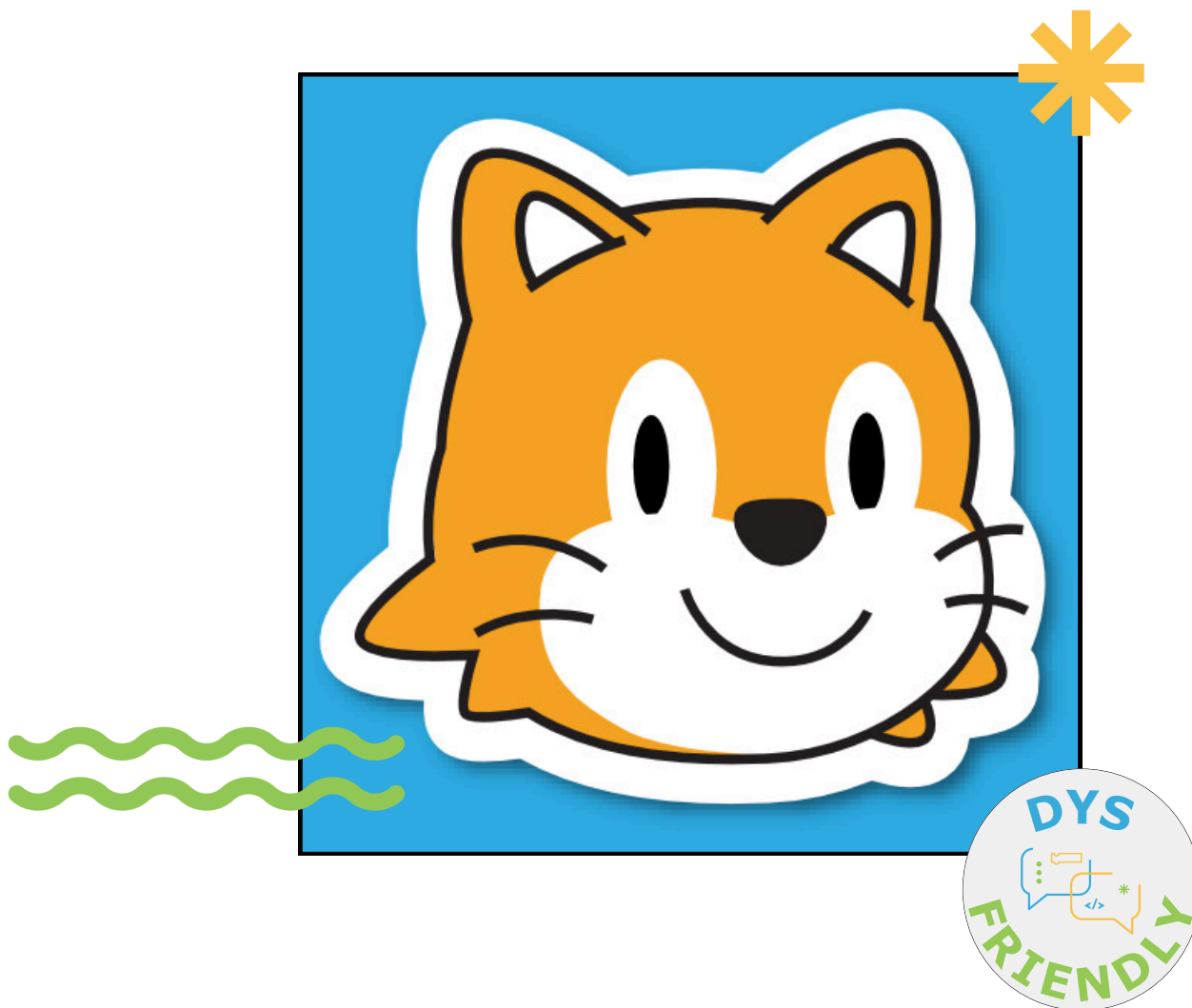




# Scratch Jr

Je m'initie à la programmation



## Carnet de l'enseignant



Cofinancé par  
l'Union européenne

# Sommaire



CodeNPlay : Un engagement pour l'accessibilité.....	p3
Pair Programming.....	p5
Les 4 concepts fondamentaux de l'informatique.....	p7
Les différents concepts de programmation.....	p.8
Découverte de l'environnement Scratch Jr.....	p.10
Description des différents bloc.....	p.11
Mes premiers pas sur Scratch Jr.....	p.13
En route vers l'aventure.....	p.15
Les fiches d'activité (fiches 6 à 11).....	p.17
Optimiser un programme en utilisant des boucles (fiche 12)...	p.19
Créer sa propre histoire (fiche 13 et 13').....	p.21

## CodeNPlay : Un engagement pour l'accessibilité

Chez Codenplay, nous avons à cœur de rendre nos jeux et contenus accessibles à tous les enfants, y compris ceux qui rencontrent des troubles de l'apprentissage.

