

Belgium



Avec le soutien financier de



Cache-cache avec la Terre

Fiche pour l'enseignant

La Terre a disparu... Comment la retrouver ? En écrivant le programme d'un mini-jeu en langage Scratch, les élèves s'exercent à manipuler les notions de base de ce langage de programmation par blocs (variables, boucles, conditions, calcul de coordonnées).



Programmation

Age : 10-14 ans

Cache-cache avec la terre

Objectif

- ✓ Programmer un jeu de « cache-cache » : le but est de retrouver la planète Terre dans l'univers par essai-erreur en cliquant sur l'écran pour s'en approcher

Notions abordées : Programmation, instructions, mathématiques

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

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

Dispositif pédagogique : 1 ordinateur par jeune pour un groupe de 12 jeunes ou 1 ordinateur par binôme pour un groupe 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 »](#)
- [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 Scientothèque. 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 – 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. Il est possible (mais non obligatoire) de profiter de cette activité pour faire un rappel sur théorème de Pythagore, suivant le niveau des jeunes.

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.

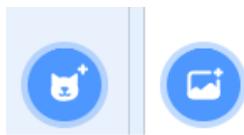
2 – Jeu de base

Le principe du jeu que les jeunes vont devoir programmer est le suivant : la planète Terre disparaît de l'écran ; le joueur est invité à la retrouver par essai-erreur en cliquant sur l'écran pour évaluer la distance avec la Terre et s'en rapprocher.

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

a) Disparition du sprite Terre

- Choisir l'arrière-plan « Galaxy » (disponible dans la base de données de Scratch)
- Remplacer le sprite par défaut (personnage de chat) par le sprite « earth » (inclus dans la base de données de Scratch)



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

- Construire ensuite la séquence de blocs dans l'onglet Code permettant au sprite Terre d'apparaître au centre de l'écran (« aller à x:0 y:0 ») et disparaître pour se positionner de manière aléatoire (« aller à position aléatoire »).

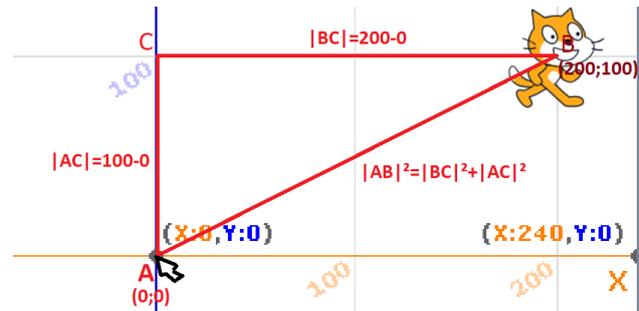


Pour rappel, les blocs sont organisés en catégorie. La couleur des blocs dépend de la catégorie à laquelle ils appartiennent. Si vous ne trouvez pas un bloc, il suffit de repérer sa couleur et de chercher dans la bibliothèque la catégorie correspondante. De plus, les catégories ont des noms assez parlants, par exemple pour déplacer un sprite, il semble logique d'aller chercher dans la catégorie « mouvement ».

b) Le calcul de la distance

Pour retrouver le sprite Terre, le joueur est invité à cliquer sur l'écran. En fait, nous allons calculer la distance entre l'endroit où la souris clique et la position du sprite Terre.

Pour rappel, les Sprites se déplacent dans un repère orthonormé. Par exemple sur l'image ci-contre, le sprite se trouve à la position $(x:200; y:100)$. La distance séparant le sprite et la position de la souris $(x:0; y:0)$ correspond à l'hypoténuse d'un triangle ABC, dont les côtés AC et BC valent la différence d'ordonnées et d'abscisses respectivement entre la position pointée par la souris et celle du centre du sprite.



