

**MOONHACK 2021**

# Héros de l'énergie domestique

VERSION FRANÇAISE

Créé par CODE CLUB AUSTRALIE et supporté  
par TELSTRA FOUNDATION



/ AUSTRALIA



POWERED BY  
TELSTRA  
FOUNDATION

Soumets ton projet à [MOONHACK.COM](https://moonhack.com)  
pour être comptabilisé





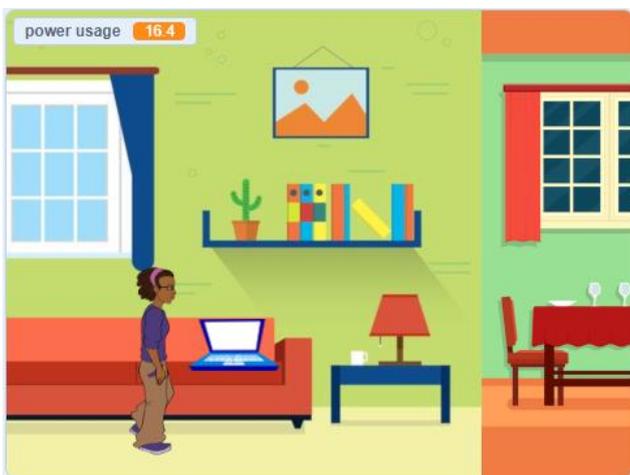
## Héros de l'énergie domestique

Nous utilisons tous beaucoup d'électricité dans nos maisons. Un des moyens les plus simples pour réduire notre consommation électrique consiste à éteindre les appareils que nous n'utilisons pas.

### INTRODUCTION

#### Ce que tu vas faire

Tu vas créer un jeu utilisant le défilement parallaxe pour aider Avery à minimiser sa consommation électrique en éteignant ses appareils.



#### Ce que tu vas apprendre

- Défilement parallaxe (déplacement de l'arrière-plan autour du personnage)

*Le Code Club Australie reconnaît les gardiens traditionnels de la terre de toute l'Australie et leur lien permanent avec la terre, les cultures et les communautés. Les propriétaires traditionnels d'Australie sont les premiers innovateurs du monde.*

#### Ce dont tu as besoin

##### MATÉRIEL

Un ordinateur capable d'exécuter Scratch 3. Cette leçon utilise la barre d'espace et peut donc ne pas être adaptée pour une tablette.

##### LOGICIEL

Scratch 3 : Soit en ligne [rpf.io/scratchon](https://scratch.mit.edu/projects/473116991/) soit hors ligne [rpf.io/scratchoff](https://scratch.mit.edu/projects/473116991/)

##### TÉLÉCHARGEMENTS

Projet de démarrage hors ligne [bit.ly/powerhero](https://bit.ly/powerhero)

#### Informations supplémentaires pour les éducateurs

Voici le lien du projet déjà terminé - <https://scratch.mit.edu/projects/473116991/>

Consultez notre blog sur ce projet avec des conseils, des programmes et du matériel de support à l'adresse suivante : [medium.com/@codeclubau](https://medium.com/@codeclubau)

# 1. UNE MAISON QUI SE DÉPLACE

Il existe différentes façons de montrer la marche dans un jeu. La première consiste à faire bouger le sprite dans la scène ; une autre consiste à faire bouger la scène autour du sprite. Cette deuxième option est appelée 'défilement parallaxe' et c'est ce que nous allons utiliser dans ce jeu.

Ouvre ton projet de départ : [bit.ly/powerhero](https://bit.ly/powerhero)

Modifie la langue de ton projet au niveau du globe que tu vois en haut et à gauche de ton écran.



D'abord, nous devons préparer l'arrière-plan pour qu'il bouge alors que Avery reste immobile. Cela va donner une impression de déplacement.



Clique sur le sprite appelé HOUSE et ajoute le code suivant :

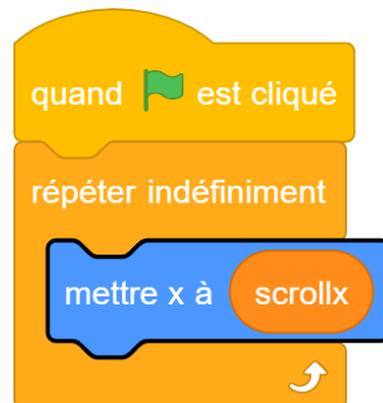
Le code va s'exécuter sans arrêt dans une boucle.



