SCRATCH TUTORIAL: CATCHING FALLING APPLES

DIGITAL CONTENT CREATION > 3.4 PROGRAMMING

TARGET GROUP	AGE GROUP	PROFICIENCY LEVEL	FORMAT	COPYRIGHT	LANGUAGE
School drop outs, Students (primary school), Students (secondary school)	Children, Teenagers	Level 1	Activity sheet	Creative Commons (BY-SA)	English, French

This tutorial explains how to make a game in which the player uses the webcam to catch falling apples to stop them from hitting a character walking along the screen. This is meant for an audience already familiar with Scratch since certain certain basic instructions are not detailed.

General Objective	Skillset building
Preparation time for facilitator	less than 1 hour
Competence area	3 - Digital content creation
Name of author	Adnan Elbarrak
Support material needed for training	Computer with webcam and Scratch or internet connection.
Resource originally created in	French

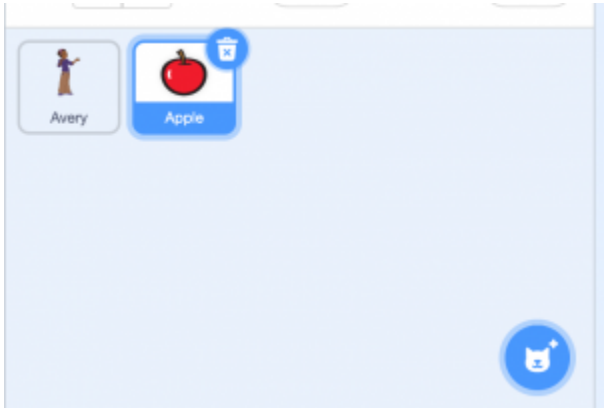
WORKSHOP DIRECTIONS

1 Introduction

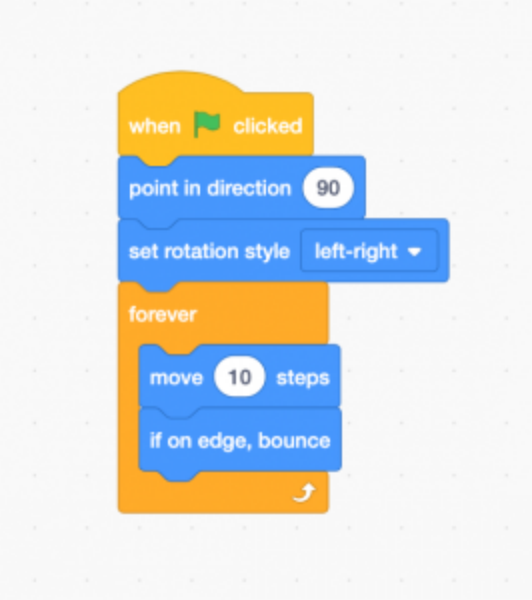
We will make a game in which the player uses the webcam to catch falling apples to stop them from hitting a character, Avery, who is walking along the screen. This is meant for an audience already familiar with Scratch. Certain basic instructions (e.g. adding and deleting sprites) are not detailed.

2 Adding Sam

Delete the default sprite and import a character from the library (for example ‘Avery’ in the ‘People’ category) as well as an apple.



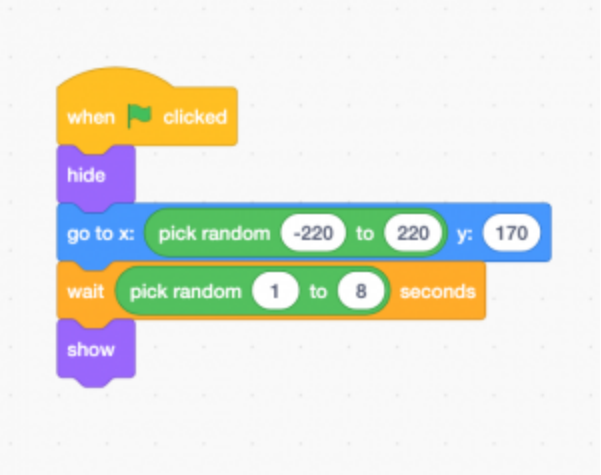
We now need to make Avery move all over the stage. For this, add the following code to Avery’s script.



When the green flag is clicked, Avery points to the right (90°), sets her rotation style to left to right (to ensure she doesn't flip upside down), then will indefinitely move by 10 coordinates on the axis and change direction when she touches the wall.

