AnnuOWL

Annual Evaluation of Occupant Wellbeing through Lighting

15 Aug 2023

This is the documentation of AnnuOWL: an opensource workflow in Rhino Grasshopper developed for performing Radiance-based annual lighting simulations on horizontal and vertical planes, and evaluating visual and non-image-forming effects of light on occupant comfort and wellbeing, along with energy performance via metrics for light sufficiency. As a new branch in the OWL family of tools, AnnuOWL features 64 original Python-based grasshopper components. This Radiance grid-based simulation workflow is designed to support early stage interventions in lighting and facade design.

This tool requires minimal inputs from the user when evaluating the following occupant-centered metrics on an annual basis: **Lighting Sufficiency** *(using editable recommendations of EN17037 for spatial daylight autonomy)*, **Non visual/circadian potential** *(using annual hourly evaluation of Circadian Stimulus, a metric proposed by Lighting Research Centre)*, and **Protection from Discomfort Glare** *(using editable recommendations of EN17037 for glare protection using DGP as a metric)*. These metrics, evaluated for each user position and view, are displayed via an intuitive visualization infographic – proposed as OVNI diagrams. Evaluation and visualization of metrics for every occupant's [Visual] **comfort**, [Circadian] **health** and **well-being**, along with **energy use** for lighting, can help in early stage interventions in façade and lighting design. Deeper investigation via hourly heatmap diagrams is also supported for each occupant position. Additionally, grid-based evaluation of various climate-based daylight metrics on the horizontal plane is supported, as well as the evaluation of spatial daylight autonomy of the entire space -- as a singular comprehensive metric for daylight performance and compliance.

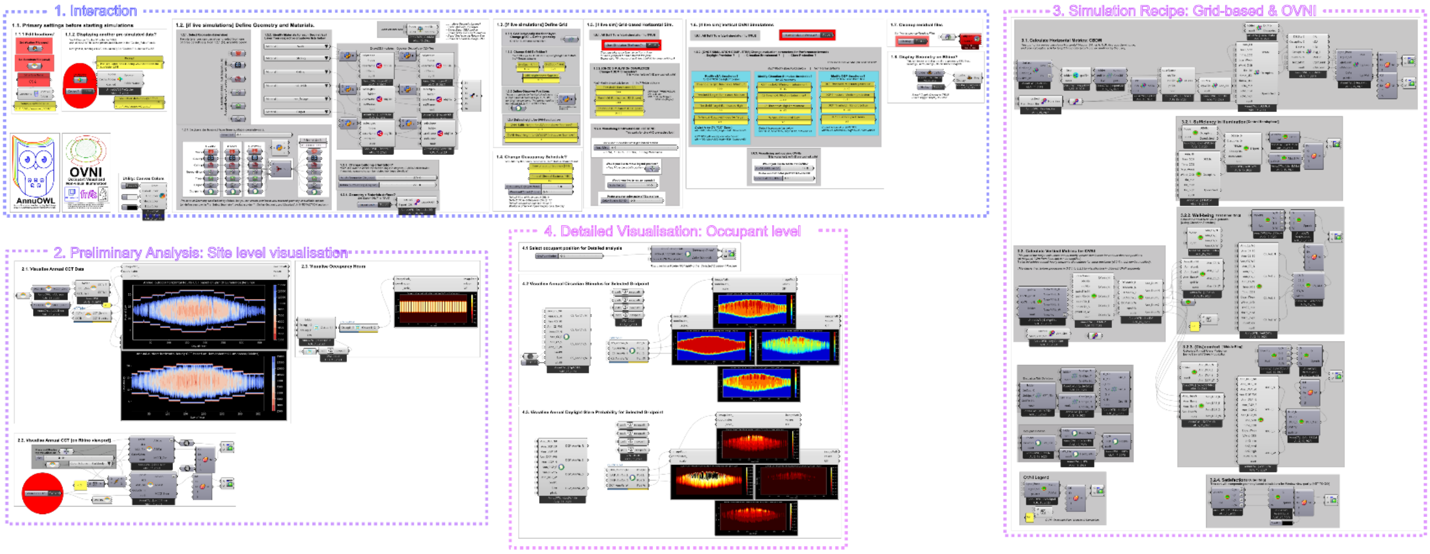
This tool is is developed by Marshal Maskarenj at LAB UCLouvain, along with Sergio Altomonte, as part of FNRS funded project SCALE (Shading Control Algorithms from Luminance-based Evaluations).

Contents

Overview

AnnuOWL is supported by 64 original Grasshopper Python components, and its workflow is divided into 4 major sections: the Interaction section, Preliminary site-level analysis, Simulation pipeline, and detailed analysis at occupant level. The components on the Grasshopper panel are presented in the figure below.

A snapshot of the entire grasshopper workflow, and the placement of components within these sections, is presented in figure below.



Functionality of AnnuOWL

AnnuOWL is a radiance-based tool, and includes multiple models and standard-based protocols for evaluating metrics of visual and non-visual lighting performance.

As user-input, AnnuOWL requires building geometry, weather data *(as *.epw file)*, and pre-calculated annual spectral data as *(*.aowl file)*. Other parameters, such as: *occupancy schedule, table height, head height, grid* *distribution*, and *thresholds for metrics to be evaluated,* are pre-populated with standard defaults, but are editable to match user needs. With these inputs, AnnuOWL can evaluate the following metrics:

- Grid-based CBDMs (DA, cDA, UDI) and Average illuminance through occupied hours.
- Occupant-centered DA performance (High/Med/Min/Non-Compliant) at table height using editable EN17037 target thresholds.
- Occupant-centered Circadian potential as non-visual performance (High/Med/Min/NC) at eye-height in four orientations.
- Occupant-centered Protection from discomfort glare (High/Med/Min/NC) at eye-height via editable EN17037 thresholds.
- Annual grid-based Spatial Daylight Autonomy compliance throughout floor area (High, Minimum, Medium).

These metrics are displayed in an intuitive visualization proposed as OVNI diagrams. Early stage interventions in façade and lighting design may be supported through such integrated visualizations encompassing occupant [Visual] comfort, [Circadian] health and well-being, and energy use for lighting.

The OVNI Diagrams

The OVNI diagram is an infographic placed at every occupant's position, which describes the occupant-centered annual performance of the design – towards energy use, comfort, and circadian health and well-being.

Each OVNI includes a hemisphere, surrounded by three rings. Each ring is sub-divided into four segments, facing North, East, South, and West orientations.

- The hemisphere represents sufficient illuminance at table height, and uses radiance grid-based evaluation for occupied hours.

Sub-divisions on the rings represent vertical viewing positions for four principal orientations, evaluated at the occupant eye-height.

- The inner ring represents Circadian potential & Non-visual performance, as proxy for occupant health & well-being.
- The middle ring represents protection from discomfort glare for each orientation, and describes occupant comfort.
- The outer ring represents view quality. This is yet to be incorporated, and will describe occupant satisfaction.

Installation Instructions

Most installation instructions below are the same as for the initial OWL tool, so if you already work with OWL, you can skip few steps and jump directly to export OBJ instructions below.

Installing Python

1. Python 2.7 can be downloaded from this link.
2. The following need to be added to path: <<C:\Python27>> and <<C:\Python27\Scripts>> more information on adding path variables can be accessed here.
3. Using Pip, the following components must be installed: *numpy*, *scipy*, and *matplotlib*... Sample command for pip installation of these packages is like <<pip install numpy>>

Installing Radiance

1. Radiance needs to be installed on Windows. It is available at this location. The latest release as of this day is Radiance 5.4a
2. Radiance needs to be added to Path, if not automatically done in installation process.

Installing latest Ladybug 1.xx for access to Ladybug Image Viewer component

1. The latest Ladybug 1.x can be downloaded from this location (latest version is 1.6, as of this day).
2. With the folder unzipped, and *installer.gh* opened in Grasshopper – the installation instructions can be followed.

Installing GH_Cpython component

1. The component can be downloaded in a zip file from https://www.food4rhino.com/en/app/ghcpython.
2. In Grasshopper, appropriate folder needs to be opened as File > Special Folders > Component Folders
3. From the zip file, *GH_CPython* and *FastColoredTextBox.dll* need to be pasted into Libraries folder

Rhino-side settings for exporting OBJ files

In order to correctly export the geometry (via ExportOBJ component), the following settings need to be done on Rhino:

- With Rhino open, draw any surface (e.g. using *Rectangular Plane: Corner to Corner*) on Rhino Viewport.
- With the object selected, go to File > Export Selected > [Save as type: OBJ] > and click on the Options box that appears underneath.
- Under Formatting, uncheck the “Map Rhino Z to OBJ Y” option.
- Check the “Always use these settings. Do not show this dialog again.” Option. Click OK.
- Save the OBJ with any file name.

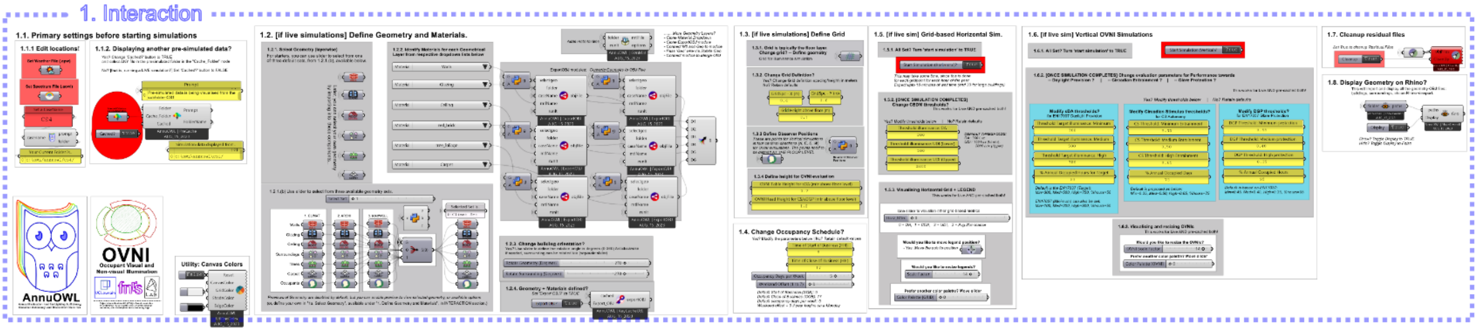
Having done these settings once, Rhino will export OBJs in the fashion intended by the ExportOBJ component here onwards.

The following part of this document presents the 4 major sections of the workflow, and the elements of each of the 64 Grasshopper Python components nested under each section.

Following the detailed description of the workflow and components, various supporting components for customizing the simulation outputs (using sliders, panels, etc.) are presented. These components can be useful while modifying the default settings, and performing simulations for evaluating specific demands.

Going through detailed description of the components, as well as the different sections of the workflow, is optional. You may choose to skip directly to user-customization section. However, it is recommended to go through the interaction section to learn about minimum required user-inputs.

1. Interaction Section



The user-inputs for performing simulations are expected only in this section.

The main inputs required from users, are: location of the weather file *(*.epw)*, location of the spectrum file *(*.aowl)*, and the case-name (which generates a new subfolder with the case name in the root folder: *C:\OWL\annuowl*). Additionally, users need to select layers of their built environment (as Walls, Glazing, carpet, etc.) into specific geometry components (or use slider to choose from among default options) and define the material for each geometry layer from the dropdown material menu. In order to perform parametric simulations for various orientations, the slider input may also be used to reorient elements of the geometry.

As described previously, the workflow also includes editable parameters e.g. occupancy schedule, table height, head height (*i.e. the vertical distance between the floor and an occupant's eyes*), grid distribution (*sensor spacing in the x- and the y- dimensions for the grid, as well as the height of evaluation grid above the floor*), and thresholds for various metrics (*DA and UDI illuminance thresholds, sDA and DGP compliance thresholds etc.*). These parameters are pre-populated with standard defaults, but may be customized to match user needs.

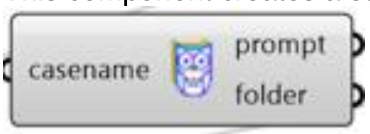
Options for visualization, such as use of different color palettes, the placement of legends, the resizing of the OVNI diagrams, etc. are also provided, and may be used for generating graphics to suit specific needs.

The 8 original python-based components in this section, their descriptions, input and output nodes, as well as their key associations with other components in the workflow is presented as follows. This also includes links to the openly available code for each of these components, as well as hyperlinks to other associated components.

1.1.1. GenDir

Component Name: GenDir [Code]

Component Description:
This component creates a subfolder in your C:\OWL/annuowl/ folder for saving simulation data.



Inputs:
casename: The name of this specific case, needs to be unique else it will overwrite previous simulation data

Outputs:
prompt: User text
folder: location of folder for further processing

Key Connections (output):
PreCache

1.1.2. PreCache

Component Name: PreCache [Code]

Component Description:
This component helps visualise pre-simulated data in case needed. While running live simulation for fresh cases, the 'Cached' button can be set to FALSE.

Inputs:
Folder: Link to the current folder
Cache_Folder: (if selecting a pre-simulated solution) Any file in the pre-simulated folder
Cached: YES if displaying presimulated results, NO if running live simulation

Outputs:
Prompt: Some information about data processing
FolderName: Location of files to process, depending upon Cached or Live

Key Connections (Input):
GenDir
Key Connections (output):
CacheDltMtxH, CacheDltMtxV, CacheCSMtxV, CacheDGPMtxV, CacheGrDef, GrDefFile, CacheHPts, HptFile, ShowGeomA, ImportSRF

1.2.2. ExportOBJ

Component Name: ExportOBJ [Code]

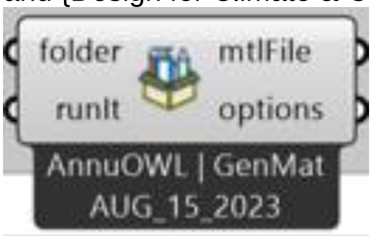
Component Description:
This component exports selected rhino geometry as .obj files for further processing in Radiance commandline. Groups of objects with similar material compositions (eg: Walls, or Glazing, etc) should be selected together in the geometry for parallel export -- ideally using Rhino layers.

Inputs:
selectgeo: Connect to a grasshopper Geometry object, which has Rhino objects/Layers via 'Set Multiple Geometries'.
folder: Location of the folder where the .obj files are to be saved.
caseName: Name for the specific case
mtlName: case-specific name for the objects with same material (eg: Walls, Glazing, etc). Look at GenMat options for the possible material names.
runIt: A boolean toggle to run this component.
Outputs:
objFile: Location of the saved object file for further processing.

1.2.2. GenMat

Component Name: GenMat [Code]

Component Description:
This component generates a 'materials.rad' file at the intended folder, for performing the obj2mesh function inside Obj2HDR component. mtlFile connects to the Obj2HDR input, and options-output contains possible materials for using as mtlName in ExportOBJ component. Regenerates the materials.rad file in su2rad repository (maintained by Thomas Bleicher) at https://github.com/tbleicher/su2rad/blob/master/su2rad/su2radlib/ray/materials.rad
Additional material data (3-channel) taken from spectral-database (spectraldb.com) maintained by Alstan Jakubiec and {Design for Climate & Comfort Lab} at UToronto.



Inputs:
folder: Location of the folder where the materials.rad file needs to be saved.
runIt: A boolean toggle to run this component.
Outputs:
mtlFile: location of the generated materials.rad file to be connected with Obj2HDR component input.
options: Connect to a panel to view the possible materials, which may be used as mtlName in the ExportOBJ component.
Connections (output):
DCMtxH, DCMtxV

1.2.4. KeyCacheOBJ

Component Name: KeyCacheOBJ [Code]

Component Description:
A logical component that feeds into the runIt of ExportOBJ components, taking inputs both from dedicated toggle button, but also from PreCache component's Toggle. If Precache is True, this results into 'False' output irrespective of dedicated toggle key. If PreCache is False, it results into True or False depending upon state of dedicated Toggle key.

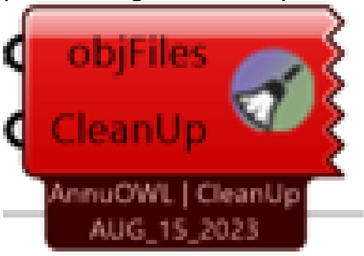
Inputs:
cached: The toggle key of PreCache. This is the dominant key input.
Export_OBJ: The dedicated toggle key. This is the secondary key input.
Outputs:
exportOBJ: This connects to the runIt node of ExportOBJ component(s).
Key Connections (output):
ExportOBJ, GenMat, ExportSRF, ImportSRF

1.7. CleanUp

Component Name: CleanUp [Code]

Component Description:

Each simulation run generates multiple interim files, many of which are not useful for data visualization and processing. This component removes such heavy files from the simulation folder.



Inputs:
objFiles: This links to the (merged) objFiles generated each simulation.
CleanUp: A toggle button, set TRUE for cleaning the residual bulky interim files.
Outputs:
None
Key Connections (Input):
ExportOBJ

1.8. ShowGeomA

Component Name: ShowGeomA [Code]
Component Description:
The first one of the two components designed to display pre-cached geometry of the scene on the Rhino canvas. Objects such as windows, ceiling, glazing, etc., saved as .obj files in each pre-simulated (or live) folder, are imported and displayed. The output node connects to the second Show-Geometry component.

