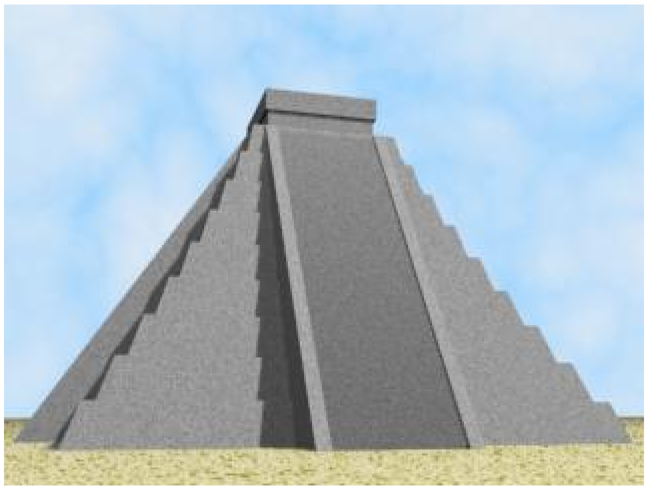
How to modeling Step Pyramid using Straight Stair



There is many ways to complete any task. I chose this method (using **Straight Stair** primitive) because they gives us the freedom to choose the number of steps in advance and this is an advantage. Also I will use **MacroRecorder** to capture script code for future use. Of course, this tutorial can be accomplished without *MacroRecorder* but the idea of this tutorial is not only to show you new "how to" modeling trick but also to familiarize yourself with the benefits of **MacroRecorder** in modeling process.

Here we will use *MacroRecorder* to optimize our work without entry into the study of *MAXScripts*(!) The point is to make script based template for future use. For this purpose you don't need advanced MAXScript knowledge, just starting but worth it.

Well, let`s start...

[1] First step is optional but creates good habits. In your file browser navigate to Max "*Scripts*" folder and create new subfolder, name it "*obj_templetes*" (for example) or something suitable for you. Here we will store and keep whole our working pieces of code for our convenience. Yep, because this snap-codes do not have UI (interface) and usually will run them from MAXScript Editor, it's a good idea not to mix them with other scripts.

[2] Start 3ds Max. Open menu **MAXScript > MAXScript Listener** (**F11**) and into Listener open menu **MacroRecorder** and select **Enable** to activate it.

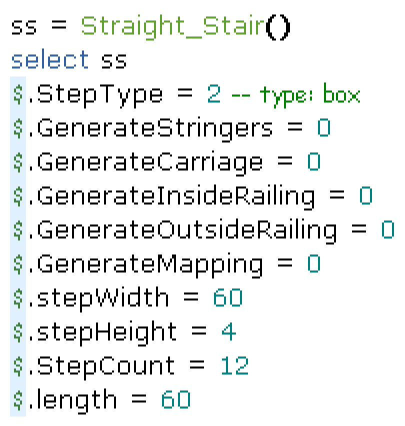
[3] In **Create panel > Geometry** select **Stairs from** dropdawn menu and press **Straight Stair** to create it. As you can see, recording each change and we must sort what of all params realy needed. Here we will work with Type:Box. Then select it, recorder return:
$.StepType = 2.

Also we don't need stringers, carriage, railing and mapping, so then disable them recorder will return:

```
$.GenerateStringers = 0
$.GenerateCarriage = 0
$.GenerateInsideRailing = 0
$.GenerateOutsideRailing = 0
$.GenerateMapping = 0
```

There what we will use: step width = 60, step height = 4, step counts = 12 and length = 60.
$.stepWidth = 60
$.stepHeight = 4
$.StepCount = 12
$.length = 60

[4] Now delete "*Straight Stair*".



```
ss = Straight_Stair()
select ss
$.StepType = 2 -- type: box
$.GenerateStringers = 0
$.GenerateCarriage = 0
$.GenerateInsideRailing = 0
$.GenerateOutsideRailing = 0
$.GenerateMapping = 0
$.stepWidth = 60
$.stepHeight = 4
$.StepCount = 12
$.length = 60
```

[5] Now start new script (**MAXSript > New Script**). Copy from Listener only what we need and paste code into this new script in *MAXScript Editor*.

