

Paris in 1789 - Research

Kaloyan Ivanov

545447

City Layout

18th century Paris is a stunning and diverse city. It is geographically divided by the **Seine river** through the middle and administratively into **7 districts** - each unique on its own. The most **distinctive** of the regions being '**île de la Cité**'; it is situated at the center of the city and comprised of **two islands**, surrounded by the river.



Infrastructure

Looking at the map on the left, we can clearly see that the city **infrastructure** is quite **chaotic** and with **no visible building pattern**. Contrary to common grid-based approaches that focus on street/road placement, in its ideology, it is heavily **house-block-oriented**, leaving the streets around the houses to adapt to the house positioning, which may end up in **varying street widths**¹.



However, there are also **main roads of larger width** that **divide separate regions**, therefore being of **higher priority** compared to **houses**.

Following this tendency, we can also notice that in each region where important landmarks are present, **house blocks' positioning** seems to be **based around** the given **landmarks**'.

As a result of those key findings, I can **accurately design** my procedurally building algorithm around the following **priority list**: **Main streets** → **Landmarks** → **House blocks** → **Secondary Streets**



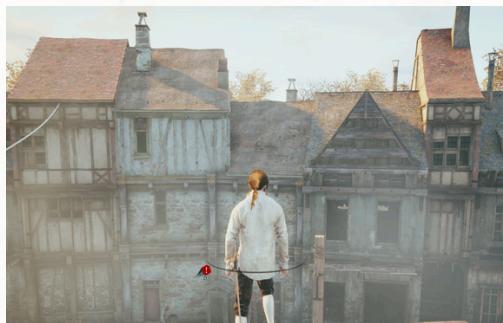
Building Styles

However, regions can be divided into **three categories** - **poor**, **mid** and **rich**. Depending on the region a house is part of, its **building style** and **materials used** vary.

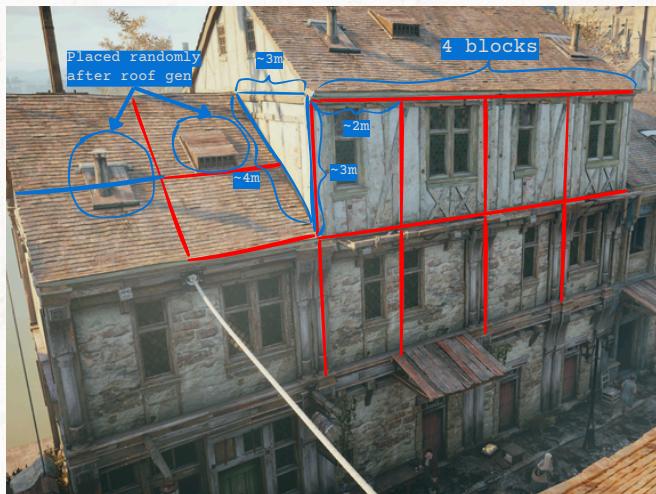
References:

1: [Assassin's Creed Syndicate: London Wasn't Built in a Day](#)

Poor

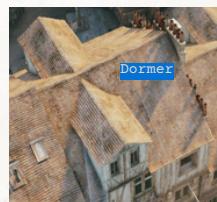


As you can see from the map above, it is the rarest category, and it also happens to be the most unique - with a more old-village-type houses feel.



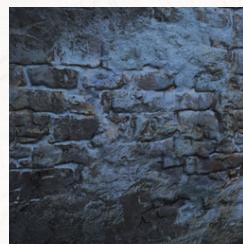
Regarding **dimensions**, poor houses keep it mostly **consistent**, with a **singular block dimensions** of **~2x3x3 meters**.

Interestingly, houses from different categories differ the most by **roof types** - the **poor** ones having mostly **straight roofs** as seen in the image on the left. However, there are also some **exceptions**:



When it comes to **materials**, there are the following materials **unique** for the given **richness category**: **walls, roofs, and ground**.

Walls:



Worn/Rotten wood

Worn Plaster

Cement covered brick wall

Roofs:



Worst



Mid



Best



Mud

Ground:

Procedural Materials to use with properties adjustments:



With different color values



With different parameters for different quality of roofs



Doors:

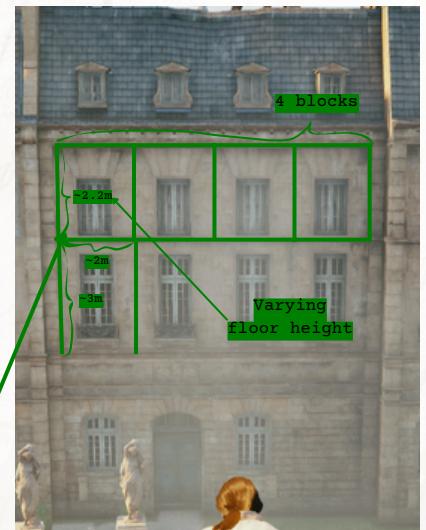


Mid



Something that **mid** and **rich** houses have in common is their **mansard type roof**. Since there are **multiple variants** of this type, I have decided to **generate them procedurally** using **curves**, instead of having premade modules, but more on that **later in the document**.

