

# Informatique 6e

## Fiches MITIC

INSTITUT  
*FLORIMONT*

01000001

0111010101110100011001010  
111010101110010011100110010000000111010001000000100001001  
100101011011100110111101101001011101000010000001001110011  
000010110010001100001011101010110010000100000011001010111  
010000100000010100110110100101101101011011110110111000100  
000010101100110010101110010011001000110000101101110001011  
100010000001001001011011100111001101110100011010010111010  
001110101011101000010000001000110011011000110111101110010  
011010010110110101101111011011100111010000101110000000000

Informatique – 6<sup>e</sup> MITIC

Institut Florimont

Petit-Lancy (Suisse)

© Tout droit réservé. Crédit photographie couverture : Institut Florimont. Illustration des premières pages de chapitre issue de *Codex Leicester* de Leonardo da Vinci (domaine public).

2<sup>e</sup>me édition, v2.0

juin 2021

I N S T I T U T *fl*ORIMONT

Informatique  
6<sup>e</sup> MITIC  
Institut Florimont

Ce livret appartient à .....



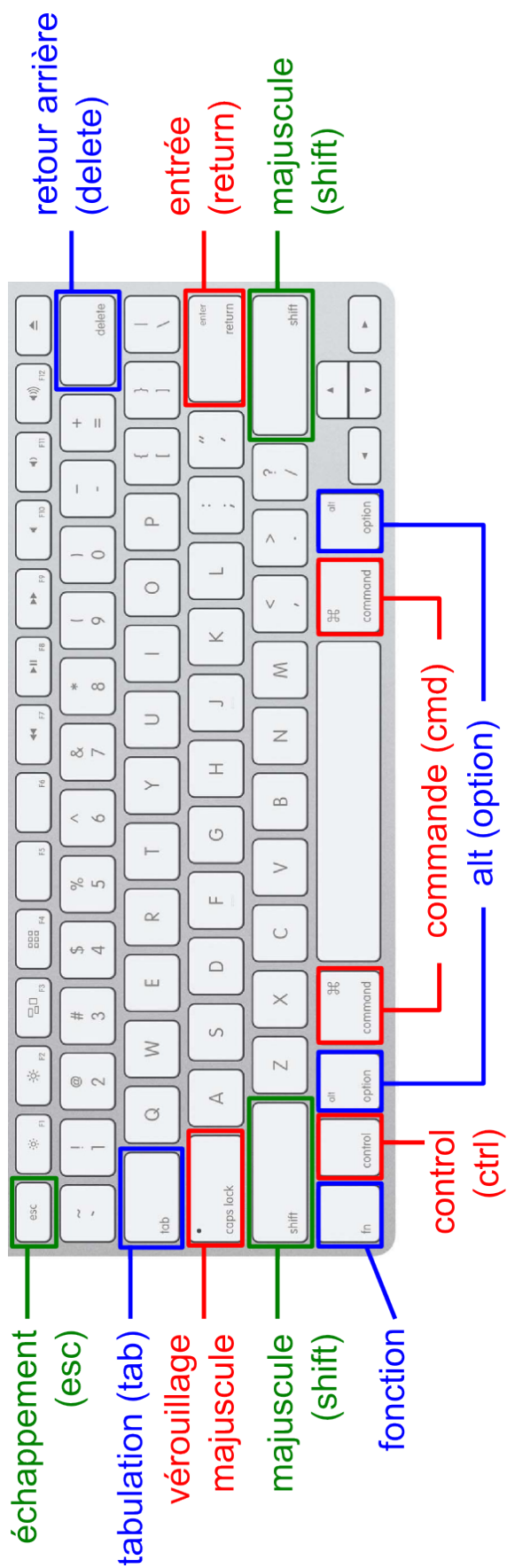
# Table des matières

<b>1</b>	<b>Programmation Scratch.....</b>	<b>1</b>
1	Séance 1 : un premier programme	2
1.1	Premiers pas avec Scratch...	2
1.2	Pour préparer l'activité...	3
1.3	Pour bien démarrer...	3
1.4	Sujet de l'activité...	3
1.5	Pour aller plus loin...	4
2	Séance 2 : dessiner avec Scratch	5
2.1	Pour bien démarrer...	5
2.2	Sujet de l'activité...	5
2.3	Pour aller plus loin...	6
3	Séance 3 : créer un petit jeu en Scratch	7
3.1	Pour bien démarrer...	7
3.2	Sujet de l'activité...	7
3.3	Pour aller plus loin...	8
4	Aide pour réaliser les activités	9
4.1	Aide pour la séance 1...	9
4.2	Aide pour la séance 2...	12
4.3	Aide pour la séance 3...	13





## Les touches spéciales du clavier



Pour sauvegarder son travail : cmd + S

Pour annuler la dernière opération : cmd + Z



# Philosophie du document

Vous avez entre les mains le premier exemplaire d'une série de quatre fascicules qui accompagneront les élèves des classes de 6<sup>e</sup>, 5<sup>e</sup>, 4<sup>e</sup> et 3<sup>e</sup> dans leur découverte et maîtrise de l'outil informatique.

Ce document se présente sous la forme d'un fascicule qui rassemble des fiches MITIC<sup>1</sup> permettant aux élèves d'apprendre à utiliser les logiciels et espaces numériques mis à leur disposition. Pour l'année de 6<sup>e</sup>, sont traités les logiciels *Microsoft Word* (traitement de texte), *Microsoft Excel* (tableur grapheur), *Gimp* (retouche d'image), *Scratch* (programmation) ainsi que l'outil *Microsoft Teams* présent sur l'espace numérique de travail (ENT) de notre école.

