



INFORMATIQUE

Sequence 0 : From scratch

TP 0.1

v1

IUT d'Annecy, 9 rue de l'Arc en Ciel, 74940 Annecy

LES CONCEPTS DE LA PROGRAMMATION AVEC SCRATCH

1 TP 0 : Introduction à la programmation avec Scratch



Objectif du TP

Pour cette semaine 0, nous allons (re)découvrir les bases de la programmation à travers le langage Scratch.

L'utilisation de Scratch nous permettra de nous familiariser avec les concepts fondamentaux de la programmation sans avoir à se soucier de la syntaxe complexe des langages de programmation traditionnels.

A la fin de cette séance, vous connaîtrez les principaux concepts de la programmation :

- Variables
- Boucles
- Conditions

Durant toute cette séance, s'il y a le moindre détail qui n'est pas clair pour vous, n'hésitez pas à poser une question à l'enseignant ou à un camarade.



Anglais ou français ?

Sur cette page, les images des blocs sont en anglais. Vous pouvez utiliser Scratch dans la langue de votre choix.

1.1 Présentation de l'interface

L'interface de Scratch est divisée en plusieurs sections :

- **Palette de blocs** : où vous trouverez les différents blocs de code que vous pouvez utiliser
- **Mouvement** : pour déplacer les sprites
- **Apparence** : pour modifier l'apparence des sprites -> Parler, changer de costume, etc.
- **Événements** : pour déclencher des actions
- **Contrôle** : pour gérer le flux du programme
- **Sons** : pour ajouter des effets sonores
- **Zone de script** : où vous assemblez les blocs pour créer votre programme
- **Zone de scène** : où vous pouvez voir votre projet en action
- **Liste des sprites** : où vous pouvez ajouter ou sélectionner les sprites de votre projet

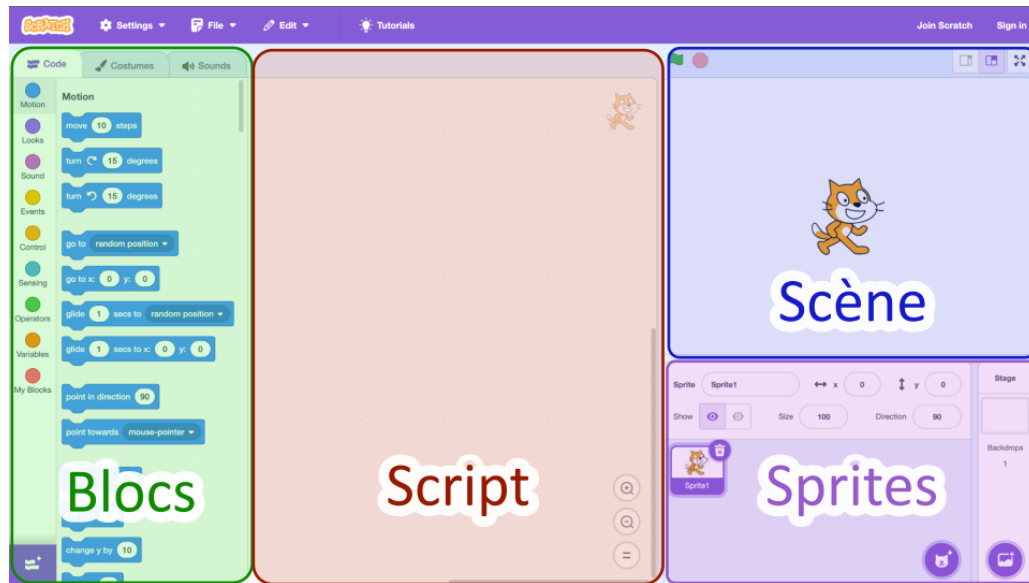


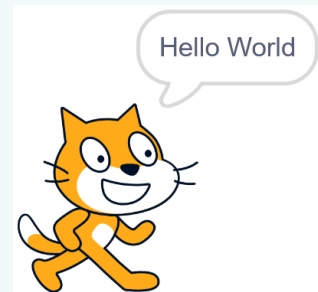
FIGURE 1 – Description de l'interface Scratch

2 Là où tout commence : Hello World

Pour commencer, nous allons créer un programme simple qui affiche "Hello World" à l'écran.


