bdGenerator

Release 1.0.0

Benjamin Danneville

Oct 14, 2021

CONTENTS:

CHAPTER

ONE

INSTALLATION

Overview

This will cover how to install the bdGenerator script
on Maya 2020 (Python 2) and Maya 2022 (Python 3).

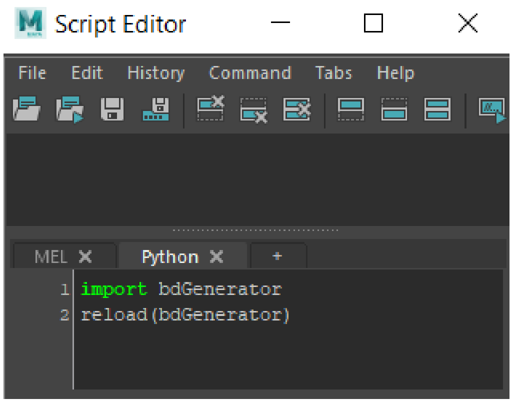
1.1 Installation of the script

1- Download bdGenerator.py and put it into the folder Documents/maya/scripts/

1.1.1 Maya 2020 (Python 2)

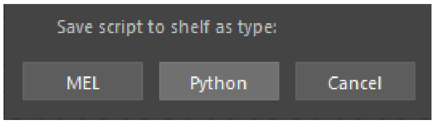
2- Copy/paste this into your python script editor in Maya

```
import bdGenerator
reload(bdGenerator)
```



```
import bdGenerator
reload(bdGenerator)
```

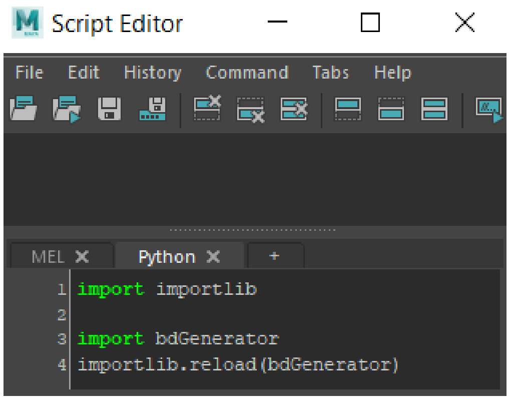
3- Select everything and drag it with the left mouse button onto your shelf and select python


Save script to shelf as type:

1.1.2 Maya 2022 (Python 3)

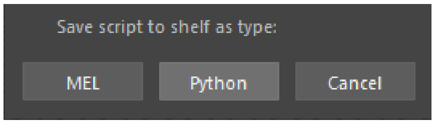
2- Copy/paste this into your python script editor in Maya

```
import importlib

import bdGenerator
importlib.reload(bdGenerator)
```



```
import importlib

import bdGenerator
importlib.reload(bdGenerator)
```

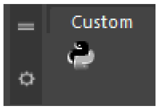
3- Select everything and drag it with the left mouse button onto your shelf and select python



Save script to shelf as type:

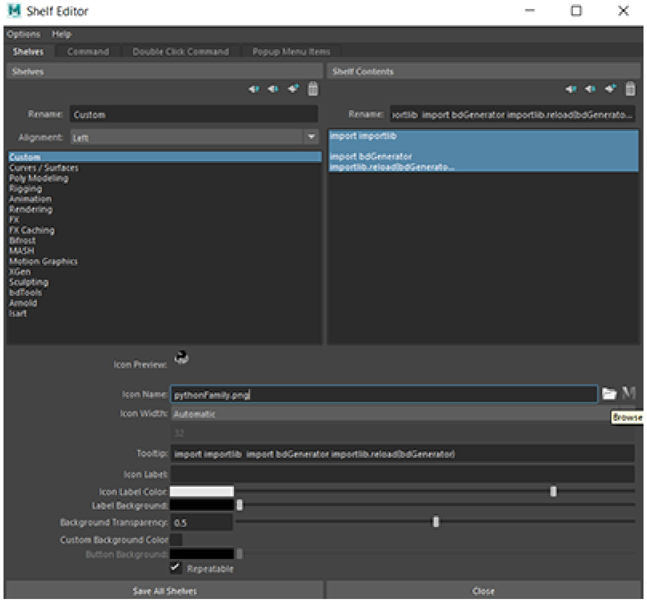
1.2 Adding the icon

You should now have the basic python icon in the shelf that represents your script



Now, if you want to have a more fancy icon, download my bdGenerator Icon

Now to replace the python icon by this one, just right click on the button and hit **edit**
In the **"Shelves"** tab, click on the Icon Name folder and go find the icon you just downloaded



Rename: Custom

Alignment: Left

Custom
Curves / Surfaces
Poly Modeling
Rigging
Animation
Rendering
FX
FX Caching
Bifrost
MASH
Motion Graphics
XGen
Sculpting
bdTools
Arnold
Isart

Rename: ›ortlib import bdGenerator importlib.reload(bdGenerato...

```
import importlib

import bdGenerator
importlib.reload(bdGenerato...
```

Icon Preview:

Icon Name: pythonFamily.png

Icon Width: Automatic

32

Tooltip: import importlib import bdGenerator importlib.reload(bdGenerator)

Icon Label:

Icon Label Color:

Label Background:

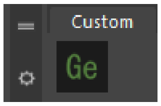
Background Transparency: 0.5

Custom Background Color

Button Background:

☑ Repeatable

Now it should look like this !



The installation of the script is done

CHAPTER

TWO

MAYA SCENE REQUIREMENT

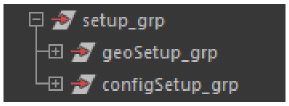
Overview

2.1 Setup

The setup is composed of meshes and configurations.
The meshes correspond to the final 3D models that will replace the blocking.
The configurations allows you to set how the meshes
assembles together to form the final 3D model.

