WHAT WE GOT AFTER OUR 800TH COFFEE

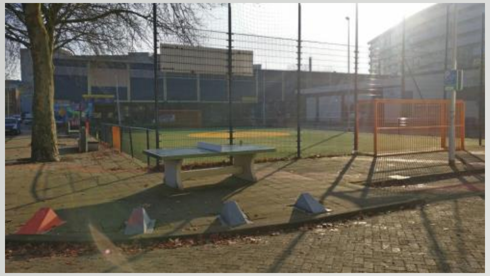
Bram vd Berg || 4687949

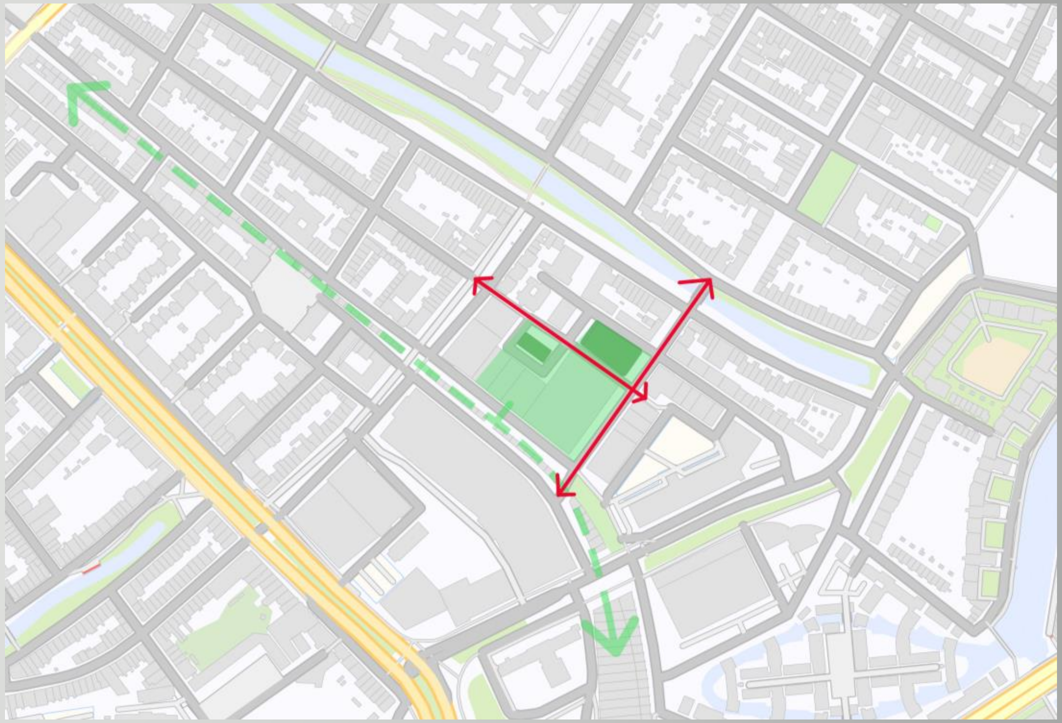
Jordy van Eijk || 4566297

LAY-OUT

- Pitch
- Recap Midterm
- Flowchart
- Code
  - Stacking
  - Don’t float
  - Greenroofs
  - Elevatorshafts
  - Corridors
  - Facade implemention
- Progress Visualisation