Chaque fiche est conçue pour être exploitée à trois occasions et dans trois matières différentes, à chaque fois lors d'une séance de 45 minutes. La fiche sur le tableur, par exemple, est découverte en mathématiques (*Séance 1*), exploitée à nouveau en physique-chimie (*Séance 2*) puis en histoire-géographie (*Séance 3*) selon un calendrier proposé en début de fiche. Nous avons à chaque fois essayé de faire coïncider les notions abordées dans la fiche avec le programme de la matière concernée. Remarque : les séances 2 et 3 peuvent être inversées si nécessaire, puisqu'elles reprennent les notions découvertes dans la première séance de la fiche.

Au début de l'année, chaque titulaire de 6<sup>e</sup> doit emmener les élèves dont il a la charge en salle informatique et leur faire découvrir la plateforme *Teams* (fiche page ??). Au cours de l'année, les professeurs de chaque matière concernée par une fiche sont responsables de sa réalisation avec les élèves.

Professeurs, c'est à vous que revient la tâche délicate d'inclure le contenu de ces fiches dans votre progression. À vous de le faire vivre : arriver en salle informatique et demander aux élèves de remettre en forme un texte de Jonathan Swift ne présente que peu d'intérêt pédagogique. Donnez du sens à ces fiches et profitez-en pour diversifier votre enseignement. N'hésitez pas à exploiter dans vos cours les techniques présentées dans ce fascicule afin que les élèves utilisent plusieurs fois leurs nouvelles compétences et, par là-même, les pérennisent.

À la fin de ce fascicule sont proposées des idées d'activités supplémentaires sous forme de

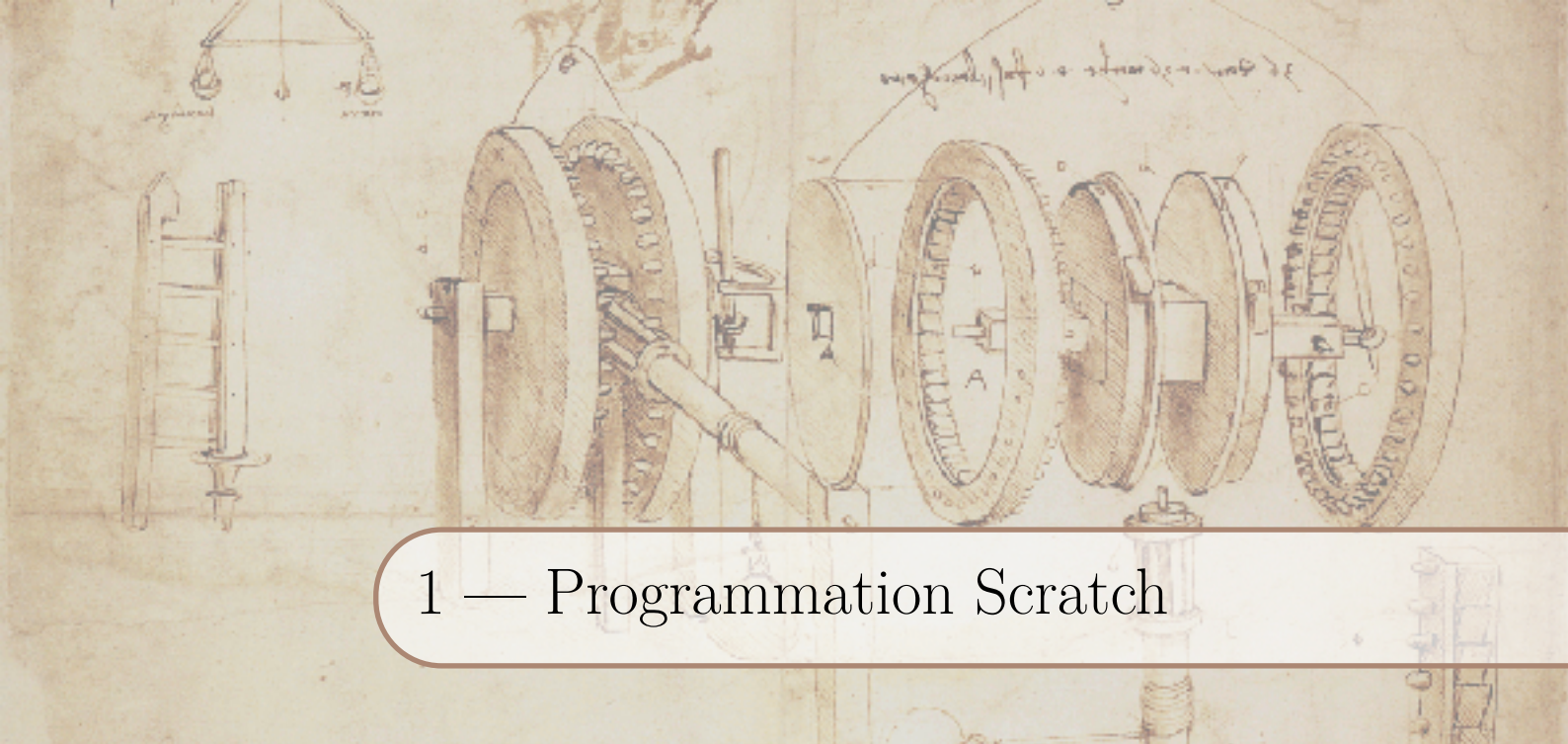
---

1. MITIC : Médias, Images et Technologies de l'Information et de la Communication.

projet : ainsi les élèves exploiteront-ils les connaissances acquises au cours de l'année. Ces activités pluridisciplinaires permettent aussi de faire comprendre aux élèves que les cours qu'ils suivent ne sont pas des entités cloisonnées, mais, qu'à l'inverse, les compétences qu'ils y développent sont transposables d'une matière à une autre. Dès lors, ils comprendront que ce qu'ils étudient en classe fait partie d'un socle de connaissances transdisciplinaires utiles et nécessaires à leur avenir.

