

X2 Blender Importer

Import aller 3d-Objekte aus dem Spiel X (Beyond the Frontier 2) nach Blender in Python

```
import Blender
```

```
strTexDir=Blender.Draw.Create("")
strBODDir=Blender.Draw.Create("")
strSzeDir=Blender.Draw.Create("")
strBODFile=Blender.Draw.Create("")
blnLOD=0
```

```
def configer(zustand):
    #Hole/Schreibe Konfiguration
    # global strTexDir, strBODDir, strSzeDir
    if zustand==1:
        dictKonf=Blender.Registry.GetKey("X2Blender", True)
        if dictKonf:#wenn es die gibt
            try:
                strBODFile.val=dictKonf["bodfile"]
                strTexDir.val=dictKonf["texturdir"]
                strBODDir.val=dictKonf["boddir"]
                strSzeDir.val=dictKonf["szenedir"]
            except:
                pass
        else:
            dictKonf={}
            print "hier", strBODFile.val
            dictKonf["bodfile"]=strBODFile.val
            dictKonf["texturdir"]=strTexDir.val
            dictKonf["boddir"]=strBODDir.val
            dictKonf["szenedir"]=strSzeDir.val
            Blender.Registry.SetKey("X2Blender", dictKonf, True)
```

```
def dirseparator1(filename):
    #Funktion trennt Pfad von der Datei und dient als callback
    global strBODDir
    strBODDir.val=Blender.sys.dirname(filename)
    configer(0)
    return
```

```
def dirseparator2(filename):
    #Funktion trennt Pfad von der Datei und dient als callback
    global strTexDir
    strTexDir.val=Blender.sys.dirname(filename)
    configer(0)
    return
```

```
def dirseparator3(filename):
    #Funktion trennt Pfad von der Datei und dient als callback
    global strSzeDir
    strSzeDir.val=Blender.sys.dirname(filename)
    configer(0)
```

```

    return

def fileselect(filename):
    #Callback fuer BOD Datei
    # global strBODFile
    strBODFile.val=filename
    configer(0)
    return

def split(text,trenner):
    lstErg=[]
    d=""
    for i in text:
        if i!=trenner:
            d=d+i
        else:
            lstErg.append(d)
            d=""
    if text[-1]!=trenner:
        lstErg.append(d)
    return lstErg

def imgexistenz(dateiname):
    #prueft die Existenz der Bilddatei
    if Blender.sys.exists(dateiname+".jpg"):
        return dateiname+".jpg"
    elif Blender.sys.exists(dateiname+".tga"):
        return dateiname+".tga"
    else:#fehler
        return Blender.Get("scriptsdir")+Blender.sys.sep+"ersatz.jpg"

def texturholer(nummer):
    #Erstellt oder ruft Textur
    blnTexExist=0
    for j in Blender.Texture.Get():
        if j.getName()=="x_"+nummer:
            blnTexExist=1
            break
    if blnTexExist==0:
        textur=Blender.Texture.New("x_"+nummer)
        textur.setType("Image")
        bilddatei=imgexistenz(strTexDir.val+Blender.sys.sep+nummer)
        img=Blender.Image.Load(bilddatei)
        textur.image=img
    else:
        textur=Blender.Texture.Get("x_"+nummer)
        bilddatei=textur.getImage().getFilename()
    return (textur,bilddatei)

def find(text,suche):
    intErg=-1
    if suche in text:
        for i in range(len(text)):
            if text[i:i+len(suche)]==suche:

```

```

        intErg=i
        break
    return intErg

def newObject(name):
    scene = Blender.Scene.getCurrent()
    object= Blender.Object.New("Mesh",name)
    scene.link(object)
    return object

def findscale(strObjName,bodysize):
    #liefert den skalierungswert zurueck
    absm=0
    for k in range(1000):
        try:obj=Blender.Object.Get(strObjName+"_"+str(k))
        except:break
        m=[]
        for j in range(3):
            d=[]
            for i in obj.getBoundingBox():d.append(i[j])
            d.sort()
            m.append(d[7]-d[0])
        m.sort()
        if absm<m[2]:absm=m[2]
    fltProp=(bodysize/500.0)/absm
    # for k in range(1000):
    # try:
    #     obj=Blender.Object.Get(strObjName+"_"+str(k))
    #     obj.setSize(fltProp,fltProp,fltProp)
    # except:
    #     break
    return fltProp

def addEmpty(strObjname,level,location,rotation,sizer):
    #verlinkt alle BodyObjekte mit einem Empty
    emp=Blender.Object.New("Empty")
    emp.setName(strObjname)
    scene = Blender.Scene.getCurrent()
    scene.link(emp)
    lstObjs=[]
    for k in range(1000):
        try:
            obj=Blender.Object.Get(strObjname+"_"+level+"_"+str(k))
            lstObjs.append(obj)
        except:
            break
    emp.makeParent(lstObjs)
    emp.setLocation(location)
    emp.setEuler(3.1416*rotation[0],3.1416*rotation[1],3.1416*rotation[2])
    emp.setSize(sizer,sizer,sizer)

def vertfaces(coord,obj,punktliste,boddatei,txlist):
    hlp=split(coord, ";")
    fcoord=[punktliste[int(hlp[1])],punktliste[int(hlp[2])],punktliste[int(hlp[3])]]