I recommend you two things :

- Have the Setup and the Blocking on two different maya scenes, so that you can just import the Setup into the Blocking scene and then delete it after you've generated you're final scene
- Group everything under this organization where `geoSetup_grp` contains your Meshes and `configSetup_grp` contains your Configuration groups



```
⊟ setup_grp
  ⊞ geoSetup_grp
  ⊞ configSetup_grp
```

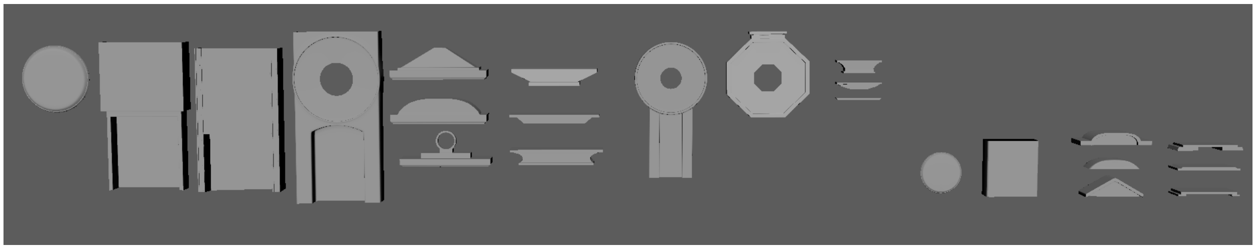
2.1.1 Meshes

- Each mesh has a name easy to identify
- Each pivot has been carefully placed

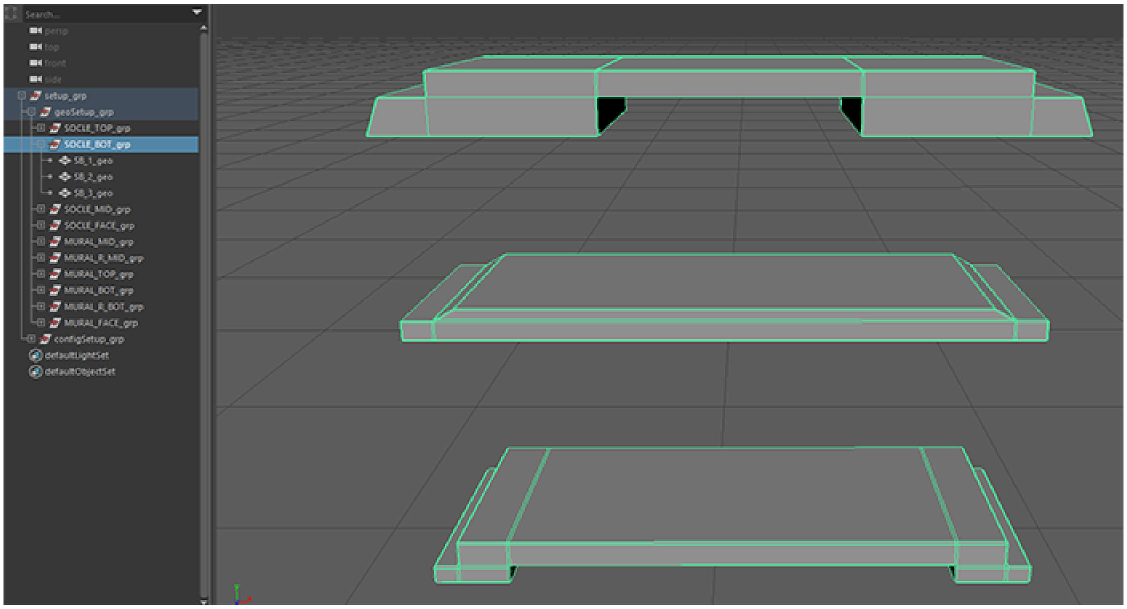
Each mesh has a name easy to identify :
We'll use the name of our meshes in our configurations' locator,
So the easiest it is to identify our mesh, the better

Each pivot has been carefully placed :
When the script will assembles all the meshes together,
it will uses the pivot point to move our mesh

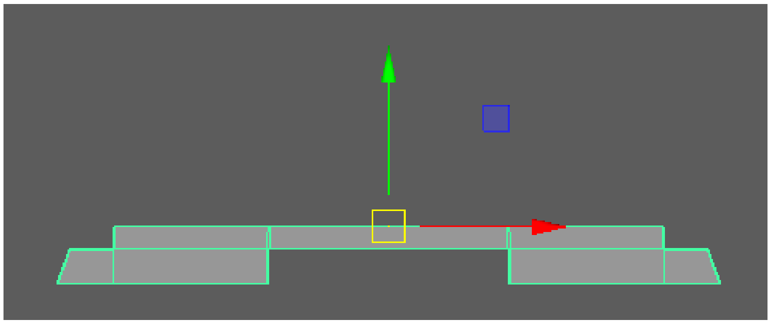
We'll take the exemple of using bdGenerator for generating random clocks,
So here you can see all our Meshes

