Sketchify Tutorial
Scripting

sketchify.sf.net

Željko Obrenović
z.obrenovic@tue.nl

Scripts

- In Sketchify scripting languages can be used to quickly outline the behavior of sketches
- Scripts are proven, highly productive and simple to learn and use end-user development paradigms
- With such tools designers, who are usually not experienced programmers, can quickly define more complex interaction scenarios, without requiring intensive programming

Scripts

- We currently support several higher-level scripting languages including Javascript, Python and BeanShell
  (experimental support for Groovy, Ruby, TCL, Sleep, Haskell, and Prolog)

Sketchify Extends Scripting Languages

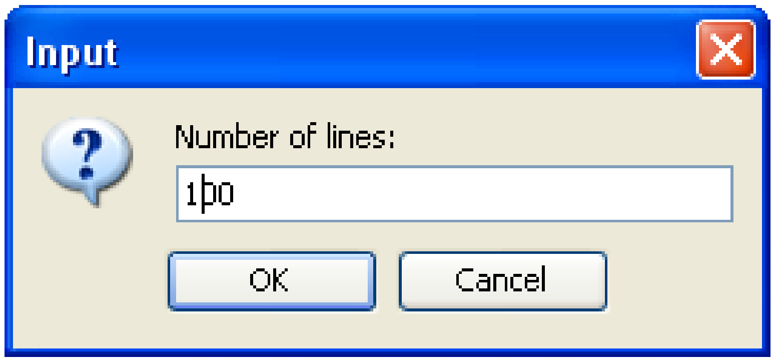
- Sketchify Scripting Extensions
  - Working with Variables
  - Getting User Input
  - Pause and Wait
  - Graphics

Extensions – Working with Variables

- **amico.update**(String variable, String value)
- **amico.update**(String variable, int value)
- **amico.update**(String variable, double value)
- String **amico.get**(String variable)
- String **amico.getString**(String variable)
- int **amico.getInteger**(String variable)
- double **amico.getDouble**(String variable)
- int **amico.getCount**(String variable)
- int **amico.getTimestamp**(String variable)

Extensions – Getting User Input

- String **amico.ask**(String question)
- String **amico.askString**(String question)
- int **amico.askInteger**(String question)
- double **amico.askDouble**(String question)



Extensions – Pause and Wait

- **amico.pause**(double seconds)
- **amico.waitForUpdate**(String variable)
- **amico.waitUntil**(String expression)

Extensions – Graphics

- **amico.clearCanvas**()
- **amico.repaint**()
- **amico.setColor**(int r, int g, int b)
- **amico.setColor**(int r, int g, int b, int transparency)
- **amico.setTransparency**(float transparency)
- **amico.setLineWidth**(double width)
- **amico.setFont**(String name, String style, int size)
- **amico.translate**(int x, int y)
- **amico.rotate**(double angle, int x, int y)
- **amico.scale**(double x, double y)
- **amico.shear**(double x, double y)};

Extensions – Graphics

- **amico.drawText**(String text, int x, int y)
- **amico.drawLine**(int x1, int y1, int x2, int y2)
- **amico.drawRect**(int x, int y, int w, int h)
- **amico.drawEllipse**(int x, int y, int w, int h)
- **amico.drawCircle**(int center_x, int center_y, int r)
- **amico.fillRect**(int x, int y, int w, int h)
- **amico.fillEllipse**(int x, int y, int w, int h)
- **amico.fillCircle**(int center_x, int center_y, int r)
- **amico.drawImage**(String strPathOrURL, int x, int y)
- **amico.drawImage**(String strPathOrURL, int x, int y, int w, int h)
- **amico.getTextWidth**(String text)
- **amico.getTextHeight**(String text)

