



# Sketchify Tutorial

## Defining Interaction and Logic in Sketchify

[sketchify.sf.net](http://sketchify.sf.net)

Željko Obrenović

[z.obrenovic@tue.nl](mailto:z.obrenovic@tue.nl)



# Events and Actions

- Event – what happens
- Action – what to do when something happens
- Logic – connecting actions with events



# Defining Interaction in Sketchify

- **Events**
  - Mouse Events, Keyboard Events, Regions Overlap and Interaction, Variable Events
- **Actions**
  - Sketch Transition, Variable Actions, Timer Actions, Macro Actions
- **Connecting Actions and Events**
  - Drag-and-drop
  - Templates and Formulas



# Defining Interaction in Sketchify

- **Events**
  - Mouse Events, Keyboard Events, Regions Overlap and Interaction, Variable Events
- **Actions**
  - Sketch Transition, Variable Actions, Timer Actions, Macro Actions
- **Connecting Actions and Events**
  - Drag-and-drop
  - Direct Specifying



# Events

- Mouse Events
- Keyboard Events
- Sketch Events
- Variable Events
- Region Overlap Events



# Mouse Events

- Mouse Events are captured by Active Regions
- Two Types of Mouse Events
  - Continuous motions
  - Discrete Mouse Events





# Discrete Mouse Events

- 14 types of mouse events
  - Left Button Click, Left Button Press, Left Button Release, Right Button Click, Right Button Press, Right Button Release, Middle Button Click, Middle Button Press, Middle Button Release, Double Click, Mouse Entry, Mouse Exit, Mouse Wheel Up, Mouse Wheel Down
- Discrete Mouse Events are defined in the properties of active regions

Image	Mouse Event	Action	Param1	Param2
Properties	Left Button Press	Go to sketch		
Move & Rotate	Left Button Press	Variable update		
<b>Mouse Events</b>				
Overlap & Touch				
Embedded Sketch				





# Keyboard Events

- Keyboard events are defined within the sketch settings
- You can capture “pressed” or “released” events for any keyboard key

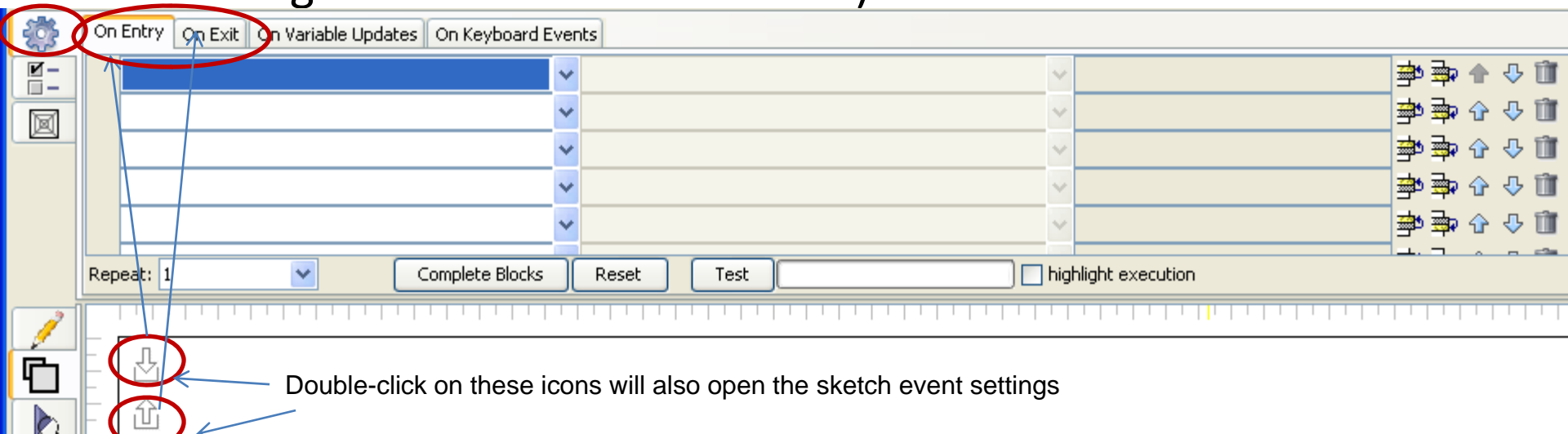
Shift	Ctrl	Alt	Key	Event	Action	Param1	Param2
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		pressed			
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Enter				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Backspace				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tab				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cancel				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Caps Lock				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Esc				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Space				