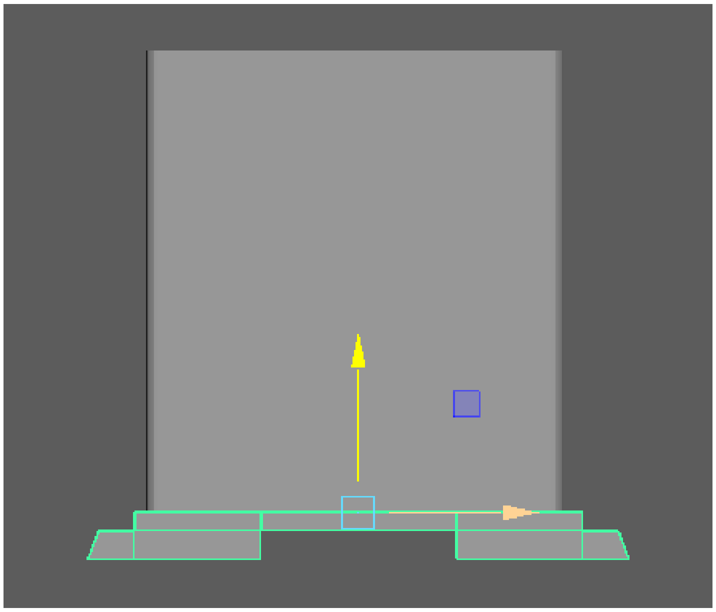


The emplacement of your meshes in the outliner doesn't matter,
but I recommend you to group each variations together.
For exemple, here we have all the variations of the bottom of a clock
inside a group inside `geoSetup_grp`.



Here for instance, the pivot point of each bottom variations is placed at the top,
so that when bdGenerator will move it, it will "snap" to the bottom of our core mesh





2.1.2 Configurations

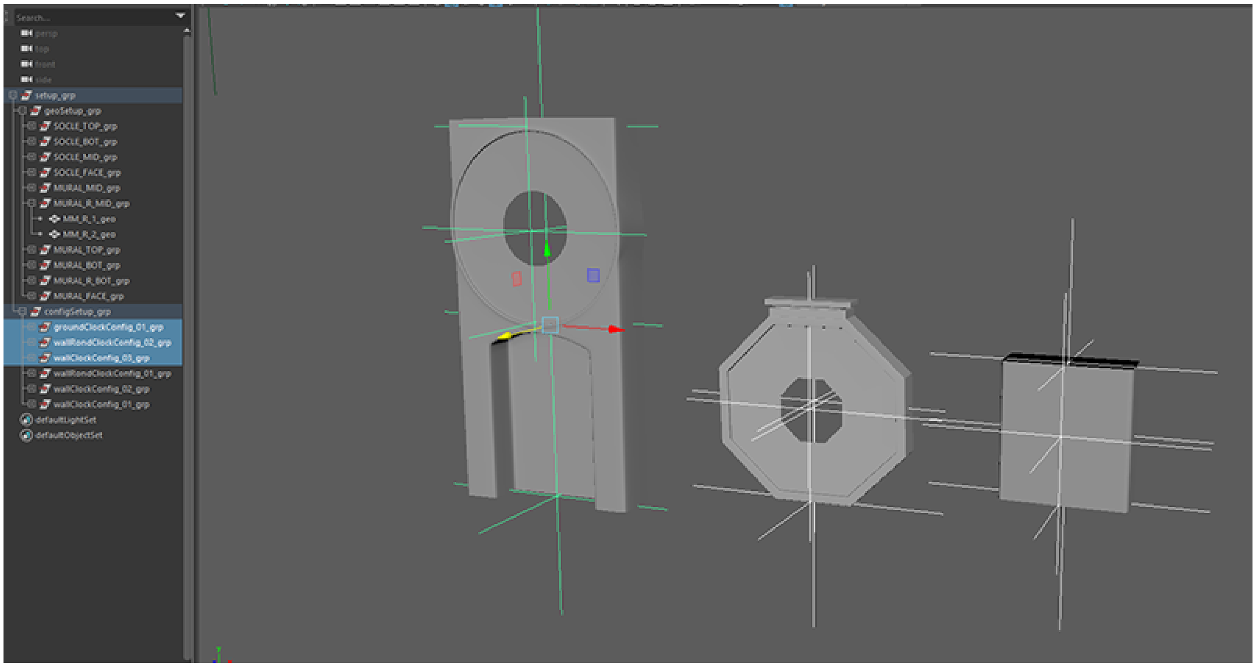
- Configuration groups only contains locators
- Each locators must have a string attribute named “Variation_obj”

Configuration groups only contains locators :
It’s important because otherwise, the script can not make the difference between a configuration group and a blocking group

Each locators must have a string attribute named “Variation_obj” :
The `Variation_obj` attribute will be the emplacement where we will put the name of the meshes that the locator can take. It will take a random mesh from the list and place it at the locator’s position.

