

Belgium



Avec le soutien financier de



Les débris spatiaux

Fiche pour l'enseignant

En écrivant un programme d'un petit jeu de récupération des débris spatiaux, les élèves approfondissent la découverte du langage de programmation Scratch (variables, boucles, conditions, événements, manipulation des sprites).



Programmation

Age : 10-14 ans

Les débris spatiaux

Objectif

- ✓ Programmer à l'aide du logiciel de programmation Scratch un jeu consistant à ramasser des débris suite à une explosion dans l'espace

Notions abordées : Programmation, instructions, mathématiques

- 1 – Contexte
- 2 – Prérequis
- 3 - Jeu de base
- 4 - Options supplémentaires du jeu

Durée : ~ 2h (+1h avec les options supplémentaires du jeu)

Dispositif pédagogique : 1 ordinateur par jeune pour les groupes de 12 jeunes ou 1 ordinateur par binôme pour les groupes de 24 jeunes

Matériel

- Ordinateurs avec le logiciel Scratch Desktop

Références & liens utiles

- Scratch mit : <https://scratch.mit.edu/>
- Fiche pédagogique pour s'initier au logiciel Scratch et à la programmation en blocs « [Parcours Découverte Thymio & Scratch](#) »
- Conférence Vidéo ESERO Belgium « [L'espace, une opportunité et une responsabilité: l'enjeu des débris spatiaux](#) » par le professeur Olivier Chaziot du Von Karman Institute (Belgique)
- [Programme pour le jeu de base](#) sur le site Scratch mit
- [Programme pour le jeu avec options](#) sur le site Scratch mit

Remarque

Pour des raisons d'ergonomie de lecture, le texte de cette fiche pédagogique n'est pas rédigé en écriture inclusive mais il s'adresse néanmoins tant aux hommes qu'aux femmes, ainsi qu'aux personnes non-binaires.

Droits d'auteur

Le contenu de cette fiche pédagogique est publié sous licence Creative Commons Attribution - Pas d'utilisation commerciale - Partage dans les mêmes conditions ([CC-BY-NC-SA](#)) :

Attribution [BY] (*Attribution*) : l'œuvre peut être librement utilisée, à la condition de l'attribuer à l'auteur en citant son nom : La Scientotheque. Cela ne signifie pas que l'auteur est en accord avec l'utilisation qui est faite de ses œuvres.

Pas d'utilisation commerciale [NC] (*Noncommercial*) : le titulaire de droits peut autoriser tous les types d'utilisation ou au contraire restreindre aux utilisations non commerciales (les utilisations commerciales restant soumises à son autorisation). Elle autorise à reproduire, diffuser, et à modifier une œuvre, tant que l'utilisation n'est pas commerciale.

Partage dans les mêmes conditions [SA] (*ShareAlike*) : le titulaire des droits peut autoriser à l'avance les modifications ; peut se superposer l'obligation [SA] pour les œuvres dites dérivées d'être proposées au public avec les mêmes libertés que l'œuvre originale (sous les mêmes options Creative Commons).

Description détaillée

1 – Contexte

L'espace est un formidable terrain d'exploration et de développement technologique, mais souffre également de son exploitation. Les débris y prolifèrent, qu'ils résultent d'évènements accidentels ou intentionnels. Ces débris ont déjà un impact et constituent une menace pour les activités spatiales présentes et futures.

[Une page dédiée](#) à la thématique des débris spatiaux est disponible sur le site d'ESA Education.

Il est également possible de visionner la Conférence Vidéo ESERO Belgium [« L'espace, une opportunité et une responsabilité : l'enjeu des débris spatiaux »](#) par le professeur Olivier Chaziot, chef du département Aérospatiale et Aéronautique de l'Institut von Karman de Dynamique des Fluides (Rhode-Saint-Genèse) qui étudie de façon théorique mais aussi expérimentale la façon dont les satellites sont détruits (ou non) en rentrant dans l'atmosphère terrestre.

2 – Prérequis

La réalisation de ce jeu fait appel à un certain nombre de notions récurrentes en programmation : variables, conditions, boucles, etc. **Il est recommandé de se familiariser avec ces notions et de s'être initié au logiciel Scratch au préalable.** La fiche pédagogique [« Parcours Découverte Thymio & Scratch »](#) mentionnée dans la section « Références et liens utiles » permet de prendre en main le logiciel Scratch et de découvrir les bases du langage de programmation en blocs.

