



Séquence : Scratch

Niveau	Domaine	Compétences visées	Période	Nombre de séances
Cycle 3 : CM1	Géométrie	1- Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran en utilisant un logiciel de programmation	-	6+

1. Découvrir l'environnement de programmation Scratch

Objectifs :

- S'inscrire sur la plate-forme Scratch
- Découvrir les premiers principes de bases de la programmation par blocs.

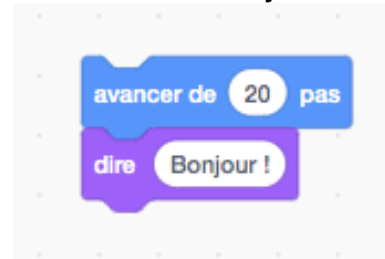
Séances	Déroulement Tâches des élèves	Matériel
1 Date : Durée : 50 min Type : Découverte	<p>1.Présentation du projet (5 min)</p> <ul style="list-style-type: none"> - L'enseignant présente le projet final (cliquer ici) que les élèves vont programmer dans cette séquence. - Il s'agit de l'exploration d'un extra-terrestre dans l'espace à la recherche de ressources pour survivre en évitant quelques dangers. <p>2.Création du compte Scratch (15 min)</p> <ul style="list-style-type: none"> - Les élèves se rendent sur le site Scratch (https://scratch.mit.edu/) et clique sur « Rejoindre Scratch » pour créer un compte. - Ils doivent renseigner un nom d'utilisateur et un mot de passe ainsi que répondre à quelques questions (genre, date de naissance, adresse email) - Cette étape permet d'aborder avec eux la pertinence et les critères d'un mot de passe « fort » comme la présence de lettres et de chiffres, de minuscules et de majuscules, de caractères spéciaux... <p>3.Exploration libre du logiciel (10 min)</p>	<p>iPad</p> <p>Fiche bilan</p>

- Pendant 10 minutes, les élèves explorent librement Scratch. Pour le moment, ils ne doivent pas chercher à modifier la scène ou le lutin (phase suivante) mais l'enseignant les encourage plutôt à s'intéresser à des instructions simples, à les manipuler et à les tester pour observer ce qu'il se passe. Mettre l'accent notamment sur les catégories Mouvement (**bleu foncé**), Apparence (**violet**), Événement (**marbré**) et Contrôle (**orange**)

4.Manipulation (20 min)

- Les élèves réalisent les exercices suivants (notés au tableau au fur et à mesure)

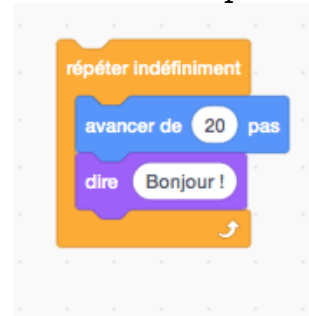
Exercice 1 : Faire avancer le chat de 20 pas et lui faire dire bonjour.



Exercice 2 : Répéter 3 fois : Faire avancer le chat de 20 pas et lui faire dire bonjour.



Exercice 3 : Répéter indéfiniment : Faire avancer le chat de 20 pas et lui faire dire bonjour.



Exercice 4 : Même chose quand on clique sur le drapeau vert.



5. Correction/trace écrite (5 min)

- Distribution de la **fiche bilan**. Les élèves colorient dans la bonne couleur les blocs qu'ils ont utilisé lors de la séance.

Fiche bilan - Scratch : les blocs-instruction

Je colorie, au crayon de couleur, les blocs que je connais dans le logiciel Scratch.