Purpose of the configurations

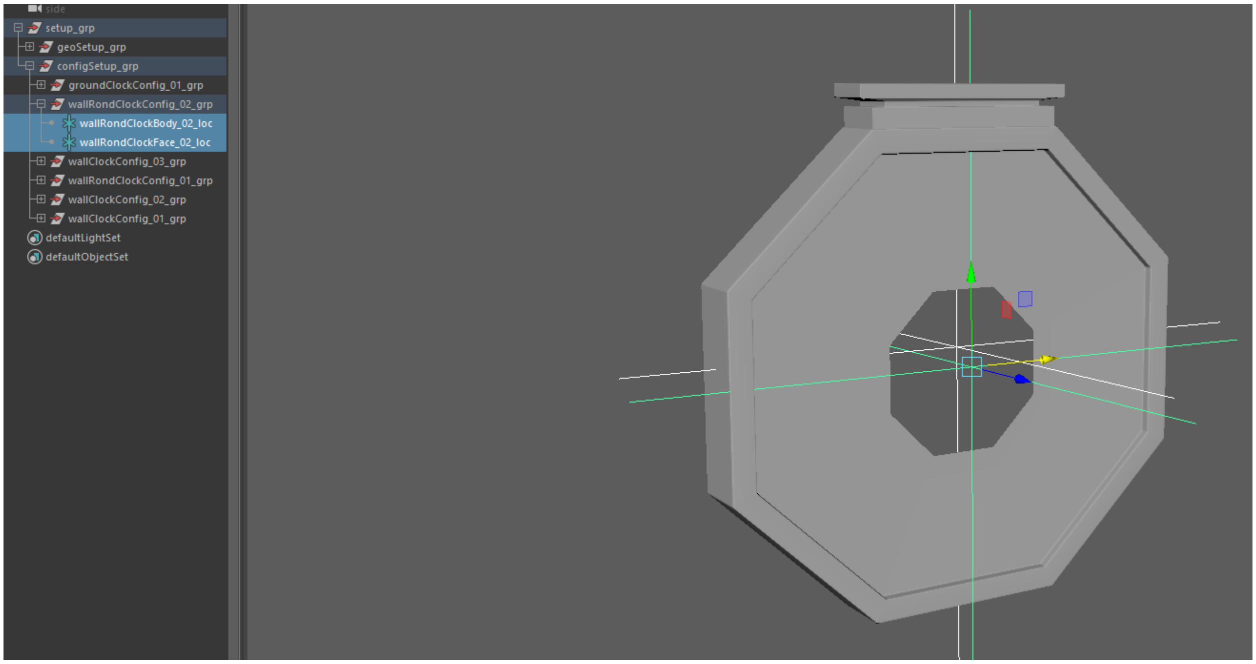
Having configurations allows you to have different setups so that you can have controls over what you are generating. For exemple if we continue on our clock exemple,
we could have a configuration for grounded clocks and walled clocks.
But we could go further, if you wanted for instance a circular clock,
which doesn’t need any top variations or even bot variations,
you could create another configuration group which doesn’t have a bot and a top locator.



Basically, you need a new configuration each time you need to move locators or when you want to add or removes locators.

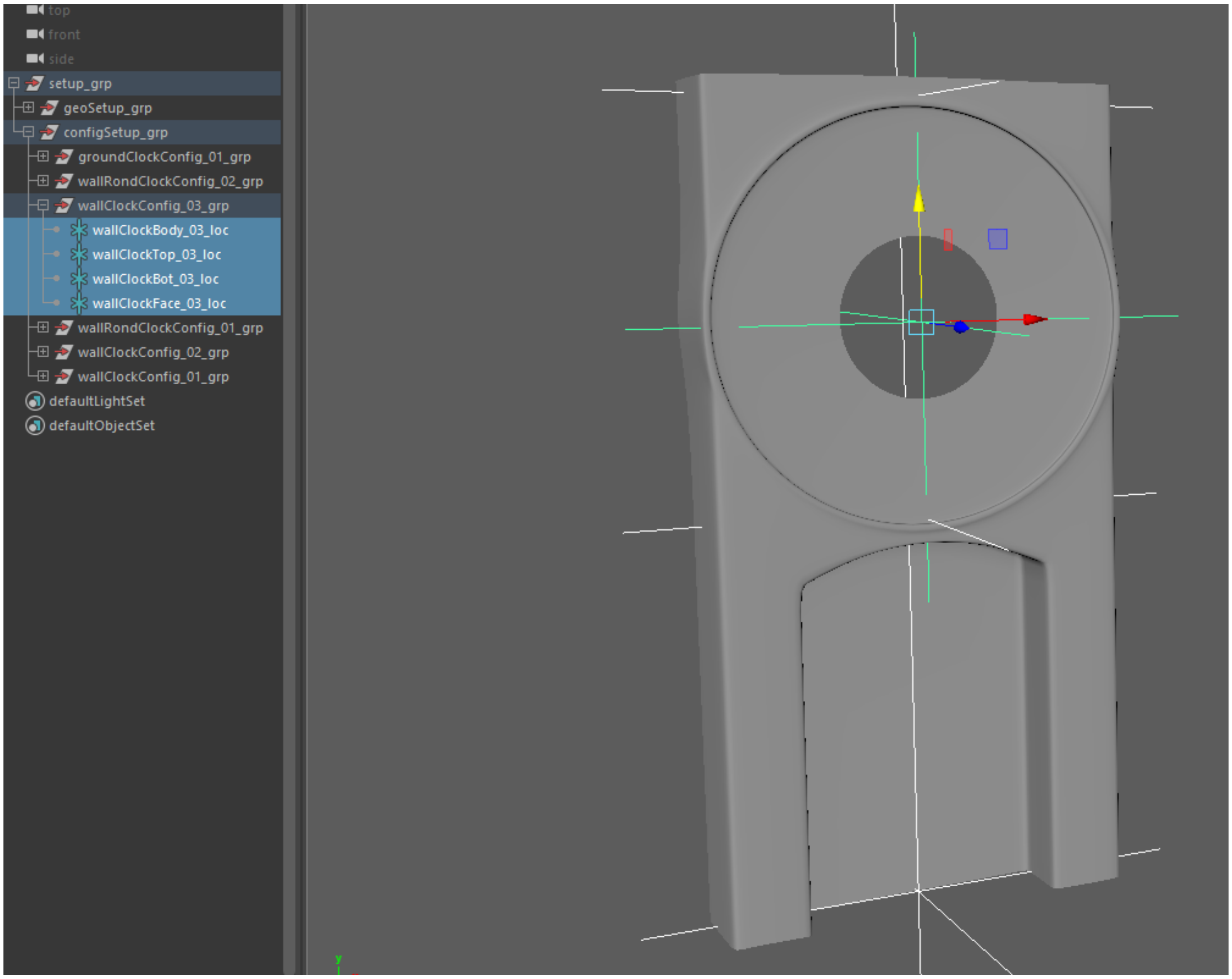
Purpose of the locators

Locators corresponds to parts of our final object.
Continuing with our clocks. You could have a clock with only a body, and a face which would correspond to a configuration with two locators,
one corresponding to the body, and the other to face



