

Niveau : 3ème

Un jeu pour se remettre doucement à Scratch

Matériel : PC/Mac, logiciel Scratch ou mBlock

Durée : 2 séances

Notions travaillées et Compétences du socle associées

Pré-requis : Bases de Scratch

Evaluation : sur table et démo

TECH	1.4 : Participer à l'organisation et au déroulement de projets.
TECH	3.3 : Présenter à l'oral et à l'aide de supports numériques multimédia des solutions techniques au moment de leur réalisation.
TECH	4.2 : Appliquer les principes élémentaires de l'algorithme et du codage à la résolution d'un problème.
TECH	5.1 : Simuler numériquement la structure et/ou le comportement d'un objet.
TECH	5.2 : Organiser, structurer et stocker des ressources numériques.

Ressources :
fredtechnocollege.org
« Scratch », Réviser »

ASTEROID Attack !!

Je te propose de construire un jeu de plateforme à un seul joueur. L'objectif sera de sauver le plus longtemps possible la planète **Sbork** de l'attaque de méchants envahisseurs lançant des astéroïdes (et même parfois des pastèques !).

Le joueur est le héros aux commandes de sa fusée, il pilotera à l'aide des touches de direction et tirera des missiles à l'aide de la barre d'espace.

Ce jeu sera construit en plusieurs étapes, progressives et autant de parties (1 à 5).

- Tu auras au final **trois lutins** différents...et trois scripts leur correspondant.
- Tous **les scripts démarreront ensemble lorsque le drapeau vert sera cliqué.**
- A chaque fin d'étape, fait vérifier ton travail avant de passer à la suite.



Si tu n'y arrives (vraiment) pas, demande un « coup de pouce » à ton prof pour te débloquent...mais n'en abuse pas, c'est en te heurtant toi-même à ces difficultés que tu apprendras à les dépasser (eh oui, c'est comme la vie...)



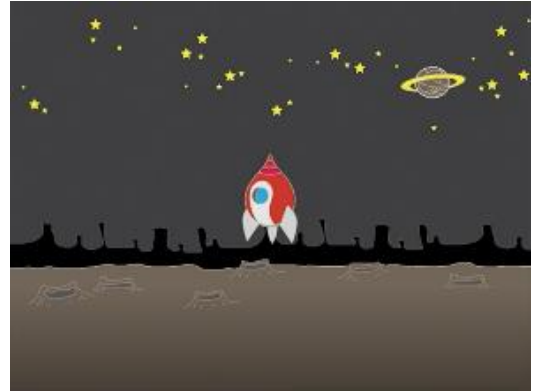
1ère partie

Crée le décor et déplace ta fusée en xy

1-Crée le décor :

Choisis un décor de ton choix parmi ceux proposés dans Scratch, celui de l'illustration n'est qu'une possibilité parmi d'autres. Ta fusée sera également choisie parmi les lutins disponibles de Scratch.

C'est fait ? Passe à la suite...



2- Anime la fusée à l'aide du clavier :

Afin de contrôler la fusée, on souhaite utiliser les quatre flèches du clavier pour la diriger dans les quatre directions.



Code le script utilisant les commande « Ajouter à x » et « Ajouter à y » si les touches flèches sont enfoncées :

Astuce : pour que la fusée reste verticale et ne tourne pas quand tu changes de direction, modifie l'option du lutin dans ses informations :



Quelques trucs pour programmer :

- Pour savoir quelle touche est enfoncée au clavier, il te faudra faire un test et agir en conséquence.
- Il te faudra faire la même chose pour les quatre flèches de direction.
- Sachant que ces quatre touches devront toujours être scrutées, il te faudra nécessairement répéter tout ton script grâce à la fonction « Répéter indéfiniment » :



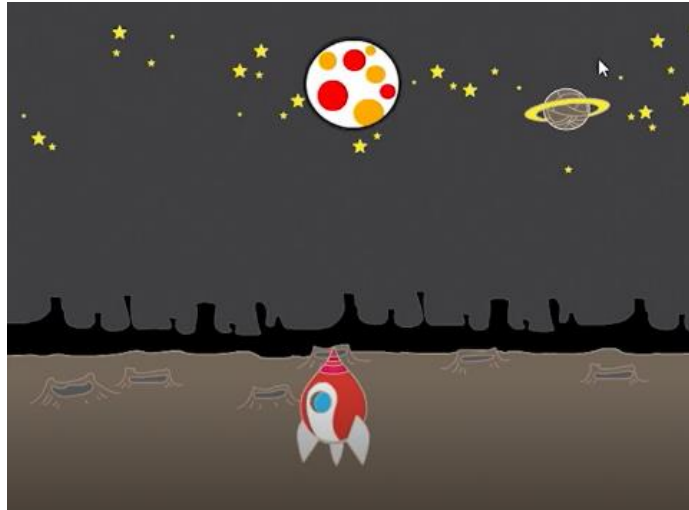
Fais vérifier tout ça par le prof...et passe à la suite

2^{ème} partie

Crée les astéroïdes qui chutent

Des astéroïdes doivent apparaître en haut de l'écran, chuter jusqu'en bas de façon continue.