Mouvement

- avancer de 10
- tourner de 15 degrés
- tourner de 15 degrés
- s'orienter à 90
- s'orienter vers
- aller à x: 0 y: 0
- aller à pointeur de souris
- ajouter 10 à x
- donner la valeur 0 à x
- ajouter 10 à y
- donner la valeur 0 à y
- rebondir si le bord est atteint

Apparence

- dire Hello! pendant 2 secondes
- dire Hello!
- montrer
- cacher
- basculer sur l'arrière-plan arrière-plan 1
- ajouter 10 à la taille
- mettre à 100 % de la taille initiale

Données

- créer une variable
- mettre variable à 0
- ajouter à variable 1

Capteurs

- touché?
- touche espace pressée?

Contrôle

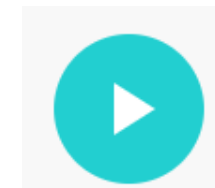
- attendre 1 secondes
- répéter 10 fois
- répéter indéfiniment
- si alors
- si alors
- sinon
- attendre jusqu'à
- répéter jusqu'à
- stop tout

Evenement

- quand presse
- quand espace est pressé
- quand l'arrière-plan bascule sur arrière-plan 1
- quand je reçois message 1
- envoyer à tous message 1

Opérateurs

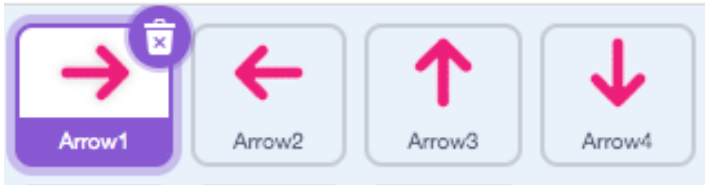
- + - x /
- nombre aléatoire entre 1 et 100
- < = >
- et ou non



2. Réalisation d'un premier programme

Objectifs :

- Déplacer le sprite à l'aide des touches du clavier.

	Séances	Déroulement Tâches des élèves	Matériel
2	<p>Date :</p> <p>Durée : 50 min</p> <p>Type :</p>	<p>1.Présentation des objectifs de la séance (5 min)</p> <ul style="list-style-type: none"> - Les élèves vont apprendre aujourd'hui à personnaliser la scène dans Scratch (sprite et arrière-plan), enregistrer leur travail, se familiariser avec le système de coordonnées et réaliser un premier programme leur permettant de déplacer l'extra-terrestre avec les flèches du clavier. <p>2.Changer le lutin et l'arrière-plan (10 min)</p> <ul style="list-style-type: none"> - Supprimer le lutin « chat ». - Rechercher dans la bibliothèque le lutin « Ghost » - Changer l'arrière-plan et sélectionner « Stars » - Cliquer sur « enregistrer maintenant » pour sauvegarder son travail <p>3.Expérimentation : programmer le lutin « Ghost » (45 min)</p> <p>1. Ajouter 4 sprites « Arrow ». Changer le costume des flèches pour avoir les 4 directions.</p> 	<p>iPad</p> <p>Claviers</p> <p>Fiche bilan</p>

2. Programmer une flèche pour pouvoir déplacer le fantôme.

L'enseignant montre la programmation de la flèche de droite. Voici la logique :

Quand je clique sur la flèche de droite à l'écran, j'envoie un message qui s'appelle « droite » à tous les autres sprites.

Quand mon fantôme reçoit le message « droite », on ajoute 10 à x (cela le fera avancer vers la droite).



Programme de la flèche



Programme du Ghost

3. Programmer les autres flèches.

Les élèves recopient le programme et cherchent la programmation pour les autres flèches afin de déplacer le fantôme dans les 4 directions.

4. Orienter le fantôme.

Les élèves complètent ensuite le code du fantôme pour que le sprite s'oriente vers le haut quand il reçoit le message « haut », vers le bas quand il reçoit le message « bas » et ensuite de suite. À partir du modèle ci-dessous (pour s'orienter en haut), les élèves testent l'orientation du sprite et complètent les autres directions.