As you can see, **the easiest way to place our locator is to place it over our meshes**

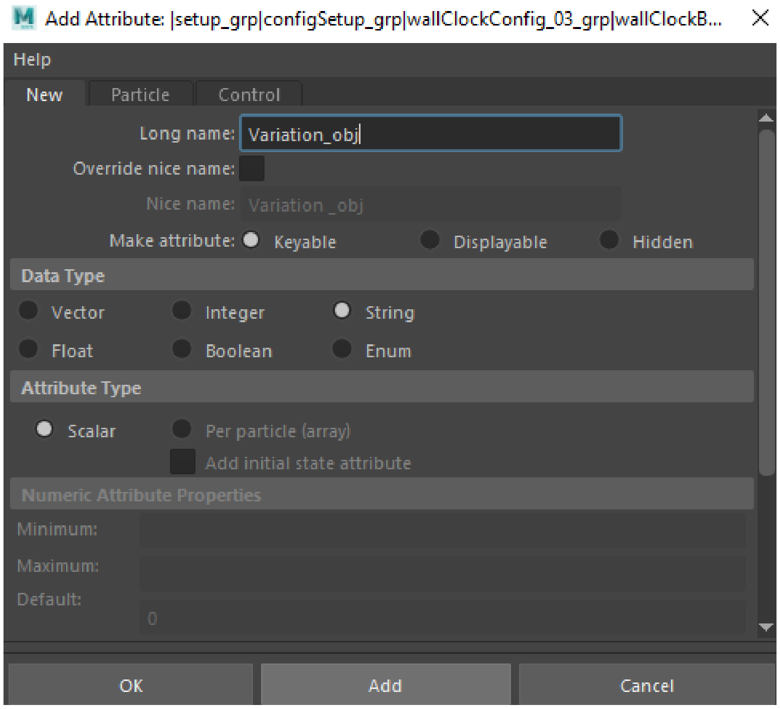
Now if we wanted a clock with a body, a face, a top and a bottom,
we would need to have 4 locators



The string attributes Variation_obj

As we said, the attribute `Variation_obj` makes the link between the locator and the meshes.

To add it, select the desired locator and go to Modify > Add Attribute. Choose **String**, enter the Long name `Variation_obj` and you can **Add**


Add Attribute: |setup_grp|configSetup_grp|wallClockConfig_03_grp|wallClockB...

Long name: Variation_obj

Override nice name: □

Nice name: Variation _obj

Make attribute: ● Keyable ○ Displayable ○ Hidden

Data Type

○ Vector ○ Integer ● String

○ Float ○ Boolean ○ Enum

Attribute Type

● Scalar ○ Per particle (array)

□ Add initial state attribute

Numeric Attribute Properties

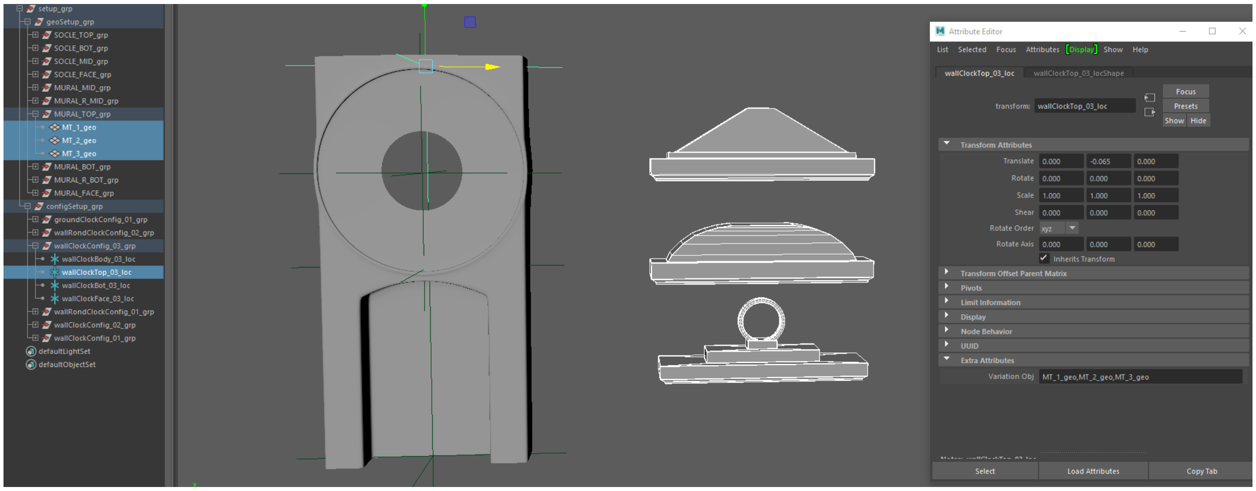
Minimum:

Maximum:

Default: 0

Now to link this to our meshes, just add the meshes that you want in the attribute like this `Variation_1_geo,Variation_2_geo,Variation_3_geo`
The meshes must be separated by only a **comma**.

