre ___ et ___ » (Opérateurs), « S'orienter en direction de ___ », « Tourner » (Mouvement).

Script : sprite balle

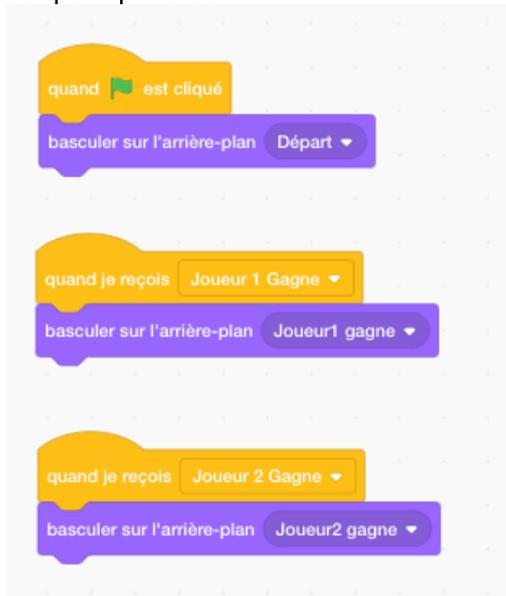




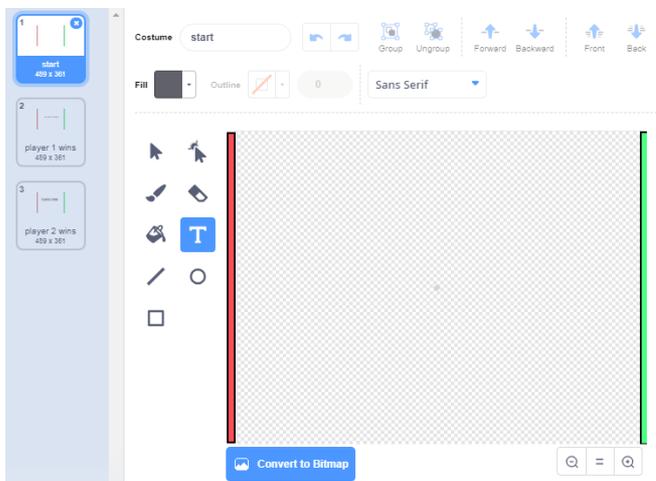
Script : sprite raquette 1



Script : sprite scène



Costumes script scène : a) version standard avec 2 lignes de but colorées aux extrémités opposées de l'écran, b) le joueur 1 gagne ! et c) Le Joueur 2 gagne !



Leçon 30 – Quizz de géographie : voyager à travers un continent

Coder un **Quizz** à l'aide de Scratch est une activité répandue dans les écoles. Les thèmes vont des connaissances générales à un sujet spécifique. Dans ce dernier domaine, il peut donner une nouvelle dimension passionnante à l'enseignement des langues, des sciences, de l'histoire et de la géographie. Les élèves peuvent avoir l'occasion de rechercher les questions et réponses dans la matière de leur choix ainsi que de créer un projet passionnant de **questions et réponses** qui captera l'intérêt des participants.

L'exemple de projet ci-dessous est celui d'un tour d'Europe. Les questions ici sont basées sur l'appellation des pays. Mais cela pourrait être basé sur la désignation des langues principales, des capitales, des chefs d'État, des destinations touristiques populaires, des montagnes, des rivières, des lacs, etc.



Jouabilité - résumé du plan de codage

Le guide s'appelle Daire, il parcourt l'Europe en s'arrêtant dans chaque pays pour demander à un joueur de taper son nom à l'écran.

La personne interrogée ne peut pas quitter le pays avant d'avoir tapé la bonne réponse. Une fois que la bonne réponse est entrée, Daire se rend dans le prochain pays et pose ensuite la question suivante jusqu'à ce que tous les pays sélectionnés soient visités. Ce processus se poursuit jusqu'à ce que tous les pays soient visités.