5. Rebondir sur les bords

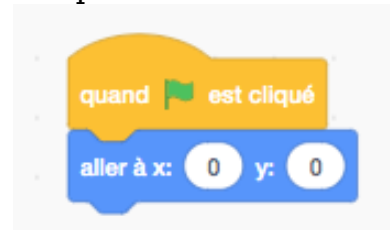
Les élèves cherchent comment faire en sorte que le sprite rebondisse sur les bords quand il atteint le bord de l'écran.

Résultat final attendu



6. Initialiser la position du Ghost

L'enseignant rappelle que, lorsqu'on lance le programme (drapeau vert), le Ghost doit se situer au centre de l'écran. Les élèves reprennent sans difficulté les instructions qu'ils avaient vues lors de la première séance.

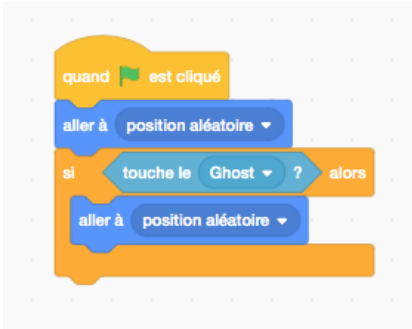




3. Compléter le programme

Objectifs :

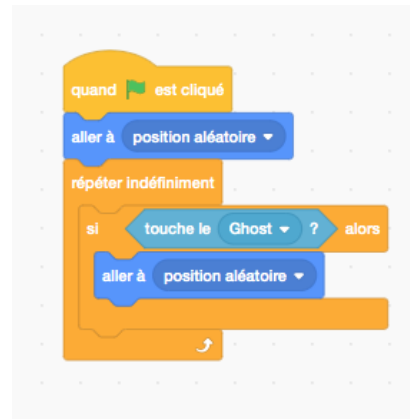
- Création de nouveaux sprites, d'une variable score.
- Apprendre à programmer des instructions conditionnelles (si...alors) et à utiliser des capteurs.

Séances	Déroulement Tâches des élèves	Matériel
<p>2</p> <p>Date : Durée : 60 min Type :</p>	<p>1.Ajouter le Sprite « Crystal » (5 min)</p> <ul style="list-style-type: none"> - Programmer : « Quand le drapeau vert est cliqué, mon sprite va à une position aléatoire ». <p>2.Déplacer le Sprite quand il est touché.(30 min)</p> <ul style="list-style-type: none"> - Nous souhaitons que le Sprite « Crystal » se déplace à un position aléatoire au moment où il est touché par notre Ghost. (cela nécessite l'utilisation de la fonction « si...alors » et aussi l'utilisation d'un capteur « touche le ? ») <p><u>Résultat attendu :</u></p>  <p>Malheureusement, lorsqu'on lance le programme et que l'on dirige le Ghost vers le Crystal, cela ne fonctionne pas.</p>	<p>iPad</p> <p>Claviers</p> <p>Fiche bilan</p>

On remarque, en lisant ce programme, que le test n'est effectué qu'une seule fois, au lancement du programme (juste après qu'on a initialisé la position du Crystal). Or, à ce moment-là, les 2 sprites ne se touchent pas. Donc, le sprite ne se déplace pas une fois touché. C'est normal.

Pour que le programme fonctionne correctement, il faut que le test « si le Crystal touche le Ghost » soit effectué en permanence. Pour cela, il suffit de placer ce test dans une boucle « **répéter indéfiniment** ».

Le résultat attendu devient donc :



3. Créer une variable score (10 min)

L'enseignant rappelle aux élèves qu'ils doivent créer un score et l'augmenter à chaque fois que la ressource est touchée. Laisser les élèves chercher.

La solution se trouve dans la partie « **variables** » des commandes puis de cliquer sur « **créer une variable** »

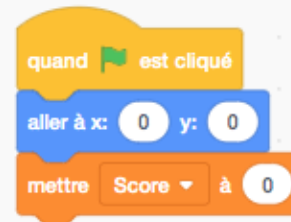
4. Augmenter le score lorsqu'on récolte un Crystal et Initialiser le score à 0 en début de partie (15 min)

Modifier le code du Sprite Crystal pour que: « Si le Crystal est touché par le Rover, alors on ajouter 1 au score ».