Pour ce faire, nos dispositifs ont été relus par Noémie Brans, orthopédagogue et enseignante dans l'enseignement spécialisé et Tatiana Matmat, collaboratrice des Pôles Territoriaux.

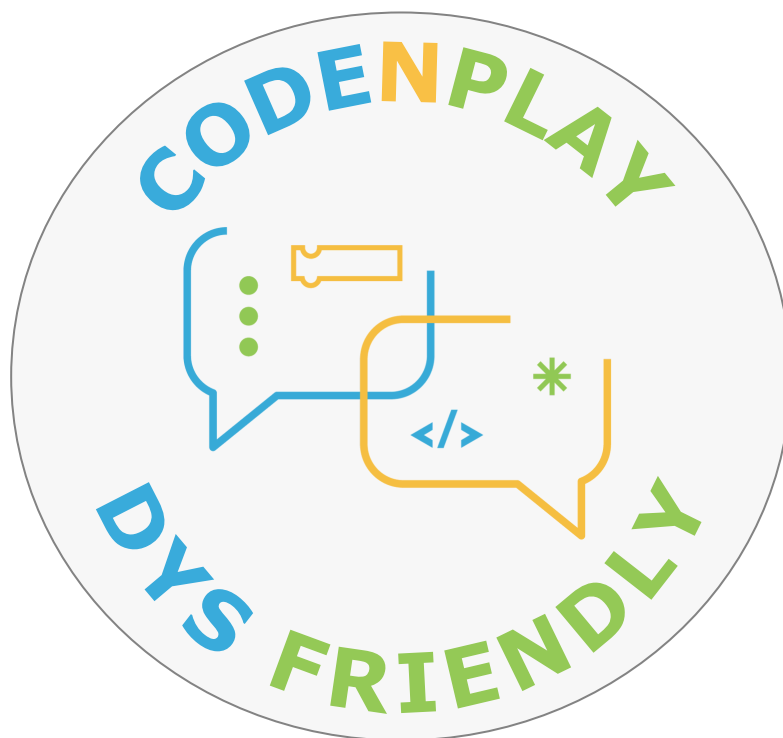
Concrètement, cela se traduit par :

- L'utilisation d'une police d'écriture claire et lisible, sans empattement et de taille suffisamment grande pour faciliter la lecture.
- Un interligne large pour aérer le texte et éviter la fatigue visuelle.
- La mise en gras des mots importants et des actions pour une meilleure compréhension.
- Simplifier le vocabulaire utilisé et à définir les mots difficiles.
- Structurer les textes de manière claire et concise, avec des paragraphes courts et des titres explicites.
- Proposer des contenus variés et stimulants, qui motivent les enfants et favorisent leur apprentissage.

Notre objectif est de permettre à tous les enfants de profiter pleinement de nos dispositifs d'apprentissage.

En plus de ces adaptations, nos contenus sont tous lisibles avec un logiciel de lecture vocale tel que Speechify.

Nous sommes convaincus que l'accessibilité est un enjeu majeur et nous nous engageons à poursuivre nos efforts en ce sens pour que tous les enfants puissent découvrir le numérique et la robotique avec Codenplay.



# Pair programming :

Le pair programming est une pratique dans laquelle deux programmeurs travaillent ensemble sur un même ordinateur pour concevoir, programmer et tester un logiciel. Voici quelques avantages que les enseignants peuvent tirer de cette pratique :

- Favoriser la collaboration et la communication : Le pair programming encourage les programmeurs à travailler ensemble, à partager des idées et à communiquer constamment. Cela peut aider à améliorer la qualité du code en permettant à chacun d'apporter sa propre expertise et son expérience.
- Améliorer la qualité du code : Le pair programming permet à deux programmeurs de travailler ensemble pour trouver des erreurs de code et des bugs plus rapidement. Ils peuvent également discuter des bonnes pratiques de programmation et prendre des décisions ensemble sur la façon d'aborder un problème complexe.
- Augmenter la productivité : Bien que cela puisse sembler contre-intuitif, le pair programming peut en fait augmenter la productivité des programmeurs. Travailler en équipe peut aider à résoudre les problèmes plus rapidement et à réduire le temps nécessaire pour tester et déboguer le code.