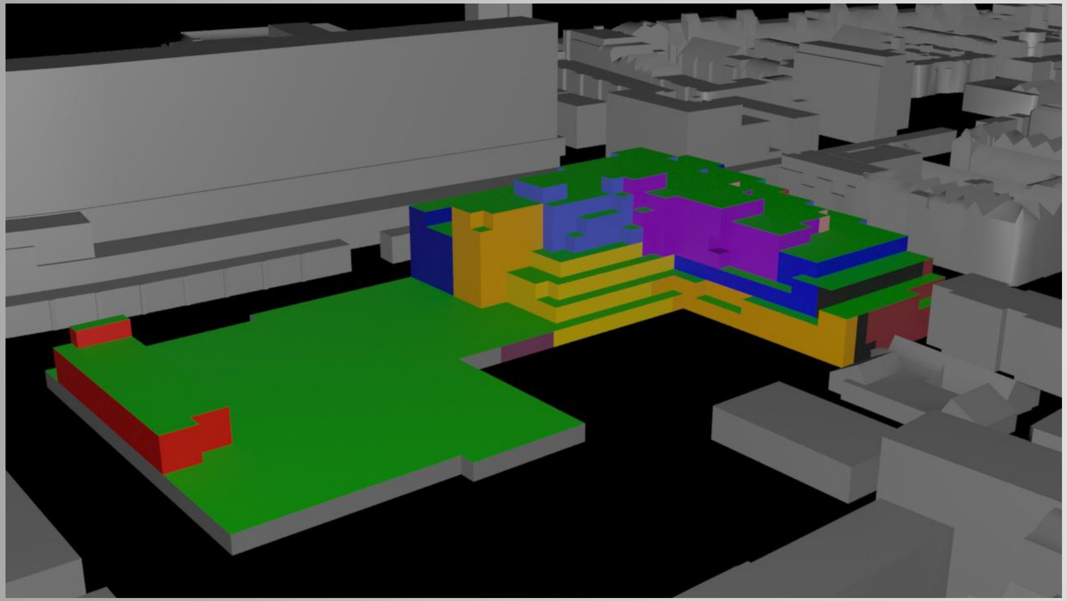
PITCH-SITUATION

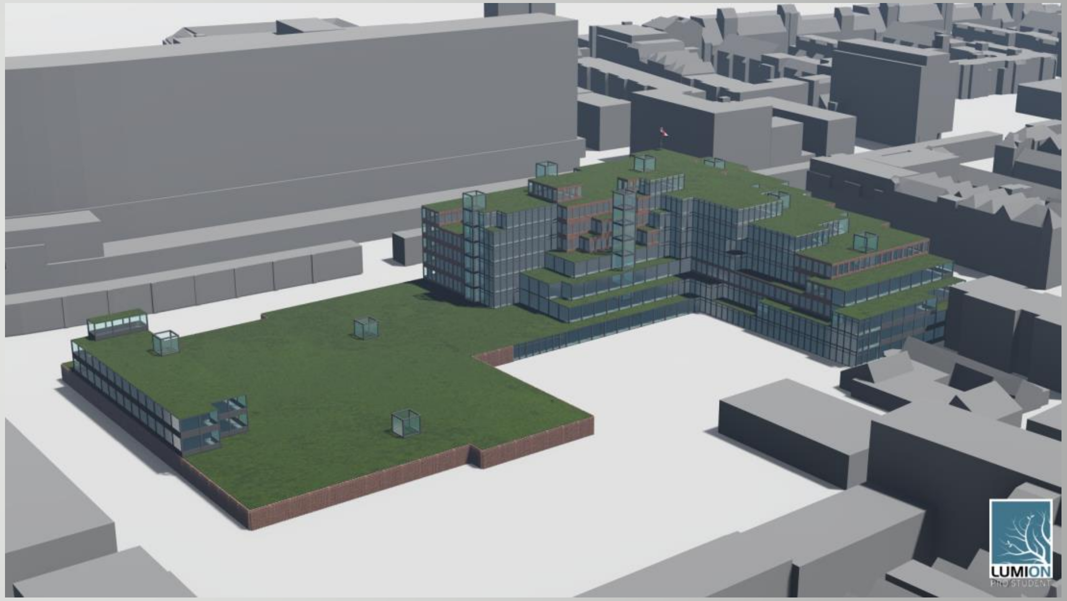




