

選んだ建築 - ラ・トゥーレット修道院 (ル・コルビュジェ/ヤニス・クセナキス)

クセナキスの初期のスケッチには屋上へ向かう斜路と、屋上回廊が構想されている。しかし実際には斜路は実現されず、屋上も一周できない。

本プロジェクトの目標は、斜路を増築することで、修道院の回廊性を増幅させることである。

与条件を大きく四種類に分け、整理した。

- 1 - 現状のラ・トゥーレットは中庭側に構造壁はなく、斜路の接続は可能。
- 2 - 柱の中間点を接続候補として116箇所プロットし、情報を埋め込む。
- 3 - 部屋同士の相性を12点満点で採点し、表にする。
- 4 - 現状の動線をエッジとし、グラフ理論を用いて特定の接続候補からの全ての接続候補への最短経路をダイクストラ法を用いて求める。

全ての接続候補を繋ぐ線は約5000本存在するが、ここから適切な斜路となる線を選抜していく。実装したシステムで物理的な制限や接続候補の評価を行うと、最終候補は10本程度となった。

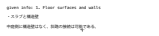
最終候補から4本の斜路を生成した2つの別解はそれぞれ、屋上の一周や、僧房の空間から屋上へのシークエンスなどを実現している。部屋の相性と距離のみで評価したにもかかわらず、回廊性を増幅させる結果を得られことは興味深く、またル・コルビュジェによる十字の通路が現状とは全く異なる要素に見える。

Le Couvent Saint Marie de la Tourette (Le Corbusier, Iannis Xenakis):[†] Ramps for Alternative Cloister

neither ramp nor cloister was realised.



generate Ramps for Alternative Cloister



node = 116 points (position, orientation, floor, type)



given info 3. score list for combination of functions ・スコアリスト シ・トゥーレット特選物の形質形成の基となる機能は大きく、	(cf. what is (later) of the monastery) ・修道院における回廊 シ・トゥーレット修道院には、修道院特有の回廊がその回廊の壁の上に成立す、ル・ム・ドゥ・ジュエは十字の形をもつて回廊の回廊を規定したと知られていて、
---	--

図解は都市と農村をつなぐ交通手段であり、都市の中心部の空間でもあり、農村の中心部の空間でもあるので、都市的要素と農村的要素が混在する。都市部に限定することでもなく、農村部に限定することでもなく、**中間的な空間である。そのため、今の中心部の空間は都市空間に近しい空間に近しい。同じような空間が農村部の中心部にもある。**



4242 (4) $\chi^2 = 0.001$



4952 JOURNAL OF CLIMATE



502 JOURNAL OF DOCUMENTATION



```
start, end, ramp length[m], exiting length[m]
```



そこで、無終端語の終端語の距離を求め、*evaluation*の上位のものが発生確率が最も高い終端語の距離が一定数を超えれば中途的な無終端語候補を取り除くと、無終端語の数を減らす。

[illegible]

2.	α_1	α_2	0.718713388934217	0.8673121033959902	0.26999421
3.	α_1	α_2	0.8709989326221138	0.816681491365064703	0.27000579
4.	α_1	α_2	0.8230155712053573	0.94068150254509008	0.23000421
5.	α_1	α_2	0.8773121033959902	0.8673121033959902	0.26999421



空間としても個人生活空間も形成され、それを核として様々な生活空間が形成され、十字の通路によって形成されていく集居生活空間の組織性を、個人生活空間においても実践したと言える。

Introduction of the building project you selected

Phase0 - 各自のシステムによって別解を与えることを前提に、既存の建築プロジェクトを一つ取り上げる

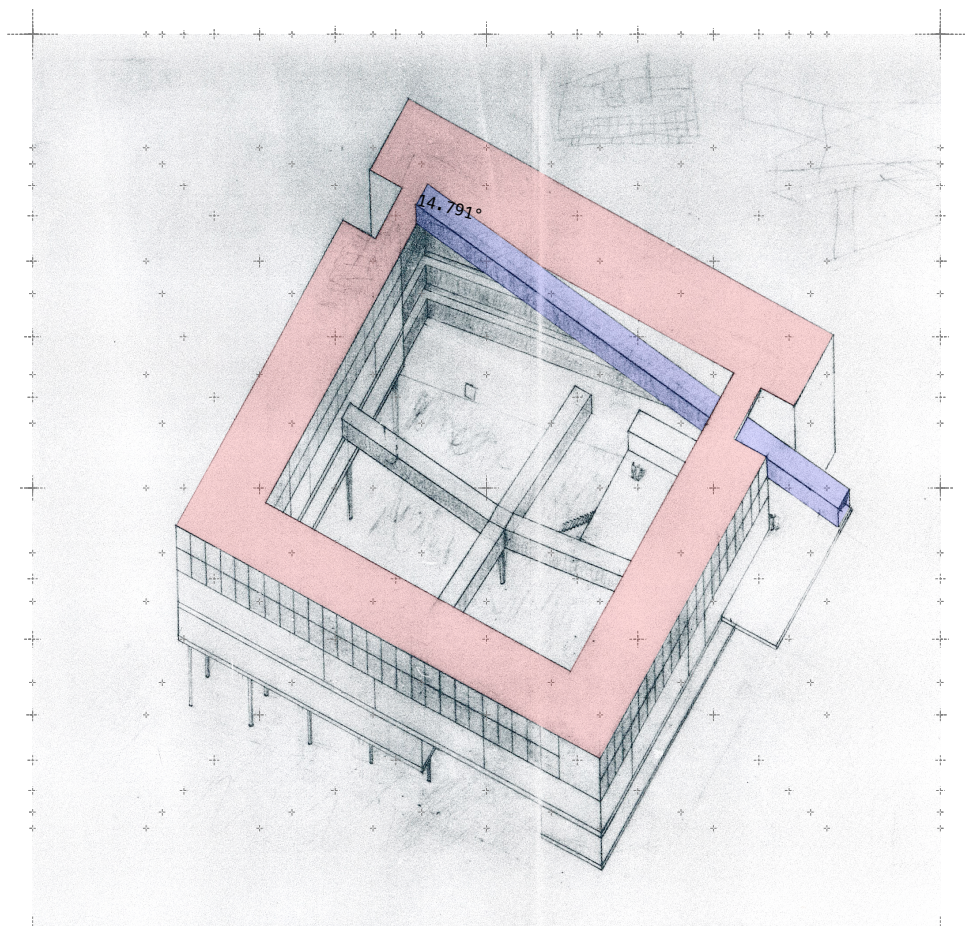
クセナキスの初期のスケッチには屋上へ向かう斜路と、屋上回廊が構想されている。しかし実際には斜路は実現されず、屋上も一周できない。
本プロジェクトの目標は、斜路を増築することで、修道院の回廊性を増幅させることである。

(cf. what is cloister of the monastery)

・修道院における回廊

ラ・トゥーレット修道院には、修道院特有の回廊がその地形の傾斜ゆえに成立せず、ル・コルビュジェは十字の通路をもって回廊の役割を補完したと言われている。

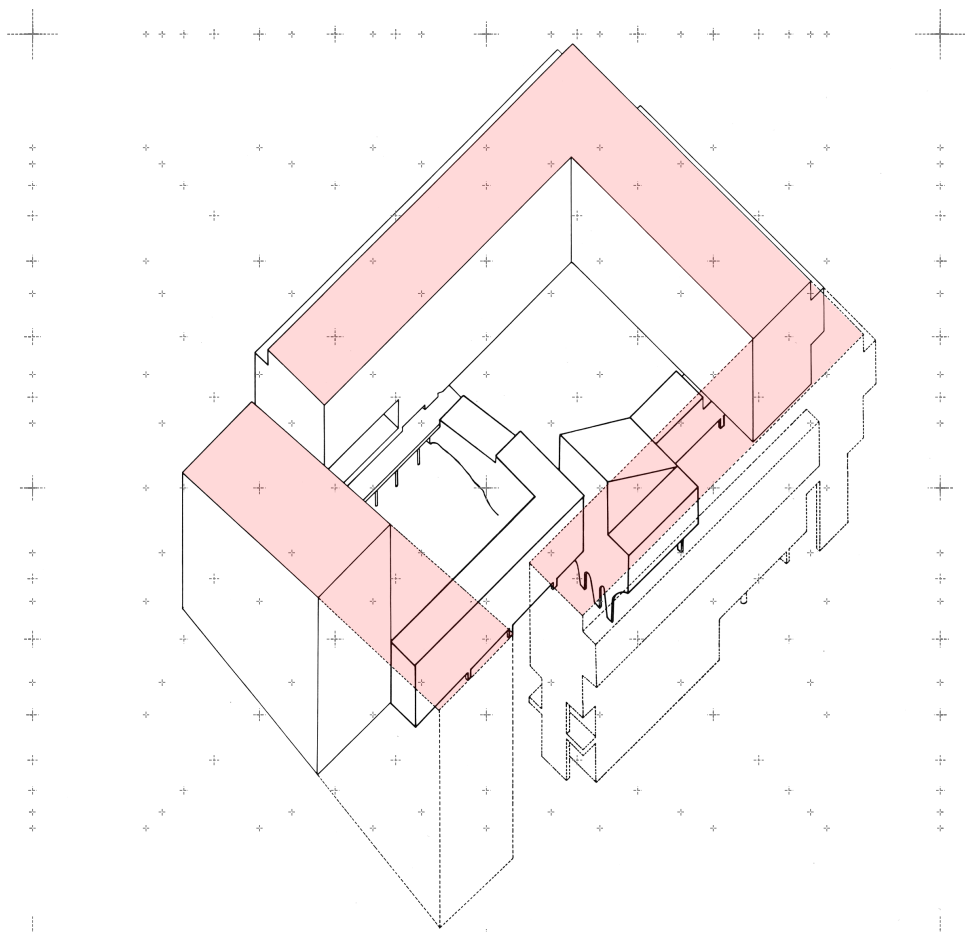
回廊は教会と僧房をつなぐ動線でもあり、儀式のための空間でもあり、個人の瞑想の場でもあるような、複合的な要素を兼ね備えた、修道院には欠かすことのできない空間である。それゆえ、-1Fの十字の通路は集団生活を豊かにしているとは言えるが、同ような通路が個人生活のためにもあってよいし、それは回廊性の増幅のためにも必要だと考える。



Axonométrie du couvent montrant rampe avec silhouettes et croquis, cotes
•前提 - 構想

クセナキスの初期スケッチには屋上へと向かう斜路と、回廊としての屋上が描かれている。

Iannis Xenakis / 30-3-54 / Crayon noir et de couleur / Calque épais / 0,60 x 0,44 / 1.244
// La Tourette and other buildings and projects, 1955-1957 / Le Corbusier. -- 1984
28.[The Le Corbusier archive / [Le Corbusier]; H. Allen Brooks, general editor } ,p518.