- Favoriser l'apprentissage : Le pair programming peut également être un outil pédagogique utile pour les enseignants. En travaillant avec un élève, les enseignants peuvent aider les élèves à comprendre les concepts de programmation et à développer leurs compétences en programmation.
- Encourager la résolution de problèmes : Le pair programming est une pratique qui encourage la résolution de problèmes en équipe. Les programmeurs doivent travailler ensemble pour trouver des solutions à des problèmes complexes et pour prendre des décisions importantes.

En résumé, le pair programming est une pratique utile pour les enseignants car elle favorise la collaboration, la communication, la qualité du code, la productivité et l'apprentissage. Les enseignants peuvent encourager les étudiants à pratiquer cette méthode en travaillant en binôme pour résoudre des problèmes de programmation ou pour travailler sur des projets de groupe.

# Les 4 concepts fondamentaux de l'informatique :

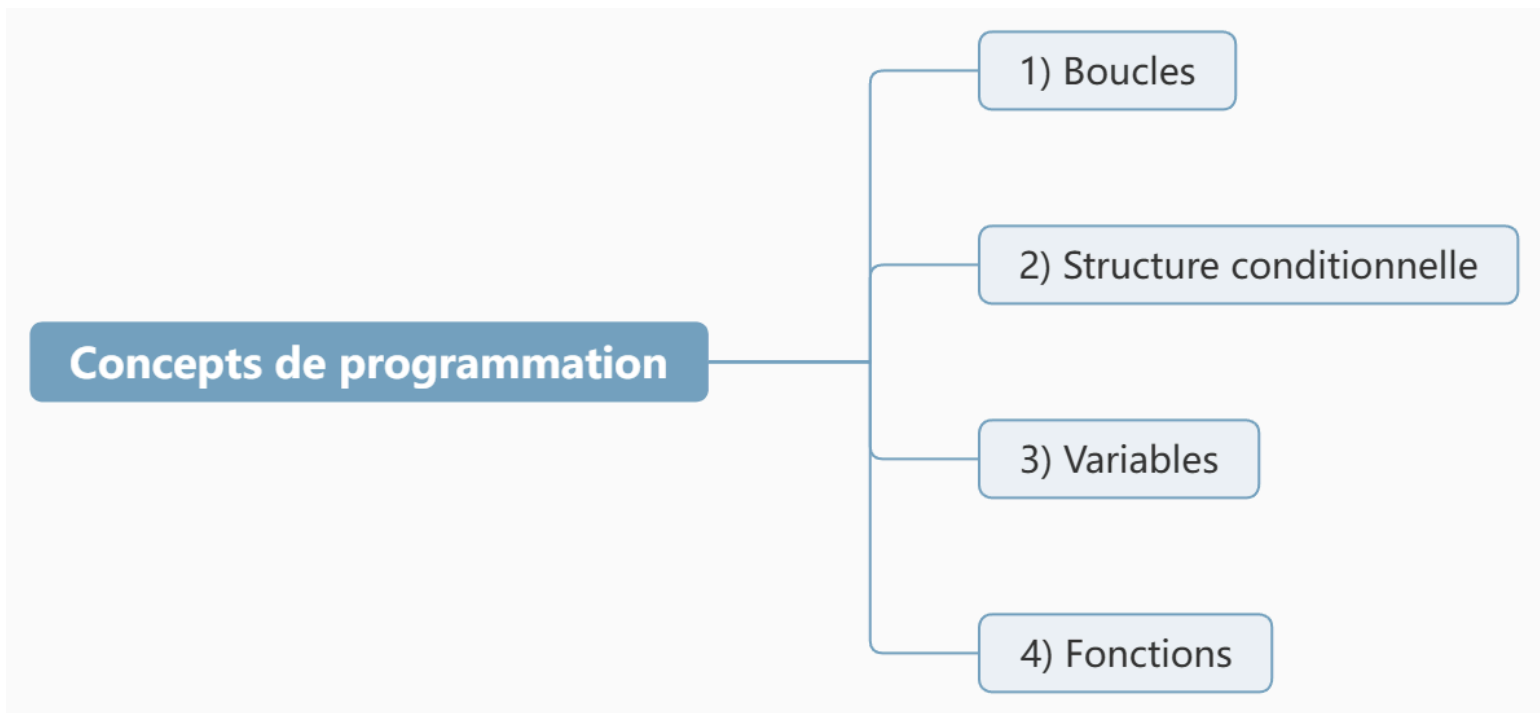
**Machines :** Les machines informatiques sont des dispositifs physiques qui peuvent stocker, traiter et communiquer des données numériques. Les ordinateurs modernes sont des exemples courants de machines informatiques, mais il existe également de nombreuses autres formes de machines informatiques, telles que les smartphones, les tablettes et les serveurs.

