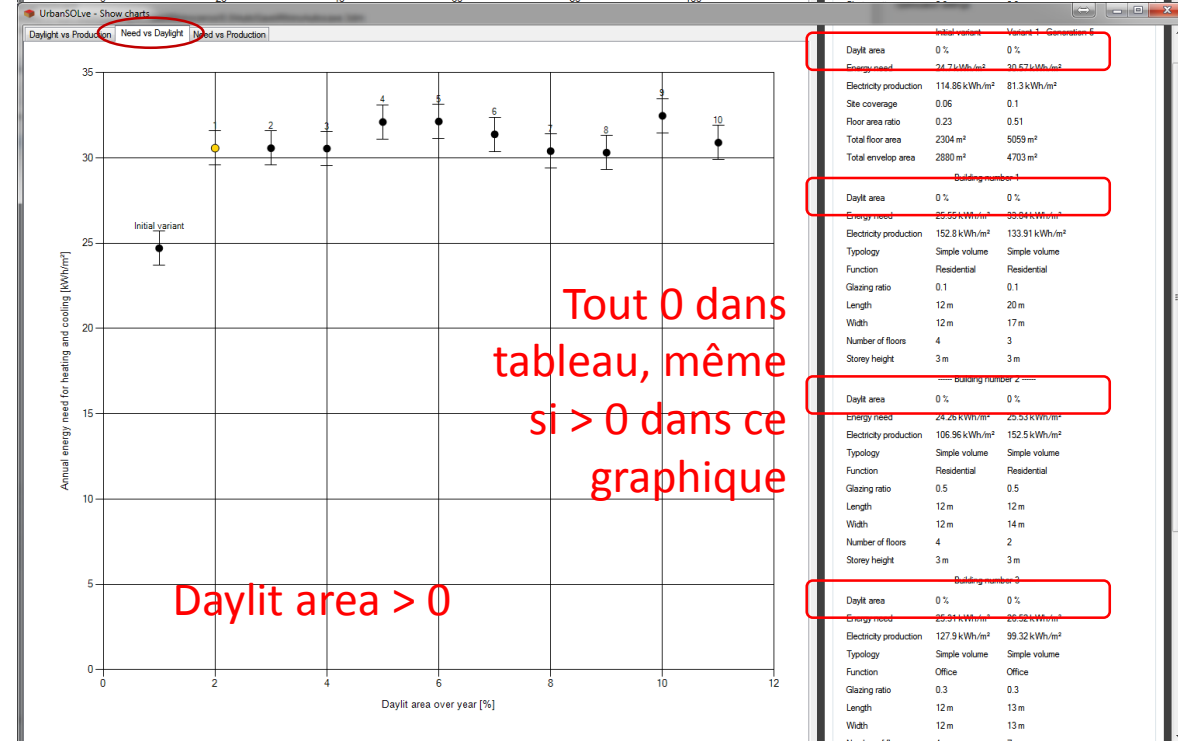


Il semble vraiment y avoir un problème avec les métamodèles du daylit area.

Déjà, les valeurs sont différentes d'un graphique à l'autre. Puis, elles semblent généralement faibles, ce qui ne devrait pas être le cas pour un bâtiment avec glazing ratio (WWRatio) assez élevé (genre > 0.5).



Je pense que l'erreur vient sans doute d'erreurs dans le calcul des valeurs pour chaque paramètre (voir prochaines slides là-dessus).

Tu peux bien reconstruire tous les paramètres ?

Je crois mieux de commencer par régler les paramètres d'irradiation et de surface par orientation (item 4 dans la liste excel) et celui du WFRatio et FormFactor (voir les prochaines slides).

```
329
330     for (int i = 0; i < meanIrradiation.Length; ++i)
331     {
332         meanIrradiation[i] /= pointNumber[i];
333     }
334     return meanIrradiation;
335 }
336
337 public double getWfRatio()
338 {
339     return description.glazingRatio / getTotalFloorArea();
340 }
341
342 // Rapport entre la surface de facade Nord et celle de l'enveloppe totale
343 public double getNorthFacRatio()
344 {
345     return getFacadeArea().ElementAt(0) / getTotalEnvelopeArea();
346 }
347
348 public double getEastFacRatio()
349 {
350     return getFacadeArea().ElementAt(2) / getTotalEnvelopeArea();
351 }
352
```