Cette fiche pédagogique se présente comme un guide pour l'enseignant ou l'animateur afin qu'il puisse accompagner les jeunes, étape par étape, dans la création et la programmation du jeu. D'un point de vue pédagogique, il est recommandé de présenter chaque étape que les jeunes doivent réaliser, de les laisser réfléchir puis de reprendre avec eux les différentes solutions trouvées afin d'en choisir une. Le programme Scratch présenté dans cette fiche est une des possibilités pour arriver au résultat visé, mais n'est pas la seule solution.

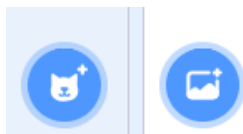
3 – Jeu de base

Le principe du jeu que les jeunes vont devoir programmer est le suivant : des débris spatiaux défilent sur l'écran, le joueur doit les intercepter en déplaçant son personnage contrôlé, ici un petit robot, en utilisant les touches fléchées du clavier.

Le [programme pour le jeu de base](#) se trouve sur le site de Scratch.

a) Le robot

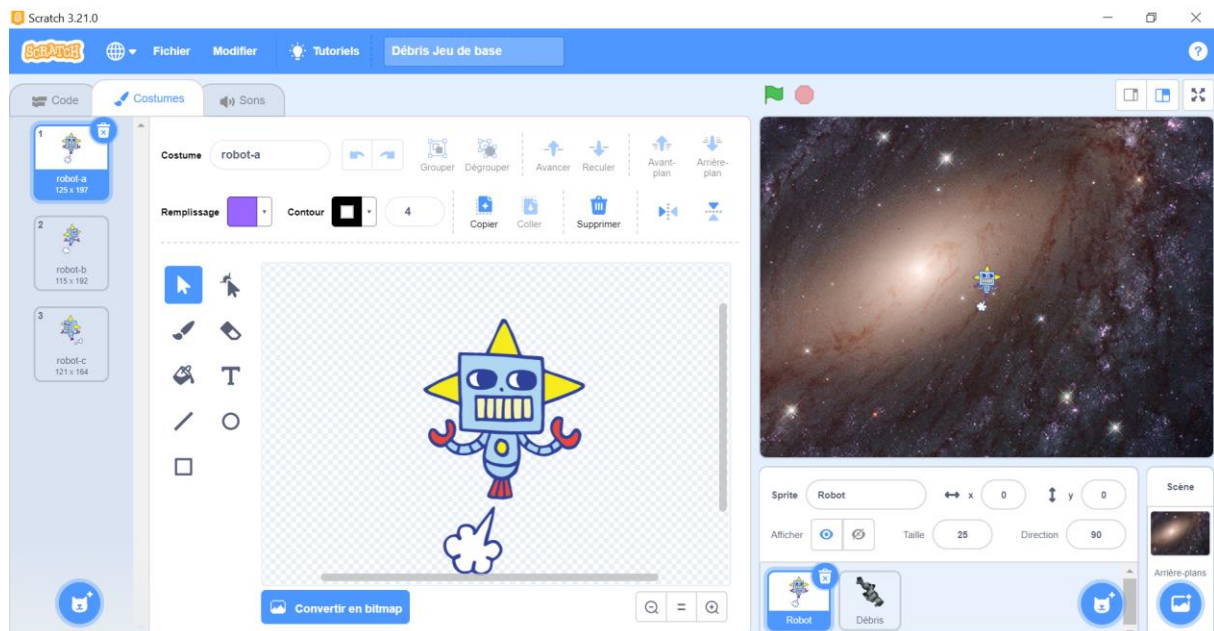
Le joueur contrôle un robot. Commençons donc par coder celui-ci.