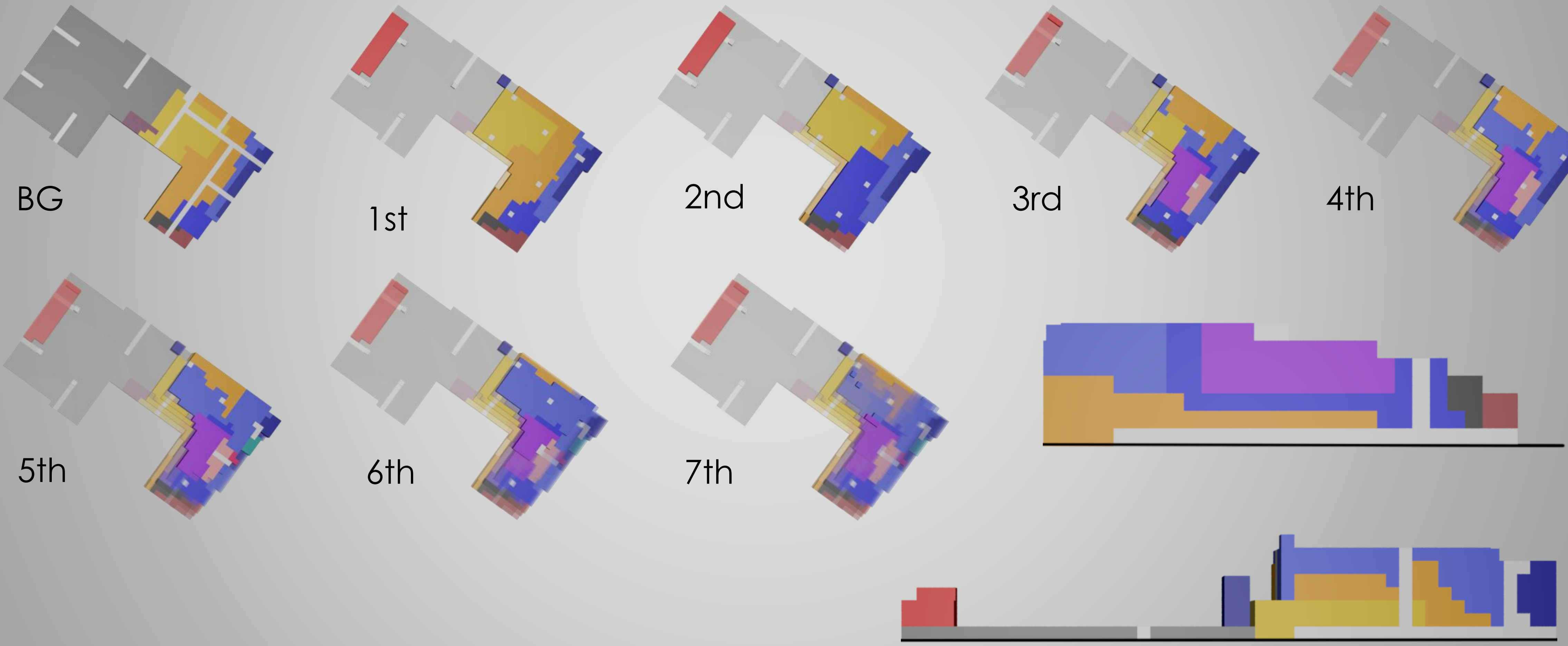
PITCH-REFERENCES

PITCH-OUR SOLUTION

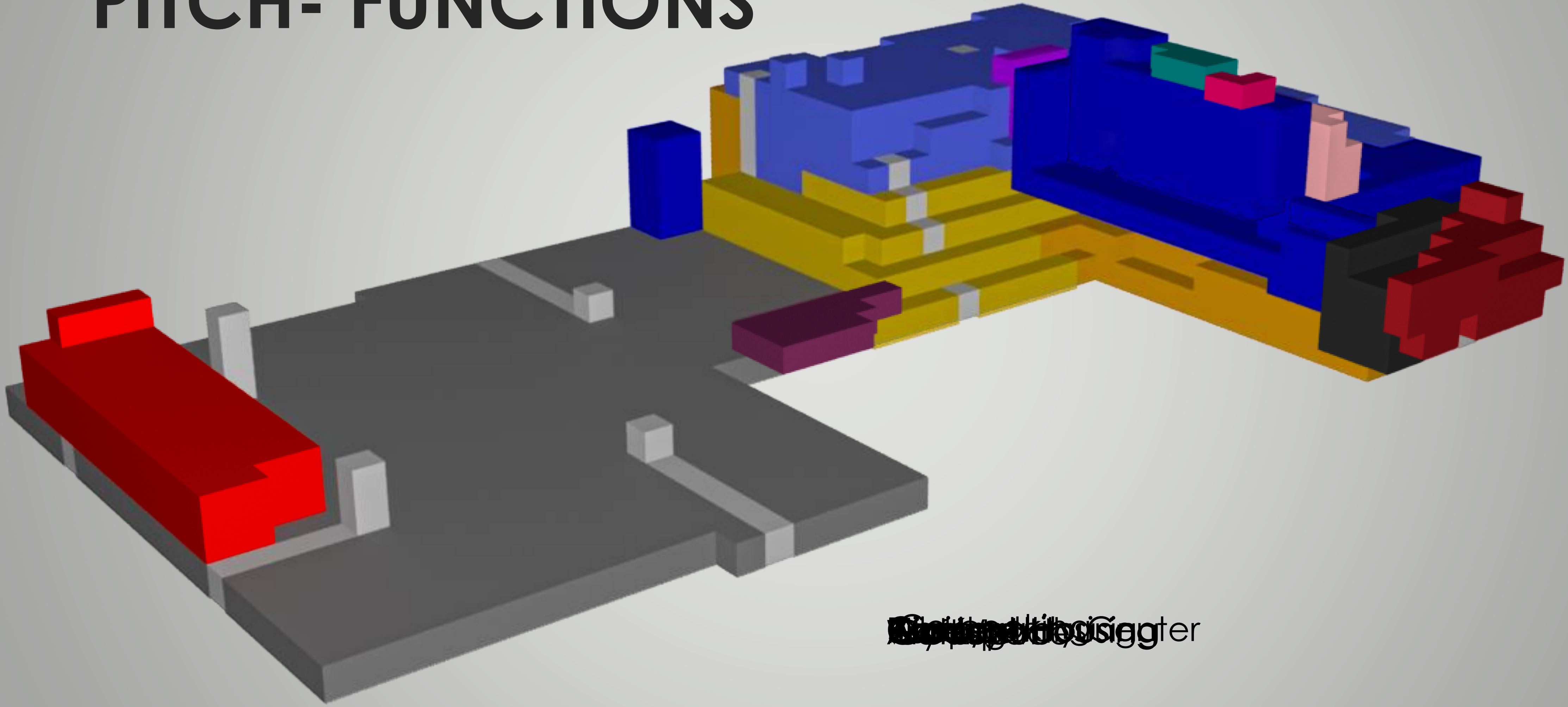