Résultats attendu :



Pour initialiser le score à 0 en début de partie, il suffit de reprendre le programme du Ghost et d'ajouter la commande « **mettre le score à 0** » quand le drapeau vert est cliqué ce qui donne le code suivant :



5. Fiche bilan (10 min)

Les élèves complète la fiche bilan en coloriant les nouvelles commandes utilisées.

Devoirs : les élèves pour la séance d'après doivent ajouter le lutin « potion » et refaire le même travail que pour le Crystal.


4. Éviter des obstacles, gérer son nombre de vies

La séance en vidéo....



Objectifs :

- Création de nouveaux sprites, d'une variable vie.
- Consolider les instructions conditionnelles (si...alors) et l'utilisation des capteurs.
- Découvrir les blocs d'envoi de messages.

Séances	Déroulement Tâches des élèves	Matériel
<p>2</p> <p>Date : Durée : 60 min Type :</p>	<p>Présentation des objectifs : <i>Pour améliorer le jeu, l'enseignant explique qu'il va falloir introduire des obstacles (une chauve-souris et des météorites) et une nouvelle variable : le nombre de « vies ». Le Ghost démarre le jeu avec 3 vies et perd une vie, en revenant à la base, chaque qu'il touche un obstacle. Le jeu s'arrête lorsque le nombre de vies atteint 0.</i></p> <p>1. Ajouter de nouveaux Sprites (5 min)</p> <ul style="list-style-type: none"> - Ajouter les Sprites « Bat » et « Rocks » - Programmer : « Quand le drapeau est cliqué, aller à une position aléatoire » <p>2. Création et initialisation d'une variable « nombre de vies » (10 min)</p> <ul style="list-style-type: none"> - Ils créent la nouvelle variable (sur le même modèle que la variable « score ») et ils l'initialisent à 3 (au même endroit que l'initialisation du score). Le programme du Ghost devient donc : 	<p>iPad</p> <p>Claviers</p> <p>Fiche bilan</p>

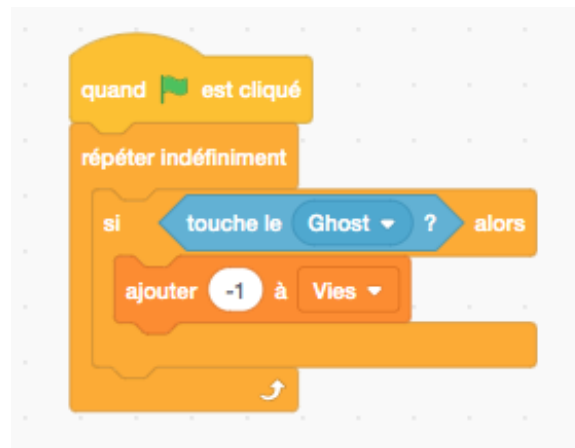
3. Perdre une vie quand le Ghost touche la chauve-souris (30 min)

À la séance précédente, toucher une ressource permettait d'augmenter de 1 la valeur de la variable « score ». De même, ici, toucher la météorite ou la chauve-souris doit faire diminuer de 1 la valeur de la variable « vie ». Nous conseillons de résoudre d'abord la tâche pour un des obstacles (la chauve-souris), puis de recommencer pour l'autre (la météorite).