Double-click on this icons will also open the keyboard event settings



# Sketch Events

- Two types of sketch events
  - On sketch entry
    - Defines what will happen when you open the sketch
  - On sketch exit
    - Defines what will happen when you close the sketch (or go move to another sketch)





# Variable Events

- Variable events are defined in sketch settings
- Define what will happen when variable is updated or has particular value

On Entry On Exit **On Variable Updates** On Keyboard Events

Variable	operator	value	action	param1	param2
	=				
	>				
	<=				
	>=				
	<				
	>=				
	<				
	<=				
	>				
	in				
	not in				

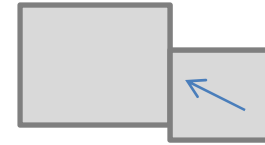
if... VAR  
VAR =

Double-click on this icons will also open the variable event settings

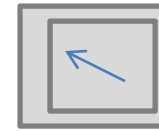


# Region Overlap Events

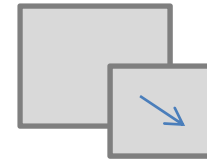
- Detected in active regions
- Define what will happen when two regions overlap
- Four types of events:
  - touches, inside, outside, completely outside



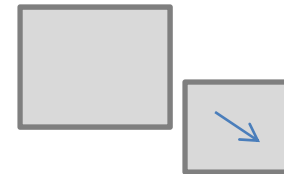
touches



inside



outside



completely outside

1

Make solid (disable walk through)

Region	Event	Action	Param1	Param2
Any region				
	touches			
	inside			
	outside			
	completely outside			

Image  
Properties  
Move & Rotate  
Mouse Events  
Overlap & Touch  
Embedded Sketch

1

Close, Refresh, Undo, Redo, Home, Back, Forward, Search, Help, Print, Zoom, Full Screen, Exit



# Defining Interaction in Sketchify

- **Events**
  - Mouse Events, Keyboard Events, Regions Overlap and Interaction, Variable Events
- **Actions**
  - Sketch Transition, Variable Actions, Timer Actions, Macro Actions
- **Connecting Actions and Events**
  - Drag-and-drop
  - Direct Specifying



# Actions

- Sketch Transitions
- Variable Actions
- Timer Actions
- Macro Actions



# Sketch Transitions

- Actions that causes that current sketch is closed and new one open
- In interface defined as  
“Go To Sketch” <sketch-name>
- When this actions occurs:
  - On exit events occurs on the old sketch
  - On entry sketch events occurs on the



# Variable Actions

- Three types of variables actions
  - **Variable update** – sets the variable to given value
  - **Variable increment** – increments current value of the variable for a given number
    - For example, if variable has the value “122”, increment with “12” will result in change of variable value to “134”
    - Variable has to contain number, otherwise nothing happens
  - **Variable append** – appends given string to the existing content of the variables
    - For example, if variable has the value “abc”, append of “de” will result in change of variable value to “abcde”





# Timer Action

- Two types of timer actions
  - Start timer
  - Stop timer



# Macro Actions

- Two types of macro actions
  - Start macro
  - Stop macro



# Defining Interaction in Sketchify

- **Events**
  - Mouse Events, Keyboard Events, Regions Overlap and Interaction, Variable Events
- **Actions**
  - Sketch Transition, Variable Actions, Timer Actions, Macro Actions
- **Connecting Actions and Events**
  - Drag-and-drop
  - Direct Specifying



# Events and Actions

- Event – what happens
- Action – what to do when something happens
- **Logic – connecting actions with events**



