"FINAL PROJECT LAVA"

```
import maya.cmds as mc

import mtoa.utils as mutila

import os

import mtoa.utils as mutils




home = os.getenv("HOME")


print(home)


mc.file(new = True, force=True)

mc.file(rename =
os.path.join(os.getenv("HOME"),"maya","projects","default","scenes","VSFX502_FinalPart1
_Chaithanya_Kasireddy.mb"))


#CREATE_PLANE

polyModel = mc.polyPlane(sw = 100,sh = 100)

mc.scale(100,100,100)


#LAVA_SHADER

ailavaBaseNode = mc.shadingNode("aiStandardSurface", asShader = True)

mc.setAttr(ailavaBaseNode+".base",1)

mc.setAttr(ailavaBaseNode+".baseColor",0,0,0, type = "float3")

mc.setAttr(ailavaBaseNode+".specular",1)
mc.setAttr(ailavaBaseNode+".specularColor",0.571,0.7891,1, type ="float3")

mc.setAttr(ailavaBaseNode+".specularRoughness",0.230)

mc.setAttr(ailavaBaseNode+".emission",8)


#Create shading group and assign surface and displacement shader

sg = mc.sets(name = ailavaBaseNode+"SG", empty = True, renderable = True, noSurfaceShader=
True)

mc.connectAttr(ailavaBaseNode+".outColor",sg+".surfaceShader")




#Assign Shading group to plane

mc.select(polyModel[0])

mc.hyperShade(assign = sg)


#####OSL_Shading#####

oslShader = mc.shadingNode("aiOslShader", asShader = True)


#Ramp color organe connected to aicolorCorrect

#Place 2d Texture

"""

place2dText1 = mc.shadingNode("place2dTexture", asUtility = True)

#Create color ramp

aiRamp = mc.shadingNode("aiRampRgb",asShader = True)

mc.setAttr("aiRampRgb1.ramp[0].ramp_Color", 1,0.11,0)

mc.setAttr("aiRampRgb1.ramp[1].ramp_Color",1,0.2,0)
mc.setAttr("aiRampRgb1.ramp[1].ramp_Position",1)

mc.connectAttr(aiRamp+".outColor", aiNoise1+".color1")


#Connect Placae 2D Texture to color ramp

mc.connectAttr(place2dText1+".outUV",Ramp+".uvCoord")

"""

#Create a noise for the lava

aiNoise = mc.shadingNode("aiNoise", asTexture = True)

mc.setAttr(aiNoise+".octaves",30)

mc.setAttr(aiNoise+".distortion",5.677)

mc.setAttr(aiNoise+".lacunarity", 3.606)

mc.setAttr(aiNoise+".amplitude", 1)

mc.setAttr(aiNoise+".color2", 0.296774,0.043121,0, type = "float3")

#Connect OslShader to aiNoise

mc.connectAttr(oslShader+".outValue",aiNoise+".color1")

"""

mc.connectAttr(Ramp+".outColor",aiNoise+".color1")

mc.connectAttr(aiRamp+".outColor",aiNoise+".color1")

"""

#Color correct the noise

colorCorrectionLava = mc.shadingNode("aiColorCorrect", asUtility = True)

mc.connectAttr(aiNoise+".outColor",colorCorrectionLava+".add")

#Connect color correction to main shader emission

mc.connectAttr(colorCorrectionLava+".outColor",ailavaBaseNode+".emissionColor")
```

```
#aiCellNoise

#Create Noise Pattern

cellNoise = mc.shadingNode("aiCellNoise", asTexture = True)

mc.setAttr(cellNoise+".pattern", 6)

mc.setAttr(cellNoise+".octaves",10)

mc.setAttr(cellNoise+".lacunarity",2.546)

mc.setAttr(cellNoise+".amplitude",1)

mc.setAttr(cellNoise+".scaleX",10)

mc.setAttr(cellNoise+".scaleY",5)

mc.setAttr(cellNoise+".scaleZ",10)

mc.setAttr(cellNoise+".offsetX",10)