Ensuite, nous allons ajouter du code dans la boucle pour faire bouger notre maison :

La variable 'scrollx' garde une trace de la distance parcourue par Avery sur l'axe des x (de gauche à droite).

En bougeant l'arrière-plan dans la direction opposée des déplacements de Avery, ce sera comme si c'était elle qui bougeait.



A chaque fois que nous jouons au jeu, nous devons réinitialiser notre variable 'scrollx' pour que le sprite HOUSE revienne à sa position de départ.

Ajoute ce code :



## 2. EN MARCHÉ

Maintenant que la maison est prête à bouger avec l'aide de la variable 'scrollx', allons-y ! Nous voulons que Avery marche vers la droite quand nous appuyons sur la touche 'flèche droite'.



Sélectionne le sprite 'Avery'.



Change la valeur de 'scrollx' lorsque la touche 'flèche droite' est enfoncée.

Teste ton projet.

Lorsque tu appuies sur la touche 'flèche droite', l'arrière-plan doit se déplacer vers la gauche, ce qui donne l'impression que Avery se déplace vers la droite.



Elle ne fait que glisser le long de l'arrière-plan. Donnons maintenant l'impression qu'elle marche.

Avery a plusieurs costumes qui, lorsqu'ils sont affichés les uns après les autres, donnent l'impression qu'elle marche.

Ajoute le block 'costume suivant' à ton code :

Teste à nouveau ton programme en appuyant sur la touche 'flèche droite'.



Avery marche vite !

Ralentissons-la en ajoutant un bloc d'attente :

Teste ton projet à nouveau. Avery devrait marcher normalement.



### 3. PLUS D'ARRIÈRE-PLANS

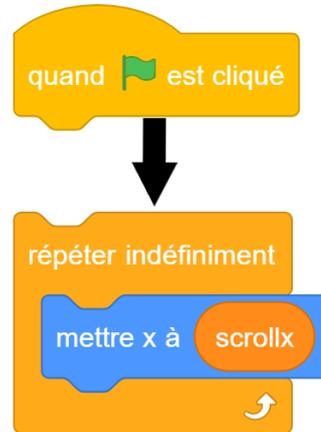
Avery ne peut pas marcher très loin car il n'y a qu'un seul arrière-plan. Ajoutons plus d'arrière-plans pour qu'elle puisse marcher sur une plus longue distance.



Clique sur le sprite HOUSE



Sépare le bloc 'lorsque le drapeau vert est cliqué' de la boucle 'répéter indéfiniment' :



Sous le bloc 'quand le drapeau vert est cliqué', ajoute du code pour définir les conditions initiales :



Ajoute une boucle qui va créer un clone pour chaque costume de la maison :



Au-dessus du code que tu as détaché, ajoute un bloc 'quand je commence comme un clone' :



Parce qu'il y a des costumes différents qui constituent l'arrière-plan, nous devons les déplacer à l'écran en se basant sur 'scrollx' mais chaque costume doit être décalé de 480 pixels par rapport au précédent (480 pixels est la largeur d'un costume).



```
quand je commence comme un clone
  répéter indéfiniment
    mettre x à scrollx + numéro du costume * 480
```

Ajoute ce code. Vérifie bien que ce soit correct

Si nous exécutons notre programme maintenant, les sprites de la maison n'apparaîtront pas, car nous avons dit à Scratch de les cacher. Mais il ne faut pas simplement montrer tous les sprites, car Scratch ne les laisse pas sortir complètement de la scène.

Nous pouvons plutôt dire à Scratch que si la position x réelle du sprite est bien celle que nous voulons, alors on le montre sinon on le cache :

```
quand je commence comme un clone
  répéter indéfiniment
    mettre x à scrollx + numéro du costume * 480
    si absisse x = scrollx + numéro du costume * 480 alors
      montrer
    sinon
      cacher
```

*Astuce : Si tu ne souhaites pas réécrire du code, tu peux le dupliquer en cliquant dessus avec le bouton droit et en appuyant sur dupliquer. (ici le bloc +)*

Teste ton programme. Tu remarques que l'arrière-plan de la maison commence à droite de l'écran. Nous ne voulons pas ça, nous voulons démarrer avec le premier costume de la maison qui prend tout l'écran.