Remarque : le plus astucieux consistera à créer le script **d'un seul astéroïde** puis, lorsqu'il fonctionnera parfaitement, de le **dupliquer** autant de fois que tu le souhaiteras.

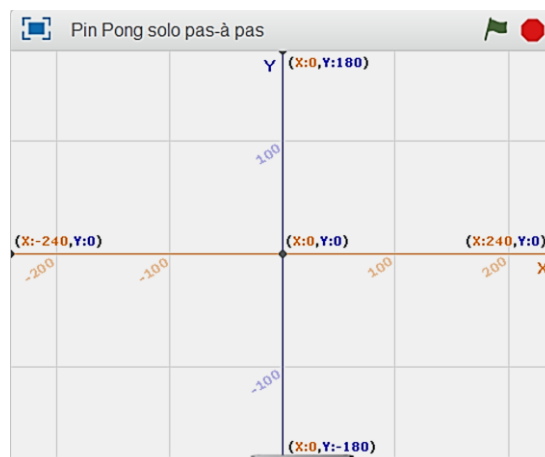


1 – Créer un astéroïde :

- Il te faut créer un nouveau lutin qui sera ton astéroïde, choisis-le dans la bibliothèque de Scratch.
- Ton lutin devra apparaître en haut de l'écran. Pour ce faire, tu pourras utiliser la fonction ci-contre qui place le lutin à une coordonnée de ton choix (ici, $x=106$ et $y=-152$ par exemple)



Rappel : Le repère de Scratch est un repère orthonormé ayant pour centre le point (0,0), c'est le centre de la fenêtre « scène ».



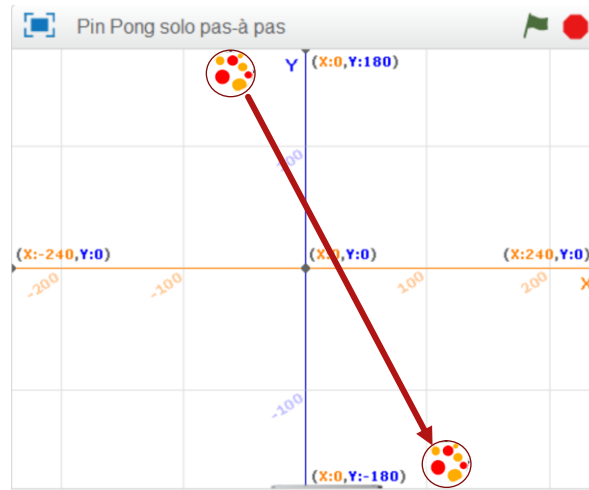
Si tu veux que ton astéroïde parte du haut, il te faudra donc choisir un « y » aux environs de 170 ou 180...

2/ Faire glisser l'astéroïde du haut vers le bas :

- Il y a plusieurs façons de déplacer un lutin, tu connais déjà la commande « **Avancer de...** » mais tu peux également utiliser la fonction ci-contre qui fera « glisser » ton lutin jusqu'à $x=-23$ et $y=-143$ (c'est un exemple) en 1 seconde -mais tu pourras changer cette valeur-.



A ce stade, ton astéroïde apparaît en haut et « glisse » jusqu'en bas...mais il fait toujours le même parcours et ne le fait qu'une seule fois, ça n'est pas très drôle 😞



3/ Faire partir l'astéroïde de n'importe où en haut et le faire arriver n'importe où en bas

Le jeu sera plus intéressant si on ne sait pas d'où part l'astéroïde et où il arrivera. On va utiliser la fonction « **nombre aléatoire** », très utile dans les jeux.

Exemple : si l'on utilise la commande suivante, alors le lutin se placera toujours à l'altitude $y=0$ mais le « x » sera choisi au hasard entre $x=-50$ et $x=60$.



Grâce à cette fonction, tu ajoute une part d'aléatoire dans ton jeu qui va devenir ainsi bien plus intéressant.

Tu pourras bien sûr en faire de même pour le « glisser » de telle manière que l'astéroïde arrive n'importe où en bas de la scène (et non toujours au même endroit).

4/ Faire apparaître de nouveaux astéroïdes indéfiniment :

Il te suffira d'inclure ton script dans une boucle infinie :



Ça fonctionne ? Il ne reste plus qu'à dupliquer ton script une ou deux fois pour rendre le jeu plus intéressant -mais plus difficile-.

Fais vérifier tout ça par le prof...et passe à la suite

3^{ème} partie

Si un astéroïde te touche...le jeu se termine

Tu peux maintenant déplacer ta fusée en x et y, des astéroïdes tombent de n'importe où et descendent...il te reste à les éviter faute de quoi ta fusée sera détruite et Sbork ira à sa perte...

1/ Comment savoir si un astéroïde a touché la fusée ?

Il y a plusieurs façons de faire, comme souvent en Scratch, en voici une :

Utiliser le test « ... touché ? » :

Scratch possède un test intégré très performant permettant de savoir si le lutin que tu programmes touche (ou non) un autre lutin (tu peux même détecter la flèche de la souris, un bord...).