**Algorithmes :** Un algorithme est une recette qui permet de résoudre un problème de manière systématique (Dowez 2011). Les algorithmes sont utilisés dans de nombreuses applications informatiques, telles que le traitement des données, l'apprentissage automatique et l'analyse des systèmes.

**Langages :** Les langages informatiques sont des systèmes formels utilisés pour communiquer avec les machines informatiques. Il existe de nombreux langages informatiques différents, tels que C++, Python, les blocs Scratch Jr et Java, qui sont utilisés pour développer des applications informatiques de toutes sortes.

**Information :** L'information est le contenu numérique qui est stocké et traité par les machines informatiques. L'information peut prendre de nombreuses formes différentes, telles que du texte, des images, des vidéos et des données structurées. Les machines informatiques sont capables de stocker, traiter et communiquer des informations de manière automatique, ce qui en fait une ressource extrêmement importante pour de nombreuses entreprises et organisations.

# les différents concepts de programmation :



## 1) Boucles

Une boucle est une instruction qui répète une action plusieurs fois. Par exemple, pour compter jusqu'à 10. Dans la programmation, les boucles sont utilisées pour optimiser l'écriture d'un programme en n'écrivant qu'une seule fois la partie redondante.

## 2) La structure conditionnelle

C'est une instruction qui vérifie si une condition est vraie ou fausse. Par exemple, si vous voulez savoir si un nombre est pair, vous pouvez utiliser une condition qui vérifie si le nombre est divisible par 2. Dans la programmation, les conditions sont utilisées pour exécuter des instructions différentes en fonction de la situation.



# les différents concepts de programmation :

## **3) Variables**

Une variable est une case mémoire que l'on peut nommer et qui peut contenir une valeur. Par exemple, si vous voulez stocker votre âge, vous pouvez utiliser une variable appelée "âge" et y stocker la valeur 8. Dans la programmation, les variables sont utilisées pour stocker des données et les utiliser plus tard dans le programme.

## **4) Fonctions**

Une fonction est un morceau de programme qui porte généralement un nom et accomplit un traitement spécifique. Par exemple, pour dessiner un carré, vous pouvez créer une fonction qui dessine un côté, puis appeler cette fonction quatre fois pour dessiner les quatre côtés du carré. Dans la programmation, les fonctions sont utilisées pour décomposer le code et le rendre plus facile à comprendre et à gérer.