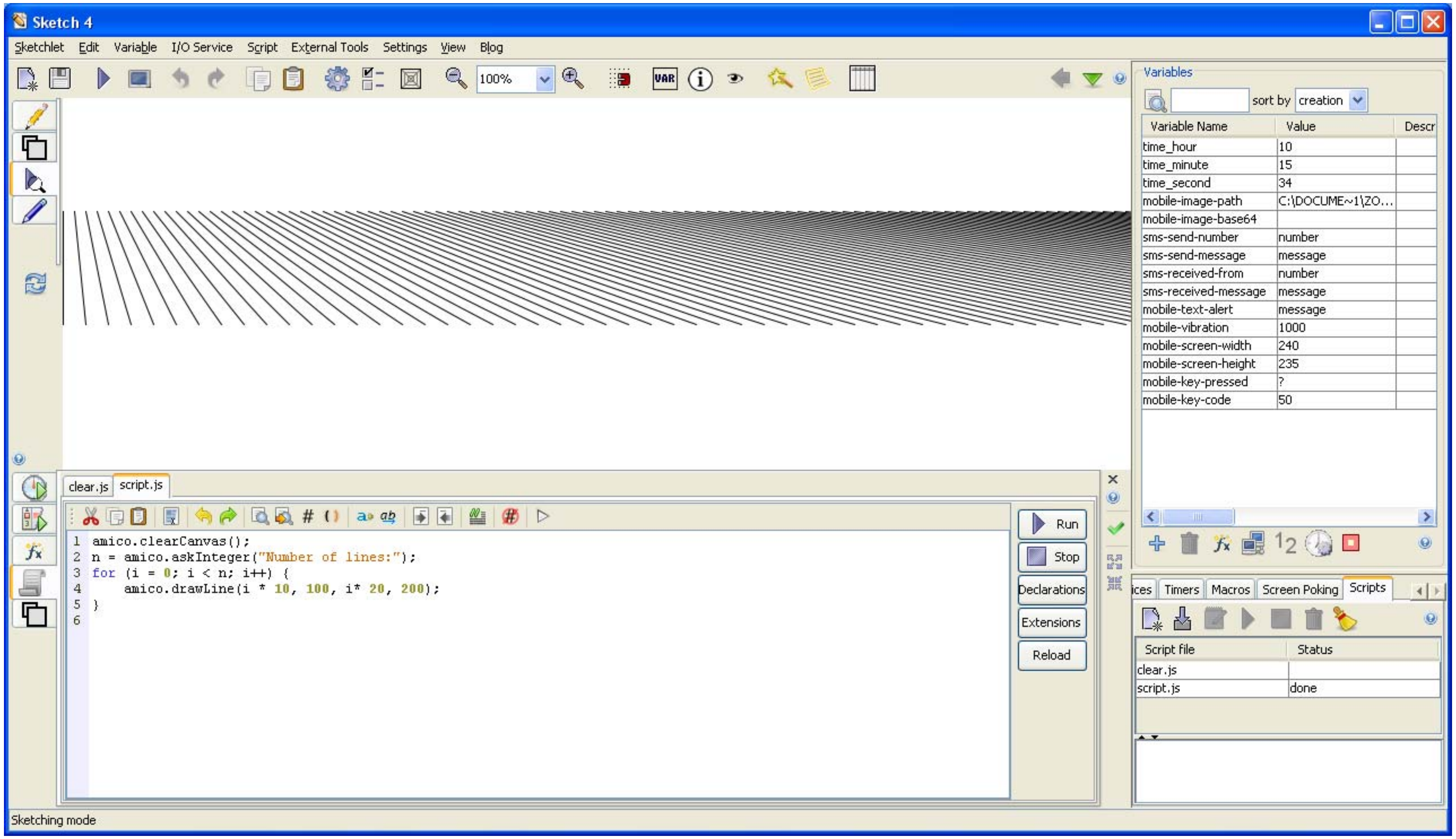
Variable Declarations Inside Scripts

- When a script is called, Sketchify variables will be redeclared within the script
  - Variables may be renamed to satify naming convention of scripting languages

Sketchify Variable Name	Declaration in scripts
position x	position_x
motion-intensity	motion_intensity
a	a

- Read-only, use *amico.update* to change the value of a Sketchify variable

Script Editor

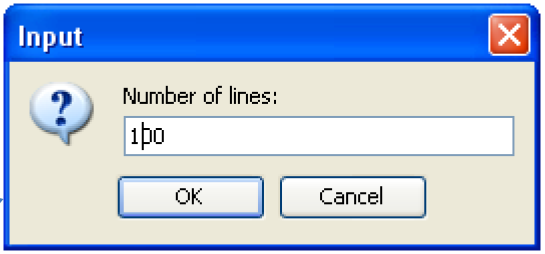

Sketch 4
Sketchlet Edit Variable I/O Service Script External Tools Settings View Blog
100%
clear.js script.js
Run
Stop
Declarations
Extensions
Reload
Sketching mode
Variables
sort by creation
ices Timers Macros Screen Poking Scripts

```
amico.clearCanvas();
n = amico.askInteger("Number of lines:");
for (i = 0; i < n; i++) {
    amico.drawLine(i * 10, 100, i* 20, 200);
}
```

