MyRoom: A Unity Plugin for Procedural and Interactive Indoor Scene Synthesis

Haocheng Du*, Yunlong Zhao*, Shuo Huang*, Jiayu Bai*, Song Tian†, Jialin Liu*
* *Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation,*
Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China
† *Center for the Humanities, Southern University of Science and Technology, Shenzhen 518055, China*

2023 IEEE Conference on Games (CoG) | 979-8-3503-2277-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/CoG57401.2023.10333189

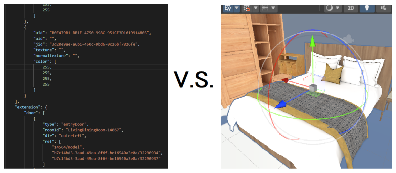
***Abstract*—The demand for indoor synthesis has increased significantly in recent years because of the emergence of computational design. This work designs and develops *MyRoom*, a Unity plugin that can import layout datasets for indoor synthesis and procedurally generate and interactively design indoor scenes. *MyRoom* enables users to easily edit and visualize non-intuitive layout description data, making it easier to generate and manipulate indoor scenes in Unity. *MyRoom* provides a user-friendly interface and powerful tools for designing and optimizing indoor layouts, including an automatic layout generator, making it ideal for game developers, interior designers, and researchers in the digital world-building industry. With *MyRoom*, users can streamline the process of creating high-quality indoor scenes in games and achieve their design goals efficiently.**

***Index Terms*—Room synthesis, furniture layout, procedural content generation, Unity plugin, VR games**

I. Introduction

Game developers are now spending increasing time engaging in interactions within virtual spaces, such as arranging furniture in virtual rooms. Crafting a well-organized room can be a daunting task if designers manually place all furnishings, resulting in a laborious and repetitive undertaking. An automatic furniture layout generation method is crucial for gaming, interior design, and virtual reality applications, as it assists in generating numerous rooms with specific sizes and styles. This type of method can be referred to as indoor scene synthesis [1].

Numerous indoor scene datasets have been developed, such as *SUNCG* [2] and *3D-FRONT* [3], typically in the `JSON` [4] format, describing the type and spacial information about the furniture in a scene. However, data description files are hard to read and there is no standardized method for effectively managing non-uniform and non-visual data across various indoor scene datasets. As a result, users have to design custom importers, viewers, and editors for each dataset to directly view and manipulate scene layout. To reduce this tedious and repetitive work, a standard tool is needed. This paper develops *MyRoom* (https://github.com/BigJIU/MyRoom/), consisting of *Place4Me* and *MyRoomEditor*. The former employs a two-phase generation approach with a machine learning model [5] to synthesize indoor scenes. The latter provides a user-friendly interface to import, visualize, edit, and export layout description files within the scenes in Unity.



```
                255,
                255
            ]
        },
        {
            "uid": "B0E479B1-BB1E-4750-998C-951CF3D1619914803",
            "aid": "",
            "jid": "3d20e9ae-a6b1-450c-9bd6-0c26bf7826fe",
            "texture": "",
            "normaltexture": "",
            "color": [
                255,
                255,
                255,
                255
            ]
        }
    ],
    "extension": {
        "door": [
            {
                "type": "entryDoor",
                "roomId": "LivingDiningRoom-14067",
                "dir": "outerLeft",
                "ref": [
                    "14564/model",
                    "b7c14bd3-3aad-49ea-8f6f-be16540a3e0a/32290934",
                    "b7c14bd3-3aad-49ea-8f6f-be16540a3e0a/32290937"
                ]
```

Fig. 1. Left: dataset layout description. Right: scene imported from dataset.

II. *MyRoom* Plugin

This section describes *Place4Me* and *MyRoomEditor*.

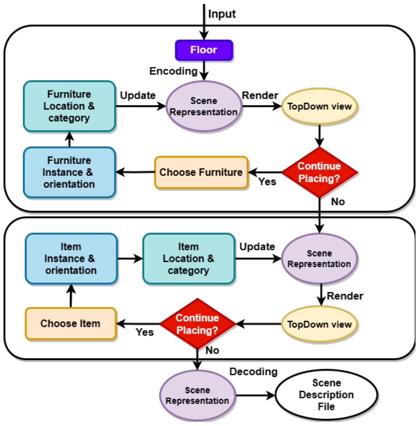


Fig. 2. Process overview of *Place4Me*.

A. Place4Me: Two-phase Indoor Scene Synthesis

Place4Me employs a machine learning model with *DeepSynth* as the primary component to produce a furniture layout from a given incomplete or empty scene. However, different

This work was supported in part by the Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), the Research Institute of Trustworthy Autonomous Systems, the Center for GBA Sci-tech Humanities and Innovation Culture of SUSTech, and the Shenzhen Humanities & Social Sciences Key Research Bases.

Corresponding author: Jialin Liu (liujl@sustech.edu.cn).

979-8-3503-2277-4/23/$31.00 ©2023 IEEE

Authorized licensed use limited to: Southern University of Science and Technology. Downloaded on December 07,2023 at 05:55:50 UTC from IEEE Xplore. Restrictions apply.

to *DeepSynth* which can only generate one-layer furniture on the floor, *Place4Me* presents *Item-on-Furniture* feature that enables the synthesis of small items laying on top of furniture.