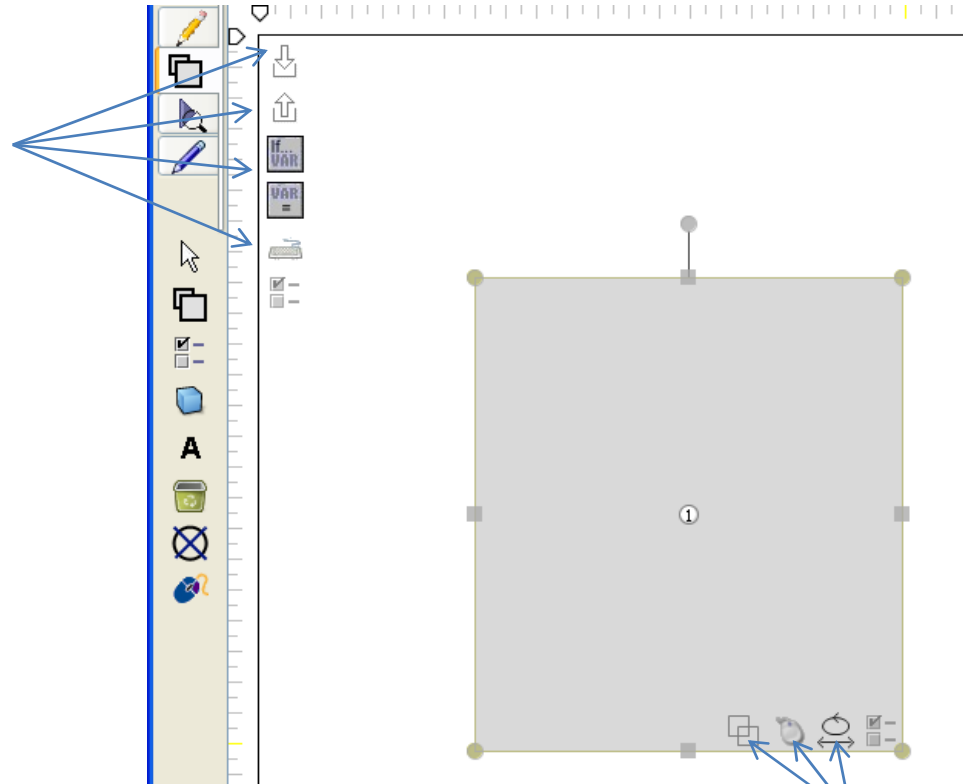
# Connecting Actions and Events

- Drag-and-drop support
  - Drag variable, sketch, timer or macro to connect it with various events
- Specify directly



# Drop Event Anchors

Anchors for connecting sketch events (on entry, on exit, on variable update, on keyboard event) by drag-and-drop of variables, timers and macros. You can also double-click on these icons to open current settings for these events and properties.