Blocs de commandes clés : « répéter indéfiniment », « Si__alors », « Répéter jusqu'à ce que__ », « Quand je reçois et Envoyer à tous » et « Stop tout » (dossier Contrôle), « Dire » (Apparence), « Touche le (une couleur) », « Demander », « Réponse » (Capteurs), « Mettre __ à » et « Ajouter__à » (Variables), « __ et__ » (Opérateurs), « S'orienter en direction de__ » (Mouvement), « Jouer son__ » (Son).

Un Sprite (Daire) et un arrière-plan (carte de l'Europe) avec un seul script pour le premier.

Scripts :

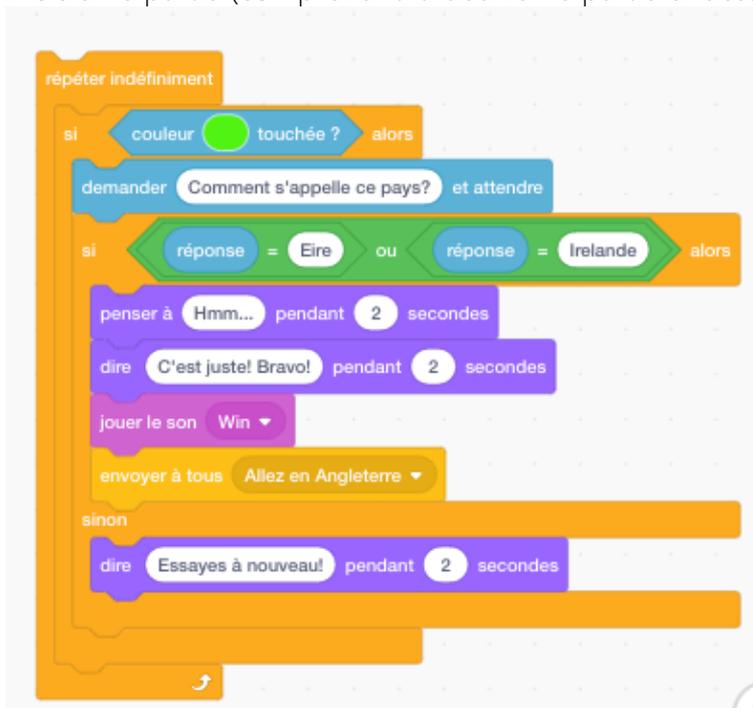
Première partie



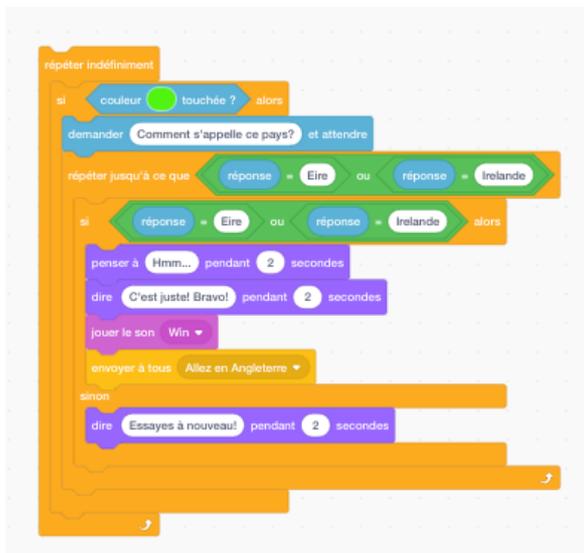
Deuxième partie



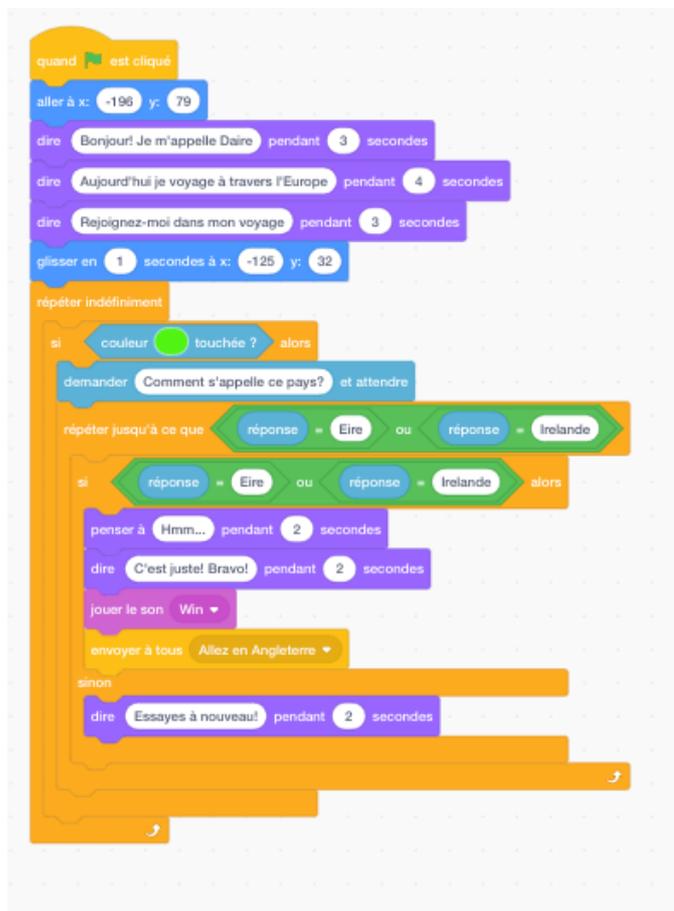
Troisième partie (comprenant la deuxième partie ci-dessus)



Quatrième partie (comportant la troisième partie ci-dessus)