Merci d'avance à tous pour votre implication.

L'équipe de rédaction.



# 1 — Programmation Scratch

Les ordinateurs sont des machines qui exécutent des programmes. On peut écrire des programmes dans différents *langages de programmation*, par exemple *Python*, *C++*, *Java*... ou encore *Scratch*.

*Scratch* est un langage de programmation **visuelle** (on place des blocs d'instructions pour créer des programmes composés de codes) et **évènementielle** (le programme réagit à des événements comme le clic de souris ou l'appui sur une touche). Il contient des **objets** : le lutin est un objet, l'arrière plan de la scène est un autre objet. On peut modifier les propriétés des objets, leur associer des codes, des costumes ou des sons.

- Logiciel<sup>1</sup> : *Scratch 3.0*
- Prérequis : aucun
- Matière concernée : mathématiques
- Objectifs : réaliser un programme simple en *Scratch* (programme rendu sur *Teams*).
- Compétences :
  - choisir et paramétrer l'objet sprite ;
  - choisir et paramétrer l'objet scène ;
  - créer/insérer un nouvel objet ;
  - écrire un code comprenant mouvements, réponses à événement, boucles et son ;
  - associer un code à un objet ;
  - écrire un programme simple qui réponde à une problématique donnée.

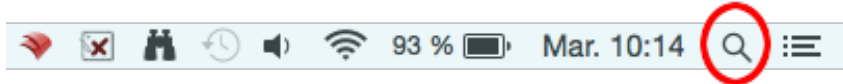
---

1. Le logiciel Scratch est librement téléchargeable : <https://scratch.mit.edu/>

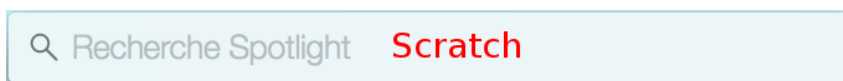
# 1 Séance 1 : un premier programme

## 1.1 Premiers pas avec Scratch...

Lancer le logiciel en utilisant la « loupe » :

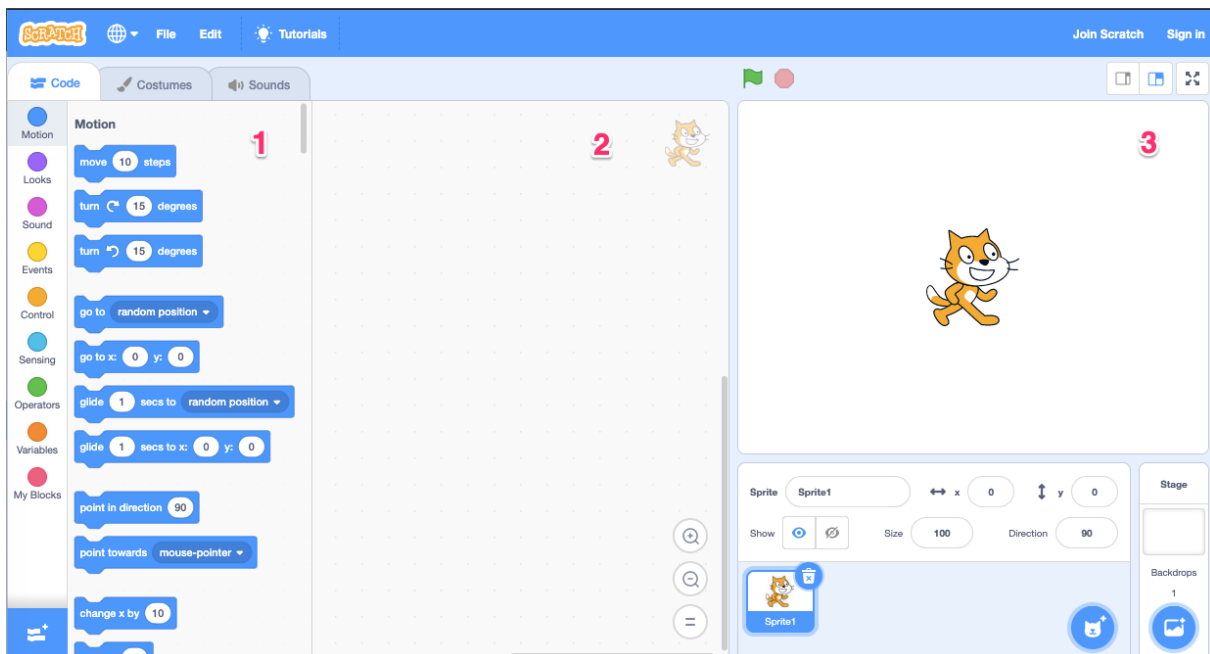


... puis en indiquant *Scratch* :




La fenêtre principale du logiciel s'ouvre. Elle se présente sous la forme de trois colonnes :

- la première colonne ① contient les différents blocs d'instructions que l'on utilise pour écrire les programmes ;
- la seconde colonne ② est la zone dans laquelle sont construits les codes qui composent le programme ;
- la troisième colonne ③ contient deux zones :
  - la zone où le programme s'exécute,
  - la zone où on peut sélectionner les différents objets présents dans le programme.