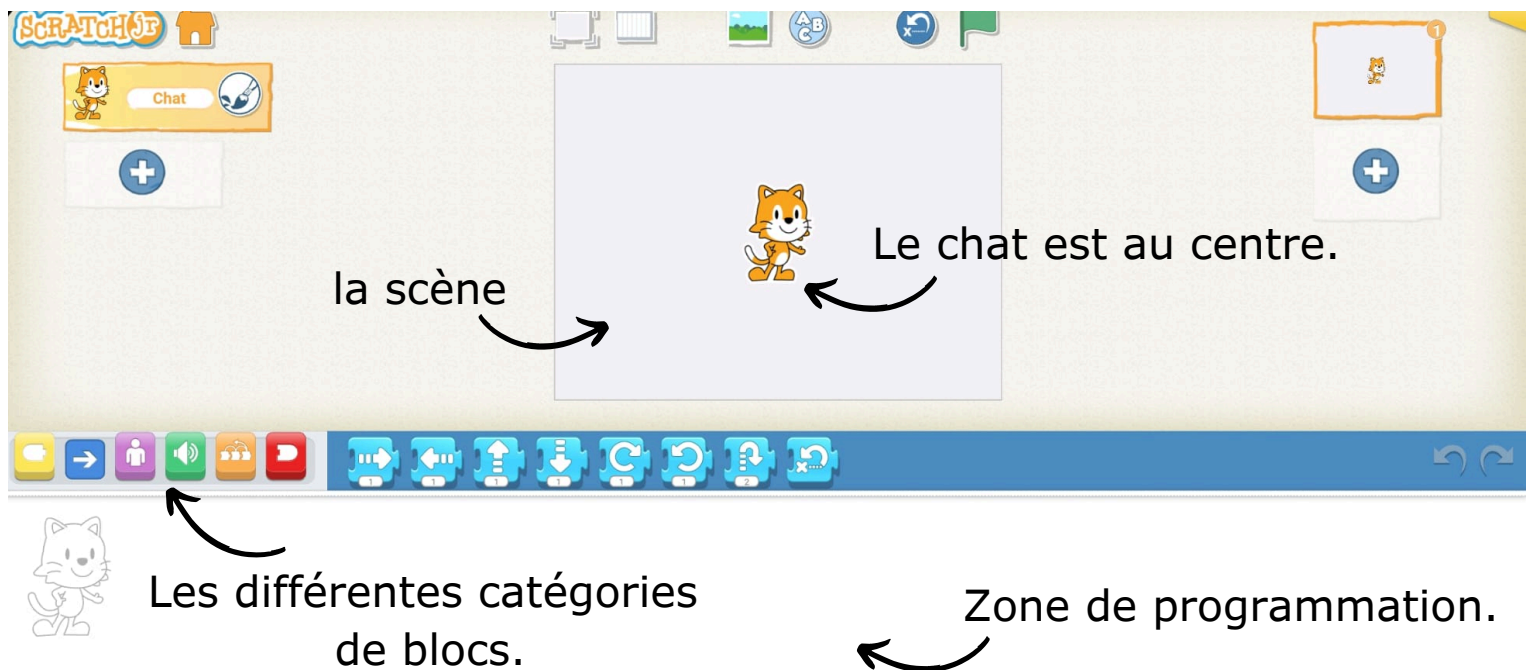
# DÉCOUVERTE DE L'ENVIRONNEMENT SCRATCH

- **Qu'est-ce que Scratch Jr ?**

Scratch Junior est un logiciel de programmation visuelle conçu pour les enfants âgés de 5 à 7 ans. Il permet aux enfants de créer des projets interactifs en glissant et en déposant des blocs de programmation colorés. Les enfants peuvent utiliser Scratch Junior pour créer des histoires animées, des jeux et des simulations, tout en développant leur pensée logique et leur créativité.

L'interface de Scratch Junior est conviviale, et permet aux enfants de voir instantanément le résultat de leur travail. Scratch Junior est une excellente ressource pour les enseignants qui souhaitent introduire la programmation aux jeunes enfants de manière ludique et accessible.

- **L'environnement scratch jr**



# DESCRIPTION DES DIFFÉRENTS BLOCS

## ● Les blocs d'événement



Démarre le script lorsqu'on touche le Drapeau vert.



Démarre le script lorsque vous touchez le personnage.



Démarre le script lorsque le personnage est touché par un autre personnage.



Démarre le script à chaque fois qu'un message de la couleur spécifiée est envoyé.



Envoie un message de la couleur spécifiée.

## ● Les blocs de mouvement



Déplace le personnage vers la droite du nombre de cases spécifié.



Fait tourner le personnage dans le sens des aiguilles d'une montre de la quantité spécifiée (tourner de 12 correspond à un tour complet)



Déplace le personnage vers la gauche du nombre de cases spécifié.



Fait tourner le personnage dans le sens inverse des aiguilles d'une montre de la quantité spécifiée (tourner de 12 correspond à un tour complet).



Déplace le personnage vers le haut du nombre de cases spécifié.



Fait sauter le personnage vers le haut du nombre de cases spécifié, puis redescendre à sa position de départ.



Déplace le personnage vers le bas du nombre de cases spécifié.



Ramène le personnage à son point de départ (pour définir une nouvelle position de départ, il faut faire glisser le personnage à la position voulue)

## ● Les blocs d'apparence



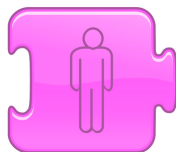
Fait apparaître un message dans une bulle au-dessus du personnage.



Remet le personnage à sa taille initiale.



Augmente la taille du personnage.



Fait disparaître progressivement le personnage jusqu'à ce qu'il soit invisible.



Diminue la taille du personnage.



Fait apparaître progressivement le personnage.

## ● Les blocs de son



Joue le son « pop »



Joue un son enregistré par l'utilisateur.

## ● Les blocs de contrôle



Met le script en pause pour un temps spécifié (en dixièmes de secondes).



Modifie la vitesse d'exécution de certains blocs.



Arrête tous les scripts du personnage.



Exécute le bloc à l'intérieur de la boucle le nombre de fois spécifié.

## ● Les blocs de contrôle



Répète le script encore et encore.



Indique la fin du script (mais n'intervient pas sur le déroulement du script).