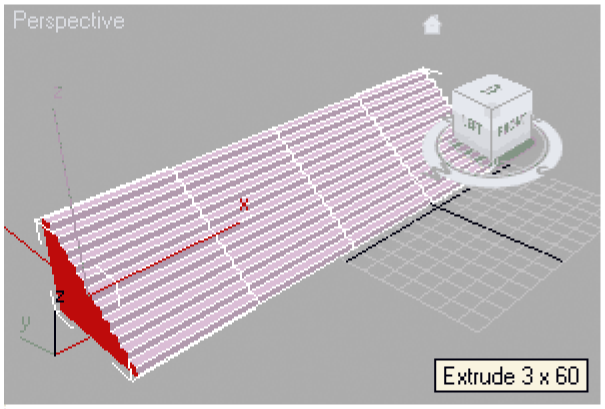
Following good script practice, assign variable name for the object and our script will start with:
ss = Straight_Stair()
This will create **Straight Stair** at [0,0,0] position. So, now we can select it calling variable name:
select ss

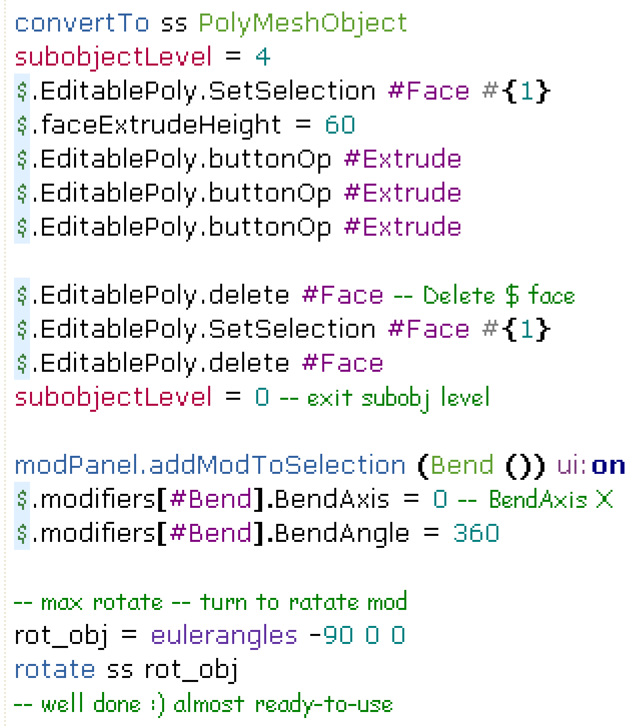
In left capture you can see what we need to create our *Straight Stair* object. You can play with this parameters, but for the moment *save* this script (in our "obj_templetes" folder) and don't close the file, we will work with it all the time in this tutorial.

I hope it's clear what we do. Here is our "Step 1 - creating Straight_Stair". That's right, in our case we will execute the operations by steps. So it's a good practice to add some comments to the scripts.

[6] Execute this code from MXS Editor. This step is needed to continue. At this stage you anointed to use "*Evaluate All*" (**Ctrl+E**) but for the next we will use only "*Evaluate Line / Selection*" (**Shift+Enter**) to work step by step.

[7] Next - "Step 2 - convertToPoly, Extrude, Bend"
Our stair is still selected. Convet to Editable Poly (convertTo $ PolyMeshObject). Go to polygon subobject level (subobjectLevel = 4). Select left side polygon ($.EditablePoly.SetSelection #Face #{1}). Press **Extrude Settings** button-box (right to the *Extrude* button) and type **60** for height (because we set stepWidth = 60), press twice "*Apply*" and once "*OK*" to extrude three times (see picture). Delete selected face and opposite face too. Exit subobject level (subobjectLevel = 0) by selecting root in the EPoly stack (do not deselect object in the scene).

