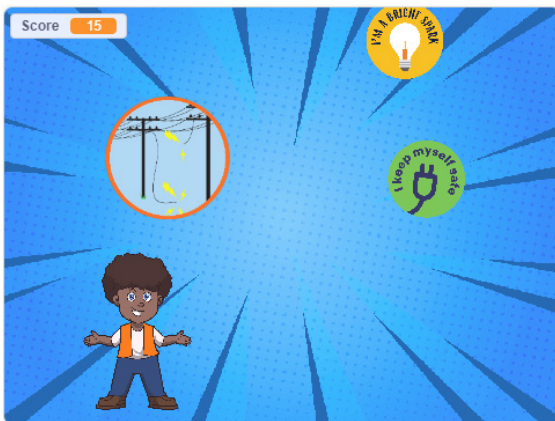


Electrical Safety Catch



Create a catch game. Code your sprite to catch the safety badges and avoid the dangerous situations. Can you reach a score of 50?

INTRODUCTION



What you will need

HARDWARE

A computer capable of running Scratch 3

SOFTWARE

Scratch 3:
either online
[rpf.io/scratchon](https://scratch.mit.edu)
or offline
[rpf.io/scratchoff](https://scratch.mit.edu)

Additional notes for educators

Here is a link to the completed project
<https://scratch.mit.edu/projects/848866804/>

This project is a collaboration between Code Club Australia and New South Wales electricity distributors Ausgrid, Endeavour Energy and Essential Energy. Read our blog to learn more about the project.

What you will learn

- How to create falling objects
- How to add a score
- How to use variables


Starter Project

<https://scratch.mit.edu/projects/849547821/>




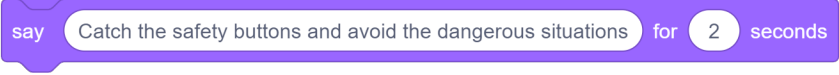
STEP 1 - CHOOSING YOUR CHARACTER

Open the starter project - <https://scratch.mit.edu/projects/849547821/>





For this project you can choose which sprite you would like to use. Your main character can be Kai or Tash. Delete the sprite that you do not want to use.

Select the sprite you are using - Kai or Tash. → 

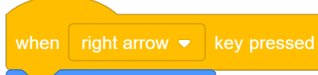
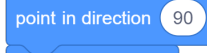


Start with the event **when the flag is clicked**. **Set y to -100** so that Kai/Tash will always travel along the bottom. **Set size to 40%**. Add a **say** block for your sprite to introduce the game.

when  clicked
set y to 
set size to 
say  for 2 seconds

Let's code how the player controls Kai/Tash. Start with the event **when left arrow is pressed**. Add blocks to **point in direction -90** and **move 10 steps**. Set the rotation to **left-right**.

when 
point in direction 
move 
set rotation style 

Now for the opposite direction. Start with the event **when right arrow is pressed**. Add blocks to **point in direction 90** and **move 10 steps**. Set the rotation to **left-right**.

when 
point in direction 
move 
set rotation style 

Test your code. Kai/Tash should move from left to right, controlled by the arrow keys.

STEP 2 - FALLING ITEMS

Select the Danger sprite → 

First we need to add code for the danger sprite to start somewhere along the top. Start with the event **when the green flag is clicked**. Add blocks to **go to random position** and to **set y to 180**. Add a **wait** block for the instructions to be spoken.

```

when green flag clicked
  go to random position
  set y to 180
  wait 3 seconds
  
```

Add a **forever** block that will **change y by -3**. This will make the danger sprite constantly fall.

```

when green flag clicked
  go to random position
  set y to 180
  wait 3 seconds
  forever
    change y by -3
  
```

Test your code. The danger sprite should 'fall' to the bottom of the screen.