Ainsi, dans le script de ton astéroïde, il te suffira de tester **si** tu touches la fusée ou pas ; si tu la touches, le jeu se termine sinon... le jeu continue et les astéroïdes continuent de tomber. Voici les blocs nécessaires :



Tu peux même écrire ce processus de test indépendamment des autres, il « tournera » en permanence dès qu'on lancera le jeu et ce indépendamment de ce qui se passe par ailleurs.

Je te laisse trouver ce qu'il faudra mettre dedans...



4ème partie

La fusée se défend et tire

Pour constituer tes munitions, tu auras besoin ici de créer un nouveau « lutin » (ici c'est une boule verte) qui partira de la fusée et montera vers le haut de l'écran (flèche jaune ci-dessous) pour abattre les astéroïdes avant qu'ils ne te touchent.



Cette « boule verte » devra partir de la pointe de la fusée et monter tout droit lorsque tu appuieras sur la « barre d'espace ».

1/ Comment faire partir un lutin depuis la fusée ?

Dans l'absolu, ç'est compliqué puisque la fusée bouge tout le temps... mais Scratch a déjà tout prévu avec l'instruction « **Aller à x=..., y=...** » et en utilisant les coordonnées de la fusée elle-même « **position x** » et « **position y** » :



Cette commande va énormément nous simplifier la vie puisque le lutin « balle verte » va maintenant suivre la fusée dans son déplacement... c'est magique. Essaie !

Programme : Si tu appuies sur la barre d'espace, alors la boule verte apparaît sur la fusée (« **Montrer** ») et monte en **répétant** le mouvement **jusqu'à** atteindre le haut de la fenêtre ou elle disparaît (« **Cacher** »). Il n'y a plus qu'à l'écrire, la structure sera la suivante... à toi de remplir ce qui manque...



5ème et dernière partie

Compter le temps et marquer les points

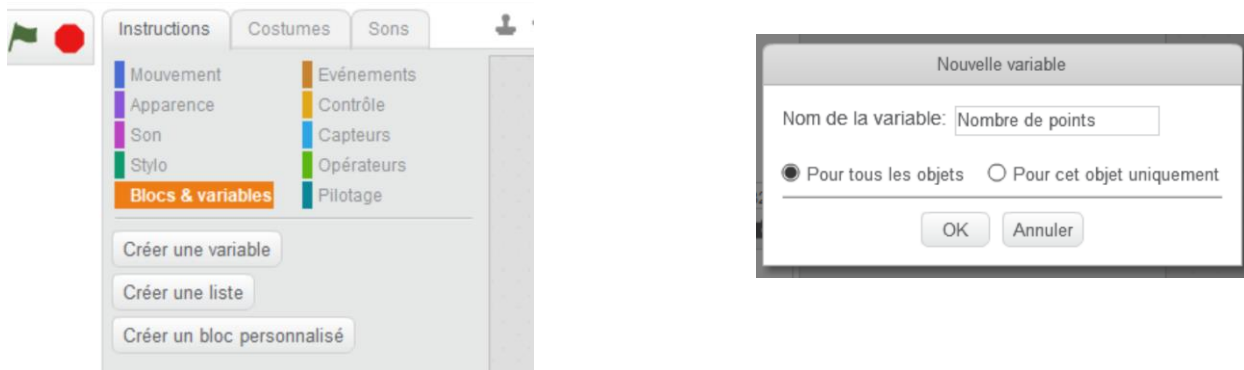
Pour compléter le jeu, il serait intéressant d'avoir le temps qui passe et le nombre d'astéroïdes abattus. Nous allons voir comment faire ça.

1/ Créer les variables « **Nombre de points** » et « **temps** » :

Une « variable est une grandeur qui varie (si !) : le temps évolue d'1s seconde toutes les secondes (si !), et tu marques un point à chaque fois que tu abats un astéroïde (promis !)

Les grandeurs « temps » et « nombre de points n'existant pas dans Scratch, c'est à toi de les **créer**, de les **initialiser** et de les modifier quand il le faut.

Clique sur la palette « **Blocs et variables** » puis sur « **Créer une variable** » et appelle ta première variable « **Nombre de points** » :



Tu verras alors apparaître de nouvelles commandes associées à ta nouvelle variable :

Tu pourras insérer ces nouvelles commandes dans tes scripts.



Ainsi, dans le script de ton astéroïde, un des process sera chargé de compter les points : si une boule touche ton astéroïde, tu rajouteras un point à ta variable « Nombre de points ». Je te laisse compléter le script ci-dessous...



Pense à initialiser le nombre de points à 0 en début de jeu, juste après le « Clic sur drapeau vert »

2/ Compter le temps qui passe :

C'est la même astuce : crée une variable « **Temps** », met-la à 0 au tout début puis ajoute lui « 1 » à chaque seconde qui passe.

Tu auras intérêt à en faire un petit process indépendant, un peu comme ça (je te laisse remplir...)



Ici il faudra attendre une seconde, ajouter 1 » à ta variable « Temps », ça sera répété indéfiniment

Te voici avec un jeu parfaitement fonctionnel que tu pourras encore compléter et améliorer si tu le souhaites : bravo !