Manipulation 1 : Hello World


- Ouvrez un éditeur Scratch sur le site du MIT
- Dans la palette de blocs, trouvez le bloc "Quand le drapeau vert est cliqué".
- Ajoutez le bloc "dire Bonjour pendant 2 secondes" sous le bloc précédent.
- Modifier le texte pour Hello World
- Cliquez sur le drapeau vert pour exécuter votre programme.
- Vous devriez obtenir le résultat ci-contre.



Besoin d'aide ?

Scratch est bien documenté sur son site internet : documentation de Scratch Vous pouvez également trouver des tutoriels en ligne.

Le bloc  se trouve dans la catégorie "Événements" ("Events").

Le bloc  se trouve dans la catégorie "Apparence".

3 Un premier pas vers la mémoire : les variables

Qu'est-ce qu'une variable ?

Une variable est un espace de stockage pour des données qui peuvent changer au cours de l'exécution du programme.

Par exemple, vous pouvez utiliser une variable pour stocker le score d'un jeu ou le nom d'un utilisateur.

Analogie

On peut imaginer une variable comme une boîte dans laquelle on place des informations. On peut ouvrir cette boîte pour voir ce qu'il y a à l'intérieur, ou y mettre de nouvelles informations.

Sur scratch, on affecte une valeur à une variable en utilisant le bloc "mettre [nom de la variable] à [valeur]".

3.1 Notre première variable

Manipulation 2 : Comment créer une variable Scratch ?

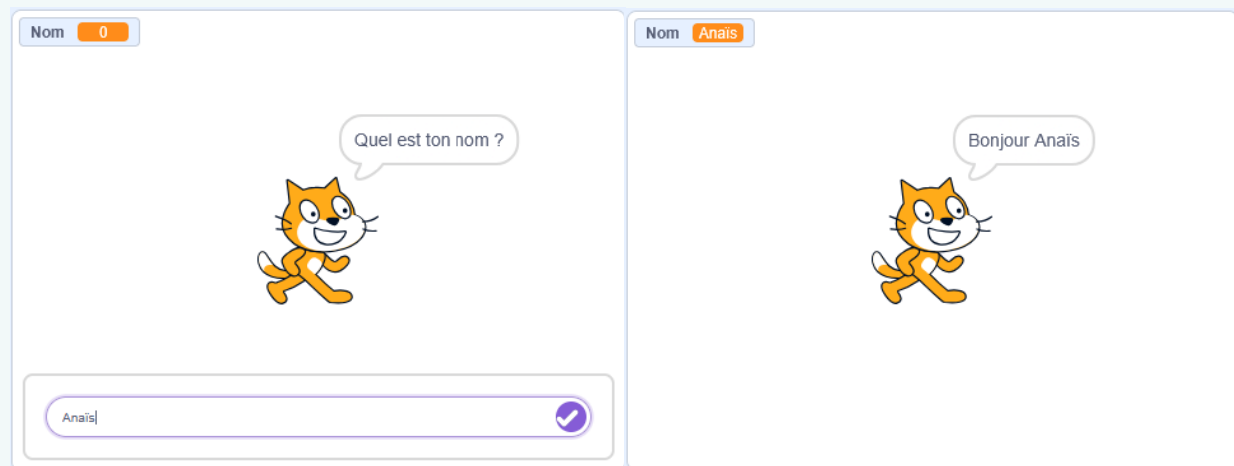
- ☐ Dans la palette de blocs, trouvez la catégorie "Variables".
- ☐ Cliquez sur "Créer une variable" et nommez-la comme vous le souhaitez.

Un bloc orange arrondi **my variable** apparaît alors dans la palette de blocs, c'est votre variable.

Manipulation 3 : Personnaliser le message

Notre premier programme était très impersonnel ! Accueillons l'utilisateur avec un message personnalisé.

