Bake Camera from Alembic

Shelf script to bake camera form Alembic camera in scene. Must select the child camera in the alembic sop.

```
import hou
obj = hou.node("/obj")



ocam = hou.node(hou.ui.selectNode(node_type_filter=hou.nodeTypeFilter.ObjCamera))
ver = ocam.parent().parent().parm("vpp_version_abc").eval().split("_")[-1]



reslist = ["1920x1080", "3840x2160", "3072x2109", "3024x2016", "2224x1548", "Keep Original",
"Custom"]
sl = hou.ui.selectFromList(reslist, message="choose resolution", sort=True, exclusive=True)
stopval = sl[0]
resolution = reslist[stopval]
if resolution is "Keep Original":
    pass
elif resolution is "Custom":
    custom = hou.ui.readInput("Enter resolution as #x# format, ie: 1920x1080. The x is
needed.", title="Enter Resolution")
    resolution = custom[1]
    ocam.parm("resx").set(resolution.split("x")[0])
    ocam.parm("resy").set(resolution.split("x")[1])

else:
    ocam.parm("resx").set(resolution.split("x")[0])
    ocam.parm("resy").set(resolution.split("x")[1])

try:
    shotinfo = hou.node("/obj/SHOTINFO")
    backplate = shotinfo.parm("backplate")
except:

    backplate = ocam.parm("vm_background")
    pass
```

```
camName = ocam.name() + "_baked"
#setback = ocam.parm("vm_background").set(backplate)
tcam = obj.createNode("cam", camName)
tcam.moveToGoodPosition()

scalenull = obj.createNode("null", "shotscale")
scalenull.parm("scale").set(.1)
scalenull.moveToGoodPosition()

tcam.setInput(0, scalenull)



#add point with normal for facing cam normals
atwr = tcam.createNode("attribwrangle", "normal")
atwr.parm("snippet").set("v@N=set(0,0,1);")
atwr.setInput(0, tcam.node("camOrigin"))
normnull = tcam.createNode("null", "OUT_POINT")
normnull.setInput(0, atwr)

#copy keyframes
cam_xform=["tx", "ty", "tz", "rx", "ry", "rz"]
cam_parms=["resx", "resy", "aspect", "focal", "aperture", "orthowidth", "shutter", "focus",
"fstop"]
tcam.parm("vm_background").set(backplate)

parms_bake = list()
parms_bake.extend(cam_xform)
parms_bake.extend(cam_parms)
start = hou.playbar.playbackRange()[0]
end = hou.playbar.playbackRange()[1]

p = hou.StringParmTemplate("version", "Version", 1)
g = tcam.parmTemplateGroup()
g.append(p)
tcam.setParmTemplateGroup(g)
tcam.parm("version").set(ver)

with hou.undos.group("bake cam"):
    for x in range(int(start), int(end +1)):
        hou.setFrame(x)
```

```
        tcam.setWorldTransform(ocam.worldTransform())
        for p in parms_bake:
            parm = tcam.parm(p)
            if parm.name() in cam_xform:
                parm.setKeyframe(hou.Keyframe(parm.eval()))
            else:
                parm.setKeyframe(hou.Keyframe(ocam.parm(p).eval()))
```

Revision #6
Created 2023-04-25 12:13:18 UTC by Anthony
Updated 2025-02-05 13:03:27 UTC by Anthony