

# Library of controls and procedures

In this document, we list several commonly used controls and procedures, which can be imported by any Scratch program. Each library is stored in a sprite. Use the "load sprite from file" option to import the library into your program. You can then copy-paste to other sprites the procedures you are interested in.

Download the library by clicking [here](#).

Radio buttons:.....	2
Geometric shapes: .....	5
Strings routines.....	7
List routines.....	9

## Radio buttons:

This sprite allows you to select one of 3 speeds: slow, medium, fast

Each click sends out an appropriate message. For example, if you click on "slow" the sprite sends "speed slow" message.

This sprite works only if you use it as is, with NO change in orientation or size.

File: radio.sprite2

## How to use the control:

1. Import the sprite into your program.
2. Place the sprite at a suitable place with NO change in its orientation or size.
3. Process the 3 messages "speed slow", "speed medium", and "speed fast" to set the desired parameter in your program.

## Scratch and CS Concepts Used

When we design this control, we made use of the following Scratch and CS concepts.

- Algorithms
  - o Designing new algorithms
- Arithmetic
  - o Expressions
  - o Basic operators (+, -, \*, /)
  - o Advanced operators (abs)
- Concurrency
  - o Synchronization using broadcasting
- Conditional statements:
  - o Conditions: YES/NO questions
  - o Relational operators (=, <, >)
  - o Conditionals (IF)
  - o Conditionals (If-Else)
  - o Conditionals (nested IF)
  - o Boolean operators (and, or, not)

- Costumes
- Data types – basic
  - o Integers
- Events
- Sequence
- Variables
  - o properties

### For advanced users:

#### If you wish to resize the control:

Presently, the distance between left (right) edge of the slow (fast) circle from the center of the med circle is 50. And the radius of all circles is 12. Notice these numbers in the code and change them according to how you resize.

#### If you wish to use the control for some other purpose (e.g. Level: high/med/low):

- Keep only the 1st costume and delete the rest.
- Change the text labels to whatever you want.
- Duplicate the costume to create 2 more. Name them 'med' and 'fast' respectively. Fill the circles as before (blue for med and purple for fast).
- Go to scripts and change broadcast messages.

### Design:

We basically use the geometry to design this control. The sprite's center is also the center of the "med" circle. When the user clicks, we find out whether they clicked on any of the 3 radio buttons, but comparing the x and y values of the pointer with the sprite's center. Here is the algorithm:

#### Algorithm which radio button was clicked:

When sprite clicked:

Given:

$X_m$  = X of mouse pointer

$Y_m$  = Y of mouse pointer

X, Y = coordinates of the sprite's center (also the center of "med")

R = radius of all 3 circles (identical)

D = how far the sprite center is from the outer edge of "slow" and "fast" circles.

Procedure:

```
If Xm is within R of X AND Ym is within R of Y
    "Med" was clicked
Else:
    If Xm is within D of X AND Ym is within R of Y
        If Xm < X
            "Slow" was clicked
        Else
            "Fast" was clicked
        End if
    End if
End if
```

## Geometric shapes:

This library consists of procedures to draw "regular" shapes such as square, pentagon, circle, etc.

File: shapes.sprite2

Here is the complete list:

- Triangle( size )
- Square( size )
- Pentagon( size )
- Hexagon( size )
- Rectangle( width, height )
- Regular polygon( size, number of edges )
- Circle with diameter input
- Circle with circumference input
- Semi-circle with diameter input
- Quarter-circle with diameter input

## How to use the library:

1. Import the sprite into your program.
2. Copy-paste procedures for the shapes you are interested in.

## Scratch and CS Concepts Used

When we designed this library, we made use of the following Scratch and CS concepts.

- Algorithms
  - o Designing new algorithms
- Arithmetic
  - o Expressions
  - o Basic operators (+, -, \*, /)
- Data types – basic
  - o Integers
- Events
- Looping – simple
- Pen art
- Sequence

- Procedures – with input

**Design:**

We use basic geometry and the TRT principle to design these procedures.

## Strings routines

This library consists of procedures to manipulate strings.

Here is the complete list:

- Split (string, letter): splits 'string' into a list tL of tokens separated by 'letter'.
- Uppercase ( letter ): changes the given letter (a thru z) to uppercase.

## How to use the library:

1. Import the sprite into your program.
2. Copy-paste procedures that you are interested in.

## Scratch and CS Concepts Used

When we designed this library, we made use of the following Scratch and CS concepts.

- Algorithms
  - o Designing new algorithms
- Conditionals
  - o IF
  - o Relational operators
  - o Boolean operators
- Data types
  - o Integers
  - o Strings, operators
  - o Lists
- Looping – conditional
- Sequence
- Procedures – with input
- Variables

## Design:

### Split:

Use string traversal.

### **Uppercase:**

Use a series of IF statements that compare with a thru z and change the case.

File: strings.sprite2

## List routines

This library consists of procedures to manipulate lists.

File: list\_routines.sprite2

Here is the complete list:

- Remove From List( item ): removes 'item' from list L. Since Scratch does not allow a list as a parameter to a custom block, you need to use a separate variable for the list.

### How to use the library:

1. Import the sprite into your program.
2. Copy-paste procedures that you are interested in.

*Author: Abhay B. Joshi (abjoshi@yahoo.com)*

*Last updated: 2 February 2021*