Pour cela, nous devons mettre la variable 'scrollx' à -480. (Souviens toi que 480 est la taille de notre sprite Maison en pixels)

Met à jour la valeur de départ de 'scrollx' en modifiant ce code :



```
quand le drapeau vert est cliqué
  mettre scrollx à -480
```

Teste ton projet à nouveau. L'arrière-plan devrait maintenant occuper tout l'écran lorsque tu appuies sur le drapeau vert !

## 4. UN ORDINATEUR MOBILE

Quand Avery marche, l'ordinateur ne bouge pas avec l'arrière-plan. On dirait qu'il flotte devant elle ! Nous allons faire bouger l'ordinateur avec l'arrière-plan, pour qu'Avery puisse l'approcher.

Le but du jeu est d'éteindre tous les appareils ménagers le plus vite possible. Lorsque le jeu démarre, l'ordinateur doit être allumé.



Clique sur le sprite de l'ordinateur et ajoute le code suivant :

```
quand [drapeau] est cliqué
  basculer sur le costume on ▼
```

L'ordinateur doit rester immobile dans l'arrière-plan, ce qui signifie qu'il doit se déplacer par rapport à Avery :

Teste ton programme.

L'ordinateur doit rester immobile alors qu'Avery se dirige vers lui. Que se passe-t-il quand il va en dehors de l'écran ? Il est entraîné sur le côté gauche de l'écran !

```
basculer sur le costume on ▼
  répéter indéfiniment
    mettre x à scrollx + 565
```

Nous devons ajouter du code qui va cacher l'ordinateur dès qu'il sort de l'écran. Pour cela, nous allons le cacher dès qu'il touche le bord :

Teste ton code à nouveau.

L'ordinateur devrait disparaître dès qu'il touche le bord de l'écran.

```
mettre x à scrollx + 565
  si touche le bord ? alors
    cacher
  sinon
    montrer
```

## 5. AVERY EST TRÈS ÉCONOME

Le but du jeu est d'éteindre les appareils, nous devons donc donner à Avery la capacité de les éteindre !

Nous allons utiliser la touche 'espace' pour éteindre l'ordinateur, mais Avery ne peut l'éteindre que si elle le touche.

Ajoute le code suivant au sprite de



l'ordinateur :



```
quand la touche espace est pressée
si touche le Avery ? alors
  [ ]
```

Nous voulons que Avery puisse allumer et éteindre l'ordinateur quand elle veut.

Tu remarqueras que le sprite de l'ordinateur a deux costumes : 'on' (en marche) et 'off' (éteint).

Nous pouvons simplement passer de l'un à l'autre en utilisant le bloc 'costume suivant' :

Teste ton projet.

Avery peut maintenant éteindre ou allumer l'ordinateur dès qu'elle est à proximité.



```
quand la touche espace est pressée
si touche le Avery ? alors
  costume suivant
  [ ]
```

Pour que ce jeu soit plus intéressant, nous voulons mesurer la consommation d'électricité et donner un score à battre.

Pour faire cela, nous voulons augmenter la mesure de la consommation électrique tant que l'ordinateur est allumé (Variable 'Power Usage').

Ajoute ces blocs de code au sprite de l'ordinateur :



Pourquoi augmenter la consommation électrique de 0.012 ? C'est le nombre de watt-heure qu'un ordinateur portable (50 watts) consomme en une seconde.

Teste ton projet.

La consommation d'énergie devrait augmenter lorsque l'ordinateur portable est allumé, mais pas s'il est éteint.

```
quand est cliqué
répéter indéfiniment
si nom du costume = on alors
  ajouter 0.012 à power usage
  attendre 1 secondes
```

## 6. PLUS D'APPAREILS

Jusqu'à présent, nous n'avons codé que notre ordinateur, mais nous avons également une lampe et une télévision, alors faisons en sorte qu'ils fonctionnent aussi.

Le code pour notre lampe sera très similaire à celui de notre ordinateur. Au lieu d'écrire tout le code, copions nos trois piles de code.

Nous pouvons le faire en cliquant et en faisant glisser chaque pile de code de notre sprite d'ordinateur vers notre sprite de lampe.

*Astuce : assure-toi de saisir le bloc supérieur pour chaque pile de code afin de tout copier.*

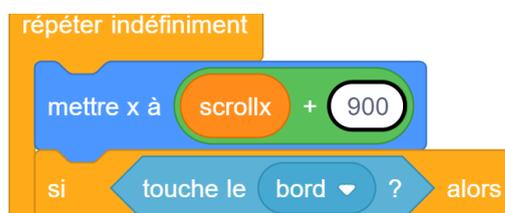


Une fois que tu as copié les trois piles, nous devons modifier quelques valeurs. Assure-toi que



le sprite de la lampe est sélectionné.