Variable Name	Value	Descr
time_hour	10	
time_minute	15	
time_second	34	
mobile-image-path	C:\DOCUME~1\ZO...	
mobile-image-base64		
sms-send-number	number	
sms-send-message	message	
sms-received-from	number	
sms-received-message	message	
mobile-text-alert	message	
mobile-vibration	1000	
mobile-screen-width	240	
mobile-screen-height	235	
mobile-key-pressed	?	
mobile-key-code	50	

Script file	Status
clear.js	
script.js	done

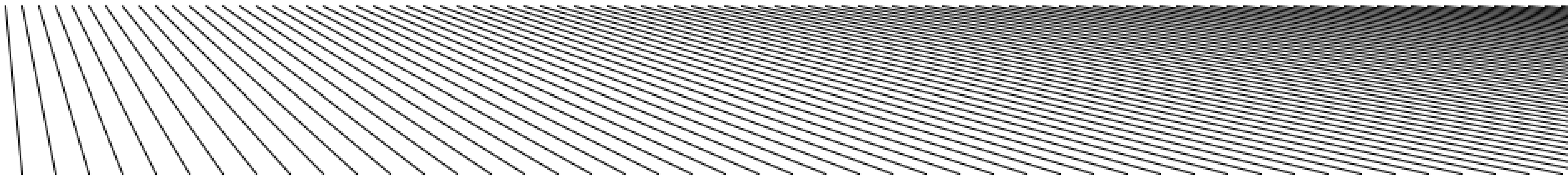
Example


Input

Number of lines:

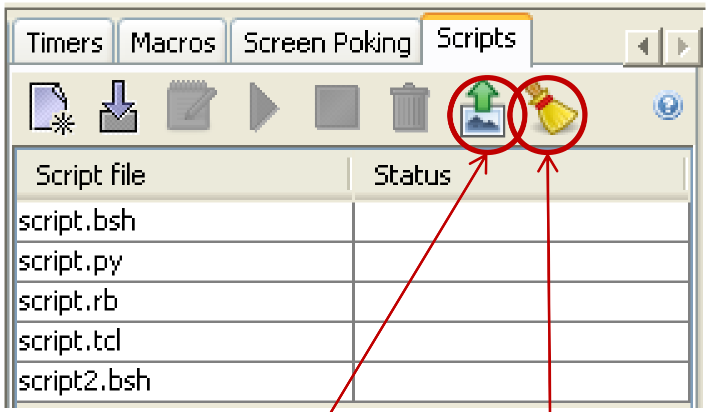
100

```
amico.clearCanvas();
n = amico.askInteger("Number of lines:");
for (i = 0; i < n; i++) {
   amico.drawLine(i * 10, 100, i* 20, 200);
}
```



Merging Image Generated by Scripts and Background Sketch Image

- Scripts draw in a separate layer on top of the sketch
- The image from this layer can be merged with the background sketch image (i.e. it becomes a part of that image)



