Procedural generation of interior settings

Manuel Lampert

Hochschule Luzern

Digital Ideation 3nd Semester

Studio Game Extended - Dragica Kahlina, Sebastian Hollstein

25th of November 2021

Abstract

Procedural content generation also known as PCG, is becoming a widespread technic for improving iterative pipelines for content creation. Especially game development can greatly benefit from this approach, as many games rely on agile iterations to ensure gameplay quality. But for now, these techniques have been mostly used for large environments such as terrains and cities. Putting a focus on the interior of buildings and houses one could improve player experiences even further, because these areas wouldn't have to be blocked or empty because production did not allow for this amount of detail. For this reason, this paper focuses on the procedural creation of interior designs with 3d furniture.

Keywords: procedural content generation, interior design, 3d

1 Procedural generation of interior settings

PCG frees up the artist of repetitive tasks and enables them to focus on creating better experiences (*Technical Artist Bootcamp*, 2019). But to get to the point of enabling creators, the problem needs to be identified and broken up into its components and build back together with parameters to control the output. So, the motivation for this paper was to enable faster creation and gaining experience in breaking up a complex problem set such as the design and placement of furniture in indoor settings.

2 Research

Comparing Google Scholar search results of "Procedural Building" (1'700'000) and "Procedural interior" (155'00)[1] roughly highlights the amount of attention spent on solving big PCG problems, in terms of real-world scale, rather than small problems such as interior design or set dressing. Nonetheless, papers such as *Rule-based layout solving and its application to procedural interior generation* (Tutenel et al., 2009) try to create novel ways of solving this problem. Looking at the recent developments in Machine Learning one could try to solve this problem using *Neural Networks such as the paper Scene Synthesis Deep Convolutional Generative* (Ritchie et al., 2019) tries to achieve. Using an algorithmic approach it's also possible to aid the development of Machine Learning Models such as described in the paper *Increasing generality in machine learning through procedural content generation* (Risi & Togelius, 2020).

3 Practical Approach

This paper does not explore the problem set based on specific mathematical theories described in the papers of the previous. Instead, this paper serves as an intuitive exploration of the problem set with the use of a 3d content creation tool called Houdini (*Houdini* -

[1] Based on Google Scholar search in November 2021
https://scholar.google.com/scholar?hl=de&as_sdt=0%2C5&q=procedural+building
https://scholar.google.com/scholar?hl=de&as_sdt=0,5&q=procedural+interior

SideFX, 2021). Houdini is by nature procedural and contains many node based tools for content creation (Bannink, 2009) but also allows the use of code, which is the main mode of operation for this paper.

To further simplify the problem set, a few constraints were set. The rooms are only of rectangular shape as well are the furniture's, this allows for easier solving of intersections.

Furnitures are represented by a class with attributes to describe their size as well their relationship with other furniture's. Following is a table with description for each attribute:

Table 1

Furniture Attributes

Attribute	Type	Description
node	Houdini Node	Houdini specific type containing the geometry
minDistance	[]Float	Distances for each side of the geo which should be kept free
parentPoint	Houdini node	Node to which this object should spawn close by
parentDistanceMin	Float	Minimum distance to parent
parentDistanceMax	Float	Maximum distance to parent
lookAt	String	A point or node which should be faced
blockParentSpace	Boolean	If the space between parent and child should be kept free

The Furnitures objects are added to an array in which the order defines the priority in the room e.g., first come first serve. The generation then consists of iterating through this array and trying to place them in the room by generation random positions, inside the room, that satisfy the constraints described in the attributes.

Following is the pseudocode for the generation:

```
For each Furniture in List as F
    While overlapping
        If F has Parent
            If F should face Parent
                Set orientation to Parent
            Else
                Set random orientation
                Set random position between minDistance and maxDistance
                from Parent
        Else
            If F has margin
                Set random position minus margin
            Else
            Set random position
                For each Furniture Placed as FP
                    If FP should block space between Parent
                        If F.Position between FP and FP.Parent
                            Break

                    If F is intersecting FP
                        Break

                    overlapping = False
                    Place F in Grid
```

4 Answers

This approach proves the possibility of automating the placement of objects in a confined space, following rules of finding and avoiding areas.

