Facility Layout Scripts User Guide

This is a guide to running the python scripts that generate facility layout xml files:

1. generateSingle.py
2. generateMultiple.py

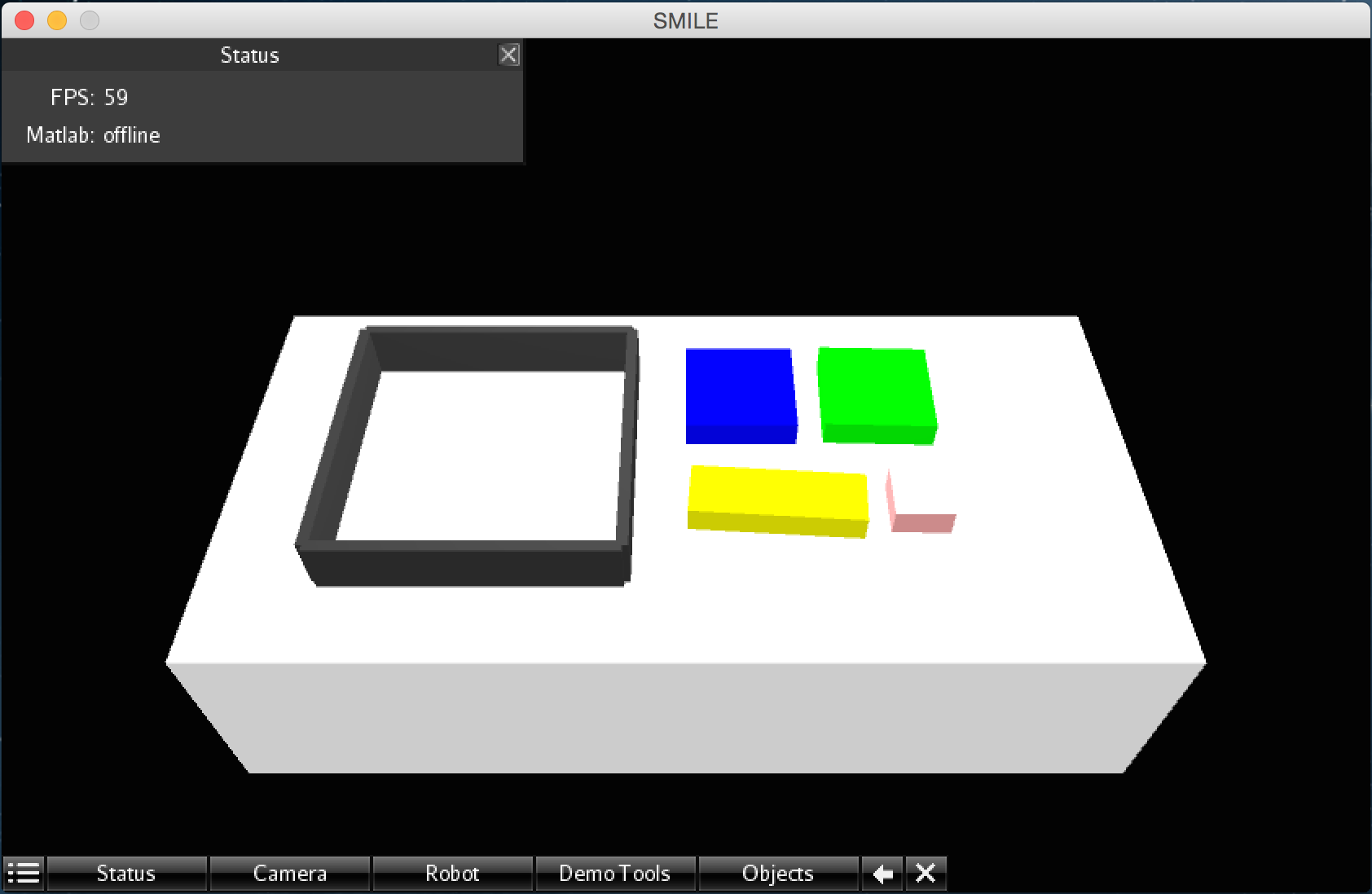
# generateSingle.py

This python script generates a gray enclosure (4 sides, open top and bottom) that marks the boundaries of the actual facility layout. Colored blocks are placed to its side and each represents an area to be put into the facility. Once in the facility, the user can move around the blocks to produce a layout of his/her choice.