Plusieurs solutions sont possibles, voici celle idéale :

Programmation de la chauve-souris

1. Lorsque le drapeau vert est cliqué :
Aller à une position aléatoire
Utiliser le bloc « Si... alors » → Si le Ghost est touché ?
alors on ajouter -1 à vie

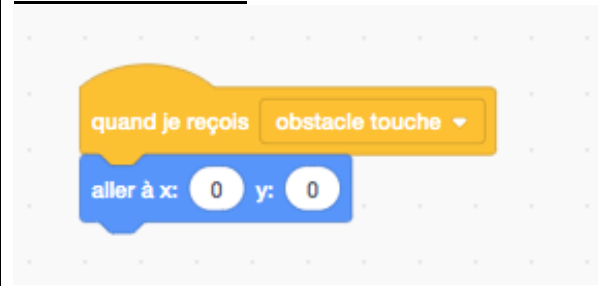


Ce programme a cependant un défaut : lorsque le Ghost entre en contact avec la chauve-souris, il le touche pendant un certain temps (quelques dixièmes de secondes, quelques secondes si on ne bouge plus) : pendant tout ce temps, la variable « vie » décroît. Au bout de quelques secondes, on se retrouve avec des valeurs négatives très grandes (- 4000...). La seule façon d'éviter cela est de rompre immédiatement le contact entre le Ghost et l'obstacle.

Puisque l'obstacle est fixe, c'est donc le Ghost qui doit se déplacer. Or, cela ne peut pas être fait dans le programme de la chauve-souris ou de la météorite : il faut être dans le programme du Ghost pour commander la position du Ghost.

Une solution consiste à garder le programme que l'on a fait pour la chauve-souris, en y ajoutant une nouvelle instruction qui va envoyer un message aux autres programmes (celui du Ghost en particulier). Ce message, dans le programme du Ghost, déclenchera une action (retourner à la base). Tout comme les noms de variables, les intitulés des messages doivent être explicites. Ici, notre message est par exemple « obstacle touché ».

Résultat attendu :



Programme du Ghost



Programme de la chauve-souris

		<p>4. Refaire la même chose avec la météorite (20 min)</p> <p>Maintenant que les élèves ont réussi à gérer un des obstacles (la chauve-souris), ils doivent refaire la même chose pour l'autre (la météorite). Ce petit exercice permet de mieux s'approprier ce qui a été fait précédemment, en particulier l'envoi et la réception de message.</p>	
--	--	---	--

Devoirs : les élèves pour la séance d'après doivent terminer la programmation de la météorite.

5. Mettre fin au jeu : « game over »

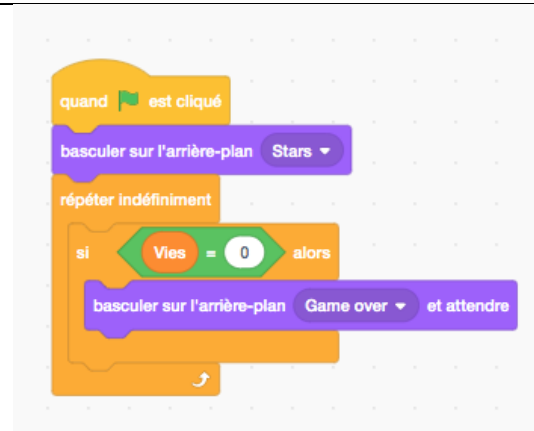
La séance en vidéo....



Objectifs :

- Faire apparaître « game over »
- Faire en sorte qu'on ne puisse plus jouer (sauf en relançant le programme)

Séances		Déroulement Tâches des élèves	Matériel
2	Date : Durée : 60 min Type :	<p>Présentation des objectifs : <i>Montrer le résultat attendu aux élèves : quand le nombre de vies atteint 0, le fond d'écran change pour faire apparaître « Game Over »</i></p> <p>1.Ajouter la scène « Game Over » (5 min)</p> <ul style="list-style-type: none">- Dupliquer la scène « Stars »- Ajouter le texte « GAME OVER » en police « Pixel » et couleur rouge au milieu de l'écran.- Renommer l'arrière-plan « Game Over'' <p>2.Faire apparaître « Game over » quand le nombre de vie vaut 0 (5 min)</p> <ul style="list-style-type: none">- La solution la plus facile consiste à programmer la scène « Game over » qui vient d'être créée de la manière suivante : « Au drapeau vert, basculer sur l'arrière-plan « Stars », si le nombre de vie = 0, alors je bascule sur l'arrière-plan « Game Over » ».- Cela sera l'occasion aussi de consolider l'emploi de la boucle « Si...alors » ainsi que la répétition indéfinie (sinon, le programme ne fonctionnera pas).	iPad Claviers Fiche bilan