Script file	Status
script.bsh	
script.py	
script.rb	
script.tcl	
script2.bsh	

Merge the image generated by scripts with the background sketch image

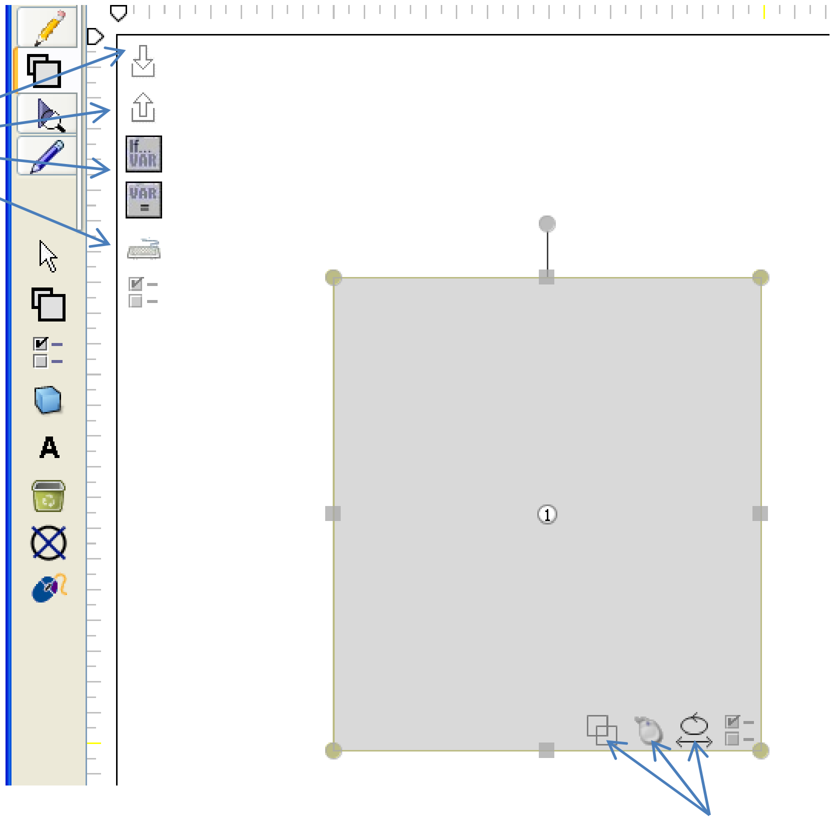
Clear the image generated by scripts

Calling Scripts

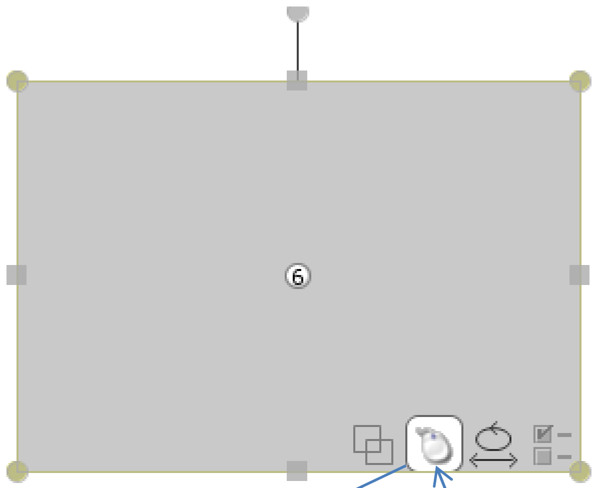
- Scripts can be called from several places
  - On active region mouse events
  - On sketch events (entry or exit)
  - On variable updates ("On Variable Update" actions)
  - On keyboard events
  - From other macros, as one of the commands
- Drag-and-Drop on any sketch or region event
- Directly specify in settings

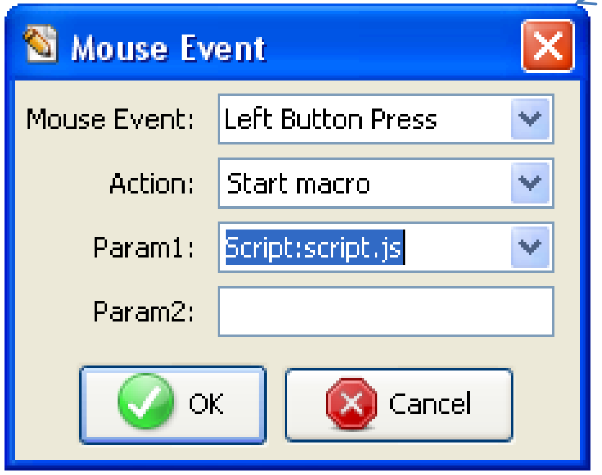
Drop Event Anchors

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.



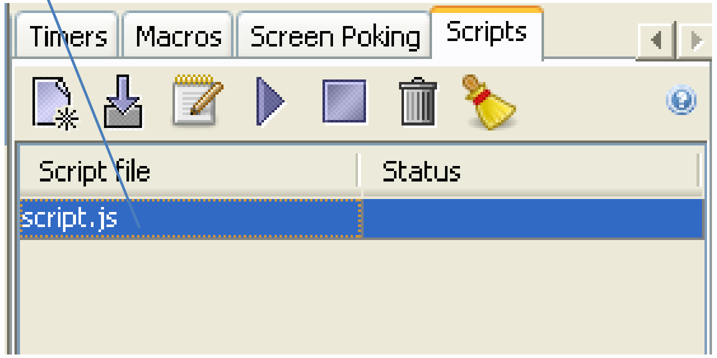
Anchors for connecting region events (region overlap, discrete mouse events, continues mouse events) by drag-and-drop of variables, timers and macros. You can also double-click on these icons to open current settings for these events and properties.





