Arcos - Codebook

This codebook is documentation for the accompanying Houdini file and has the same structure.

With special thanks to the man, the myth, the legend:
Akos

Course: BK7083

Students:
Jasper van Beusekom
Piet de Kok
Vera Lensvelt
Catalina Gormaz Peña

Environment – City Generation

This code is designed to create geometries from a list of coordinates (Json File).

Input:	9-312-520.city.json, 9-312-524.city.json
Output:	City files separated in the right geometries

- This code was made available by the course staff.

Environment – Grid

This code creates the voxelgrid inside our designated area. It does so based on the voxel outline created at the start of the code.

Input:	Building Outline, Voxel
Output:	Voxelgrid & Voxelpoints

- 1) Place a pointgrid, which is transformed to the correct area and copied upwards.
- 2) Create volume out of building outline (IN: Building Outline)
- 1+2) Selects grid points that are included in building volume.
  Remove all other points
  OUT: VoxelPoints
- Replace points with voxels (IN: Voxel)
  OUT: Voxelgrid

Environment Analysis - Sun Position

This code creates the sun paths for every hour per day from a data file of sun coordinates in Rotterdam. It outputs the sun paths as a sphere geometry.

	Sun Position Table Rotterdam
Output:	Sun Paths

Create a dome with radius r

IN: Sun Positions Table Rotterdam

Function Sun path

- Iterates through days and hours
- Store sun positions for each hour of the day in spherical coordinates
- Add positions as geometry

Create Sphere

- Add Sphere to all sun positions points
- OUT: Sun Paths

Environment Analysis – Shadow Casting

Input:	Sun Paths, VoxelPoints, City
Output:	Shadow Analysis Voxels & Points

IN: Sun Paths

IN: VoxelPoints

IN: City

Attribute Wrangle

- Check if a ray cast from current position P (in voxel points) intersects with any geometry of City
- If no intersection in first direction, check reverse direction (-dir)
- If no intersection is found -> count += 1
- Outputs count / total of rays sent from point -> Ratio

Sorting

- Sorts ratio’s from minimum to maximum
- Find range (max- min)
- If ratio < threshold -> Remove Voxel from VoxelGrid
- If ratio < threshold:
  Give colour ranging from 0 to 1
  OUT : Shadow_Analysis_Points

Copy Voxels to Points

OUT: Shadow_Analysis_Voxels, Shadow_Analysis_Voxels_Negative

Environmental Analysis – Sunlight Analysis

Input:	Sun Paths, VoxelPoints , City, Voxel
Output:	ColouredVoxels, VoxelPoints

IN: Sun_Paths

IN: VoxelPoints

IN: City

Attribute wrangle ; Calculate Sunlight

- Check if a ray cast from current position P (in voxel points) intersects with any geometry of City
- If casted ray is not blocked (no intersection) -> Count += 1
- Outputs count / total of rays sent from point -> Ratio
- Add ratio as an attribute (analysis 1)

Sorting

- Sorts ratio's from minimum to maximum
- Find range (max- min)

  OUT: VoxelPoints

IN: Voxel

- Copy Voxels to points

  OUT: Coloured Voxels

Environmental Analysis – Daylight Analysis

<table>
<tr><td>Input:</td><td>Skydome_points, VoxelPoints , City, Voxel</td></tr>
<tr><td>Output:</td><td>ColouredVoxels, VoxelPoints</td></tr>
<tr><td colspan="2">IN: Skydome_points
IN: VoxelPoints
IN: City
Attribute wrangle ; Calculate daylight
➢ Check if a ray cast from current position P (in voxel points) intersects with any geometry of City
➢ If casted ray is not blocked (no intersection) -> Count += 1
➢ Outputs count / total of rays sent from point -> Ratio
➢ Add ratio as an attribute (analysis 1)
Sorting
➢ Sorts ratio’s from minimum to maximum
➢ Find range (max- min)
OUT: VoxelPoints
IN: Voxel
➢ Copy Voxels to points
OUT: Coloured Voxels</td></tr>
</table>

Environmental Analysis – Height

Input:	VoxelPoints
Output:	VoxelPoints2

IN: VoxelPoints (from Sunlight Analysis)

- Retrieve Height Parameter from VoxelPoints

Sorting

- Sorts ratio’s from minimum to maximum
- Find range (max – min)
- Give max value = 1 & min value = 0
- Give Colour (red to green) based on y-value

OUT: VoxelPoints2

Seed finding

Input:	Function table, point grid
Output:	Seeded points

Loops over all points

Calculates the loss for each weight

Stores the best point for each function

For a more node-oriented approach, see the nodes in the Houdini file.

Growing algorithm

Input:	Last frame
Output:	New frame

```
Loops over all functions
    Loops over all voxels that are within a 1 voxel radius of this function’s points
        Keeps the best possible point to grow to and grows to it
```

For a more node-oriented approach, see the nodes in the Houdini file.