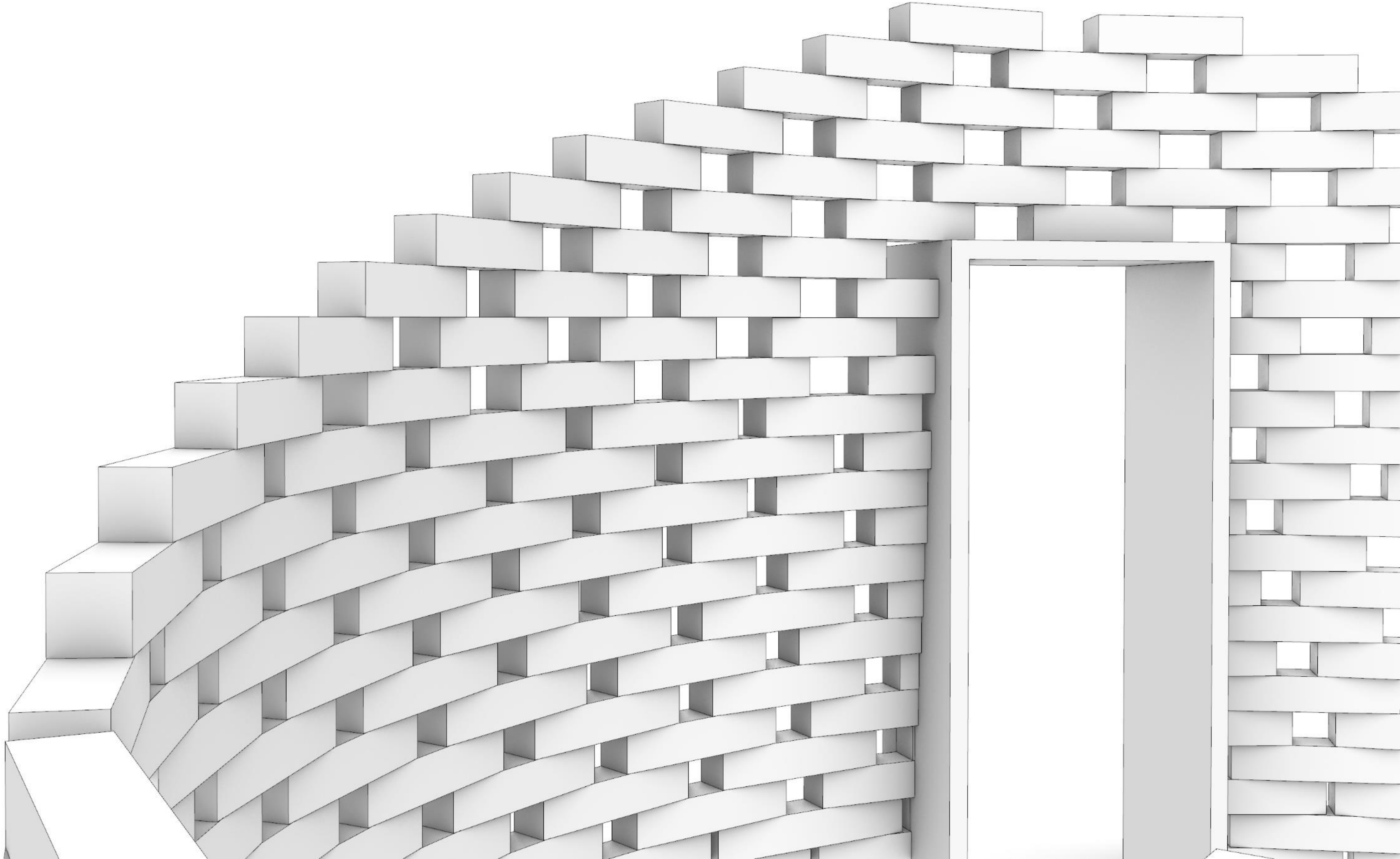
# portfolio



Pál Mészáros
2026

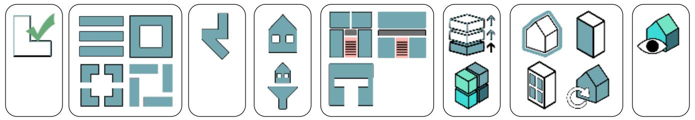
# planpal

I developed PlanPal as a Python-based generative design plug-in for Rhino/Grasshopper that creates mass housing building models from data through algorithms. The idea grew out of an internal evaluation of commercial tools such as TestFit and Digital Blue Foam: we tested them as potential solutions for generating design options for real estate developers, but they couldn't quite deliver the flexibility and control we needed. I suggested to my boss that I could build a similar engine directly in Grasshopper—less polished than a web app, but fully transparent, customizable and tailored to our workflows—and then set out to do exactly that.