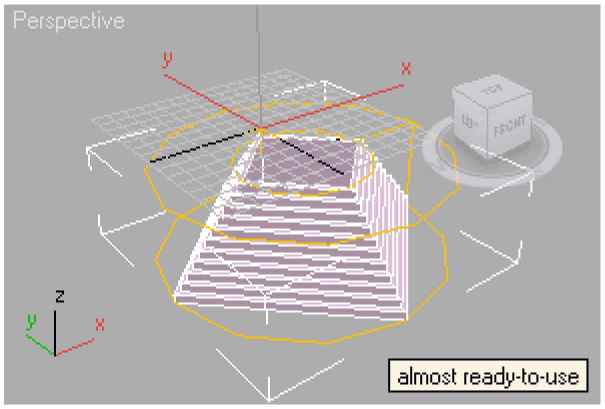




```
convertTo ss PolyMeshObject
subobjectLevel = 4
$.EditablePoly.SetSelection #Face #{1}
$.faceExtrudeHeight = 60
$.EditablePoly.buttonOp #Extrude
$.EditablePoly.buttonOp #Extrude
$.EditablePoly.buttonOp #Extrude

$.EditablePoly.delete #Face -- Delete $ face
$.EditablePoly.SetSelection #Face #{1}
$.EditablePoly.delete #Face
subobjectLevel = 0 -- exit subobj level

modPanel.addModToSelection (Bend ()) ui:on
$.modifiers[#Bend].BendAxis = 0 -- BendAxis X
$.modifiers[#Bend].BendAngle = 360

-- max rotate -- turn to ratate mod
rot_obj = eulerangles -90 0 0
rotate ss rot_obj
-- well done :) almost ready-to-use
```

[8] Add **Bend** modifier (modPanel.addModToSelection (Bend ()) ui:on). Change axis to X ($.modifiers[#Bend].BendAxis = 0) and angle to 360 ($.modifiers[#Bend].BendAngle = 360). Left only to rotate the object **-90** by X but **this operation will not recorded by MacroRecorder** !
We have two alternatives... First - to add some notes in our script to remember us to do this manualy, OR Second - to write a script command. So if look in MAXScript Help for "*rotate*" we will find how easy it is:
rot_obj = eulerangles -90 0 0
rotate ss rot_obj

End code for "Step 2" is on the right picture above.



[9] Next - "Step 3 - STL_Check, Weld vertices"
We have open edges so need to Weld vertices to fix this.
We need "STL Check" modifier to find where they are exactly.

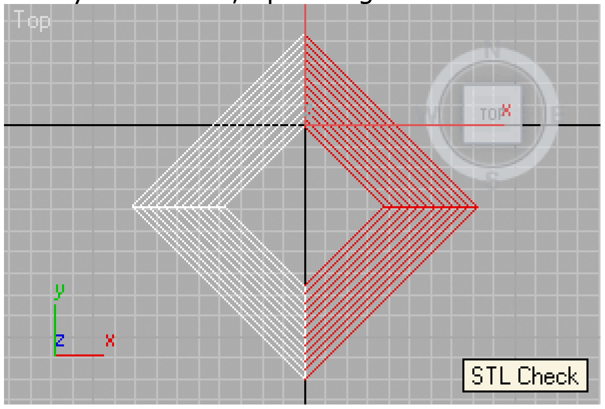
But first we need to collapse to poly our object and confirm with "Yes" in warning window. MacroRecorder will return 2 lines:
deleteModifier $ 1
maxOps.CollapseNodeTo $ 1 off

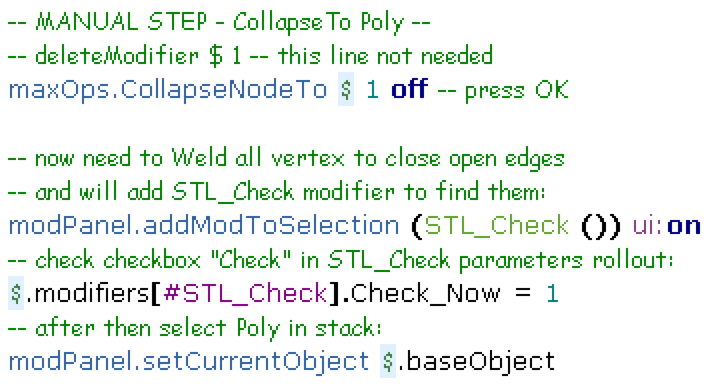
Little attention here!
We need only second line of code. Else as you see, Max will delete Bend modifier and then will collapse to poly, and we do not want this to happen. It is important to add a note here, something like "MANUAL STEP - CollapseTo Poly". That way we will know that all the code above (step 1 and 2) we can execute at once; also so this line of code (maxOps.CollapseNodeTo $ 1 off) must be executed as single. Hereinafter will be understood that the next lines of code will be executed in parts.