- Créer deux variables a et b (correspondant aux longueurs des côtés AC et BC) (catégorie Variables – puce orange foncée)



- Soustraire les coordonnées du sprite Terre et de la position pointée par la souris en faisant correspondre les résultats aux variables a et b (catégorie Contrôle – puce orange claire)



- Créer une variable « distance »
- Attribuer à la variable « distance » la valeur grâce au bloc se trouvant dans la catégorie Opérateurs (puce verte) telle qu'indiquée sur la figure ci-dessous :

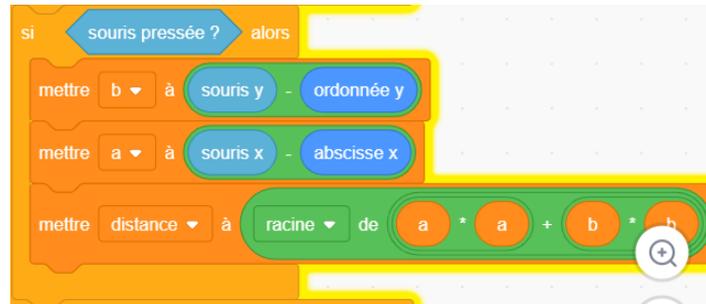


Les blocs ci-dessus traduisent bien la formule du théorème de Pythagore :

$$\text{Distance} = \sqrt{[(a*a)+(b*b)]}$$

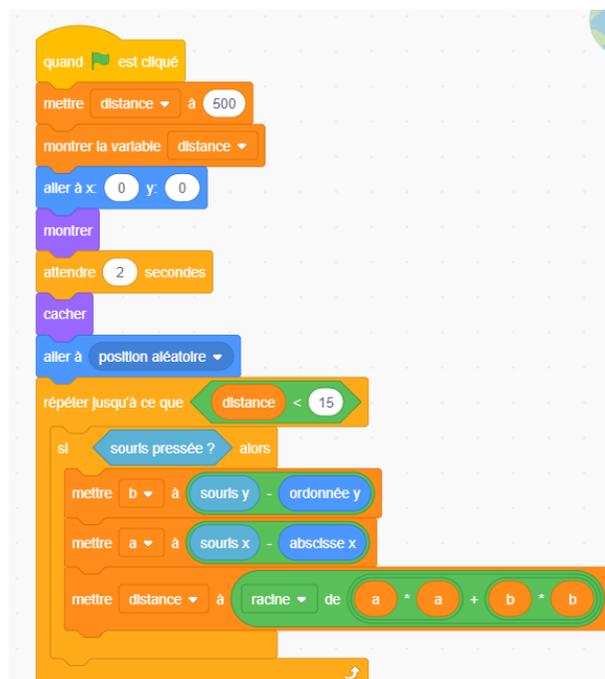
Si on imbriquait simplement cette série de blocs dans le code du sprite Terre, la distance entre le pointage de la souris et la position du sprite Terre serait calculée seulement au lancement du programme. Nous allons donc dans un premier temps définir quand la calculer (à chaque fois que la souris est pressée).

- Insérer la séquence de bloc venant d'être construite dans un bloc d'instruction conditionnelle «si...alors» (catégorie Contrôle) avec comme condition «souris pressée ? » (Catégorie Capteurs – puce bleue claire)



- Insérer le tout dans une boucle «répétez jusqu'à ce que distance < 15 ». Nous avons choisi 15 au lieu de « distance = 0 » pour ne pas avoir à cliquer exactement sur le pixel correspondant au centre du sprite Terre recherché. Il s'agit d'une marge donc.
- Initialiser la valeur de la variable « Distance » en lui attribuant une valeur élevée en début de programme, 500 par exemple. En effet, sans cela, si la variable distance était inférieure à 15 au moment de lancer le code, on passerait directement dans l'animation de victoire.
- Ajouter également un bloc « montrer la variable « Distance » »