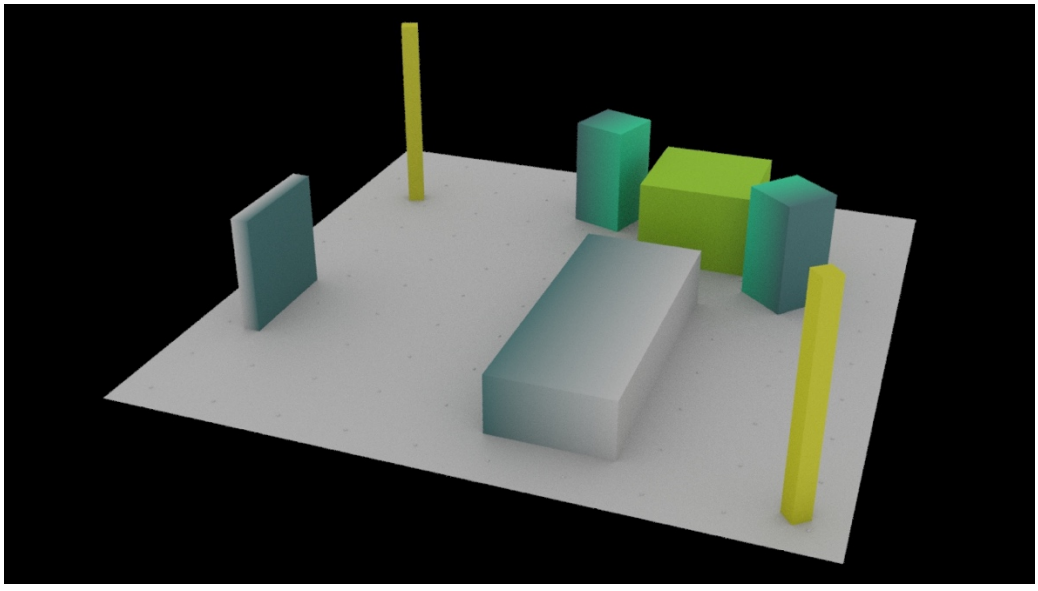

Fig. 1. Rendered result of generator

Although this implementation is successful. If evaluated over multiple generations, it becomes evident that there are edge cases which are not covered in the proposed rule set. Nevertheless, this intuitive approach of solving the problem gives a good insight in the possible approaches one could take and proves insightful for further development.

5 Conclusion

Proceduralism has a future not only in VFX but also in games and all areas of 3D. As time develops, all scales of reality will be described in processes for procedural content generation, from galaxies to cups on a coffee table. In combination with Machine Learning this will democratize 3d content creation, so everyone can create 3d scenes without the specific knowledge of a content creation tool (*Technical Artist Bootcamp*, 2019) [33:44].

As for this tool, a bigger deep dive is needed to create a robust version that covers more edge cases. Like any procedural system, the better the components of the problem set are known, the better one can control the output. As a first test to get a feel for the problems, this was worth it and will be developed further into the future.

References

Bannink, P. (2009). Houdini in a games pipeline. *SIGGRAPH 2009: Talks*, 1. https://doi.org/10.1145/1597990.1598051

Houdini—SideFX. (2021). https://www.sidefx.com/

Risi, S., & Togelius, J. (2020). Increasing generality in machine learning through procedural content generation. *Nature Machine Intelligence*, *2*(8), 428–436. https://doi.org/10.1038/s42256-020-0208-z

Ritchie, D., Wang, K., & Lin, Y.-A. (2019). Fast and Flexible Indoor Scene Synthesis via Deep Convolutional Generative Models. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6175–6183. https://doi.org/10.1109/CVPR.2019.00634

Technical Artist Bootcamp: Introduction to Proceduralism - YouTube. (2019, May 3). YouTube. https://www.youtube.com/watch?v=-LHj7-kROoo

Tutenel, T., Bidarra, R., Smelik, R. M., & de Kraker, K. J. (2009). *Rule-based layout solving and its application to procedural interior generation*. 10.