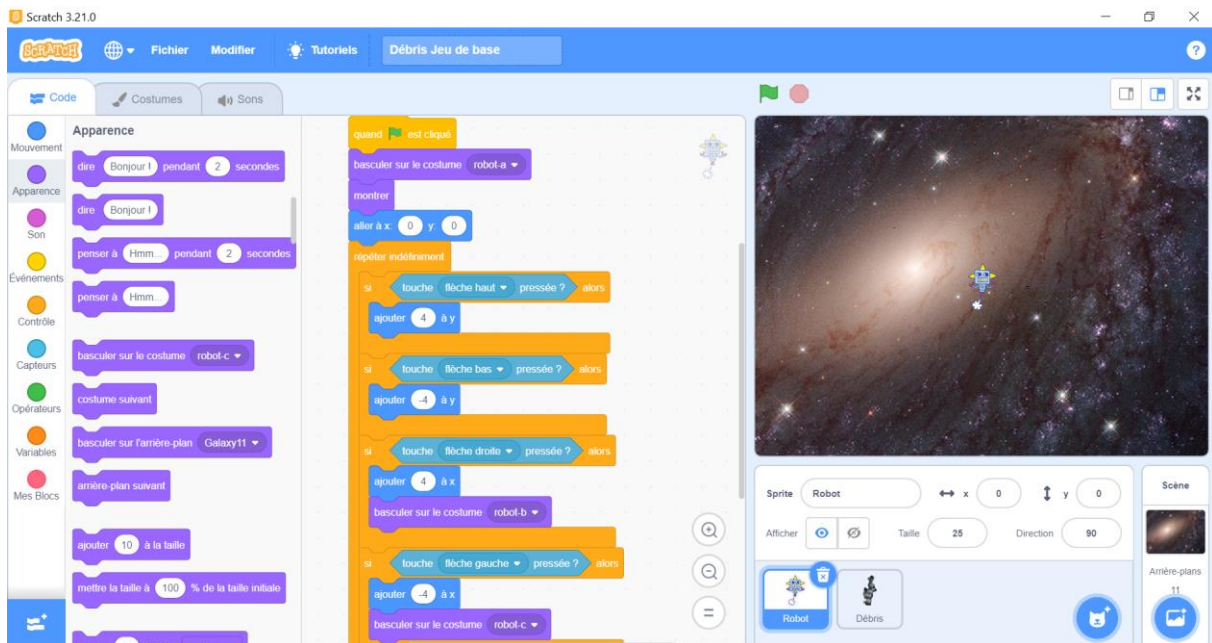
et scène.

Pour rappel, Scratch fonctionne avec des sprites (lutins), c'est-à-dire des personnages ou objets à programmer, et il est possible de personnaliser le jeu en choisissant son sprite ou son arrière-plan. Il suffit pour cela d'utiliser les puces bleues au bas des zones sprite

- Choisir l'arrière-plan « Galaxy » (disponible dans la base de données de Scratch)
- Remplacer le sprite par défaut (personnage de chat) par le robot (inclus dans la base de données de Scratch)
- Créer trois costumes possibles au robot dans l'onglet Costumes (« robot-a » de face, « robot-b » de profil à gauche, « robot-c » de profil à droite) comme présenté ci-dessous :



- Construire ensuite la séquence de blocs dans l'onglet Code permettant au robot de se déplacer sous le contrôle des touches fléchées comme visible ci-dessous :

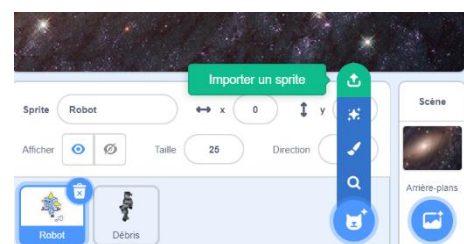


A ce niveau, vous pouvez en profiter pour faire un rappel sur les notions de boucle et d'instruction conditionnelle (Si... Alors). Il peut être utile d'insister sur la façon dont le robot va se déplacer: le code permet au robot de se déplacer grâce à des changements de costumes et des opérations sur les abscisses et les ordonnées du robot, plutôt que de le faire pivoter puis avancer. Cela permet de modifier les deux paramètres simultanément et d'avoir plus de liberté dans les mouvements.

