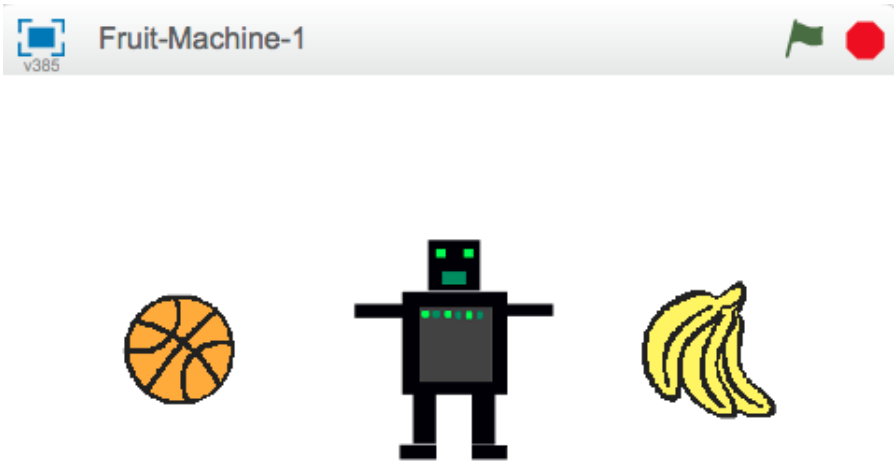
Additional Projects

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 Project	Make sure to SAVE your work now

Step 1: Create a sprite that changes costumes

Activity Checklist

Let's import the different pictures for the game

1. Start a new scratch project. Delete the cat by right clicking it and clicking Delete ☐
2. First of all let's add a new Backdrop from the library. Choose the rays backdrop from the Other category and then delete the original blank stage. ☐
3. Now add a new sprite from the library. ☐
4. Choose an image from any folder. We used things/Bananas, but you can use any image you want to. ☐
5. Click on the blue 'i' next to the sprite's picture in the Sprites window. Rename the sprite to 'Fruit'. ☐
6. Now click the Costumes tab and import two more things so there are three costumes in total (we used things/apple and things/watermelon-a, but you can use any images you like). ☐

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

Step 2: Making the picture change

Activity Checklist

1. Click the `Scripts` tab.
2. Click `Events` and drag a `when flag clicked` into the scripts area. This will be triggered when we click the green flag.
3. Click the Control tab and add a `forever` and attach it so it snaps to the bottom.
4. Click the green flag in the top right. Notice that a yellow outline is around our script. It's running because we clicked the green flag, which triggers this.
5. Now click `Looks` and drag in a `next costume`
6. How do we slow it down so it isn't changing so quickly? Click the `Control` tab and drag in a `wait 1 secs`
7. Adjust the time until it's repeating at a faster pace (a time of 0.5s looks good). What would happen if we didn't have the `wait 1 secs` block?

Test Your Project

Click the green flag.

Do the costumes change at a sensible rate?

Save your project

Things to try

- [ ] Adjust the time in the `wait 1 secs` 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

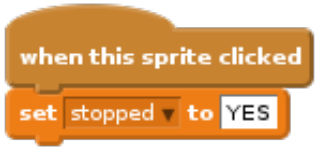
Activity Checklist

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

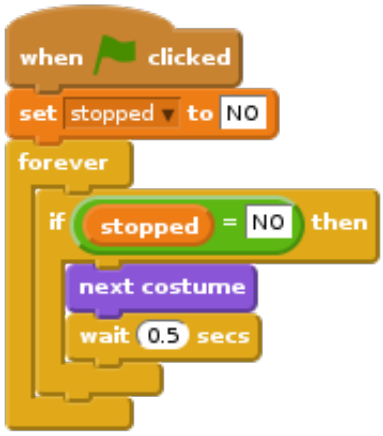
One way to do it is by using a variable to set the state of the Sprite. This will also be useful later...