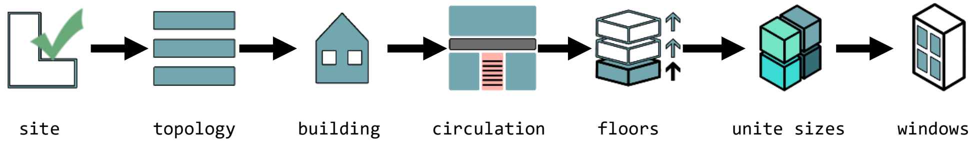
PlanPal was conceived as an open, extensible alternative to standalone feasibility tools like TestFit, Autodesk Forma or Archistar, but running natively inside Grasshopper. It focuses on early-stage mass housing workflows and is designed to integrate with the wider Grasshopper ecosystem, remaining compatible with third-party plug-ins such as Ladybug, Planfinder and Colibri. A dedicated website and a series of YouTube tutorials document the system and make it accessible for other designers who want to adopt or extend the workflow.

2023.06.
[https://www.plan-pal.xyz/](https://www.plan-pal.xyz/)

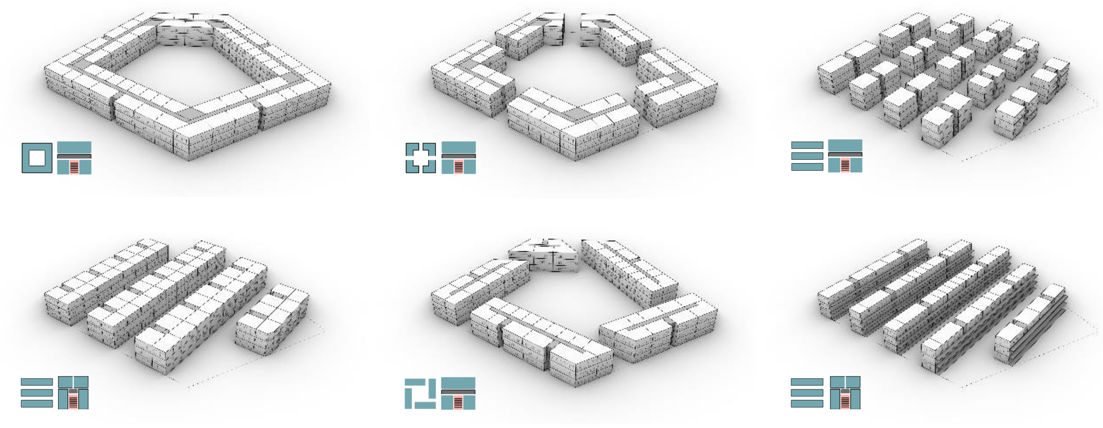
## components



## workflow




## results



# urban typology clustering

This project started from a simple discovery: I realized that Google Earth’s 3D data can be brought into Rhino, and began brainstorming how this could be turned into something useful for real estate developers. I first built a Grasshopper workflow that reconstructs 3D models of city neighbourhoods (or larger areas) using OpenStreetMap and Google Earth data, creating a parametric representation of buildings and their urban context.

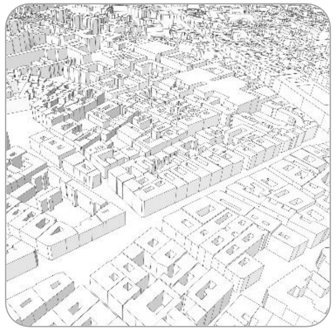
On top of this model, I developed a machine-learning clustering pipeline that groups buildings according to their urban typology and morphology—how they sit within the street and block network, their scale, and their relationship to their surroundings. By comparing each building to the dominant local pattern, the system highlights outliers that do not fit their context, often revealing underbuilt plots where zoning may allow significantly larger developments. The project was both a way to explore hidden development potential for real estate and a chance for me to dive deeper into practical machine learning applied to urban form.

2025.07.