Va à la page spécifiée du projet.

# MES PREMIERS PAS SUR SCRATCH JR

## Objectifs :

- Découvrir les différentes fonctionnalités de Scratch Jr.
- Importer un nouveau personnage.
- Choisir un décor.
- Utiliser les notions vues précédemment.

## Les compétences :

- Résoudre des problèmes et mobiliser sa pensée critique/analytique.
- Coder et décoder pour réaliser les déplacements d'un robot ou d'un personnage sur un écran.
- Ecrire un algorithme pour résoudre un problème simple.
- Lire, interpréter des représentations de l'espace et les confronter au réel.

## Les notions travaillées : "Machines" et "Langages"

- Donner des instructions à une machine en utilisant un langage spécial, appelé "langage de programmation".
- Un "algorithme" est une méthode permettant de résoudre un problème.
- Un programme est un algorithme exprimé dans un langage de programmation.

## Le matériel :

- Une tablette pour deux élèves
- Application Scratch junior installée
- Les fiches 1- 2 - 3

## Déroulement de l'activité

<b>Situation mobilisatrice :</b> Proposer aux élèves de créer leur première animation grâce à une application de programmation	
<b>Travailler en binôme</b>	<ul style="list-style-type: none"><li>• Travailler en binôme, expliquer aux élèves le rôle du pilote et du copilote (suivant le concept de "pair programming").</li><li>• Après 15 minutes, changer le rôle de chacun des élèves.</li></ul>
<b>Découverte de l'environnement</b>	<ul style="list-style-type: none"><li>• Montrer aux élèves des manipulations essentielles de l'application : le démarrage, le principe d'une séquence d'instruction.</li><li>• Découvrir collectivement la fiche 1</li><li>• Laisser quelques minutes d'exploration libre aux élèves.</li><li>• Proposer aux élèves la fiche 2 : Ecrire un programme</li><li>• Lire collectivement la fiche et laisser les élèves travailler à leur rythme pour créer leur premier programme.</li><li>• Vérifier que chaque groupe a réalisé et testé le programme.</li></ul>
<b>Les déplacements sur quadrillage</b>	<ul style="list-style-type: none"><li>• Faire apparaître le quadrillage et montrer aux élèves que lorsqu'on déplace le sprite une case bleue se déplace avec lui.</li><li>• Distribuer la fiche 3 et laisser les élèves créer un programme.</li></ul>
<b>Réalisation des premiers défis</b>	<ul style="list-style-type: none"><li>• Distribuer la fiche 4</li><li>• Réaliser un défi à la fois : lire la consigne, laisser les élèves travailler en binôme, vérifier que chaque groupe a bien réalisé le défi.</li><li>• Faire la même démarche pour tous les défis.</li></ul>

# EN ROUTE VERS L'AVENTURE ...

## **Objectifs :**

- Créer une histoire en suivant des étapes.
- Découvrir comment changer la vitesse d'un sprite.
- Découvrir les différentes fonctionnalités de Scratch Jr

## **Les compétences :**

- Coder et décoder pour réaliser les déplacements d'un robot ou d'un personnage sur un écran.
- Concevoir un algorithme pour résoudre un problème simple.
- Lire, interpréter des représentations de l'espace et les confronter au réel.

## **Les notions travaillées : "Machines" et "Langages"**

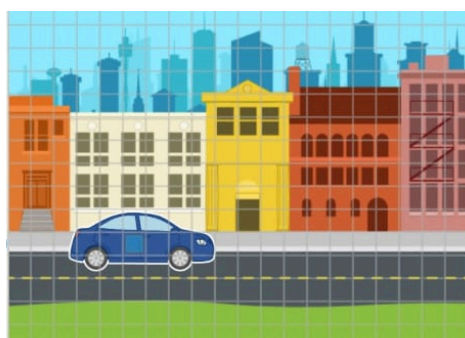
- Donner des instructions à une machine en utilisant un langage spécial, appelé "langage de programmation".
- Un "algorithme" est une méthode permettant de résoudre un problème.
- Un programme est un algorithme exprimé dans un langage de programmation.