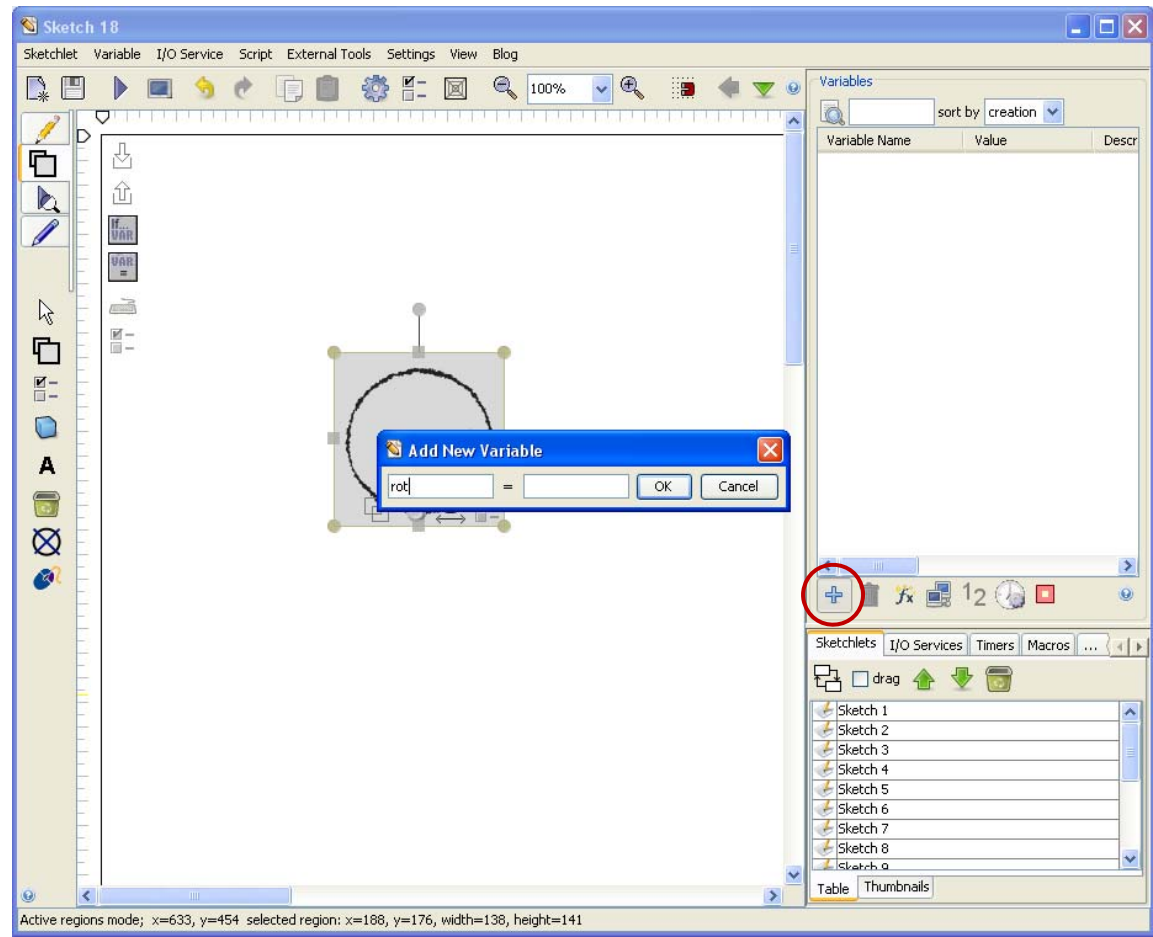


Anchors for connecting region events (region overlap, discrete mouse events, continues mouse events) by drag-and-drop of variables, timers and macros. You can also double-click on these icons to open current settings for these events and properties.



# Example: Connecting Region Continuous Mouse Event to Variable

- Step 1: Creating variable “rot”