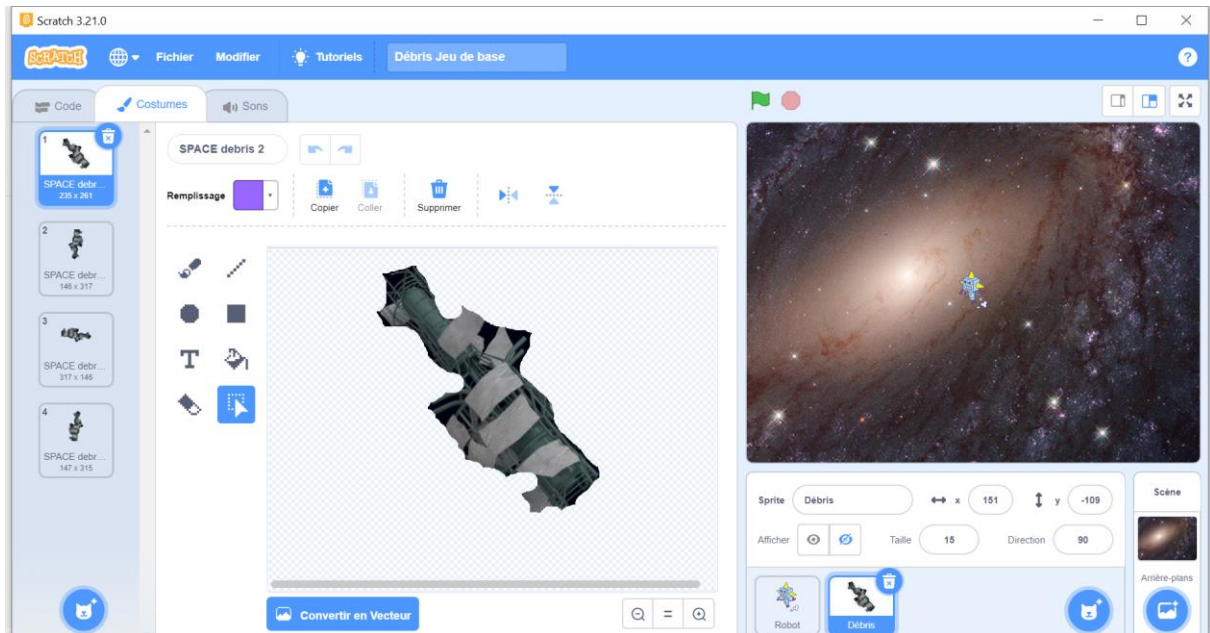
b) Les débris

Le deuxième élément important de ce jeu sont les débris à intercepter.

- Commencer par **importer un sprite** qui fera office de débris. Il est possible d'en chercher une image sur internet et de l'importer mais n'importe quel sprite inclus dans la base de données de Scratch peut faire l'affaire (tant qu'il est distinguable du robot contrôlé par le joueur).



- Comme pour le sprite robot, créer 4 costumes au sprite débris en modifiant leur orientation.

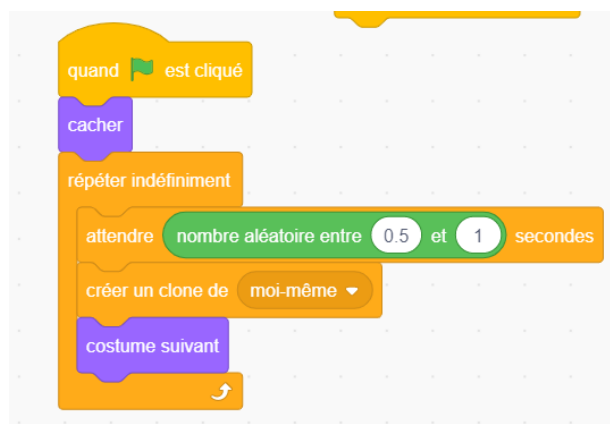


Il va ensuite falloir créer des clones de ce débris pour les faire apparaître plusieurs fois à l'écran sans avoir à coder chacun d'entre eux. Attention cependant, les clones se codent en même temps que le sprite d'origine et partageront donc le même code.

Le code du débris sera constitué de deux séquences : une qui générera les clones, et la deuxième qui définira leur comportement.

- Créer la séquence pour **générer les clones**

Le code visible ci-dessous est celui qui permet de créer les clones. Au lancement du programme, il faut cacher le sprite débris d'origine car ce n'est pas lui qui se déplace réellement mais ses clones. La boucle infinie (« répéter indéfiniment ») et les blocs qu'elle contient servent à trois choses : définir la fréquence de génération des clones (ici en l'occurrence en définissant une période comprise entre 0.5 et 1 seconde), créer le clone à proprement parler (bloc se trouvant dans la section contrôle) et le faire changer d'apparence en modifiant le costume (ce n'est pas indispensable mais cela permet d'apporter un peu de variété).



