Show String

This program takes a word (string of alphabets) and displays it in a graphical format.

Snap and CS Concepts Used

When we design this program, we will make use of the following Snap and CS concepts. Learn these concepts if you don't know them before proceeding further.

- Algorithms
    - Abstraction
    - Using algorithms
    - Designing new algorithms
- Animation using costumes
- Conditional statements:
    - Conditions: YES/NO questions
    - Relational operators (=, <, >)
    - Conditionals (IF)
    - Conditionals (If-Else)
- Data types – basic
    - Integers
- Data types – strings
    - String operations (join, letter, length of)
- Events
- Looping (iteration)
    - Looping - simple (repeat, forever)
- OOP
    - Clones
- Procedures
    - User defined (custom)
    - Simple
    - With inputs
- Program output
    - Text
- Sequence
- Synchronization using broadcasting

- User input
    - Text
- Variables
    - Simple
    - Local/global scope
- XY Geometry

High Level Design

Snap sprite library does not include sprites that look like alphabet. So, we will acquire these from outside. I am going to copy these graphic images from Scratch. A few examples are shown below along with the names I will use for their costumes:

A-Glow
B-Glow
C-Glow

We will build a sprite that would have 26 costumes each showing one of these letters. The costumes are named as "A-glow", "B-glow", and so on. So, we can figure out costume names corresponding to each alphabet easily by concatenating the letter with "-glow". We will then display each letter by creating a clone of this sprite. In addition, we will include a square costume to represent the SPACE character and call it "blank-glow".

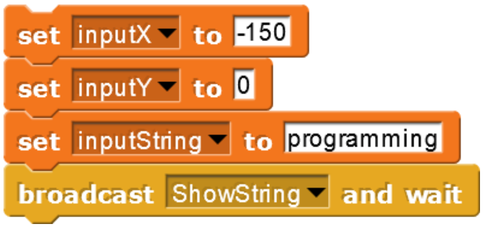
Show string algorithm

Here is the algorithm based on the idea discussed above:

```
Algorithm Show String
Given:
      string S (a sequence of letters),
      (X,Y) starting point,
      w (width of each costume)
Steps:
Go to (X,Y)
For every letter C in S
      If C is a SPACE character
```

```
            Switch costume to blank-glow
      Else
            Convert C to upper-case (we will discuss this later below)
            Switch costume to C-glow
      End if
      Create clone and display
      Change X by w
End repeat
```

Here is an example of how this component is invoked:



```
set inputX to -150
set inputY to 0
set inputString to programming
broadcast ShowString and wait
```

The string is displayed at (-150, 0).

Display:

PROGRAMMING

Changing to upper-case

In the algorithm above, we have to convert every letter to upper-case because costume names are all upper-case. For this purpose, we can simply borrow the custom block "Uppercase" from our string library. Look up this guide to figure out how to download and use the library. The component we are interested in is available as "strings.xml" which you can import into your project.

Final program

Download the program here.

Author: Abhay B. Joshi (abjoshi@yahoo.com)
Last updated: 6 February 2021