## **Le matériel :**

- Une tablette pour deux élèves
- Application Scratch junior installée
- La fiche 5

# Déroulement de l'activité

<b>Situation mobilisatrice :</b> Proposer aux élèves de créer une petite histoire sur Scratch Jr : Créer une route sur laquelle des véhicules circulent.	
<b>Travailler en binôme</b>	<ul style="list-style-type: none"><li>• Composer des groupes de 2 élèves.</li><li>• Rappeler aux élèves le rôle du pilote et du copilote (suivant le concept de "pair programming").</li></ul>
<b>Placer la voiture sur la route</b>	<ul style="list-style-type: none"><li>• Chaque groupe allume la tablette et lance l'application Scratch Jr.</li><li>• Projeter le recto de la fiche 5 et lire une étape à la fois. Marquer un temps de pause pour laisser le temps à chaque groupe d'effectuer les consignes données.</li><li>• Vérifier que chaque groupe a bien réalisé les étapes décrites.</li></ul>
<b>Réaliser le programme de la voiture</b>	<ul style="list-style-type: none"><li>• Inviter les élèves à découvrir les différentes étapes (pour les élèves non lecteurs : lire chaque étape; pour les élèves lecteurs : laisser les élèves lire les différentes étapes.</li><li>• Chaque groupe programme les déplacements de la voiture et teste le programme obtenu.</li><li>• Vérifier que chaque groupe a bien réalisé le défi. (aide aux enfants en difficulté).</li></ul>
<b>Dépassement</b>	<ul style="list-style-type: none"><li>• Ajouter d'autres véhicules sur la route.</li></ul>





# LES FICHES D'ACTIVITÉ

Chacune des activités permet d'apprendre rapidement de nouvelles notions sur ScratchJr. Les fiches sont répertoriées du plus simple au plus difficile, mais il est possible de les réaliser dans l'ordre de son choix.

## **Objectifs :**

- Créer une histoire en suivant des étapes.
- Découvrir différentes fonctionnalités de Scratch Junior

## **Les compétences :**

- Résoudre des problèmes et mobiliser sa pensée critique/analytique.
- Coder et décoder pour réaliser les déplacements d'un robot ou d'un personnage sur un écran.
- Concevoir un algorithme pour résoudre une problème simple.
- Lire, interpréter des représentations de l'espace et les confronter au réel.

## **Les notions travaillées : "Machines" et "Langages"**

- Donner des instructions à une machine en utilisant un langage spécial, appelé "langage de programmation".
- Un "algorithme" est une méthode permettant de résoudre un problème.
- Un programme est un algorithme exprimé dans un langage de programmation.

## **Le matériel :**

- Une tablette pour deux élèves
- Application Scratch junior installée
- Les fiches de 6 à 11

# Déroulement de l'activité

<b>Situation mobilisatrice : Proposer aux élèves de créer des petites histoires l'aide de fiches</b>	
<b>Travailler en binôme</b>	<ul style="list-style-type: none"><li>• Composer des groupes de 2 élèves.</li><li>• Rappeler aux élèves le rôle du pilote et du copilote (suivant le concept de "pair programming").</li></ul>
<b>Découverte de la fiche</b>	<ul style="list-style-type: none"><li>• Chaque groupe allume la tablette et lance l'application Scratch Jr.</li><li>• Proposer une fiche d'activité : mettre en évidence la notion vue de la fiche choisie.</li><li>• Vérifier la compréhension de la fiche en faisant oraliser les démarches par les élèves.</li></ul>
<b>Au travail</b>	<ul style="list-style-type: none"><li>• Chaque groupe d'élèves réalise l'activité donnée.</li><li>• Vérifier la réalisation des élèves. (aide aux enfants en difficulté)</li></ul>
<b>Dépassement</b>	<ul style="list-style-type: none"><li>• Il y a des piste de dépassement sur certaines fiches.</li></ul>
<b>Fiche 6 : Faire la course</b>	Utiliser le bloc de vitesse pour accélérer ou ralentir un personnage.
<b>Fiche 7 : Une forêt effrayante</b>	Ajouter plusieurs sprites et programmer leur propre script.
<b>Fiche 8 : Jouer au basketball</b>	Apprendre à utiliser le bloc de répétition pour dribbler avec un ballon de basket.
<b>Fiche 9 : Alors on danse</b>	Utiliser des blocs de son et de mouvement et un bloc de redémarrage pour faire danser les personnages.
<b>Fiche 10 : Une conversation</b>	Envoyer plusieurs messages entre les personnages en utilisant des enveloppes de couleurs différentes.
<b>Fiche 11 : Sur la lune</b>	Faire disparaître un sprite, utiliser le bloc d'attente, démarrer un programme en cliquant sur un sprite.

# OPTIMISER UN PROGRAMME EN UTILISANT DES BOUCLES

## Objectifs :

- Remplacer les instructions répétées par une boucle.

