

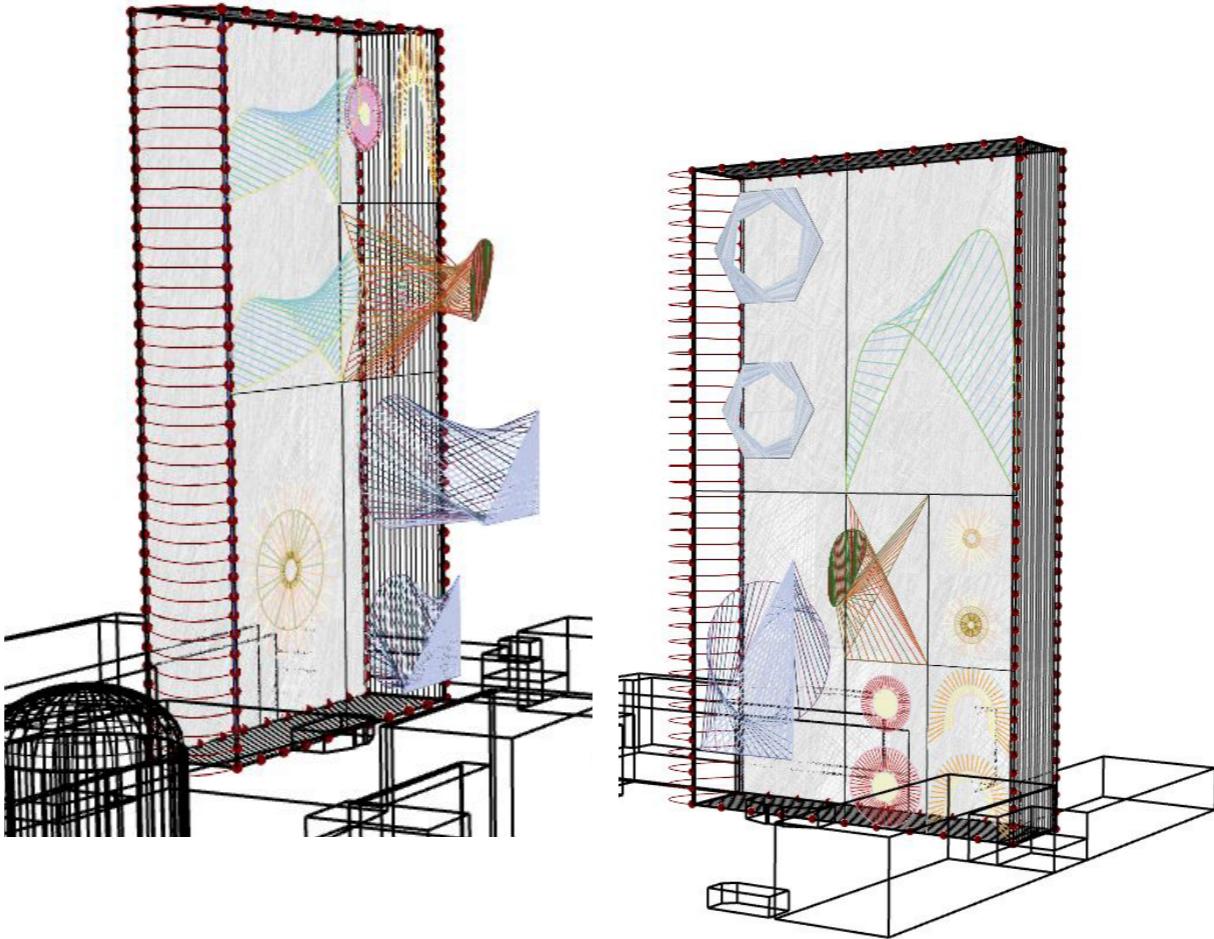


The Project

Als Projekt habe ich mich an der Buchreihe *Das Reich der Sieben Höfe* inspiriert. Die Fantasygeschichte spielt in einem Land mit dem Namen Prythian, welches in sieben Höfe/Lordschaften unterteilt ist (Frühlingshof, Sommerhof, Herbsthof, Winterhof, Hof des Morgens, Hof des Tages, Hof der Nacht). Ich habe mich entschieden, diese Höfe symbolisch an der Fassade des UN Gebäudes zu platzieren und das Gebäude an sich als Buch darzustellen.