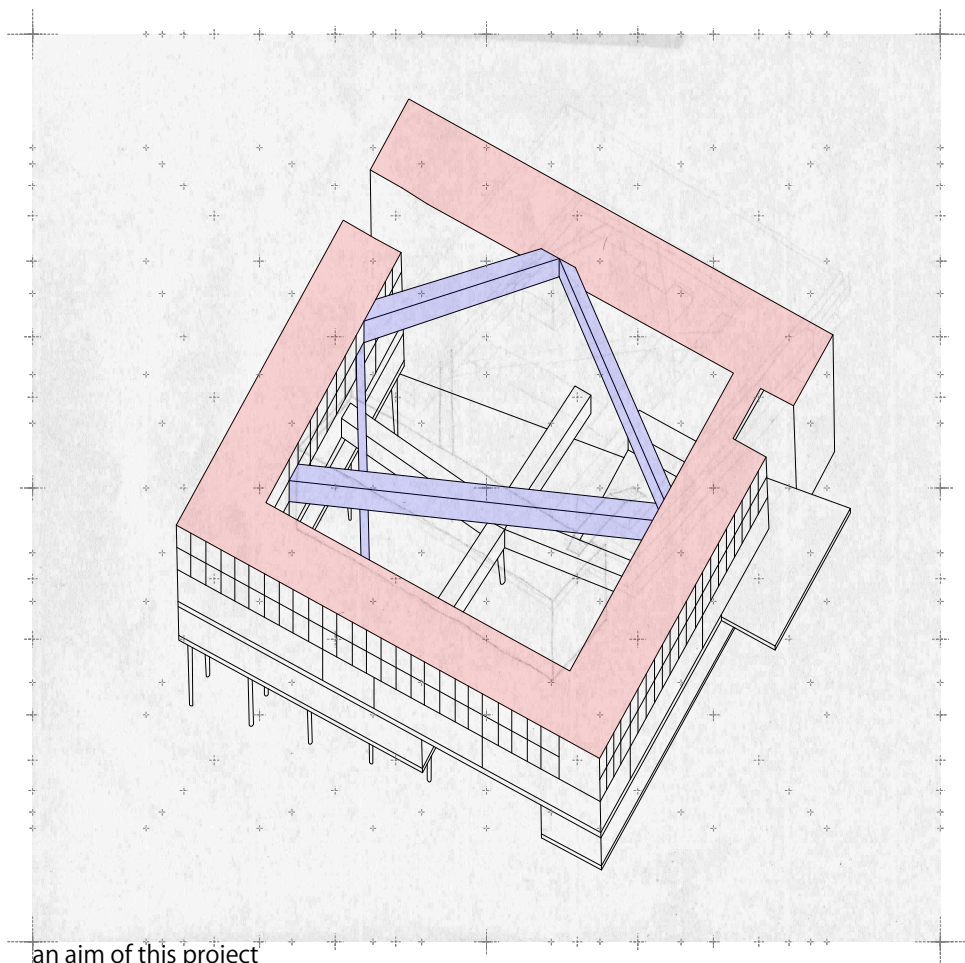


Axonométrie du couvent

•前提 - 現状

しかし実際には、斜路も屋上回廊も実現されていない。

neither ramp nor cloister was realised.



an aim of this project

・目標

ラ・トゥーレットの修道院に斜路を設け、修道院の回廊性を増幅させるのが本プロジェクトの目標である。

generate Ramps for Alternative Cloister

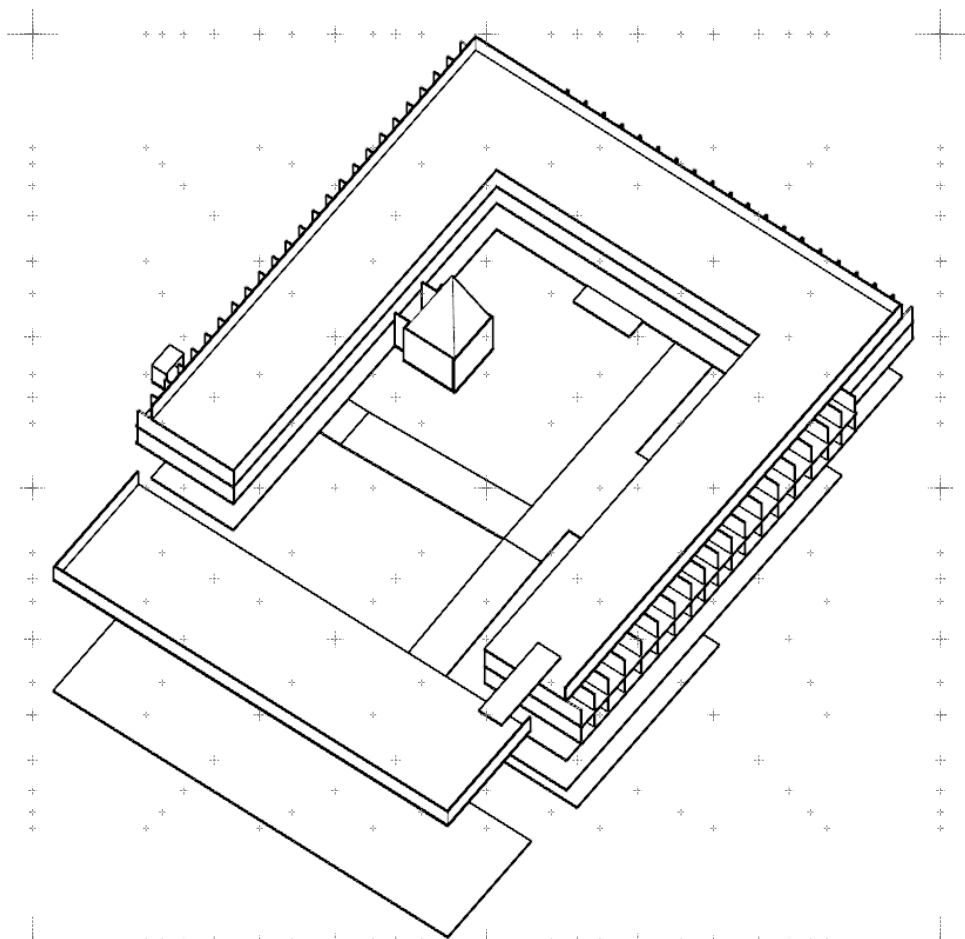
Design your data structure which will fit your design operation

Phase1 - プロジェクトの要素を分解し、どのように数値化情報化できるか、データ構造を設計する

与条件を大きく四種類に分け、整理した。

- 1 - 現状のラ・トゥーレットは中庭側に構造壁はなく、斜路の接続は可能。
- 2 - 柱の中間点を接続候補として116箇所プロットし、情報を埋め込む。
- 3 - 部屋同士の相性を12点満点で採点し、表にする。
- 4 - 現状の動線をエッジとし、グラフ理論を用いて特定の接続候補からの全ての接続候補への最短経路をダイクストラ法を用いて求める。

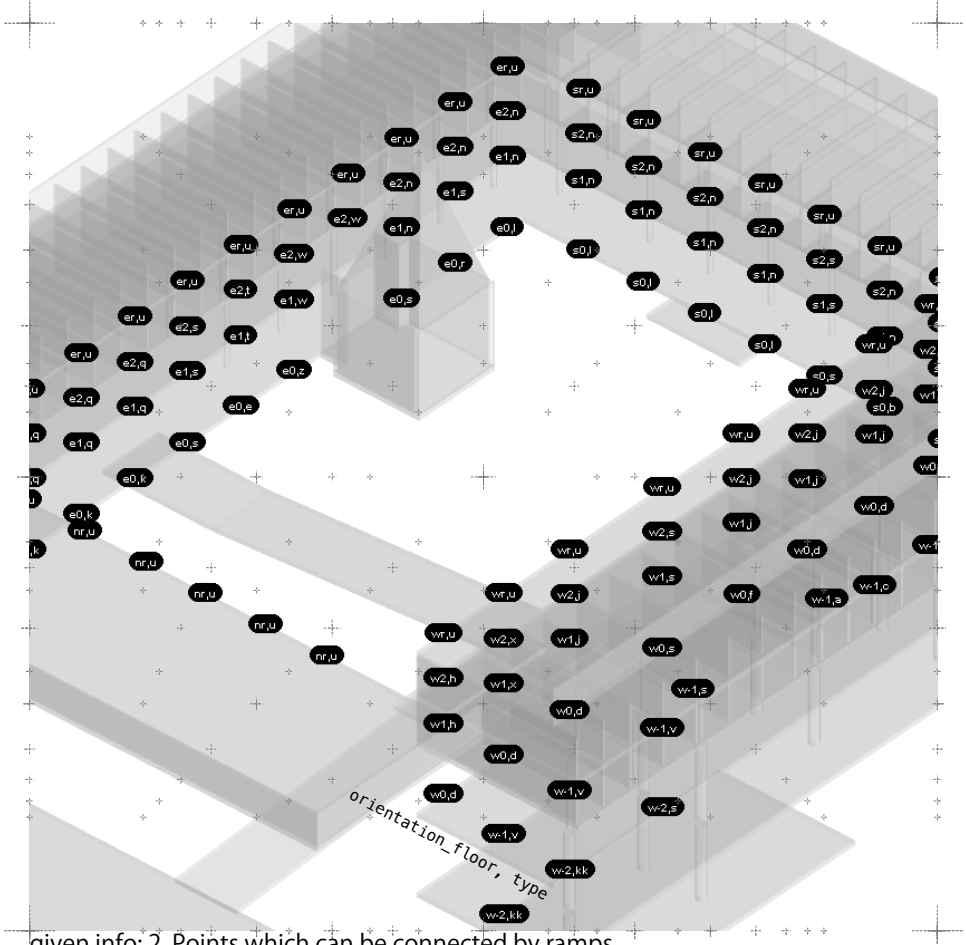
ここから部屋同士の相性と距離を用いて評価を行うようなデータ構造を考えた。



given info: 1. Floor surfaces and walls

・スラブと構造壁

中庭側に構造壁はなく、斜路の接続は可能である。



given info: 2. Points which can be connected by ramps

・接続可能な箇所

中庭側には構造壁の2スパン分毎に柱が立っていることを踏まえて、屋上階を含む各フロアにおいて、柱の中間点にエージェントとしてのノードを設定する。116個のノードには位置情報・方角・隣接する部屋の機能が情報として埋め込まれている。

node = 116 points (position,orientation_floor,type)

