Architecture Style Grammar

# WALLS PART

<walls height="3">

color-factory-type

texture-type (can be ommited)

</walls>

The building walls are defined by a set of parameters. The height of a floor is defined by the parameter **height**, in meters. The color of the walls is randomly chosen using the color-factory-type. The walls can be given a texture to enhance their appearance using the texture-type.

## COLOR-FACTORY-TYPE

<color-factory>

<hue min="10" max="30" />

<saturation min="0.6" max="0.8" />

<value min="0.8" max="1" />

</color-factory>

The color of the walls is generated randomly inside the three given intervals: **hue** (0 to 360 degrees), **saturation** (0 to 1) and **value** (0 to 1).

These parameters can be easily defined by inspection using a photo editing package such as Photoshop, Paint.NET or Gimp (the last ones free).

## TEXTURE-TYPE

<texture type="stucco" side-size="0.1" />

The texture to apply to the walls has two parameters: **type** chooses the image to load, **side-size** defines the scaling to apply when mapping the texture to the wall.

# ROOF PART

TODO

# ATTACHMENTS PART

## AttachmentS-TYPE

<attachments>

List of floor-span-TYPEs

</attachments>

This is the attachment façade filling root element.

When generating the façade, the system can apply different rules according to the floor number which is being generated. For instance, the ground floor (which is floor 1) can have a centered door and windows in the remaining space, while floor 3 can have balconies occupying the whole façade.

This element features a list of floor-span-types, each defining a set of rules to apply to the selected extent.

Attachments-type must exist one and one time alone, having at least the default floor-span-type, that is, one that spans the all the floors.

## FLOOR-SPAN-TYPE

<floor-span from=”1” to=”100” *at=”3”*>

ATTACHMENT-FACTORY-TYPE  
or  
ATOMIC-ATTACHMENT-TYPE

</floor-span >

This element selects the extent of floors to which its child has application.

It no arguments are given, **from** defaults to 1 and **to** defaults to 100. This means every floor from the 1st to the last one get a similar façade, as defined by its child element.

If only one floor is to have this aspect, instead of from and to having that floor number, one can use at (instead of from=3 to=3, using at=3).

## Attachment-FACTORY-type

randomize-attachments-FACTORY-TYPE

or

boundary-attachments-FACTORY-TYPE

A factory generates instances of its children according to a defined policy.

**Randomize** fills the façade with instances of its children. Each child has an associated probability of being spawned, so that some can appear more frequently than others.

**Boundary** attachments either spawns one instance at the left and right limits of the façade, occupying the rest with instances of the center part (**priority**=center) or spawns one instance at the center part, occupying the remaining façade with instances of left and right (**priority**=boundaries).

### Randomize-attachments-FACTORY-TYPE

<randomize-attachments –factory odds=”0.5 0.3 0.2”>

list of ATOMIC-ATTACHMENT-TYPEs

</randomize-attachments -factory>

The randomize factory populates the available façade space with instances of each ATOMIC-ATTACHMENT-TYPE defined has children.

The parameter **odds** is a list of float numbers from 0 to 1 with the probability of each ATOMIC-ATTACHMENT-TYPE being created. Therefore there must be the same number of odds in the list as there are ATOMIC-ATTACHMENT-TYPEs and the sum of their values must be 1.

### Boundary-attachments-TYPE

<boundary-attachments-factory priority=”center/boundaries”>

Left boundary ATOMIC-ATTACHMENT-TYPE

Center ATOMIC-ATTACHMENT-TYPE

Right boundary ATOMIC-ATTACHMENT-TYPE

</boundary-attachments-factory>

The boundary factory gives priority to spawning either the center part or the boundaries, according to the **priority** parameter.

The list of ATOMIC-ATTACHMENT-TYPEs must have three elements, being the first one the left boundary, followed by the center one and with the right one ending the set.

## ATOMIC-Attachment-type

SEQUENCE -OF-ATTACHMENTS-TYPE

or

SIMULTANEOUS-ATTACHMENTS-TYPE

or

ATTACHMENT-INSTANCE-TYPE

An atomic attachment is one which has its spacing defined from the start, unlike the Attachment-FACTORY-type, which spawns attachment instances to the available space.

### SEQUENCE-OF-ATTACHMENTS-TYPE

<sequence-of-attachments>

List of ATOMIC-INSTANCE-TYPEs

</sequence-of-attachments>

The children of this element are generated in the order they are defined.

The element occupies the sum of its children (spacing).

### SIMULTANEOUS-ATTACHMENTS-TYPE

<simultaneous-attachments spacing=”3”>

List of ATTACHMENT-INSTANCE-TYPEs

</simultaneous-attachments>

This element grants the grammar creator the capability of applying more than on attachment instance to the same façade space. This can useful for the creation of compound structures such as a door with a balcony.

The **spacing** parameter of its children is ignored, with its own **spacing** parameter being applied instead.

### ATTACHMENT-INSTANCE-TYPE

<attachment-instance type=”window” spacing=”3” />

The attachment instance is the most basic element available. All of the attachment grammar tree leaves must be of this type.

The **type** parameter allows the user to select the appearance of the instance, while **spacing** defines the horizontal space it occupies.