PITCH – FLOORPLAN/SECTION



PITCH- FUNCTIONS

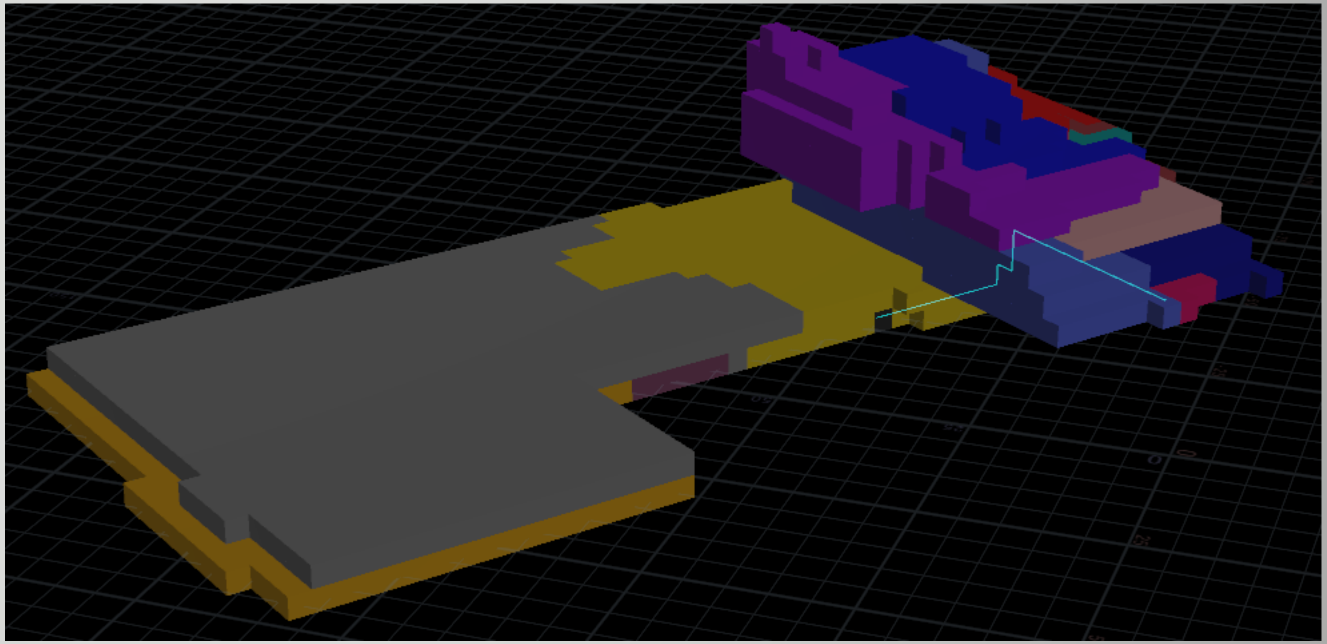


[illegible]