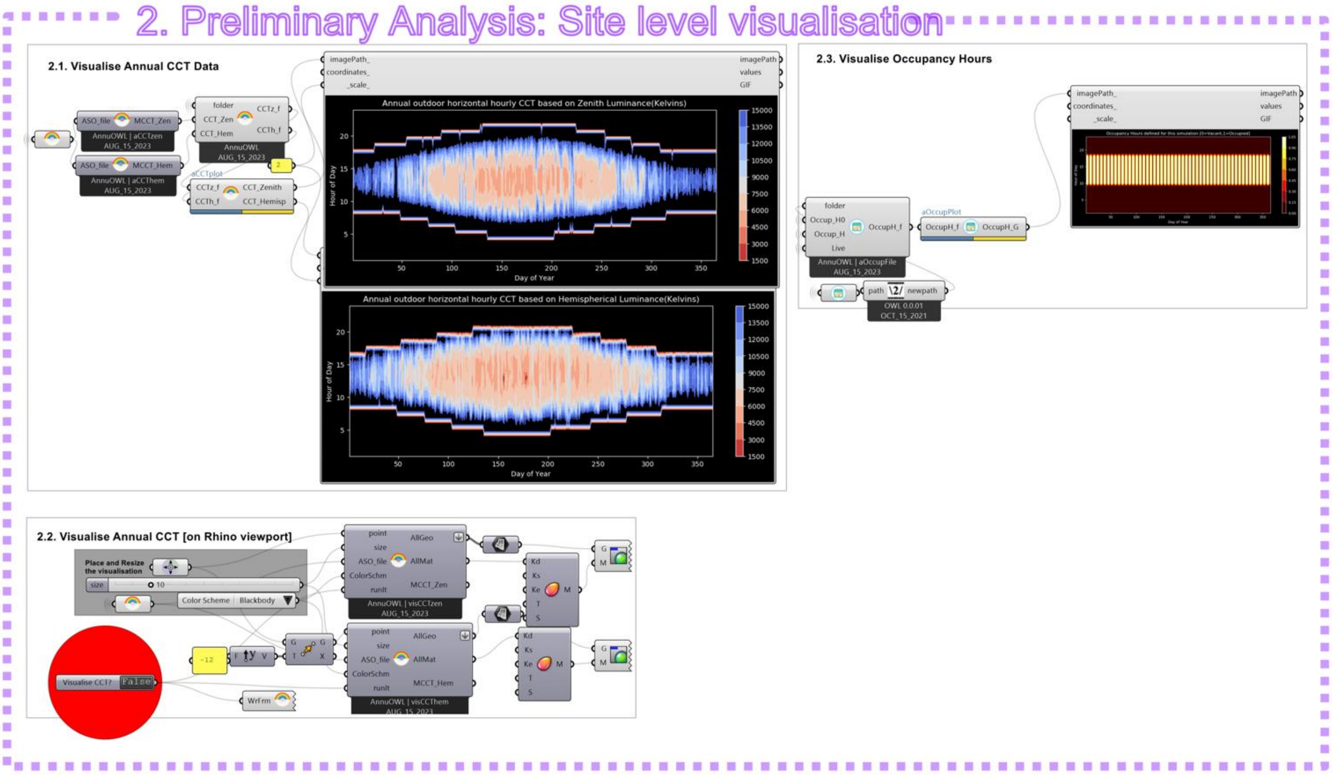
Inputs:
folder: Location of the folder with the .obj files.
Outputs:
paths: the paths of each .obj file in the pre-simulated folder. This node connects to the second Show-Geometry component
Connection (Input):
PreCache
Connection (output):
ShowGeomB

1.8. ShowGeomB

Component Name: ShowGeomB [Code]
Component Description:
The second of two components designed to display pre-cached geometry of the scene on the Rhino canvas. If Display is set to TRUE, Objects saved as .obj files in each pre-simulated (or live) folder, are imported and displayed.

Inputs:
paths: the paths of each .obj file in the pre-simulated folder. This node takes data from the first Show-Geometry component
Display: A True/False button for displaying geometry in Rhino. Set TRUE to display, FALSE to hide.
Outputs:
Geo: The geometry imported from the pre-simulated folder.
Connections (Input):
ShowGeomA

2. Preliminary Analysis: Site level visualization



This section includes 8 components for visualizing site-level annual data. 4 components in the 'Visualise Annual CCT data' subsection support generation of annual heatmaps of horizontal CCT. These include components for generating zenith-luminance based zenith CCT (using spectral sky models to convert hourly zenith luminance from annual *.epw file to respective CCT data), as well as for extracting hemispherical CCT data from precalculated *.aowl files (which uses the approach and models presented in the 2022 paper regarding the initial OWL tool, for evaluating the hourly hemispherical CCT). Two more components are developed to plot heatmaps in Rhino Viewport using customized palettes: 'CIE1931' and 'Blackbody radiation' color schemes. Additionally, components are developed to plot the annual occupancy schedule, defined by users and used for the simulations, as annual heatmaps.

Following suit, the 8 components in this section, their descriptions, their icons, the input and output nodes, as well as the key associations with other components in the workflow is presented as follows. This also includes links to the openly available code for each of these components, as well as hyperlinks to other associated components

2.1 Visualise Annual CCT Data

2.1. aCCTzen

Component Name: aCCTzen [Code]

Component Description:

Hemispherical CCT using Zenith Luminance. This component takes in the annual zenith luminance data from the weather file (in this case, the pre-simulated .aowl file, but can also take from the EPW file) and evaluates the Annual CCT for the zenith patch. The zenith-patch CCT is assumed as hemispherical CCT. For better resolution, use the aCCThem component's output.

The luminance to CCT conversion is done using the Diakite-Knoop 2021 model (Refer to https://doi.org/10.1177%2F1477153520982265 by Diakite-Kortlever and Knoop for more about this model).

Inputs:

ASO_file: Link to the Annual Spectral output file in the .aowl format for extracting zenith luminance through the annual hours.

Outputs:

MCCT_Zen: The CCT of the zenith patch through each hour of the year. Zenith CCT is being assumed equal to hemishperical CCT, use the [aCCThem] component for higher complexity in calculations

Connections (output):

aCCTfile

2.1. aCCThem

Component Name: aCCThem [Code]

Component Description:

Extraction of pre-calculated Hemispherical CCT from the .aowl file. The accompanying python-based utility can be used to convert the EPW weather data for any defined location to respective Annual Spectral data for unobstructed sky hemisphere. The utility follows the approach recommended in https://doi.org/10.1016/j.enbuild.2022.112012. A combination of Perez model with spectral sky models (https://doi.org/10.1177/1477153520982265) calculates hourly patch CCT, which is then converted to patch SPD following CIE015 standard. Merging patch SPD with cosine correction generates relative combined SPD of sky hemisphere. Tristimulus X,Y,Z values are evaluated and then chromaticity coordinates x, and y (and complementary z) are derived by factoring. McCamy's equation (1992) is used to derive CCT from chromaticity coordinates x and y for each hour.

Inputs:

ASO_file: Link to the Annual Spectral output file in the .aowl format for extracting zenith luminance through the annual hours.

Outputs:

MCCT_Hem: The hemispherical CCT through each hour of the year.

Connections (output):

aCCTfile

2.1. aCCTfile

Component Name: aCCTfile [Code]

Component Description:

This component reads the zenith-luminance-based and hemispherical precalculated CCTs through the year, and tabulates them into respective CSV files for plotting on annual heatmaps.

Inputs:

folder: location of the folder for saving simulation files.
CCT_Zen: the annual zenith-based CCT calculated by aCCTzen component.
CCT_Hem: precalculated annual hemispherical CCT extracted from .aowl file by the aCCThem component.

Outputs:

CCTz_f: CSV file with tabulated zenith-based annual CCT data (for further plotting)
CCTh_f: CSV file with tabulated hemispherical annual CCT data (for further plotting)

Connections (Input):

aCCTzen, aCCThem

Connections (output):

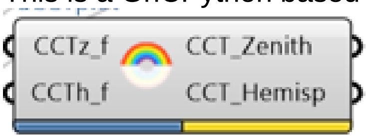
aCCTplot

2.1. aCCTplot

Component Name: aCCTplot [Code]

Component Description:

This is a GhCPython based component for plotting annual CCT heatmaps



Inputs:

CCTz_f: CSV file with tabulated zenith-based annual CCT data
CCTh_f: CSV file with tabulated hemispherical annual CCT data

Outputs:

CCT_Zenith: Location of the plot image file
CCT_Hemisp: Location of the plot image file

Connections (Input):

aCCTfile

2.2 Visualise Annual CCT [On Rhino Viewport]

2.2. visCCTzen

Component Name: visCCTzen [Code]

Component Description:

Visualise hemispherical CCT using Zenith Luminance. With similar functionality as aCCTzen component., this component visualises the heatmap of CCTs on Rhino canvas with real colors.

Two color options are possible with custom RGBs: Blackbody and CIE1931, where correlations are drawn for calculating RGBs based on respective CCTs for both palettes.

Inputs:

point: An anchor point on Rhino canvas for displaying the heatmap.
size: Scaling factor of the heatmap plot.
ASO_file: Link to the pre-calculated Annual Spectral Output (.aowl) file.
ColorSchm: Palette option for CCT to RGB, between 'Blackbody' and 'CIE 1931'
runIt: a boolean toggle switch for visualising on Rhino.

Outputs:
AllGeo: Geometry, for CustomPreview component.
AllMat: Material, for CustomPreview component.
MCCT_Zen: list of annual CCTs (similar to aCCTzen component)

Connections (output):
Custom Preview

2.2. visCCThem

Component Name: visCCThem [Code]

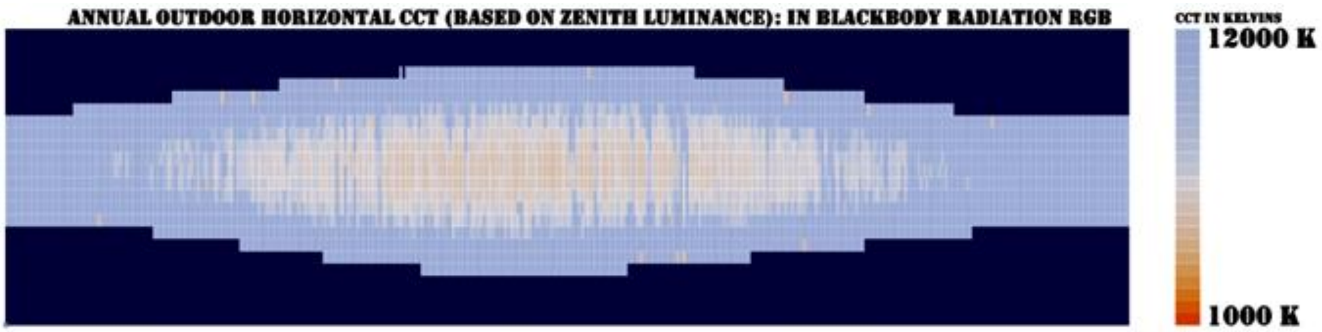
Component Description:
Visualise pre-calculated hemispherical CCT from the .aowl file.
With similar functionality as aCCThem component., this component visualises the heatmap of CCTs on Rhino canvas with real colors. Two color options are possible with custom RGBs: Blackbody and CIE1931, where correlations are drawn for calculating RGBs based on respective CCTs for both palettes.

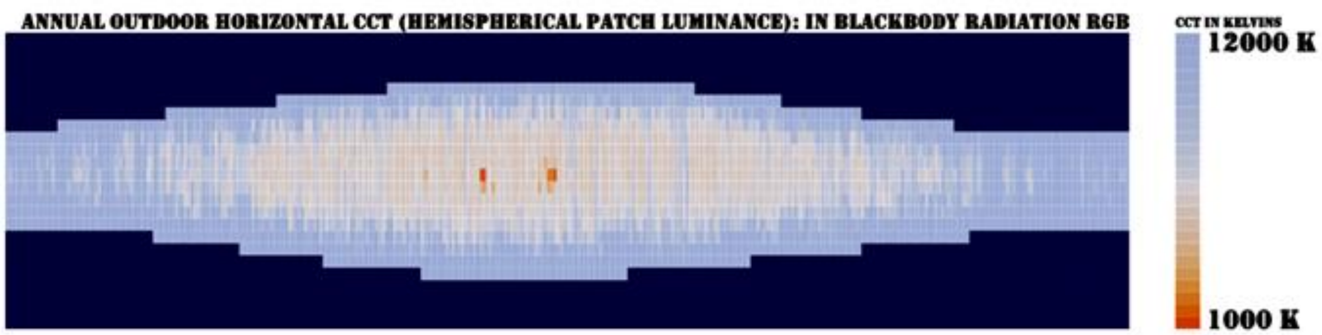
Inputs:
point: An anchor point on Rhino canvas for displaying the heatmap.
size: Scaling factor of the heatmap plot.
ASO_file: Link to the pre-calculated Annual Spectral Output (.aowl) file.
ColorSchm: Palette option for CCT to RGB, between 'Blackbody' and 'CIE 1931'
runIt: a boolean toggle switch for visualising on Rhino.

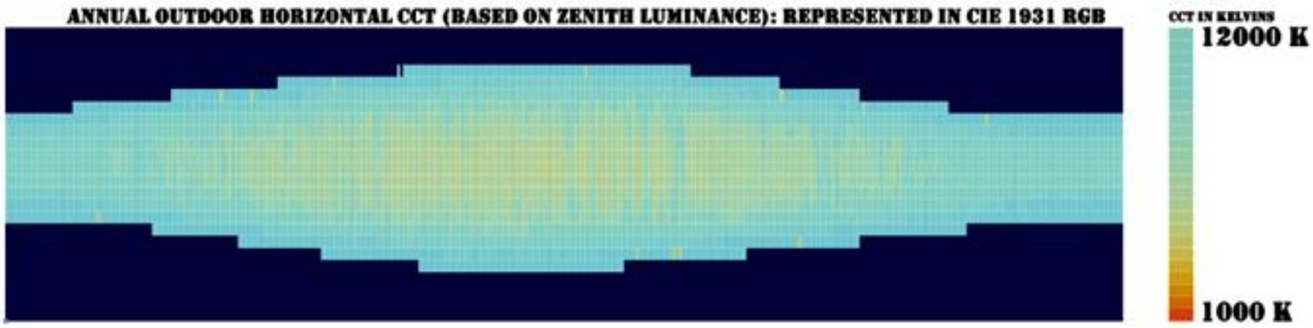
Outputs:
AllGeo: Geometry, for CustomPreview component.
AllMat: Material, for CustomPreview component.
MCCT_Zen: list of annual CCTs (similar to aCCThem component)

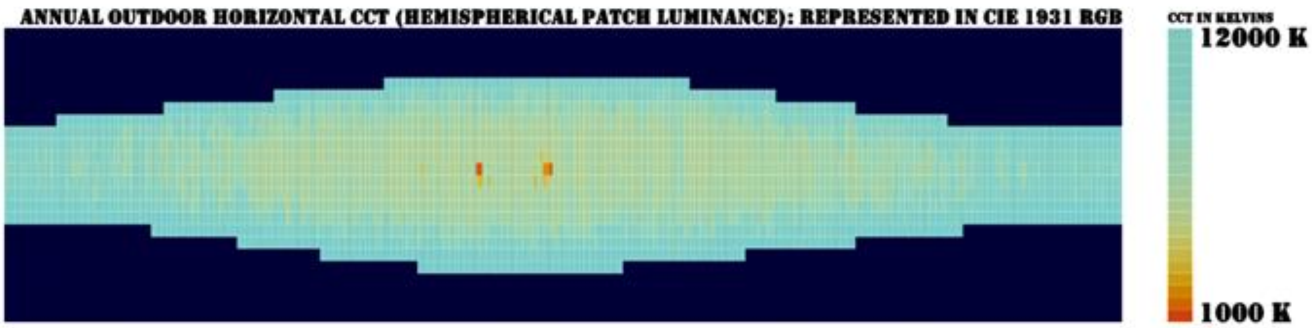
Connections (output):
Custom Preview

Presented below, are the Rhino-side visualizations of zenith-based (via visCCTzen) and hemispherical (via visCCThem) annual CCT heatmaps, for the Brussels AOWL file. These visualisations are for the horizontal plane, for unobstructed external zenith-facing hemispherical views. The heatmaps presented below are for the two color palette options: Blackbody radiation RGB and CIE 1931 RGB colors, respectively.









2.3 Visualise Occupancy Hours

2.3. aOccupFile

Component Name: aOccupFile [Code]

Component Description:

This component takes the list of annual occupied hours, and tabulates into a CSV file for further plotting and visualisation.

Inputs:
folder: location of the active folder for saving simulation data
Occup_H0: (if displaying a pre-cached simulation) list of annual hours in 1s (occupied) and 0s (unoccupied)
Occup_H: (if running live simulation) list of annual hours in 1s (occupied) and 0s (unoccupied)
Live: a boolean toggle. False = Precache. True = Live

Outputs:
OccupH_f: The location of the CSV file with annual occupancy data. This connects to aOccupPlot for plotting in 24x365 annual heatmap

Connections (Input):
PreCache, CacheDltMtxV, DltMtxV

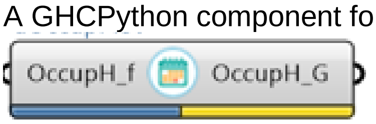
Connections (output):
aOccupPlot

2.3. aOccupPlot

Component Name: aOccupPlot [Code]

Component Description:

A GHCPython component for plotting in 24x365 annual heatmap from data generated by aOccupFile component



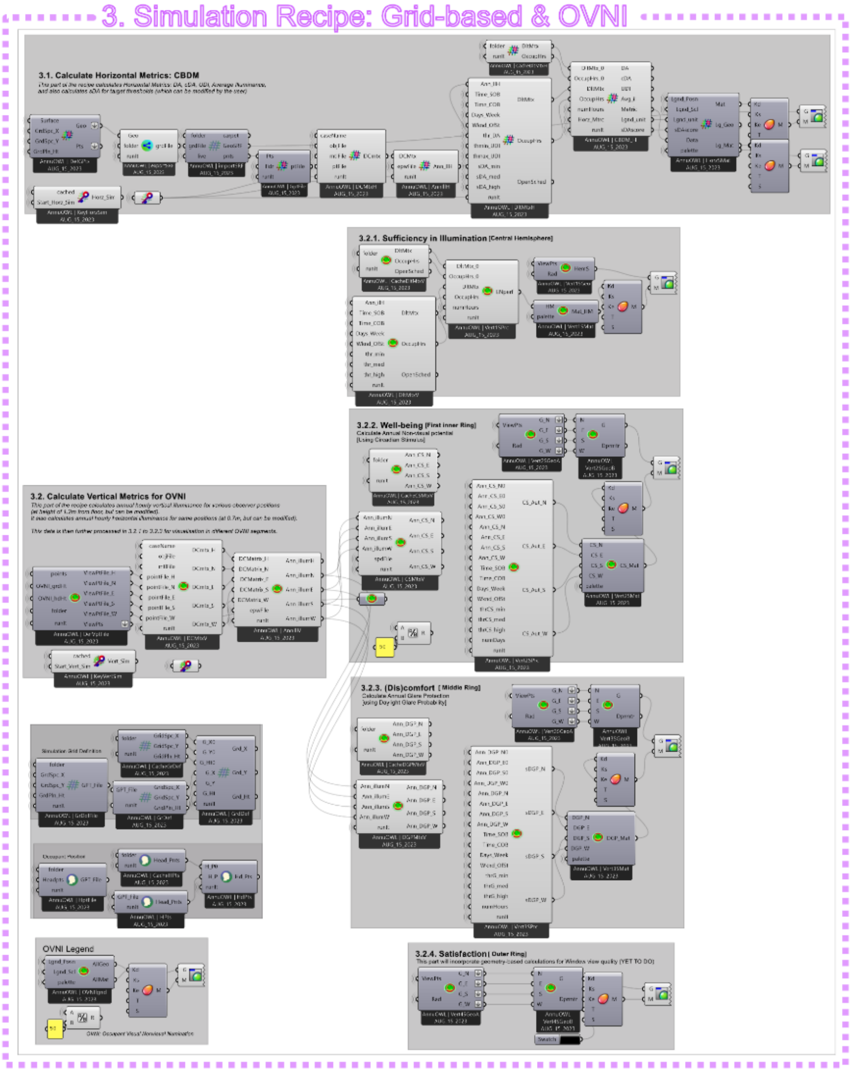
Inputs:
OccupH_f: The location of the CSV file with annual occupancy data.