Now for exemple if we want the top locator to take the top Meshes as variation
So that when it will generate it will choose between the different top meshes,
just add the meshes' name to the top locator's Variation_obj attribute



transform: wallClockTop_03_loc

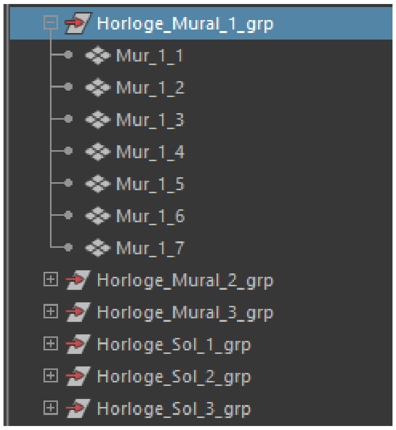
Translate	0.000	-0.065	0.000
Rotate	0.000	0.000	0.000
Scale	1.000	1.000	1.000
Shear	0.000	0.000	0.000
Rotate Order	xyz		
Rotate Axis	0.000	0.000	0.000

☑ Inherits Transform

Variation Obj MT_1_geo,MT_2_geo,MT_3_geo

2.2 Blocking

- Blocking groups only contains meshes
- Every meshes has a different names
- Blocking groups must be named with "_grp" at the end
- Meshes has not been freeze transformed



```
⊟ Horloge_Mural_1_grp
    Mur_1_1
    Mur_1_2
    Mur_1_3
    Mur_1_4
    Mur_1_5
    Mur_1_6
    Mur_1_7
⊞ Horloge_Mural_2_grp
⊞ Horloge_Mural_3_grp
⊞ Horloge_Sol_1_grp
⊞ Horloge_Sol_2_grp
⊞ Horloge_Sol_3_grp
```

Blocking groups only contains meshes :
It's important because otherwise, the script can not make the difference between a blocking group and a configuration group

Blocking groups must be named with ``_grp`` at the end :
bdGenerator will use the Blocking groups name to rename the groups that will contains the final models assembles from the meshes

Meshes has not been freeze transformed :
The group of meshes that will be created by the script will be placed
at the position and orientation of the blocking meshes' pivot.
So if you want the generation to be in the right orientation,
please don't freeze transform your blocking

CHAPTER

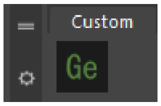
THREE

USING THE SCRIPT

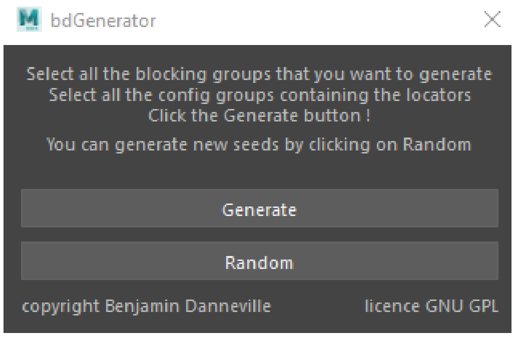
Overview

3.1 Launching

Once you're setup has been created and you have you're blocking, you're ready to go.
If you installed it right, just click on the button



It should open this



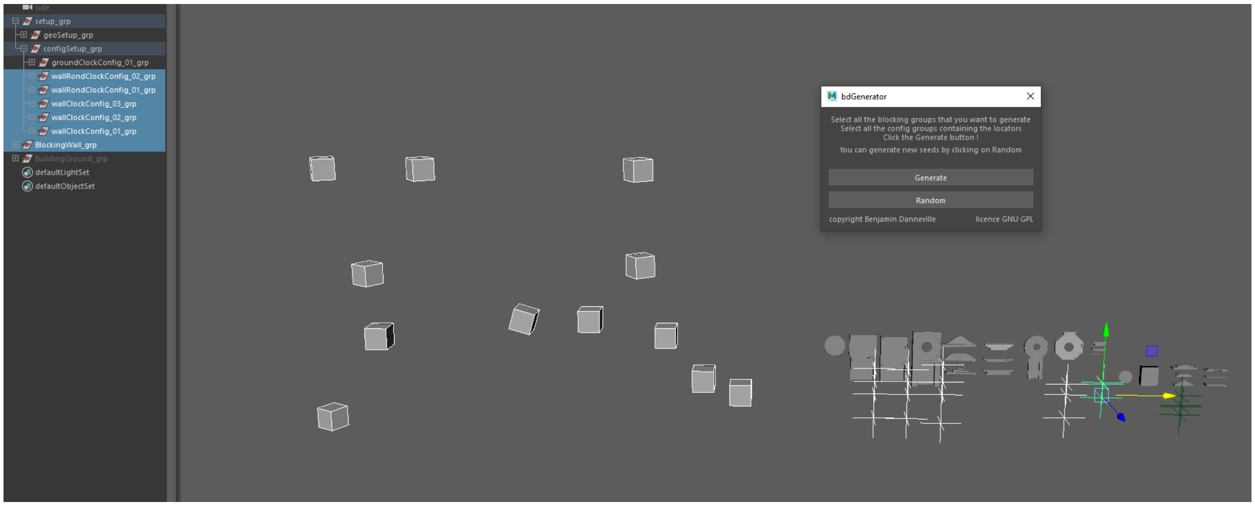
Select all the blocking groups that you want to generate
Select all the config groups containing the locators
Click the Generate button !

You can generate new seeds by clicking on Random

copyright Benjamin Danneville licence GNU GPL

3.2 Selection

Now to use the script, you must select some blocking groups and some configurations groups.
You can select one or multiple blocking and configuration groups,
but you don't have to select every blocking and configurations groups.
For instance if we continue our clock exemple, you would like to only generate the wall clocks,
then select your wall clocks blocking group and only your wall clock Configurations



Select all the blocking groups that you want to generate
Select all the config groups containing the locators
Click the Generate button !