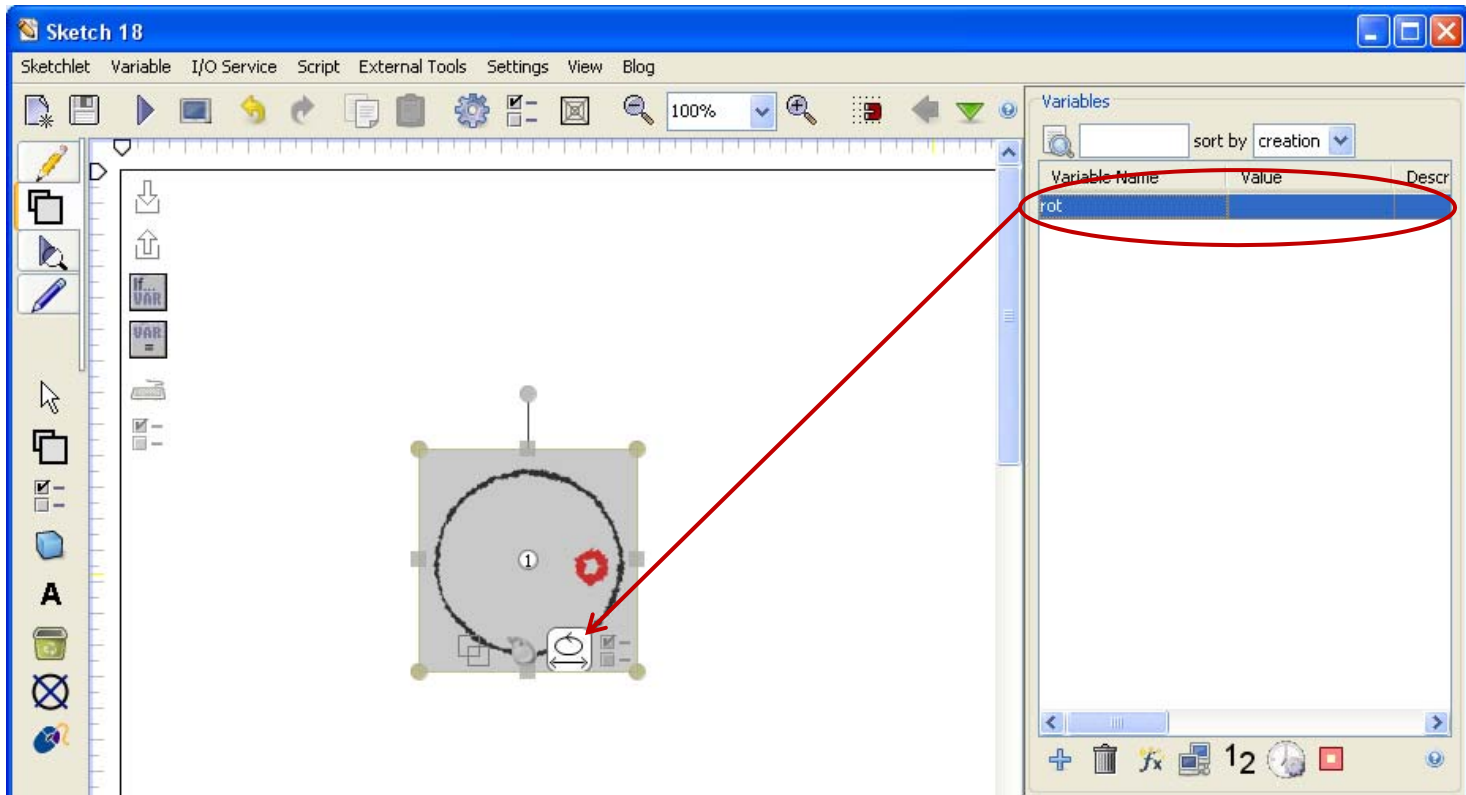


Active regions mode; x=633, y=454 selected region: x=188, y=176, width=138, height=141



# Example: Connecting Region Continuous Mouse Event to Variable

- Step 2: drag-and-drop the variable on the active region motion mapping icon







# Example: Connecting Region Continuous Mouse Event to Variable

- Step 3: select properties of motion that you want to capture in the variable
- In this example it is “rotation”

Dimension: position x  
Variable: position x  
Start value: rotation  
End value: speed  
Prefix:  
Postfix:  
Format:

Dimension	Min	Max
position x		
position y		
rotation		
speed	0	1000

Active regions mode; x=513, y=255 selected region: x=188, y=176, width=138, height=141



# Example: Connecting Region Continuous Mouse Event to Variable

- Step 4: interact with the region and observe variables changes
- In this example, dragging the region with right mouse button will rotate it and update the variable “rot”

The screenshot shows a software interface with a toolbar at the top. The toolbar includes icons for undo, redo, copy, paste, settings, and zoom. A ruler is visible below the toolbar. The main workspace contains a gray square region with a black circular outline and a red gear icon. A blue arrow points from the gear icon to the 'Variables' panel on the right. The 'Variables' panel has a search bar and a 'sort by' dropdown set to 'creation'. Below this is a table with the following data:

Variable Name	Value	Descr
rot	83.25741339328722	



# Example: Connecting Region Continuous Mouse Event to Variable

- You can drag-and-drop the variable on the region properties icon, and select property

The screenshot shows a software interface with a region (a rectangle with a smaller inner rectangle) and a red gear icon in the bottom right corner. A variables panel is open, showing a table with the following data:

Variable Name	value	Descr
rot	83.25741339328722	

A red arrow points from the 'rot' variable in the variables panel to the region properties icon in the bottom right corner of the region.

The 'Set Region Property' dialog box is shown. The 'Set' dropdown is set to 'Rotation' and the 'to' field contains '=rot'. A 'Cancel' button is visible.



# Example: Connecting Region Continuous Mouse Event to Variable

- In this example we are using variable “rot” to control rotation property of another region

The screenshot shows a software interface with a sketch of a region (a rectangle with a smaller rectangle inside) and a 'Variables' panel. The 'Variables' panel has a table with the following data:

Variable Name	value	Descr
rot	83.25741339328722	

A red oval highlights the 'rot' variable and its value. A red arrow points from this oval to a region in the sketch below.