Nous voulons mettre à jour l'emplacement de la lampe, afin qu'elle ne soit pas au même endroit que l'ordinateur :

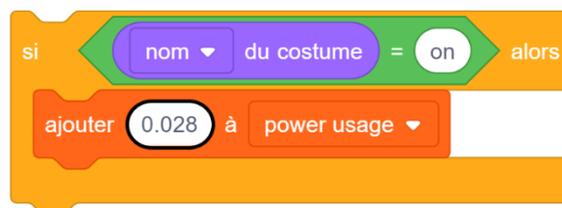


Nous voulons également modifier la puissance de consommation de la lampe.

Change la consommation d'énergie de la lampe à 0,028 (le nombre de watt-heure qu'une lampe de 100 W utilise par seconde) :

Teste ton projet.

Tu devrais maintenant avoir à la fois l'ordinateur et la lampe dans ton jeu.



## Défi

### Ajoute le téléviseur à ton jeu

- Maintenant que tu as ajouté la lampe, peux-tu faire la même chose pour ajouter le téléviseur ?
- *Astuce : tu devras changer la position de départ et la consommation d'énergie. En moyenne, un téléviseur LCD de 42 pouces consomme environ 120 W de puissance, soit 0,033 Watt-heure par seconde. Et il y a un bel endroit sur le banc de la cuisine avec une valeur x autour de 1200.*

## 7. GAME OVER

Pour rendre notre jeu stimulant, nous allons en faire une course contre la montre (ou, plus précisément, un Wattmètre). Nous allons donner à notre joueur une puissance maximale qu'il peut utiliser (nous lui donnerons 1 Watt-heure), et s'il en utilise plus, c'est GAME OVER !

1 Au début du jeu, nous voulons définir notre consommation d'énergie à 0, attendre jusqu'à ce que nous atteignons la quantité maximale de puissance, puis la partie est terminée.

Ajoute le code suivant au

sprite de la maison :



```
quand est cliqué
mettre power usage à 0
attendre jusqu'à ce que power usage > 1
envoyer à tous game over
```

2 Maintenant, clique sur le sprite 'Game Over' et ajoute un code qui réagira au message 'game over' :



```
quand je reçois game over
montrer
stop tout
```

3 Teste ton jeu.

Lorsque ta consommation d'énergie dépasse 1 watt-heure, est-ce que le message 'Game Over' apparaît ?

Que se passe-t-il lorsque tu redémarres le jeu ? Il est toujours là !

Alors, ajoutons du code pour masquer le sprite 'game over' lorsque le jeu commence :



```
quand est cliqué
cacher
```

# Défis

## Marcher à gauche

- Pour le moment, Avery ne peut marcher que vers la droite. Peux-tu la faire marcher vers la gauche ?
- *Astuce : Tu devras t'assurer qu'elle est tournée dans la direction qu'elle marche.*

## De l'électroménager plus efficace

- Peux-tu rendre le jeu plus facile en rendant les appareils plus économes ?
- Recherche la consommation d'énergie de différents appareils et trouve des alternatives plus économique à l'ordinateur, à la lampe et à la télévision. Tu peux peut-être aussi ajuster la consommation d'énergie maximale pour rendre le jeu plus difficile.
- *Astuce : pour obtenir le nombre de watts-heure par seconde, tu peux diviser le nombre de watts qu'un appareil utilise par 3600.*

## Défi avancé : Des visuels plus fluides

- Pour le moment, les appareils disparaissent dès qu'ils touchent le bord et n'apparaissent que si'ils ne touchent plus le bord. Cela signifie que les sprites disparaissent alors qu'ils devraient être à l'écran.
- Peux-tu changer le code pour que les appareils ne disparaissent pas avant qu'ils ne soient presque complètement hors de l'écran ?
- *Astuce : Tu devras vérifier si la position x du sprite est inférieure à une certaine valeur ou si la position x du sprite est supérieure à une certaine valeur.*

**Félicitations, tu es un artisan du changement Moonhack !**

**N'oublie pas de demander à un adulte d'enregistrer ta participation sur [moonhack.com](https://moonhack.com)**