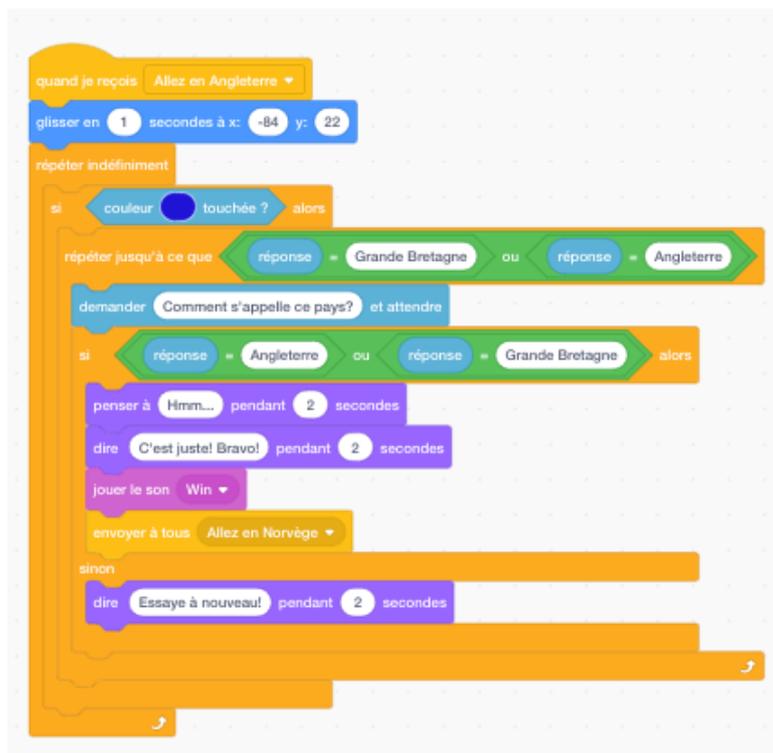


Quatrième partie (comportant les première et troisième parties)



Remarque : Éire ou Eire est le nom irlandais de l'Irlande.

Script pour le pays Angleterre



Suivez la même procédure pour tous les autres pays marqués.

Exercice

Demandez aux enfants ans la classe ou aux participants à la session, individuellement ou en groupe, de réaliser un projet similaire basé sur le continent africain.

Cela nécessitera beaucoup de recherche et de planification en avance afin d'obtenir les informations pertinentes nécessaires (par exemple : capitales de pays, langues parlées, caractéristiques géographiques telles que principaux fleuves, montagnes, etc.).

ⁱ Remarque : Tous les plans de cours sont préparés par Brendan Smith, Camden Education Trust, Galway, Irlande.