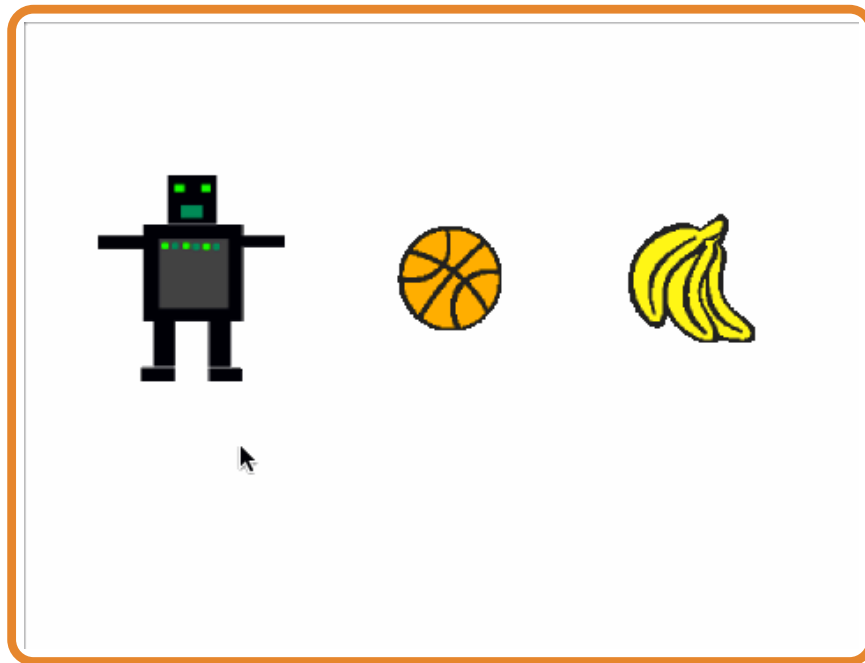


Level 1

Fruit Machine

Introduction

This is a game that has three sprites that change costume. You have to stop them when they're showing the same picture (like a fruit machine!).



Activity Checklist - Follow these **Instructions** one by one.



Test Your Project - Click on the green flag to **Test** your code.



Save your work - Click this to **Save** your work.

STEP 1: Create a sprite that changes costumes

Let's import the different pictures for the game



Activity Checklist

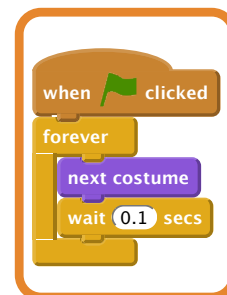
- Start a new scratch project. Delete the cat by right clicking it and clicking **Delete** ☐
- Import a new sprite. ☐
- Choose an image from any folder. We used things/bananas1, but you can use any image you want to. ☐
- Click the **Costumes** tab and import two more things so there are three in total (we used animals/bee1 and things/lego, but you can use any images). ☐

Now we've got some costumes, we want the sprite to change between them.

STEP 2: Making the picture change

✓ Activity Checklist

- Click the **Scripts** tab. ☐
- Click **Control** and drag a **when flag clicked** into the scripts area. This will be triggered when we click the green flag. ☐
- Add a **forever** and attach it so it snaps to the bottom. ☐
- Click the green flag in the top right. Notice that a white outline is around our script. It's running because we clicked the green flag, which triggers this. ☐
- Now click **Looks** and drag in a next costume ☐
- How do we slow it down so it isn't changing so quickly? Click **Control** and drag in a **wait 1 secs** ☐
- Adjust the time until it's repeating at a faster pace (a time of 0.1s looks good). What would happen if we didn't have the wait block? ☐



🚩 Test Your Project

Click the green flag.

- Do the costumes change at a sensible rate? ☐

📁 Save your project

Things to try

1. Adjust the time in the wait block. What numbers do you think would make the game too easy, or too hard?

STEP 3: Making it stop when we click on it

Great! We can make the sprite change costumes forever, but how do we make it stop when we click on it?