Si l'application **Scratch** n'est pas installée sur votre ordinateur, la version en ligne vous sera proposée. En cliquant dessus, vous pourrez alors ouvrir une fenêtre similaire à celle présentée ci-dessus.

Vous pouvez dès à présent choisir la langue de l'interface en cliquant sur l'icône en haut à gauche . Nous choisirons par exemple Français



## 1.2 Pour préparer l'activité...

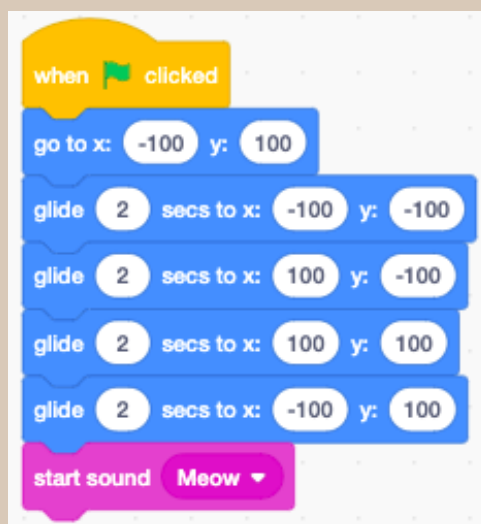
QR-code ici avec url

## 1.3 Pour bien démarrer...

Dès que vous avez ouvert un nouveau document dans *Scratch*, sauvegardez-le au format Nom-seance1.sb3 : dans le menu Fichier, choisir Sauvegarder sur votre ordinateur. Pendant que vous travaillez, pensez à sauvegarder régulièrement votre travail.

## 1.4 Sujet de l'activité...

*Voici un premier programme : avant de l'écrire, essayez de deviner ce qu'il se passe lorsque le drapeau vert est pressé ! Le programme est volontairement écrit en anglais. A vous de retrouver les blocs correspondant en français. Recopiez ensuite le programme en cherchant à comprendre le rôle de chaque bloc de code.*



*Une fois votre programme terminé, vous devrez l'enregistrer au format .sb3 (le fichier doit être nommé à partir de votre nom : Nom-seance1.sb3) et le rendre sur Teams dans le dossier de remise de devoir, à l'endroit indiqué par votre professeur. Si nécessaire, se reporter à la fiche méthode Remettre son devoir, page ??.*

Pour obtenir de l'aide, rendez-vous à la page 9

## 1.5 Pour aller plus loin...

Rendez-vous dans la rubrique **Costumes** à côté des rubriques **Code** et **Sons**, et modifiez le costume de votre sprite à votre convenance, comme par exemple :



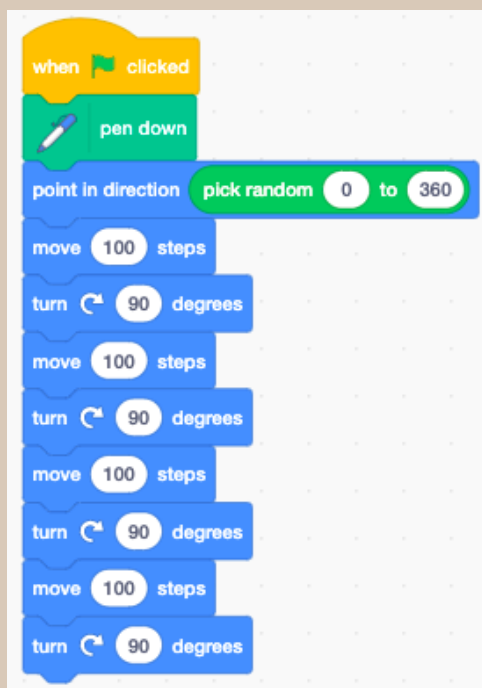
## 2 Séance 2 : dessiner avec Scratch

### 2.1 Pour bien démarrer...

Dès que vous avez ouvert un nouveau document dans *Scratch*, sauvegardez-le au format Nom-seance2.sb3 : dans le menu Fichier, choisir **Sauvegarder sur votre ordinateur**. Pendant que vous travaillez, pensez à sauvegarder régulièrement votre travail.

### 2.2 Sujet de l'activité...

Lire le code suivant et essayer de deviner ce qu'il va se passer lorsque le programme est lancé. Construire ensuite le code en essayant de comprendre le rôle de chaque bloc de code. Le programme est volontairement écrit en anglais. A vous de retrouver les blocs correspondant en français.

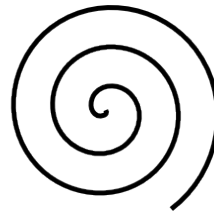
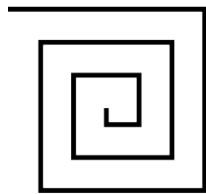


Une fois votre programme terminé, vous devrez l'enregistrer au format .sb3 (le fichier doit être nommé à partir de votre nom : **Nom-seance2.sb3**) et le rendre sur Teams dans le dossier de remise de devoir, à l'endroit indiqué par votre professeur.