[10] Add "**STL Check**" modifier, check checkbox "*Check*" in him parameters rollout and after then select Poly in stack:

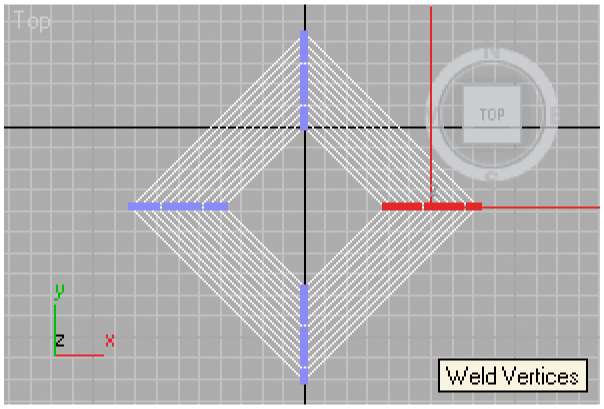
```
maxOps.CollapseNodeTo $ 1 off
modPanel.addModToSelection (STL_Check ()) ui:on
$.modifiers[#STL_Check].Check_Now = 1
modPanel.setCurrentObject $.baseObject
```

Now you can see, open edges are selected.





```
-- MANUAL STEP - CollapseTo Poly --
-- deleteModifier $ 1 -- this line not needed
maxOps.CollapseNodeTo $ 1 off -- press OK

-- now need to Weld all vertex to close open edges
-- and will add STL_Check modifier to find them:
modPanel.addModToSelection (STL_Check ()) ui:on
-- check checkbox "Check" in STL_Check parameters rollout:
$.modifiers[#STL_Check].Check_Now = 1
-- after then select Poly in stack:
modPanel.setCurrentObject $.baseObject
```

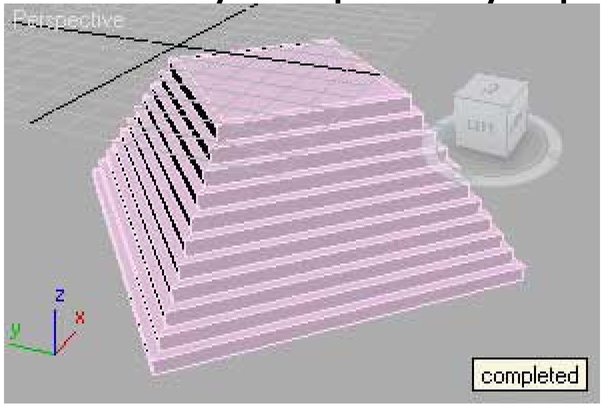


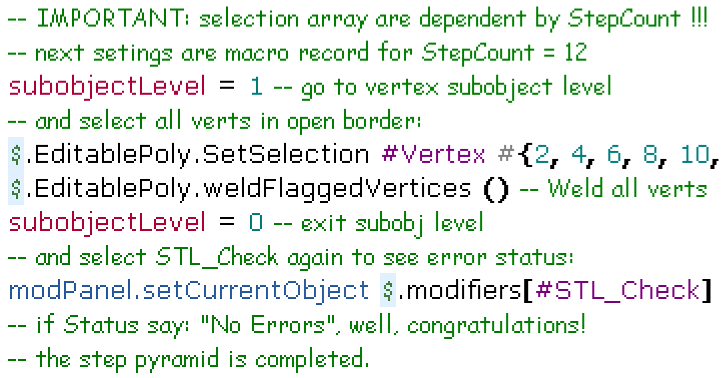
[11] Now in vertex subobject level we select all vertices in this corner (see picture). Next, press "**Weld**", exit subobject level and select "**STL Check**" modifier. If him status say: "*No Errors*", well, congratulations! The step pyramid is completed.