Outputs:
OccupH_f: Location of the plot file.

Connections (Input):
aOccupFile

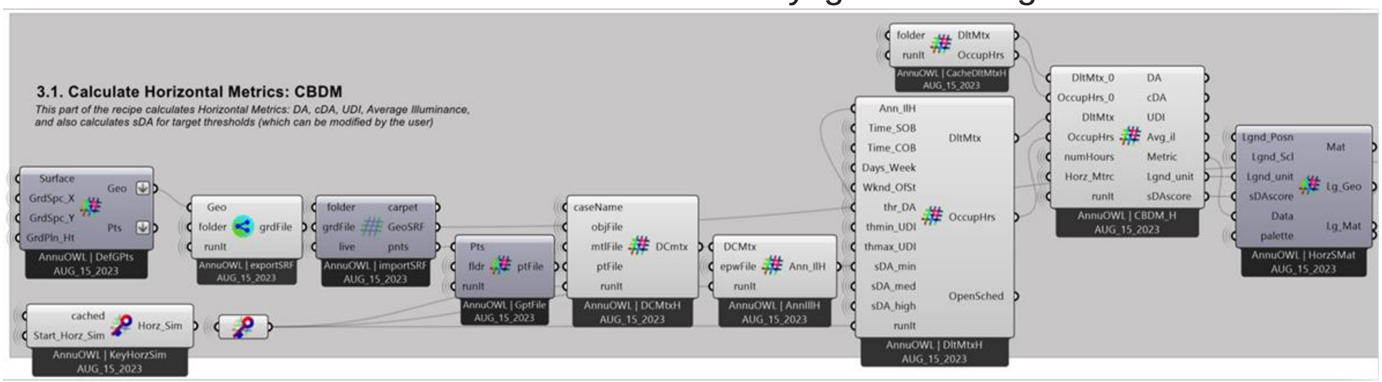
Connections (output):
LBImageViewer

3. Simulation Pipeline: Grid-based and OVNI



This is the pipeline with 43 components, nested under 3 subsections: for Horizontal grid-based evaluations, for OVNI-based evaluations, and supporting components. The components in this section form the main engine behind the AnnuOWL evaluations. The horizontal and vertical illuminance calculations are supported by Radiance 2-Phase Daylight Coefficient method for grid-based evaluations. This section does not need user-modifications.

3.1. Calculate Horizontal Metrics: Climate Based Daylight Modeling



With 11 components, this subsection evaluates the grid-based CBDMs, as well as the consolidated SDA compliance score. These evaluations are influenced by user inputs, such as the set thresholds for DA and UDI limits, the grid-point definition (x-spacing, y-spacing, grid height), and the occupancy schedule – which are taken from the Interaction section. The 11 components in this subsection, their descriptions, their icons, the input and output nodes, as well as the key associations with other components in the workflow is presented as follows.

3.1. DefGPts

Component Name: DefGPts [Code]

Component Description:

This component divides the defined floor/grid Surface into points and sub-patches, depending upon the X and Y grid spacing. The Gridplane height is also to be defined.

Inputs:
Surface: Surface for horizontal grid-based simulation. Generally the floor
GrdSpc_X: X grid spacing in meters (eg: 0.5)
GrdSpc_Y: Y grid spacing in meters (eg: 0.5)
GrdPln_Ht: Height of gridplane above floor level, in meters (eg: 0.7)

Outputs:
Geo: Geometry of sub-patches
Pts: List of points for further processing

Connections (Input):
GrdDef

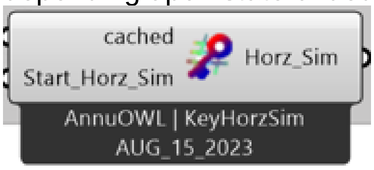
Connections (output):
ExportSRF

3.1. KeyHorzSim

Component Name: KeyHorzSim [Code]

Component Description:

A logical component that feeds into the runIt of Horizontal Metrics components, taking inputs both from dedicated toggle button ("Start Horizontal Simulation?"), but also from PreCache component's Toggle. If Precache is True, this results into 'False' output irrespective of dedicated toggle key. If PreCache is False, it results into True or False depending upon state of dedicated Toggle key.



Inputs:
cached: The toggle key of PreCache. This is the dominant key input.
Start_Horz_Sim: The dedicated toggle key. This is the secondary key input.

Outputs:
Horz_Sim: This connects to the runIt node of various horizontal simulation component(s).

3.1. exportSRF

Component Name: exportSRF [Code]

Component Description:

This component saves the divided floor/grid Surface as a **.3DM* file, to be imported for live and cached simulations later.

Inputs:
Geo: Geometry of sub-patches imported from DefGpts.
folder: Location of folder where live or cached files are stored.
runIt: A boolean switch for this component.

Outputs:
grdFile: Location of the 3DM file with divided surface geometry, for further Radiance processing and Rhino visualisation.

Connections (Input):
DefGpts

Connections (output):
importSRF

3.1. importSRF

Component Name: importSRF [Code]

Component Description:

This component imports the 3DM file with divided floor/grid Surface as sub-patches, for further processing. The subpatches are used for Rhino visualisation, and their central points are used for generating points for Radiance grid-based simulations.

Inputs:
folder: Location of folder where live or cached files are stored.
grdFile: Location of the 3DM file with divided surface geometry, for further Radiance processing and Rhino visualisation.
live: A boolean toggle for this component. True = Live simulation, False = Precached data from folder.

Outputs:
carpet: Location of the .3DM file
GeoSRF: List of individual sub-patches for Rhino grid-based visualisation.

pnts: List of points for Radiance grid-based simulations
Connections (Input):
ExportSRF, PreCache, KeyCacheOBJ
Connections (output):
Custom Preview, GptFile

3.1. GptFile

Component Name: GptFile [Code]
Component Description:
This component takes the list of points, and saves it as a point-file in the defined folder. This point-file is used by Radiance for further processing.

Inputs:
Pts: List of gridpoints
fldr: Folder to save the pointfile in.
Outputs:
ptFile: Location of the point-file
Connections (Input):
importSRF, KeyHorzSim
Connections (output):
DCMtxH

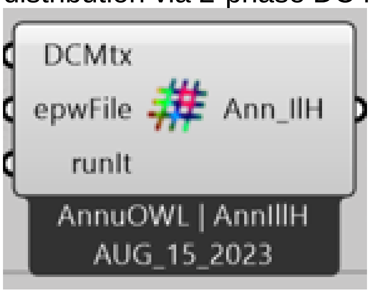
3.1. DCMtxH

Component Name: DCMtxH [Code]
Component Description:
This component reads OBJfiles with material files (while also generating an interim skyfile), and renders a Daylight Coefficient Matrix as precursor to Radiance grid-based evaluation. **This component uses "-ab 5 -ad 10000" for generating the Daylight coefficient matrix.**

Inputs:
caseName: Name for the specific case.
objFile: Takes location of .obj files. Connect to the output node of ExportOBJ component.
mtlFile: Takes location of Material file (in .RAD format) from GenMat component.
pointFile: Takes a point-file with a list of gridpoints.
runIt: A boolean toggle to run this component.
Outputs:
DCmtx: Location of the saved .mtx file containing pointwise Daylight Coefficients.
Connections (Input):
ExportOBJ, GenMat, GptFile, KeyHorzSim
Connections (output):
AnnIllH

3.1. AnnIllH

Component Name: AnnIllH [Code]
Component Description:
This component takes the DC matrix (from SCALE_DCMtxH) along with EPW file, to generate annual illuminance distribution via 2-phase DC method.



Inputs:
DCMatrix: Link to .mtx file output of SCALE_DCMtxH
epwFile: Connect to the EPW file of the location
runIt: A boolean toggle to run this component.
Outputs:
Ann_illum: Location of the annualillum.csv file generated, which contains annual illuminance values.
Connections (Input):
DCMtxH, KeyHorzSim
Connections (output):
DltMtxH

3.1. CacheDltMtxH

Component Name: CacheDltMtxH [Code]
Component Description:

A pre-cached alternative of the DltMtxH component.

Inputs:
folder: Location of pre-simulated folder.
runIt: A toggle switch

Outputs:
DltMtx: Location of a pre-calculated CSV file with various daylight metrices (DA/cDA/UDI/Average-Illuminance) for each grid-point, and Occupancy hours of the year.
OccupHrs: Occupied hours of the year, calculated based on user inputs (SOB, COB, Days of Week, etc.)

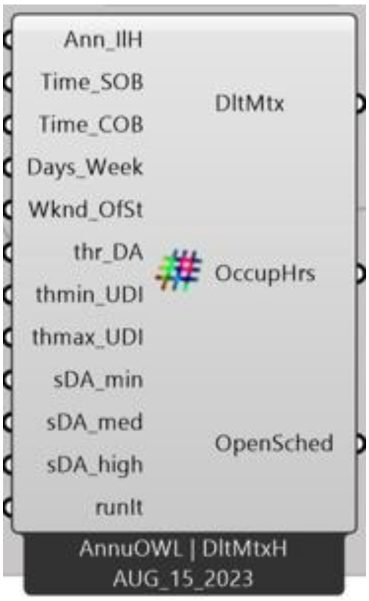
Connections (Input):
PreCache, KeyCacheOBJ

Connections (output):
CBDM_H

3.1. DltMtxH

Component Name: DltMtxH [Code]

Component Description:
[If running Live Simulation] This component takes the horizontal illuminance data for each point for each hour of the year (from AnnIllH), along with occupancy hours parameters (SoB, CoB, etc), as well as user defined thresholds for DA and UDI, and evaluates the Daylight Metrics (DA, cDA, UDI, Average Illuminance) for each point. This evaluates Daylight Autonomy, Continuous Daylight Autonomy, Useful Daylight Illuminance, and Average Illuminance over the year. Refer to https://patternguide.advancedbuildings.net/using-this-guide/analysis-methods/daylight-autonomy.html for more on the metrics.



Inputs:
Ann_illH: The annual grid CSV file generated by AnnIllH component.
Time_SOB: Start of business each day (eg: 9)
Time_COB: Close of business each day (eg: 18)
Days_Week: Working days each week (eg: 5)
Wknd_OfSt: (Weekend offset) definition of working weeks. If the first day of the year is Sunday, this parameter is 0. If any other day, this parameter is the difference between that day and Sunday (eg: 3, if 1 Jan was Wednesday).
thr_DA: Minimum threshold for Daylight Autonomy (and cDA). Default is 300 (lux) but user can choose a different number for minimum.
thmin_UDI: Lower limit of Useful Daylight Illuminance. Default is 100 (lux) but user can choose a different lower threshold.
thmax_UDI: Upper limit of Useful Daylight Illuminance. Default is 2000 (lux) but user can choose a different upper threshold.
sDA_min: Threshold for Daylight Autonomy - Minimum (eg: by EN17037 Target/Minimum). Default is 300 (lux) but user can choose a different number for minimum threshold.
sDA_med: Threshold for Daylight Autonomy - Medium (eg: by EN17037 Target/Medium). Default is 500 (lux) but user can choose a different number for medium threshold.
sDA_high: Threshold for Daylight Autonomy - High (eg: by EN17037 Target/High). Default is 750 (lux) but user can choose a different number for high threshold.
runIt: A boolean Toggle for running this component.

Outputs:
DltMtx: Link to the CSV file containing DA, cDA, UDI and other parameters.
OccupHrs: Occupied hours of the year, depending upon user defined parameters (SOB, COB, etc.)
OpenSched: Occupancy schedule through the 8760 hours, where 1 = occupied and 0 = unoccupied

Connections (Input):
AnnIllH, KeyCacheOBJ

Connections (output):
CBDM_H

3.1. CBDM_H

Component Name: CBDM_H [Code]

Component Description:
This component isolates Daylight Metrics from the computed CSV file into DA, cDA and UDI data. Takes the CSV from DltMtxH component.

Inputs:
DltMtx_0: (if pre-cached) Link to CSV file from pre-cached folder containing DA, cDA and UDI data, generated by DltMtxH component.
OccupHrs_0: (if pre-cached) Number of occupied hours of the pre-simulated data.

DltMtx: (if live simulation) Link to CSV file containing DA, cDA and UDI data, generated by DltMtxH component.
OccupHrs: (if live simulation) Number of occupied hours calculated from user defined parameters (SOB, COB, Days of Week).
numHours: Percentage annual occupied hours as threshold for compliance (eg: 50 for EN17037 Target, 95 for EN17037 Minimum
Horz_Mtrc: Slider input defining which metric to visualise: 0 = DA, 1 = cDA, 2 = UDI, 3 = Average Illuminance
runIt: a boolean switch for running this component. Set to TRUE.

Outputs:
DA: Daylight Autonomy.
cDA: Continuous Daylight Autonomy
UDI: Useful Daylight Illuminance.
Avg_il: Average illuminance.
Metric: selected metric (DA/cDA/UDI/Avg.Il) based on slider input. This connects to Horizontal Simulation Material component (HorzSMat).
Lgnd_unit: Depending upon the slider input, this generates the text for Legend.
sDAscore: sDA performance of the design for Minimum, Medium and High compliance.

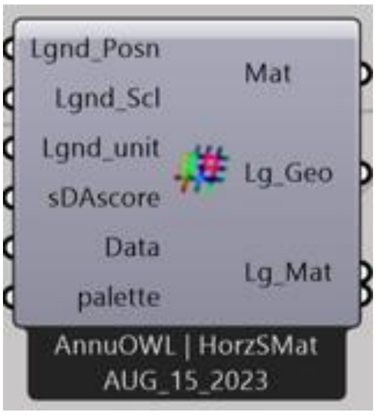
Connections (Input):
CacheDltMtxH, DltMtxH

Connections (output):
HorzSMat

3.1. HorzSMat

Component Name: HorzSMat [Code]

Component Description:
This component takes in the data to be visualised over the grid (Metric from CBDM_H) and converts that to rhino material for visualisation.



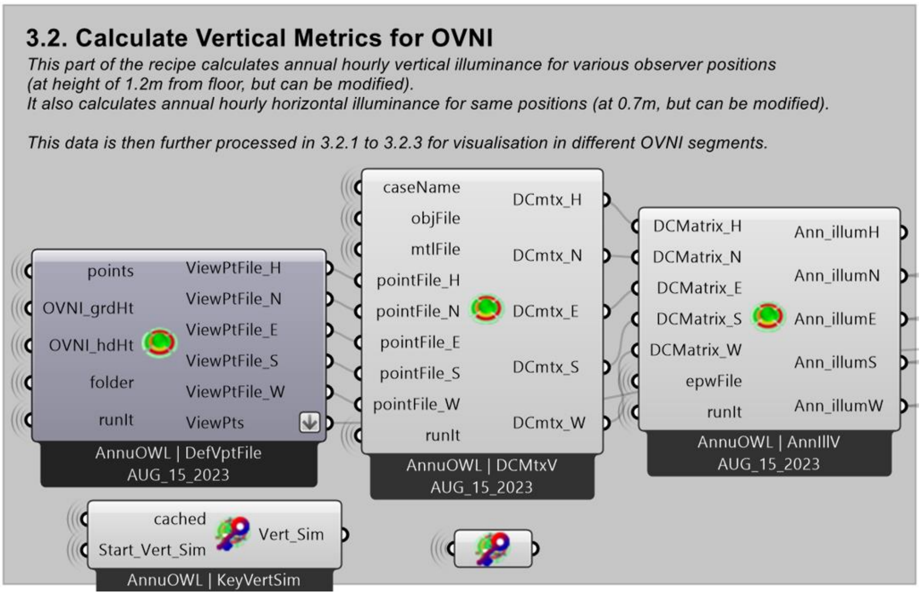
Inputs:
Lgnd_Posn: Legend Position as a point on Rhino.
Lgnd_Scl: Slider to resize the legend.
Lgnd_unit: CBDM-specific textual description. Takes data from CBDM_H component.
sDAscore: sDA performance of the design for Minimum, Medium and High performance. Takes input from CBDM_H component.
Data: The grid-based data to be visualised.
palette: (0-5) a slider for different visualisaiton palettes. Default is 0.

Outputs:
Mat: Material for each geometry subpatch for visualisation through Custom Preview component.
Lg_Geo: Geometry of the Legend.
Lg_Mat: Material/colors of the legend. connects to Custom Preview along with Geometry

Connections (Input):
CBDM_H

Connections (output):
Custom Preview

3.2. Calculate Vertical Metrics for OVNI



This subsection is the pre-cursor for its 4 sub-subsections, each of which evaluate various constituents of the OVNI diagrams. The annual illuminance for each occupant position (at horizontal plane at the table height for the OVNI hemisphere, and on vertical plane facing four orientations at the eye-height for the OVNI rings) are evaluated as CSV files in this sub section.

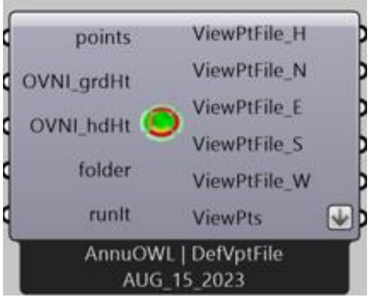
This subsection comprises 4 components, and its sub-subsections comprise 17 more components. Their descriptions, their icons, the input and output nodes, as well as the key associations with other components in the workflow is presented as follows:

3.2. DefVptFile

Component Name: DefVptFile [Code]

Component Description:

This component converts defined occupant positions as points into point-files in four principal orientations, for use in radiance grid-based simulations.