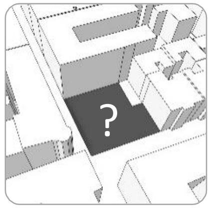
osm

google maps

data



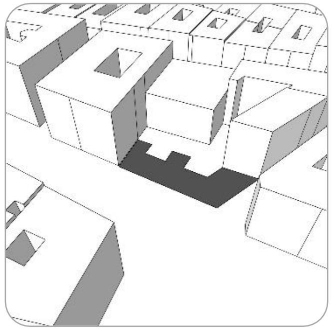
3d city



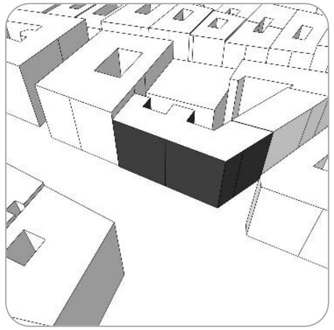

Which site is **underbuilt** compare to its environment?

**Categorizing** all building according to their urban morpholgy

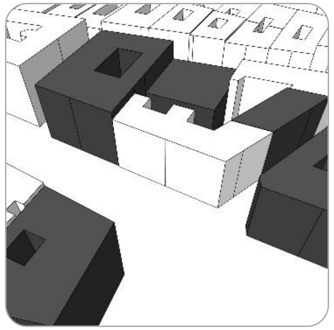
Morphology data:



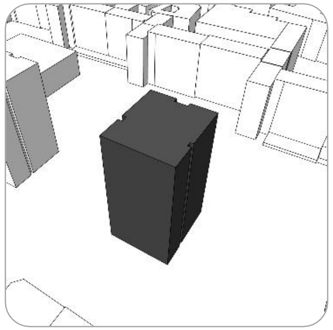
area, thickness



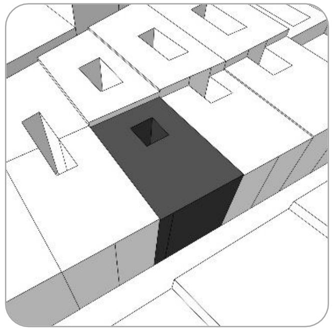
height



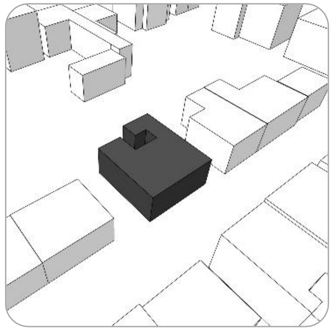
neighbour relations



topology (example 1)



topology (example 2)



topology (example 3)

typology map
example findings (left)

# parametric masterplan

I developed a parametric masterplan model for a new mixed-use city neighbourhood in Budapest, created for a real estate developer's master thesis and published as an interactive tool on ShapeDiver. The Grasshopper-based script generates entire building configurations for the site and provides detailed area breakdowns that developers can use for feasibility calculations. Users can explore a wide variety of design options by adjusting parameters such as building heights, densities and mix of functions—or simply input a target FAR value to obtain a corresponding design solution.

The project focuses on a large area (360 000 m²) next to a railway station, where the long-term vision is to move the tracks underground and reclaim the surface for development. The parametric model supports this study by testing scenarios for residential and office buildings, commercial ground floors, underground parking and generous park areas, helping to evaluate how different urban configurations perform both spatially and economically.

2025.10.
https://www.shapediver.com/app/

site

## adjust parameters -> design options

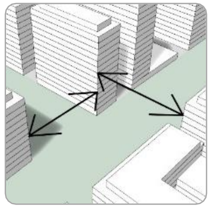
two versions:

1. multiple parameters, precise design modification
2. 1 parameter, FAR value

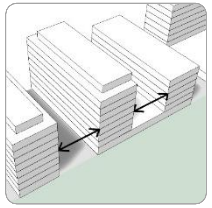
divided into **zones**

apply **grid** system

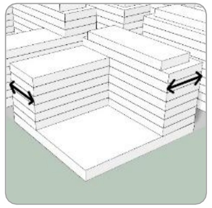
## version 1 – parameters:



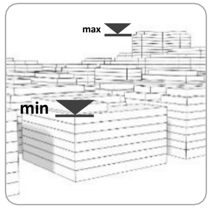
green path width



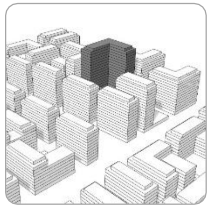
building distance



building thickness




building height

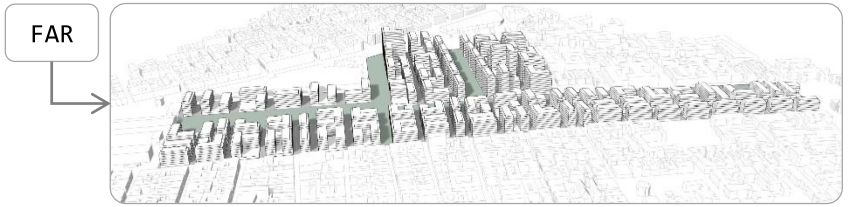


tower location

- floor height
- parking floor area ratio
- penthouse
- design randomizer
- …

etc.

## version 2

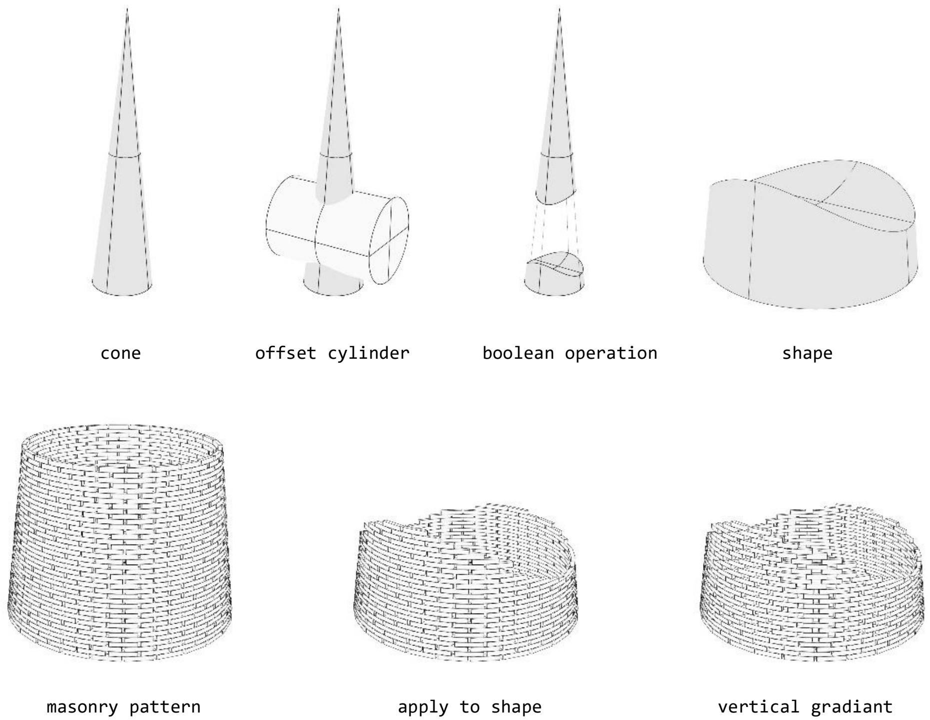


# firepit installation

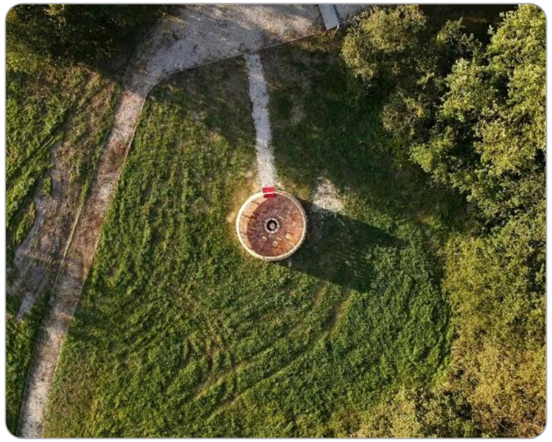
As part of a corporate social responsibility project with Opinion Builders, I helped organize and execute a 4-day construction camp in Enying, a small underdeveloped town where the company was already engaged in ongoing projects.

