Interactive Role Play - Variables

Introduction

The idea of this role play is to try and help the kids understand the concept of what a variable is and how they can be used.

By getting the kids to participate in a real world activity, they can hopefully make the connection between what they are doing and what they are coding.

Basic Premise

The basic idea is that the volunteer will attempt to “program” the children. This is done by firstly selecting a volunteer to stand in the front of the class on the `stage` and act as the `sprite`. The volunteer then writes a `script` on the board and selects another volunteer to “run” the code by reading out each command, slowly, allowing time for the `sprite` to perform the command.

The commands can become more complex and start to include variables.

Concepts covered

This role play should help the kids understand:

1. The `stage`
2. What a `sprite` is
3. What a `script` is
4. What a `variable` is
5. Why a `variable` is used
6. The difference between `variables` for ‘all sprites’ and for ‘this sprite only’

Example script

First start with a simple `script`:

`- When Aaron claps his hands - Walk to Sally - Say 'Hello Sally'`

The `sprite` will probably be a bit confused for the first one, so encourage the person “running” the `script` to read each command slowly, and encourage the `sprite` to execute the command.

Now pick a new name:

`- When Aaron claps his hands - Walk to Sam - Say 'Hello Sam'`

Again, have someone “run” the `script`.

Now try and illustrate to the kids how, when you want to change the name, you have to do so in multiple places. Ask them how we might be able to save ourselves from changing the name in multiple places. If they don’t think of it already, suggest that maybe we could put a placeholder in there. Maybe a `variable`.

Change the `script` as follows:

`` - When Aaron claps his hands - Walk to - Say ‘Hello ’

---

variable = Jenny ``

Ask a child to ‘execute’ the program. They will more than likely read out <name> instead of substituting the real name in for the `variable`. Guide them through this.

Then ask the children to change the name `variable` and “see what the program does”.

`` - When Aaron claps his hands - Walk to - Say ‘Hello ’

---

variable = James ``

By now, the kids should be getting a hang of how changing a `variable` changes the way the `script` executes. Time to step it up a notch.

Now ask for 3 more volunteers. We are about to run a program with 3 `sprite`s at once.

Once again choose a new value for the <name> `variable`:

``` - When Aaron claps his hands - Walk to - Say ‘Hello ’

---

variable = Sarah ``

Ask the class what they think will happen when they execute the program. Have the class to execute the program that now has 3 `sprite`s and ask them why all 3 `sprite`s went to the same person. By now they should be able to allude to the fact that all 3 `sprite`s were sharing the same <name> `variable`.

Now for the kicker. Ask the kids how we might be able to get the 3 `sprite`s to each go to a different person. You may well find that they suggest you give each `sprite` their own `variable`.

So do that:

``` - When Aaron claps his hands - Walk to - Say ‘Hello ’

---

variable for sprite1: = Melissa variable for sprite2: = Mark variable for sprite3: = Matthew ``

Execute the `script` and the `sprite`s should end up at their designated people.

Now try to get the class to reiterate why the `sprite`s went to the different people this time as opposed to the time before.

BOOM, you’re done. Your kids now understand the basic concept of `variable`s..

Ideas for “commands”

- Highfive <name>
- Walk <number_of_steps> steps forward
- Turn <direction>
- Perform some action: <action>