Inputs:

points: Takes a list of points, as x,y,z coordinates.

OVNI_grdHt: Height above floor level for evaluating sDA (typically table height, around 0.7m above floor level)

OVNI_hdHt: Vertical distance between an observer's eyes and the floor level, for evaluating visual/non-visual metrics (typically around 1.2m above floor level)

folder: Takes location of the folder, for saving the point files.

Outputs:

ViewPtFile_H: pointfile for zenith-facing horizontal position.

ViewPtFile_N: pointfile for north-facing vertical view.

ViewPtFile_E: pointfile for east-facing vertical view.

ViewPtFile_S: pointfile for south-facing vertical view.

ViewPtFile_W: pointfile for west-facing vertical view.

ViewPts: list of points as x,y,z coordinates.

Connections (Input):

HdPts, KeyVertSim

Connections (output):

DCMtxV, GptPointer, Vert1SGeo, Vert2SGeoA, Vert3SGeoA, Vert4SGeoA

3.2. KeyVertSim

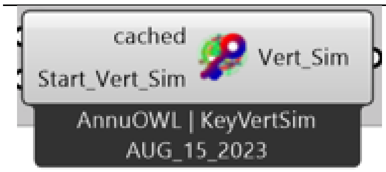
Component Name: KeyVertSim [Code]

Component Description:

A logical component that feeds into the runIt of Vertical Metrics components, taking inputs both from dedicated toggle button ("Start Vertical Simulation?"), but also from PreCache component's Toggle. If Precache is True, this results into 'False' output irrespective of dedicated toggle key. If PreCache is False, it results into True or False depending upon state of dedicated Toggle key.

Inputs:

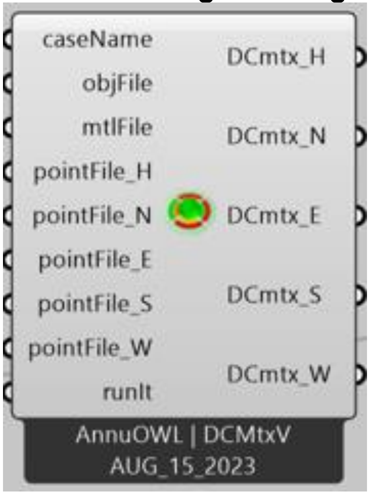
cached: The toggle key of PreCache. This is the dominant key input.



Start_Vert_Sim: The dedicated toggle key. This is the secondary key input.
Outputs:
Vert_Sim: This connects to the runIt node of various horizontal simulation component(s).
Connections (output):
AnnIllV, DCMtxV, CSMtxV, Vert2SPrc, DGPMtxV, Vert3SPrc, DefVptfile, GptCSfile, GptDGPfile, Vert1SPrc, aOccupFile, DltMtxV, CacheDltMtxV, CacheCSMtxV, CacheDGPMtxV, HptFile, HPts, CacheHPts, HdPts

3.2. DCMtxV

Component Name: DCMtxV [Code]
Component Description:
(Live Simulation only) This component reads OBJfiles with material files (while also generating a Radiance skyfile), and renders Daylight Coefficient Matrices for 4 principal orientations and for the zenith facing vector for each occupant position, as precursor to Radiance grid-based illuminance simulations. **This component uses "-ab 5 -ad 10000" for generating the 5 Daylight coefficient matrices.**



Inputs:
caseName: Name for the specific case.
objFile: Takes location of .obj files. Connect to the output node of ExportOBJ component.
mtlFile: Takes location of Material file (in .RAD format) from GenMat component.
pointFile_H: pointfile for zenith-facing horizontal position.
pointFile_N: pointfile for north-facing vertical view.
pointFile_E: pointfile for east-facing vertical view.
pointFile_S: pointfile for south-facing vertical view.
pointFile_W: pointfile for west-facing vertical view.
runIt: A boolean toggle to run this component. True = Live, False = Precache
Outputs:
DCmtx_H: Location of the saved .mtx file containing pointwise Daylight Coefficients for zenith-facing horizontal position.
DCmtx_N: Location of the saved .mtx file containing pointwise Daylight Coefficients for north-facing vertical view.
DCmtx_E: Location of the saved .mtx file containing pointwise Daylight Coefficients for east-facing vertical view.
DCmtx_S: Location of the saved .mtx file containing pointwise Daylight Coefficients for south-facing vertical view.
DCmtx_W: Location of the saved .mtx file containing pointwise Daylight Coefficients for west-facing vertical view.
Connections (Input):
ExportOBJ, GenMat, DefVptFile, KeyVertSim
Connections (output):
AnnIllV

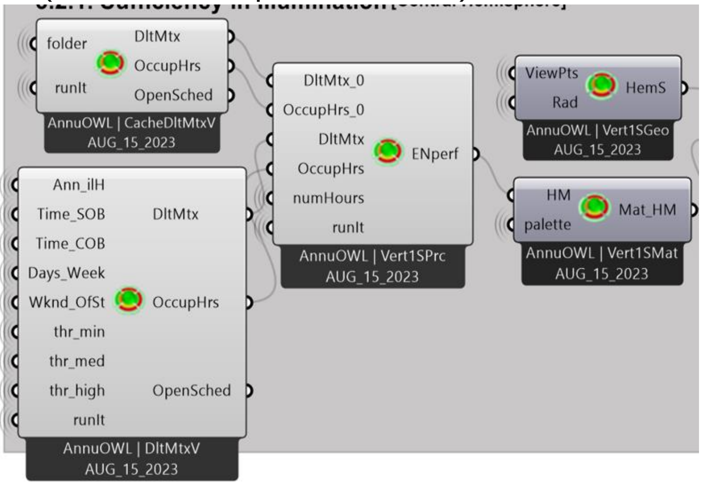
3.2. AnnIllV

Component Name: AnnIllV [Code]
Component Description:
(Live Simulation only) This component takes the DC matrix (from SCALE_DCMatrix) along with EPW file, to generate annual illuminance distribution via 2-phase DC method. This is done for the horizontal plane, as well as for vertical plane for 4 orientations.

Inputs:
DCMatrix_H: Location of the saved .mtx file containing pointwise Daylight Coefficients for zenith-facing horizontal position.
DCMatrix_N: Location of the saved .mtx file containing pointwise Daylight Coefficients for north-facing vertical view.
DCMatrix_E: Location of the saved .mtx file containing pointwise Daylight Coefficients for east-facing vertical view.
DCMatrix_S: Location of the saved .mtx file containing pointwise Daylight Coefficients for south-facing vertical view.
DCMatrix_W: Location of the saved .mtx file containing pointwise Daylight Coefficients for west-facing vertical view.
epwFile: Connect to the EPW file of the location.
runIt: A boolean toggle to run this component. True = Live, False = Precache.
Outputs:
Ann_illumH: Location of the csv file generated, which contains annual illuminance values for zenith-facing horizontal position.

Ann_illumN: Location of the csv file generated, which contains annual illuminance values for north-facing vertical view.
Ann_illumE: Location of the csv file generated, which contains annual illuminance values for east-facing vertical view.
Ann_illumS: Location of the csv file generated, which contains annual illuminance values for south-facing vertical view.
Ann_illumW: Location of the csv file generated, which contains annual illuminance values for west-facing vertical view.
Connections (Input):
DCMtxV, KeyVertSim
Connections (output):
DltMtxV, CSMtxV, DGPMtxV

3.2.1. Sufficiency in Illumination (Central Hemisphere of OVNI)



The calculations for visualization in OVNI hemisphere: **Daylight sufficiency performance** -- based on Daylight Autonomy compliance thresholds calculated for annual occupied hours, are performed by the 5 components in this sub-subsection.

3.2.1. CacheDltMtxV

Component Name: CacheDltMtxV [Code]
Component Description:
A pre-cached alternative of the DltMtxV component.

Inputs:
folder: Location of pre-simulated folder.
runIt: A toggle switch
Outputs:
DltMtx: Location of a pre-calculated CSV file with various daylight metrices (DA/cDA/UDI/Average-Illuminance) and compliance metrics (EN17037 Minimum, Medium, High for each occupant-position on horizonal plane, and Occupancy hours of the year.
OccupHrs: Occupied hours of the year, calculated based on user inputs (SOB, COB, Days of Week, etc.).
OpenSched: Occupancy state through the 8760 annual hours (0=Unoccupied, 1=Occupied)
Connections (Input):
PreCache, KeyVertSim
Connections (output):
Vert1SPrc, aOccupFile

3.2.1. DltMtxV

Component Name: DltMtxV [Code]
Component Description:
[If running Live simulation] This component takes the horizontal illuminance data for each occupant's position for each hour of the year (from AnnIllV component), along with occupancy hours parameters (SoB, CoB, etc), as well as user defined thresholds for minimum, medium and high targets (such as for EN17037), and evaluates the spatial daylight autonomy (Target - Min/Med/High or custom) for each occupant's position.
Inputs:
Ann_illH: The CSV file containing hourly annual illuminance, generated for each occpant's position by the AnnIllV component.
Time_SOB: Start of business each day (eg: 9)

Time_COB: Close of business each day (eg: 18)
Days_Week: Working days each week (eg: 5)
Wknd_OfSt: (Weekend offset) definition of working weeks. If the first day of the year is Sunday, this parameter is 0. If any other day, this parameter is the difference between that day and Sunday (eg: 3, if 1 Jan was Wednesday).
thr_min: Threshold for Daylight Autonomy - Minimum (eg: by EN17037 Target/Minimum). Default is 300 (lux) but user can choose a different number for minimum threshold.
thr_med: Threshold for Daylight Autonomy - Medium (eg: by EN17037 Target/Medium). Default is 500 (lux) but user can choose a different number for medium threshold.
thr_high: Threshold for Daylight Autonomy - High (eg: by EN17037 Target/High). Default is 750 (lux) but user can choose a different number for high threshold.
runIt: A boolean Toggle for running this component. FALSE = cached data (this component does not run)

Outputs:
DltMtx: Link to the CSV file containing calculated sDA parameters for the defined minimum, medium, high thresholds.
OccupHrs: Occupied hours of the year, depending upon user defined parameters (SOB, COB, etc.)
OpenSched: Occupancy schedule through the 8760 hours, where 1 = occupied and 0 = unoccupied

Connections (Input):
AnnIllV, KeyVertSim

Connections (output):
Vert1SPrc, aOccupFile

3.2.1. Vert1SPrc

Component Name: Vert1SPrc [Code]

Component Description:
This component isolates Daylight Metrics from the computed CSV file into various thresholds, and processes the data to evaluate how each occupant position performs, as High, Medium or Low Performance, or as Non-compliant. If any gridpoint, for defined percent of occupied hours does not qualify for High, then it checks for Medium, and firther for Low, and thus assigns performance evaluation for each occupant's position.

Inputs:
DltMtx_0: (if pre-cached) Link to CSV file from pre-cached folder containing threshold binned data, generated by DltMtxV component.
OccupHrs_0: (if pre-cached) Number of occupied hours of the pre-simulated data.
DltMtx: (if live simulation) Link to CSV file containing threshold-binned data, generated by DltMtxV component.
OccupHrs: (if live simulation) Number of occupied hours calculated from user defined parameters (SOB, COB, Days of Week).
numHours: Percentage annual occupied hours as threshold for compliance (eg: 50 for EN17037 Target, 95 for EN17037 Minimum
runIt: a boolean switch for running this component. Set to TRUE.

Outputs:
ENperf: Performance of each occupant's position. 0=Non compliant, 1=Minimum, 2=Medium, 3=High

Connections (Input):
CacheDltMtxV, DltMtxV, KeyVertSim

Connections (output):
Vert1SMat

The logical flowchart for performance evaluation, and assigning compliance for visualization in the OVNI hemisphere is presented later in the document.

3.2.1. Vert1SGeo

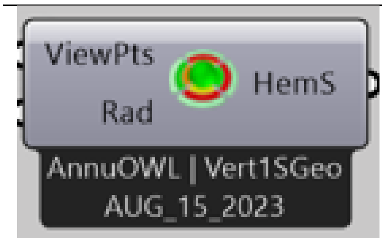
Component Name: Vert1SGeo [Code]

Component Description:
This component generates the OVNI inner hemisphere's Geometry. Output connects to Custom Preview Geometry.

Inputs:
ViewPts: Points representing occupants' head positions.
Rad: Radius of OVNI inner hemisphere. Can be rescaled.

Outputs:
HemS: Hemispherical Geometry, connects to Custom Preview

Connections (Input):
DefVptFile



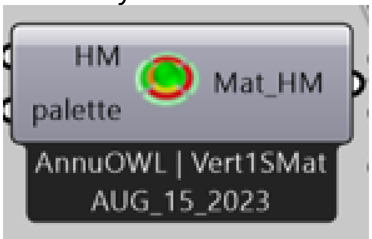
Connections (output):
Custom Preview

3.2.1. Vert1SMat

Component Name: Vert1SMat [Code]

Component Description:

This component generates the OVNI inner hemisphere's Materials/Colors. Output connects to Custom Preview Geometry.



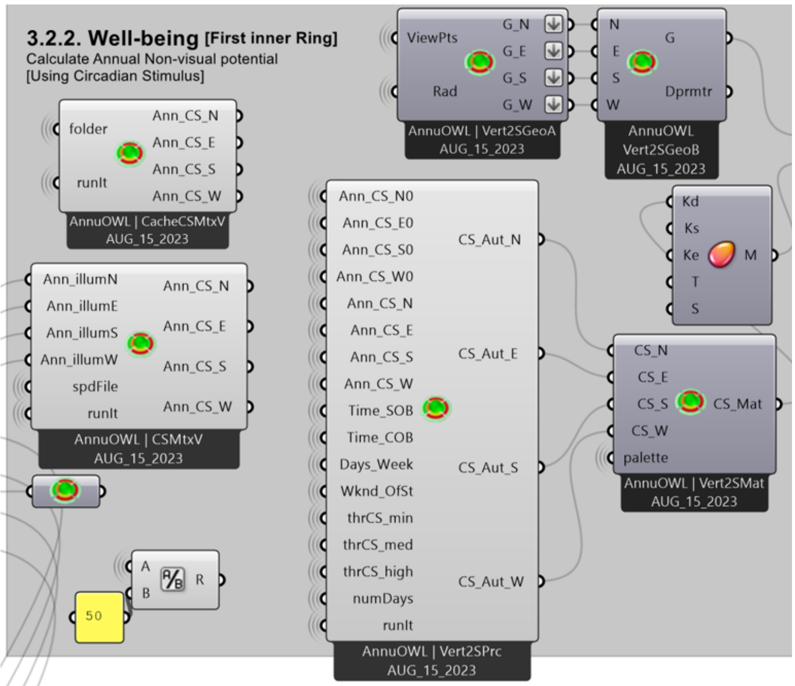
Inputs:
HM: (Horizontal metric) Data to be displayed for each gridpoint/occupant position.
palette: Coloring Palette for visualising data, with 5 options.

Outputs:
Mat_HM: Materials for Hemispherical Geometry, connects to Custom Preview

Connections (Input):
Vert1SPrc

Connections (output):
Custom Preview

3.2.2. Well-being



The calculations for visualization in OVNI first inner ring: **Circadian/Non-visual potential** using evaluation of Circadian Stimulus for annual occupied hours – facing the four principal orientations, are performed by following 6 components:

3.2.2. CacheCSMtxV

Component Name: CacheCSMtxV [Code]

Component Description:

A pre-cached alternative of the CSMtxV component.

Inputs:
folder: Location of pre-simulated folder.
runIt: A toggle switch. False = Precache, True = Live

Outputs:
Ann_CS_N: A pre-simulated csv file containing Annual hourly CS (vertical plane) for each occupant position, facing North.
Ann_CS_E: A pre-simulated csv file containing Annual hourly CS (vertical plane) for each occupant position, facing East.

Ann_CS_S: A pre-simulated csv file containing Annual hourly CS (vertical plane) for each occupant position, facing South.
Ann_CS_W: A pre-simulated csv file containing Annual hourly CS (vertical plane) for each occupant position, facing West.
Connections (Input):
PreCache, KeyVertSim
Connections (output):
Vert2SPrc, GptCSFile

3.2.2. CSMtxV

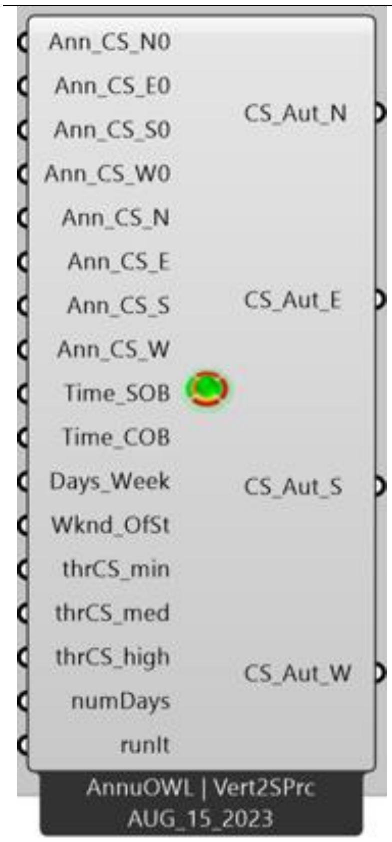
Component Name: CSMtxV [Code]
Component Description:
(if running live simulation) This component takes the annual sky SPD data (.aowl file) which includes pre-calculated CIE_z value for each annual hour in addition to spectral data -- along with vertical illuminance data at each occupant position at eye level in 4 orientations, and uses Truong's approximation to evaluate CS for each point over the year. Truong's approximation makes the code at least 900X fast (tested to be 1826sec vs 2.7sec for a specific case, for regular calculation as in OWL1 approach and with Truong's approximation, respectively). For more info on this approximation, refer to DOI: https://doi.org/10.1177/1477153519887423 and https://doi.org/10.1177/14771535211044664

Inputs:
Ann_illumN: Location of the csv file with calculated annual hourly illuminance data for each occupant's position, on eye level at the vertical plane, facing North.
Ann_illumE: Location of the csv file with calculated annual hourly illuminance data for each occupant's position, on eye level at the vertical plane, facing East.
Ann_illumS: Location of the csv file with calculated annual hourly illuminance data for each occupant's position, on eye level at the vertical plane, facing South.
Ann_illumW: Location of the csv file with calculated annual hourly illuminance data for each occupant's position, on eye level at the vertical plane, facing West.
spdFile: Location of the annual SPD data file (.aowl format) which also has CIE x y z, spectral data and CCT values, with CIEz being the relevant data for each annual hour.
runIt: A boolean toggle for this component. True = Live, False = Precache.
Outputs:
Ann_CS_N: A csv file containing Annual hourly CS (vertical plane) for each occupant position, facing North.
Ann_CS_E: A csv file containing Annual hourly CS (vertical plane) for each occupant position, facing East.
Ann_CS_S: A csv file containing Annual hourly CS (vertical plane) for each occupant position, facing South.
Ann_CS_W: A csv file containing Annual hourly CS (vertical plane) for each occupant position, facing West.
Connections (Input):
AnnIllV, KeyVertSim
Connections (output):
Vert2SPrc, GptCSFile

3.2.2. Vert2SPrc

Component Name: Vert2SPrc [Code]
Component Description:
This component takes in the Annual CS for each hour (by reading CSV files for live or pre-cached simulations) for 4 orientations, takes in occupancy hours (SOB, COB, Days of week) and bins the circadian potential performance into High, Medium, Minimum and non compliant based on the defined thresholds.
For each occupied day, non-visual performance for each bin (High/Med/Min/NC) is defined as the percentage of occupied days in a year when CS exceeds defined threshold for at least 1 occupied hour in the morning (between 0800-1200 inclusive, depending also on SOB/COB).
Further, at the occupant position level, if any point, for defined percent of occupied hours does not qualify for High, then it checks for Medium, and further for Low, and thus assigns performance evaluation for each occupant's position.

