

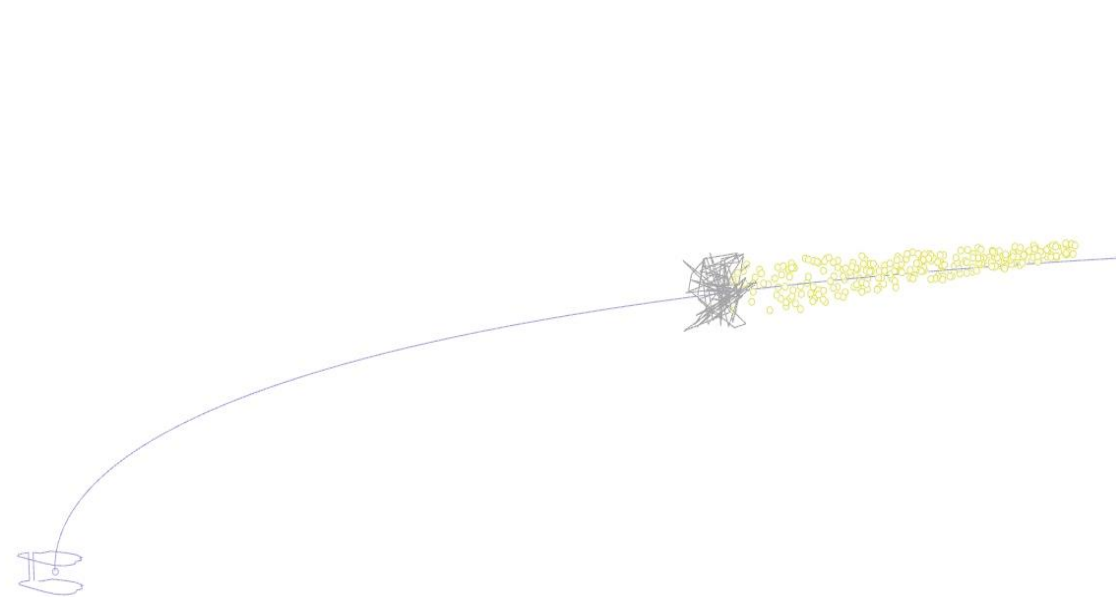
Final Project DM2

Homage to Urania - Meteorites over Vienna



The Person

Urania is the Muse of Astronomy in Greek mythology, her name derived from Uranos, the personification of the sky. She is often depicted with symbols such as a celestial globe and a pointer, which she typically holds in her hands, sometimes gesturing with the pointer towards a globe on the ground. In Late Antiquity, the Pythagorean concept of the harmony of the spheres was elaborated upon, with each Muse assigned to a specific celestial sphere. In this framework, Urania was associated with the fixed stars, representing the highest tone in the harmony of the spheres.



The project

As the foundation for my project, I used the "Urania" Observatory in Vienna. Named after Urania, the Muse of Astronomy and the stars, the observatory is one of the oldest public observatories in Austria and embodies the spirit of exploration and the quest for knowledge about the cosmos. In my project, I envisioned meteorites descending upon the observatory, each following different celestial paths. As they shower down upon the building, they create a stunning spectacle, illuminating the sky and cover the building the process. This project pays homage to Urania, not only as a symbol of celestial navigation and discovery but also as a representation of humanity's enduring fascination with the stars. The meteoric paths are a reminder of the dynamic nature of the universe, where cosmic events continuously unfold, shaping our understanding of space and time.

The code

```
targets = [ centerUrania, rs.VectorAdd(centerUrania, dm.randVec(50,100)) ]
###Mittelpunkt + random Vec ergibt target
frames = 30# 250
path = "C:/Users/annas/OneDrive/Dokumente/uni-dokumente/dm2/meteoriten_fotos/"

####Meteorit Bahnen####

anzMet = 3

meteoriten = []
bahnen = []
schweife = []

for i in range(anzMet):
    for tar in targets[0:1]:
        pOben = rs.VectorAdd(tar, [0,0, random.uniform(200,500)] )
        pStart= rs.VectorAdd(pOben, [1000,1000,0] )
        kometBahnCoords = [pStart, pOben, tar ]
        #rs.AddCurve( kometBahnCoords, 1)
        bahn = rs.AddCurve( kometBahnCoords, 2) ###Flugbahn
        coords = rs.DivideCurve( bahn, frames, 0) ####Flugbahn wird in abhaengigkeit von
        frames geteilt
        rs.ObjectColor(bahn, [255,255,255])

        meteoriten.append(bahn)
        bahnen.append(coords)
        for cor_index, cor in enumerate(coords[0:]): ###Flug von Punkt A nach Punkt B
            for meteorit in rs.ObjectsByName("meteorit"):
                if meteorit not in flach_Meteorit:
                    rs.DeleteObjects(rs.ObjectsByName("meteorit*"))
                    meteoritCoords = []

####Meteorit####
        for i in range(80):
            vecX = dm.randVec(-30, 30)
            corX = rs.VectorAdd( cor, vecX )
            #rs.AddPoint( corX )
            meteoritCoords.append( corX )
            meteorit = rs.AddCurve( meteoritCoords, 1)
            rs.ObjectColor(meteorit, [102,102,102])

####Meteorit flachdruecken####
        flach_Meteorit = []
        if cor_index == len(coords)-1:
            stauchung = random.uniform(0.1,0.8)
            rs.ScaleObject(meteorit, cor, [1,1,stauchung])
            flach_Meteorit.append(meteorit)

####tangente = schweif#####
        paramX = rs.CurveClosestPoint(bahn, cor)
        tanVec = rs.CurveTangent( bahn, paramX )
        print rs.VectorLength(tanVec) ###ueberpruefung ob normalvektor
        schweifLenx = random.uniform(200,500)
        tanVec = rs.VectorScale( tanVec, -schweifLenx)####Einheitsvektor wird um die
        Schweiflenx vergroesert
        schweif = rs.AddLine( cor, rs.VectorAdd(cor,tanVec) )
        schweife.append(schweif)

####schweif koma####
        divY = 20
        coordsKoma = rs.DivideCurve(schweif, divY, 0)

        min_distance = 5
        max_distance = 20

        CoordsKoma = []
        for index, cor in enumerate(coordsKoma[2:]): ###Start bei zweiten Punkt
            if cor_index < len(coords)-2: ###Sobald meteorit auf
                vorletzten punkt von coords angekommen ist, wird koma nicht mehr generiert
                auskrag = dm.reMap( index, 0, divY-2, max_distance, min_distance)
                ###Hilfe von chatgpt bei remap befehl
                for i in range(15):
                    vecY = dm.randVec(-auskrag,auskrag)
                    corY = rs.VectorAdd(cor, vecY)
                    Koma = rs.AddPoint( corY)
                    rs.ObjectColor( Koma, [200,200,0])
                    rs.ObjectPrintWidth( Koma, 0.5)

                CoordsKoma.append( Koma)

        rs.ObjectName( schweif, "meteorit schweif")
        rs.ObjectName( meteorit, "meteorit")
        rs.ObjectName(CoordsKoma, "meteorit koma")
        rs.ObjectName(flach_Meteorit, "flach_Meteorit")
        rs.ObjectName(bahn,"bahn")
        rs.DeleteObject(schweif)
        rs.Redraw()
        dm.esc()
```