## Les compétences :

- Résoudre des problèmes et mobiliser sa pensée critique/analytique.
- Coder et décoder pour réaliser les déplacements d'un robot ou d'un personnage sur un écran.
- Concevoir un algorithme pour résoudre un problème simple.
- Lire, interpréter des représentations de l'espace et les confronter au réel.

## Les notions travaillées : "Algorithme"

- Une boucle permet de répéter plusieurs fois la même instruction ou la même séquence d'instructions.


## Déroulement de l'activité

### Situation mobilisatrice :

Montrer aux élèves le programme suivant :



### Observation

- Demander aux élèves d'observer attentivement, de repérer les instructions qui se répètent afin d'utiliser moins de blocs.
- Montrer ce bloc d'instruction orange .
- Les élèves expérimentent pour comprendre la signification de ce bloc. Ils tentent d'optimiser le programme qui contrôle le déplacement du sprite.

<b>Recherche</b>	<ul style="list-style-type: none"> <li>• Proposer deux missions aux élèves :</li> <li>• Mission 1 : Faire déplacer un sprite en escalier en utilisant la boucle. (monter 7 marches)</li> <li>• Mission 2 : Faire faire le tour de la scène au sprite (3 fois de suite)</li> </ul>
<b>Mise en commun</b>	<ul style="list-style-type: none"> <li>• Réaliser à la correction collective : les groupes d'élèves proposent leur solution et justifient leur choix. Bien insister sur l'utilisation du vocabulaire adéquat.</li> <li>• Indiquer que l'on peut introduire une boucle chaque fois que des instructions se répètent.</li> </ul>
<b>Exercices</b>	<ul style="list-style-type: none"> <li>• Projeter au tableau ce programme :</li> </ul> <div data-bbox="596 786 1286 904" data-label="Image"> <p>The image shows a Scratch script starting with a 'when green flag clicked' event block. This is followed by a 'repeat' loop block set to 2 iterations. Inside the loop are the following movement blocks in sequence: 'move up 2 steps', 'move right 4 steps', 'move up 3 steps', 'move left 2 steps', 'move right 2 steps', and 'move left 1 step'.</p> </div> <ul style="list-style-type: none"> <li>• Demander aux élèves de colorier le déplacement du sprite sur la grille. (v. fiche 12)</li> <li>• Les élèves vérifient le programme sur leur tablette.</li> </ul>

# CRÉER SA PROPRE HISTOIRE

## Objectifs :

- Créer sa propre histoire.
- Réinvestir les notions rencontrées lors des séquences précédentes.

## Déroulement de l'activité

### Situation mobilisatrice :

Annoncer aux élèves qu'ils vont pouvoir créer leur propre histoire.  
Conseiller aux élèves de ne pas créer un récit trop compliqué.

<b>Travailler en binôme</b>	<ul style="list-style-type: none"><li>• Composer des groupes de 2 élèves.</li><li>• Rappeler aux élèves le rôle du pilote et du copilote (suivant le concept de "pair programming").</li></ul>
<b>Découverte de la fiche</b>	<ul style="list-style-type: none"><li>• Chaque groupe allume la tablette et lance l'application Scratch Jr.</li><li>• Distribuer la fiche avec les consignes (fiche 13).</li><li>• Lire collectivement les consignes afin de vérifier la bonne compréhension des consignes.</li><li>• Bien insister sur le fait que les élèves devront intégrer dans leur programme : des événements pour coordonner leurs sprites, employer des boucles, au moins un bloc de contrôle et au moins un bloc apparence.</li><li>• Montrer la grille pour vérifier le travail réalisé.</li></ul>
<b>Conception</b>	<ul style="list-style-type: none"><li>• Les élèves discutent des éléments qu'ils veulent intégrer dans leur projet. S'ils éprouvent des difficultés à trouver une idée, proposer des exemples. (voir fiche 13')</li><li>• Vérifier si le projet proposé par les élèves est cohérent.</li><li>• Donner des pistes d'amélioration pour faciliter la création du projet.</li></ul>
<b>Présentation</b>	<ul style="list-style-type: none"><li>• Chaque groupe présente son projet à la classe.</li></ul>



www.codenplay.be  
Cantersteen 12  
1000 Bruxelles  
contact@codenplay.be

Scratch Jr, je m'initie à la programmation  
2023

Autrice : Trichili Sonia

Relecture :

Marielle Léonard, Ingénieure d'étude en didactique de  
l'informatique (Université de Lille)

Julie Henry et Fanny Boraita (Digifactory ASBL)