- Créer la séquence pour **définir le comportement des clones**

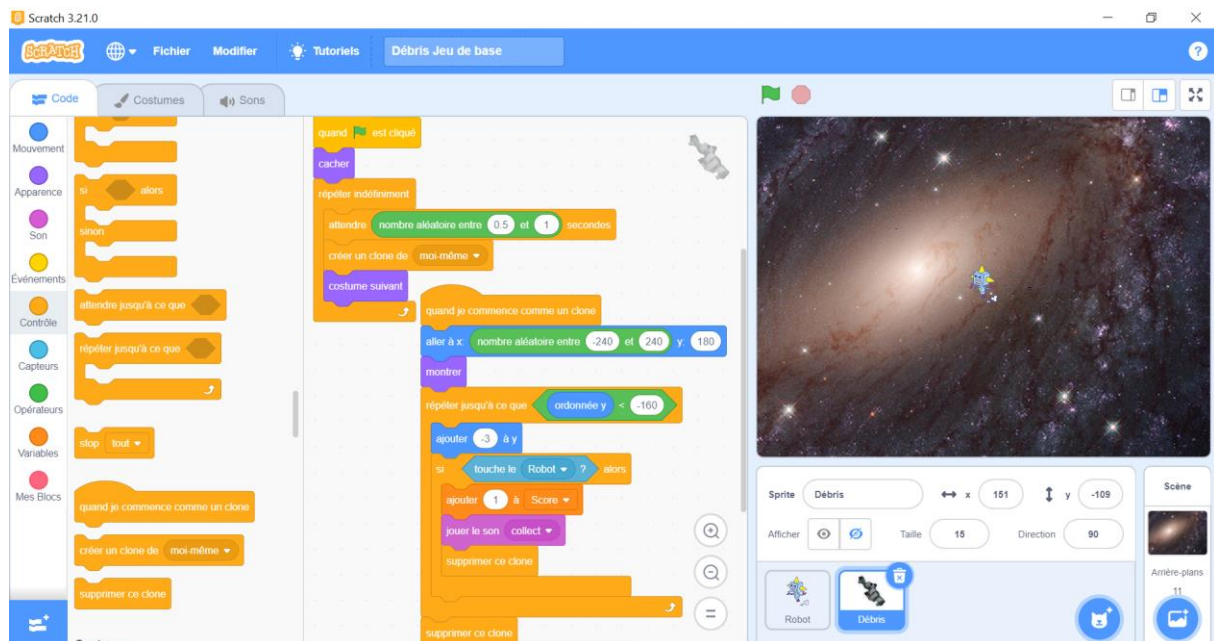
Le bloc de départ « quand je commence comme un clone » (section contrôle) permet de faire en sorte que la séquence de code qui le suit s'applique à chaque clone créé. Il faut ensuite définir la position de départ du clone et le montrer (puisque c'est une copie de l'original qui est caché). L'utilisation d'une valeur aléatoire pour l'abscisse x correspondant à la position du clone permet de le faire apparaître aléatoirement à l'écran.

Enfin, une boucle se déroule jusqu'à ce que l'ordonnée y correspondant à la position du clone atteigne une valeur inférieure à -160, c'est-à-dire lorsque le clone atteint le bas de l'écran. Dans la boucle, le bloc « ajouter -3 à y » permet de faire se déplacer le clone vers le bas de l'écran. Une instruction conditionnelle indique ensuite que, dès que le sprite robot touche le clone, le score doit être augmenté de 1 point, un son doit être joué et le clone peut disparaître. Si le robot n'a pas touché le clone, celui-ci peut disparaître grâce au bloc final « supprimer ce clone ».

```

quand je commence comme un clone
  aller à x: nombre aléatoire entre -240 et 240 y: 180
  montrer
  répéter jusqu'à ce que < ordonnée y < -160 >
    ajouter -3 à y
    si touche le Robot ? alors
      ajouter 1 à Score
      jouer le son collect
      supprimer ce clone
  supprimer ce clone
  
```

Voici le programme final obtenu pour le sprite Débris :



Le jeu est maintenant programmé: il est possible de le lancer et vérifier que tout fonctionne!

3 – Options supplémentaires du jeu

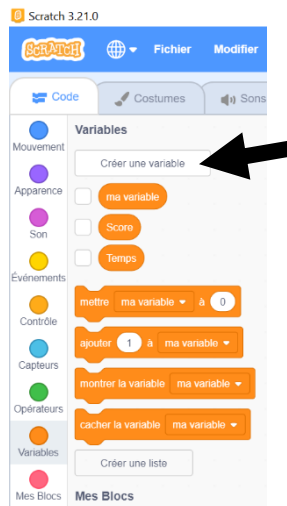
Il est possible d'ajouter un certain nombre d'options pour sophistication le jeu : éléments de décors, effets visuels, fonctions supplémentaires...

Le [programme du jeu avec options](#) se trouve sur le site de Scratch.

a) Le temps et le score

Une option intéressante pour complexifier le jeu est d'ajouter un intervalle de temps dans lequel le jeu se déroule ainsi qu'un score (nombre de fois que le sprite robot attrape un débris).