**Sample Input (0):**

The input is a simple command line typed into bash/terminal.

python generateSingle.py “room,5,5:paintingRoom,blue,2,2:machineRoom,green,2,2:weldingRoom,yellow,3,1:packingRoom,pink,1,1:facility.xml”



**Fig 1.1: Initial Environment of the facility layout in SMILE after running Sample Input (0)**

**Breaking down the command line arguments:**

(1) First argument must be the actual python script

generateSingle.py

(2) Second argument is a colon (“:”) separated string consisting of n components.

“room,5,5:paintingRoom,blue,2,2:machineRoom,green,2,2:weldingRoom,yellow,3,1:packingRoom,pink,1,1:facility.xml”

Once split,

(2.1) The first component is for the layout

room,5,5

room,<layoutX>,<layoutY>

* + - layoutX is the xspan of the layout
    - layout Y is the yspan of the layout

(2.2) The 2nd to the (n-1)th components are for each area.

paintingRoom,blue,2,2

machineRoom,green,2,2

weldingRoom,yellow,3,1

packingRoom,pink,1,1

<area\_name>, <area\_color>, <area\_xspan>, <area\_yspan>

* + - area\_name is the designated name of the area like paintingRoom, machineRoom, weldingRoom, packingRoom
    - area\_color is the color assigned to the area. Ex: blue for paintingRoom, green for machineRoom
    - area\_xspan is the xpsan of the area. Ex: paintingRoom has xspan of 2, packingRoom has xspan of 1
    - area\_yspan is the ypsan of the area. Ex: weldingRoom and packingRoom have yspan of 1