(Dans le fichier Building.cs)

Faux !

$$\text{WfRatio} = \frac{\text{glazingRatio} * \text{FacadeArea}}{\text{TotalFloorArea}}$$

Cela pourrait expliquer les problèmes avec les métamodèles du Daylit Area car ce paramètre est très important (a un coefficient large pour les 3 typologies).

```
130
131    /// <summary>
132    /// Surface de plancher totale / surface de l'enveloppe totale
133    /// </summary>
134    /// <returns>
135    /// form factor
136    /// </returns>
137    public double getFormFactor()
138    {
139        return getTotalFloorArea() / getTotalFacadeArea();
140    }
141
142    /// <summary>
143    /// Surface d'enveloppe totale
144    /// </summary>
145    /// <returns>
146    /// total envelop area [m^2]
147    /// </returns>
148    public double getTotalEnvelopeArea()
149    {
150        return getTotalFacadeArea() + getRoofArea();
151    }
152
153    /// <summary>
154    ///
155    /// </summary>
156    /// <returns>
157    /// Roof area [m^2]
158    /// </returns>
159    public double getRoofArea()
160    {
161        return getFootprintArea();
162    }
163
```

(Dans le fichier Building.cs)

Faux !

FormFactor = $\frac{\text{TotalFloorArea}}{\text{TotalEnvelopeArea}}$

```

49 // Daylit area
50 public abstract double getDaylightAutonomy(double floorAreaRatio, double siteCoverage);
51
52
53 public double getActiveSolarEnergy()
54 {
55     List<IrradiationPoint> roofSensors = getRoofSensors();
56     List<IrradiationPoint> facadeSensors = getFacadeSensors();
57     double roofProduction = 0.17 * ((getRoofArea() / roofSensors.Count) * roofSensors.Sum(sensor => sensor.irradiationValue));
58     double facadeProduction = (1 - description.glazingRatio) * 0.17 * ((getTotalFacadeArea() / facadeSensors.Count) * facadeSensors
59     activeSolarEnergy = (roofProduction + facadeProduction) / getTotalFloorArea();
60     return roofProduction + facadeProduction;
61 }
62
63 public double getEnergyNeed(double floorAreaRatio, double siteCoverage)
64 {
65     double buildingPerformance = 32.2978018230063
66         + 18.9284668032393 * getFormFactor()
67         + -3.42095697920017 * description.glazingRatio
68         + -2.84751491537644 * getWfRatio()
69         + -0.0317452282044676 * getMeanIrradiation().ElementAt(6) // MeanFacIrrad
70         + -0.000576084353749911 * getMeanIrradiation().ElementAt(0) // MeanNorthFacIrrad
71         + 0.000523589268194919 * getMeanIrradiation().ElementAt(1) // MeanSouthFacIrrad
72         + -0.0226333857127084 * getMeanIrradiation().ElementAt(3) // MeanWestFacIrrad
73         + -0.0109206283265959 * irrPerFloorArea().ElementAt(1) // SouthFacIrradPerFA
74         + -0.00724773184901299 * getFormFactor() * getMeanIrradiation().ElementAt(1)
75         + 0.0534332403142441 * getWfRatio() * getMeanIrradiation().ElementAt(6)
76         + 0.00917774739839259 * getWfRatio() * getMeanIrradiation().ElementAt(0)
77         + 0.0000412735619418759 * getMeanIrradiation().ElementAt(6) * getMeanIrradiation().ElementAt(3);
78
79     /*
80     switch (description.program)
81     {
82         case Program.office:
83             buildingPerformance = CST.off_constant
84                 + CST.off_plotRatio * floorAreaRatio
85                 + CST.off_siteCoverage * siteCoverage
86                 + CST.off_formFactor * getFormFactor()
87                 + CST.off_vwRatio * description.glazingRatio
88                 + CST.off_wfRatio * getWfRatio()
89                 + CST.off_meanRoofIrr * getMeanIrradiation().ElementAt(4)
90                 + CST.off_meanSouthFacIrr * getMeanIrradiation().ElementAt(1)
91                 + CST.off_envelopIrrPerFa * irrPerFloorArea().ElementAt(5)
92                 + CST.off_meanEnvelopIrr * getMeanIrradiation().ElementAt(5)
93                 + CST.off_roofIrrPerFa * irrPerFloorArea().ElementAt(4)
94                 + CST.off_northFacIrrPerFa * irrPerFloorArea().ElementAt(0)
95                 + CST.off_southFacIrrPerFa * irrPerFloorArea().ElementAt(1);
96             break;
97         case Program.residential:
98             buildingPerformance = CST.app_constant
99                 + CST.app_nbOffFloors * numberOfFloors