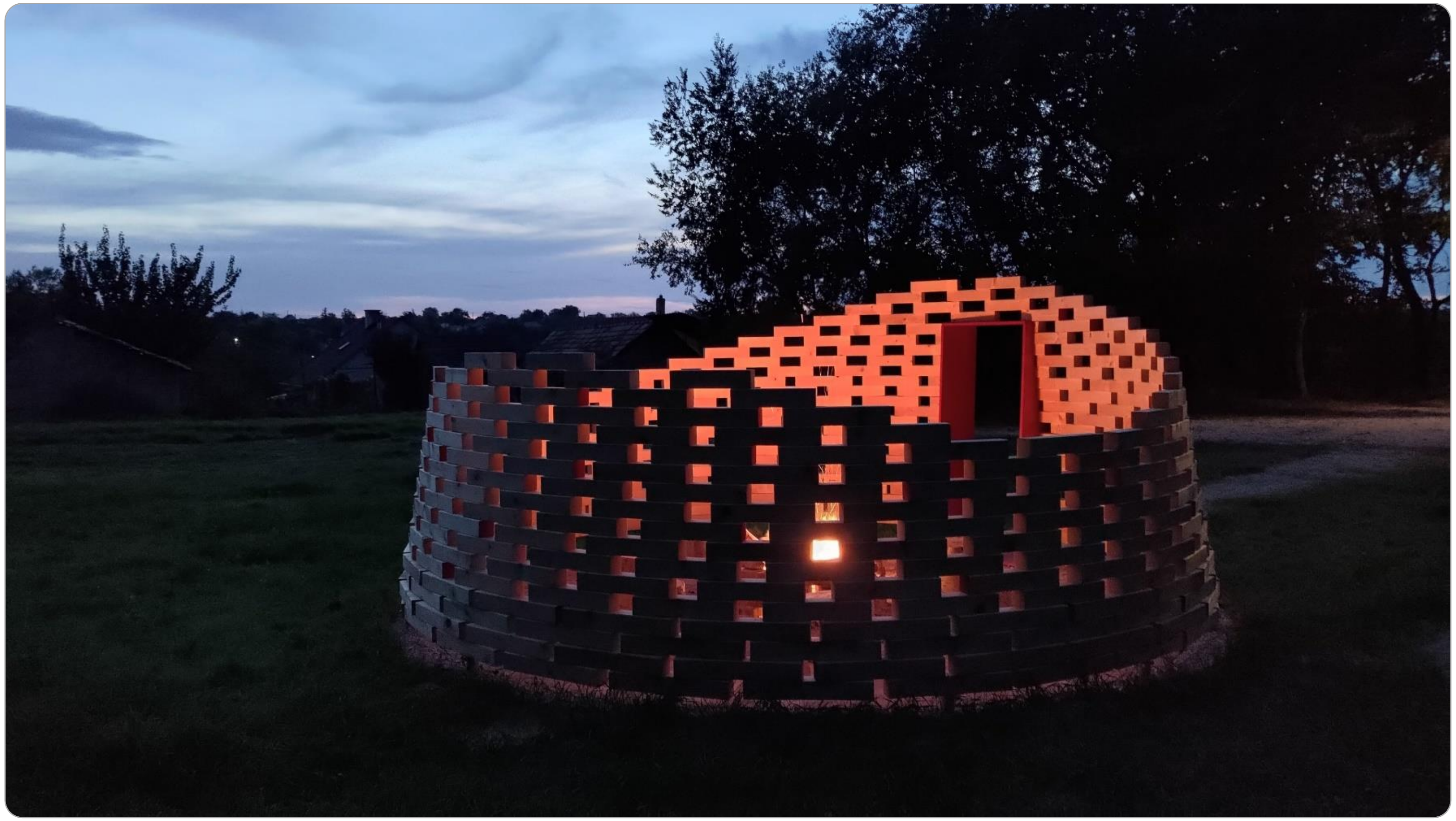
We designed and built two installations; my main contribution was the firepit, which we developed using parametric design tools to achieve the intended geometry and construction logic. I translated the design concept into a Grasshopper algorithm that could drive the fabrication and assembly process on site. The project successfully combined CSR impact and team-building, and it allowed me to experience the direct, hands-on implementation of computational design in a real construction setting.