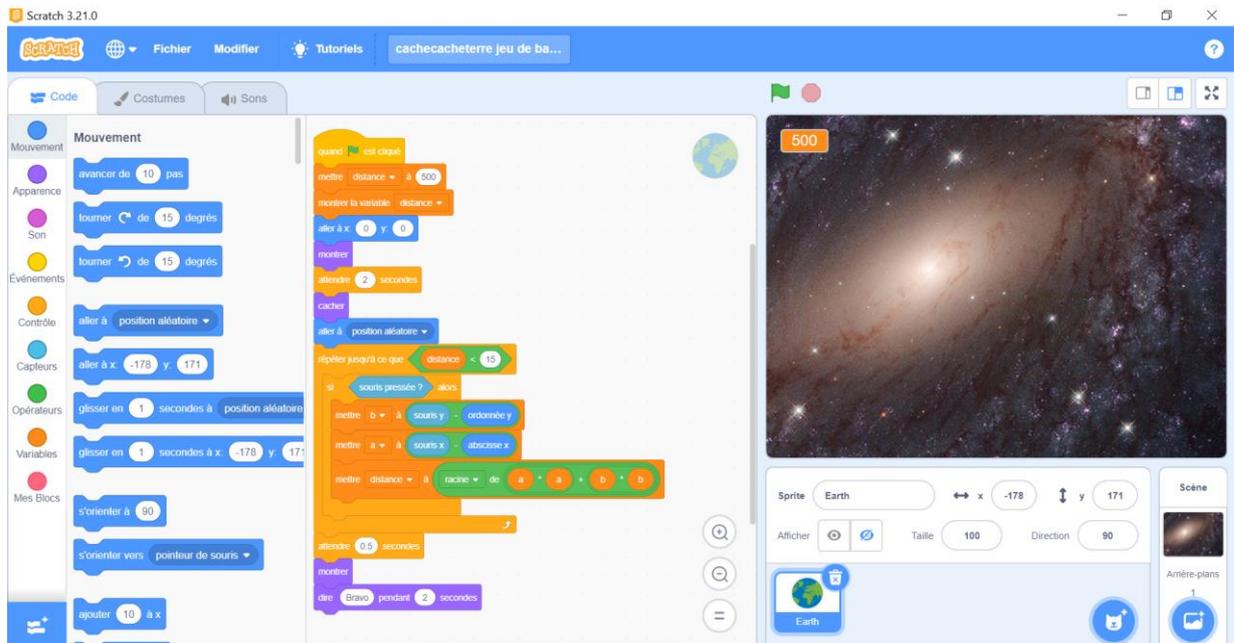
Vous devriez obtenir quelque chose qui ressemble à ceci :



c) La fin du jeu

Le code du paragraphe précédent s'arrête une fois le sprite Terre trouvé par le joueur. Il reste maintenant à voir ce qu'il se passe ensuite.

- Placer le sprite Terre au centre de l'écran (« aller à x:0 y:0 »)
- Le faire réapparaître
- Faire dire « Bravo » grâce au bloc « dire » (Catégorie Apparence – puce violette)