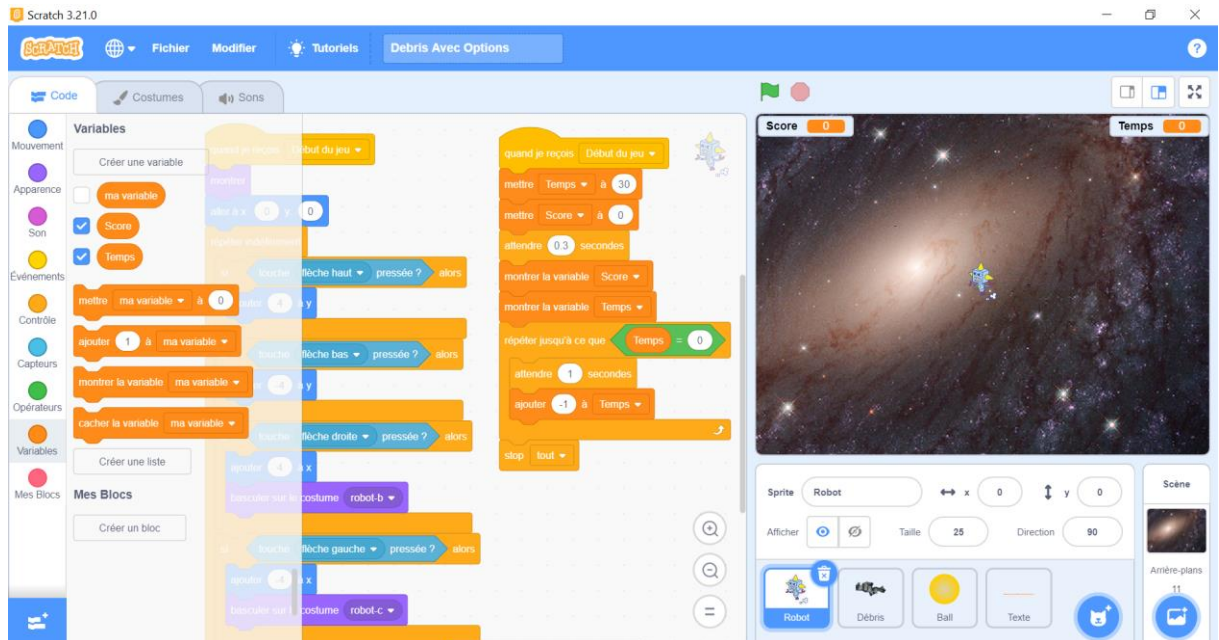
- Créer une variable « Score » et une variable « Temps » dans la section Variables.



- Dans la zone de code du sprite Robot, construire la séquence suivante qui consiste à initialiser les variables Temps et Score (30 secondes et 0 point) avant de les faire apparaître à l'écran. Une boucle est ensuite ajoutée pour lancer le compte à rebours (« ajouter -1 à Temps »). Enfin, le bloc « stop tout » met fin au programme.

```

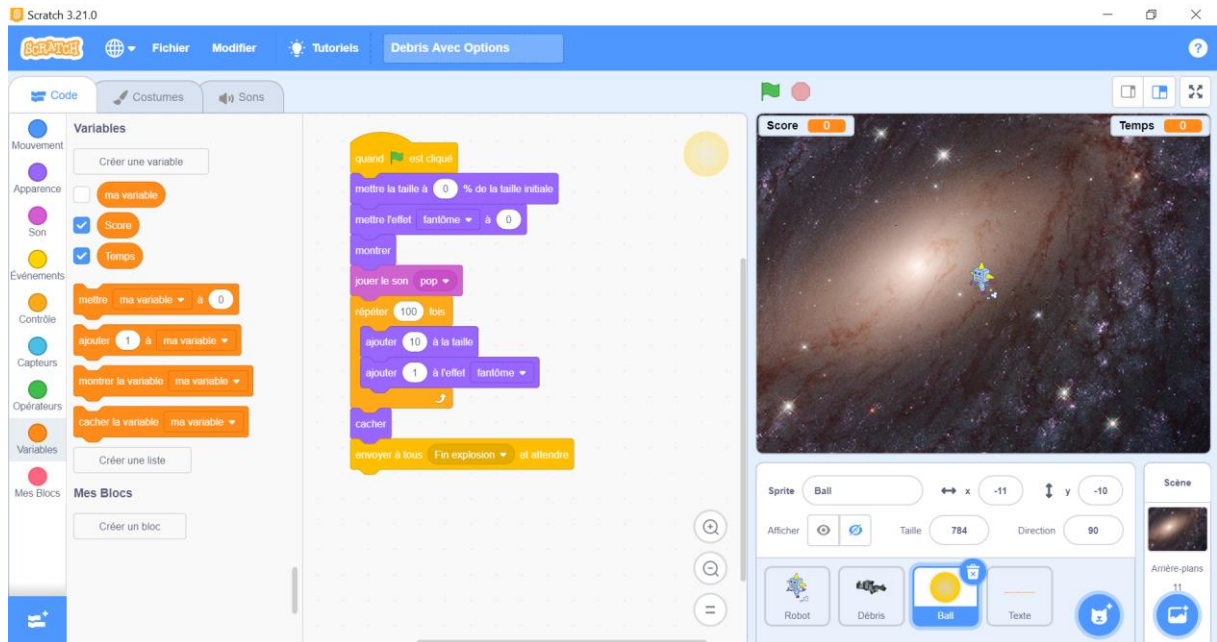
quand est cliqué
mettre Temps à 30
mettre Score à 0
attendre 0.3 secondes
montrer la variable Score
montrer la variable Temps
répéter jusqu'à ce que Temps = 0
attendre 1 secondes
ajouter -1 à Temps
stop tout
    
```