Inputs:
Ann_CS_N0: (if precached) Link to CSV file from pre-cached folder containing Annual hourly CS, facing north.
Ann_CS_E0: (if precached) Link to CSV file from pre-cached folder containing Annual hourly CS, facing east.



Ann_CS_S0: (if precached) Link to CSV file from pre-cached folder containing Annual hourly CS, facing south.
Ann_CS_W0: (if precached) Link to CSV file from pre-cached folder containing Annual hourly CS, facing west.
Ann_CS_N: (if live simulation) Link to calculated CSV file containing Annual hourly CS, facing north.
Ann_CS_E: (if live simulation) Link to calculated CSV file containing Annual hourly CS, facing east.
Ann_CS_S: (if live simulation) Link to calculated CSV file containing Annual hourly CS, facing south.
Ann_CS_W: (if live simulation) Link to calculated CSV file containing Annual hourly CS, facing west.
Time_SOB: Start of business each day (eg: 9)
Time_COB: Close of business each day (eg: 18)
Days_Week: Working days each week (eg: 5)
Wknd_OfSt: (Weekend offset) definition of working weeks. If the first day of the year is Sunday, this parameter is 0. If any other day, this parameter is the difference between that day and Sunday (eg: 3, if 1 Jan was Wednesday).
thrCS_min: Threshold for Non visual performance - Minimum. Default is set at 0.35 (CS) but user can choose a different number for minimum threshold.
thrCS_med: Threshold for Non visual performance - Medium. Default is set at 0.50 (CS) but user can choose a different number for medium threshold.
thrCS_high: Threshold for Non visual performance - High. Default is 0.65 (CS) but user can choose a different number for high threshold.
numDays: Percentage annual occupied days as threshold for compliance. Default is set at 75 (as in, minimum 75% of occupied days) but user can choose a different time threshold.
runIt: A boolean Toggle for running this component. True = Live, False = Precached

Outputs:
CS_Aut_N: The non-visual potential/performance of each occupant-position, facing North (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
CS_Aut_E: The non-visual potential/performance of each occupant-position, facing East (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
CS_Aut_S: The non-visual potential/performance of each occupant-position, facing South (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
CS_Aut_W: The non-visual potential/performance of each occupant-position, facing West (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)

Connections (Input):
CacheCSMtxV, CSMtxV, KeyVertSim
Connections (output):
Vert2SMat

The logical flowchart for performance evaluation, and assigning compliance for visualization in the OVNI inner ring, is presented later in the document.

3.2.2. Vert2SGeoA

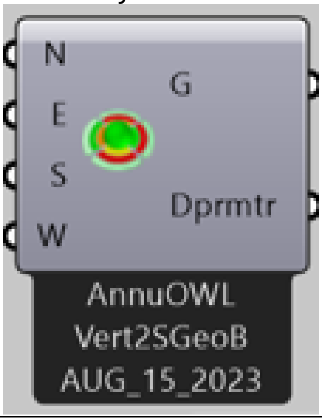
Component Name: Vert2SGeoA [Code]
Component Description:
The first of two related components, that generates the OVNI first ring (Well-being/Circadian) Geometry. The second component -- Vert2SGeoB -- merges these geometries and its Output connects to Custom Preview Geometry.

Inputs:
ViewPts: Points representing occupants' head positions.
Rad: Radius of OVNI diagrams. Can be rescaled.
Outputs:
G_N: Geometry of OVNI inner ring, facing North
G_E: Geometry of OVNI inner ring, facing East
G_S: Geometry of OVNI inner ring, facing South
G_W: Geometry of OVNI inner ring, facing West
Connections (Input):
DefVptFile
Connections (output):
Vert2SGeoB

3.2.2. Vert2SGeoB

Component Name: Vert2SGeoB [Code]
Component Description:

The second of the two related components, that takes in the geometry of the OVNI first ring (Well-being/Circadian) in four orientations from Vert2SGeoA, and merges these geometries. Its Output connects to Custom Preview Geometry.



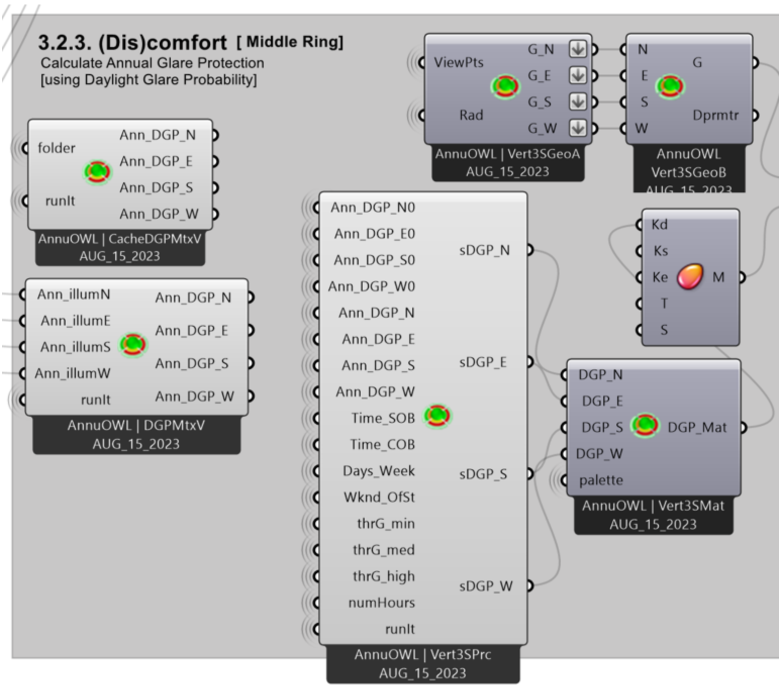
Inputs:
G_N: Geometry of OVNI inner ring, facing North
G_E: Geometry of OVNI inner ring, facing East
G_S: Geometry of OVNI inner ring, facing South
G_W: Geometry of OVNI inner ring, facing West
Outputs:
G: Merged Geometry of OVNI first ring (Well-being/Circadian), connects to Custom Preview Geometry.
Connections (Input):
Vert2SGeoA
Connections (output):
Custom Preview

3.2.2. Vert2SMat

Component Name: Vert2SMat [Code]
Component Description:
This component takes in the non-visual potential for each orientation, and calculates the color (material) for visualisation, depending upon the selected palette.

Inputs:
CS_N: The non-visual potential of each occupant-position, facing North (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
CS_E: The non-visual potential of each occupant-position, facing East (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
CS_S: The non-visual potential of each occupant-position, facing South (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
CS_W: The non-visual potential of each occupant-position, facing West (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
Outputs:
CS_Mat: Combined materials for all orientations. Connects to Custom Preview
Connections (Input):
Vert2SPrc
Connections (output):
Custom Preview

3.2.3. Discomfort



The calculations for visualization in OVNI middle ring: **Protection from discomfort glare** using evaluations of Daylight Glare Probability (DGP) for annual occupied hours – facing the four principal orientations, are performed by the following 6 components:

3.2.3. CacheDGPMtxV

Component Name: CacheDGPMtxV [Code]

Component Description:

A pre-cached alternative of the DGPMtxV component.

Inputs:
folder: Location of pre-simulated folder.
runIt: A toggle switch. False = Precache, True = Live

Outputs:
Ann_DGP_N: A pre-simulated csv file containing Annual hourly DGP (vertical plane) for each occupant position, facing North.
Ann_DGP_E: A pre-simulated csv file containing Annual hourly DGP (vertical plane) for each occupant position, facing East.
Ann_DGP_S: A pre-simulated csv file containing Annual hourly DGP (vertical plane) for each occupant position, facing South.
Ann_DGP_W: A pre-simulated csv file containing Annual hourly DGP (vertical plane) for each occupant position, facing West.

Connections (Input):
PreCache, KeyVertSim

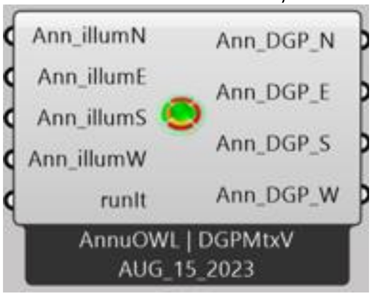
Connections (output):
Vert3SPrc, GptDGPfile

3.2.3. DGPMtxV

Component Name: DGPMtxV [Code]

Component Description:

(if running live simulation) This component takes the vertical illuminance data at each occupant position at eye level in 4 orientations, -- and uses simplified DGP approach to evaluate DGPs for each point over the year.



Inputs:
Ann_illumN: Location of the csv file with calculated annual hourly illuminance data for each occupant's position, on eye level at the vertical plane, facing North.
Ann_illumE: Location of the csv file with calculated annual hourly illuminance data for each occupant's position, on eye level at the vertical plane, facing East.
Ann_illumS: Location of the csv file with calculated annual hourly illuminance data for each occupant's position, on eye level at the vertical plane, facing South.
Ann_illumW: Location of the csv file with calculated annual hourly illuminance data for each occupant's position, on eye level at the vertical plane, facing West.
runIt: A boolean toggle for this component. True = Live, False = Precache.

Outputs:
Ann_DGP_N: A csv file containing Annual hourly sDGP (vertical plane) for each occupant position, facing North.
Ann_DGP_E: A csv file containing Annual hourly sDGP (vertical plane) for each occupant position, facing East.
Ann_DGP_S: A csv file containing Annual hourly sDGP (vertical plane) for each occupant position, facing South.
Ann_DGP_W: A csv file containing Annual hourly sDGP (vertical plane) for each occupant position, facing West.

Connections (Input):
AnnIllV, KeyVertSim

Connections (output):
Vert3SPrc, GptDGPfile

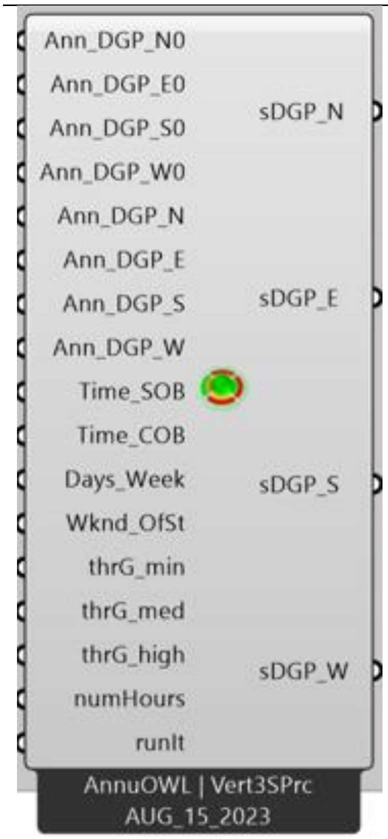
3.2.3. Vert3SPrc

Component Name: Vert3SPrc [Code]

Component Description:

This component takes in the Annual sDGP for each hour (by reading CSV files for live or pre-cached simulations) for 4 orientations, takes in occupancy hours (SOB, COB, Days of week) and bins the glare protection performance into High, Medium, Minimum and non compliant based on the defined thresholds. Further, at the occupant position level, if any positions glare protection performance for defined percent of occupied hours does not qualify for High, then it checks for Medium, and further for Low, and thus assigns performance evaluation for each occupant's position.

Inputs:



Ann_DGP_N0: (if precached) Link to CSV file from pre-cached folder containing Annual hourly DGP, facing north.
Ann_DGP_E0: (if precached) Link to CSV file from pre-cached folder containing Annual hourly DGP, facing east.
Ann_DGP_S0: (if precached) Link to CSV file from pre-cached folder containing Annual hourly DGP, facing south.
Ann_DGP_W0: (if precached) Link to CSV file from pre-cached folder containing Annual hourly DGP, facing west.
Ann_DGP_N: (if live simulation) Link to calculated CSV file containing Annual hourly DGP, facing north.
Ann_DGP_E: (if live simulation) Link to calculated CSV file containing Annual hourly DGP, facing east.
Ann_DGP_S: (if live simulation) Link to calculated CSV file containing Annual hourly DGP, facing south.
Ann_DGP_W: (if live simulation) Link to calculated CSV file containing Annual hourly DGP, facing west.
Time_SOB: Start of business each day (eg: 9)
Time_COB: Close of business each day (eg: 18)
Days_Week: Working days each week (eg: 5)
Wknd_OfSt: (Weekend offset) definition of working weeks. If the first day of the year is Sunday, this parameter is 0. If any other day, this parameter is the difference between that day and Sunday (eg: 3, if 1 Jan was Wednesday).
thrG_min: Threshold for occupied-hourly glare protection performance - Minimum. Default is set at 0.45 (DGP) based on EN17037 but user can choose a different number for minimum threshold.
thrG_med: Threshold for occupied-hourly glare protection performance - Medium. Default is set at 0.40 (DGP) based on EN17037 but user can choose a different number for medium threshold.
thrG_high: Threshold for occupied-hourly glare protection performance - High. Default is 0.35 (DGP) based on EN17037 but user can choose a different number for high threshold.
numHours: Percentage annual occupied hours as threshold for compliance. Default is set at 95 (as in, minimum 95% of occupied hours) based on EN17037, but user can choose a different time threshold.
runIt: A boolean Toggle for running this component. True = Live, False = Precached

Outputs:
sDGP_N: The glare protection performance for each occupant-position, facing North (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
sDGP_E: The glare protection performance for each occupant-position, facing East (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
sDGP_S: The glare protection performance for each occupant-position, facing South (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
sDGP_W: The glare protection performance for each occupant-position, facing West (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)

Connections (Input):
CacheDGPMtxV, DGPMtxV, KeyVertSim
Connections (output):
Vert3SMat

The logical flowchart for performance evaluation, and assigning compliance for visualization in the OVNI second ring, is presented later in the document.

3.2.3. Vert3SGeoA

Component Name: Vert3SGeoA [Code]
Component Description:
The first of two related components, that generates the OVNI middle ring (Glare Protection) Geometry. The second component -- Vert3SGeoB -- merges these geometries and its Output connects to Custom Preview Geometry.

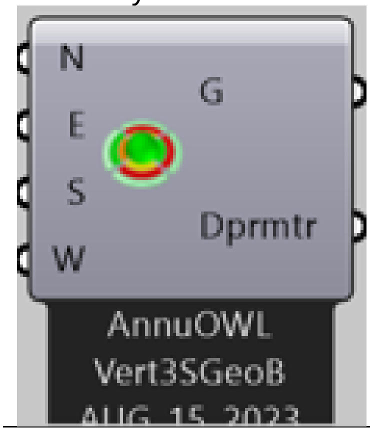
Inputs:
ViewPts: Points representing occupants' head positions.
Rad: Radius of OVNI diagrams. Can be rescaled.
Outputs:
G_N: Geometry of OVNI middle ring, facing North
G_E: Geometry of OVNI middle ring, facing East
G_S: Geometry of OVNI middle ring, facing South
G_W: Geometry of OVNI middle ring, facing West
Connections (Input):
DefVptFile

Connections (output):
Vert3SGeoB

3.2.3. Vert3SGeoB

Component Name: Vert3SGeoB [Code]

Component Description:
The second of the two related components, that takes in the geometry of the OVNI middle ring (Glare Protection) in four orientations from Vert3SGeoA, and merges these geometries. Its Output connects to Custom Preview Geometry.



Inputs:
G_N: Geometry of OVNI middle ring, facing North
G_E: Geometry of OVNI middle ring, facing East
G_S: Geometry of OVNI middle ring, facing South
G_W: Geometry of OVNI middle ring, facing West
Outputs:
G: Merged Geometry of OVNI middle ring (Glare Protection), connects to Custom Preview Geometry.
Connections (Input):
Vert3SGeoA
Connections (output):
Custom Preview

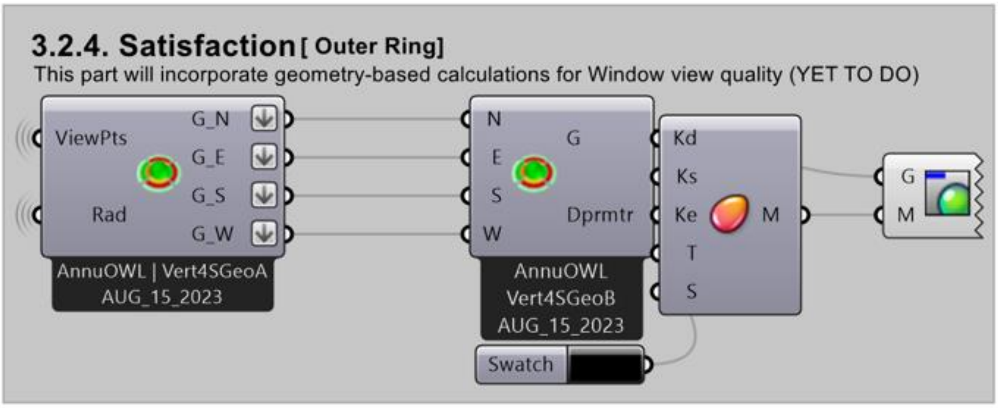
3.2.3. Vert3SMat

Component Name: Vert3SMat [Code]

Component Description:
This component takes in the glare protection performance for each orientation, and calculates the color (material) for visualisation, depending upon the selected palette.