You can generate new seeds by clicking on Random

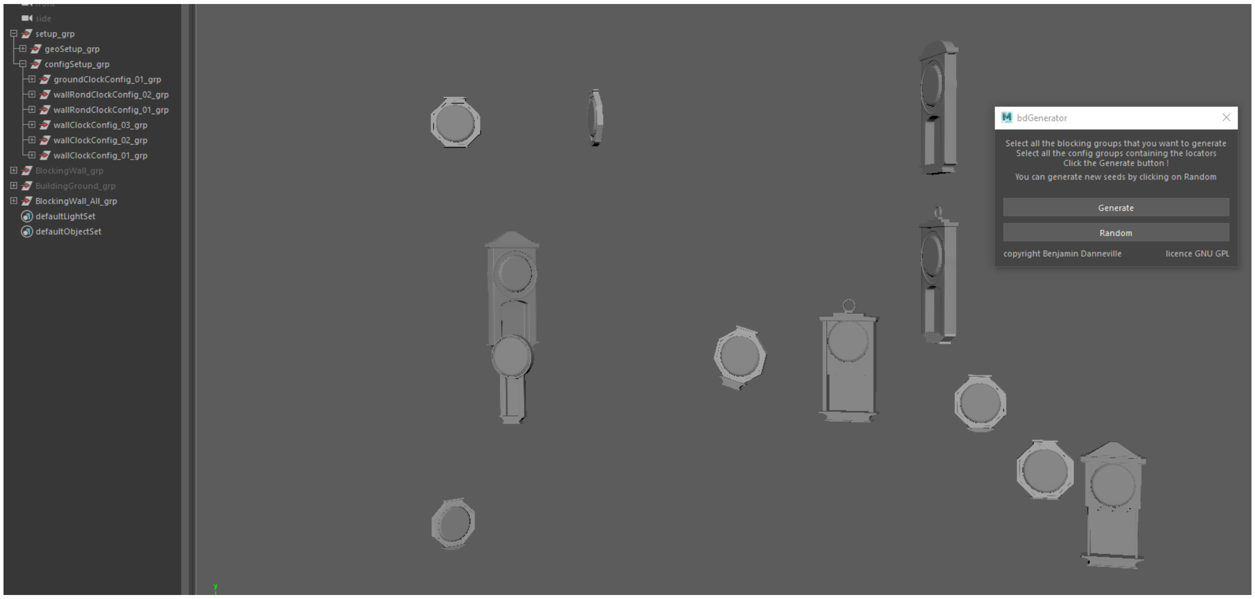
copyright Benjamin Danneville

licence GNU GPL

For more visibility when you generate your final models, hide your blocking groups

3.3 Generation

Now that you have selected your desired blocking and configuration groups,
Just click on `Generate` !

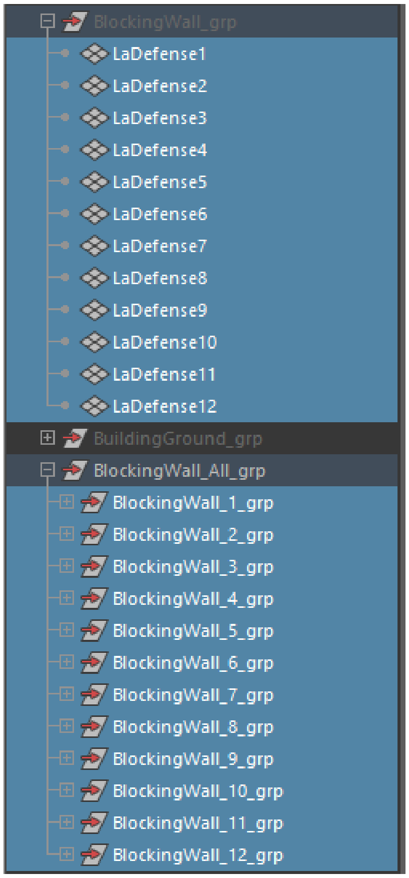


Select all the blocking groups that you want to generate
Select all the config groups containing the locators
Click the Generate button !

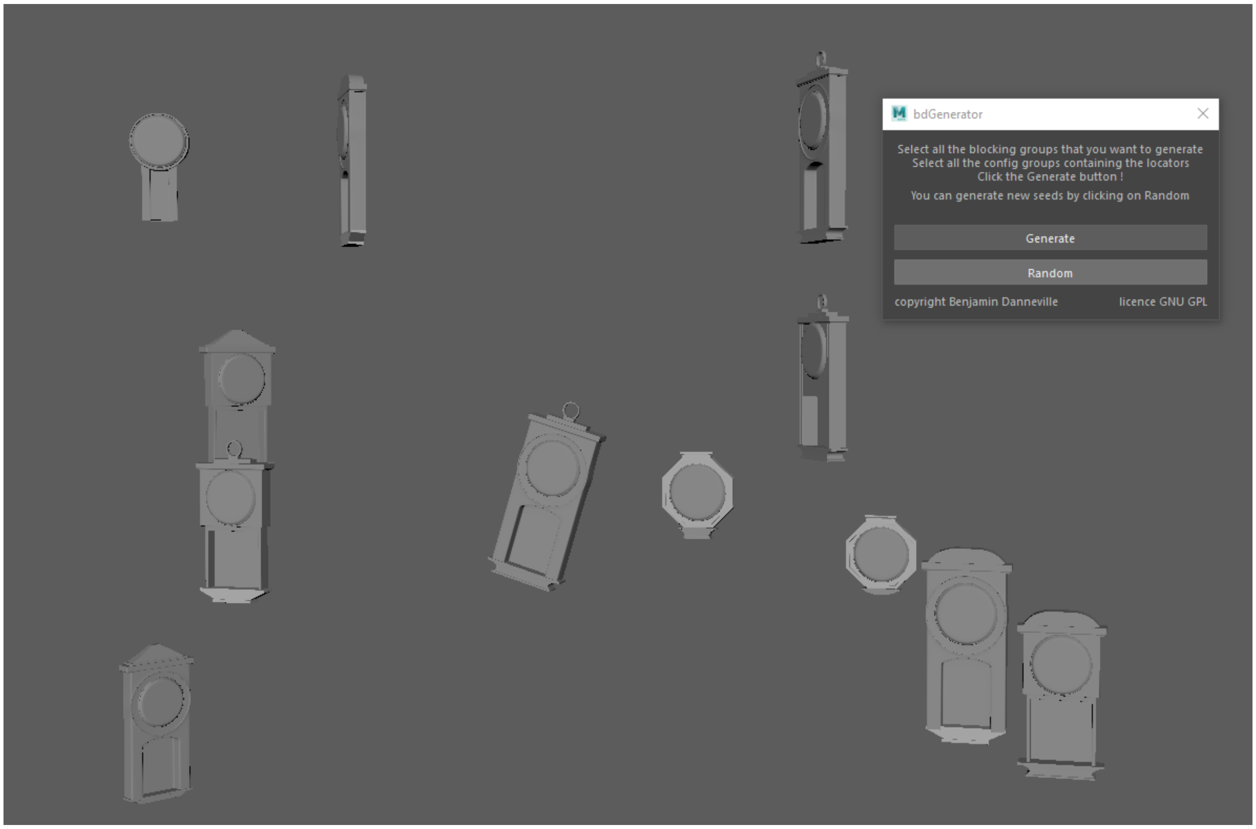
You can generate new seeds by clicking on Random