b) Personnalisation de l'interface

Il est possible d'ajouter des éléments visuels pour personnaliser l'interface de jeu comme par exemple :

- Une animation figurant une explosion initiale
 - Importer le sprite Ball (disponible dans la banque de données de Scratch)

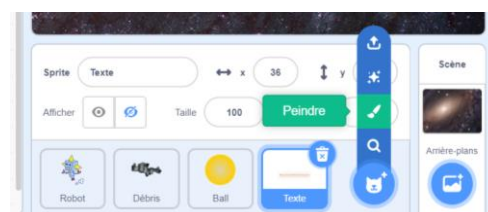


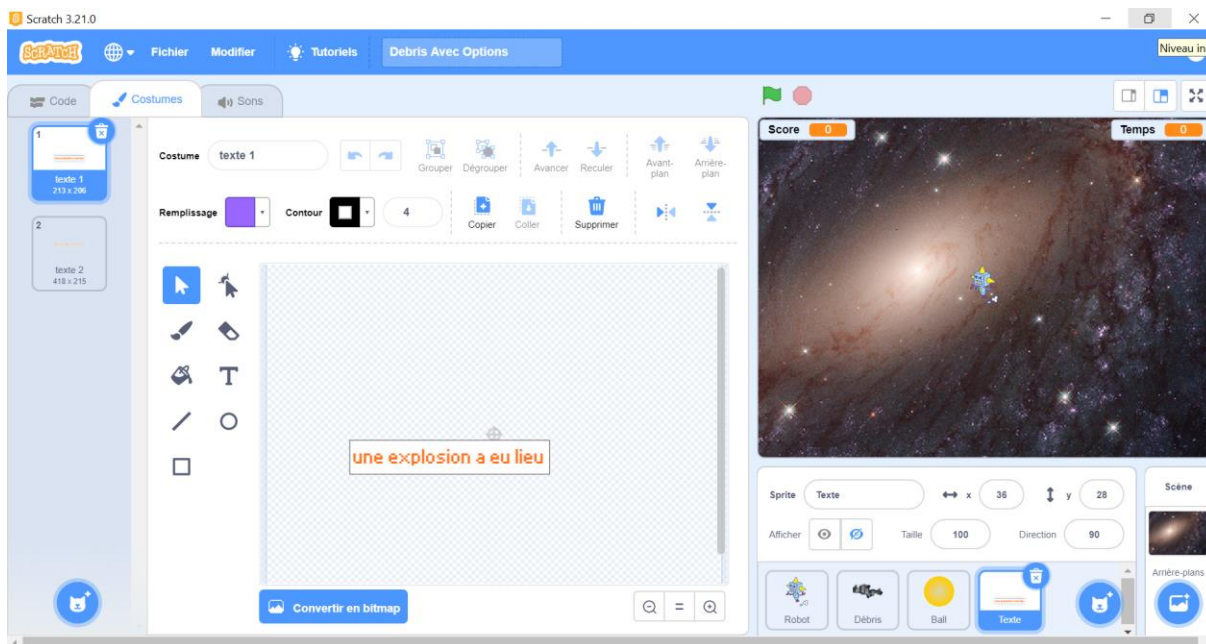
- Construire la séquence ci-contre qui consiste d'abord à réduire la taille du sprite ball à 0%, et mettre l'effet fantôme à 0 (qui correspond à un effet de transparence nulle) avant de le faire apparaître. Un son Pop est ensuite joué au moment de l'apparition, puis le sprite grossit et devient de plus en plus transparent pour simuler l'effet d'une explosion grâce à une boucle répéter et aux blocs « ajouter 10 à la taille » et « ajouter 1 à l'effet fantôme ». Une fois la boucle terminée, le sprite est caché une fois. Pour que cette animation commence avant que le sprite robot n'apparaisse à l'écran, il faut ajouter un bloc « envoyer à tous Message » (dans la section Événements) et de créer un nouveau message, par exemple « Fin explosion ».