1. Create a new variable by clicking `Data` and `Make a variable`. Call it `stopped` and make it for only this sprite, then uncheck the box next to it so it doesn't display on the stage. ☐
2. At the start of the game, the sprite won't have been clicked so we'll set the variable to be equal to "NO". ☐

3. Now we'll set the variable `stopped` to "YES" when someone clicks on the sprite. ☐



4. Finally we need to make the sprite stop changing costume when the variable `stopped` changes to "YES". Add an `if...then` loop and use a new equals `[] = []` operator block (found under the Operators tab) to check if `stopped` is still "NO".



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 program again. Does it stop when you put the mouse pointer on it, without clicking?
Does the sprite stop when you click anywhere else on the Stage?

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

1. Duplicate the sprite (Fruit) by right-clicking on it in the bottom

right corner.

2. Duplicate it again so there are 3 sprites on the screen. ☐
3. 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 by clicking on each one in turn!

Save your project

Step 5: Start each sprite with a random costume

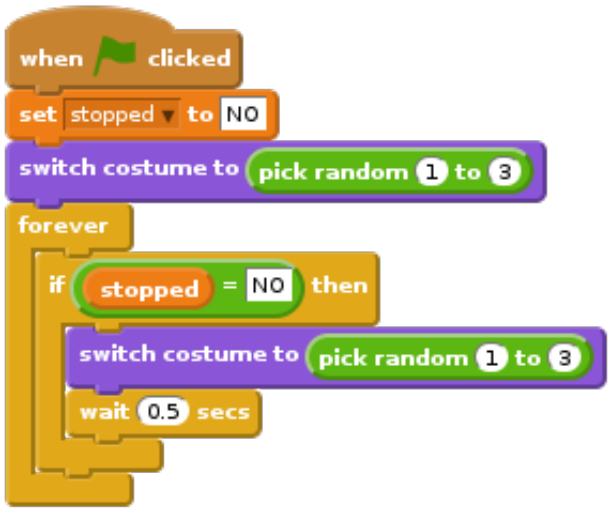
Let's make the sprites change to a random costume when the green flag is clicked.

When you start the game just after you've loaded it, all the sprites show the same costume and change in unison. It would make the game more interesting (and harder) if they changed in a less predictable way.

Activity Checklist

1. If you look under the `costumes` tab for a sprite then you'll see that each costume has a number. You can specify which costume a sprite is wearing using either its name or its number. ☐
2. To make the sprite start with a random costume, let's add a `switch costume to` block with `pick random (1) to (3)` to choose the costume number. ☐
3. We can also use exactly the same block in the `forever` loop so that the sprite switches to a different costume each time it ☐

changes during the game.



4. Do the same thing for each of your sprites. ☐

Test Your Project

Click the green flag. All the sprites should change their costumes in an unpredictable sequence.

How would we need to change our script if we added another costume?

Save your project

Things to try

Make the game harder

Change the difficulty of the game somehow. Just making the costumes change quicker is fairly easy. Can you come up with something more imaginative?

Some ideas you might like to try:

- [ ] Change the number of costumes each sprite has.
- [ ] Make some sprites have unique costumes.
- [ ] Have different times between costume changes.

Have fun coming up with your own things! Every time you make a change, think about whether it makes the game easier or harder. Is the game too easy or too hard? How can you adjust the difficulty so it's just right?

Step 6: Display a message when the game has finished.

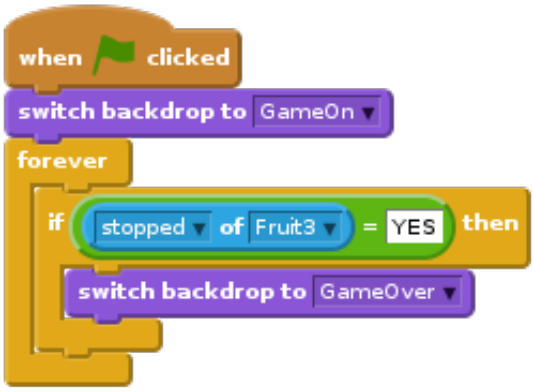
Let's show the player a "Game Over" message when they've finished

Activity Checklist

First of all, let's create a different Backdrop to display when the game has finished.