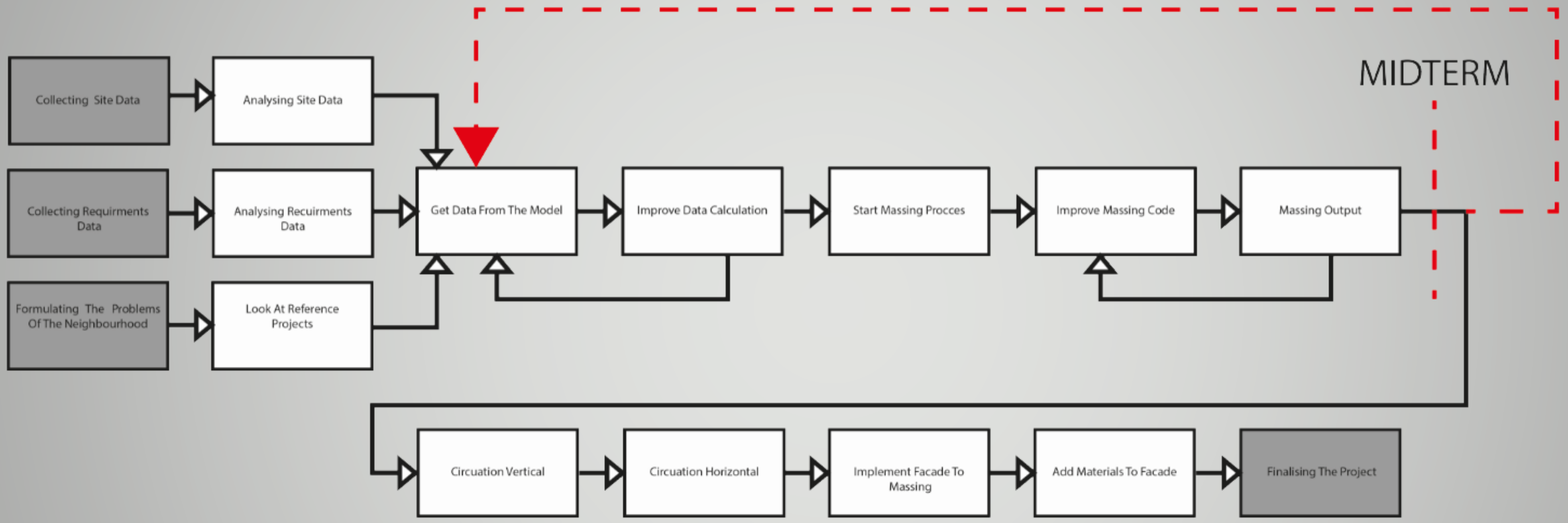
PITCH- FINAL INPRESSION

RECAP MIDTERM

- The “zeppelin”
- Correct Wp calculation
- Improve date calculation
- Add stacking to growth algo.
- Add greenroofs
- Add elevators and corridors
- Implement facade design



