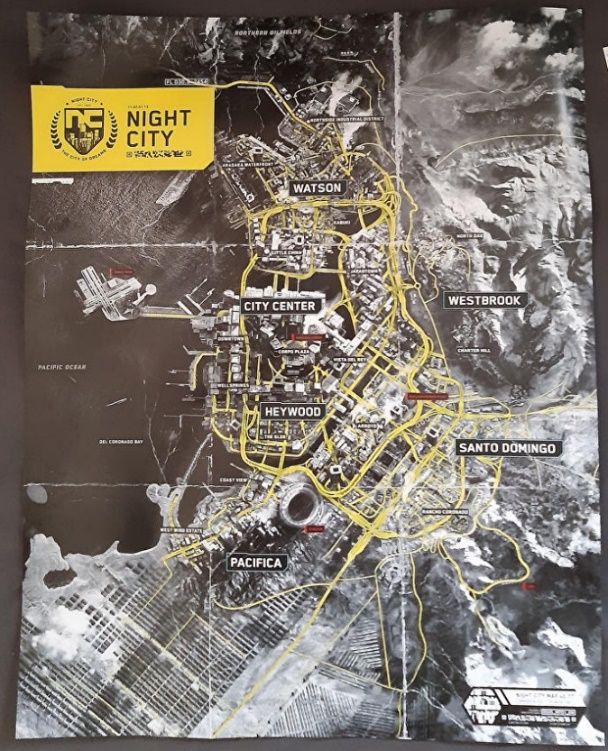
Research Procedural Art Nils Meijer 466301

**Chosen theme:** Downtown Night City

**Time of day:** preferably dynamic, otherwise deep night or sunset.

A picture containing text, indoor

Description automatically generated**Visual references:**

**Reference analyzation:**

During the night, when the city is less active, the insane amount of light emitting from the buildings replaces the sun. Colour palettes that are rather common in a cyberpunk city *during night* are combinations of blue, purple, pink, white, green. There is a certain “glow” present slightly above the city, also known as “light pollution”. It allows for a very artificial and active mood, characteristic for the city that never sleeps.

Architectural elements

The skyscrapers consist of countless tiny windows and are relatively straight with some exceptions. Sometimes, they are connected by the use of huge bridges/corridors high in the sky. Floors often look like they have been duplicated multiple times, with minor adjustments each iteration.

City shape

Street-wise/city layout-wise, the organization is very structured, with streets often being parallel to each other, especially in the city center.

Materials

The buildings often have a metallic material, reflecting the light cast upon them. Depending on the location of the building for which the material is used, the metal/glass is either squeaky-clean (corporate buildings, part of the elite/upper class of the city), or dirty and covered in dust/smudges (poorer areas of the city). A concrete material is also rather common, often being used for walls, next to metal.

Procedural techniques/tools

**Tools required for a cyberpunk style city/what am I going to work on:**

**Spline placement system**

A spline placement system, to determine where roads are located. Based on this pre-defined roadmap, the city blocks are generated. Depending on the shape of the available space, the proper building should be generated for each place in the city block.

Using the location of the city block, the height of buildings should be modified. The closer a block is to the city center, the higher a building has a chance to be. This should give a nice skyline, with the tall buildings mainly focused in the middle of the city.