2022.08.
https://bigsee.eu/firepit-and-panoramp-in-enying/





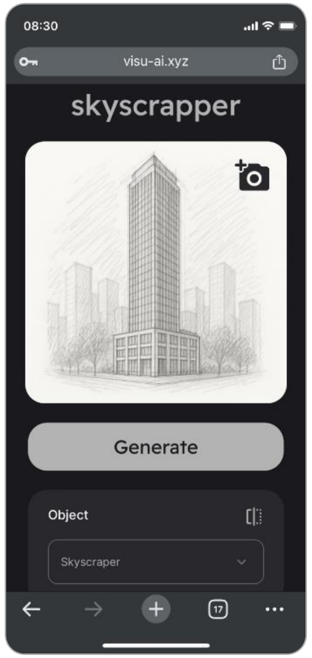

# visuAI

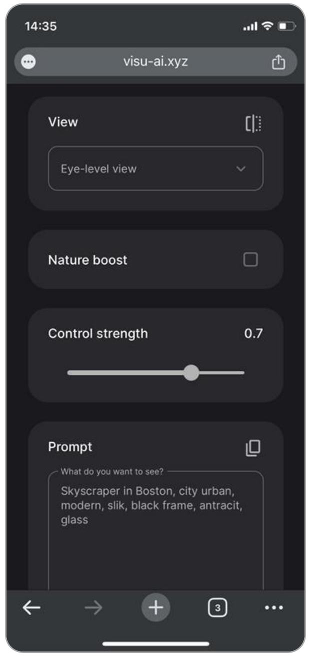
I was interested in mobile app development, because it is the most used device today, and Check was the starting point when we implemented it for tablet use.

I built VisuAI with FlutterFlow and Stability AI as a focused mobile and web app that turns hand-drawn sketches into realistic architectural renders. The goal is to bring hand-drawing back into the concept phase while still giving clients the polished visuals they expect. Architects sketch on paper, capture it with their phone and generate renders without opening CAD or a 3D package.

The project was also my way of learning how to integrate modern AI image-generation APIs into a simple product. VisuAI never became a commercial tool—we use it internally occasionally, and I chose not to compete with the many AI image generators on the market—but it has been a valuable prototype and learning experience in mobile app development and AI-assisted design workflows.

2025.03.
[https://visu-ai.xyz/](https://visu-ai.xyz/)

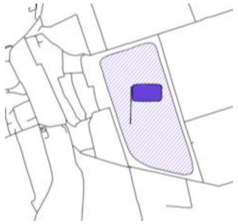

# check

Building on the experience from PlanPal, my boss—who works closely with developers focused on warehouses and industrial buildings—asked for a similar tool tailored to logistics projects. Existing feasibility software didn't fit their specific needs, so we decided to create our own web-based decision support app.

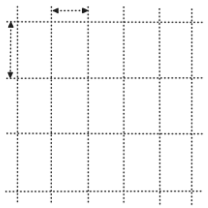
I designed and implemented the backend, which generates industrial building configurations and aggregates detailed building and financial metrics. A colleague, an architect whom I had previously introduced to programming and who later specialized in frontend development, leads the user interface and interaction design. Over time the project evolved: for this tool, a clear and intuitive UI/UX became as important as the underlying algorithms, and the focus shifted from pure generative design to fast, understandable decision support.

The app allows users to sketch warehouse and industrial layouts directly in the browser—optimised for both desktop and tablet—then review key quantitative outputs for feasibility studies and business planning. While the current version emphasizes interactive layout creation and analysis, our roadmap includes integrating more advanced generative capabilities in future iterations.

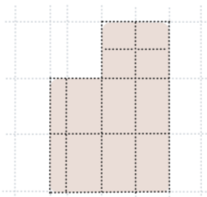
2025.09.
[https://check.novu.eu/](https://check.novu.eu/)