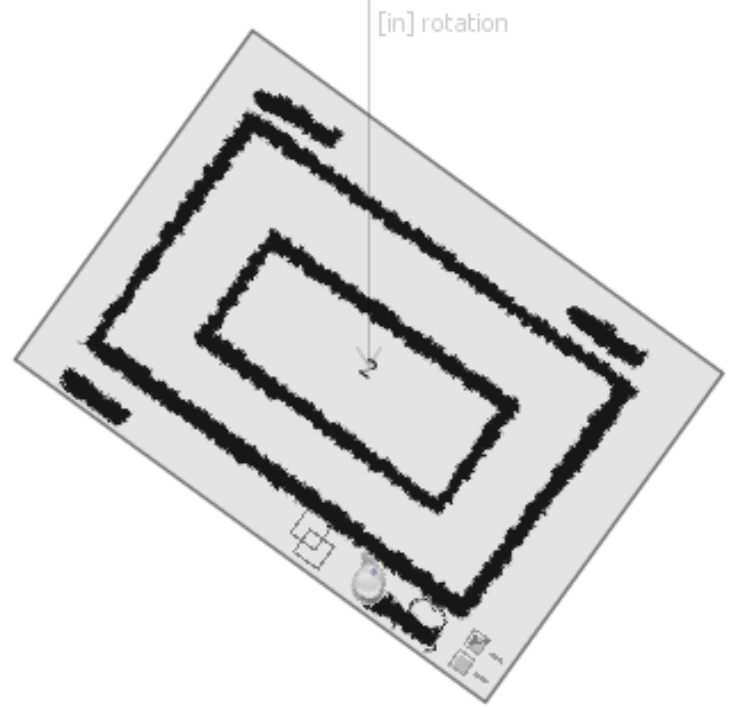
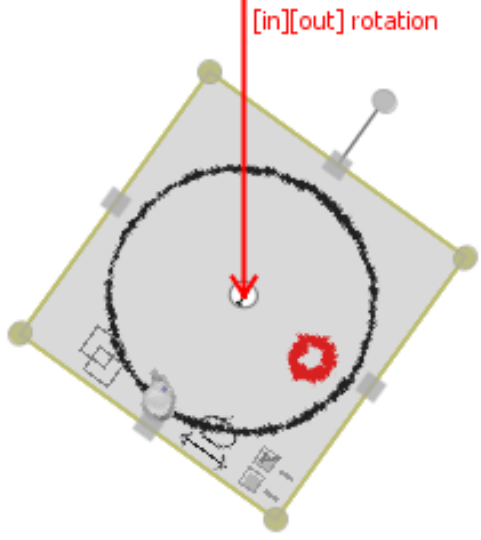
The 'Set Region Property' dialog box shows the following configuration:

- Set: Rotation
- to: =rot
- Buttons: Cancel



rot : 35.962867373302636 >

[YouTube Video](#)





# More Examples

- **Sketch Transitions Based on Mouse Events**
  - [YouTube Video](#)

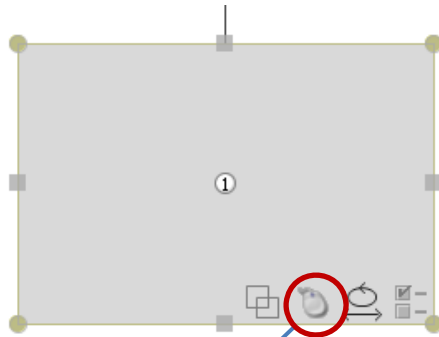
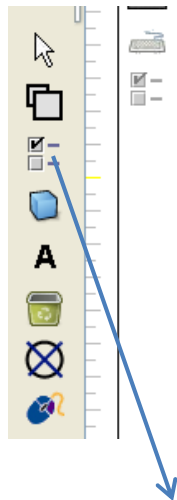