3 – Options supplémentaires du jeu

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

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

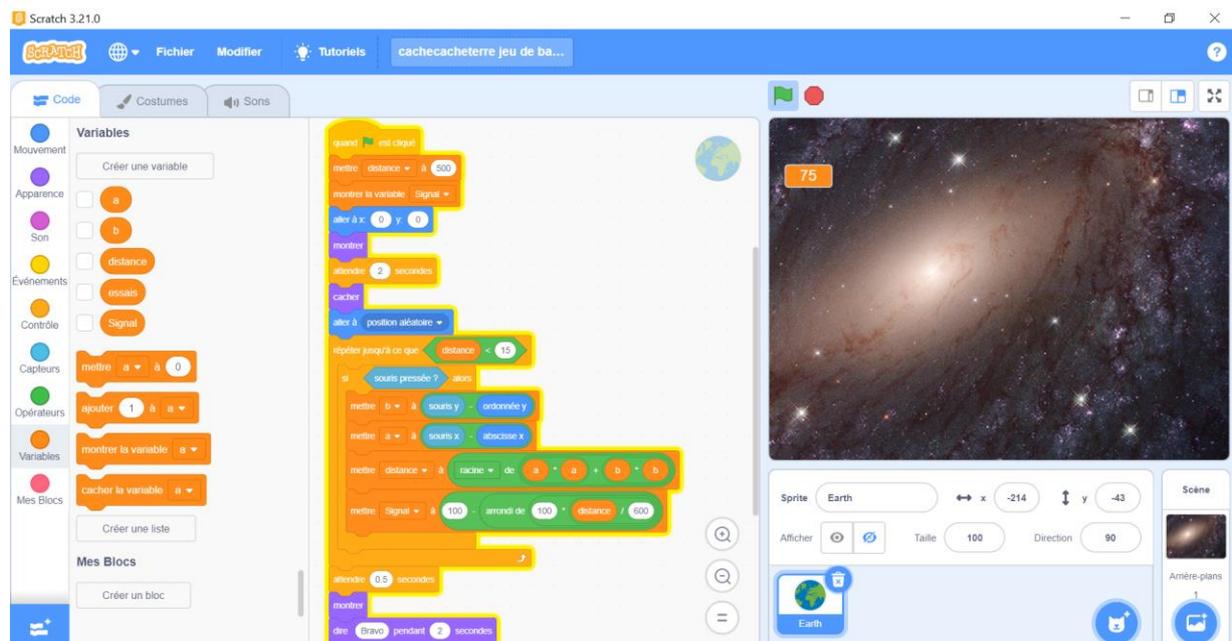
a) Convertir la distance en « signal »

Plutôt que de montrer la valeur de la distance entre le sprite Terre et la position pointée par la soirée, il peut être plus lisible de montrer la valeur d'un signal exprimé en pourcentage. Plus le joueur se trouvera proche de la Terre, plus le Signal approchera la valeur de 100.

- Créer une variable Signal
- Diviser la valeur de la « distance » par sa valeur maximum (il s'agit de la diagonale de l'écran, qui fait 480 sur 360 pixels, la distance maximale est donc la racine carrée de $480^2 + 360^2$, soit 600) et multiplier par 100.
- Soustraire ce résultat à 100
- Attribuer la valeur du signal
- Ajouter le bloc ainsi construit dans le code du sprite Terre (sous le bloc correspondant à la formule de la distance)
- Montrer la variable Signal (à la place de la distance)

Ainsi si on clique sur le sprite Terre, on obtiendra une distance de 0, ce qui donnera un signal de 100% ($100 - 0 = 100$)

Remarque : le bloc « arrondi » n'est pas obligatoire, mais rend l'interface plus lisible (par défaut le programme affiche 6 décimales, ce qui allonge la case).



b) Limiter le nombre d'essais

Il est possible d'introduire un nombre limité d'essais pour ajouter un peu de challenge au jeu.

