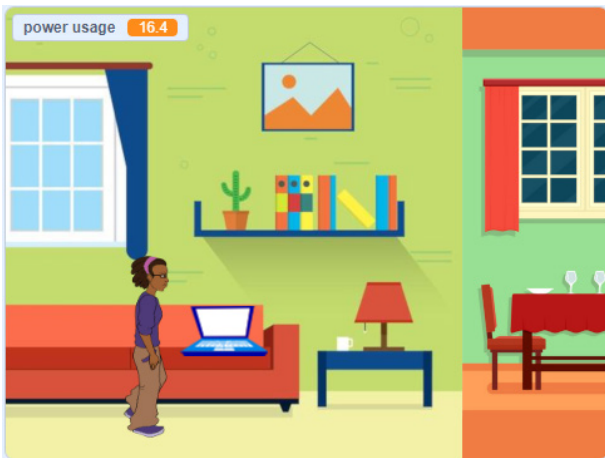# MOONHACK 2021

Home Power Hero

We use a lot of power in our homes! One of the simplest ways to reduce our energy consumption is to turn off appliances!

INTRODUCTION

What you will make

You will make a game that uses parallax scrolling to help Avery minimise her energy consumption by turning off appliances in her home.

What you will learn

- Parallax scrolling (moving the background around the character)

What you will need

HARDWARE

A computer capable of running Scratch 3. This project uses the space bar, so may not be suitable for an iPad.

SOFTWARE

Scratch 3:
either online
rpf.io/scratchon
or offline
rpf.io/scratchoff

DOWNLOADS

Offline starter project
bit.ly/powerhero

Additional notes for educators

Here is a link to the completed project
https://scratch.mit.edu/projects/473116991/

Check out our blog post for this project with tips, curriculum and supporting material at
medium.com/@codeclubau

Code Club Australia recognises the Traditional Custodians of the land across Australia and their continuing connection to land, cultures, and communities. Australia's traditional owners are the world's first innovators.
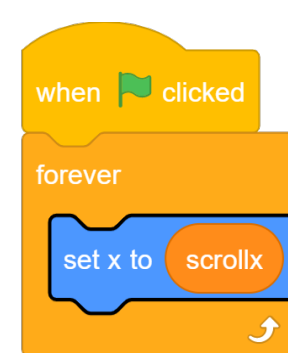
1. MOVING HOUSE

There are a few different ways to show walking in a game. One is to have the sprite move around the scene; another is to have the scene move around the sprite. This second option is called "parallax scrolling" and is what we're going to use in this game.

- Open the starter project at bit.ly/powerhero

- First, we need to set up the Background so that it moves while Avery stays still, simulating motion.
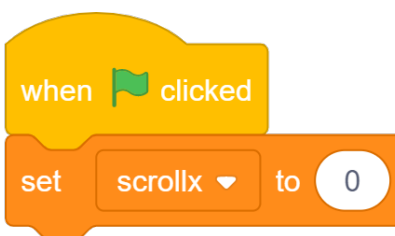
- Add this code to your House Sprite.

```
when [flag] clicked
forever
```

- Next, we'll add code in this forever loop that will move the House Sprite.

- The variable 'scrollx' keeps track of how far Avery moves in the x axis (left to right).

- By moving the background in the opposite direction to Avery's movement, it will make it look like she is moving relative to the background.

```
when [flag] clicked
forever
    set x to (scrollx)
```

- Every time we play the game, we need to reset our scrollx variable so that the House Sprite goes back to the start.

- Add this code.

- (It won't do anything yet, don't worry!)

```
when [flag] clicked
set [scrollx ▾] to (0)
```

2. WALKING

Now we have the House ready to move according to the value of the 'scrollx' variable, let's make it happen!
We want to make Avery walk to the right when we press the right arrow key.

- • Select the Avery Sprite.
- • Change Scrollx when the right arrow key is pressed.
- • Test your code so far. When you press the right arrow key, the background should move to the left, making it look like Avery is moving to the right relative to the background.

```
when  right arrow  key pressed
change  scrollx  by  -10
```

- • Avery is just gliding along, let's make it look like she's walking.
- • Avery has several costumes that, when played one after another, make it look like she's walking.
- • Add the "next costume" sprite to your code.
- • Test your progam again by pressing the right arrow key.

```
when  right arrow  key pressed
change  scrollx  by  -10
next costume
```