- ☐ Créez une variable appelée "Nom".
- ☐ Ajoutez un bloc "demander [Quel est votre nom ?] et attendre" avant le bloc "dire".
- ☐ Le bloc Demander est associé à une variable spéciale appelée "réponse". Ces deux blocs se trouvent dans la catégorie Capteur ("Sensing").
- ☐ Utilisez le bloc "mettre [Nom] à [réponse]" pour stocker la réponse de l'utilisateur dans la variable "Nom".
- ☐ Utiliser un bloc "regrouper" pour rassembler "Bonjour" et la variable "Nom" dans le bloc "dire".
- ☐ Cliquez sur le drapeau vert pour exécuter votre programme.
- ☐ Vous devriez voir le sprite demander votre nom et afficher un message personnalisé.



3.2 Nos premiers calculs

Notre première variable était un texte, mais on peut aussi utiliser des nombres.

Manipulation 4 : Calculer l'année de naissance

- ☐ Créez une variable appelée "Âge" ainsi qu'une variable "Année de naissance".
- ☐ Ajoutez un bloc "demander [Quel est votre âge ?] et attendre" avant le bloc "dire".
- ☐ Utilisez le bloc "mettre [Âge] à [réponse]" pour stocker la réponse de l'utilisateur dans la variable "Âge".
- ☐ Utilisez un bloc "mettre [Année de naissance] à [2025 - Âge]" pour calculer l'année de naissance de l'utilisateur.
- ☐ Utilisez un bloc "dire" pour afficher "Vous êtes né en [Année de naissance]".
- ☐ Cliquez sur le drapeau vert pour exécuter votre programme. Vous devriez voir le sprite demander votre âge et afficher l'année de naissance calculée.

3.3 A vous de jouer

Manipulation 5 : Sprite tient un tacos

- ☐ Créer un programme qui demande à l'utilisateur combien de tacos et de kebab il veut commander.
- ☐ Ce programme répondra le montant total de la commande.
- ☐ Faire valider votre programme par l'enseignant.

4 Conditions : Si, alors, sinon



Qu'est-ce qu'une condition ?

Une condition est une instruction qui permet de prendre des décisions dans un programme.

Par exemple, si l'utilisateur a plus de 18 ans, on peut lui afficher un message différent que s'il a moins de 18 ans.

Sur Scratch, on utilise le bloc "si [condition] alors" pour créer une condition. On peut aussi utiliser le bloc "sinon" pour exécuter un bloc de code si la condition n'est pas remplie.

4.1 Conditions simples

Manipulation 6 : Condition simple

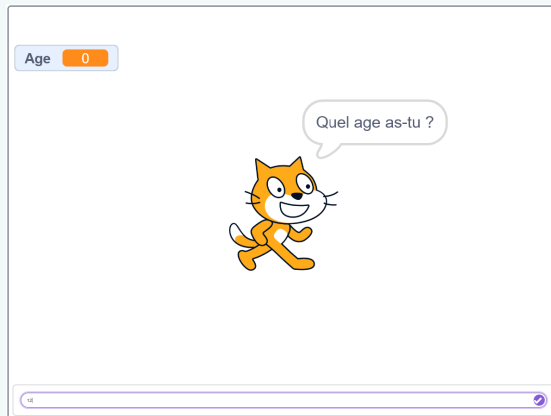
- ☐ Demander à l'utilisateur son âge et stocker la réponse dans une variable "Âge".
- ☐ Utiliser un bloc "si [Âge > 18] alors"
- ☐ Dans le bloc "si", ajouter un bloc "dire [Vous êtes majeur]" pour afficher un message si l'utilisateur est majeur.
- ☐ Ajouter un bloc "sinon" pour afficher un message différent si l'utilisateur est mineur.
- ☐ Cliquez sur le drapeau vert pour exécuter votre programme. Vous devriez voir le sprite demander votre âge et afficher un message différent selon que vous êtes majeur ou mineur.

4.2 Conditions imbriquées

Les conditions peuvent aussi être imbriquées, c'est-à-dire qu'on peut mettre une condition à l'intérieur d'une autre condition.