- Créer une variable « Essais »
- Incorporer un bloc « ajouter -1 à essais » dans la boucle « si souris pressée » pour augmenter le compteur à chaque clic de souris.
- Ajouter un bloc « attendre jusqu'à ce que non souris pressée ? » pour faire en sorte que la variable Essais ne baisse de 1 qu'une fois la souris relâchée. Sans ce bloc, le compteur d'essais défilerait tant que la souris n'est pas relâchée.

```

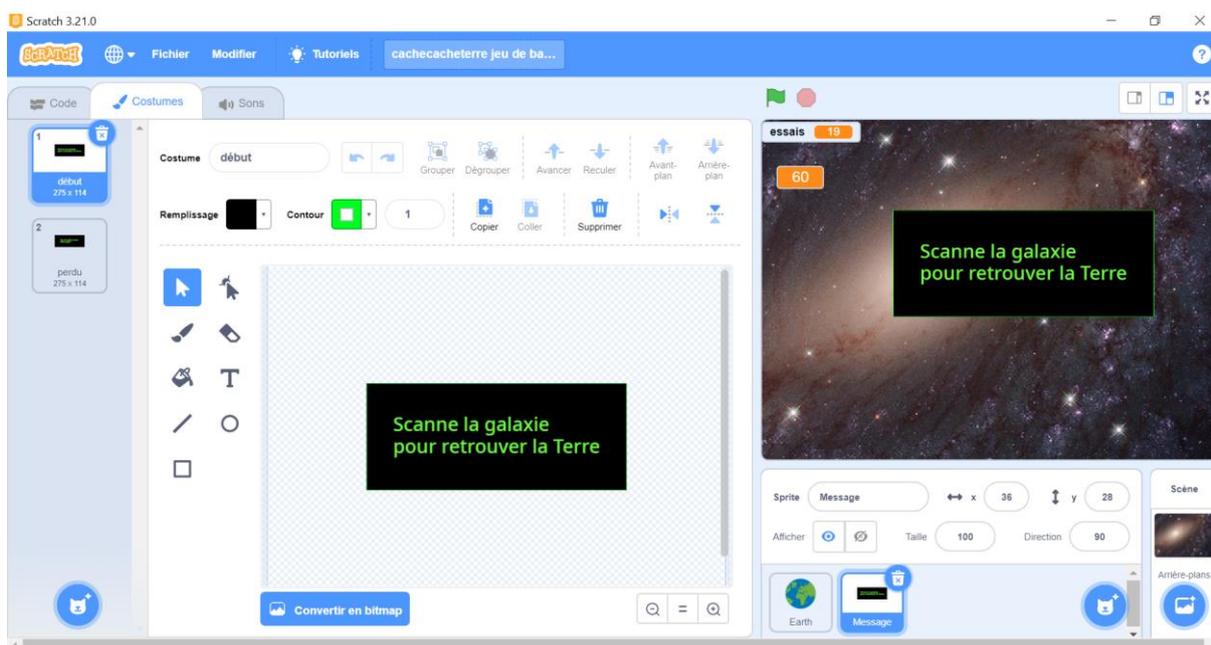
    quand cliqué
    mettre distance à 500
    mettre essais à 20
    montrer la variable Signal
    montrer la variable essais
    aller à x: 0 y: 0
    montrer
    attendre 2 secondes
    cacher
    aller à position aléatoire
    répéter jusqu'à ce que distance < 15
    si souris pressée ? alors
        envoyer à tous dommage
        mettre b à souris y - ordonnée y
        mettre a à souris x - abscisse x
        mettre distance à racine de a * a + b * b
        mettre Signal à 100 - arrondi de 100 * distance / 800
        ajouter -1 à essais
        attendre jusqu'à ce que non souris pressée ?
    attendre 0.5 secondes
    montrer
    dire Bravo pendant 2 secondes
  
