D.A.V.E: A Prototype for Automatic Environment Decoration

Callum James Glover[1] and Eike Falk Anderson[1]

[1]The National Centre for Computer Animation, Bournemouth University, UK

Figure 1: *Example of kitchens produced by our prototype.*

Abstract

To fully immerse players in video games, elements of the game world must form a rich, coherent and believable universe, which is time consuming and expensive to achieve manually. We propose an artists' tool for Autodesk Maya to aid environment decoration, allowing artists to focus on the more important elements of the worlds they create.

1. Introduction

Large scale environments in video games are becoming more and more common, with many buildings represented by simple facades with no interior detail. The limiting factor tends to be the cost of artists' effort and time that place a burden on the developer. Richer environments can be produced when artists focus their time on important elements of the environment, with procedural generators automating some of the artists' tasks. To this end we have developed an artist's tool to undertake much of the generic work necessary to populate a scene with objects, acting as a refined canvas for the artist to work on. To prove the viability of the tool, various kitchen layouts have been produced (Figure 1).

2. Previous and Related Work

The automatic placement of props (theatrical 'stage properties' – movable objects in the environment) is an established concept and different approaches exist, recently also including evolutionary and machine learning techniques, leading to increasingly accurate approximations of how humans might place objects around a scene. Nevertheless, artists are often still required to explicitly intervene in the generation of environments, or to correct the results of procedural generation.

The most effective existing approaches such as CAPS [XSF02] employ constraint-based semantic systems, where CAPS also employs 'Pseudo-physics' to achieve physically plausible object placement. A different constraint-based method that incorporates interior design guidelines, ensuring that interiors are both logical and visibly balanced, was introduced by Merrell et al. [MSL*11]. The system analyses user placement of objects and suggests improvements derived from the constraints, also allowing the authoring of new constraints, which can, however, result in room layouts not desired by the artist. More recently, Kán and Kaufmann presented a fully autonomous furniture placement system employing a greedy optimization algorithm [KK18].

Other work has been concerned with procedurally generating entire floor plans for buildings [TBSdK09, LTS*10], or possibly even the buildings themselves [Bra05, Eli17].

3. Artist Directed Automatic Environment Decoration

Our prototype tool acts as an aid for placing props (e.g. furniture) in an existing environment – rooms and buildings placed in the virtual environment by the user. We aim for a simple, computationally inexpensive rule-based method driving an easily extensible tool. The prototype relies largely on pseudo-random number generation for the automatic propagation of objects within selected areas, aiming

© 2019 The Author(s)
Eurographics Proceedings © 2019 The Eurographics Association.

DOI: 10.2312/egp.20191052

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

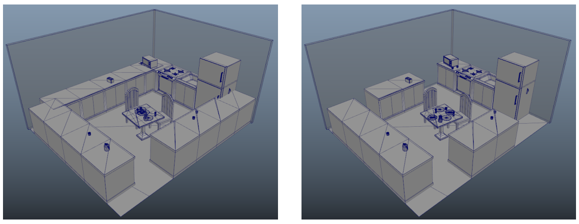

Figure 2: *A particularly disorderly scene generated by our prototype (left) and the results of under a minute of cleanup (right).*

to place objects in plausible locations, and in case of undesirable results, allow users to easily move or delete badly placed objects (Figure 2). The generated numbers are used to offset the positions and rotations of the props within their defined areas, with the exact ‘style’ of object being chosen at random from a given set.

3.1. Identifying Object Shapes

Before placement, the shape of objects used to decorate the environment is determined. Based on user choice, either a convex hull is generated by employing Graham’s scan [Gra72], or a user-created surface acting as the hull is used (in which case all vertices must be numbered successively and lie on a single edge loop). The resulting mesh is then triangulated, becoming a set of many smaller convex hulls. For placing objects around or on top of the resulting hulls, our prototype processes each edge individually rather than the shape as a whole, essentially trying to emulate a human approach – a person would determine how to place chairs around a table based on their distance around the edge, rather than e.g. placing strictly four chairs around an oblong table even if six would fit.