1. Click on the stage and then the `Backdrops` tab. Change the name of the existing backdrop to "GameOn". ☐
2. Duplicate the backdrop and then add some text to the copy that says "Game Over". You can change the size of the text by clicking on it and then dragging one of the corners. Rename this backdrop to be "GameOver". ☐
3. Click on the `Scripts` tab for the stage and set the "GameOn" backdrop to be the one displayed when the game is started. ☐

4. How can we detect when all the sprites have stopped? Remember we use the `stopped` variable to record whether each sprite has been clicked? Let's check the `stopped` variable for the Fruit3 sprite to see if the game is over. Select the Fruit3 sprite and then can use a `x position of Fruit3` block from the `Sensing` tab, but change x position to `stopped`.



Test Your Project

Click the green flag. Does the "Game Over" message appear when you click on Fruit3?

What happens if you stop Fruit3 before you've clicked on both of the other fruit sprites? Let's
modify our script so that it will work regardless of the order in which the sprites are stopped.

- To check to see that all three fruit sprites have had their `stopped` variables set to YES, we can use the `and` operator. This is a compicated block that can be quite fiddly to assemble, so try and put it together one step at a time.

Test Your Project

Click the green flag. Does the "Game Over" message appear when you all 3 Fruits are stopped, regardless of the order you clicked on them?

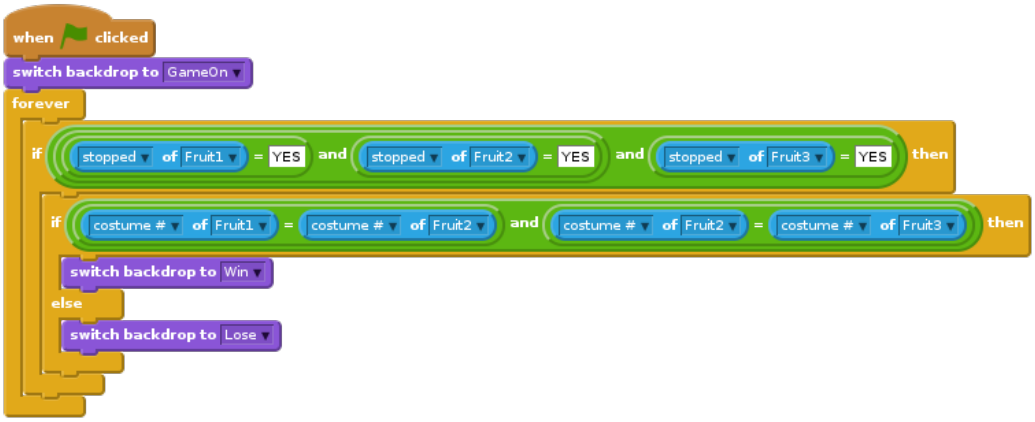
Save your project

Step 7. Tell the player whether they've won or lost.

The aim of the game is to click on the sprites so they stop while showing the same costume. It would be nice to also display a message that told you whether you'd won or lost.

Activity Checklist

1. We wrote the code to check that the game was over in step 6, so all we need to do now is check to see if the player has won. Go back to the backdrops and add some more text to the GameOver backdrop so that also displays the word "WIN". Then change its name to "Win". ☐
2. Copy the backdrop again to create one with a "Lose" message. Give it the name "Lose". ☐
3. Now we need some code to work out which backdrop to display once the game is over. We can use an `if...then...else` block to see if the player has won or lost by comparing each `costume #` (costume number) using a similar `x position of Sprite` block like we did before. This time, instead of looking at the `stopped` variable, we can check the `costume #` and see if Fruit1 has the same costume as Fruit2, and if Fruit2 has the same costume as Fruit3. ☐



Test Your Project

Click the green flag. Does the correct message appear when the game has finished? What will happen if each sprite's costume numbers don't match (for example, if Fruit2's costume number 3 is an apple and Fruit3's costume number 3 is a melon)?

Save your project

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

Challenge: 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?

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).

You'll probably need to think about using a different way of starting the game each time it is played instead of the `when flag clicked` Then you can store values in variables that are remembered between each round of the game.

Save your project

Well done, you've finished! Now you can enjoy your game!

Don't forget you can share your game with all your friends and family by clicking on Share on the menu bar!