Här är ett mindre script för att skriva ut alla objekt som finns i en blend fil. Tanken är att detta ska byggas på och användas som filformat för byggnad av spelvärlden.

För att importera objekt till blender ska File > Link användas.

Koden kräver som ni ser att det finns en “Blender_Export” mapp i C disken. Går dock att ändra.

TODO: Lägg till rotations värden. Kanske taggar så att man kan även stödja mer än modeller.

```
import bpy
import os

# Ensure all folders of the path exist
#path = "C:/Blender_Export/"
path = "C:/Users/Hampus/Desktop/1337xXSlayer/Project/core/assets/map/"
fileName = "BasicMap.txt"
os.makedirs(path, exist_ok=True)

#Store lux elements in these dicts. Luxelements contains list of objects
luxElements = {}
luxContainers = {};


#Get tag string from element
def getTag(obj):
    if obj.type in 'CAMERA':
        return "Camera"
    if obj.type in 'MESH':
        #Ob.library returns null if object is not linked
        if ob.library:
            return "Model"
        else:
            return "Mesh"
    if obj.type in 'LAMP':
        return "Light"
    #If not found return nothing
    return None

#Returns True if _proxy element
def handleLuxObject(obj):
    if "_proxy" in obj.name:
        print("Test   " + obj.name.split("_")[0] + "\n")
        if obj.name.split("_")[0] in luxElements:
            luxElements[obj.name.split("_")[0]].append(obj)
        else:
            luxElements[obj.name.split("_")[0]] = [obj]
        return True
    else:
        luxContainers[obj.name] = obj
        return False


def generateLuxObjects():
    for luxName in luxContainers.keys():
        ob = luxContainers[luxName]

        values = "\t"
        tag = getTag(ob)
        values += "<" + tag + " "

        position = "x=\"%f\" y=\"%f\" z=\"%f\" rotationX=\"%f\" rotationY=\"%f\"
rotationZ=\"%f\"" % (ob.location.x, ob.location.y, ob.location.z, ob.rotation_euler.x,
ob.rotation_euler.y, ob.rotation_euler.z)
        values += position

        #Iterate thorugh all custom properties of object
        for key in ob.keys():
            #if key not in ['_RNA_UI']:
            if key not in ['_RNA_UI' ,'luxrender_object']:
                values += (" " + key + "=\"" + str(ob[key]) + "\"")


        values += ">\n"
        #Iterate through childern
        for obChild in luxElements[luxName]:
            valuesChild = "\t\t"
            tagChild = getTag(obChild)
            valuesChild += "<" + tagChild + " "

            positionChild = "x=\"%f\" y=\"%f\" z=\"%f\" rotationX=\"%f\" rotationY=\"%f\"
rotationZ=\"%f\"" % (obChild.location.x, obChild.location.y, obChild.location.z,
obChild.rotation_euler.x, obChild.rotation_euler.y, obChild.rotation_euler.z)
            valuesChild += positionChild

            for key in obChild.keys():
                if key not in ['_RNA_UI' ,'luxrender_object']:
                    valuesChild += (" " + key + "=\"" + str(obChild[key]) + "\"")

            values += valuesChild + "></" + tagChild + ">\n"



        file.write(values + "\t</" + tag + ">\n")


# Write data out to file
with open(path + fileName, "w") as file:
    file.write("<?xml version=\"1.0\"?>\n<Map>\n")
    #Iterate through all objects in blend file
    for ob in bpy.data.objects:
        luxObject = False
        values = "\t"
        tag = getTag(ob)

        values += "<" + tag + " "
        #print(ob.type)

        position = "x=\"%f\" y=\"%f\" z=\"%f\" rotationX=\"%f\" rotationY=\"%f\"
rotationZ=\"%f\"" % (ob.location.x, ob.location.y, ob.location.z, ob.rotation_euler.x,
ob.rotation_euler.y, ob.rotation_euler.z)
        values += position



        #Iterate thorugh all custom properties of object
        for key in ob.keys():

            #if key not in ['_RNA_UI']:
            if key not in ['_RNA_UI' ,'luxrender_object']:
                values += (" " + key + "=\"" + str(ob[key]) + "\"")
            if key in 'luxrender_object':
                luxObject = handleLuxObject(ob)
                if luxObject is True:
                    break

        if luxObject is False:
            file.write(values + "></" + tag + ">\n")


        #If object is linked
        if ob.library:
            print("Found a linked object")

    generateLuxObjects();

    file.write("</Map>")
    print("Done -- -- -- - -")
```