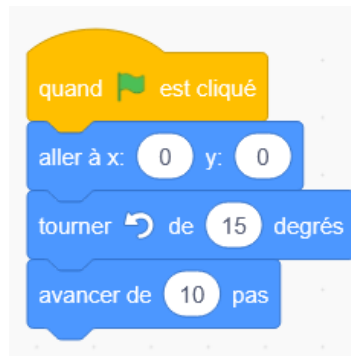
Manipulation 7 : Condition imbriquée

- ☐ Créer un programme qui dit à l'utilisateur s'il est un enfant (0-12 ans), un adolescent (13-17 ans), un adulte (18-64 ans) ou un senior (65 ans et plus).
- ☐ Pour cela, vous devrez utiliser des conditions imbriquées.
- ☐ Faire valider votre programme par l'enseignant.



5 Boucles et répétitions

5.1 Introduction : Un peu de mouvement



Manipulation 8 : Ajouter du mouvement

- Copier le code ci-dessus.
- Cliquez sur le drapeau vert pour exécuter votre programme. Vous devriez voir le sprite se déplacer et tourner.
- Cliquer autant de fois que vous voulez pour faire faire un tour complet à votre sprite.

Question 1 Combien de fois faut-il cliquer pour faire un tour complet ?

Question 2 Comment faire pour que le sprite fasse un tour complet sans avoir à cliquer 24 fois ?

5.2 Boucles



Qu'est-ce qu'une boucle ?

Une boucle est une structure de programmation qui permet de répéter un bloc de code plusieurs fois sans avoir à le réécrire.

Par exemple, si vous voulez faire tourner un sprite de 15 degrés 24 fois, vous pouvez utiliser une boucle pour éviter de répéter le bloc "tourner de 15 degrés" 24 fois.

Sur Scratch, on utilise le bloc "répéter [nombre] fois" pour créer une boucle.

Manipulation 9 : Utiliser une boucle

- Ajoutez un bloc "répéter 24 fois" autour des blocs "tourner de 15 degrés" et "avancer de 10 pas".
- Cliquez sur le drapeau vert pour exécuter votre programme. Vous devriez voir le sprite faire un tour complet en 24 étapes.
- Modifier le programme pour que le sprite fasse deux tours complets.

5.3 Boucles infinies



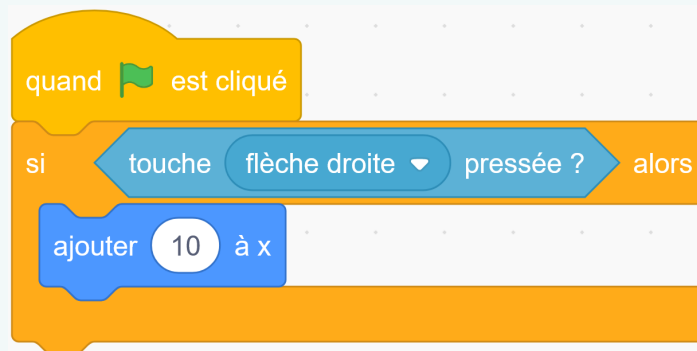
Qu'est-ce qu'une boucle infinie ?

Une boucle infinie est une boucle qui ne s'arrête jamais.

Par exemple, si vous utilisez le bloc "répéter indéfiniment", le programme continuera à exécuter le bloc de code à l'intérieur de la boucle sans jamais s'arrêter.

5.3.1 Déplacer un sprite au clavier

Manipulation 10 : Déplacer un sprite au clavier



☐ Copier le code ci-dessus.

Question 3 Que se passe-t-il si vous appuyez sur la touche "flèche droite" ?



Pourquoi ?

Lorsque le programme est lancé, il vérifie si la touche "flèche droite" est pressée. Si c'est le cas, il déplace le sprite de 10 pas vers la droite. Sinon, il ne fait rien.

Le problème est que le programme ne vérifie pas en continu si la touche "flèche droite" est pressée. Il ne le fait qu'une seule fois au début du programme.