```

```

#Punkt hinzufuegen
neulist=[]
for i in fcoord:
    punktda=0
    for j in obj.verts:
        if i==j:
            neulist.append(j)
            punktda=1
            break
    if punktda==0:
        obj.verts.append(i)
        neulist.append(obj.verts[-1])
#Flaechen hinzufuegen
f=Blender.NMesh.Face()
for n in neulist:f.v.append(n)
if texlist[str(int(hlp[0]))]!="0":
    #UV Koordinaten hinzufuegen
    uv=[]
    intUvKorr=0
    if len(hlp)==12:intUvKorr=1
    uv.append((float(hlp[6-intUvKorr]),float(hlp[7-intUvKorr])*-1.0))
    uv.append((float(hlp[8-intUvKorr]),float(hlp[9-intUvKorr])*-1.0))
    uv.append((float(hlp[10-intUvKorr]),float(hlp[11-intUvKorr])*-1.0))
    f.uv=uv
    img=Blender.Image.Load(texlist[str(int(hlp[0]))])
    f.image = img
obj.faces.append(f)
return

def bodimporter(bodfile):
    #Funktion zum Importieren der BODs
    global blnLOD,strTexDir,strBODDir,strSzeDir
    #den reinen BOD-Namen extrahieren
    bodatei=Blender.sys.basename(bodfile)[-4]
    #BOD-File in Liste einlesen
    ff=file(bodfile,"r")
    lstBOD=ff.readlines()
    ff.close()
    intMatAnz=0
    TexList={}
    #Materialblock auslesen (x_bodNummer_intID)
    for i in lstBOD:
        if i[:8]=="MATERIAL":
            intMatAnz+=1
            lstMatDef=split(split(i,":")[1],";")
            strMatID=str(int(lstMatDef[0]))
            strMatName="x_"+bodatei+"_"+strMatID
            #Texturliste UV muss immer erstellt werden
            strTexNr=str(int(lstMatDef[1]))
            TexList[strMatID]=strTexNr
            texhelper=texturholer(strTexNr)
            if strTexNr!="0":TexList[strMatID]=texhelper[1]
            #ueberpruefen ob es dieses Material schon gibt
            blnMatExist=0

```

```

for j in Blender.Material.Get():
    if j.getName()==strMatName:
        blnMatExist=1
        break
#Material anlegen
if blnMatExist==0:
    material = Blender.Material.New(strMatName)
    #Materialreihenfolge
    #0 Textur
    #1 BumpMap
    #2 EnvironmentMap
    #3 Lightmap
    #Textur
    if strTexNr!="0":
        textur=texhelper[0]
        material.setTexture(0,textur,Blender.Texture.TexCo.UV,Blender.Texture.MapTo.COL)
    #Bumpmap
    if int(lstMatDef[21])!=0:
        strBumpTex=str(int(lstMatDef[21]))
        textur=texturholer(strBumpTex)[0]
        material.setTexture(1,textur,Blender.Texture.TexCo.UV,Blender.Texture.MapTo.NOR)
    #Lightmap
    # if len(lstMatDef)>23:
    #     if int(lstMatDef[23])!=0:
    #         strLightNr=str(int(lstMatDef[23]))
    #         textur=texturholer(strLightNr)[0]
    #         material.setTexture(3,textur,Blender.Texture.TexCo.UV,Blender.Texture.MapTo.COL)
    #Environmap
    if int(lstMatDef[19])!=0:
        strEnvTex=str(int(lstMatDef[19]))
        textur=texturholer(strEnvTex)[0]
        material.setTexture(2, textur,Blender.Texture.TexCo.ORCO,Blender.Texture.MapTo.COL)
    #AmbientColor??
    material.setRGBCol([float(lstMatDef[5]),float(lstMatDef[6]),float(lstMatDef[7])])
    material.setSpecCol([float(lstMatDef[8]),float(lstMatDef[9]),float(lstMatDef[10])])
    material.setAlpha((100-int(lstMatDef[11]))/100.0)
    material.setEmit(int(lstMatDef[12])*0.01)
    material.setSpec(int(lstMatDef[13])*0.02)
    material.setHardness(int(int(lstMatDef[14])*5.11))
    material.setMode("TexFace")
#Automatic Body Size Block
for i in lstBOD:
    if find(i,"/ Automatic Object Size")!=-1 or find(i,"/ Automatic body size")!=-1:
        lngBodySize=long(split(i,";")[0])
        break

#LOD Block
intLOD=0
blnFace=0

#erstelle fuer jedes Material ein Objekt
lstObj=[]
for i in range(intMatAnz):
    objekt=newObject(boddatei+"_"+str(intLOD+1)+"_"+str(i))