Inputs:
DGP_N: The glare protection performance of each occupant-position, facing North (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
DGP_E: The glare protection performance of each occupant-position, facing East (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
DGP_S: The glare protection performance of each occupant-position, facing South (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
DGP_W: The glare protection performance of each occupant-position, facing West (0 = Non Compliant, 1=Minimum, 2=Medium, 3=High)
Outputs:
DGP_Mat: Combined materials for all orientations. Connects to Custom Preview
Connections (Input):
Vert3SPrc
Connections (output):
Custom Preview

3.2.4. Satisfaction (Yet to be incorporated)



These two components generate the geometry for the third and external ring, which is not yet operational and is part of forthcoming development. This ring will describe the view quality from the occupant's position.

3.2.4. Vert4SGeoA

Component Name: Vert4SGeoA [Code]

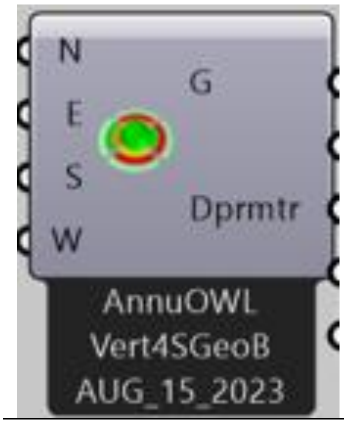
Component Description:
The first of two related components, that generates the OVNI outer ring (View Quality) Geometry. The second component -- Vert4SGeoB -- merges these geometries and its Output connects to Custom Preview Geometry.

Inputs:
ViewPts: Points representing occupants' head positions.
Rad: Radius of OVNI diagrams. Can be rescaled.
Outputs:
G_N: Geometry of OVNI outer ring, facing North
G_E: Geometry of OVNI outer ring, facing East
G_S: Geometry of OVNI outer ring, facing South
G_W: Geometry of OVNI outer ring, facing West
Connections (Input):
DefVptFile
Connections (output):
Vert4SGeoB

3.2.4. Vert4SGeoB

Component Name: Vert4SGeoB [Code]

Component Description:
The second of the two related components, that takes in the geometry of the OVNI outer ring (View Quality) in four orientations from Vert4SGeoA, and merges these geometries. Its Output connects to Custom Preview Geometry.



Inputs:
G_N: Geometry of OVNI outer ring, facing North
G_E: Geometry of OVNI outer ring, facing East
G_S: Geometry of OVNI outer ring, facing South
G_W: Geometry of OVNI outer ring, facing West
Outputs:
G: Merged Geometry of OVNI outer ring (View Quality), connects to Custom Preview Geometry.
Connections (Input):
Vert4SGeoA
Connections (output):
Custom Preview

3.3. Supporting Pipeline

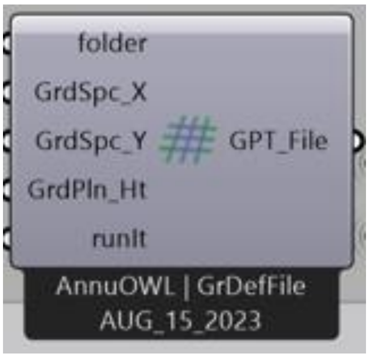
The 9 components in the following three sub-subsections support the simulation pipeline.

3.3.1. Definition of the Simulation Grid

3.3.1. GrDefFile

Component Name: GrDefFile [Code]

Component Description:
This component saves the grid definition for each simulation as a cache for further pre-cached visualisation, and also for the live simulations. The data to be cached as CSV includes Grid-X, Grid-Y and Gridplane Height.



Inputs:
folder: Location of folder for saving grid definition file as csv.
GrdSpc_X: Grid spacing in X direction.
GrdSpc_Y: Grid spacing in Y direction.
GrdPln_Ht: Height of gridplane above floor level.
Headpts: Position of observers (The points need to be defined AT FLOOR LEVEL).
runIt: a boolean switch for this component. If YES, the CSV is saved to the folder.
Outputs:
GPT_File: Location of the grid definition CSV file.
Connections (Input):
PreCache, KeyHorzSim
Connections (output):
GrDef

3.3.1. CacheGrDef

Component Name: CacheGrDef [Code]
Component Description:
Precached grid definitions, extracted from a pre-simulated folder.

Inputs:
folder: location of folder containing all simulated files.
runIt: A boolean toggle. True=Live, False=Precached.
Outputs:
GridSpc_X: Grid Spacing in the X-direction.
GridSpc_Y: Grid Spacing in the Y-direction.
GridPln_Ht: Height of grid plane above floor level
Connections (Input):
PreCache, KeyHorzSim
Connections (output):
GrdDef

3.3.1. GrDef

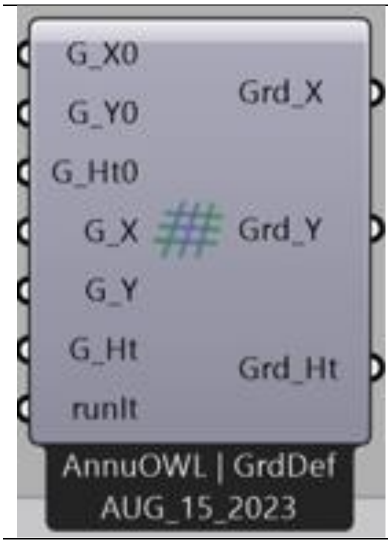
Component Name: GrDef [Code]
Component Description:
Grid definitions from live simulations, extracted from live-simulated folder.

Inputs:
GPT_File: location of Grid definition file.
runIt: A boolean toggle. True=Live, False=Precached.
Outputs:
GridSpc_X: Grid Spacing in the X-direction.
GridSpc_Y: Grid Spacing in the Y-direction.
GridPln_Ht: Height of grid plane above floor level
Connections (Input):
GrDefFile, KeyHorzSim
Connections (output):
GrdDef

3.3.1. GrdDef

Component Name: GrdDef [Code]
Component Description:
Grid definitions used for visualising grid-based data (such as CBDMs) on Rhino viewports.

Inputs:
G_X0: (if pre-cached) Grid Spacing in the X-direction.
G_Y0: (if pre-cached) Grid Spacing in the Y-direction.
G_Ht0: (if pre-cached) Height of grid plane above floor level.
G_X: (if live-sim) Grid Spacing in the X-direction.
G_Y: (if live-sim) Grid Spacing in the Y-direction.
G_Ht: (if live-sim) Height of grid plane above floor level.
runIt: A boolean toggle for this component. True = Live, False = Precache.
Outputs:
Grd_X: Grid Spacing in the X-direction.
Grd_Y: Grid Spacing in the Y-direction.
Grd_Ht: Height of grid plane above floor level.



Connections (Input):
CacheGrDef, GrDef, KeyHorzSim
Connections (output):
DefGPts

3.3.2. Occupant Position File generation

3.3.2. HptFile

Component Name: HptFile [Code]

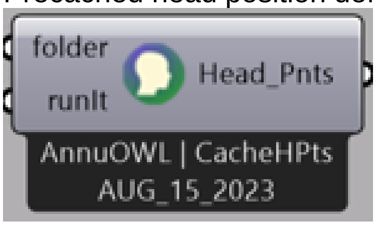
Component Description:
This component saves the head positions for occupants, for each simulation as a cache for further pre-cached visualisation. The data to be cached as CSV includes the positions of occupants on the floorplane.

Inputs:
folder: Location of folder for saving grid definition file as csv.
GrdSpc_X: Grid spacing in X direction.
GrdSpc_Y: Grid spacing in Y direction.
GrdPln_Ht: Height of gridplane above floor level.
Headpts: Position of observers (The points need to be defined AT FLOOR LEVEL).
runIt: a boolean switch for this component. If YES, the CSV is saved to the folder.
Outputs:
GPT_File: Location of the grid definition CSV file.
Connections (Input):
PreCache, KeyVertSim
Connections (output):
HPts

3.3.2. CacheHPts

Component Name: CacheHPts [Code]

Component Description:
Precached head position definitions, extracted from a pre-simulated folder.



Inputs:
folder: location of folder containing all simulated files.
runIt: A boolean toggle. True=Live, False=Precached.
Outputs:
Head_Pnts: Head positions of the occupants.
Connections (Input):
PreCache, KeyVertSim
Connections (output):
HdPts

3.3.2. HPts

Component Name: HPts [Code]

Component Description:
Occupant head positions from live simulations, extracted from live-simulated folder.

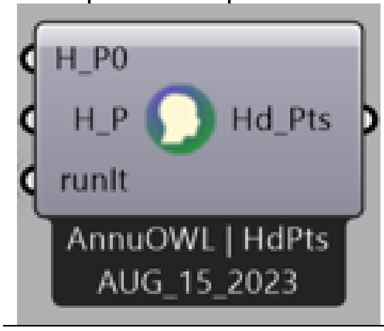
Inputs:
GPT_File: location of Grid definition file.
runIt: A boolean toggle. True=Live, False=Precached.
Outputs:
Head_Pnts: Head positions of the occupants.
Connections (Input):
HptFile, KeyVertSim

Connections (output):
HdPts

3.3.2. HdPts

Component Name: HdPts [Code]

Component Description:
Occupant head-position based definitions used for visualising OVNI-related data on Rhino viewports.



Inputs:
H_P0: (if pre-cached) Head positions of the occupants
H_P: (if live-sim) Head positions of the occupants
runIt: A boolean toggle for this component. True = Live, False = Precache.

Outputs:
Hd_Pts: Head positions of the occupants

Connections (Input):
CacheHPts, HPts, KeyVertSim

Connections (output):
DefVptFile

3.3.3. OVNI Legend

3.3.3. OVNIlgnd

Component Name: OVNIlgnd [Code]

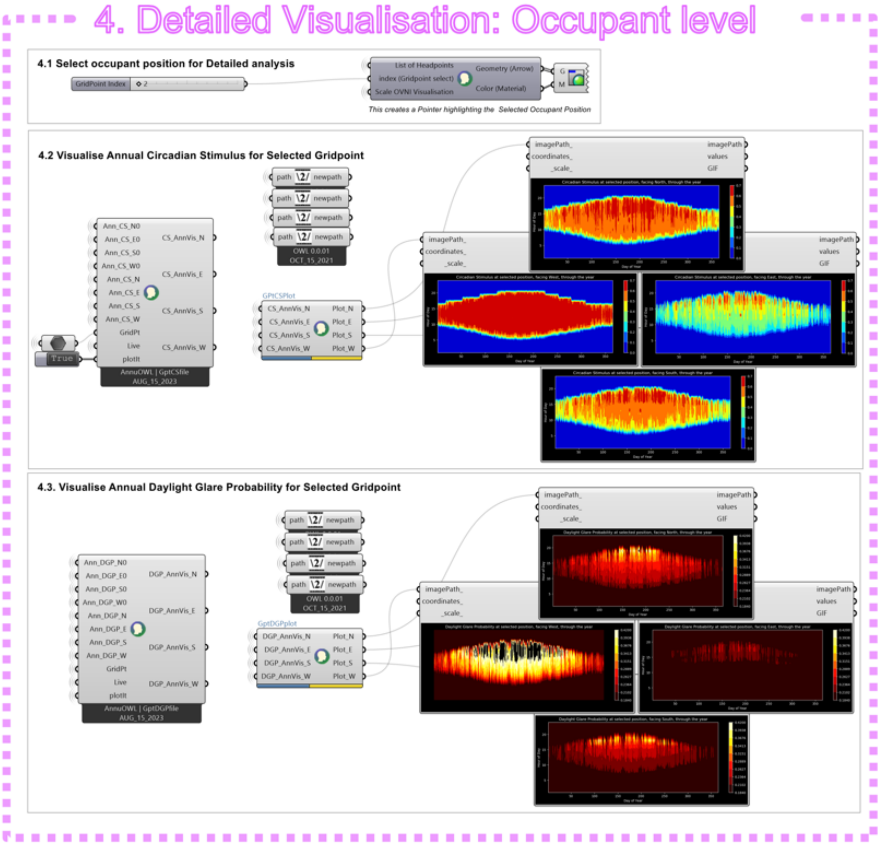
Component Description:
This component makes a legend for OVNIs performance in the defined palettes. The legend can be placed and rescaled based on user inputs. 5 Color palettes are supported.

Inputs:
Lgnd_Posn: Legend position on rhino viewport, defined through a 3D point.
Lgnd_Scl: A slider input for rescaling legends.
palette: Slider input for selecting color palettes. 5 supported options.

Outputs:
AllGeo: Geometry of legend, connects to CustomPreview.
AllMat: Materials/Colors of legend, connects to CustomPreview.

Connections (output):
Custom Preview

4. Detailed Visualisation: Occupant Level



In order to perform a detailed assessment beyond the OVNI diagrams, this section supports the generation of Annual Hourly heatmaps of CS and DGP for the four orientations, for each occupant position. Using a slider-input, the user can identify any occupant position index (which generates an arrow over the selected position to indicate the selection), and the 4 components in this section generate annual heatmaps for that occupant position. This can support a detailed exploration of time-bound opportunities for intervention, such as scheduling of blinds (for further minimizing DGP for a specific view-direction) or scheduling circadian lighting (for complementing available CS for any view-direction).

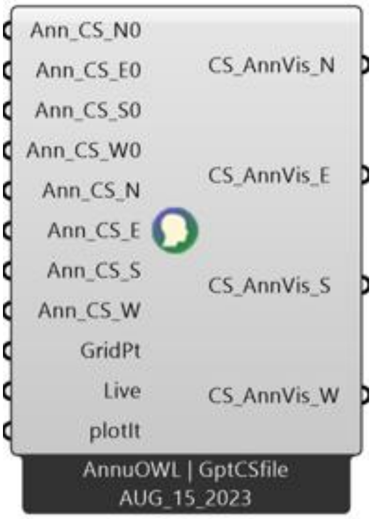
4.2. Visualise annual Circadian Stimulus for Selected Gridpoint

4.2. GptCSfile

Component Name: GptCSfile [Code]

Component Description:

This component extracts the CS data for a specific position over the year for 4 orientations. This data can then be plotted as heatmaps for visualisation.



Inputs:

Ann_CS_N0: (if Cached) the location of CSV file containing annual CS data for each grid point facing North
Ann_CS_E0: (if Cached) the location of CSV file containing annual CS data for each grid point facing East
Ann_CS_S0: (if Cached) the location of CSV file containing annual CS data for each grid point facing South
Ann_CS_W0: (if Cached) the location of CSV file containing annual CS data for each grid point facing West
Ann_CS_N: (if Live) the location of CSV file containing annual CS data for each grid point facing North
Ann_CS_E: (if Live) the location of CSV file containing annual CS data for each grid point facing East
Ann_CS_S: (if Live) the location of CSV file containing annual CS data for each grid point facing South
Ann_CS_W: (if Live) the location of CSV file containing annual CS data for each grid point facing West
GridPt: Index of gridpoint (Occupant Position) under investigation.
Live: A boolean Toggle. OFF = Cached, ON = Live
plotIt: A boolean Toggle to plot heatmaps.

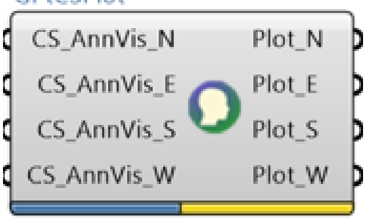
Outputs:

CS_AnnVis_N: Link to the CSV file containing Extracted Annual hourly Circadian Stimulus for the specified occupant position, Facing North.
CS_AnnVis_E: Link to the CSV file containing Extracted Annual hourly Circadian Stimulus for the specified occupant position, Facing East.
CS_AnnVis_S: Link to the CSV file containing Extracted Annual hourly Circadian Stimulus for the specified occupant position, Facing South.
CS_AnnVis_W: Link to the CSV file containing Extracted Annual hourly Circadian Stimulus for the specified occupant position, Facing West.

Connections (Input):
CacheCSMtxV, CSMtxV, KeyVertSim
Connections (output):
GptCSPlot

4.2. GptCSPlot

Component Name: GptCSPlot [Code]
Component Description:
This is a GHCPython component that plots in 24x365, the annual heatmaps of CS in four orientations – for the select occupant position at eye level on the vertical plane.

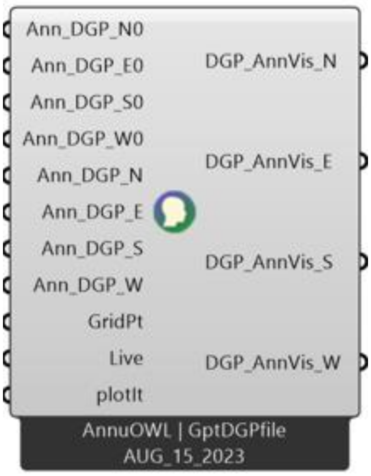


Inputs:
CS_AnnVis_N: Link to the CSV file containing Extracted Annual hourly Circadian Stimulus for the specified occupant position, Facing North.
CS_AnnVis_E: Link to the CSV file containing Extracted Annual hourly Circadian Stimulus for the specified occupant position, Facing East.
CS_AnnVis_S: Link to the CSV file containing Extracted Annual hourly Circadian Stimulus for the specified occupant position, Facing South.
CS_AnnVis_W: Link to the CSV file containing Extracted Annual hourly Circadian Stimulus for the specified occupant position, Facing West.
Outputs:
Plot_N: Annual CS Heatmap for the occupant position, facing North.
Plot_E: Annual CS Heatmap for the occupant position, facing East.
Plot_S: Annual CS Heatmap for the occupant position, facing South.
Plot_W: Annual CS Heatmap for the occupant position, facing West.
Connections (Input):
GptCSfile
Connections (output):
LB ImageViewer

4.3. Visualise annual Daylight Glare Probability for Selected Gridpoint

4.3. GptDGPfile

Component Name: GptDGPfile [Code]
Component Description:
This component extracts the DGP data for a specific position over the year for 4 orientations. This data can then be plotted as heatmaps for visualisation.