```

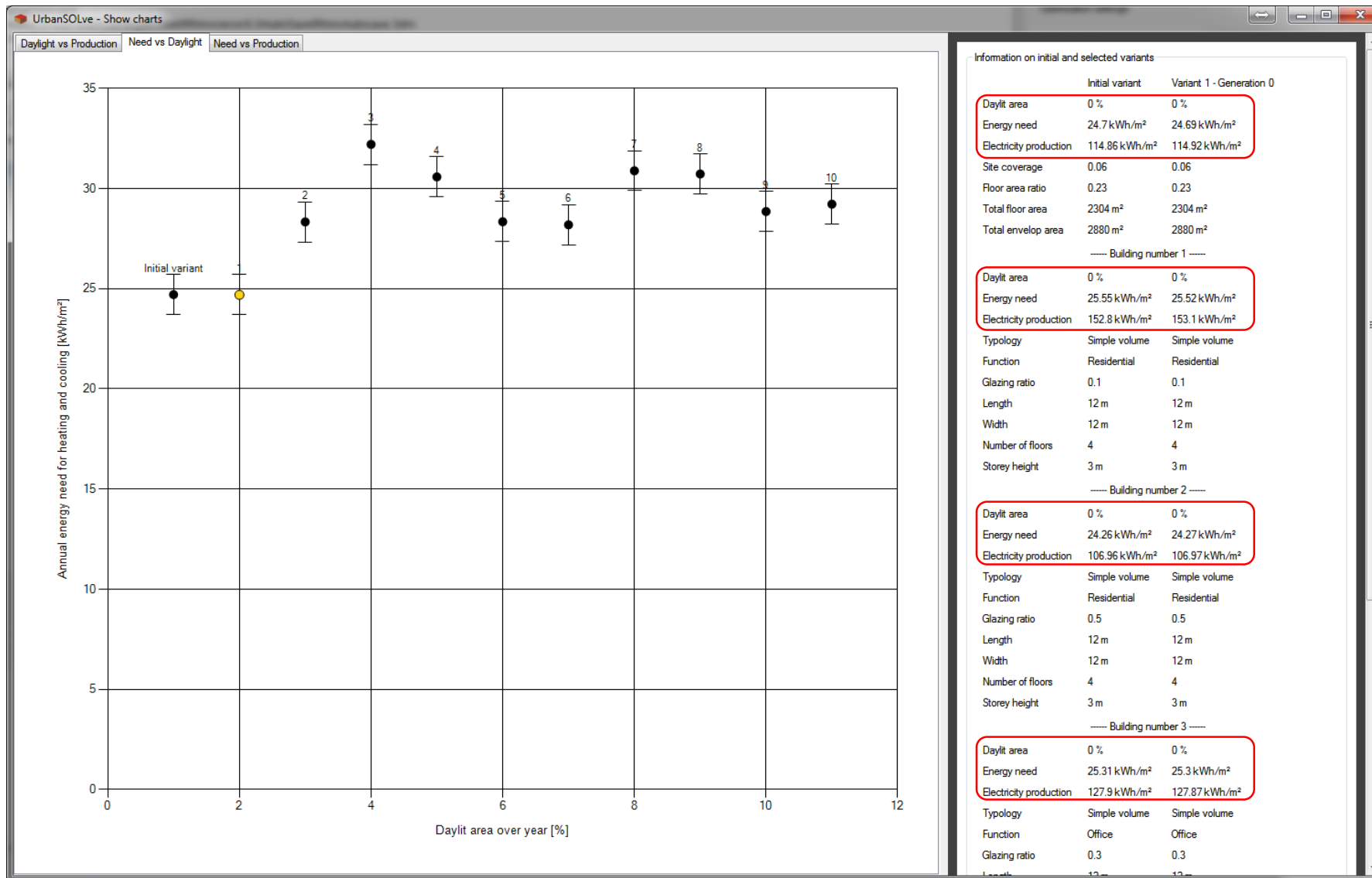
(Dans le fichier Building.cs)

Correspond au métamodèle pour les bâtiments avec fonction 'office', ok, mais est-ce que celui pour les bâtiments avec fonction 'apartment' se trouve ailleurs..?