3.2. Semantic Annotation

In order for any decoration to occur, first the prototype must be able to parse the various objects in the scene. Our prototype uses a set of UI elements (Figure 3) to allow users to tag objects and props in the scene. It also employs Maya’s attribute system to expose elements to the user, allowing for overriding of automatic operations to avoid undesirable results. With the UI the user can tag each wall with a building and room identifier and our prototype will use this semantic information to automatically fit appropriate non-wall objects into each room. Each object’s position is tested if it is located within the room – the prototype will deduce the room and building in which an object exists, but occasionally generates false negatives, incorrectly determining objects to not be inside, in which case the UI prompts the user to make a decision. Edges on the floor plan that correspond to doors of the room are identified and ‘locked off’ to prevent objects being placed across them.

3.3. Proof of Concept – Decorating Kitchens

Rule sets for different types of rooms (here: kitchen) use the preprocessed rooms and objects and methodically place props:
If the kitchen has a greater area than 20m^2, a dining table will be placed and decorated in its centre. Randomly spaced runs of counter-tops and appliances are placed with simple constraints, e.g. preventing a gas hob to be placed if an oven with a gas hob already exists. Edges are randomly skipped as a simple method to attain variation. Counter-tops are decorated by placing up to two objects from a set of toasters, kettles, microwaves etc.

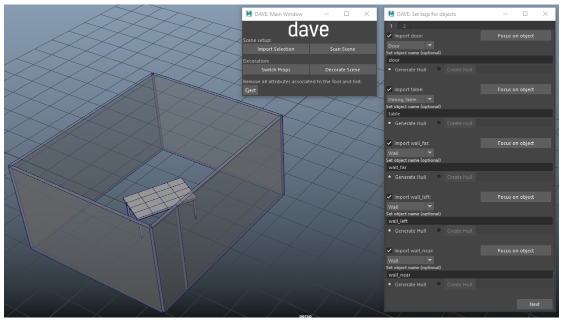

Figure 3: *The UI seen when specifying tags for objects.*

4. Discussion & Future Work

Our proof-of-concept prototype tool generates scenes that appear natural (Figure 1), i.e. with somewhat human-like object placement, which we hope to verify with a future user study. As such, we believe our approach is as effective as existing methods, yet simpler and computationally less expensive. Our prototype provides a framework for a more comprehensive tool – with infrastructure for hull generation, room determination, and interactive semantic annotation (tagging) by the user – and can potentially process thousands of rooms, the bottleneck being user input (setting up three buildings with six rooms takes approximately 4 minutes by the user, whereas the processing takes less than a second). Further automation could improve this (procedural buildings with automatic semantic annotation) and in the future we also plan to extend the system with more complex rule-sets and room descriptions.

References

[Bra05] BRADLEY B.: *Towards the Procedural Generation of Urban Building Interiors.* Master’s thesis, The University of Hull, 2005. 1

[Eli17] ELINDER T.: *General Methods for the Generation of Seamless Procedural Cities.* Master’s thesis, Lund University, 2017. 1

[Gra72] GRAHAM R. L.: An efficient algorith for determining the convex hull of a finite planar set. *Information Processing Letters 1* (1972), 132–133. 2

[KK18] KÁN P., KAUFMANN H.: Automatic furniture arrangement using greedy cost minimization. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2018), pp. 491–498. 1

[LTS*10] LOPES R., TUTENEL T., SMELIK R. M., DE KRAKER K. J., BIDARRA R.: A constrained growth method for procedural floor plan generation. In *Proceedings of GAME-ON 2010: the 11th International Conference on Intelligent Games and Simulation* (2010). 1

[MSL*11] MERRELL P., SCHKUFZA E., LI Z., AGRAWALA M., KOLTUN V.: Interactive furniture layout using interior design guidelines. In *ACM SIGGRAPH 2011 Papers* (2011), pp. 87:1–87:10. 1

[TBSdK09] TUTENEL T., BIDARRA R., SMELIK R. M., DE KRAKER K. J.: Rule-based layout solving and its application to procedural interior generation. In *Proceedings of the CASA’09 Workshop on 3D Advanced Media in Gaming and Simulation (3AMIGAS)* (2009), pp. 15–24. 1

[XSF02] XU K., STEWART J., FIUME E.: Constraint-based automatic placement for scene composition. In *Proceedings of the Graphics Interface 2002 Conference* (2002), pp. 25–34. 1

© 2019 The Author(s)
Eurographics Proceedings © 2019 The Eurographics Association.