```

```

objekt.layers=[intLOD+1]
Mesh = objekt.getData()
#fuege alle Materialien hinzu
for j in Blender.Material.Get():
    if j.getName()=="x_"+bodatei+"_"+str(i):
        Mesh.materials.append(j)
lstObj.append(Mesh)

lstPunkte=[]

for i in lstBOD:
    helfer=split(i,",")
    #Punkte auslesen
    if len(helfer)==4:
        xco=float(long(helfer[0])/1000.0)
        yco=float(long(helfer[1])/1000.0)
        zco=float(long(helfer[2])/1000.0)
        if xco!=-0.001 or yco!=-0.001 or zco!=-0.001:
            #Punktobjekt erstellen
            v=Blender.NMesh.Vert(xco,yco,zco)
            lstPunkte.append(v)

#Flaechen auslesen
elif len(helfer)==13 or len(helfer)==12:#Faces mit Kommentaren
    vertfaces(i,lstObj[int(helfer[0])],lstPunkte,bodatei,TeXList)
elif len(helfer)==3:#Faces zu Ende
    if int(helfer[0])==-99 and int(helfer[1])==0:
        intLOD+=1
        for j in lstObj:
            j.update()

    if blnLOD==1:
        #erstelle fuer jedes Material ein Objekt
        lstPunkte=[]
        lstObj=[]
        for h in range(intMatAnz):
            objekt=newObject(bodatei+"_"+str(intLOD+1)+"_"+str(h))
            objekt.layers=[intLOD+1]
            Mesh = objekt.getData()
            #fuege alle Materialien hinzu
            for j in Blender.Material.Get():
                if j.getName()=="x_"+bodatei+"_"+str(h):
                    Mesh.materials.append(j)
            lstObj.append(Mesh)
        else:
            break

Blender.Redraw()
return lngBodySize

def sceneimporter(scefile):
    # Importiert ganze Szene
    ff=file(scefile)
    lstSCE=ff.readlines()

```

```

ff.close()
lstAllSce=[]
blnObj=0
scn = Blender.Scene.getCurrent()
for i in lstSCE:
    if i[0]=="P":
        helfer=str(int(split(split(i, ";")[1], "B")[1]))
        bodname="00000"[:-len(helfer)]+helfer
        boddatei="C:\\Spiele\\X2 - Die Bedrohung\\X2_ModderKIT14\\v"+Blender.sys.sep+bodname+".bod"
        #Existenzpruefung
        if Blender.sys.exists(boddatei):
            blnObj=1
            #Kopie oder Original des Empties
            intCopyAnzahl=lstAllSce.count(bodname)
            if intCopyAnzahl==0:
                #original erstellen
                sizer=bodimporter(boddatei)
                emptyname=bodname
            else:
                #Kopie aller Objekte erstellen
                for j in Blender.Object.Get():
                    if j.getName()[:len(bodname+"_1_")]==bodname+"_1_":
                        type = j.getType()
                        objneu = Blender.Object.New(type)
                        objneu.shareFrom(j)
                        objneu.setName("c_"+str(intCopyAnzahl)+"_"+j.getName())
                        scn.link(objneu)
                        #objneu.setMatrix(j.getMatrix())
                        #objneu.drawMode = j.drawMode
                        objneu.Layer = j.Layer
                        scn.update(1)
                        emptyname="c_"+str(intCopyAnzahl)+"_"+bodname
                    #in Liste
                    lstAllSce.append(bodname)
        if i[0]!="P" and blnObj==1:
            #Empty zum erstellten Objekt hinzufuegen und Positionieren
            helfer=split(i, ";")
            XPos=long(helfer[1])/1000.0
            YPos=long(helfer[2])/1000.0
            ZPos=long(helfer[3])/1000.0
            factor=float(helfer[4])
            XRot=float(helfer[5])
            YRot=float(helfer[6])
            ZRot=float(helfer[7])
            sizeempty=findscale(bodname+"_1_",sizer)
            addEmpty(emptyname, "1", [XPos, YPos, ZPos], [XRot, YRot, ZRot], sizeempty)
            blnObj=0
    return

##
# grafische Oberflaeche
##
def fenster():