```

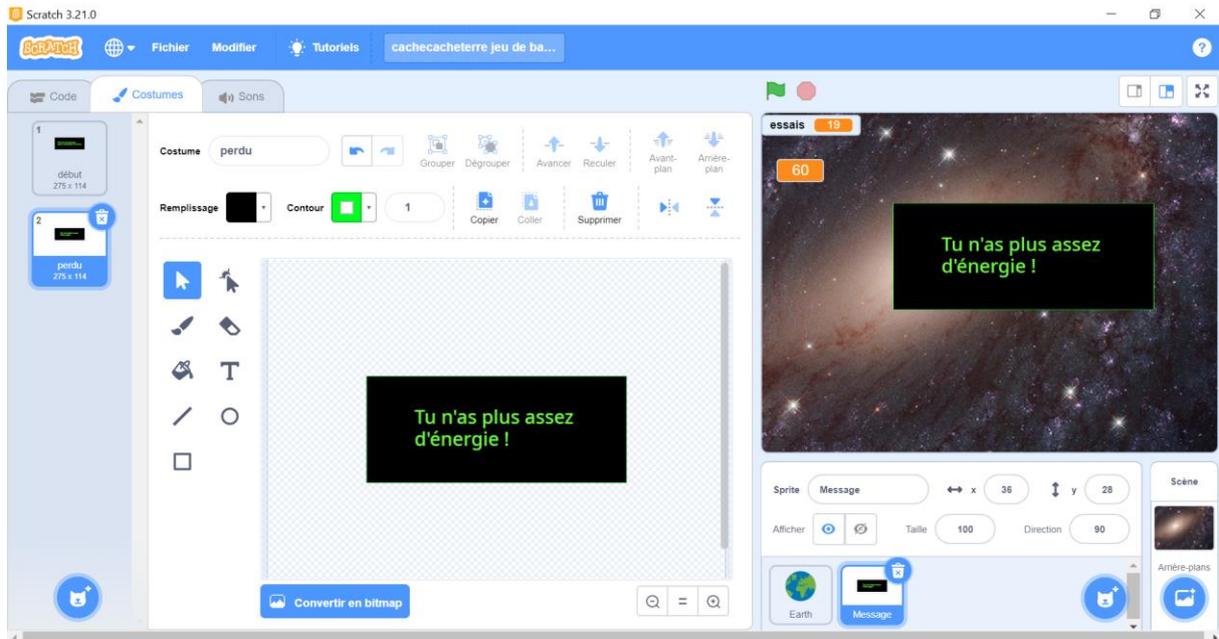
- Ajouter la nouvelle séquence ci-contre qui permettra d'envoyer le message « Dommage » (en le créant grâce au bloc « envoyer à tous ») une fois qu'il y aura eu 20 essais
- Initialiser la variable Essais à la valeur 20 et la montrer au début de la séquence
- Envoyer le message dommage à tous les sprites



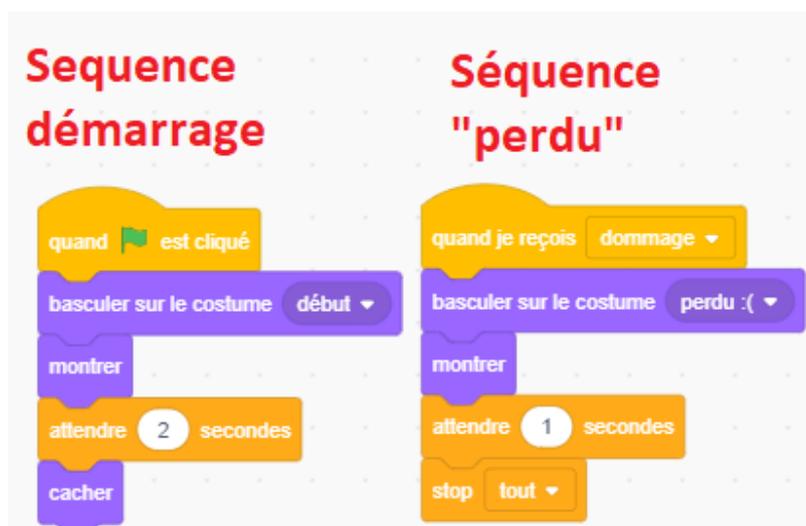
Une fois le nombre d'essais maximum défini comme ci-dessus, vous pouvez créer un nouveau sprite « Message » avec deux costumes : un message de début de jeu et un message en cas de défaite du joueur.