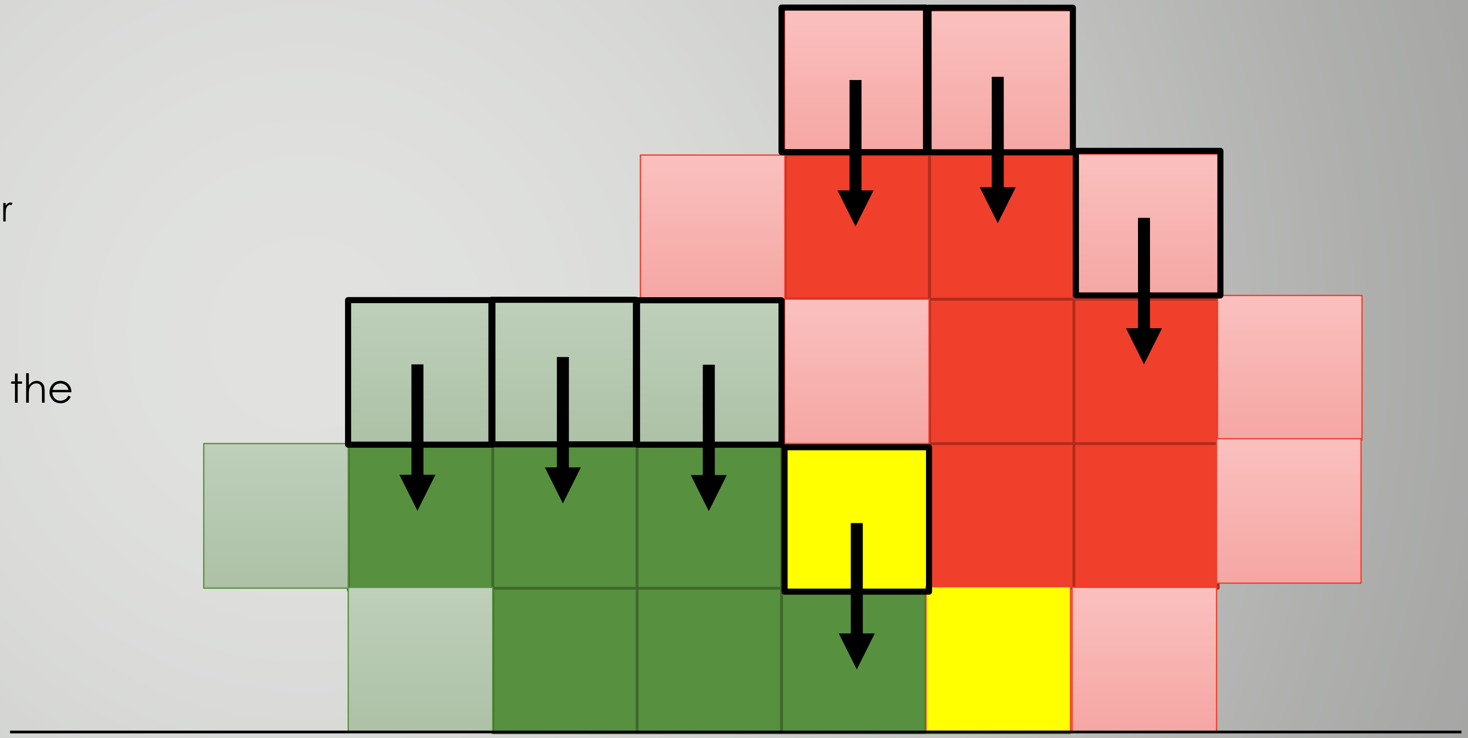
FLOWCHART


MIDTERM
Collecting Site Data
Analysing Site Data
Collecting Requirments Data
Analysing Recuirments Data
Formulating The Problems Of The Neighbourhood
Look At Reference Projects
Get Data From The Model
Improve Data Calculation
Start Massing Procces
Improve Massing Code
Massing Output
Circuation Vertical
Circuation Horizontal
Implement Facade To Massing
Add Materials To Facade
Finalising The Project

CODE-STACKING

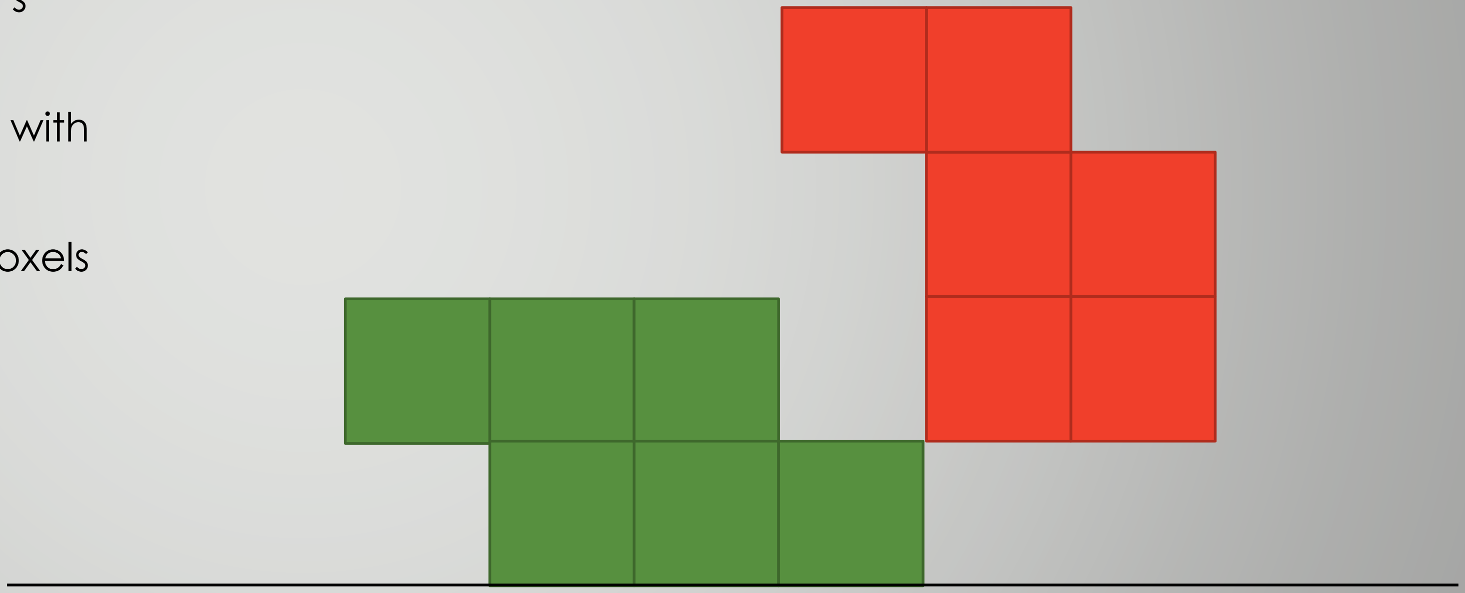
- Find neighbours
- For each neighbour
  - Check if a voxel is directly below it
  - If yes, multiply Wp with Stacking Factor