3 Making apples appear randomly

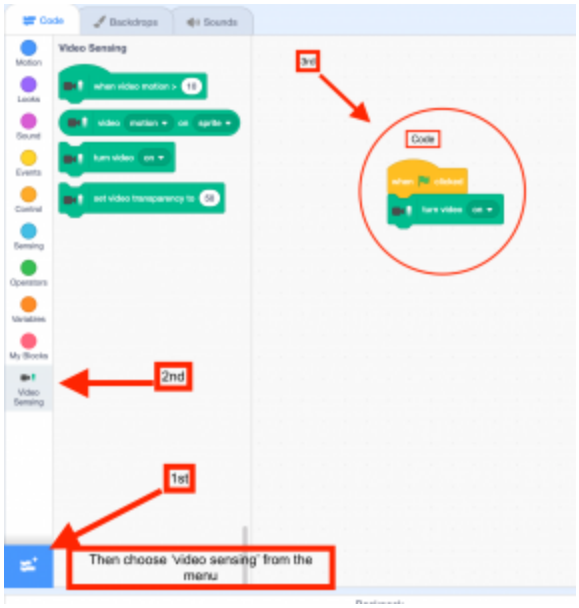
To make apples appear randomly at different times and positions, add the following blocks to the apple's script:



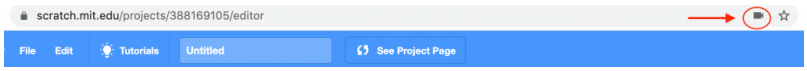
Code explanation: when the game starts (i.e. the green flag is clicked), the apple is hidden. It places itself randomly on the x axis (between -220 and 220) and stays high on the y axis (170). It will wait a random amount of time (between 1 and 8 seconds) before appearing.

4 Touching apples

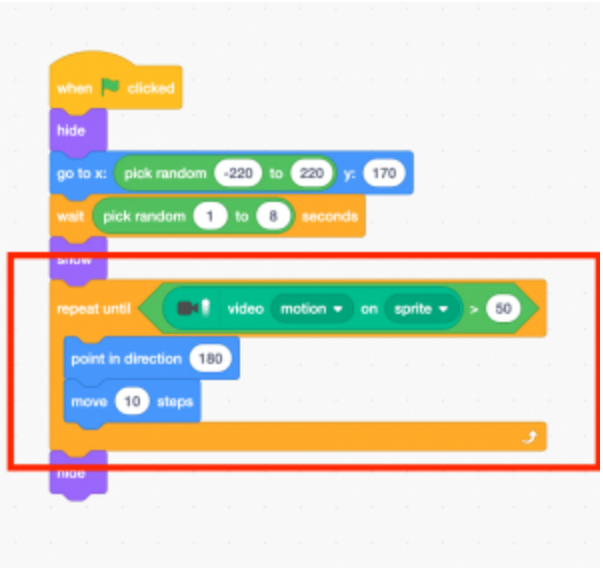
To turn our program into a game, we need to give the player something to do. In our example, they have to touch the apples to make them disappear. For this we will use the camera, which will lets the game directly detect the player's movement and its direction. Add the following block to the stage's script to activate the camera:



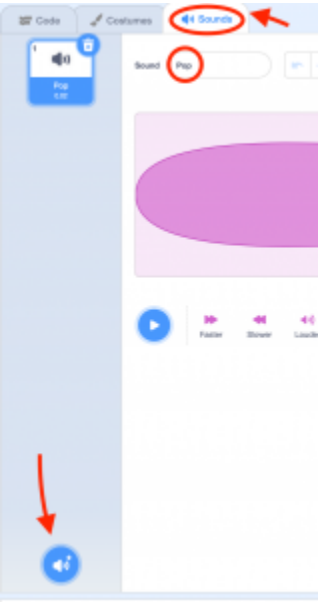
When the green flag is clicked, the computer's camera will activate and the stage displays the capture. Try it now. Note: if it's not working, you may need to give permission to Scratch to access your computer's camera. If using Chrome for example you can do that by clicking on the camera icon shown below.



Next, we have to have it so the apple disappears when it is touched by the player. Add the following blocks to the apple's script:



Once the apple appears, in points downwards (180°) and repeatedly moves by 10 steps until the camera detects reasonably quick movement on the apple. When the movement is detected, the apple hides. Test the code. You can change the value of ‘move … steps’ to make the game easier or more difficult. We have the apple play a sound effect when we touch it. Go to the ‘Sound’ tab and click on the speaker icon. Then choose a sample from the library (for example, ‘pop’)