(2.3) The nth component

facility.xml

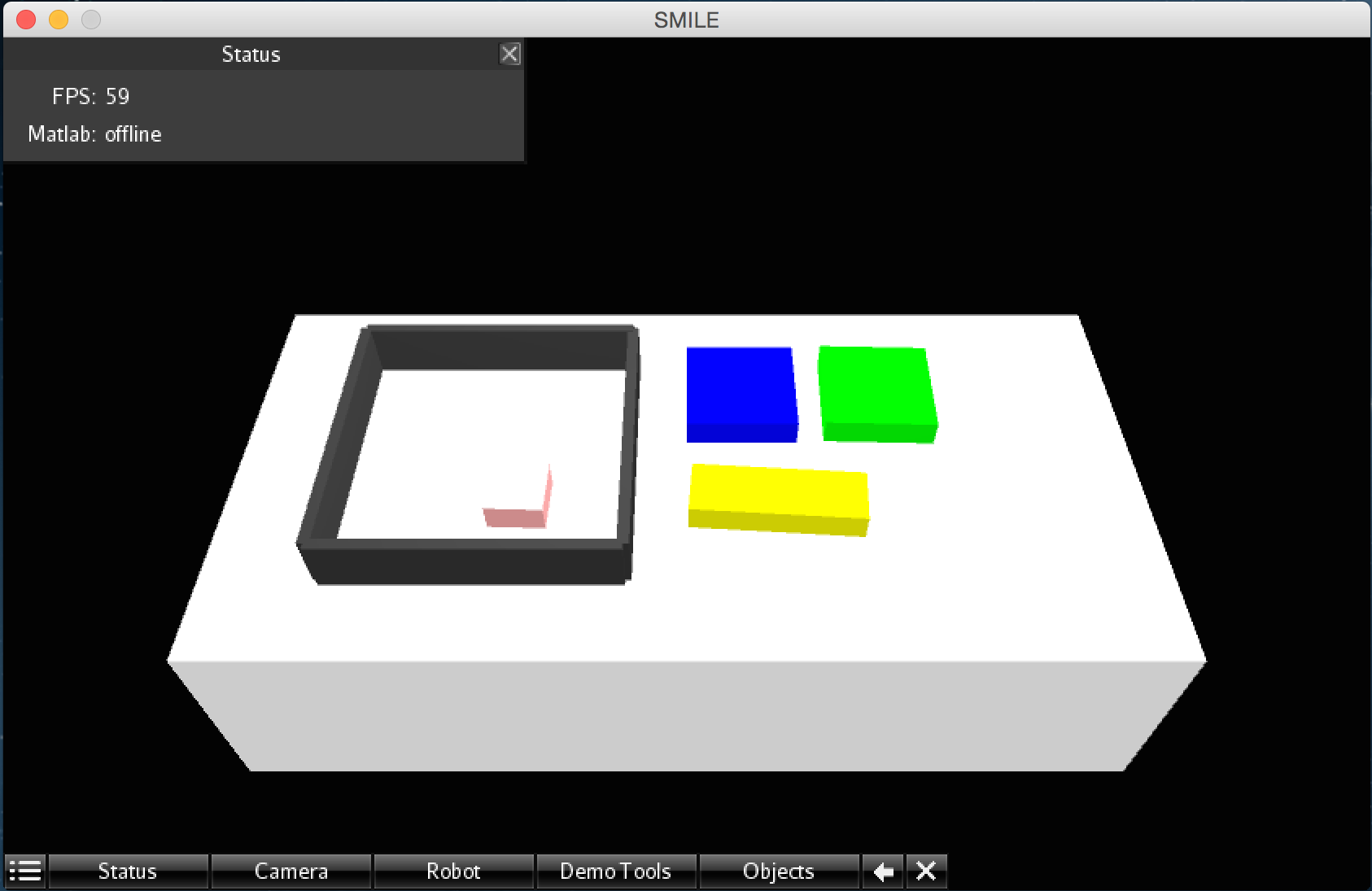
<destinationFilename.xml>

* This holds the name of the destination file, must have .xml extension.

**Integration into SMILE**

* Place the <destinationFilename.xml> into SMILE-1.1.0/ tablesetup.
* On running smile, select <destinationFilename.xml> in the *Objects* tab, and click *Load XML*

Below is an example of an xml file after a block is moved into the facility.