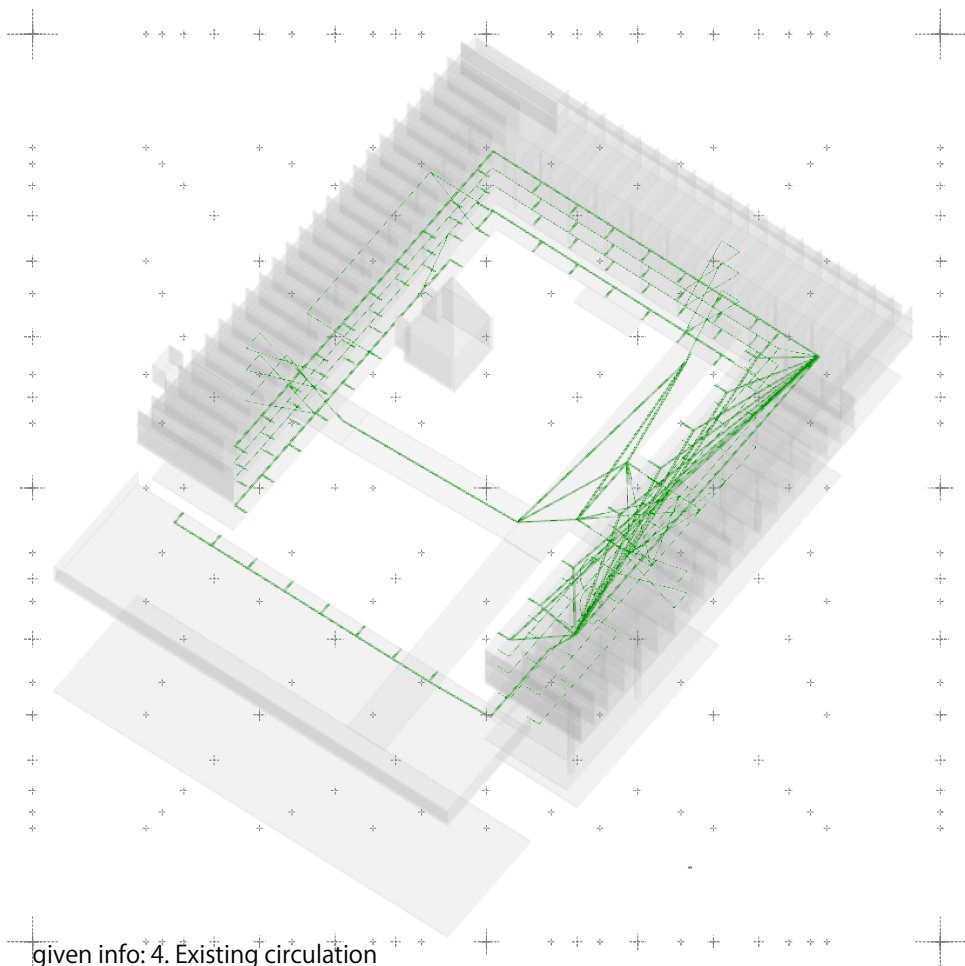
Floor			1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2	other	roof top
No.	Name	Individual Space____	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		Common Space____	unit (people)	h	i	g	j	m	w	n	q	t	x	y	s	u	
-1	1 Church	1	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-1	2 Refectory	1	v	3	3	3	6	6	4	4	4	4	0	4 1/2	4 2/3	6	8
-1	3 Chapter's Room	1	c	0	3	3	6	6	0	0	0	0	0	4 1/2	2	6	6
-1	4 Atrium	1	a	3	3	3	6	6	4	4	4	4	0	4 1/2	4 2/3	6	8
-1	5 Conduit	1	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-1	6 Conduit	1	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0	1 Workroom S	2	b	3	0	0	2	3	4	4	4	4	0	1 3/2	2/3	6	6
0	1 Workroom L	2	d	3	0	0	2	3	4	4	4	4	0	1 3/2	2/3	6	6
0	2 Father's Room	1	f	0	0	0	12	12	3	4	2	0	6	6 1/3	6	6	6
0	3 Student Priest's Room	1	p	0	0	0	3	3	12	4	3	0	1 1/2	6 1/3	6	6	6
0	4 Laybrother's Room	1	e	0	0	0	3	3	3	3	12	0	1 1/2	3	6	6	6
0	5 Library	1	l	3	3	3	4	4	5	5	3	0	3 1/2	4 2/3	6	6	6
0	6 Reading Room	1	r	3	3	3	4	4	5	5	3	0	3 1/2	4 2/3	6	6	6
0	7 Oratory	1	o	3	3	3	6	6	3	3	3	0	4 1/2	4	6	6	6
0	8 Visitor's Cell	5	k	0	0	12	0	0	0	0	0	0	0	0	0	6	6
0	evr	1	z	1 1/2	1 1/2	1 1/2	3 1/2	3 1/2	4	4	7 1/2	0	2 1/2	3 5/6	6	6	6
		Individual Space____		h	i	g	j	m	w	n	q	t	x	y	s	u	
1,2	1 Cell for Patient	4	h	3	12	0	3	2	1	1	1	4	7 1/2	1 1/3	6	4	4
1,2	2 Infirmary	2	i	12	3	0	3	2	0	0	0	4	3	2/3	6	4	4
1,2	3 Cell for Guest	4	g	0	0	3	1	1	0	0	0	4	1/2	3	6	8	8
1,2	4 Cell for Professor Father	22	j	3	3	1	3	2	0	0	0	4	3	2/3	6	8	8
1,2	5 Cell for Father as Master Student	4	m	2	2	1	2	3	0	0	0	4	2	1	6	8	8
1,2	6 Cell for Student Priest	16	w	1	0	0	0	0	3	2	2	4	0	1 2/3	6	8	8
1,2	7 Cell for Student Friar	46	n	1	0	0	0	0	2	3	2	4	0	1 2/3	6	8	8
1,2	8 Cell for Laybrother	12	q	1	0	0	0	0	2	2	3	4	0	1 1/3	6	8	8
1,2	9 Sanitary	4	t	4	4	4	4	4	4	4	4	3	4	4	6	6	6
1,2	10 i+j	2	x	7 1/2	3	1/2	3	2	0	0	0	4	3	2/3	6	6	6
1,2	11 n+m+w	2	y	1 1/3	2/3	1/3	2/3	1	1 2/3	1 2/3	1 1/3	4	2/3	3	6	8	8
other	Stairs	4	s	6	6	6	6	6	6	6	6	6	6	6	3	5	5
rf	Alternative Cloister	33	u	4	4	8	8	8	8	8	8	6	6	8	5	3	3

given info: 3. Score List for combination of functions
・スコアリスト

ラ・トゥーレット修道院の形態形成の基本となる機能は大きく、

個人生活(Individual Space) / 集団生活(Common Space) / 精神生活

というように分類できる。特に個人生活と集団生活の空間はフロアによって明快に分割されているので、ここでは個人と集団、あるいは個人と個人の空間が機能的に接続されるように、それぞれの部屋の相性を機能から読み解き、12点満点で採点し、表を作成した。