- • Avery sure is a fast mover!
- • Let's slow her down by adding a wait block.
- • Test your program again. Avery should be walking at a normal speed now.
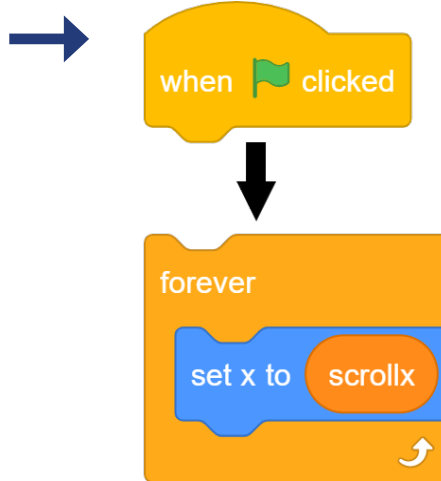
```
when  right arrow  key pressed
change  scrollx  by  -10
next costume
wait  0.05  seconds
```

3. MORE BACKGROUND

Avery can't walk very far before she runs out of background. Let's add more background for her to walk on.

- Select the House Sprite.
- Separate the 'When Green Flag Clicked' block from the forever loop.

```
when 🏳 clicked

forever
    set x to (scrollx)
```

- Under the 'When Green Flag Clicked' block add code to set up the initial conditions.

```
when 🏳 clicked
switch costume to (0)
hide
```

- Add a loop that will create a clone for each costume of the sprite.

```
switch costume to (0)
hide
repeat (3)
    create clone of (myself)
    next costume
```
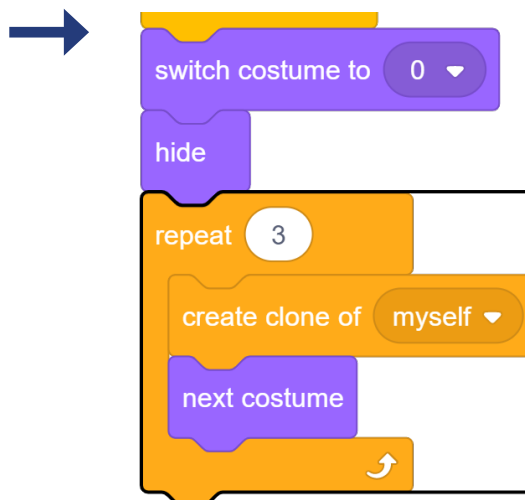
- On the code that you detached, add a 'When I Start As A Clone' block.

```
when I start as a clone
forever
    set x to (scrollx)
```

- Because we have multiple costumes that make up the background, we need to move them across the screen with scrollx, but each costume needs to be offset by 480 pixels from the previous one (480 pixels is the width of the costume).

- Add this code - make sure to check it carefully.

```
when I start as a clone
forever
    set x to (scrollx + (costume number) * 480)
```

- If we run our program now, the House Sprite won't appear, because we've previously told Scratch to hide them. We don't just want to show all the sprites, because Scratch doesn't let our sprites go all the way off the screen.

- Instead we can tell Scratch that if the actual x coordinate of the sprite is where we think it should be, then show it, otherwise, hide it.

```
when I start as a clone
forever
    set x to (scrollx + (costume number) * 480)
    if (x position = scrollx + (costume number) * 480) then
        show
    else
        hide
```

- Tip: If you don't want to rewrite the code for the position of the surface, you can duplicate it by right clicking on the + block and pressing duplicate

- Test your program.
- You'll notice that the House background starts to the right of the screen, or you may not even be able to see it! We don't want that; we want it to start so that the first House costume takes up the whole screen.
- To do this, we need to set scrollx to -480. (Remember 480 is the size of our House sprite in pixels).
- Update the starting value of scrollx by modifying this code.
- Test your program again. The background should now take up the whole screen when you press the green flag to start!

```
when [green flag] clicked
set scrollx to (-480)
```

4. MOVING LAPTOP

When Avery is walking, the laptop isn't moving with the background, which makes it look like it's floating along in front of her! In this step, we'll make the laptop move with the background, so that Avery can walk up to it.

- The aim of the game is to turn off all the appliances as quickly as possible. When the game starts your laptop should be turned on.
- Click on the laptop sprite, and add the following code.

```
when [green flag] clicked
switch costume to (on)
```

- The laptop needs to stay still relative to the background, which means it needs to move relative to Avery.
- Add this code.
- Test your program. The laptop should stay still as Avery walks towards it. What happens when it goes off screen? It gets dragged along the left side of the screen!

```
switch costume to (on)
forever
    set x to (scrollx + 565)
```

- We need to add code that will hide the laptop when it should be off screen. To do this, we will hide the laptop when it's touching the edge.
- Add this code.
- Test your code again. The laptop should now disappear when it reaches the edge of the screen.

```
set x to ( scrollx + 565 )
if < touching ( edge ) > then
    hide
else
    show
```

5. ENERGY SMART AVERY

The purpose of the game is to turn off appliances, so we need to give Avery the ability to turn them off!