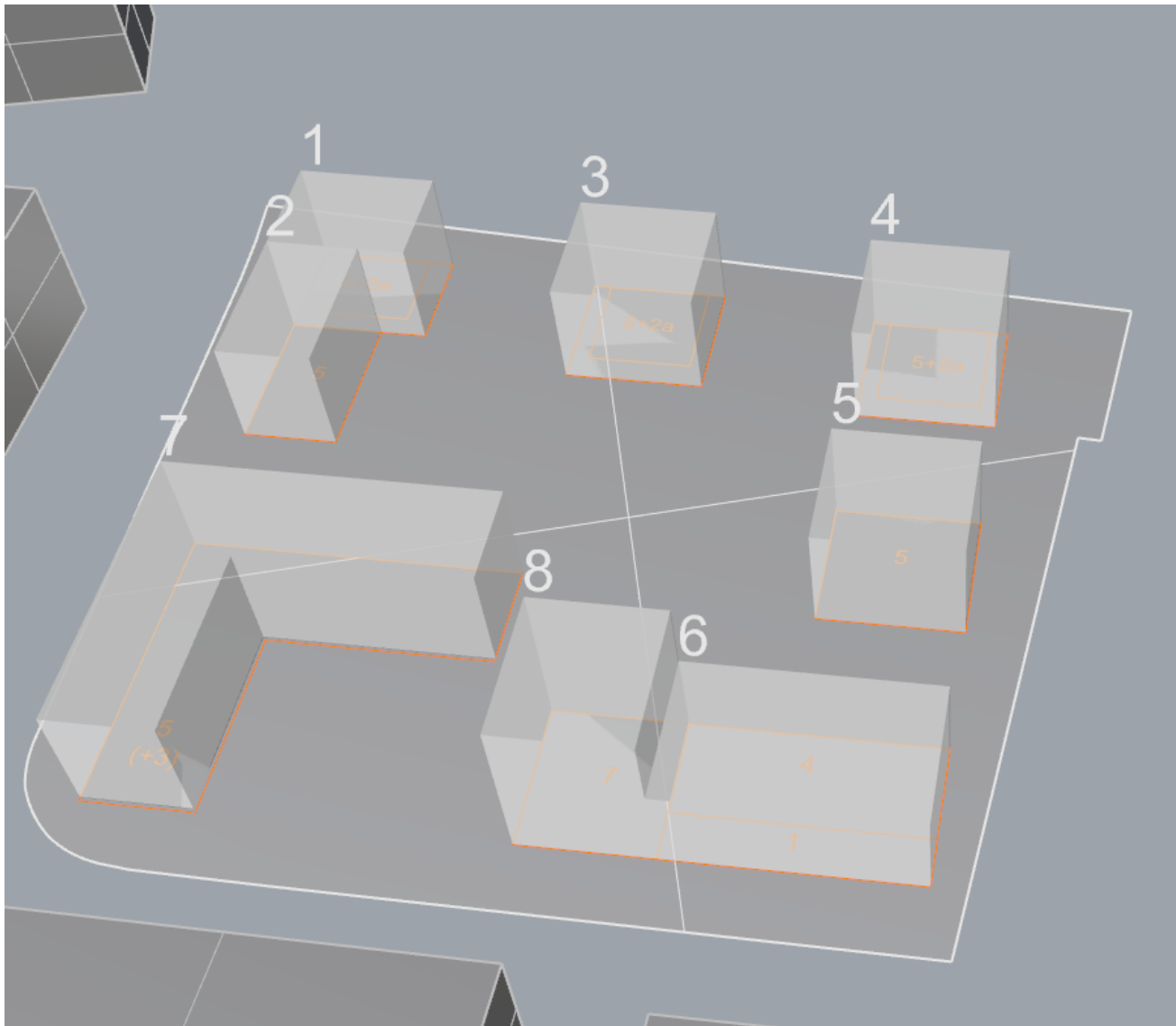
```
95
96     public const string BAKE_LAYER_NAME = "UrbanSolve";
97     public const string TMP_LAYER_NAME = "tmp";
98     public static Color BAKE_LAYER_COLOR = Color.Red;
99
100     #region OPTIMIZATION
101     public const int MESH_DENSITY = 30;
102     public const double DAYLIGHT_AUTONOMY_CST = 47.6673;
103
104     // DAYLIGHT AUTONOMY -----
105     public const double da_off_constant = -80.742562485692;
106     public const double da_off_plotRatio = -9.6393337165692;
107     public const double da_off_siteCoverage = 547.096561071363;
108     public const double da_off_exEnvArea = 0.0042596026831734;
109     public const double da_off_footprintArea = 0.166269008734934;
110     public const double da_off_formFactor = -40.8010358472293;
111     public const double da_off_eastFacRatio = -174.926844561956;
112     public const double da_off_southFacRatio = -18.2548209923902;
113     public const double da_off_wvRatio = -161.346611900301;
114     public const double da_off_wfRatio = 304.091625539356;
115     public const double da_off_meanRoofIrr = -0.00124669278508557;
116     public const double da_off_meanFacIrr = 0.0563368857359545;
117     public const double da_off_meanSouthFacIrr = -0.0103907708870867;
118     public const double da_off_facIrrPerFa = 0.187308861661125;
119     public const double da_off_eastFacIrrPrFa = -0.0280547955951425;
120     public const double da_off_PlotRatio_ExpEnvArea = 0.00123832889465642;
121     public const double da_off_SiteCoverage_MeanFacIrrad = -0.857390511693495;
122     public const double da_off_ExpEnvArea_FootprintArea = -0.0000321764820669752;