The Person

Sarah J. Maas ist eine amerikanische Fantasy-Autorin, welche in New York aufgewachsen ist. Mit nur 16 Jahren schrieb sie die erste Fassung der *Throne of Glass* Buchreihe, durch welche sie internationalen Erfolg erntete. Die Buchreihen *Das Reich der Sieben Höfe* und *Crescent City* haben ihren internationalen Erfolg bestärkt.



The Code

Ich habe das UN Gebäude als Buch umkleidet. Die Fassade durch sieben Paneele geteilt, damit sich in jeweils einem Panel ein Symbol befindet. Um die Symbole mehrfach darzustellen, habe ich die Paneele jeweils nochmals unterteilt. Die Paneele sind auch so eingestellt, dass sie die Position auf der Fassade ändern bei neuladen des Scripts. Die Symbole sind farblich und größentechnisch angepasst, dass sie sich bei neuladen des Scripts verändern.

Finale

Lamprecht Katharina

DM2 W24 - Gruber 03

```

108     seiten_line2 = rs.ObjectColor(rs.AddCurve([coordsOh[1], coordsO[1][1], 2], col_2))
109     seiten_line3 = rs.ObjectColor(rs.AddCurve([coordsUh[1], coordsU[1][1], 2], col_2))
110     rs.DeleteObjects([crv_seiten_o, crv_seiten_u])
111
112 ##Buchstaben
113     for z in range (40):
114         p5 = dm.getUnoCoord(10, 0, z)
115         p6 = rs.VectorAdd(p5, rs.VectorScale ( lVec, -10 ) )
116         p7 = dm.getUnoCoord(10, 0, z)
117         spine_crvo_coords = [p5, p6, p7]
118         spine_crvo = rs.ObjectColor(rs.AddCurve(spine_crvo_coords, 2), col_2)
119
120 ##Punkteumrandung
121     for x in range (0, 11):
122         rs.ObjectColor(rs.AddPoint( dm.getUnoCoord(x, 0, 0) ), col_2)
123         rs.ObjectColor(rs.AddPoint( dm.getUnoCoord(x, 3, 0) ), col_2)
124         rs.ObjectColor(rs.AddPoint( dm.getUnoCoord(x, 0, 39) ), col_2)
125         rs.ObjectColor(rs.AddPoint( dm.getUnoCoord(x, 3, 39) ), col_2)
126
127     for z in range (0, 40):
128         rs.ObjectColor(rs.AddPoint( dm.getUnoCoord(10, 0, z) ), col_2)
129         rs.ObjectColor(rs.AddPoint( dm.getUnoCoord(10, 3, z) ), col_2)
130         rs.ObjectColor(rs.AddPoint( dm.getUnoCoord(0, 0, z) ), col_2)
131         rs.ObjectColor(rs.AddPoint( dm.getUnoCoord(0, 3, z) ), col_2)
132
133 #####
134
135 random.shuffle( allPanels )
136 def dividePanel ( panel=allPanels[0], maxPanels=25, minHig=4.0 ) :
137     dm.esc()
138
139     if maxPanels<=4: maxPanels=4
140     if rs.Distance ( panel[0], panel[1] ) > minHig*2:
141         allPanels.remove ( panel )
142         phid = dm.pntInbetween(panel[0],panel[2])
143         allPanels.append(panel[0], dm.pntInbetween(panel[0],panel[1]), phid, dm.pntInbetween(panel[3],panel[0]))
144         allPanels.append(dm.pntInbetween(panel[0],panel[1]), panel[1],dm.pntInbetween(panel[1],panel[2]), phid)
145         allPanels.append(phid, dm.pntInbetween(panel[1],panel[2]), panel[2], dm.pntInbetween(panel[2],panel[3]))
146         allPanels.append(dm.pntInbetween(panel[0],panel[3]), phid, dm.pntInbetween(panel[2],panel[3]), panel[3])
147
148     if len(allPanels) <= maxPanels :
149         dividePanel ( panel=random.choice( allPanels[0: int(len(allPanels)*0.666)] ), maxPanels=maxPanels )
150
151
152 dividePanel(panel=allPanels[0], maxPanels=7, minHig=12.0 )
153 print " :: len(allPanels)",len(allPanels)
154
155 #####
156 ##Fruehling
157 def springpanel ( panel, normPanel = 1 ) :
158
159 ##Sommer
160 def summerpanel ( panel, normPanel = 1 ) :
161
162 ##Herbst
163 def autumnpanel ( panel, normPanel = 1 ) :
164
165 ##Winter
166 def winterpanel(panel, anzahl=50, blend=5, blendSize=0.9):
167
168 ##Morgen

```