**Fig 1.2 Environment after placing ‘packingRoom’ into the facility**

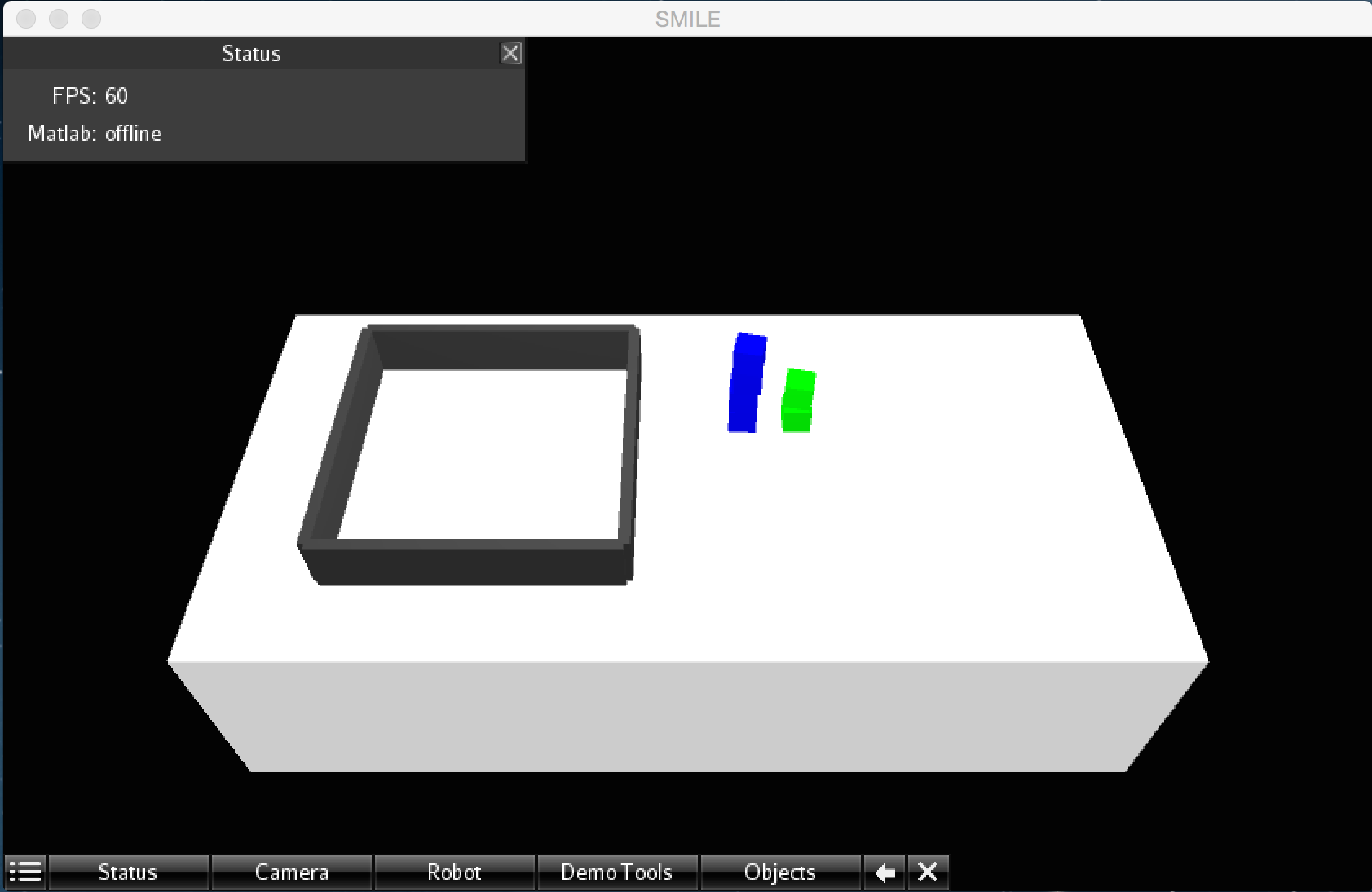
# generateMultiple.py

This python script generates a gray enclosure (4 sides, open top and bottom) that marks the boundaries of the actual facility layout. Towers of colored blocks are placed to its side. Each tower of blocks is an area in the layout. The blocks can be moved around and put together contiguously to form different shapes of a single area. The user is not constrained to one specific shape as before in generateSingle.py.

**Sample Input (1):**

The input is a simple command line typed into bash/terminal.

python generateMultiple.py “room,5,5:paintingRoom,blue,4:machineRoom,green,2:facility.xml”



**Fig 2.1** **Initial Environment of the facility layout in SMILE after running Sample Input (1)**

**Breaking down the command line arguments:**

(1) First argument must be the actual python script

generateMultiple.py

(2) Second argument is a colon (“:”) separated string consisting of n components.

“room,5,5:paintingRoom,blue,4:machineRoom,green,2:facility.xml”

Once split,

(2.1) The first component is for the layout

room,5,5

room,<layoutX>,<layoutY>

* + - layoutX is the xspan of the layout
    - layout Y is the yspan of the layout

(2.2) The 2nd to the (n-1)th components are for each area.

paintingRoom,blue,4

machineRoom,green,2

<area\_name>, <area\_color>, <num\_of\_blocks>

* + - area\_name is the designated name of the area like paintingRoom, machineRoom, weldingRoom, packingRoom
    - area\_color is the color assigned to the area. Ex: blue for paintingRoom, green for machineRoom
    - <num\_of\_blocks> is the number of blocks that need to be generated for that area