Activity Checklist

- Create a new variable by clicking **Variables** and Make a variable. Call it **stopped** and make it for all sprites, then uncheck the box next to it so it doesn't display on the stage. ☐
- Set **stopped** to 1 when someone clicks on the image using **when sprite1 clicked** and **set stopped to 0** making sure to change the value from 0 to 1. ☐
- Now we need to make the image stop changing when the variable stopped equals 1. Click **Control** and change the forever loop to a **forever if** and use a new equals operator to check if **stopped** equals 0 ☐
- Finally, add a **set sprite1-stopped to 0** underneath the **when flag clicked** ☐



Test Your Project

Click the green flag, wait for a moment, then click on the sprite.

- Does it change costume before you click on it? ☐
- Does it stop when you do click on it? ☐
- **Start the sprite again.** Does it stop when you put the mouse pointer on it, without clicking? Does the sprite stop when you click somewhere else on the Stage? Somewhere else in the Scratch window? Somewhere outside the Scratch window? ☐



Save your project

Step 4: Creating the other sprite

Now we need to make the other sprites so we can play our game!



Activity Checklist

- Duplicate the sprite (Sprite1) by right clicking on it in the bottom right corner. ☐
- Duplicate it again so there are 3 sprites on the screen. ☐
- Move each sprite so they are in a line. Make them a bit smaller with if you need to. ☐



Test Your Project

Click the green flag. All the sprites should change. Try to stop them all on the same picture!



Save your project

Things to try

1. When you start the game just after you've loaded it, all the sprites show the same costume and change in unison. How about you make the sprites change to a random costume when the green flag is clicked? Hint: try picking a random costume for each sprite when the game is started.

Well done you've finished the basic game. There are more things you can do to your game though. Have a go at these challenges!

Challenge 1: Make the game harder

Change the difficulty of the game somehow. Just making the costumes change quicker is easy. Try and do something more imaginative. Some ideas you might like to try:



Activity Checklist

- Change the number of costumes each sprite has. ☐
- Make some sprites have unique costumes. ☐
- Have different times between costume changes. ☐
- Have each sprite move to a random costume rather than the next one. ☐



Test Your Project

Have fun coming up with your own things!

Every time you make a change, think about whether it make the game easier or harder. Is the game too easy or too hard? How can you adjust the difficulty so it's just right?

Challenge 2: Make the game get harder and easier over time

Different people will have different skills at playing the game. **How could you make the game adjust its difficulty depending on the player?**



Activity Checklist

- One way you could do it is to **adjust the speed the costumes change at**. You can use a variable, called **delay**, to give the duration of each sprite's **wait** block. If the player wins the round, the delay can be reduced a little (to make the game harder). If the player loses the round, the delay can be increased a little (to make the game easier).



Test Your Project



Save your Project

Challenge 3: Detect when all the sprites have stopped on the same costume

The aim of the game is click on the sprites so they're stopped while showing the same costume. It would be nice if the stage detected when you'd finished playing and then told you if you had won or lost by checking to see if each sprite had the same costume.



Activity Checklist

- First, the stage needs to know when the player has finished. We can do this by having the stage check to see if all sprites have stopped moving when we click on one of them. Go back and modify each of the sprite clicked blocks to broadcast a new message, **checkForEnd** ☐
- The Stage can respond to this message and check if the game is over by seeing if all three sprites' **stopped** variables are set to 1, by using the **x position of Sprite** block for each sprite, and changing **x position** to **stopped**. If all three sprites have a stopped value of 1, we know the game is over and we can check to see if the player has won. ☐
- To do this, we can use the same **x position of Sprite** block, but instead of looking at the **stopped** variable, we can look at the **costume #** and see if Sprite1 has the same costume as Sprite2, and if Sprite2 has the same costume as Sprite3. ☐
- To do this, you'll need an **if** block to check each stopped variable, and inside that an **if... else** block to see if the player has won or lost by comparing each **costume #**. ☐
- From this point on, you could announce the result of the game using a **broadcast** and respond to this with another sprite. Maybe get Felix back to congratulate or commiserate the player? ☐



Test Your Project

Well done you've finished, now you can enjoy the game! Don't forget you can share your game with all your friends and family by clicking on **Share** on the menu bar!