[12] What left to do? - Copy the code from *MacroRecorder*:

```
subobjectLevel = 1
$.EditablePoly.SetSelection #Vertex #{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 97..120}
$.EditablePoly.weldFlaggedVertices ()
subobjectLevel = 0
modPanel.setCurrentObject $.modifiers[#STL_Check]
```

Little attention here!
Selection array are dependent by StepCount. So here should add a note too.





```
-- IMPORTANT: selection array are dependent by StepCount !!!
-- next setings are macro record for StepCount = 12
subobjectLevel = 1 -- go to vertex subobject level
-- and select all verts in open border:
$.EditablePoly.SetSelection #Vertex #{2, 4, 6, 8, 10,
$.EditablePoly.weldFlaggedVertices () -- Weld all verts
subobjectLevel = 0 -- exit subobj level
-- and select STL_Check again to see error status:
modPanel.setCurrentObject $.modifiers[#STL_Check]
-- if Status say: "No Errors", well, congratulations!
-- the step pyramid is completed.
```

Also can add note so this *SetSelection* is a record for *StepCount=12* preset. In the future we can record and save more presets and execute that needed or continue with the rest operations without script assistant.

If we are satisfied with this result, we may save the script end break up here. Our object templete is done. But if we want pyramid object like this one from picture above, ie without the square tunnel in the middle, then go to next step.

[13] Next - "Step 4 - make solid pyramid primitive"
Here we will select and delete unnecessary faces, and finaly cap open holes. As you guess and face selection array dependent by *StepCount*. So what we do here:

```
deleteModifier $ 1 -- delete STL_Check
subobjectLevel = 4 -- go to polyg. subobj level
$.EditablePoly.SetSelection #Face #{23..24, 27..28, 51
$.EditablePoly.delete #Face -- delete selected faces
subobjectLevel = 3 -- go to border subobj level
$.EditablePoly.SetSelection #Edge #{56, 82, 127, 161}
$.EditablePoly.capHoles #Edge -- Cap top of pyramid
$.EditablePoly.SetSelection #Edge #{35, 81, 126, 160}
$.EditablePoly.capHoles #Edge -- Cap bottom of pyramid
subobjectLevel = 0 -- exit subobj level
```

(*) Delete "STL Check" modifier
(*) Go to polygon subobject level
(*) Select 4 inside faces and 4 bottom faces
(*) Delete them all
(*) Go to border subobject level
(*) Select top open border
(*) Press "Cap" button
(*) Select bottom open border
(*) Press "Cap" button
(*) Exit subobject level

```
-- optionaly we can check for errors again:
modPanel.addModToSelection (STL_Check ()) ui:on
$.modifiers[#STL_Check].Check_Now = 1
-- but hardly necessary :)
```

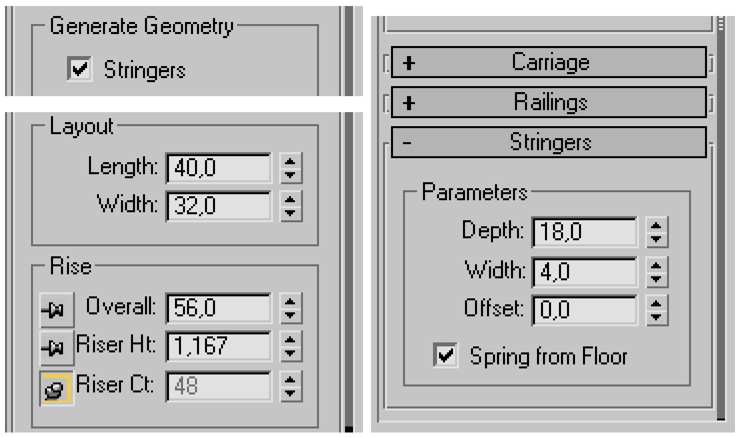
Well done :)
Optionaly we can check for errors again
(but hardly necessary)

Now you can save and close this script.
Also in **Listener** menu *MacroRecorder* select **Enable** to uncheck/trurn off recorder.