123     public const double da_off_ExpEnvArea_FormFactor = -0.00508061791889866;
124     public const double da_off_FootprintArea_MeanRoofIrrad = -0.000120092877308541;
125     public const double da_off_FormFactor_FacIrradPerFa = 0.0828921713701716;
126     public const double da_off_EastFacRatio_MeanFacIrrad = 0.296426561524879;
127     public const double da_off_WvRatio_MeanFacIrrad = 0.258065769115172;
128     public const double da_off_WfRatio_FacIrradPerFa = -0.3854215973439;
129
130     public const double da_app_constant = 81.1632920947334;
131     public const double da_app_expEnvArea = -0.0000983165166846544;
132     public const double da_app_formFactor = -125.77610906595;
133     public const double da_app_wvRatio = -6.1028869159489;
134     public const double da_app_wfRatio = 135.486601154908;
135     public const double da_app_meanRoofIrr = -0.11084066706723;
136     public const double da_app_meanFacIrr = 0.0620336225845539;
137     public const double da_app_roofIrrPerFa = 0.00481984071744582;
138     public const double da_app_northFacIrrPerFa = 0.212764031275194;
139     public const double da_app_southFacIrrPerFa = -0.0689059101335217;
140     public const double da_app_ExpEnvArea_WfRatio = 0.00801325064995131;
141     public const double da_app_FormFactor_MeanRoofIrrad = 0.0962278093607832;
142     public const double da_app_WvRatio_WfRatio = -253.454793675204;
143     public const double da_app_WfRatio_MeanFacIrrad = 0.323756788929631;
144     public const double da_app_WfRatio_RoofIrradPerFa = 0.140164041356989;
145
146     // -----
147     public const double off_constant = 19.1141836219013;
148     public const double off_plotRatio = 0.545327925719241;
```