When the sprite reaches the bottom, we need it to look like another one falls from the top. Inside the forever block and an **if then block** placing a **less than** operator inside it. Add **y position** bubble. Then **wait 1 second**, go to a **random position** and **set y to 180**.

```

set y to 180
wait 3 seconds
forever
  change y by -3
  if y position < -195 then
    wait 1 seconds
    go to random position
    set y to 180
  
```

Test your code. Can you still see the top of the sprite when it is 'waiting'? Let's add **show** and **hide** buttons so that when it reaches the bottom it **hides**, and when it starts at the top it **shows**. Let's also add in a **next costume** button.

```

forever
  show
  change y by -3
  if y position < -195 then
    hide
    next costume
  wait 1 seconds
  go to random position
  set y to 180
  
```

If you go to the costume tab you will see 6 costumes that show dangerous situations.

Test your code. The sprite should fall to the bottom, disappear, and then reappear at the top of the screen, showing the next costume.

Copy the code onto the Safe and Bright sprites (drag the code until it hovers over the sprite you are copying to). On the Safe sprite make these changes -

- inside the **operator** make the y position less than -200



```
when green flag clicked
  go to random position
  set y to 180
  wait 3 seconds
  forever loop
    show
    change y by -3
    if y position < -200 then
      hide
      wait 1 seconds
      go to random position
      set y to 180
```

On the Bright sprite make these changes -

- **change y by -5** to make it fall faster
- **wait 2 seconds** to add more time before a new one falls
- inside the **operator** make the y position less than -200

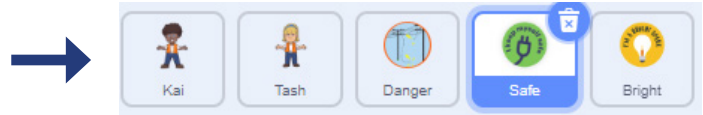


```
when green flag clicked
  go to random position
  set y to 180
  wait 3 seconds
  forever loop
    show
    change y by -5
    if y position < -200 then
      hide
      wait 2 seconds
      go to random position
      set y to 180
```

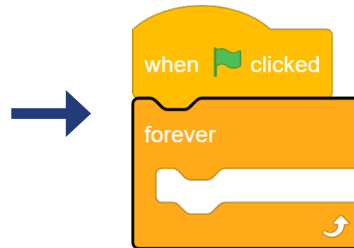
Test your code. All 3 sprites should be falling and reappearing.

Let's code how Kai/Tash will catch the other sprites. We want the player to catch the Safe and Bright sprites to earn points, and to avoid the Danger sprite.

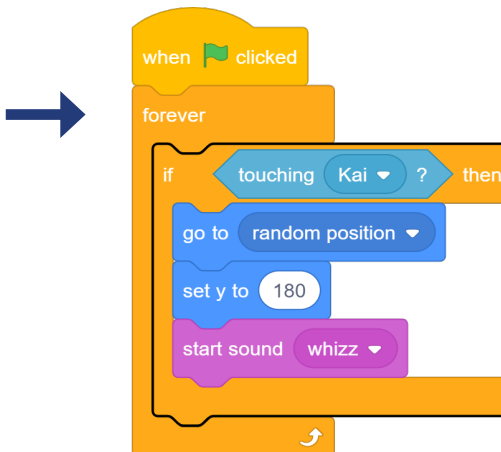
Select the Safe sprite



Start with the event **when the green flag is clicked** and then add a **forever** block.

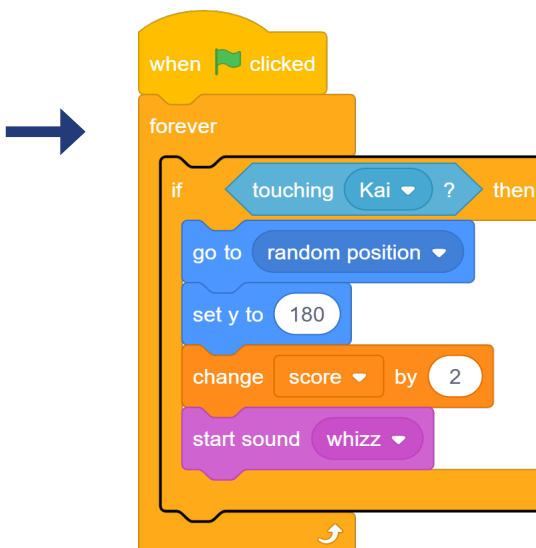


When the Safe sprite touches Kai/Tash we need it to look like it has been caught. Inside the forever block and an **if then block**, and code it if **touching Kai/Tash** then it **goes to a random position** with **y set to 180**. Go into the sound tab and choose a sound. Add a block to **start sound**. Make sure you 'add' a sound in the sound tab.