3. Arrêter le jeu quand apparaît « Game Over » (30 min)

En l'état actuel, on peut continuer à jouer lorsque « game over » s'affiche, ce qui n'est pas satisfaisant. Pour mettre fin au jeu, la solution consiste à stopper tout grâce à la commande « **Stop tout** » dans la catégorie « **contrôle** ».

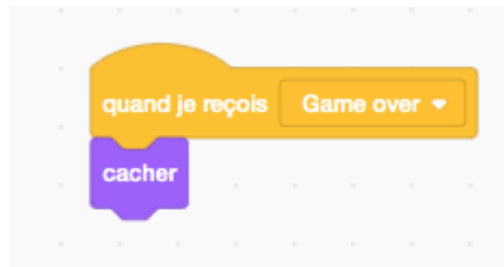
Avant cela, le programme de l'arrière-plan « Game Over » doit envoyer un message à tous les autres lutins. On nommera le message « Game over ».

Chaque lutin qui reçoit le message « Game Over » se cache. De cette manière, tout disparaîtra une fois le nombre de vie arrivé à 0.

Voici le résultat attendu :



Programme de l'arrière-plan



Programme des autres sprites

Note importante :

Ne pas oublier, pour chaque lutin, d'ajouter le bloc « **montrer** » quand je clique sur le drapeau vert.


6. Améliorer le jeu

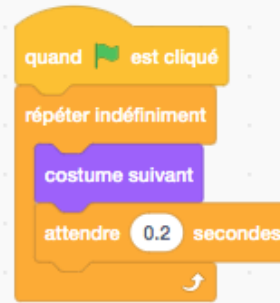
La séance en vidéo....



Objectifs :

- Peaufiner le jeu en ajoutant du son.
- Ajouter de la difficulté au jeu en faisant bouger les obstacles.

Séances	Déroulement Tâches des élèves	Matériel
<p>2</p> <p>Date : Durée : 60 min Type :</p>	<p>Présentation des objectifs : <i>Les élèves auront remarqué que leur jeu est jouable mais qu'il manque d'intérêt car il ne présente pas de difficulté majeure. Si l'on fait attention aux obstacles, le jeu peut ne jamais s'arrêter. Proposer à la classe de faire bouger la chauve-souris ainsi que la météorite.</i></p> <p>1. Faire bouger la chauve-souris (15 min)</p> <ul style="list-style-type: none"> - Quand le drapeau vert est cliqué, le sprite va à une position aléatoire, puis il se déplace en 2 sec à une autre position aléatoire et ce, indéfiniment.  <ul style="list-style-type: none"> - Il est possible aussi d'animer la chauve-souris en changeant son costume. 4 costumes différents sont déjà créés pour la chauve-souris. Il suffit donc de programmer la chauve-souris pour qu'elle change de costume toutes les 0.2 secondes. 	<p>iPad</p> <p>Claviers</p> <p>Fiche bilan</p>



2. Faire bouger la météorite (20 min)

- A ce stade de la séquence, il est possible de laisser les élèves libres de chercher une solution pour la météorite. Ils peuvent choisir de faire disparaître la météorite et la faire réapparaître aléatoirement.