- Voxel below does not have to be in the same function group



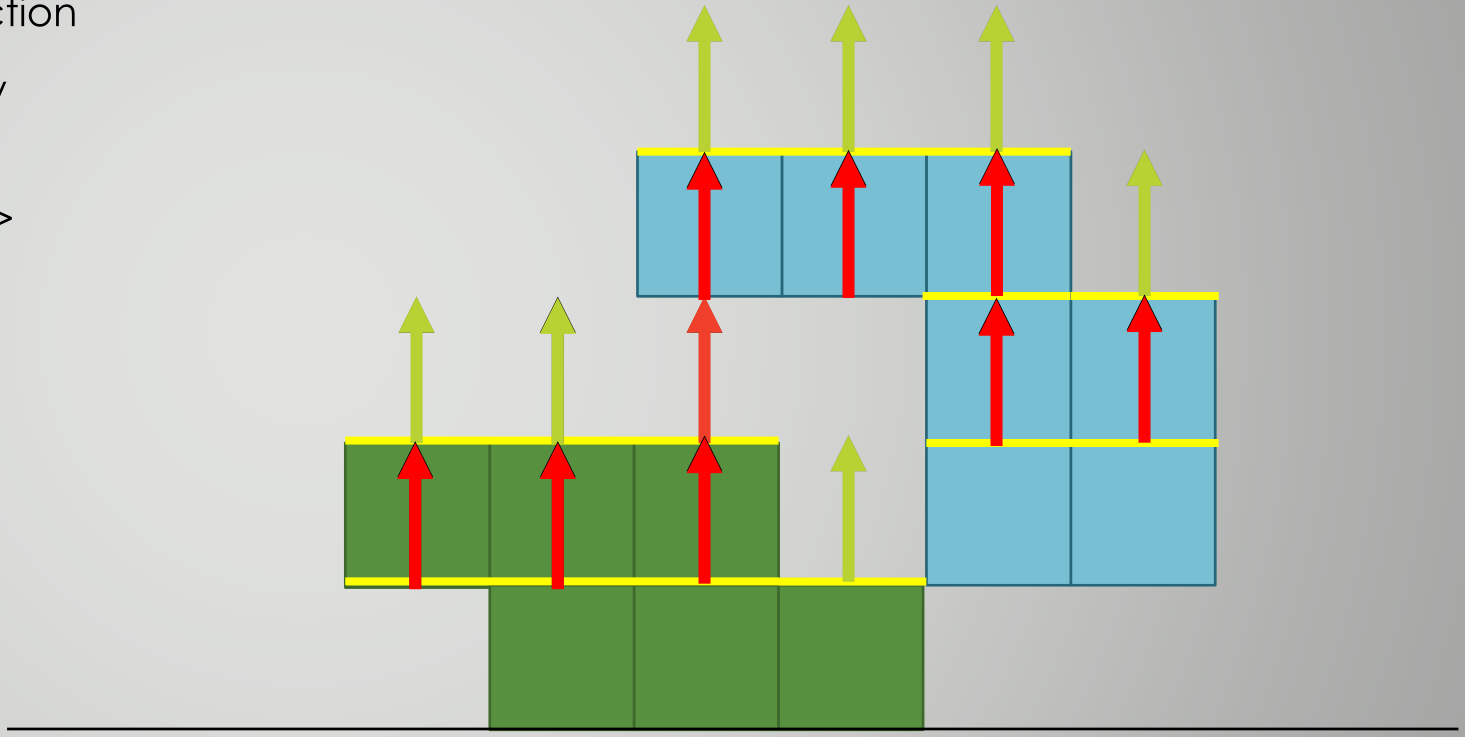
CODE- DON'T FLOAT

- When all voxels are placed, for each function find the neighbours
- If neighbours position is lower than it's childvoxel, add to list
- Swap top voxel to lowest neighbour with same **X,Z**-value
- Continue until there are no empty voxels below the childvoxel



CODE-GREENROOF

- For each voxel shoot a ray in Y direction
- If distance from intersection is below threshold -> voxel is not sutiable
- If distance is above threshold or -1 -> voxel is sutiable
- Chance parent of topprim to green function

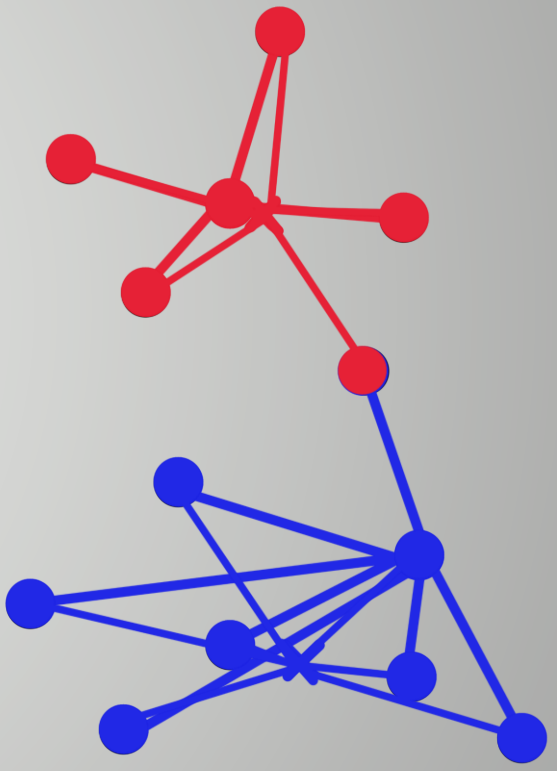


CODE-ELEVATORSHAFTS

- K-mean sorting algorithm

- Cluster occupied points in 2D field
- Start with 2 clustercenters
- Measure each distance from cluster-point to –center
- If **one** distance is above treshold, restart with numclusters+1