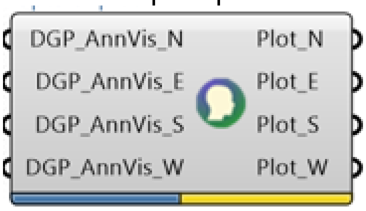


Inputs:
Ann_DGP_N0: (if Cached) the location of CSV file containing annual sDGP data for each grid point facing North
Ann_DGP_E0: (if Cached) the location of CSV file containing annual sDGP data for each grid point facing East
Ann_DGP_S0: (if Cached) the location of CSV file containing annual sDGP data for each grid point facing South
Ann_DGP_W0: (if Cached) the location of CSV file containing annual sDGP data for each grid point facing West
Ann_DGP_N: (if Live) the location of CSV file containing annual sDGP data for each grid point facing North
Ann_DGP_E: (if Live) the location of CSV file containing annual sDGP data for each grid point facing East
Ann_DGP_S: (if Live) the location of CSV file containing annual sDGP data for each grid point facing South
Ann_DGP_W: (if Live) the location of CSV file containing annual sDGP data for each grid point facing West
GridPt: Index of gridpoint (Occupant Position) under investigation.
Live: A boolean Toggle. OFF = Cached, ON = Live
plotIt: A boolean Toggle to plot heatmaps.

Outputs:
DGP_AnnVis_N: Link to the CSV file containing Extracted Annual hourly sDGP for the specified occupant position, Facing North.
DGP_AnnVis_E: Link to the CSV file containing Extracted Annual hourly sDGP for the specified occupant position, Facing East.
DGP_AnnVis_S: Link to the CSV file containing Extracted Annual hourly sDGP for the specified occupant position, Facing South.
DGP_AnnVis_W: Link to the CSV file containing Extracted Annual hourly sDGP for the specified occupant position, Facing West.
Connections (Input):
CacheDGPMtxV, DGPMtxV, KeyVertSim
Connections (output):
GptDGPplot

4.3. GptDGPplot

Component Name: GptDGPplot [Code]
Component Description:
This is a GHCPython component that plots in 24x365, the annual heatmaps of DGP in four orientations – for the select occupant position at eye level on the vertical plane.



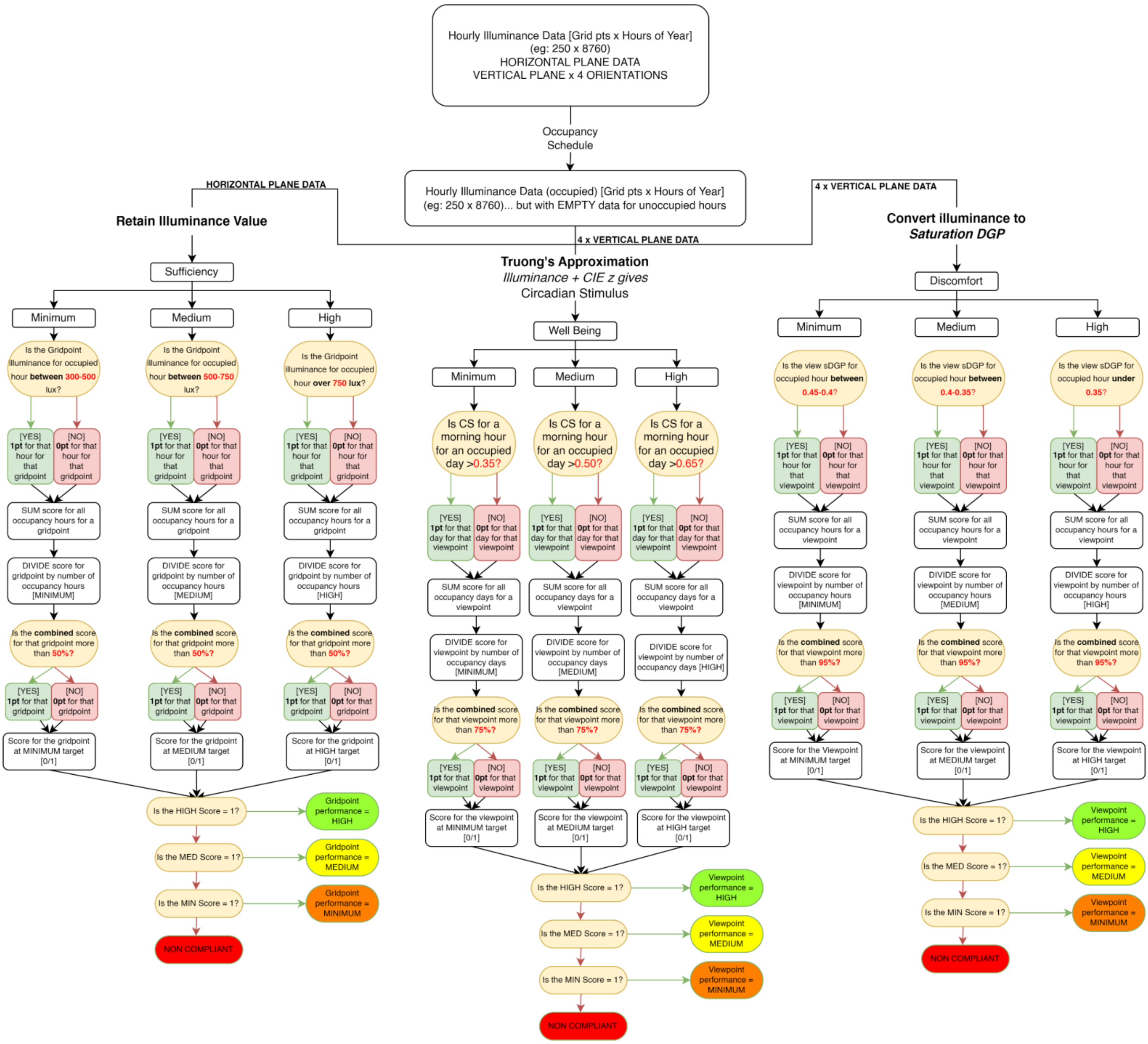
Inputs:
DGP_AnnVis_N: Link to the CSV file containing Extracted Annual hourly sDGP for the specified occupant position, Facing North.
DGP_AnnVis_E: Link to the CSV file containing Extracted Annual hourly sDGP for the specified occupant position, Facing East.
DGP_AnnVis_S: Link to the CSV file containing Extracted Annual hourly sDGP for the specified occupant position, Facing South.
DGP_AnnVis_W: Link to the CSV file containing Extracted Annual hourly sDGP for the specified occupant position, Facing West.
Outputs:
Plot_N: Annual DGP Heatmap for the occupant position, facing North.
Plot_E: Annual DGP Heatmap for the occupant position, facing East.
Plot_S: Annual DGP Heatmap for the occupant position, facing South.
Plot_W: Annual DGP Heatmap for the occupant position, facing West.
Connections (Input):
GptDGPfile
Connections (output):
LB ImageViewer

Logical Flowchart for assigning compliance

The following flowchart presents the pathway for evaluation of compliance (High/Medium/Low/Non-compliant) for sufficiency, circadian/non-visual potential, and protection from glare risk, which is then visualized via OVNI hemisphere, OVNI inner ring, and OVNI middle ring, respectively.

The calculation begins with the outputs of the AnnIllV component: Radiance DC method, grid-based evaluation of horizontal and vertical illuminance, for each occupant position, across all 8760 annual hours. In the component, horizontal illuminance values are evaluated at the (editable) table height, and the vertical illuminance values are evaluated for four principle orientations at the (editable) eye height.

Evaluation of compliance/performance in daylight autonomy and daylight sufficiency is executed by DltMtxV and Vert1SPrc components, and visualized in the OVNI hemisphere. Performance towards circadian/non-visual potential is evaluated by CSMtxV and Vert2SPrc components, and visualized by OVNI inner ring. Performance towards protection from risk of discomfort glare is evaluated by DGPMtxV and Vert3SPrc components, and visualized by OVNI second ring.



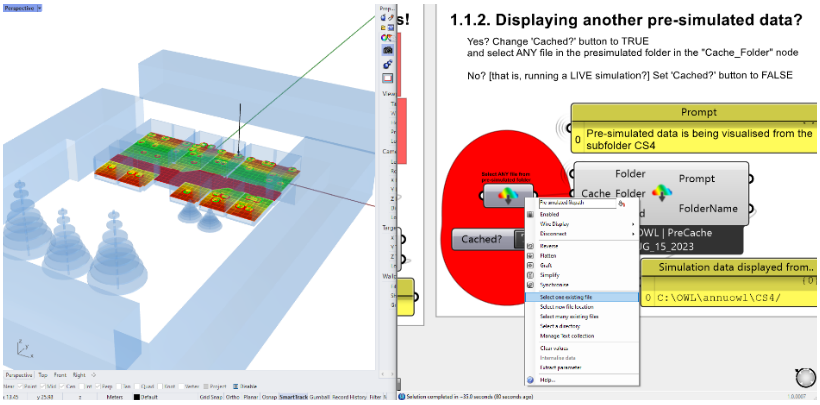
Options for user-customizations

The following section visually presents the possibility of customizing the simulation outputs, supported by AnnuOWL. This includes:

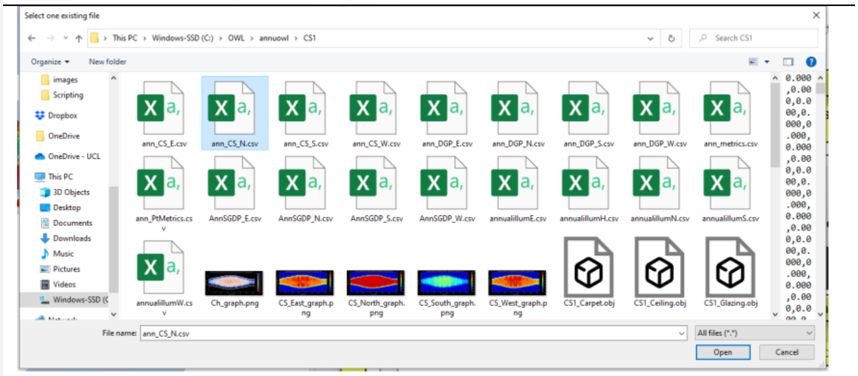
- Selection between pre-simulated cases
- Selection between template geometries, and options for reorienting geometries
- Selection between metrics (available CBDMs)
- Visualization (placement, resizing, color palettes)
- Hourly heatmaps at occupant positions
- Options for re-calculation (e.g. geometry reorientation, variation in thresholds, change of schedules, etc.)

1. Displaying a pre-simulated case.

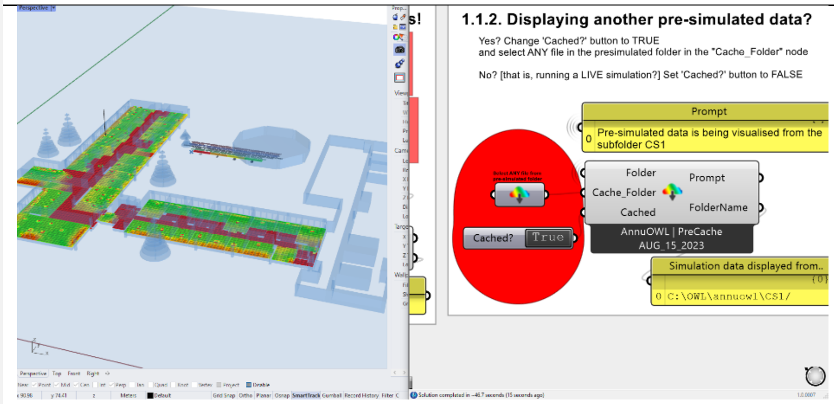
Right click on ‘Select File’ in 1.1.2 (Display pre-simulated data?’)



Select any file in another pre-simulated folder, click ‘Open’

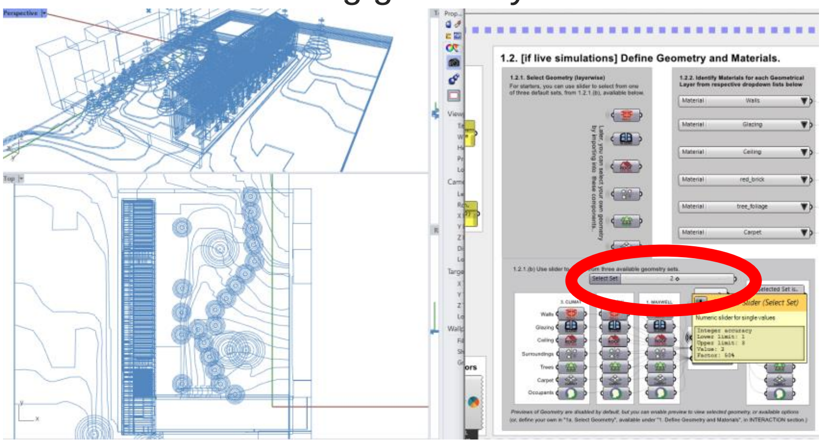


AnnuOWL loads the pre-simulated evaluations

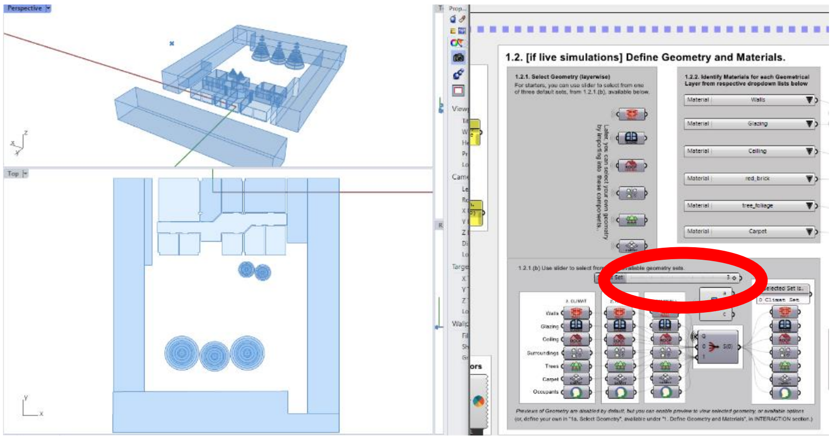


2. Using template Geometry for simulation AND Reorienting geometry

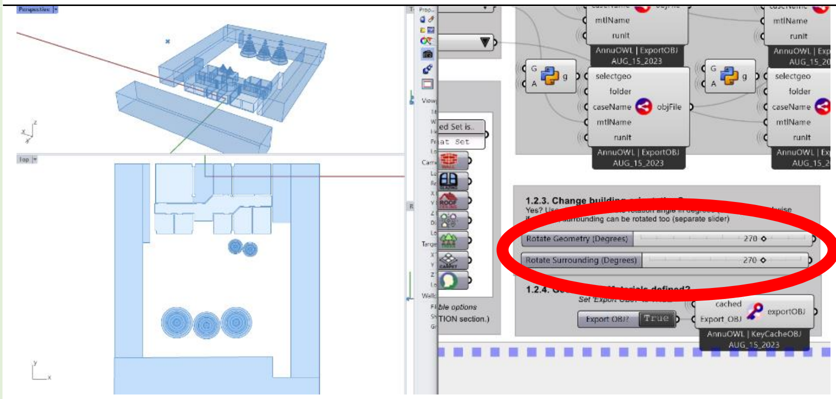
Click on ‘select set’ slider, and choose another option.