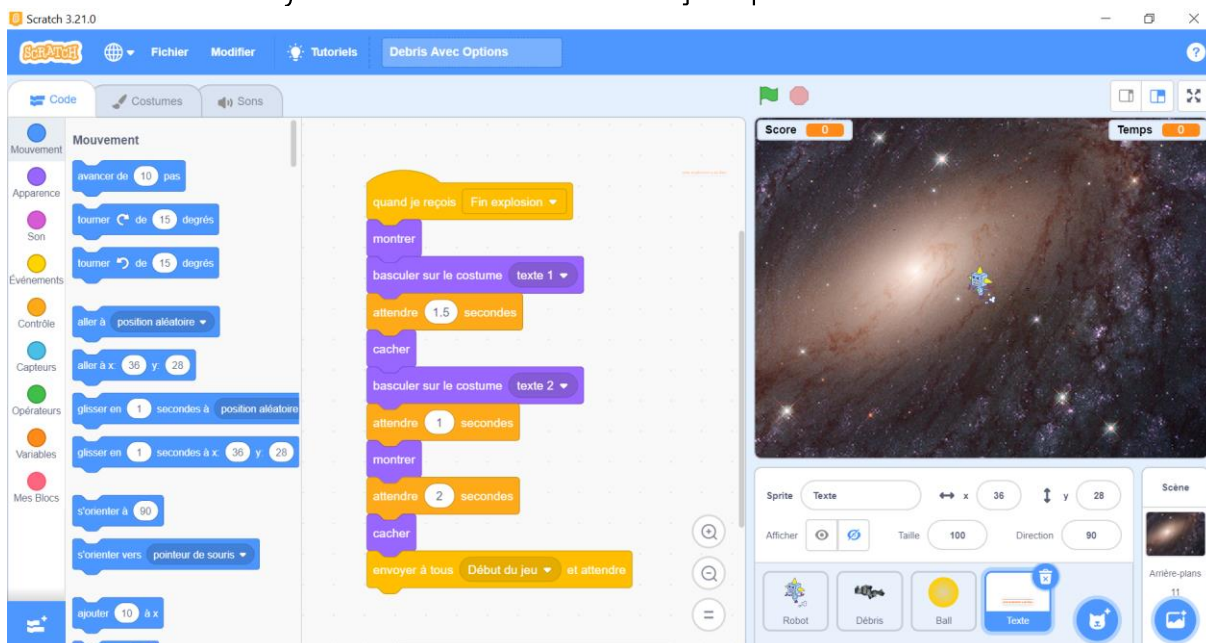
- Une animation avec un texte d'introduction

- Créer un sprite « Texte » en choisissant l'option peindre en faisant par exemple deux costumes (Texte 1: Une explosion a eu lieu et Texte 2: Ramassez le plus de débris dans le temps imparti)





- Créer la séquence suivante pour le code du sprite texte qui consiste à montrer le costume texte 1 du sprite texte pour le faire apparaître à l'écran puis le second.
- Faire commencer la séquence avec un bloc « quand je reçois Fin explosion » afin que l'animation commence juste après celle de l'explosion
- Terminer avec un bloc « envoyer à tous « Début du jeu » et attendre
- Remplacer chaque bloc « quand le drapeau vert est cliqué » par « quand je reçois Début de jeu » pour les autres sprites (robot et débris) afin de faire commencer le jeu après le texte d'introduction



Pour aller plus loin, vous pouvez également laisser les jeunes donner libre cours à leur imagination en ajoutant des effets sonores ou visuels (à la fin par exemple), en modifiant les vitesses de déplacements des sprites, en ajoutant d'autres débris...