Vous pouvez le vérifier en maintenant la touche "flèche droite" pendant que vous lancez le programme. Le sprite se déplace car la touche est pressée au moment du lancement du programme. Il ne le fait qu'une seule fois, et ne continue pas à vérifier si la touche est pressée.

Manipulation 11 : Corriger le programme

- ☐ Entourez le bloc "si [touche flèche droite pressée] alors" avec un bloc "répéter indéfiniment".
- ☐ Cliquez sur le drapeau vert pour exécuter votre programme. Vous devriez maintenant pouvoir déplacer le sprite en maintenant la touche "flèche droite" enfoncée.
- ☐ Ajoutez des blocs similaires pour les touches "flèche gauche", "flèche haut" et "flèche bas" pour déplacer le sprite dans toutes les directions.
- ☐ **ASTUCE** : Vous pouvez dupliquer tout le bloc "si [touche flèche pressée] alors" avec un clic droit et en sélectionnant "Dupliquer". Cela vous permettra de gagner du temps pour créer les autres directions.
- ☐ Cliquez sur le drapeau vert pour exécuter votre programme. Vous devriez maintenant pouvoir déplacer le sprite dans toutes les directions en utilisant les touches fléchées.

5.3.2 Un premier jeu

Nous allons ajouter un second sprite qu'il faudra attraper.



Un code pour chaque sprite

Chaque sprite peut avoir son propre code. Vous pouvez ajouter un nouveau sprite en cliquant sur le bouton "Choisir un sprite" en bas à droite de l'interface. Chaque sprite peut avoir son propre code, ce qui permet de créer des interactions entre les sprites.

Manipulation 12 : Attraper le sprite

- Ajoutez un nouveau sprite (par exemple, un chat) à votre projet

Dans le code de ce nouveau sprite :

- Ajoutez un bloc "quand le drapeau vert est cliqué"
- Ajoutez un bloc "Aller à position aléatoire" pour que le sprite apparaisse à une position aléatoire sur la scène.
- Ajoutez un bloc "répéter indéfiniment" à l'intérieur duquel vous ajouterez un bloc si touche [votre sprite] alors aller à une position aléatoire.
- Testez votre programme et faites valider votre programme par l'enseignant.

6 A vous de jouer !

A vous de créer un programme Scratch qui vous ressemble et qui montre vos compétences en programmation.

Ce premier rendu peut être un jeu, une animation, une histoire interactive, ou tout autre projet qui vous inspire. Les seules contraintes sont :

Travail à rendre 1

Projet Scratch

- Votre projet doit utiliser au moins deux lutins (sprites), dont au moins un ne doit pas être un chat.
- Votre projet doit comporter au moins trois scripts au total (c'est-à-dire pas nécessairement trois par sprites).
- Votre projet doit utiliser au moins une conditionnelle, au moins une boucle et au moins une variable.
- Votre projet doit utiliser au moins un bloc personnalisé que vous avez créé vous-même (via Créer un bloc), qui doit prendre au moins une entrée.
- Votre projet doit être interactif, c'est-à-dire que l'utilisateur doit pouvoir interagir avec le programme d'une manière ou d'une autre (par exemple, en cliquant sur un sprite, en appuyant sur une touche, etc.).

Le rendu se fera sur la plateforme moodle, où vous devrez soumettre le lien vers votre projet Scratch.



En manque d'idées ?

Voici quelques propositions de projets, classées par niveau de difficulté.

Facile : Pour les moins à l'aise

- Une balle qui rebondit sur les bords de l'écran.
- Deux personnages qui racontent une histoire
- Un mini-piano où chaque touche du clavier joue une note.

Intermédiaire : Si vous comprenez bien comment fonctionne Scratch

- Une course entre deux lutins contrôlés par le clavier.
- Un petit quizz à choix multiples.
- Une histoire interactive où l'utilisateur choisit la suite du récit.

Avancé : Pour les plus motivés !

- Un jeu de labyrinthe avec un chronomètre.
- Un simulateur de feu d'artifice avec sons et effets visuels.
- Un jeu de plateforme avec plusieurs niveaux.
- Un jeu de combat entre deux personnages avec points de vie.