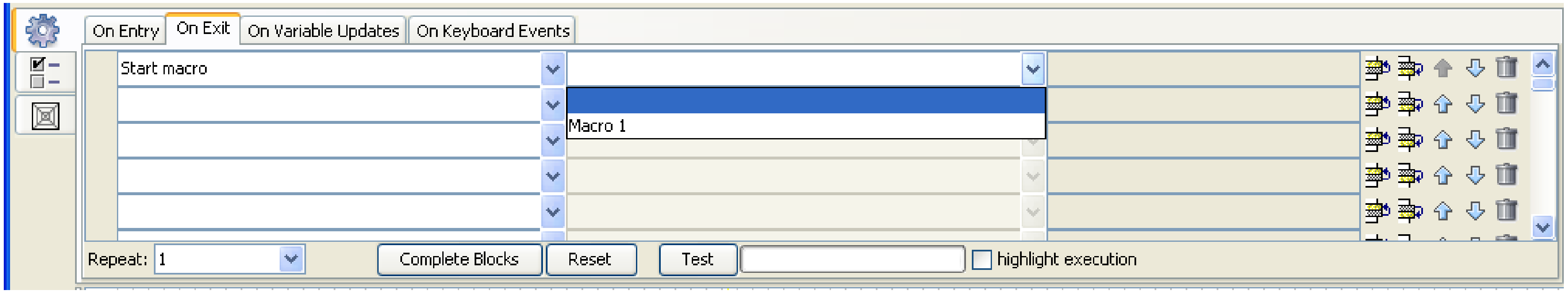
Mouse Event:	Left Button Press
Action:	Start macro
Param1:	Script:script.js
Param2:	

Drag-and-drop of the script
on the mouse event icon of
the active region.



Script file	Status
script.js	

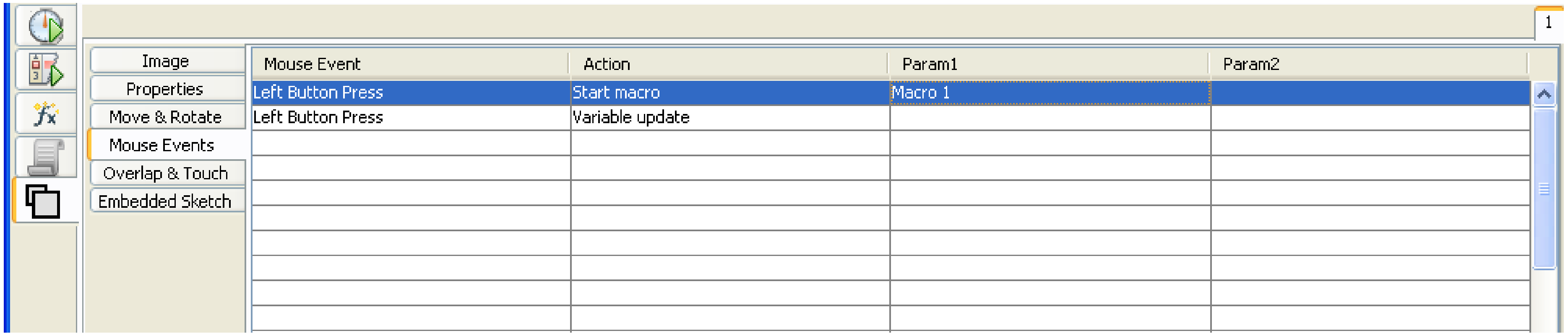
Directly Specify in Settings



Start macro

Macro 1

Repeat: 1



Mouse Event	Action	Param1	Param2
Left Button Press	Start macro	Macro 1	
Left Button Press	Variable update		

To Learn More About Scripting Languages

- JavaScript
  - http://www.w3schools.com/js
  - https://developer.mozilla.org/en/JavaScript
  - https://developer.mozilla.org/en/A_re-introduction_to_JavaScript
- BeanShell
  - http://www.beanshell.org/
- Groovy
  - http://groovy.codehaus.org/
- Python
  - http://www.python.org/
- …