copyright Benjamin Danneville licence GNU GPL

The script create a group for each blocking groups, they are named using the blocking groups name. These group should contains a group for each meshes that were present in the blocking groups



If you're not happy with the random seed / generation that you got,
you can hit on Random so that it will generate again the last generation



bdGenerator

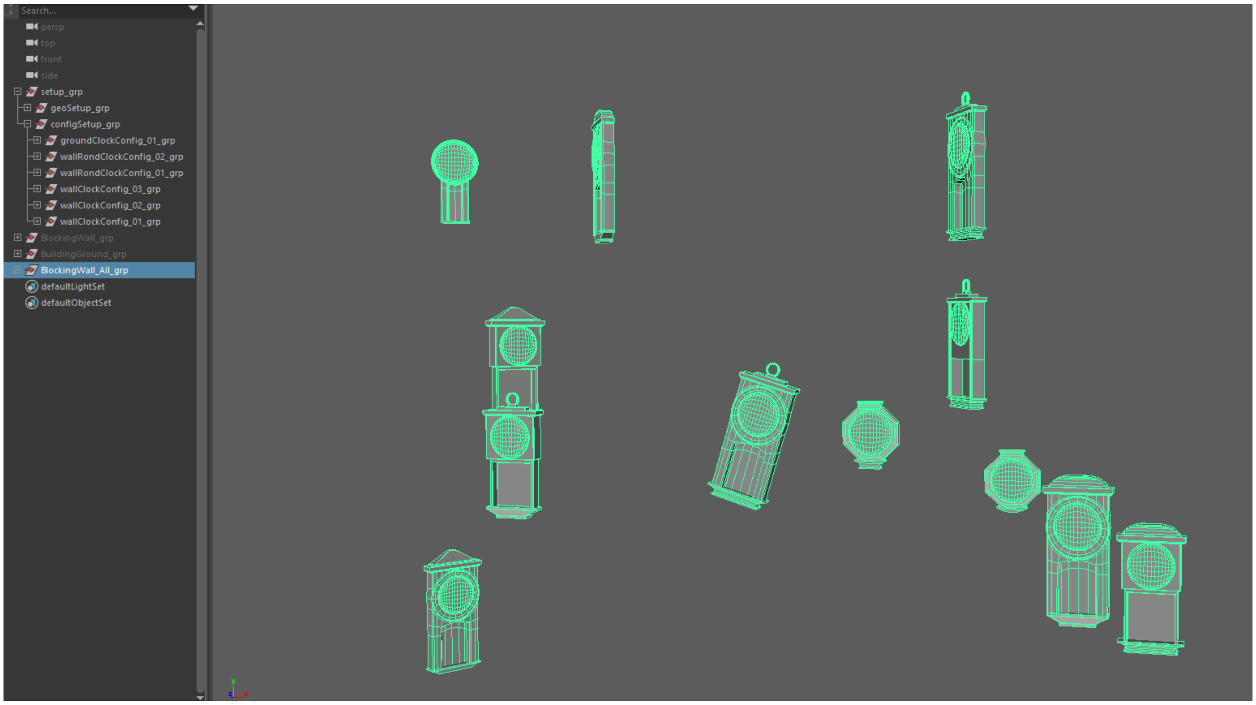
Select all the blocking groups that you want to generate
Select all the config groups containing the locators
Click the Generate button !

You can generate new seeds by clicking on Random

copyright Benjamin Danneville licence GNU GPL

3.4 Deleting a seed

If you want to delete the generation you did in a clean way,
you just have to delete the groups that bdGenerator created,
easely identified by the _All_grp name at the end


Search...
persp
top
front
side
setup_grp
geoSetup_grp
configSetup_grp
groundClockConfig_01_grp
wallRondClockConfig_02_grp
wallRondClockConfig_01_grp
wallClockConfig_03_grp
wallClockConfig_02_grp
wallClockConfig_01_grp
BlockingWall_grp
BuildingGround_grp
BlockingWall_All_grp
defaultLightSet
defaultObjectSet

CHAPTER FOUR

DOWNLOAD SCENE EXEMPLE

Here you can download the scene exemple to check if
you installed the script the right way.

Also, you can have fun with the configurations
by adding locators, removing or moving ones to understand how it works.

Download it here !

CHAPTER

FIVE

SOURCE

- Source code: https://github.com/benjamin-danneville/bdGenerator

CHAPTER

SIX

CONTACT

- My website: https://www.benjamindanneville.com/
- My LinkedIn: https://www.linkedin.com/in/benjamin-danneville/