Well, now what? If we want to make step pyramid like mayan in Chichen Itza (Mexico), we must create and human stairs.

Well, let's do it.

[14] First in *Hierarhy* panel press "**Affect Pivot Only**" button and in Alignment group press "**Align to Object**". Press "**Affect Pivot Only**" again to exit this mod. Now press **W** key (max move mode) and *right click* over the coordinate sprinners to *reset* it to zero [0,0,0]. But our bottom face is under **-27.5** by Z axis, so in Z type **27.5** and hit Enter. Now the pyramid aligned to the world.



Generate Geometry
- [✓] Stringers

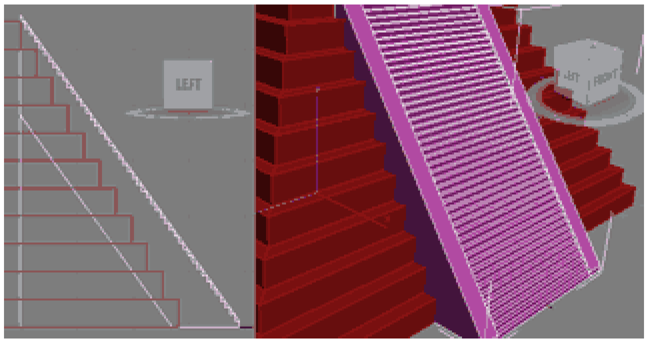
Layout
- Length: 40,0
- Width: 32,0

Rise
- Overall: 56,0
- Riser Ht: 1,167
- Riser Ct: 48

Parameters
- Depth: 18,0
- Width: 4,0
- Offset: 0,0
- [✓] Spring from Floor

[15] *Execute* (**Shift+Enter**) script code from "Step 1" (only), ie create new **Straight Stair** primitive.

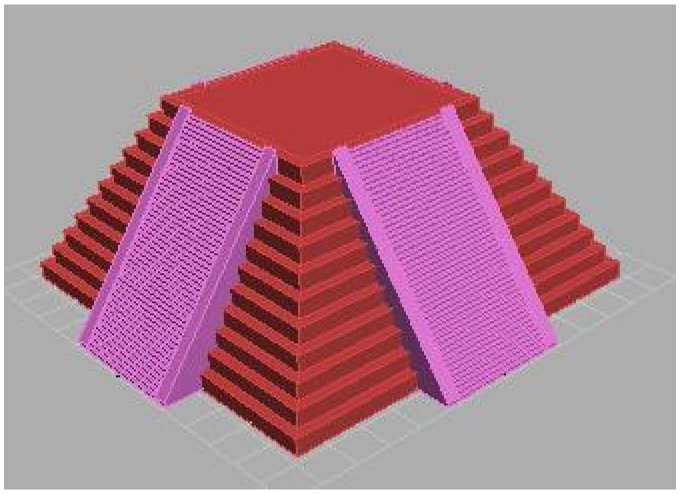
[16] Insert settings like in the left picture.



[17] Align this new stair to the pyramid and *then* change it *pivot only* by pressing "**Affect Pivot Only**" button in *Hierarhy* panel and press **"Align to World"** or *right click* over the coordinate spinners to *reset* all to zero [0,0,0]. Press "**Affect Pivot Only**" again to exit this mod.

[18] Use "**Mirror**" by **Y**, Offset: **0**, Clone: *Instance* or *Reference* .

[19] Select this 2 stairs objects and use "**Tools > Array**" with *Count* = **2** and *Rotate* by **Z = 180**, Clone: *Instance* or *Reference*.



Picture on the left shows what you need to obtain.

Remains only to build the temple on the top of the pyramid, but this is an easy box-modeling task, so this tutorial finished their work and I conclude here.

I hope this tutorial has been useful to you.

written by **Anubis**

http://3dmyths.blogspot.com/