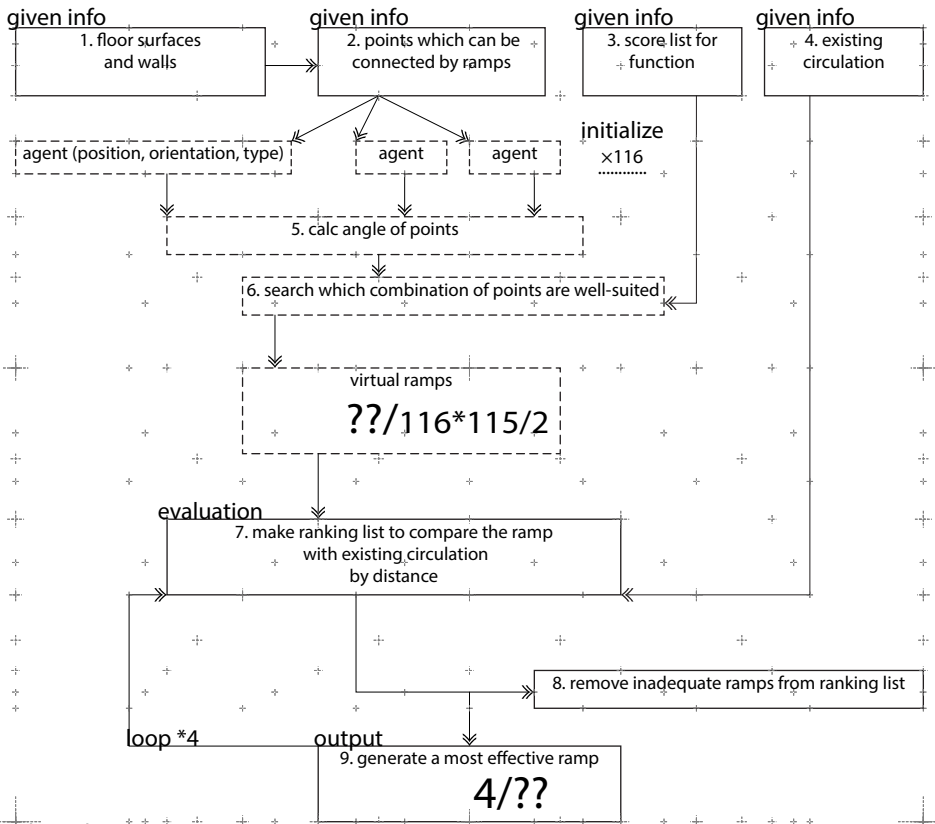


given info: 4. Existing circulation

•現状の動線

ラ・トゥーレット修道院の現状の動線をエッジとし、グラフ理論を用いて最短経路を求める。本プロジェクトでは、ダイクストラ法を利用して、特定の1つのノードから他の全ノードまでの最短経路を求め、その全長を評価に用いる。

pseudocode



Flow chart

・フローチャート

本プロジェクトで実装したシステムは以下の通りである。

現状の修道院を与件とし、そこから条件を設定する(given info)。斜路が接続可能な点をエージェントとし(initialize)、物理的な制約・部屋の機能から接続候補を選択(5. 6.)、接続候補を既存の通路と比較し、評価する(7. 8.)ことで、ラ・トゥーレット修道院に最適な、4本の斜路(9.)を考える。

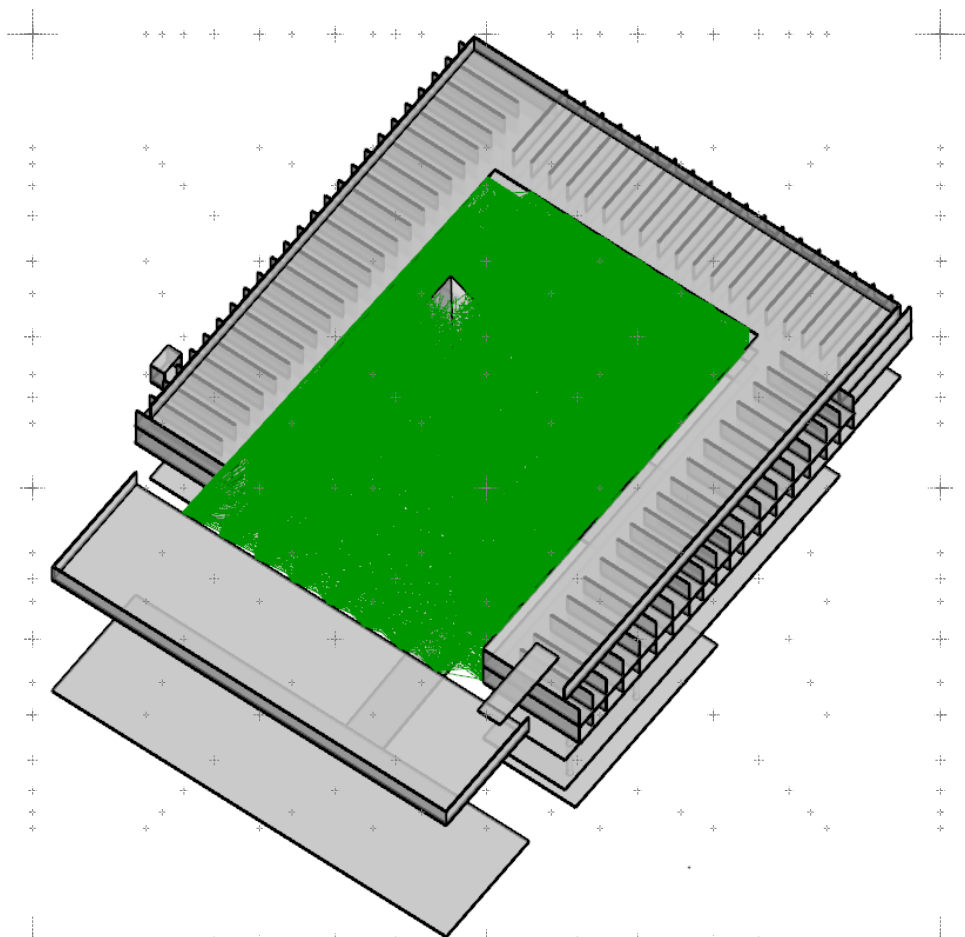
Design your evaluation method and formula

Phase2 - 評価方法、評価指標、評価関数を検討、デザインする

具体的なコード内では、接続可能な点をエージェントとし、物理的な制約・部屋の機能から接続候補を選択し、それぞれの組合せを線分で繋ぐことで、エッジを生成しようとしている。すると、特定の1つのノードから他のノードまでの最短経路は変化する。

そこで、それぞれのエッジに対応する最短経路がどの程度短くなったかを評価の指標とするために、評価関数を検討した。

また、現状の動線の評価、及びスコアリストと評価関数の関係など、再考すべき箇所も存在する。

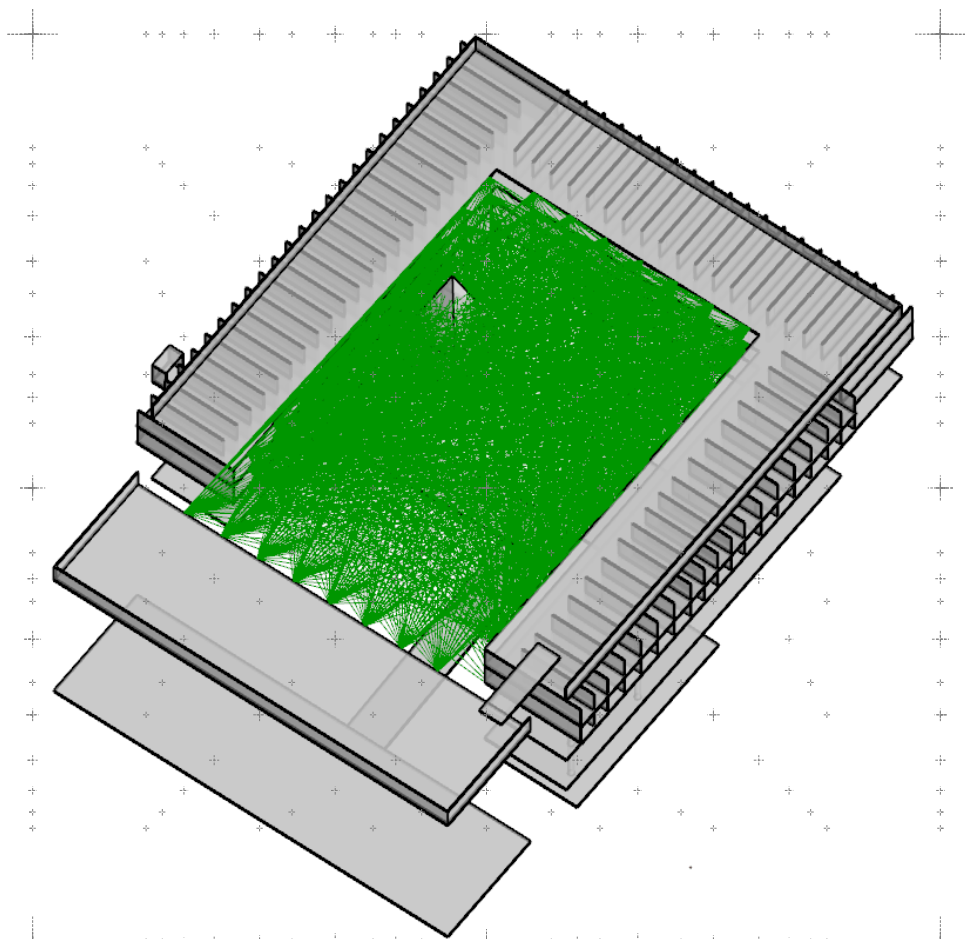


(cf. all the points connected)