Pour obtenir de l'aide, rendez-vous à la page 12

### 2.3 Pour aller plus loin...

S'il vous reste du temps, essayez de dessiner une spirale à angles droits, puis une véritable spirale.





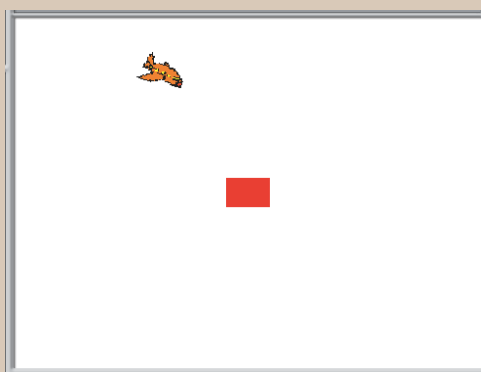
## 3 Séance 3 : créer un petit jeu en Scratch

### 3.1 Pour bien démarrer...

Dès que vous avez ouvert un nouveau document dans *Scratch*, sauvegardez-le au format *Nom-seance3.sb3* : dans le menu **Fichier**, choisir **Sauvegarder sur votre ordinateur**. Pendant que vous travaillez, pensez à sauvegarder régulièrement votre travail.

### 3.2 Sujet de l'activité...

*L'objectif de cette activité est de créer un petit jeu en Scratch. Le but est simple : piloter un avion tout en évitant des obstacles. Voici à quoi ressemble ce jeu :*



*Pour programmer ce jeu, vous devrez procéder par étapes :*

- 1. ajouter un nouvel objet : l'avion ;*
- 2. faire avancer l'avion dans la scène de manière continue, et rebondir lorsqu'il touche les bords ;*
- 3. ajouter la gestion des touches du clavier pour piloter l'avion ;*
- 4. ajouter un nouvel objet : l'obstacle ;*
- 5. ajouter la gestion de la collision entre l'avion et l'obstacle.*

*Attention, ce programme n'est pas si facile. Il va falloir réfléchir un peu avant de construire le code.*

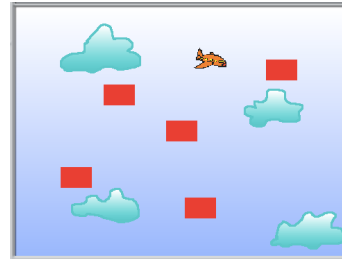
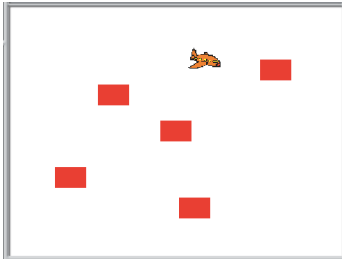
*Une fois votre programme terminé, vous devrez l'enregistrer au format `.sb3` (le fichier doit être nommé à partir de votre nom : `Nom-seance3.sb3`) et le rendre sur Teams dans le dossier de remise de devoir, à l'endroit indiqué par votre professeur.*

**Pour obtenir de l'aide, rendez-vous à la page 13**

### 3.3 Pour aller plus loin...

Si vous avez du temps, améliorez votre jeu. Vous pouvez par exemple :

- ajouter une action aux flèches *haut* et *bas* (par exemple respectivement avancer de 5 pas et avancer de  $-5$  pas) ;
- ajouter d'autres obstacles à éviter (voir image à gauche ci-dessous) ;
- modifier la scène pour qu'elle représente un ciel (voir image à droite ci-dessous).



## 4 Aide pour réaliser les activités

### 4.1 Aide pour la séance 1...

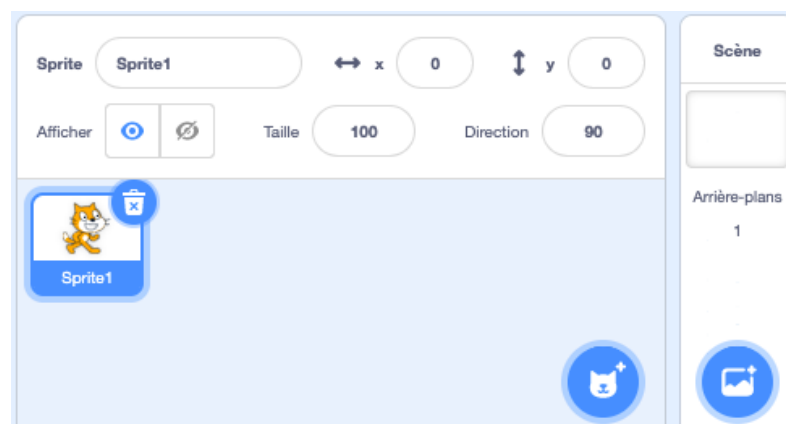
Nous allons écrire le programme étape par étape.

#### Modifier la scène où se passe l'action

La scène correspond à l'arrière-plan (blanc au départ) où se passe l'action. La scène est un objet qui peut être modifié. Pour cela, la première étape est de cliquer sur l'icône scène



en bas à droite.

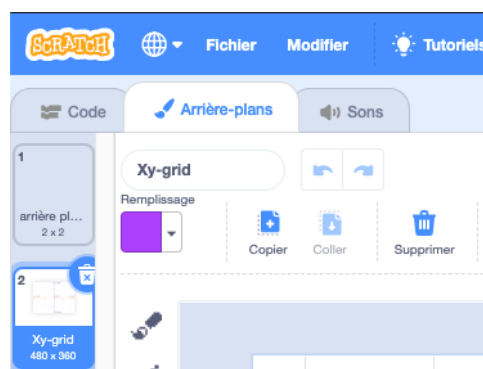


Une fois la scène sélectionnée (elle est alors entourée en couleur), suivre les 4 étapes suivantes :

1. Cliquer sur l'onglet Choisir un arrière-plan.
2. Chercher l'arrière-plan Xy-grid.
3. Cliquer sur l'arrière-plan Xy-grid pour remplacer le fond blanc qui était installé par défaut.