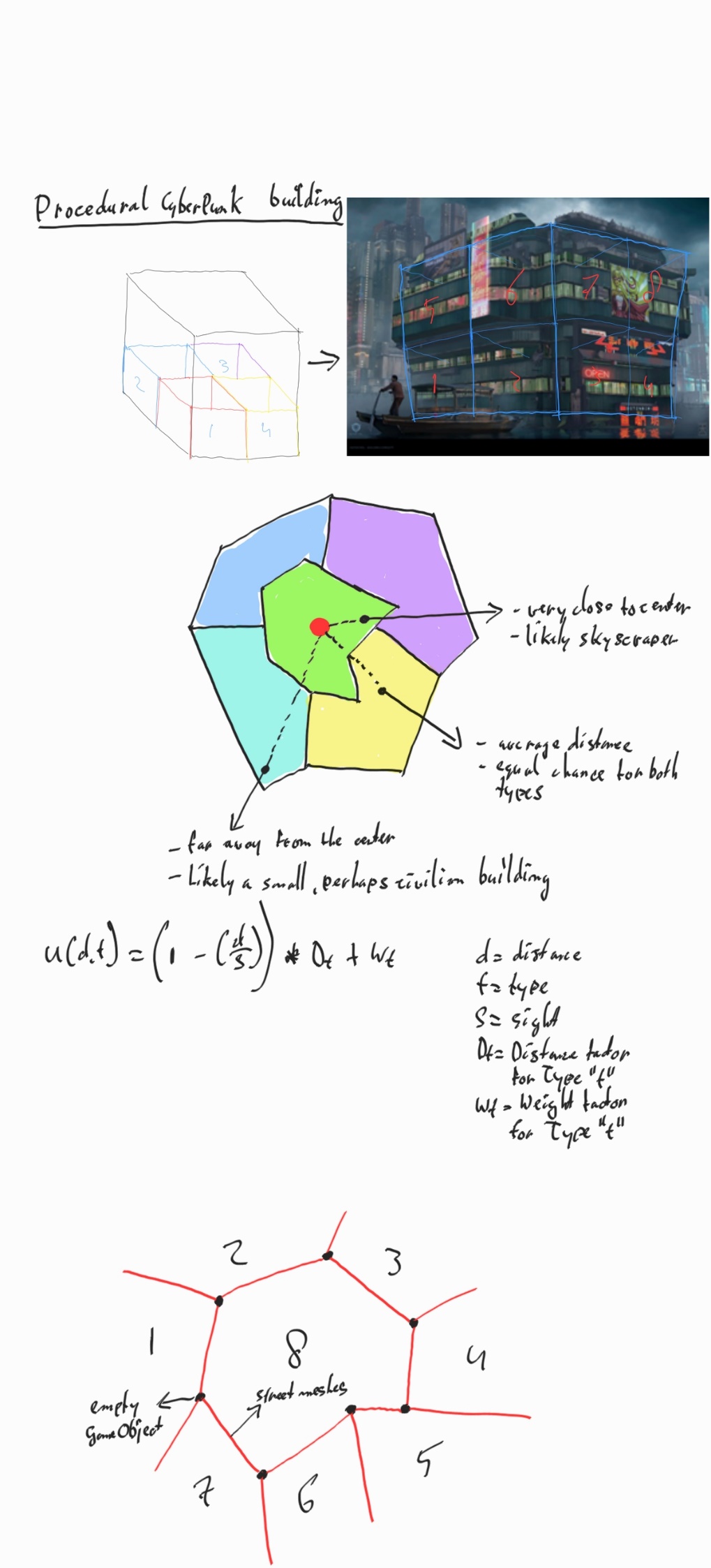


Figure Spline system

This system is used to place so-called “nodes”, which connect streets together. At the same time, these connections between nodes represent the streets themselves. When all desired nodes have been placed, the city layout will be generated (where are all the city blocks located?). In figure 1, you can see a visualization. The black dots (as said in the image, though possibly not readable due to my handwriting) represent the nodes (empty GameObject) that can be placed by me, and the red strokes represent the streets which get created because of the nodes being connected to any given node (determined by an index). Based on the resulting map, the city blocks are generated (which is what the numbers 1/8 represent; although 8 city blocks is entirely arbitrarily.

**Building procedural generation**

To generate the procedural buildings themselves (because not only do I want the city blocks to be procedural, the buildings should be unique as well instead of choosing from a list of preselected complete buildings), they will consist of multiple different components (as seen in Figure 1). Depending on the parameters given by me, fitting prefabs will be chosen. After the full city has been generated, I should be able to modify individual buildings by selecting specific “stocks” of the building, and either choose a new one from a list, and/or regenerate the building itself entirely.To make sure all given prefabs work together nicely, I’ll have to combine the parts by hand myself and make sure the origin and rotation of the prefab is at the same place *every time*, no matter the size of the finished piece. For example, the right bottom corner. This way, there shouldn’t be any unexpected overlap problems or issues where the prefab is facing the wrong way.



Figure 2 Building layout

**City procedural generation**

I am planning to adjust the “Utility function” often used for AI characters in games to determine what kind of building should be generated (tall skyscraper in the center of the city, or a civilian property with only a few floors and a different appearance than a corporate building). See Figure 2 to see a visualization of this.