•全ての組合せ

同フロア・同側面には斜路を設けないようなノードの組合せは4796通りとなるが、この中には物理的に斜路とはならない候補も含まれる。

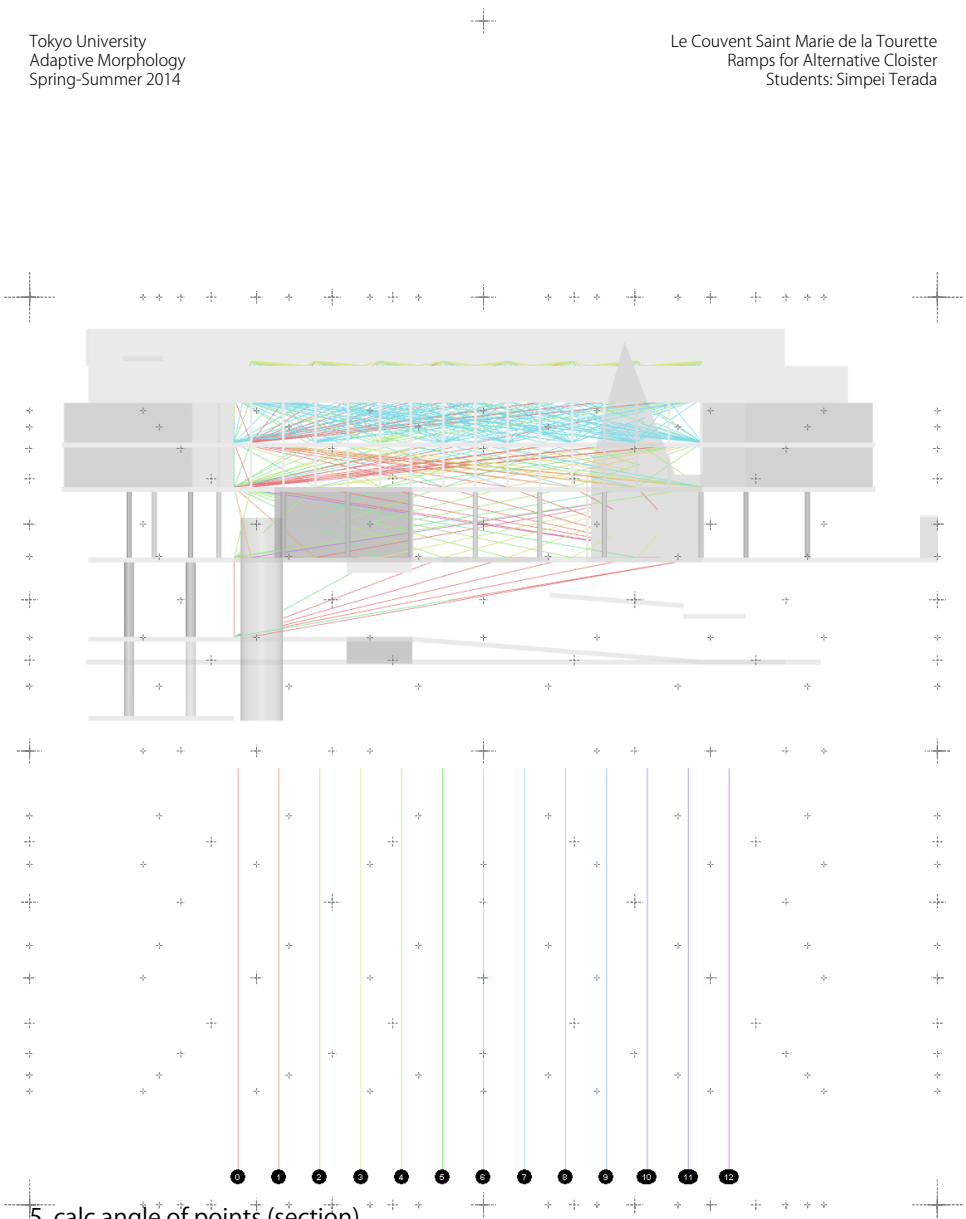
combinations = 4796



5. calc angle of points
•角度の制約

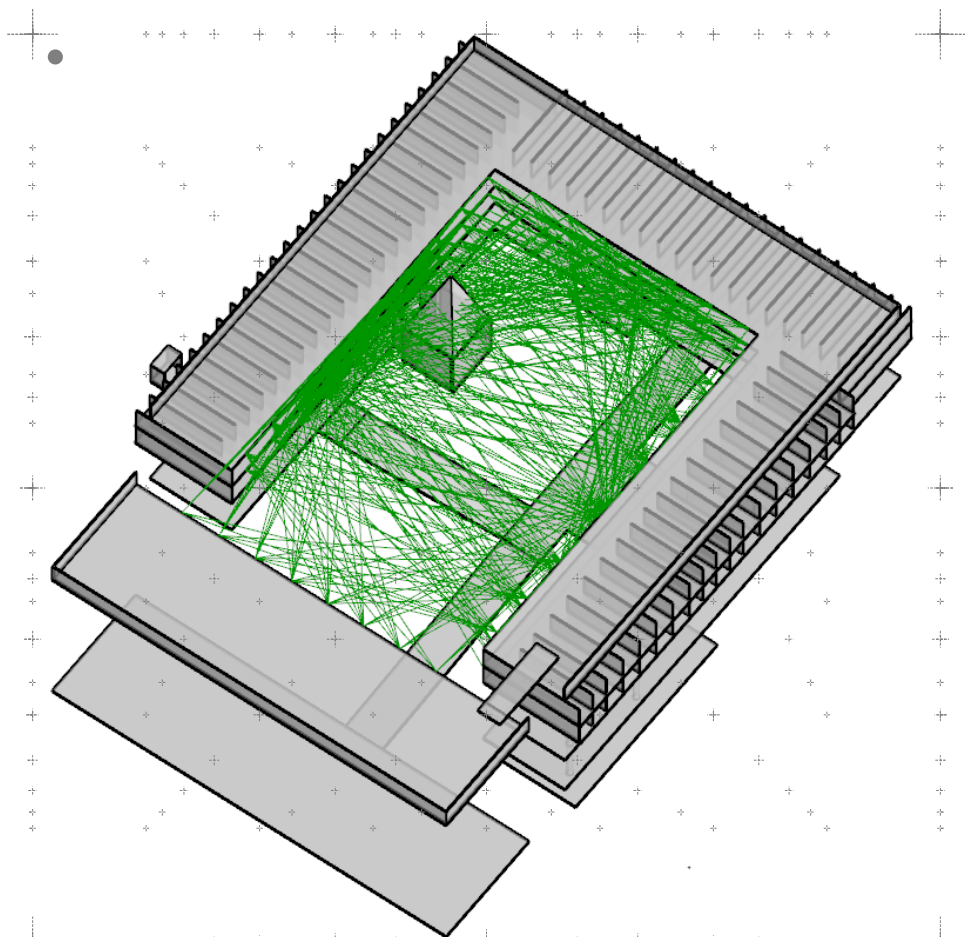
斜路として適切な角度となる、ノード二点間の角度が 8° 以下の組合せは1343通りとなる。

combinations = 1343 (Angle ≤ 8.00)



5. calc angle of points (section)
•角度の制約(立面図)

combinations = 1342 (Angle <= 8.00)

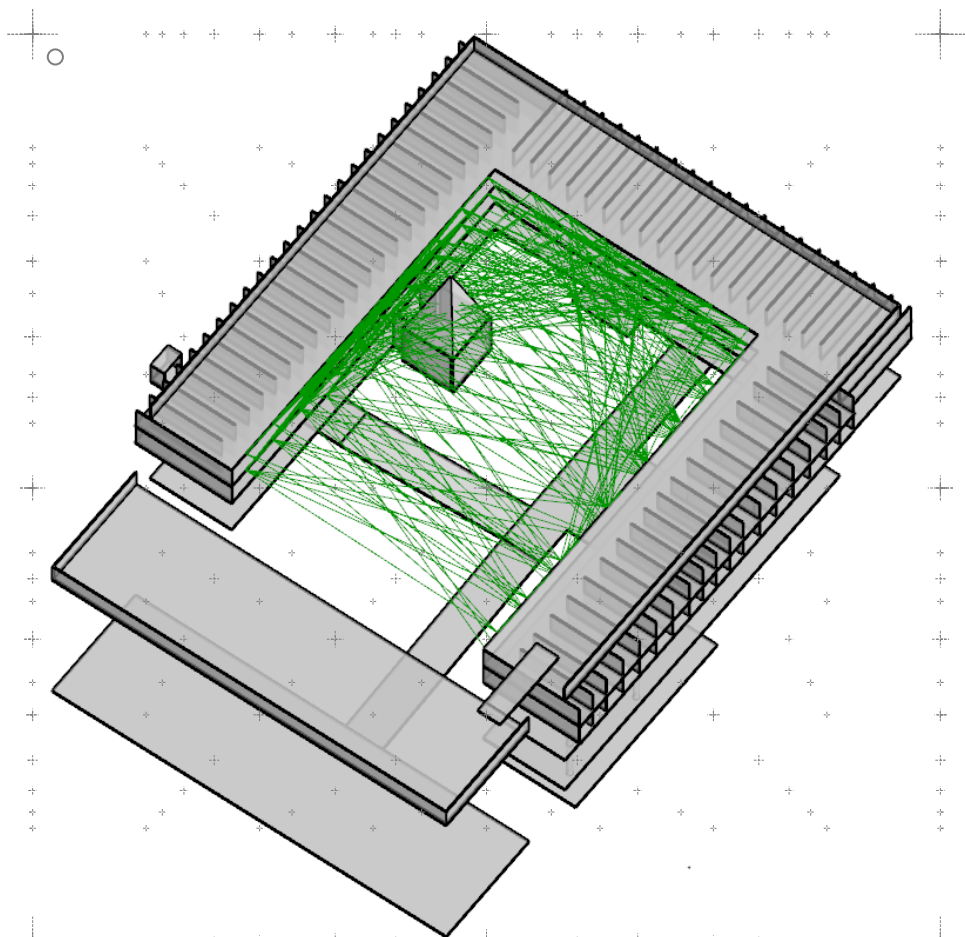


6-A. score combinations according to the Score List

・部屋同士の機能の相性

スコアリストに基づき、点数が2点以上になるノードの組合せ = 接続候補は1030通りとなる。

combinations = 1030 ($2 \leq \text{Score}$)