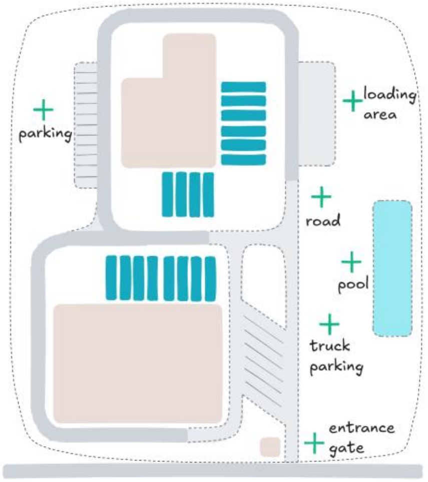
Draw your site

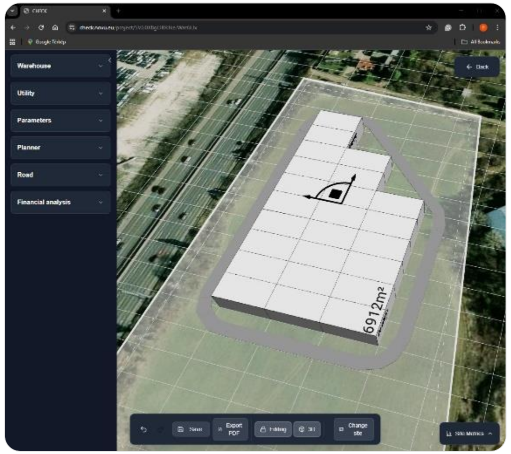


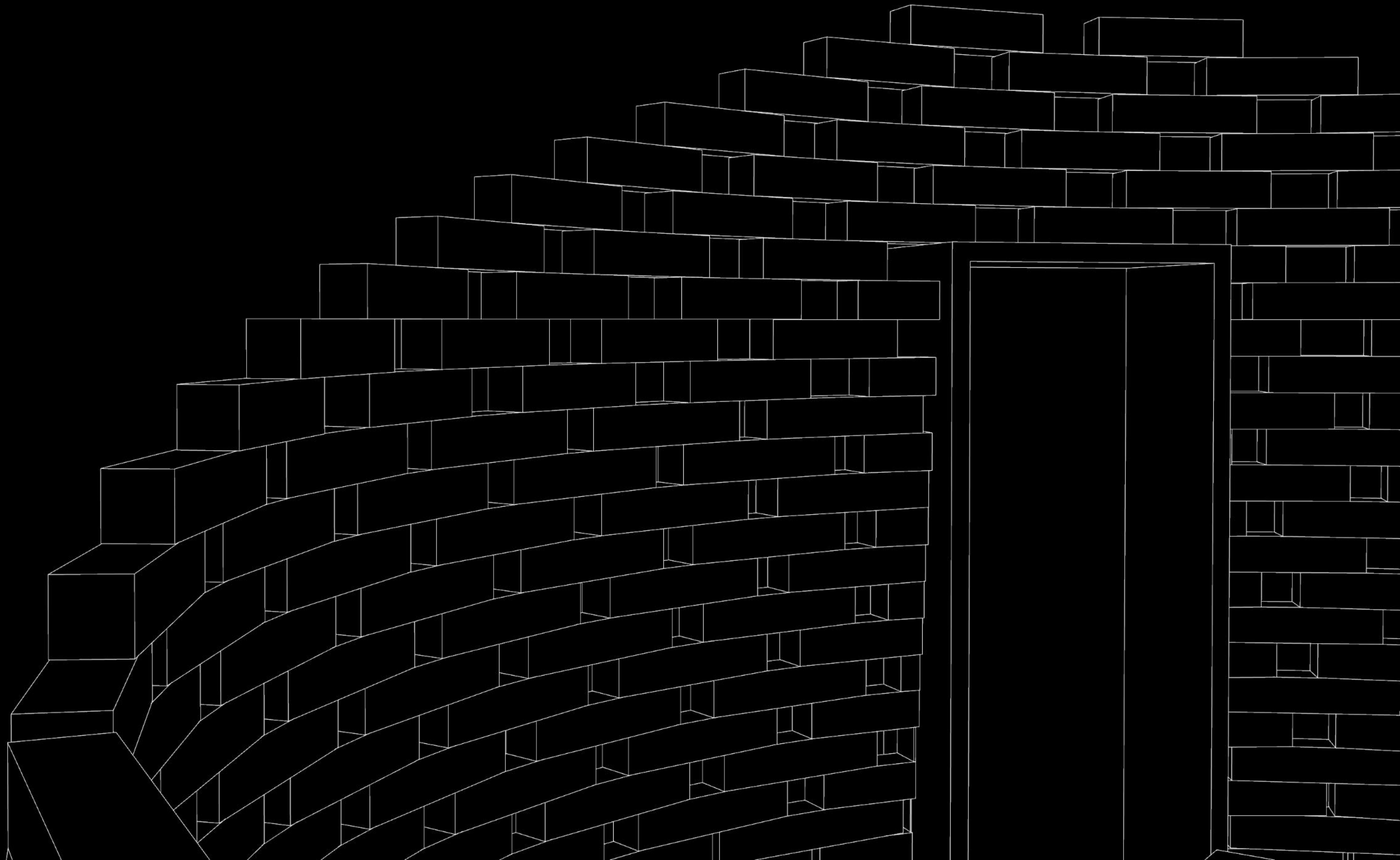
Set raster sizes



Design your building



Pál Mészáros
2026