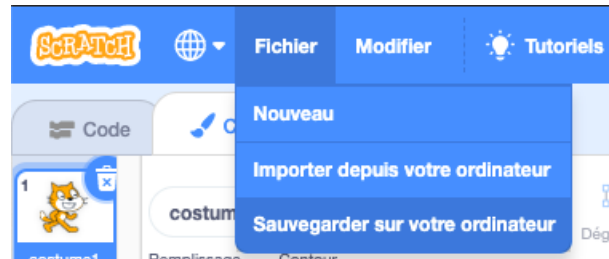
A noter qu'il est également possible d'importer un arrière-plan à partir d'images présentes sur votre ordinateur.

Notre programme comporte maintenant deux scènes différentes : arrière-plan1 et Xy-grid. C'est cette dernière qui est sélectionnée (elle est entourée en couleur).



## Enregistrer le programme

Pour enregistrer votre programme, cliquer sur **Sauvegarder sur votre ordinateur** :












Le fichier s'enregistre alors par défaut au format **.SB3** dans le dossier **Téléchargements** de votre ordinateur.

Comme toujours en informatique, il ne faut pas oublier d'enregistrer régulièrement le travail. Pour cela, cliquer régulièrement sur **Sauvegarder sur votre ordinateur** . :


## Ajouter un code associé au sprite


Le sprite est un autre objet. C'est lui qui réalise l'action principale du programme. On va lui associer un programme (nommé **code**) qui contient une succession d'ordres (les **instructions**) qu'il devra réaliser.


Pour construire ce premier code, nous allons suivre les différentes étapes indiquées ci-dessous :


1. Sélectionner le sprite  dans la zone des objets (en bas à droite)  : nous allons créer un code associé à ce sprite.
2. Choisir les blocs de contrôle en cliquant sur  (colonne de gauche).
3. Tirer le bloc  vers la zone de programmation (colonne du milieu).
4. Choisir les blocs de mouvement en cliquant sur  (colonne de gauche).
5. Tirer le bloc  vers la zone de programmation et l'accrocher sous le bloc .
6. Tirer ensuite le bloc  vers la zone de programmation et l'accrocher sous le bloc .
7. Régler les options du bloc en cliquant dans les zones de saisie et en écrivant la valeur de durée et les coordonnées  $x$  et  $y$  indiquées dans le programme ci-dessus.

8. Ajouter les trois autres blocs  et régler leurs options comme indiqué plus haut.

9. Choisir les blocs de sons en cliquant sur  (colonne de gauche).

10. Tirer le bloc  vers la zone de programmation, l'accrocher aux blocs précédents et régler ses options comme indiqué plus haut.

Après avoir terminé et vérifié le code, lancer le programme en appuyant sur le drapeau vert  en haut à droite. Aviez-vous deviné correctement ce qu'il allait se passer ?


Pour arrêter l'exécution du programme avant sa fin, appuyer sur le panneau stop  en haut.

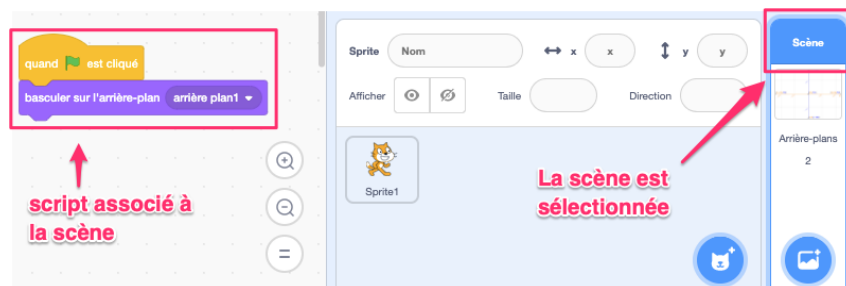
Pour que le programme s'exécute en plein écran, cliquer sur .

Pour quitter le mode plein écran, cliquer sur .

### Ajouter un code associé à la scène

Nous allons maintenant ajouter un deuxième code à notre programme : ce code va permettre de modifier la scène lorsque le drapeau vert est pressé.

Pour cela, la première étape est de cliquer sur l'icône scène  en bas à droite.



Une fois la scène sélectionnée (elle est alors entourée en couleur), créer le code suivant :



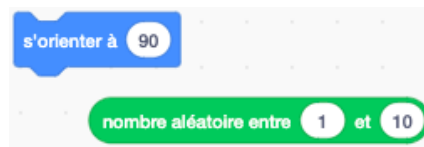
Une fois le code écrit et vérifié, lancer le programme en appuyant sur le drapeau vert  en haut.