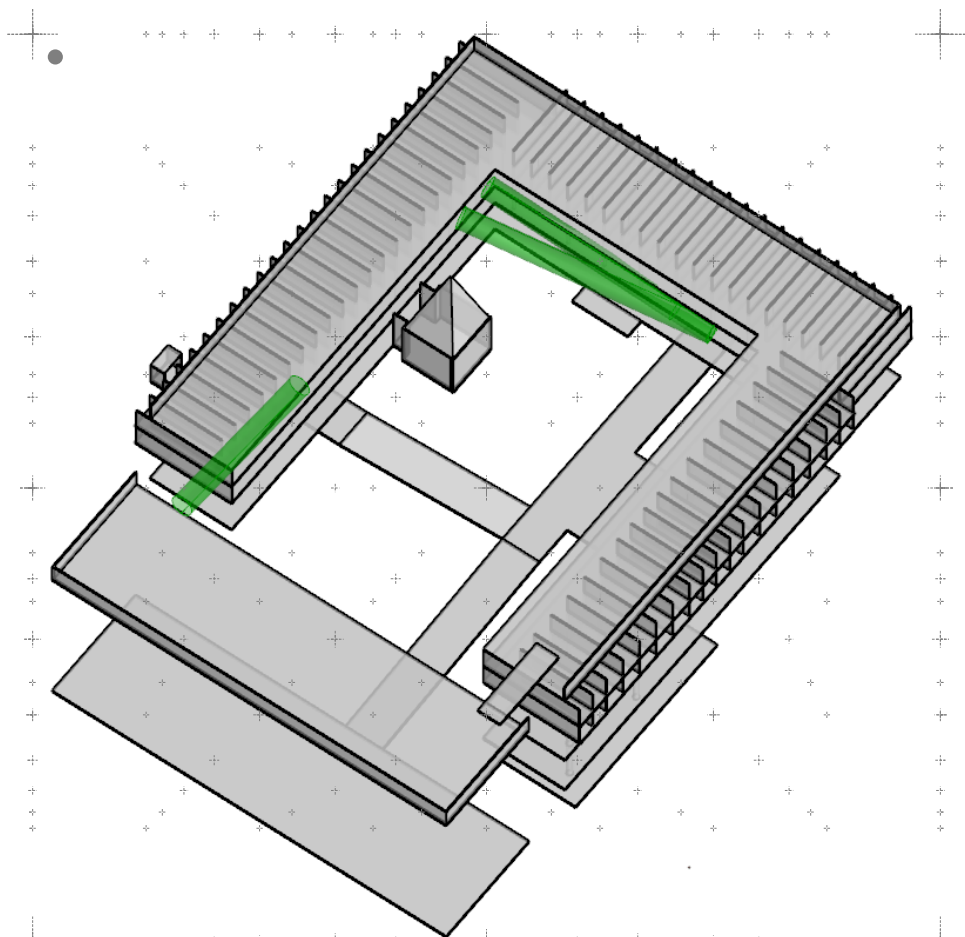


6-B. score combinations according to the Score List

•部屋同士の機能の相性

同様に、点数が6点以上になる接続候補は570通りとなる。

combinations = 570 (6 ≤ Score)



evaluation: 7-A. compare the ramp with existing circulation

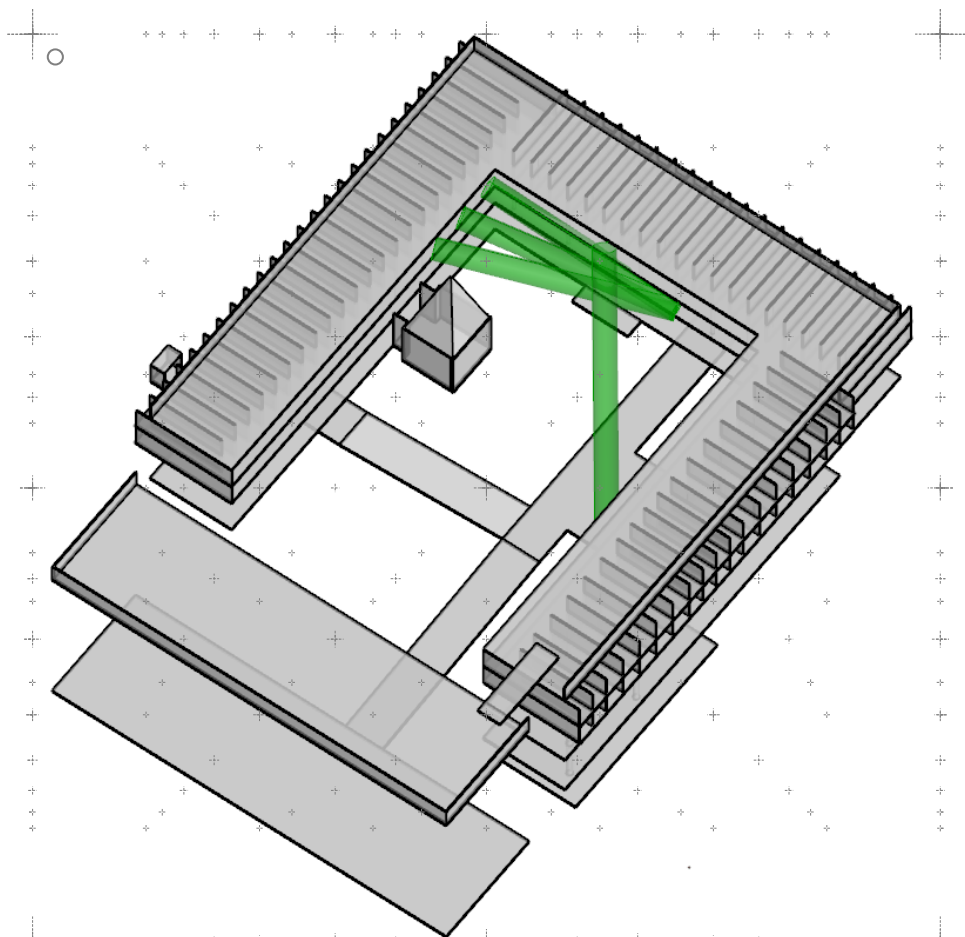
•斜路と現状の動線の比較

特定の1つの接続候補に対して、ノード二点間の長さを現状の動線の最短距離から減ずる式を評価関数とする。

__[evaluation value] = existing length - ramp length__

start, end, ramp length[mm], existing length[mm]

1. n, n, 22116.891592664975, 145552.01682469237
2. s, n, 18489.809093150256, 141892.01793580345
3. u, u, 16103.177580258423, 136611.58547895079
4. n, n, 22573.782479303703, 141892.01793580348



evaluation: 7-B. compare the ramp with existing circulation

•斜路と現状の動線の比較

評価関数から接続候補をランキング順にし、上位4本を生成した。

start, end, ramp length[mm], existing length[mm]

1. s, n, 18489.809093150256, 141892.01793580345
2. s, n, 19033.964432097848, 138232.01904691459
3. s, n, 25852.582606505777, 143875.78831362305
4. s, n, 20236.149521016243, 134572.0201580257

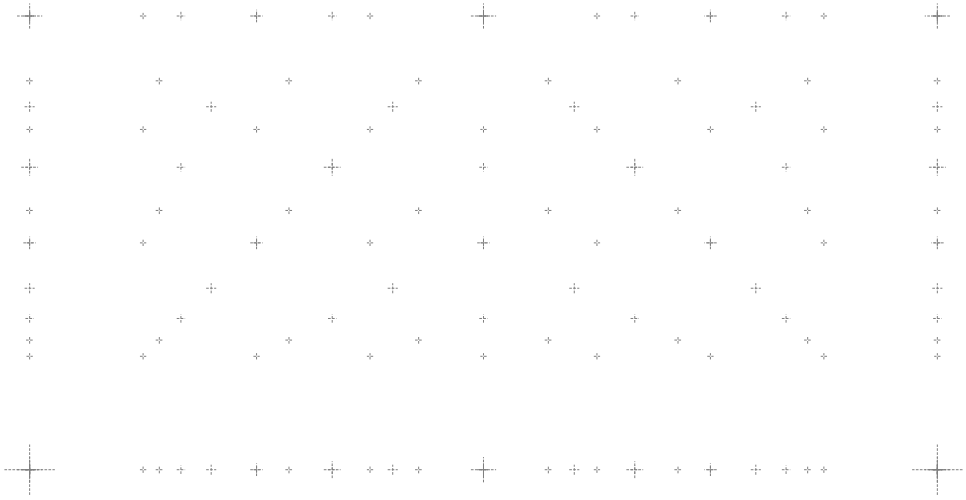
Design and implement your system which has loop cycle of operation and evaluation Phase3 - 評価とデザイン(設計操作)を抱き合わせたシステムの構築を目指し実装する

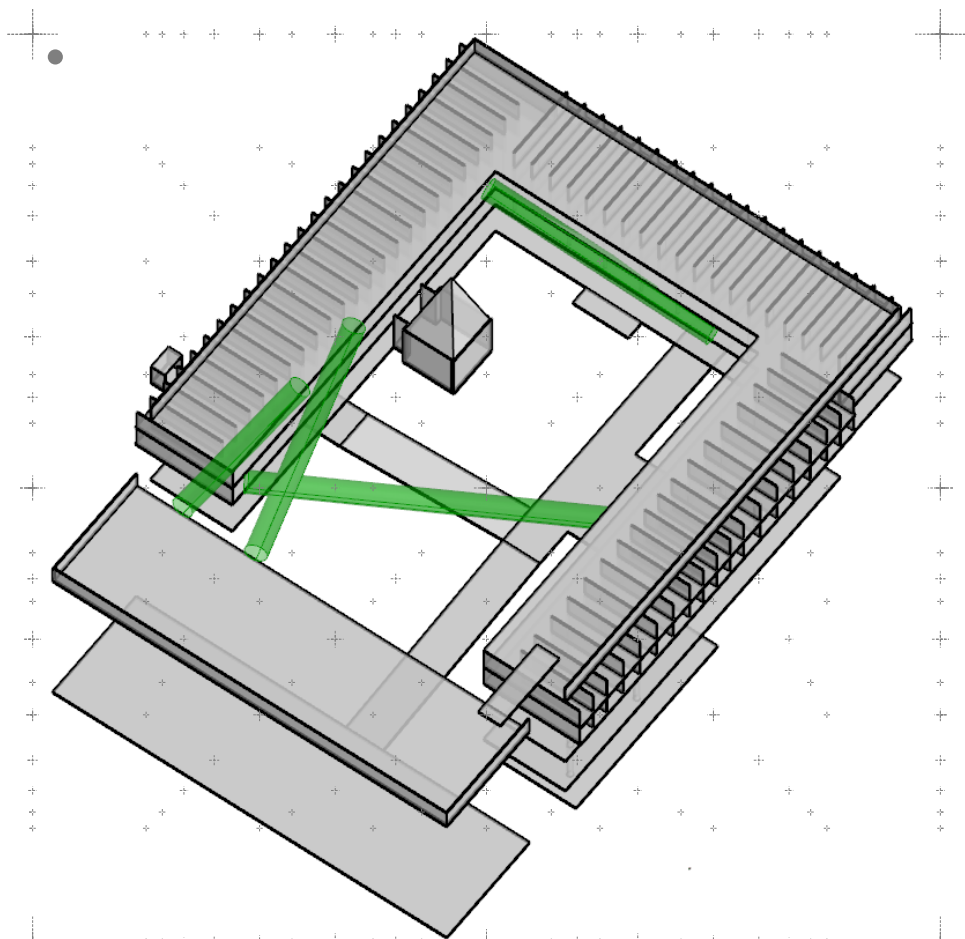
接続候補を制限した結果、接続候補としてのエッジは限られてくる。そこで今度はエッジをエージェントとして扱い、最適な斜路を見つけ出す。