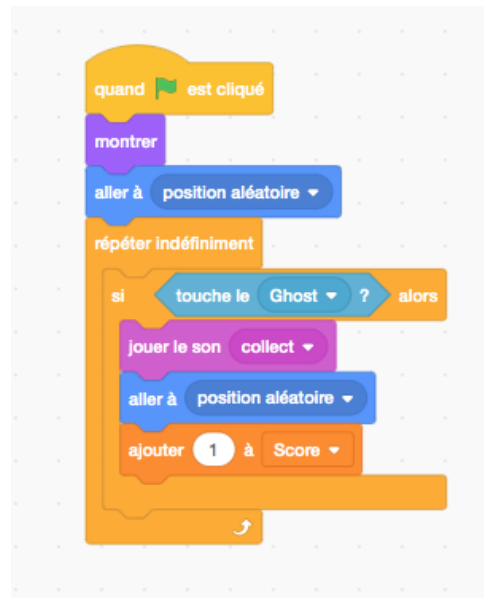
- Une autre solution bien plus complexe (celle présentée dans la version finale du jeu en début de séquence) consiste à créer une pluie de météorite défilant de la droite vers la gauche de l'écran.

Les élèves peuvent, pour cela, s'aider du tutoriel suivant : <https://ladigitale.dev/digiview/#/v/67751f5ca8b36>

3.Ajouter le son (20 min)

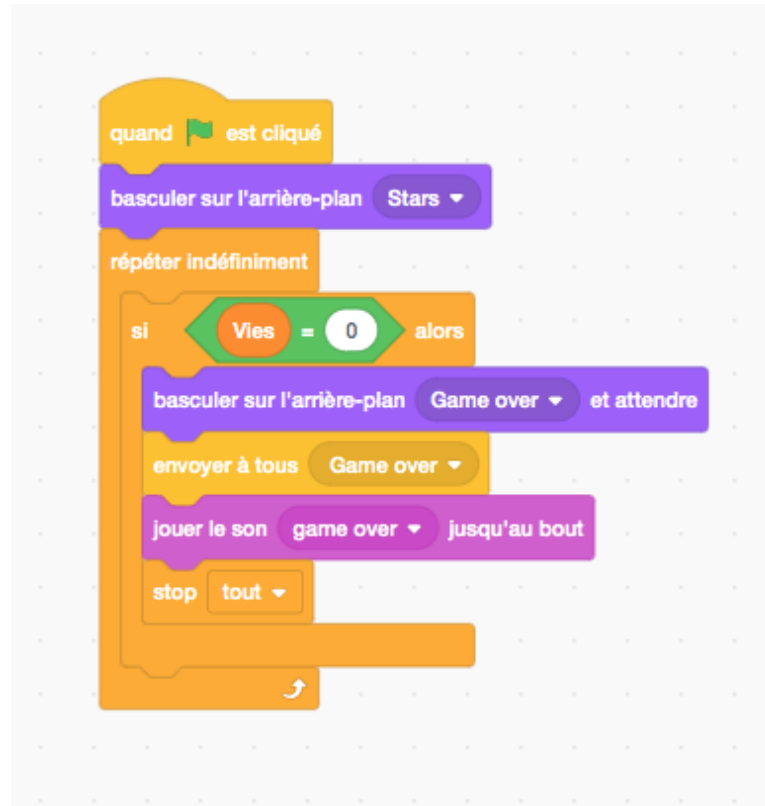
Les élèves terminent leur jeu en ajoutant maintenant le son. Pour cela, pour chaque sprite (rocks, bat, potion et crystal), un son est joué chaque fois que le Ghost est touché. Il suffit de compléter le programme en ajoutant le bloc son de son choix au bon endroit.

Exemple du Crystal :



Pour le programme de l'arrière-plan, les élèves peuvent enregistrer un son eux-mêmes (disant « Game Over ») et l'ajouter au programme. Pour cela, ils sélectionnent l'arrière-plan et vont ensuite dans l'onglet « **Son** ». Ils ajoutent un son en sélectionnant l'option de s'enregistrer.

Une fois le son enregistré, ils le renomment (« Game over » par exemple) et ajoute le bloc au programme.



Le jeu est maintenant terminé et intéressant. Les élèves peuvent colorier tous les blocs qu'ils ont utilisé pour programmer dans leur fiche bilan.