- Choisir l'option « Peindre » pour créer le sprite Message et ses deux costumes

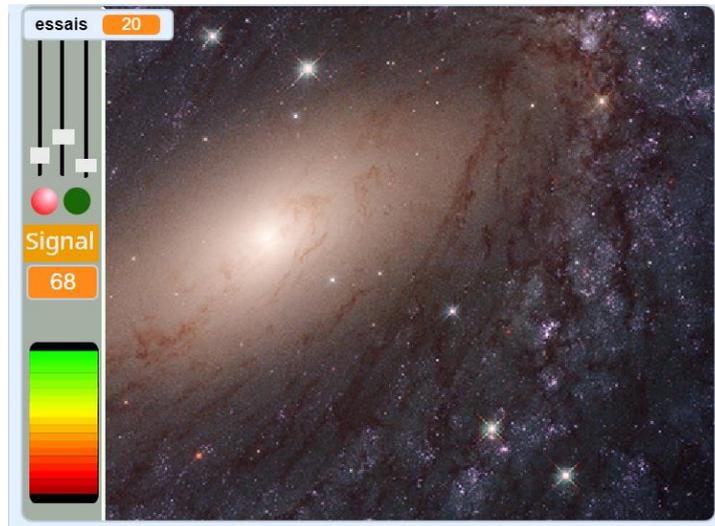




Le code du sprite Message est assez simple :



c) Personnaliser l'interface



Il est possible d'ajouter des éléments visuels pour personnaliser l'interface de jeu.

Pour la fenêtre grise sur la gauche de l'écran, nous avons simplement dessiné un panneau avec des curseurs sur la scène et intégré la variable Signal au-dessus.

Il est également possible de rajouter d'autres sprites :

- la jauge de vie (en bas à gauche) pour laquelle il a fallu créer un costume pour chaque niveau de vie, soit 20 étant donné que le joueur a droit à 20 essais. Il suffit ensuite de coder le sprite tel qu'il bascule sur les costumes dont le numéro correspond au nombre d'essais restants.



- un cercle vert qui s'anime lorsque le joueur clique sur l'écran avec le code correspondant ci-dessous :



```

quand je reçois Clic
aller à x: souris x y: souris y
mettre l'effet fantôme à 0
mettre la taille à 0 % de la taille initiale
montrer
répéter 20 fois
    ajouter 5 à la taille
    ajouter 5 à l'effet fantôme
cacher
    
```

Pour aller plus loin, il est possible de laisser libre cours à la créativité des jeunes en les invitant à ajouter et coder autant d'éléments visuels qu'ils le souhaitent.

Enfin, vous pouvez aussi ajouter des animations en insérant par exemple des blocs dans le code du sprite Terre comme présenté ci-dessous :

- la Terre qui tourne deux fois alternativement vers la droite et la gauche au début du jeu

```

répéter 2 fois
    attendre 1 secondes
    tourner de 15 degrés
    attendre 1 secondes
    tourner de 15 degrés
    
```

- la Terre qui disparaît progressivement (plutôt que d'être simplement cachée)

```

répéter 5 fois
    ajouter -10 à la taille
    attendre 0.1 secondes
    
```

La séquence de blocs finale pour le sprite Terre est représentée ci-dessous :

```

    quand est cliqué
    mettre distance à 500
    mettre essais à 20
    montrer la variable essais
    montrer la variable Signal
    mettre la taille à 100 % de la taille initiale
    aller à x: 0 y: 0
    montrer
    répéter 2 fois
    attendre 1 secondes
    tourner de 15 degrés
    attendre 1 secondes
    tourner de 15 degrés
    répéter 5 fois
    ajouter -10 à la taille
    attendre 0.1 secondes
    cacher

    quand est cliqué
    attendre jusqu'à ce que essais = 0
    envoyer à tous dommage

    mettre x à nombre aléatoire entre -180 et 240
    mettre y à nombre aléatoire entre -180 et 180
    répéter jusqu'à ce que distance < 15
    si souris pressée ? alors
    envoyer à tous Clic
    mettre b à souris y - ordonnée y
    mettre a à souris x - abscisse x
    mettre distance à racine de a * a + b * b
    mettre Signal à 100 - arrondi de 100 * distance / 600
    ajouter -1 à essais
    attendre jusqu'à ce que non souris pressée ?
    attendre 0.5 secondes
    montrer
    dire Bravo! pendant 2 secondes
    
```