Dimensions



Mid materials

Walls:



Plaster

Roofs:



Mid



Best

Ground:



Mud/Paving stones

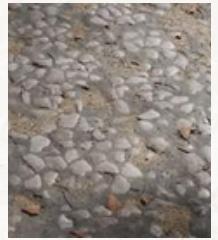
Procedural Materials to use with properties adjustments:



With different color values



With different parameters for different quality of roofs



Dusty random paving stones

Rich materials

Walls:



Roofs:



Ground:



Dusty paving stones

**Procedural Materials to use
with properties adjustments:**



Procedural generation techniques

The **city generation** will be split into two main phases: **Initial Generation** and **Fine-Tuning**

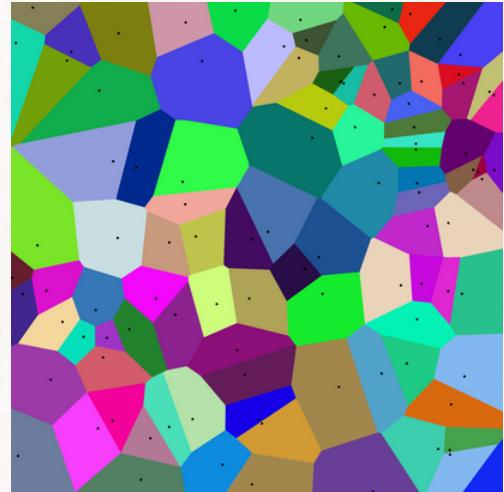
Initial Generation

As mentioned previously, Paris' infrastructural characteristics are of **organic/irregular street patterns** where the **focus** is put on **houses formation, rather than street consistency**. To satisfy those requirements, I have researched various approaches and found utilizing the **Voronoi Diagram** to be the most suitable for **procedurally generating the house shapes**.

Combined with **Editor Tooling in Unity**, I will be able to move **seeds**(black dots in image on the right) around and generate new ones. This will enable **finer control** over the **cells'**(colored polygon containing a seed) **size and density**, which will then directly translate to **houses' properties**.

For the **main streets and landmarks**, **Editor Tooling** will also come in hand as they will be able to be placed as **2D lines and shapes** from a top-down perspective to **modify** the already generated **Voronoi pattern** or **influence** it before its generation happens.

Regarding the **secondary streets between the house blocks and their widths**, I have decided take the following approach: the further two neighboring seeds are from one other, the wider the street between the actual house blocks is going to be. This makes sense if we go back to the map displaying the economical distribution of the city: **richer house blocks** are **more secluded** from their neighboring than the **poor ones** - they are smaller in area and therefore **closer to one another**.



Voronoi Diagram¹

Fine-Tuning



Ubisoft House Generator²

While I was researching the **development process behind AC Unity**, I stumbled upon this great resource² where the developers disclose parts of how their **proprietary house generator** works.

References:

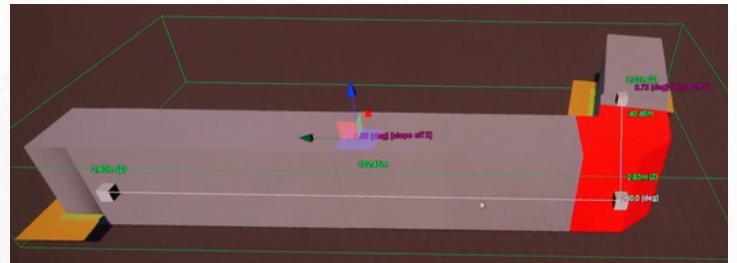
1: [Voronoi diagram](#)

2: [Making Assassin's Creed Unity: Part 2 - Next Generation Technology](#)

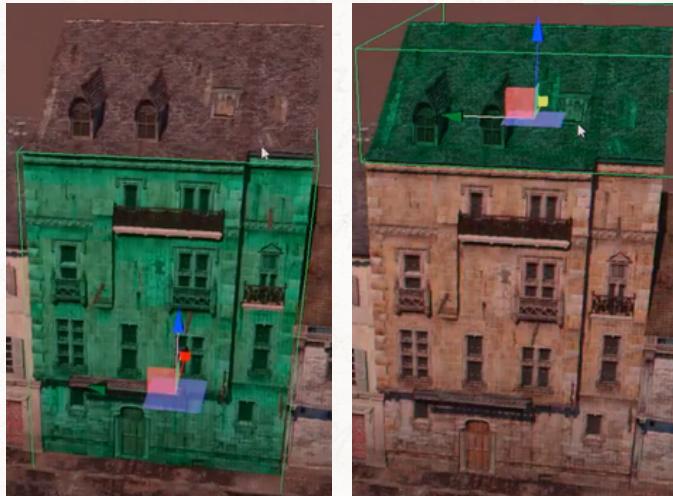
First, they start by **drawing a house block** by **putting down points** which when connected, form a coherent structure (as seen on the first image).

Then, they choose a **theme** out of a list used for the **population** of the already formed **structure**. The themes in my recreation could be the three different economy districts: **poor**, **mid**, and **rich**. Each **theme** will have its extensive **list with settings** depicting **house properties** regarding height, width, facades, roofs, materials, etc. **Based on those settings, the house block will be populated**, achieving the desired look and feel.

In my case, the **Voronoi cells' borders** would act as the **connections between those Editor Tool's points**. The user would have the ability to **select** a given **Voronoi cell** and **set its theme** before the buildings get generated.



Ubisoft House Generator¹



Ubisoft House Generator¹

After being **generated**, each **house block** can have its **theme modified** or **completely replaced** by another. That holds true also for the **separate house block lines** and **individual houses**.

For more **fine-tuning control**, each **house** will be able to be **selected individually** and **its properties modified**.

Terrain, VFX and Post-processing

Based on how I manage with the main house generation tasks, I will decide if I will have additional time to spend working on the following two groups of tasks:

Adding Seine River

To add space for the river, I would need to add support for **terrain modification** so I can elevate Paris buildings and make space for the river bed.

Research how to implement the **river** (maybe shaders) and do it.

Adding Environment FX

Implement **Fog/Dust effect** with P-p or shaders as shown on the image:



Implement **day-night cycle** for different environment settings.

References:

1: [Making Assassin's Creed Unity: Part 2 - Next Generation Technology](#)