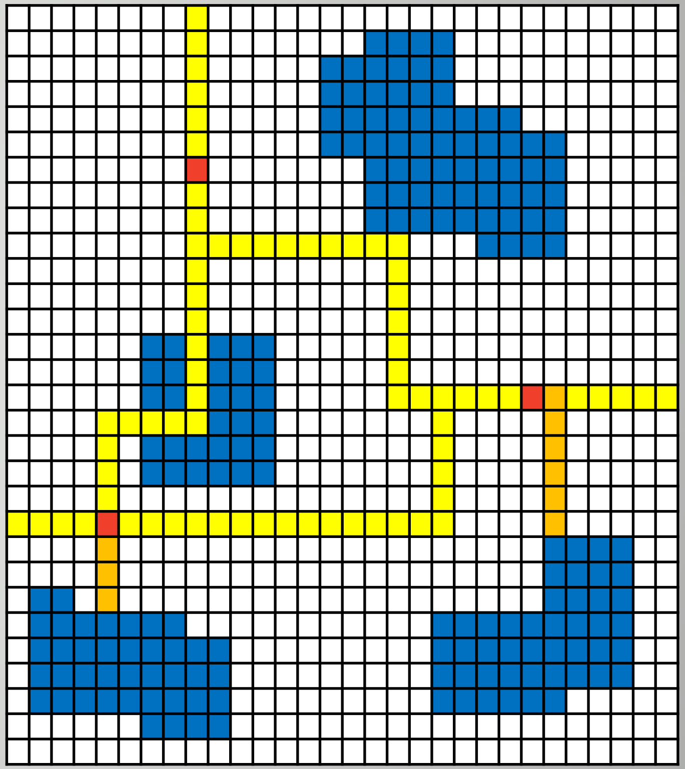
- In cluster group find highest point, divide Y-value by voxel height ->num floors per elevator



CODE-CORRIDORS

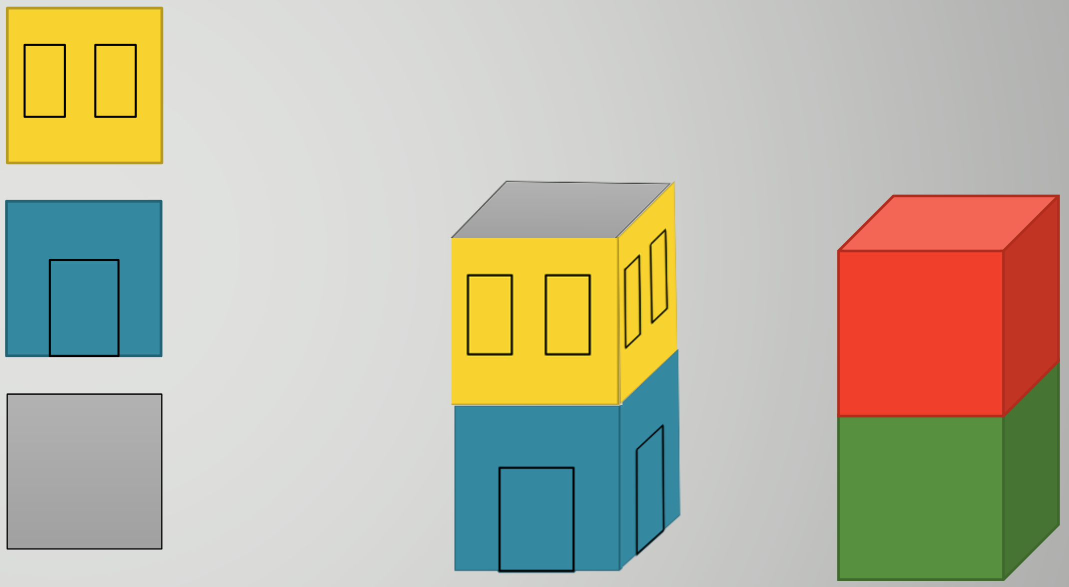
- For each elevator
  - Find shortest path to the street
  - Find shortest path between elevators

- For each function that is not connected to the formed paths, find shortest paht to closest elevator



CODE – FACADE IMPLEMENTION

- Add parent ID to voxelfaces
- Remove inside voxels en faces
- For each face determine the Normal vector
- Add corresponding template to faces using the normal and parent ID
- Export to Lumion for final representation



PROGRESS VISUALISATION

THAT’S IT

- Questions?