## 4.2 Aide pour la séance 2...

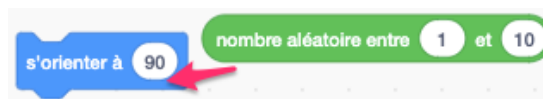
Construire le code : puisqu'il est associé à l'objet sprite, vérifier qu'il est bien sélectionné avant de le construire (voir si nécessaire le paragraphe 4.1 page 10 pour sélectionner le lutin

avant de construire le programme). Pour construire le bloc il faut procéder en deux temps :


1. Positionner les deux blocs d'instructions **s'orienter à...** et **nombre aléatoire entre...** dans la zone de programme ;



2. Tirer le bloc **nombre aléatoire entre...** dans la zone de saisi du bloc **s'orienter à...**

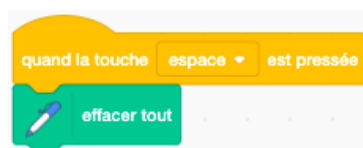



Il suffit ensuite de régler les valeurs et d'accrocher le bloc obtenu sous le bloc **abaisser le stylo**.

Une fois que vous avez terminé et vérifié le code, lancer le programme en appuyant sur le drapeau vert  en haut. Aviez-vous deviné correctement ce qu'il allait se passer ?

### Ajouter l'effacement de l'écran

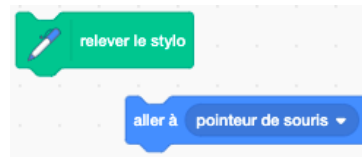
À côté du code précédent, construire le code ci-dessous qui permet d'effacer l'écran lorsque la touche **Espace** est pressée.



A noter que pour accéder aux fonctionnalités du stylo, on peut utiliser l'icône **Ajouter une extension** en bas à gauche .

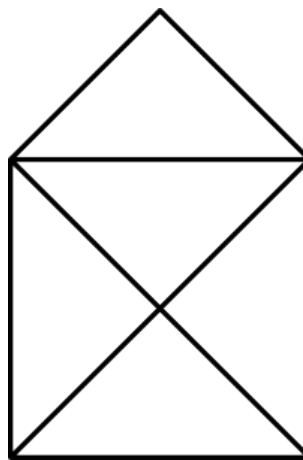
### Un carré où on veut !

En utilisant les instructions ci-dessous, modifier le code pour que le carré soit dessiné à l'endroit où se trouve le pointeur de la souris.



### Dessiner une enveloppe

Créer un nouveau code, toujours pour l'objet sprite, qui permette de dessiner une enveloppe identique à celle ci-dessous. Le but est de réaliser cette enveloppe sans jamais lever le crayon ni repasser deux fois sur le même trait.




### Modifier la taille et la couleur du stylo


En utilisant les instructions suivantes, modifier la couleur des traits et la taille du crayon.