これまでのプロセスでは、全ての接続候補に対して一様な制限を与えているため、似た情報をもつ接続候補から斜路が生成され、よって斜路がある範囲にかたまってしまっている。よって、上部に生成される斜路や、既存のボリュームが障害となり、通り抜けのできないような斜路もうまれている。

あるいは斜路を上位から生成していくと、現状の動線が変化し、評価値・ランキングも変化する。

以上のことを踏まえてシステムを実装する。





evaluation: 8-A,B. remove inadequate ramps

・不適切な斜路を取り除く

接続候補の線分間の距離を求め、evaluationの上位のものが生き残りながら線分間の距離が一定数を上回るように不適切な斜路接続候補を取り除くと、接続候補は8通り、または10通りとなる。

combinations = 8 (2 <= Score), 10 (6 <= Score)

```
list = []
for ed in self.intersect:
    line = rs.AddLine(ed.node_s.pos, ed.node_e.pos)
    isGoodEdge = True
    for edo in list:
        lineo = rs.Line(edo.node_s.pos, edo.node_e.pos)
        dist = rs.LineMinDistanceTo(line, lineo)
        if dist < 2000:
            isGoodEdge = False
            break
    if ed == edo:
        isGoodEdge = False
        break
    if isGoodEdge:
        list.append(ed)
        #create reference
        ref_e = Node_ref(ed.node_e, line)
        ref_s = Node_ref(ed.node_s, line)
        #add edge, new connection, rebuild network
        ed.node_s.connections.append(ref_e)
        ed.node_e.connections.append(ref_s)
        for edoo in self.intersect:
            gu.Dijkstra(edoo.node_s, self.agents)
            ex_len = edoo.node_e.cost
            #re-write the edge property info
            edoo.ex_len = ex_len
            edoo.len_ev = ex_len - edoo.new_len
        self.intersect.sort(sort_length)#re-sort at every step
        #0 wo list no saisyo ni mottekuru
    self.intersect = list

print "list of final candidate =", len(self.intersect)
for ed in self.intersect:
    print ed.len_ev, ed.node_s.id, ed.node_e.id

for n in range(min(int(fin_edgenum), len(self.intersect))):
    ed = self.intersect[n]
    self.fin_con.append(ed)
```

evaluation: 8. remove inadequate ramps

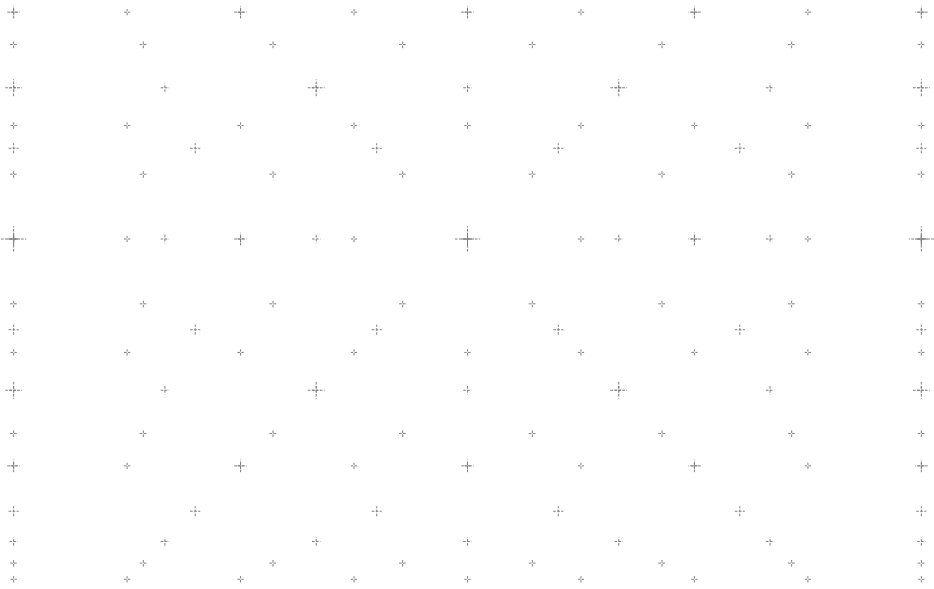
・不適切な斜路を取り除く

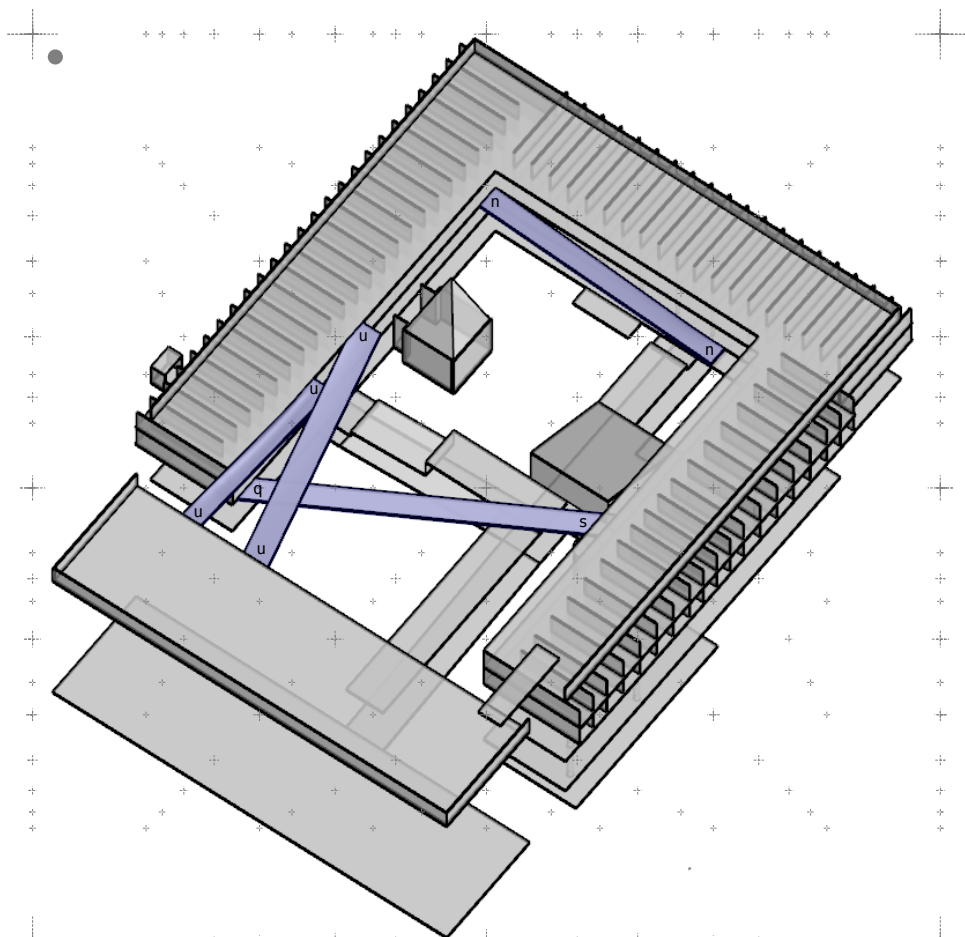
あるいは斜路を一本生成する毎に、ダイクストラ法を再計算し、ランキングを変化させるコードを考える。(システム未実装)

Presentation your alternative solution(proposal)

Phase4 - 別解の提示

最終候補から4本の斜路を生成した別解はそれぞれ、屋上の一週や、僧房の空間から屋上へのシークエンスなどを実現している。部屋の相性と距離のみで評価したにもかかわらず、回廊性を増幅させる結果を得られことは興味深く、またル・コルビュジェによる十字の通路が現状とは全く異なる要素に見えてくる。





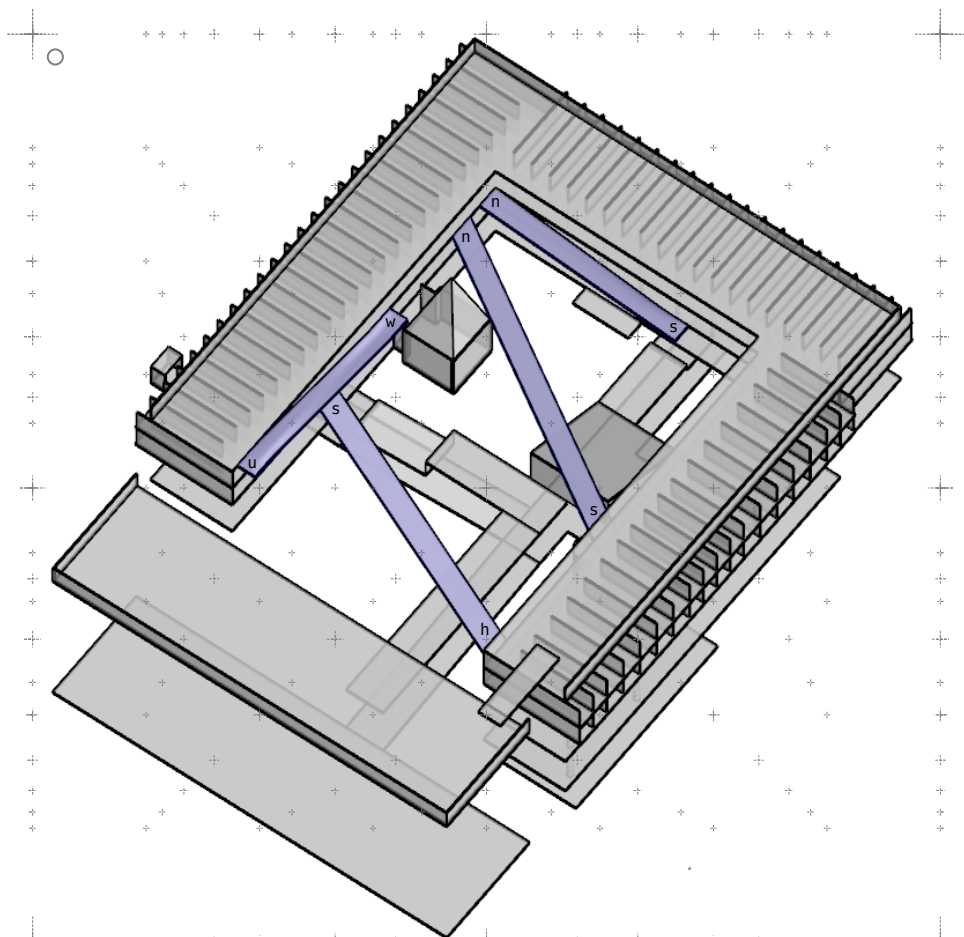
output: 9-A. generate most effective ramps