- We're going to use the space key to turn off the laptop, but Avery can only turn off the laptop when she's touching it.
- Add this code to the laptop sprite.

```
when space key pressed
if < touching ( Avery ) > then
```

- Avery needs to be able to turn the laptop on and off as she pleases. You'll notice the laptop sprite has two costumes "on" and "off". We can just cycle between them using the "next costume" block.
- Test your code.

```
when space key pressed
if < touching ( Avery ) > then
    next costume
```

- To make the game more exciting, we want to measure the power usage so that we can use it to give the player a challenge later.

- To do this, we want to increase the power usage variable whenever the laptop is on.

- **0.012** is the number of Watt Hours an average (50 watt) laptop uses in a second.

- Add this code to the Laptop Sprite.

- Test your code.

```
when [flag] clicked
forever
    if < costume [name ▾] = ( on ) > then
        change [power usage ▾] by (0.012)
    wait (1) seconds
```

6. MORE APPLIANCES

So far, we've only added code to our laptop, but we've also got a lamp and a TV, so let's make those work, too.

- The code for our Lamp will be very similar to our Laptop. Instead of writing all the code, let's copy our three code stacks across.

- We can do this by clicking and dragging each stack of code from our laptop sprite to our lamp sprite.

```
Laptop
when [space ▾] key pressed
    if < touching [Avery ▾] ? >
        next costume
```

- Once the three stacks are copied across, we'll need to edit a couple of values.

- Make sure you have the lamp sprite selected.

- Update the location of the lamp, so it's not starting on top of the laptop.

```
forever
    set x to ( scrollx + (900) )
    if < touching [edge ▾] > then
```

- We also want to change the power usage for the lamp.

- Change the power usage of the lamp to **0.028** (the number of Watt Hours used per second by a 100W incandescent lamp).

- Test your program. You should now have both the laptop and the lamp in your game.

```
if ( costume name = on ) then
    change power usage by (0.028)
```

Challenge:
What about the TV?

- Now that you've added the lamp, can you use the same process to add the TV?

- Hint: You'll need to change the starting position and the power usage. An average 42 Inch LCD TV uses about 120W of power, which is **0.033** Watt Hours per second. And there's a nice spot on the kitchen bench with an x value around **1200**.

---

7. GAME OVER

To make our game challenging, we're going to make it a race against the clock (or, more accurately, e-meter).

House

- At the start of the game we want to set our power usage to 0, wait until we reach the maximum amount of power, and then it's game over.

- Add the following code to your House Sprite.

```
when [flag] clicked
set power usage to 0
wait until ( power usage > 1 )
broadcast game over
```

Sprite1

- Now, click on the "Game Over" sprite.
- Add code that will react to the "game over" message.

```
when I receive game over ▼
show
stop all ▼
```

- Test your game. When your power usage goes above 1 watt hour, does the game over message appear?
- What happens when you start the game again? It's still there!

```
when 🚩 clicked
hide
```
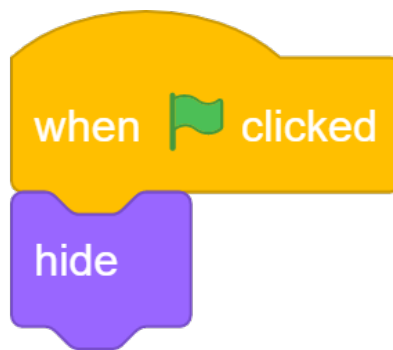
Challenges:

Walk Left

- Currently, Avery can only walk to the right? Can you make her walk to the left. You'll want to make sure she's facing the direction she's walking in.

Other appliances

- What other appliances do you have in your home?
- Research how much wattage they use and add them to Avery's house. What else do you need to change to help Avery turn everything off as quickly as possible?

Your school

- Can you create the same game for your school?

- What appliance do you have in the school that should be turned off? What do you need to change in your code if there is more than one computerW or tv in your school?

More efficient appliances

- Can you make the game easier by making the appliances more efficient.

- Research the power usage of different appliances and to find more efficient alternatives to the existing laptop, lamp, and TV.

- You might also want to adjust the maximum power usage to make the game more challenging.

Advanced Challenge:

- At the moment, the appliances disappear as soon as they hit the edge, and don't appear until they're no longer touching the edge. This means that the sprites are disappearing when they should be on screen.

- Can you change the code so that the appliances don't disappear until they're almost completely off the screen?

- Hint: You'll need to check if the x position of the sprite is less than a certain value or if the x position of the sprite is greater than a certain value.

Congratulations you're a Moonhack changemaker!

Don't forget to talk to an adult about registering your participation at
moonhack.com