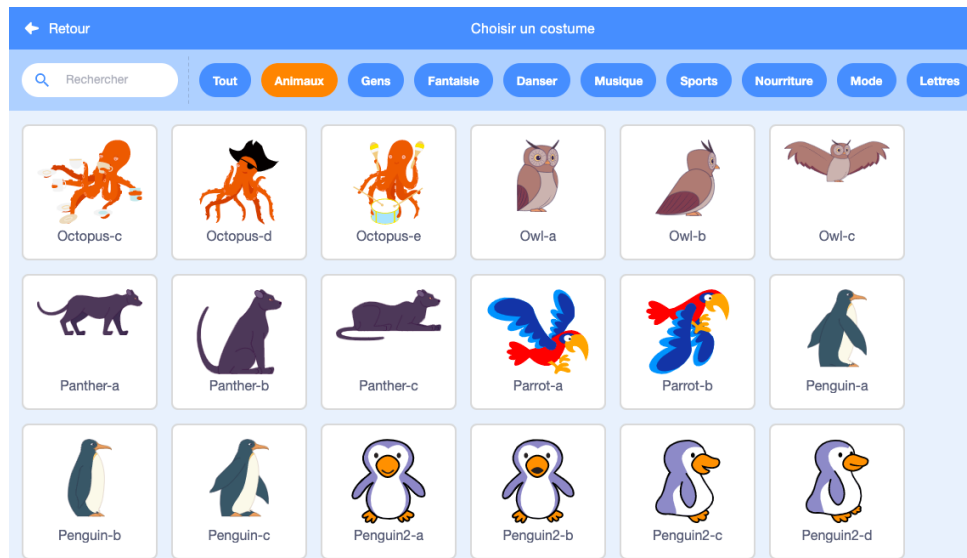


## 4.3 Aide pour la séance 3...

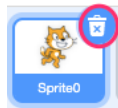
### Ajouter un nouvel objet

1. Cliquer sur l'icône Choisir un sprite , puis sélectionner le costume que vous souhaitez.

2. Par exemple dans l'onglet Animaux, choisir Parrot-a .

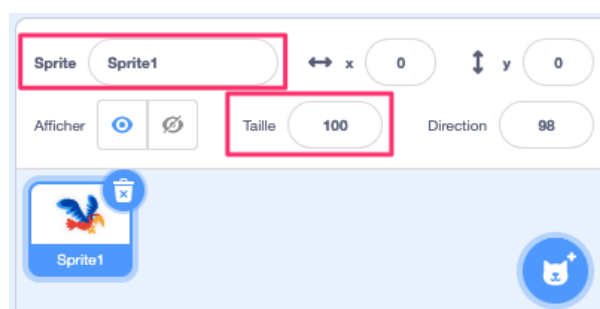


3. Supprimer alors le sprite chat **Sprite0** en cliquant sur le bouton la poubelle liée à



ce sprite

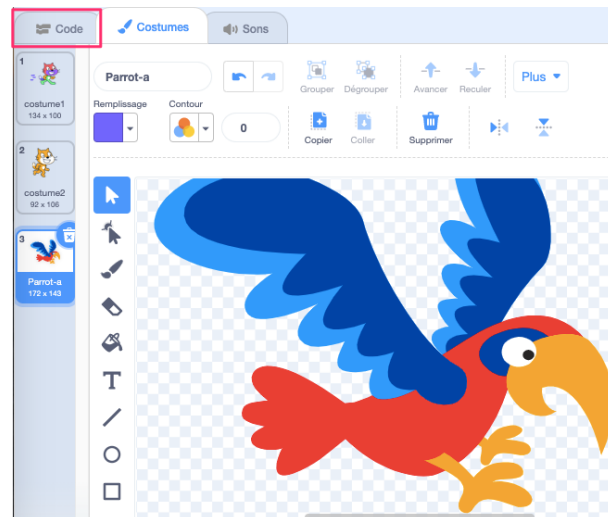
4. Modifier la taille de votre sprite en remplaçant 100 par 50 dans la rubrique **Taille**. Vous pouvez également changer le nom de votre objet en l'appelant par exemple **Sprite1** dans la rubrique **Sprite**, et en appuyant sur la touche **Entrée** de votre clavier.



On a maintenant un plus petit perroquet (objet remplaçant le chat), sur un fond blanc (objet scène).

Vérifier que l'objet **Sprite1** est bien sélectionné et cliquer sur l'onglet **Code** :





### Gérer les mouvements du perroquet dans la scène

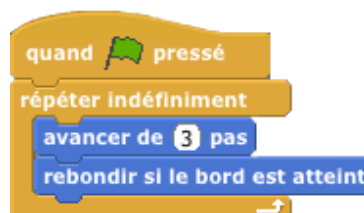
On va maintenant ajouter un code pour notre perroquet. Puisque durant la partie, le perroquet doit toujours avancer, nous allons utiliser une **boucle infinie**.

#### À retenir...

La **boucle** est une structure importante en programmation : elle permet de répéter un bloc d'instructions plusieurs fois, tant qu'une condition est vérifiée ou même indéfiniment. Dans notre programme, nous utilisons une boucle infinie.





1. Construire le code suivant associé à l'objet perroquet (il faut donc que l'objet perroquet soit sélectionné) :



2. Ajouter les deux codes suivants, également associés à l'objet perroquet :



3. Pour tester votre programme, démarrer en appuyant sur le drapeau vert  en haut. Appuyer sur le panneau stop  en haut pour mettre fin au programme.

Pour que le programme s'exécute en plein écran, cliquer sur




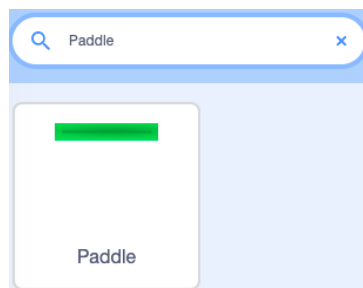
Pour quitter le mode plein écran, cliquer sur



## Un nouvel objet : l'obstacle

On va maintenant ajouter un obstacle que le perroquet devra éviter.

1. Ajouter un nouvel objet en appuyant sur . Choisir maintenant par exemple comme obstacle un **Paddle**



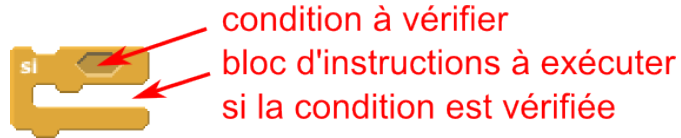
2. Nous avons maintenant un nouvel objet nommé **Paddle**, présent dans la rubrique des sprites à gauche de l'écran.
3. Sélectionner à nouveau l'objet **Sprite1** car c'est à lui que nous allons associer un nouveau code.


## Gérer la collision entre le perroquet et l'obstacle

On va maintenant traiter le cas où le perroquet entre en collision avec l'obstacle. La partie sera alors perdue. Nous allons utiliser ici une **structure conditionnelle** : un bloc d'instructions sera exécuté *si* la condition *Sprite1 percute Paddle* est vérifiée.

**À retenir...**

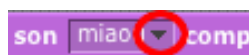
La **structure conditionnelle « si »** est une structure importante en programmation : elle permet d'exécuter un bloc d'instructions **si** une condition est vérifiée.



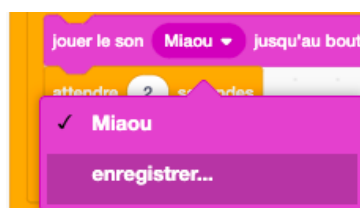
1. Modifier le code pour qu'il corresponde à celui ci-dessous. Le bloc  se trouve dans les blocs Capteur (colonne de gauche).





2. On ne veut pas le son par défaut miaou mais plutôt un son qui annonce que la partie est perdue. Pour cela, il faut enregistrer un nouveau son. Cliquer sur la flèche à droite du nom du son...



...puis choisir enregistrer...



3. Démarrer l'enregistrement à l'aide du bouton  **Enregistrer**, dire « *Perdu !* », puis l'arrêter à l'aide du bouton  **Arrêter l'enregistrement**. Faire plusieurs essais jusqu'à être satisfait du son enregistré.



4. Un nouveau son nommé **recording1** est maintenant disponible.



5. On ajoute encore un quatrième et dernier code associé à **Sprite1**. Ce script permet de repositionner le perroquet lorsque la touche **espace** est pressée :