・最適な斜路の生成

4本の斜路はそれぞれ、学生修道士用個室同士(3点)、屋上同士(3点)、無品級修道士用個室と階段(6点)を繋ぎ、結果として屋上を一周することが可能にしている。

start, end, angle[deg], ramp length[mm], existing length[mm], [evaluation value]

1. n, n, 6.568569623696629, 22116.891592664975, 145552.01682469237, [123435.125232]
2. u, u, 7.493230058574078, 16103.177580258423, 136611.58547895079, [120508.407899]
3. u, u, 4.917945388245219, 24495.800495381733, 121900.76819419311, [97404.9676988]
4. q, s, 4.861098001226742, 29855.883304079842, 119639.06497187962, [89783.1816678]
5. u, u, 3.696708450854082, 32570.774207446968, 110920.75894495856, [78349.9847375]
6. s, n, 6.858628412376028, 21185.736911390304, 98421.7934777734, [77236.0565664]
7. s, u, 7.864577364155707, 18489.80726259412, 51670.721626012149, [33180.9143634]
8. g, u, 7.737701520932276, 18791.095470106779, 32279.529303823812, [13488.4338337]

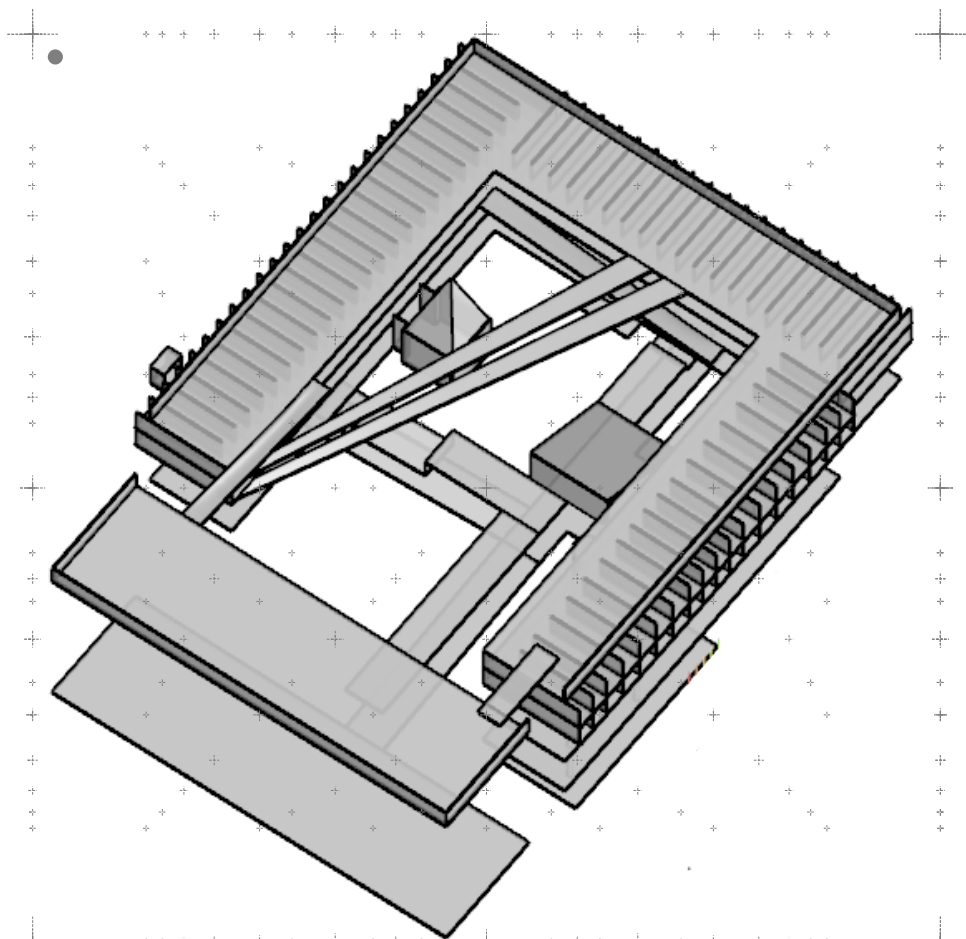


output: 9-B. generate most effective ramps

・最適な斜路の生成

個人生活の空間を經由して屋上に到達できるような斜路が生成され、十字の通路に実現されている集団生活空間の回廊性を、個人生活空間においても実現したと言える。

1. s, n, 7.864576580607733, 18489.809093150256, 141892.01793580345, [123402.208843]
2. w, u, 7.87134085616219, 18474.019743999062, 120902.05523724943, [102428.035493]
3. s, n, 4.670998526201138, 31068.093165064703, 125804.6408134969, [94736.5476484]
4. s, h, 4.920135571818573, 29498.502545649888, 123619.03330521291, [94120.5307596]
5. q, u, 5.429712485613837, 26737.238924789592, 108065.40858012867, [81328.1696553]
6. n, u, 7.335401080632735, 36263.257962265066, 111699.44321053327, [75436.1852483]
7. t, s, 5.433179465968324, 26720.228676478429, 88211.106531542857, [61490.8778551]
8. s, j, 5.311046001908349, 27332.868245415291, 63561.168193587815, [36228.2999482]
9. j, u, 5.066380365373455, 28649.131851281858, 62081.3308323858, [33432.1989811]
10. n, u, 6.568570080202787, 22116.890062312814, 48010.722737123258, [25893.8326748]

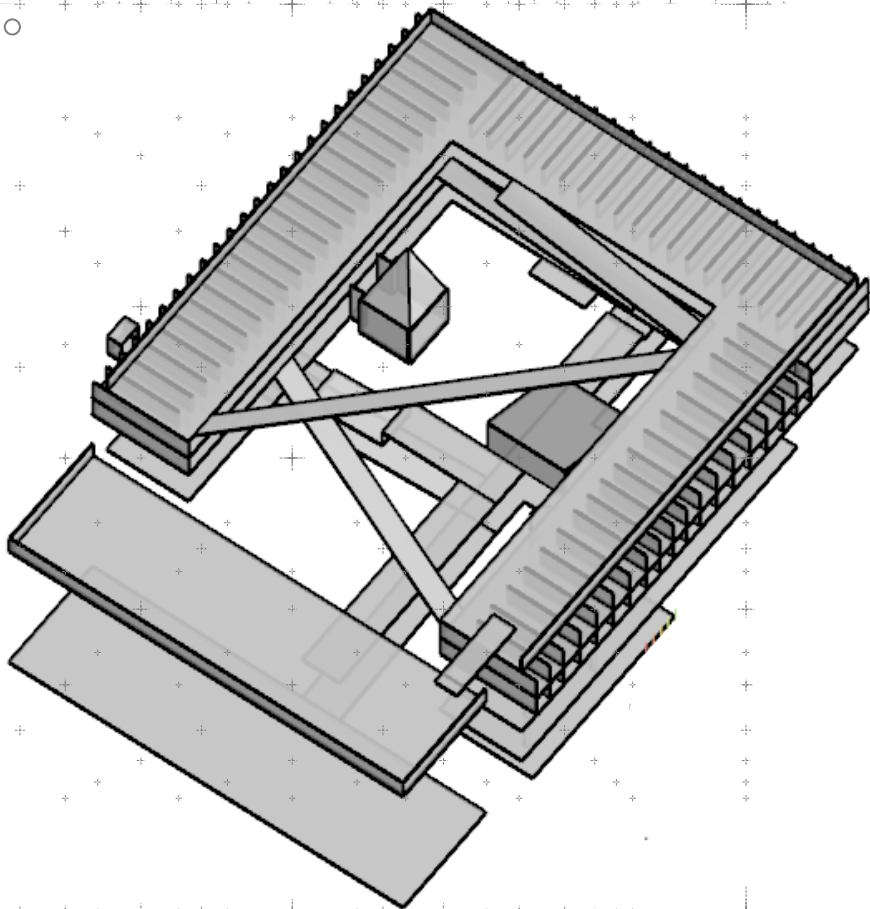


output: 10-A. generate most effective ramps

・最適な斜路の生成(ダイクストラ法を再計算する場合)

システムとして実装できていないが、ダイクストラ法を再計算し、最短経路を逐一再建した結果である。

ramps = 4 (2 <= Score)



output: 10-B. generate most effective ramps

・最適な斜路の生成(ダイクストラ法を再計算する場合)

システムとして実装できていないが、ダイクストラ法を再計算し、最短経路を逐一再建した結果である。

ramps = 4 (6 <= Score)