(2.3) The nth component

facility.xml

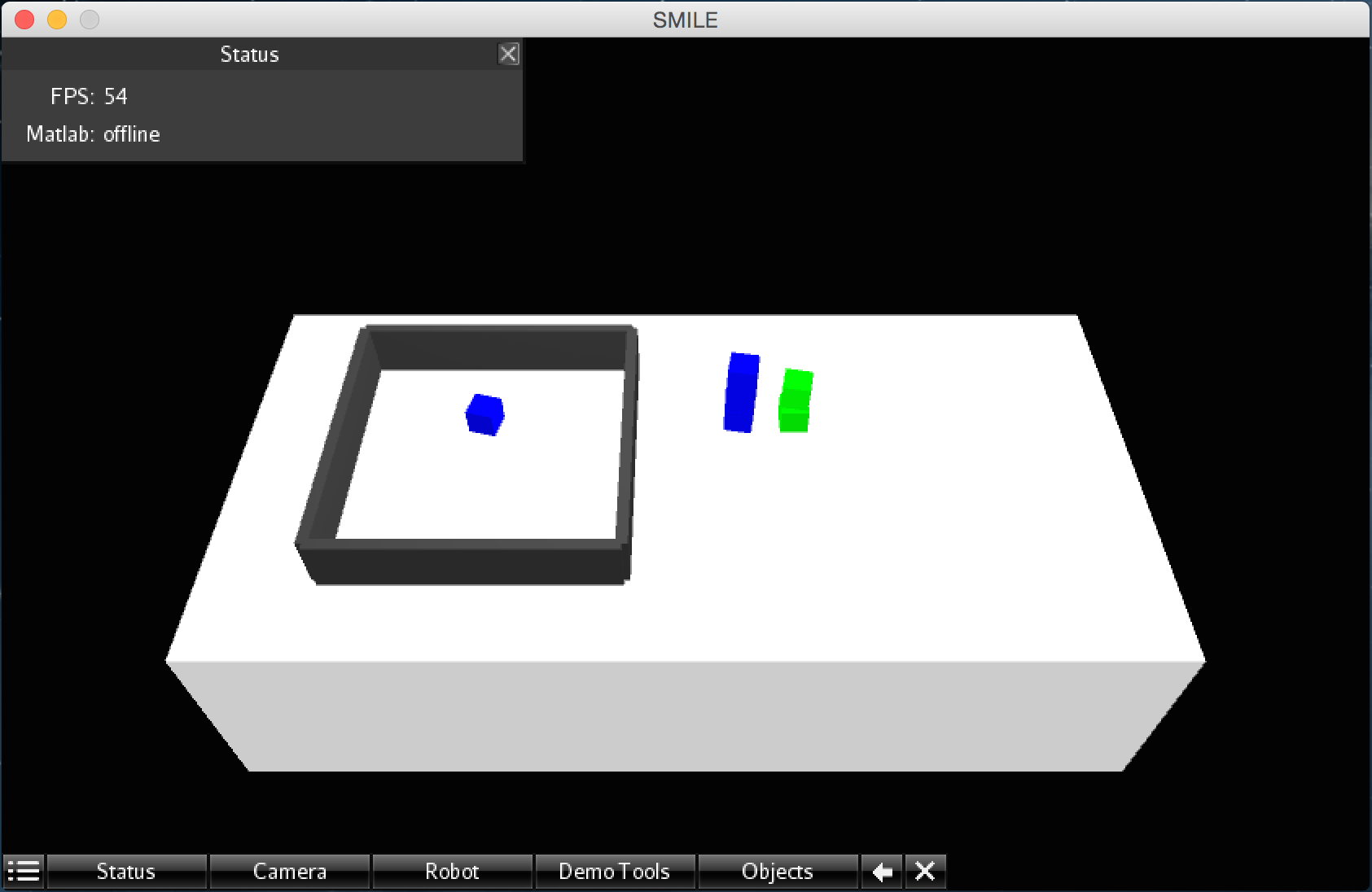
<destinationFilename.xml>

* This holds the name of the destination file, must have .xml extension.

**Integration into SMILE**

* Place the <destinationFilename.xml> into SMILE-1.1.0/ tablesetup.
* On running smile, select <destinationFilename.xml> in the *Objects* tab, and click *Load XML*

Below is an example of an xml file with a block moved into the facility.



**Fig 1.2 Environment after placing one block of ‘paintingRoom’ into the facility**