(Dans le fichier CST.cs)

On dirait bien une version antérieure des
métamodèles de Daylit Area. Ne sont plus
utilisées ?

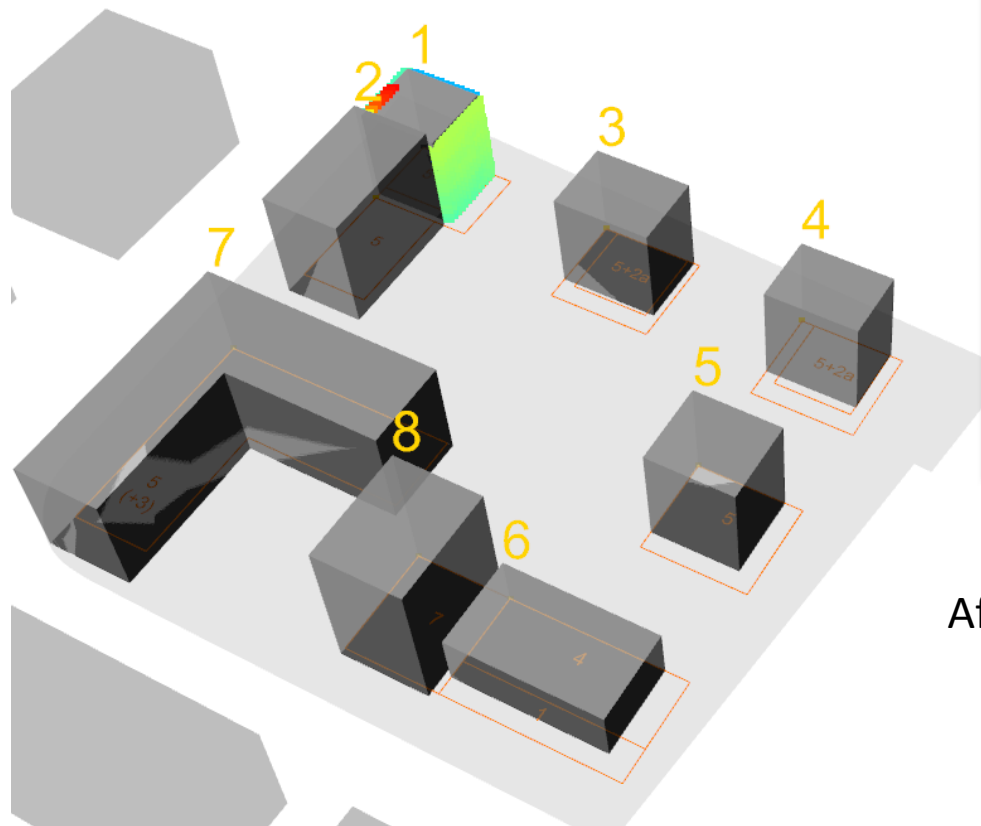


Arrondir les
valeurs de
performance
à 1 chiffre
après le point



Ce design initial est non valable on dirait (pas de carte d'irradiation affichée...), tu crois que c'est parce que j'ai mis des bâtiments collés ? (1 et 2 ainsi que 6 et 8)

Si oui, peut-on enlever la contrainte d'une distance 0 entre 2 bâtiments (qui serait alors acceptable), et ne garder que la contrainte sur deux bâtiments qui se rentrent vraiment dedans (distance < 0 finalement) ?



Site coverage min 0.0

Site coverage max 1.0

Floor area ratio min 0.0

Floor area ratio max 10.0

Rhino scene

Ground layer name Parcelle

Context layer name (optional) Contexte

Generation 0

Variant 4

Bake initial variant

Bake selected variant

Export results

Show charts

Show selected variant ☒

Show initial variant ☐

Annual Irradiation (kWh/m²)

0 400 800 1200

Affichage très étrange...

UrbanSOLve - Edit buildings

Typology

☒ Simple volume
 ☐ Courtyard
 ☐ L-shaped

Rotation

☐ 0°
 ☐ 90°
 ☐ 180°
 ☐ 270°
 ☒ θ

Alignment

☒ Corner
 ☐ Center

Dimensions (initial and ranges)

	Initial	Minimum	Maximum
Length [m]	<input type="text" value="12"/>	<input type="text" value="12"/>	<input type="text" value="15"/>
Width [m]	<input type="text" value="12"/>	<input type="text" value="12"/>	<input type="text" value="15"/>
Depth [m]	<input type="text" value="8"/>	<input type="text" value="6"/>	<input type="text" value="8"/>
Number of floors	<input type="text" value="4"/>	<input type="text" value="2"/>	<input type="text" value="5"/>

Other

Function:
 Climate:

Window-to-wall ratio:
 Compliance:

Storey height [m]:

Add one building
 Delete one
 Delete all

Close

Rotation avec angle 'custom' : peut juste aller de 0 à 270, pourquoi pas jusqu'à 360 ?

- On dirait que les valeurs pour le Energy need augmentent à chaque génération (et que celle pour le Electricity production diminuent un peu) → est-ce que les objectifs sont bien codés ?
 - Energy need -> on veut minimiser (directement ok ainsi alors car l'algorithme d'optimisation chercher à minimiser)
 - Electricity production -> on veut maximiser (alors on veut donner à minimiser -x à l'algorithme)
 - Daylit area -> on veut maximiser (alors on veut donner à minimiser -x à l'algorithme)