Test your code. The Safe sprite should disappear as soon as it touches Kai/Tash.

Let's add in a scoring system. Make a new variable called score. Add a block to **change score by 2**.





Copy the code onto the Bright sprite and the Danger sprite (drag the code until it hovers over the sprite you are copying to).

On the Bright sprite make these changes -

- **choose a new sound**
Click on the sound tab to select a new sound. The drop-down menu will now show your new sound to choose.



```

when green flag clicked
  forever loop
    if touching Kai then
      go to random position
      set y to 180
      change score by 2
      start sound High Whoosh
  
```



On the Danger sprite make these changes -

- **choose a new sound**
- **change score by -1**



```

when green flag clicked
  forever loop
    if touching Kai then
      go to random position
      set y to 180
      change score by -1
      start sound Wobble
  
```



Test your code. You have added a score element. Let's make sure we set the score to 0 at the start of the game, when the green flag is clicked.



Select the Kai/Tash sprite



Add a block onto your algorithm that sets the score to 0



```

when green flag clicked
  set y to -100
  set size to 40 %
  say Catch the safety buttons and avoid the dangerous situations for 2
  set Score to 0
  
```

Test your code. You should be able to see your score on the screen. It should increase every time you catch a safe or bright sprite, and should decrease if you touch a dangerous situation. Finally, let's create the end of the game when the player reaches a score of 50.

Add a **forever** block to the end of the algorithm. Inside add an **if then** block that runs until the **score is larger than** 50.

The code block diagram shows the following sequence of blocks: a yellow 'when green flag clicked' block, a blue 'set y to -100' block, a purple 'set size to 40 %' block, a purple 'say Catch the safety buttons and avoid the dangerous situations for 2 seconds' block, an orange 'set Score to 0' block, and an orange 'forever' loop block. Inside the 'forever' loop is an orange 'if score > 50 then' block.

Inside the if then block add a block to **say a message for 3 seconds** and a block that will **stop all**.

The code block diagram shows the same sequence of blocks as the previous diagram, but with the 'if score > 50 then' block expanded. It now contains a purple 'say Congratulations! You are an electricity safety whiz! for (3) seconds' block and an orange 'stop all' block.

Test your code. Your game is complete. How quickly can you reach 50 points?

Challenges:

Finishing the game

The finish to the game is really simple. Tell someone the right things to do if you were in these situations in real life. Can you add these messages to your game? Some messages may be -

- Play it safe! Stay clear when powerlines are near
- Spot a fallen powerline? Keep at least 8 metres away and tell an adult
- Don't be a stranger to electrical danger. Use appliances safely and correctly
- Felt a zap from a tap or appliance? Keep away and tell an adult to report it
- Electrical equipment and enclosures are no go zones. Keep well away
- Storms can be dangerous! Always seek shelter and play indoors.

Difficulty

Can you increase the difficulty of catching the sprites? Perhaps you will change their speed, size, points or number of sprites falling at the same time. Maybe you could make changes to your sprite so that it starts moving slowly and then speeds up.

Multi-level game

Can you add in a second level to the game? There are many things we could use to help others learn about electrical safety. Create a 2nd level that teaches more.

Congratulations!

You have created a game that will help others learn about being safe around electricity.

For more information head to your energy providers website:

- [Ausgrid](#)
- [Endeavour Energy](#)
- [Essential Energy](#)

If you have permission to share your projects we would love to host them in our Electricity Safety Week studio.

[Click here](#) to share your project