```

```

global strBODFile, strTexDir, strBODDir, strSzeDir

#Zeichne Rahmen
Blender.BGL.glColor3f(0,0,0)
Blender.BGL.glBegin(Blender.BGL.GL_LINE_STRIP)
Blender.BGL.glVertex2i(5,190)
Blender.BGL.glVertex2i(5,5)
Blender.BGL.glVertex2i(470,5)
Blender.BGL.glVertex2i(470,190)
Blender.BGL.glVertex2i(150,190)
Blender.BGL.glEnd()

#Zeichne Texte
Blender.BGL.glRasterPos2i(8,187)
Blender.Draw.Text("XBlender Im-/Exporter")
Blender.BGL.glRasterPos2i(10,170)
Blender.Draw.Text("BOD-/Szene-Datei")
Blender.BGL.glRasterPos2i(250,10)
Blender.Draw.Text("LOD rendern")
Blender.BGL.glRasterPos2i(10,135)
Blender.Draw.Text("Body-Verzeichnis")
Blender.BGL.glRasterPos2i(10,100)
Blender.Draw.Text("Textur-Verzeichnis")
Blender.BGL.glRasterPos2i(10,65)
Blender.Draw.Text("Szenen-Verzeichnis")

Blender.BGL.glColor3f(1,1,1)
Blender.BGL.glRasterPos2i(10,10)
Blender.Draw.Text("X2Blender v 0.5 by XIshhtar & Earthpilot")

#Zeichne Schalter
Blender.Draw.PushButton("Open",103,430,150,35,15)
Blender.Draw.PushButton("Open",600,430,115,35,15)
Blender.Draw.PushButton("Open",601,430,80,35,15)
Blender.Draw.PushButton("Open",602,430,45,35,15)
colSlider=Blender.Draw.Slider("",28,330,7,40,15,blnLOD,0,1)
Blender.Draw.PushButton("Import",104,390,7,75,30)

#Zeichne Eingabefelder
strBODFile=Blender.Draw.String("",0,10,150,415,15,strBODFile.val,100)
strBODDir=Blender.Draw.String("",0,10,115,415,15,strBODDir.val,100)
strTexDir=Blender.Draw.String("",0,10,80,415,15,strTexDir.val,100)
strSzeDir=Blender.Draw.String("",0,10,45,415,15,strSzeDir.val,100)

Blender.Draw.Redraw(1)

##
# Auswerten von allgemeinen Ereignissen
##
def allg_ereignis(erg,wert):
    if erg==Blender.Draw.ESCKEY or erg==Blender.Draw.QKEY:Blender.Draw.Exit()

##
# Auswerten von Schalter-Ereignissen

```



```

##
def schalter_ereignis(erg):
    global blnLOD
    if erg==103:# v Datei
        Blender.Window.FileSelector(fileselect,"open file",strBODFile.val)
    elif erg==104:#import start
        #check ob alle Verzeichnisse gesetzt wurden
        if strBODDir.val==" or strTexDir.val==" or strSzeDir.val=="":
            Blender.Draw.PupMenu("Bitte alle Verzeichnisse angeben")
        else:
            if strBODFile.val!="":
                if find(strBODFile.val,strBODDir.val)!=-1:
                    bodimporter(strBODFile.val)
                    Blender.Draw.PupMenu("fertig")
                elif find(strBODFile.val,strSzeDir.val)!=-1:
                    sceneimporter(strBODFile.val)
                    Blender.Draw.PupMenu("fertig")
            else:
                Blender.Draw.PupMenu("Ermittlung des Types BOD/Scene nicht moeglich")
    elif erg==28:#LOD
        blnLOD=not(blnLOD)
    elif erg==600:# BOD Dir
        Blender.Window.FileSelector(dirseparator1,"open file")
    elif erg==601:# Tex Dir
        Blender.Window.FileSelector(dirseparator2,"open file")
    elif erg==602:# Szene Dir
        Blender.Window.FileSelector(dirseparator3,"open file")

configer(1)
Blender.Draw.Register(fenster,allg_ereignis,schalter_ereignis)

```