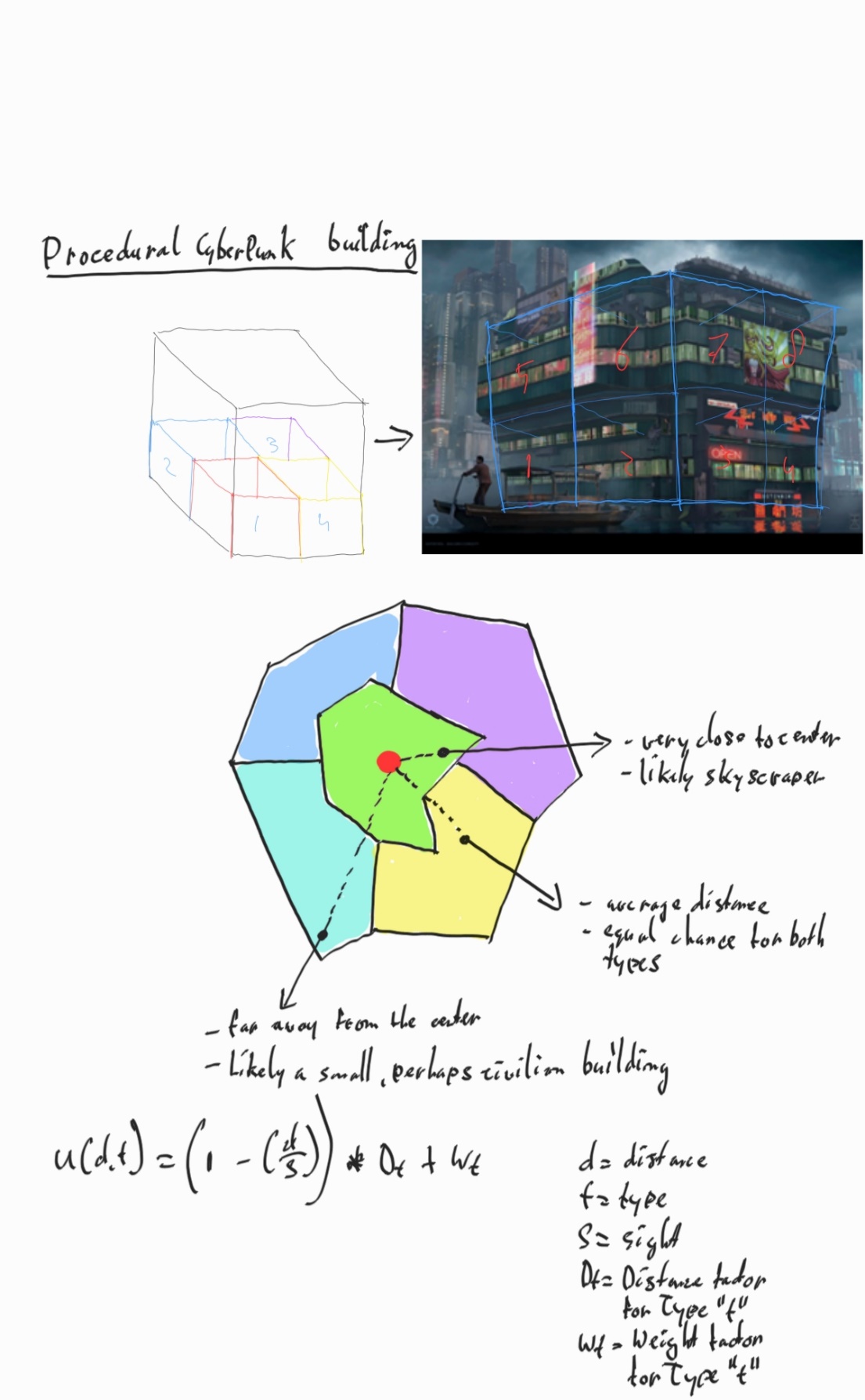


Figure 3 City layout

Utility function

Formula given during the Advanced Tools course, with annotations added by me to adjust to my idea:

U (d, t) = (1 – (d / S)) \* Dt + Wt

d = distance to the center (red dot)

t = type (e.g. 70% is corporate/skyscraper, 30% is civilian/housing)

S = Sight (is the building close enough to the center to even be considered for the

Dt = Distance factor for Type *t*

Wt = Weight factor for Type *t*

Depending on what city block the building is in, I can assign certain values to the yet-to-be made buildings in that specific city block. To make sure there is still diversity in that block, every building has a randomized value that’s included in the utility function, so it’s still *possible* a skyscraper is slightly further away from the center and a house is slightly closer, but only as an exception.