# Directly Connecting Events and Actions

- For each event, you can directly open its settings, and there select actions



# Directly Connecting Events and Actions – Mouse Events



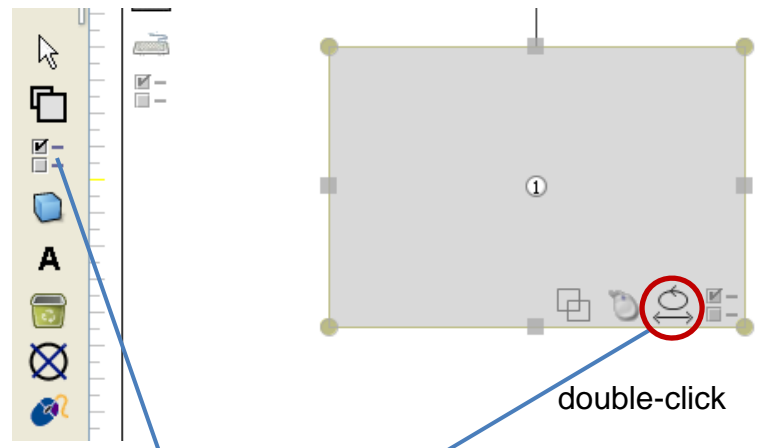
double-click

Image	Mouse Event	Action	Param1	Param2
Properties	Left Button Press	Go to sketch		
Move & Rotate	Left Button Press	Go to sketch		
<b>Mouse Events</b>		Variable update		
Overlap & Touch		Variable append		
Embedded Sketch		Variable increment		
		Start timer		
		Stop timer		
		Start macro		
		Stop macro		





# Directly Connecting Events and Actions – Mouse Continuous Events



1

**Image Properties**

Enable moving by mouse     Enable rotating by mouse

**Move & Rotate**

Motion and rotation limits

Dimension	Min	Max
position x		
position y		
rotation		
speed	0	1000

Trajectory

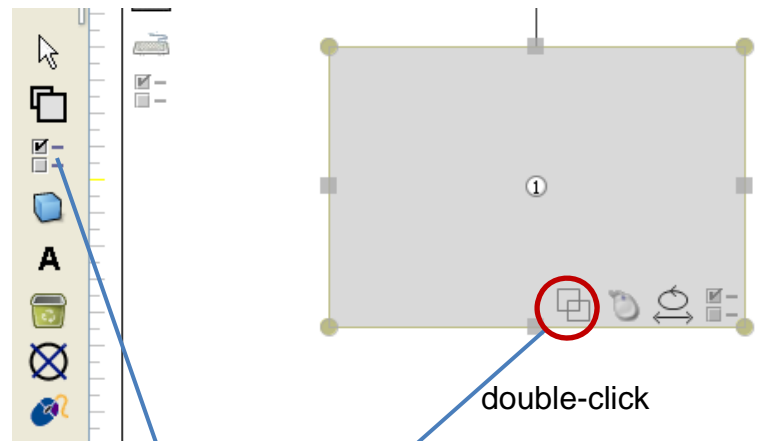
Stick to Trajectory     Control Orientation   

Mapping motion to variable updates

Dimen...	Variable	Sta...	End ...	Prefix	Postfix	Format
position x						
position y						
rotation	wii-vibration					
speed	wii-accel-raw-x					
trajecto...	wii-accel-raw-y					
	wii-accel-raw-z					
	wii-accel-x					
	wii-accel-y					
	wii-accel-z					



# Directly Connecting Events and Actions – Regions Overlap



double-click

1

Make solid (disable walk through)

Region	Event	Action	Param1	Param2
Any region	touches			
		Go to sketch		
		Variable update		
		Variable append		
		Variable increment		
		Start timer		
		Stop timer		
		Start macro		

Delete Duplicate Move Up Move Down Edit

Image Properties Move & Rotate Mouse Events **Overlap & Touch** Embedded Sketch



# Directly Connecting Events and Actions – Sketch Events

On Entry On Exit On Variable Updates On Keyboard Events

Repeat: 1 Complete Blocks Reset Test highlight execution

Double-click on these icons will also open the sketch event settings



# Directly Connecting Events and Actions – Variable Events

On Entry On Exit **On Variable Updates** On Keyboard Events

Variable	operator	value	action	param1	param2
	=				
	>				
	<				
	>=				
	<=				
	>>				
	<<				
	in				
	not in				

if... VAR  
VAR =

Double-click on this icons will also open the variable event settings



# Directly Connecting Events and Actions – Keyboard Events

On Entry On Exit On Variable Updates **On Keyboard Events**

Shift	Ctrl	Alt	Key	Event	Action	Param1	Param2
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		pressed			
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Enter				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Backspace				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tab				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cancel				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Caps Lock				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Esc				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Space				

Double-click on this icons will also open the keyboard event settings