mc.setAttr(cellNoise+".offsetY",0)

mc.setAttr(cellNoise+".offsetZ",5)

#Color Correct pattern

colorCorrectionRocks = mc.shadingNode("aiColorCorrect",asUtility = True)

mc.connectAttr(cellNoise+".outColor",colorCorrectionRocks+".input")

mc.setAttr(colorCorrectionRocks+".gamma",0.065)

mc.setAttr(colorCorrectionRocks+".hueShift",0.084)

mc.setAttr(colorCorrectionRocks+".contrast",0.968)

mc.setAttr(colorCorrectionRocks+".contrastPivot",0)

mc.setAttr(colorCorrectionRocks+".invert",1)

mc.setAttr(colorCorrectionRocks+".alphaMultiply",0)

#ai range node
```

```
colorRange = mc.shadingNode("aiRange", asUtility = True)

mc.connectAttr(colorCorrectionRocks+".outColor",colorRange+".input")

mc.setAttr(colorRange+".inputMin",0.184)

mc.setAttr(colorRange+".inputMax",0.752)

mc.setAttr(colorRange+".outputMin",0)

mc.setAttr(colorRange+".outputMax",2.147)

mc.setAttr(colorRange+".contrast",0.8)


#Connect aiRange to a mask to the aiColorCorrect

mc.connectAttr(colorRange+".outColorR",colorCorrectionLava+".mask")

#Connect shading group and assign surface and displacement shader

DisplacementRock = mc.shadingNode("displacementShader", asShader = True)

mc.connectAttr(colorRange+".outColorR",DisplacementRock+".displacement")

mc.setAttr(DisplacementRock+".scale",-0.008)

mc.setAttr(DisplacementRock+".aiDisplacementPadding", 0.1)

mc.connectAttr(DisplacementRock+".displacement", sg+".displacementShader")



#Create Noise

aiNoise = mc.shadingNode("aiNoise", asTexture = True)

mc.setAttr(aiNoise+".octaves",20)

mc.setAttr(aiNoise+".distortion",5)

mc.setAttr(aiNoise+".lacunarity",1.470)

mc.setAttr(aiNoise+".amplitude",1)
mc.setAttr(aiNoise+".scaleX",15)

mc.setAttr(aiNoise+".scaleY",10)

mc.setAttr(aiNoise+".scaleZ",15)

mc.setAttr(aiNoise+".color1",0,0,0, type = "float3")

mc.setAttr(aiNoise+".color2",1,1,1, type = "float3")

#Create Bump node

rockBump = mc.shadingNode("bump2d",asUtility = True)

mc.setAttr(rockBump+".bumpDepth",25)#rockTextBump

mc.connectAttr(aiNoise+".outColorR", rockBump+".bumpValue")

#connect bumb to main shader(NormalCamera)

mc.connectAttr(rockBump+".outNormal",ailavaBaseNode+".normalCamera")

#SkyDomeLighting

skydome = mutils.createLocator('aiSkyDomeLight', asLight=True)

#Hdri

filetex1 = mc.shadingNode("file", asTexture = True)

mc.connectAttr(filetex1+".outColor",skydome[0]+".color")

hdriFile =
os.path.join(home,"Documents","maya","Projects","default","sourceimages","belfast_sunset_p
uresky_4k.hdr")

mc.setAttr(filetex1+".fileTextureName",hdriFile,type = "string")

placeTex2= mc.shadingNode("place2dTexture", asUtility = True)

mc.defaultNavigation(connectToExisting = True, source = placeTex2, destination = filetex1)

#mc.connectAttr(filetex1+".outColor",skydome[0]+".color")

mc.select(skydome)

#mc.setAttr(skydome[0]+".intensity",0.55)
```

```
#seting Plane Subdivison

mc.select(polyModel[0])

mc.setAttr(polyModel[0]+".aiSubdivType",1)

mc.setAttr(polyModel[0]+".aiSubdivIterations",2)
```