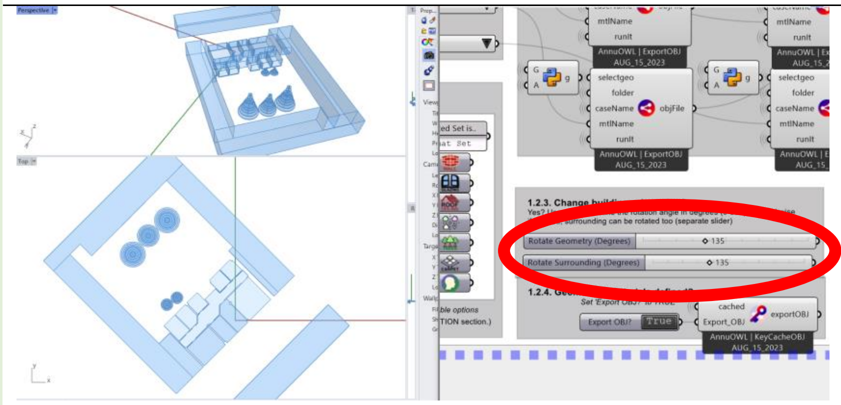
AnnuOWL displays another template geometry that can be simulated



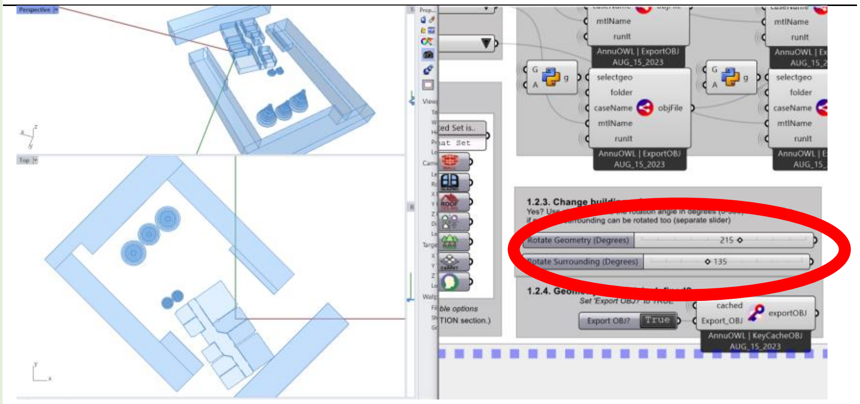
Use sliders: “Rotate Geometry” and “Rotate Surroundings”



AnnuOWL displays the reoriented geometry
(1. With equal variation in both sliders)



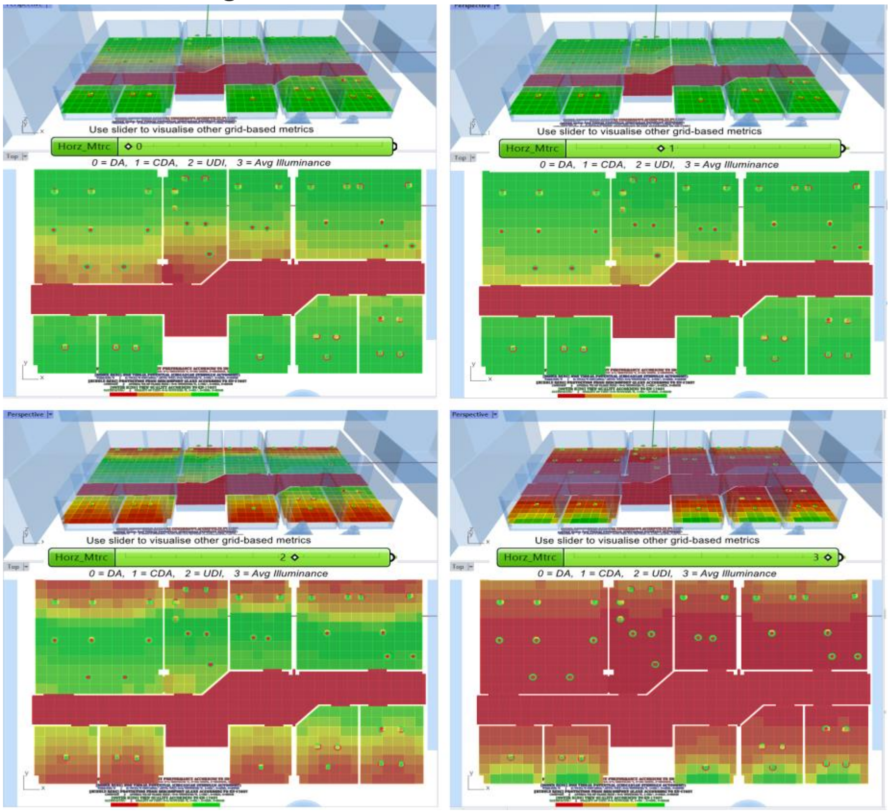
AnnuOWL displays the reoriented geometry
(2. With differential variation in both sliders)



3. Displaying a different CBDM metric on the grid

Use the “Horizontal Metric” slider to shuffle between visualization options:

- Daylight Autonomy,
- Continuous Daylight Autonomy,
- Useful Daylight Illuminance, and
- Average annual Illuminance

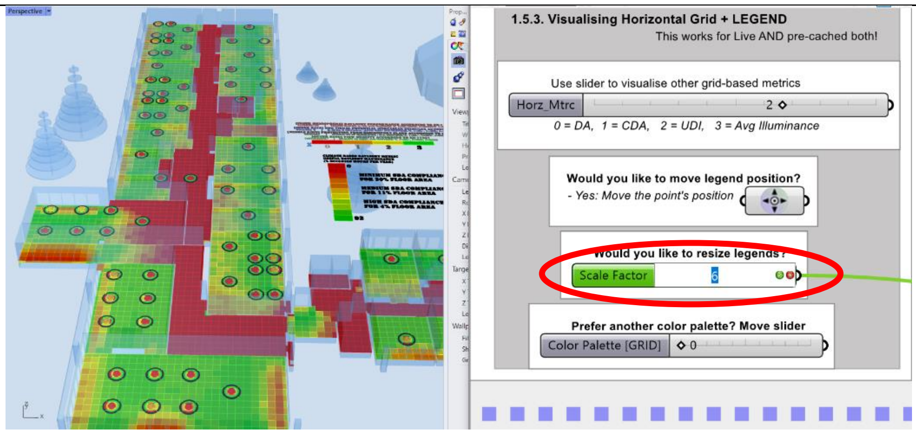


4. Moving and Resizing Legends

Set the preview of Legend position to TRUE

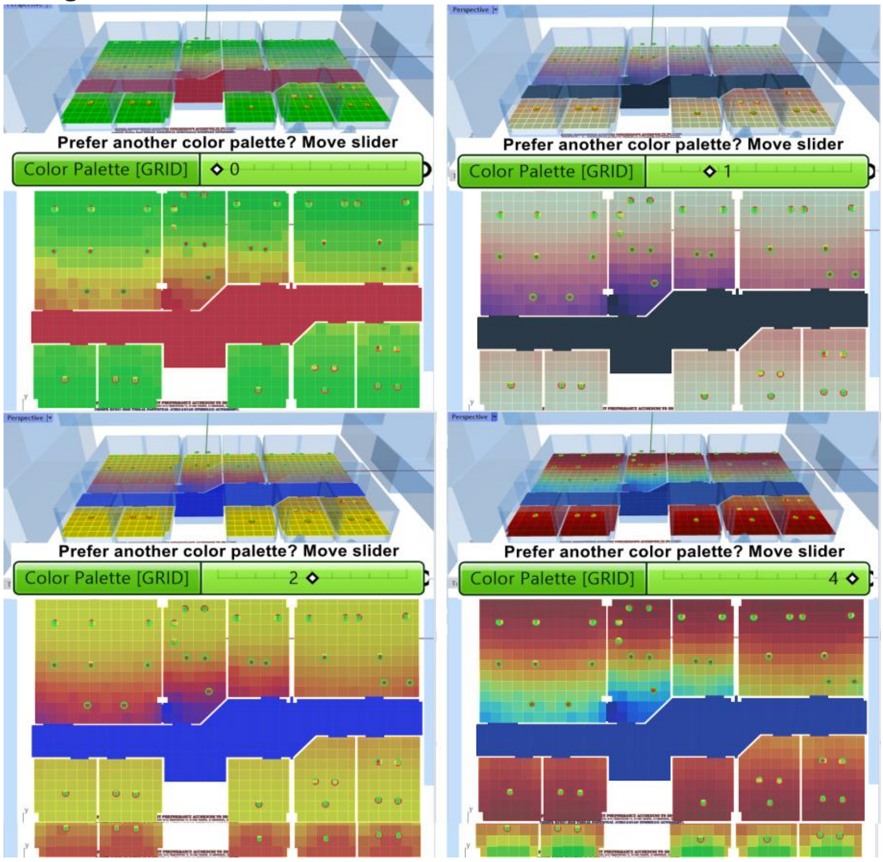
Move the highlighted point (on Rhino) in x, y, and/or z direction

Use the resize legend slider to change the size of the legend



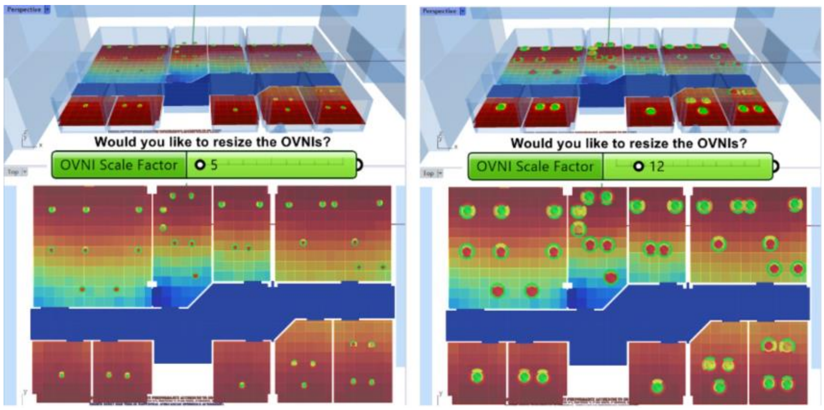
5. Using a different color palette for grid visualization

Use the color palette slider to display from among 5 available palette options



6. OVNI Visualization

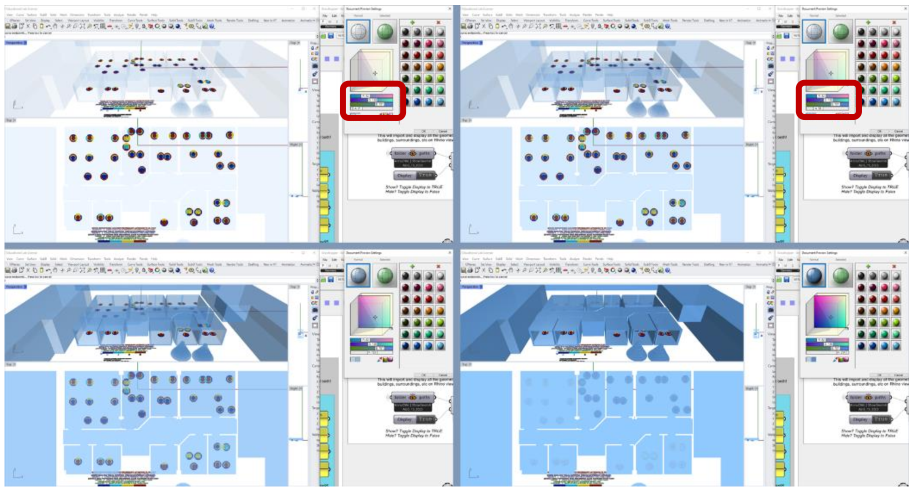
To resize the OVNI diagrams, use the 'resize OVNI' slider



For using a different color palette, use the 'color palette' slider to choose from 5 available options.

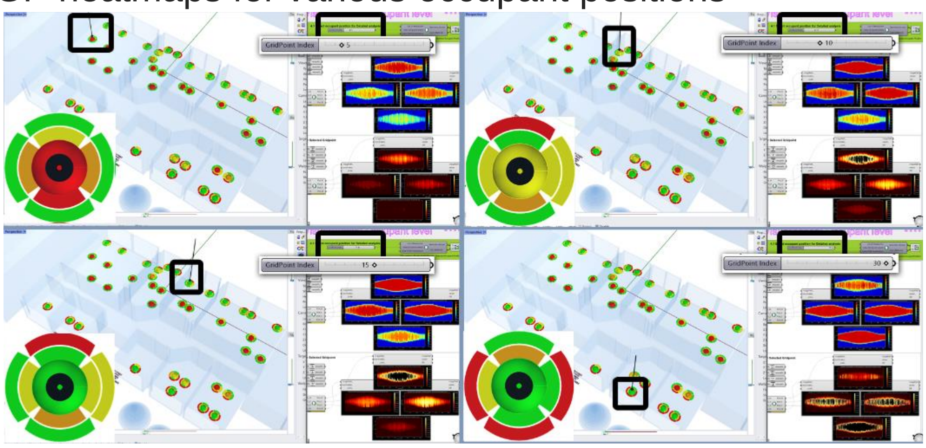
7. Displaying geometry on Rhino Canvas

To vary the opacity of geometry on Rhino canvas, change the alpha value as demonstrated.



8. Generating Annual CS and DGP heatmaps for various occupant positions

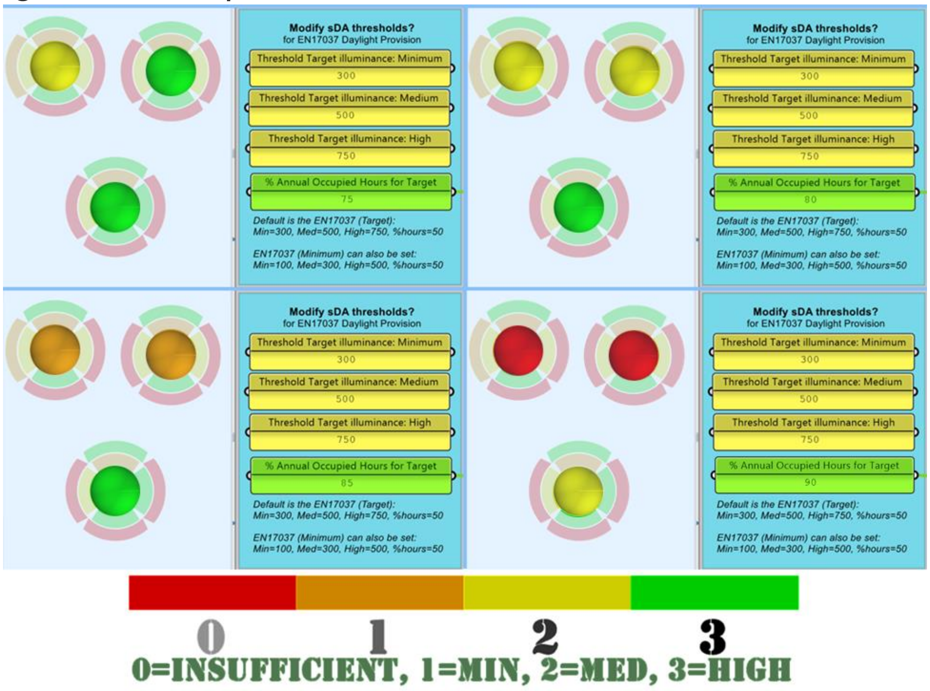
For hourly analysis of CS and DGP at occupant level, the slider can be used to select the occupant position (this generates an indicator arrow above the selected position) and AnnuOWL generates annual hourly heatmaps of CS and DGP for 4 orientations from the selected position. This allows a deeper investigation of hourly intervention opportunities, than supported by the OVNIs.



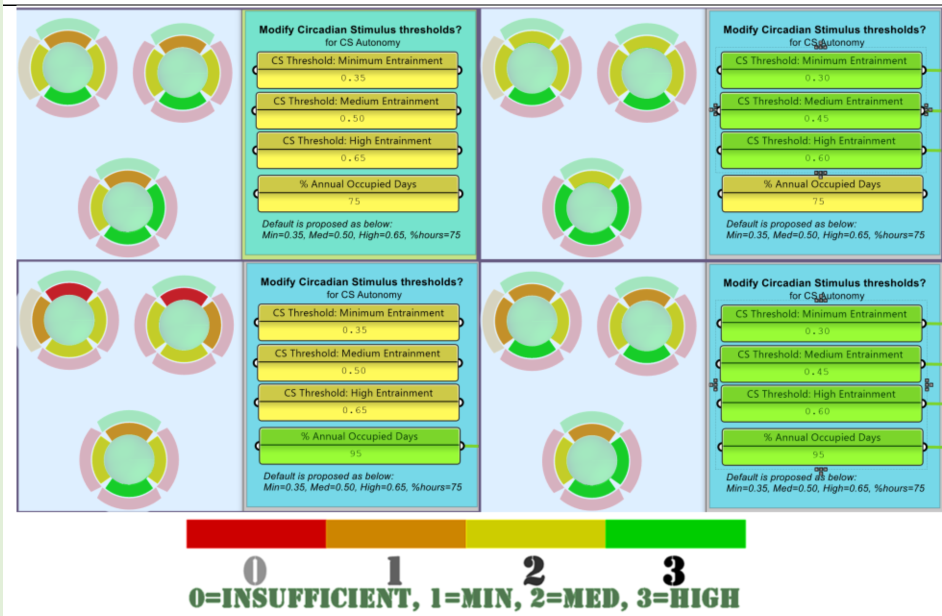
9. Options for recalculation through revised inputs

By modifying panel inputs, the evaluated metrics are recalculated and visualized on OVNI.

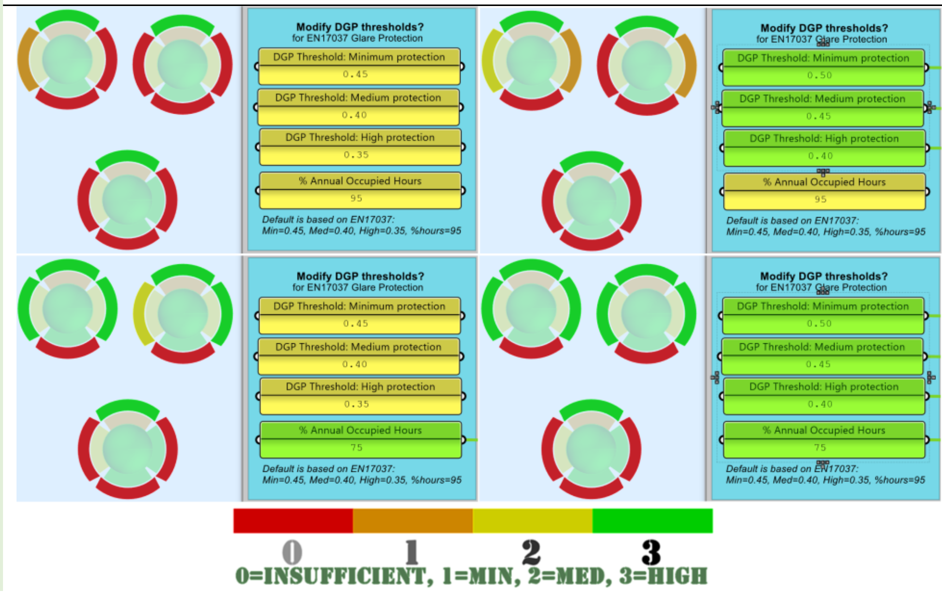
For example,
1. Varying the time thresholds for DA (% occupied annual hours) recalculates daylight autonomy compliance, and changes the visualization on the OVNI hemisphere, as demonstrated.
(Note: The hemisphere of OVNI is highlighted while the rings are faded via image-processing, to emphasize on the hemisphere's visualization)



2. Varying the CS thresholds for compliance *(minimum CS required for at-least one hour in the morning on occupied days, for each view direction for each occupant's position, under various compliance thresholds: Minimum, Medium, High)*, as well as the time thresholds *(% of annual occupied days)* changes the visualization on the OVNI first ring, as demonstrated.
(Note: to emphasize on the first ring, the rest of OVNI is faded via image processing)



3. Varying the DGP thresholds for compliance *(maximum allowed DGP each occupied hour, for each view direction for each occupant's position, under various compliance thresholds: Minimum, Medium, High)*, as well as the time thresholds *(% of annual occupied hours)* changes the visualization on the OVNI second ring, as demonstrated.
(Note: to emphasize on the second ring, the rest of OVNI is faded via image processing)



Variation in Grid-visualization using panel inputs

Modifying grid spacing (x, y) and grid height above floor level changes the evaluation of grid-based CBDMs, as well as the compliance metric of sDA, as demonstrated.
In this example, grid spacing is changed from 1.5mx1.5m to 0.4m x 0.4m.

DA and UDI visualizations are presented. sDA compliance increases from 64/59/47% floor area to 70/61/51% floor area – for EN17037 Minimum/Medium/High target compliance.