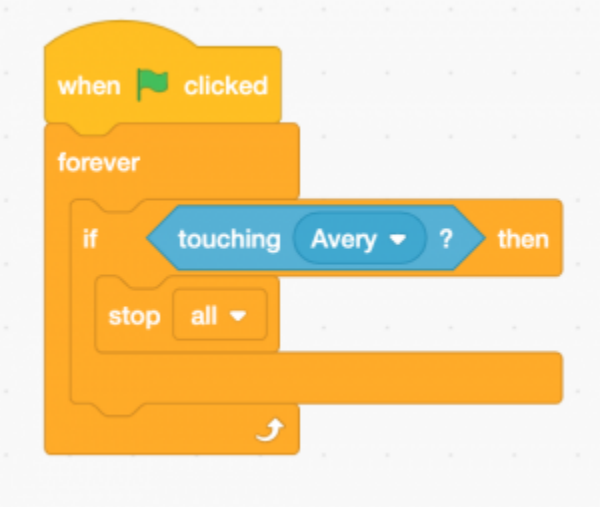


Add the following block to make the chosen sound effect play when an apple is touched:



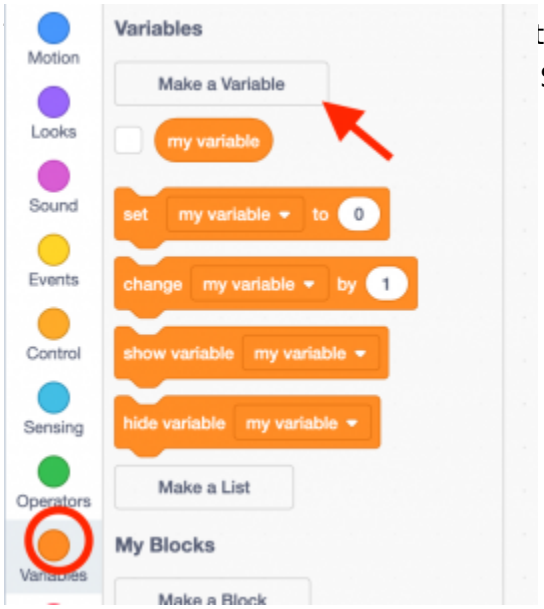
5 Ending the game

Now that the apples are falling and the player can catch them to protect Avery, we need to make the game stop once the character is hit. Add the following to the apple's script:

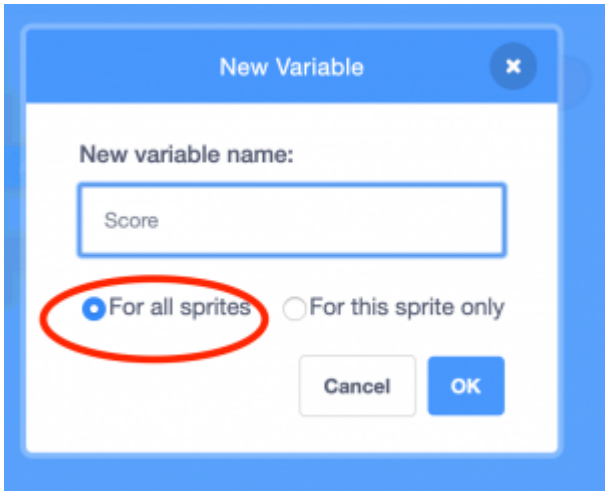


When the green flag is clicked, the game will constantly verify whether Avery is touched. If the condition becomes true, the game will stop.

6 Counting points



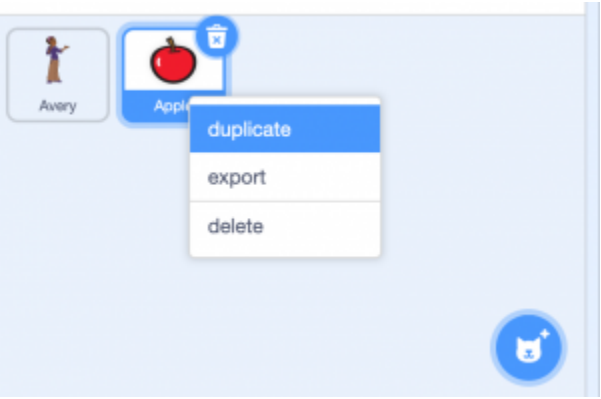
touch an apple. We will use a variable to count points. Create a Score'.



Change the script so that the score increases by one point every time the player hits an apple:



Finally, clone the apple to have several at the same time:



As we coded the game in the apple's script, each cloned apple will have a copy of the same script and will behave identically. You won't need to do any additional coding.

7 Going further

Here are some ways to improve the game:

- Add other objects with special properties (e.g. speeding up or slowing down the game)
- Increase the speed of the apples gradually
- Change the backdrop to indicate various difficulties