We adopt a top-down view same as *DeepSynth* to capture information of object placement. Specifically, a 512 × 512 image with multiple channels is used to represent a given scene. *Place4Me* performs a two-phase procedure to place furniture on the floor (*Furniture-on-Floor*) and then place small items on furniture (*Item-on-Furniture*) as illustrated in Fig. 2. In each phase, three CNNs are trained to perform the following actions. (i) *Continue placing?*: Given the current layout, this model determines whether to continue placing furniture or not. (ii) *Location & category*: Given a layout, it determines a location and a category of furniture to be placed. (iii) *Instance & orientation*: Given a layout and a location to place furniture of a certain category, it determines the actual furniture (i.e., instance) and its orientation.

The same network architecture and parameters as in *DeepSynth* were used. Models were trained on an NVIDIA GeForce RTX 2080Ti GPU. The *Continue placing?* module was trained for 260 epochs, while the others were trained for 160 epochs.

B. MyRoomEditor

MyRoomEditor is a visual interface designed for Unity written in `C#`. We choose Unity as our visualizing interface since it is a common 3D platform that is widely used by developers and learners. *MyRoomEditor* offers three key features: importing dataset, editing scene, and exporting scene (e.g., Fig. 3). With these functionalities, users can easily manipulate layout description data with Unity editor's visual assistant.

The functionalities of *MyRoomEditor* are detailed as follows. (i) *MyRoomEditor* allows to import various layout datasets into Unity scene typically in `JSON` [4] format based on modification from the work of OBJImporter [6]. Users can directly import *SUNCG* [2] and *3D-FRONT* [3] dataset with the plugin. Users can also implement importers for their own layout dataset with the help of the script interface provided in *MyRoomEditor*. The plugin also supports in-time model importing which frees users from manually importing massive 3D furniture models into Unity since *MyRoomEditor* automatically loads the required model in the current synthesizing scene. (ii) *MyRoomEditor* provides users with a visual interface to conveniently view and adjust furniture in the scenes. Users can modify the generated objects' spacial information, including position, rotation, and scale under the Unity Editor window in Fig. 1. Users can also add and delete furniture in the scene. All of the adjustments above are synchronized with the layout description file in the dataset. With Unity's interface, editing the dataset would be simple and intuitive for users. (iii) *MyRoomEditor* provides an easy way to export the modified scene as a standardized 3D object file format (`OBJ` [7]) for other usages.

III. Evaluation and Empirical Study

To assess the quality of scenes generated by *Place4Me*, a perceptual study is employed. To judge the scalability of *MyRoomEditor*, it is integrated into a virtual campus project.

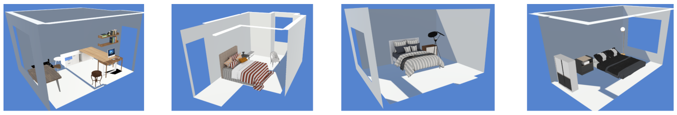


Fig. 3. Scenes generated and imported into Unity with *MyRoom*.

A perceptual study is conducted to evaluate the plausibility of indoor scenes generated by *Place4Me* compared with human-crafted scenes from *3D-FRONT*. 36 volunteers participated in this experiment. Each participant is shown 10 different rendered images of indoor scenes randomly selected from the dataset and generated scenes. Participants need to rate these images from 1 to 5. A higher score means a more plausible room layout. Although scenes from *3D-FRONT* received higher rating scores on average, the difference between the average rating scores of *3D-FRONT* (3.86 ± 0.26) and *Place4Me* (3.27 ± 0.76) is not huge. This indicates that *Place4Me* has the ability to synthesize plausible rooms.

To examine the scalability of *MyRoomEditor*, this paper integrated the plugin into a virtual campus simulator of the Southern University of Science and Technology [8]. The origin project simulates a campus area with raw building models in VRChat, a VR environment. After importing the plugin into the project, it generates several rooms within the buildings on campus. The rooms' spacial information is self-adjusted based on the target room floor, ensuring that their shape and position fit the local world's requirements. This application shows that *MyRoomEditor* is capable of integrating with Unity projects.

IV. Conclusion and Future Work

This paper presents a Unity plugin called *MyRoom* for designing indoor scenes. In the future, we will focus on three aspects: (i) making *MyRoomEditor* compatible with more datasets; (ii) accelerating the generation process of *Place4Me*; (iii) inviting more developers and designers to further assess the experience of using *MyRoom*.

References

[1] S.-H. Zhang, S.-K. Zhang, Y. Liang, and P. Hall, "A survey of 3D indoor scene synthesis," *Journal of Computer Science and Technology*, vol. 34, no. 3, pp. 594–608, 2019.

[2] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576.

[3] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao *et al.*, "3D-FRONT: 3D furnished rooms with layouts and semantics," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 933–10 942.

[4] "Javascript object notation," http://www.json.org/, accessed: March 3, 2023.

[5] K. Wang, M. Savva, A. X. Chang, and D. Ritchie, "Deep convolutional priors for indoor scene synthesis," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[6] Dummiesman, "Objimporter," https://assetstore.unity.com/packages/tools/modeling/runtime-obj-importer-49547, 2019.

[7] "Wavefront obj file format," https://en.wikipedia.org/wiki/Wavefront_.obj_file, accessed: March 3, 2023.

[8] J. Bai, "Virtual campus of the southern university of science and technology based on Unity and VRChat," 2022, undergraduate's thesis.

Authorized licensed use limited to: Southern University of Science and Technology. Downloaded on December 07,2023 at 05:55:50 UTC from IEEE Xplore. Restrictions apply.