Project DisCo: Choreographing Discrete Building Blocks in Virtual Reality

Jan Philipp Drude[1], Andrea Rossi[2]
and Mirco Becker[1]

[1] Leibniz University Hannover, Welfengarten 1, 30167 Hannover, Germany
[2] TU Darmstadt, Karolinenplatz 5, 64289 Darmstadt, Germany
drude@iat.uni-hannover.de

Abstract. Current excursions within architectural research are exploring the potential of discrete design strategies at different scales. Starting with the introduction of the Great Invention Kit (GIK) and the subsequent development of reversible 3D printing processes based on "digital materials" at the MIT Center for Bits and Atoms [1] similar concepts of additive manufacturing have recently entered the field of architecture. This development hints at the potential for new reversible fabrication methods [2], as well as new ways to define architectural shapes as bottom-up syntactical aggregations of modular building blocks.

Within this emerging field of "Discrete Architecture", Gilles Retsin showcases prototypical architectural designs with his Diamond House among other projects [3], also focusing on the possibilities for robotic assembly, while José Sanchez explores techniques borrowed from game-design to define loose assemblies based on their specific "topological diagrams" [4].

This paper introduces Project DisCo (Discrete Choreography), an application to integrate bottom-up aggregation of modular building blocks and intuitive spatial design into Virtual Reality (VR). The work presented here builds on Sanchez's approach to discrete interactive design within gaming environments, though it is neither based on a sequential placement of individual parts, nor does it utilize static vector fields. In contrast, it allows the designer to choreograph large amounts of building blocks interactively through physics simulations as a means of form generation.

Keywords: Virtual Reality, discrete assemblies, digital material.

1 Introduction

1.1 Virtual Reality

With the advent of commercial VR headsets (Oculus Rift 2012), VR has opened up to a mass audience. While mainly focused on the gaming market, some tools for geometric modeling have emerged as well. Concept modeling tools like gravity sketch® are advanced applications and deliver quick results to the skilled designer, whereas drawing tools like Google's Tilt Brush® have enthused people, especially in the consumer segment.

The ability to paint in three dimensions, exploiting full body movement, quickly leads to spatial results, although the inherent lack of precision makes it unusable for anything but concept models. Additionally none of these existing tools is tailored for architectural design and, while some architectural research is being done, both in the context of real-time simulation [5], and in their didactic potential [6], the choice of tools for the creation of architectural objects, building on the capabilities of VR and game-engines, is relatively sparse.

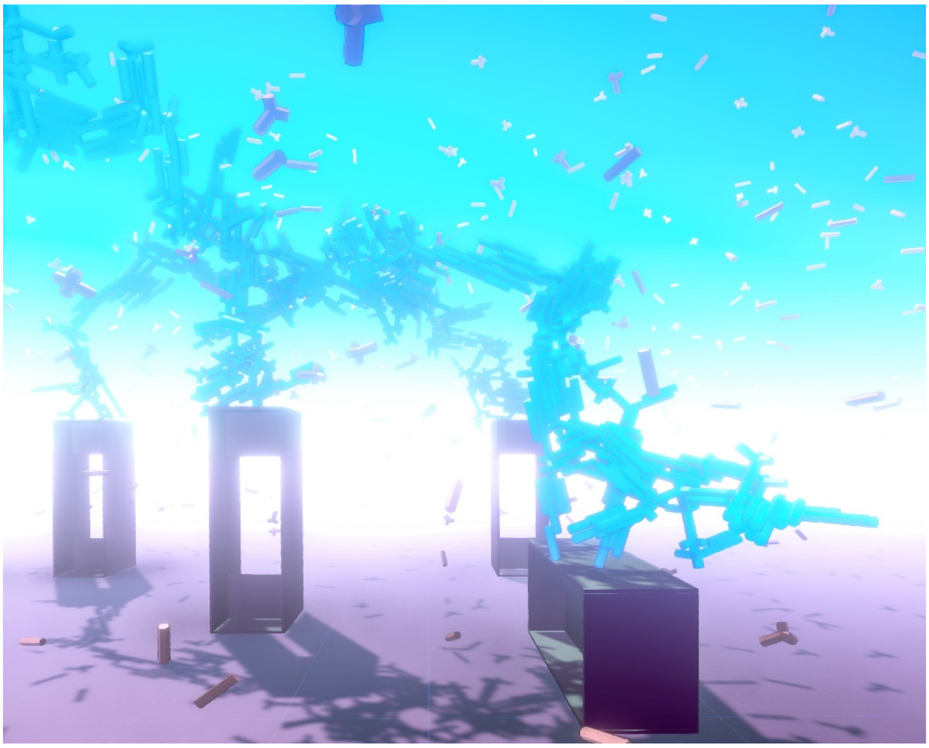

Fig. 1. Project DisCo: Screenshot

1.2 Proposal

To fill this gap, we propose a novel way of generating architectural aggregate structures inside VR: A choreographing methodology, in which the designer/choreographer is allowed to use intuitive physics-based interactions to assemble discrete modular structures.

This assembly mechanism is achieved by allowing interaction with a field of disconnected digital building blocks, initially floating in space freely. By using controller movements to apply forces to them intuitively, the building blocks are aggregated according to their inherent connection logic, allowing the choreographer to control the overall process while leaving the exact detailing of the assembly to a combination of pre-defined assembly rules and chance since the sequence of realized connections is computed by the system.

This process of modeling with virtual physics formally harks back to Greg Lynn's thesis of "animate form" [7], though the use of discrete building blocks allows embedding the discretization needed for assembly directly within the design process, avoiding the need for complex post-rationalization of the generated form.

2 Project DisCo

2.1 Designing with Digital Material in (programmatic) CAD: Wasp

The initial definition of building blocks and assembly rules builds on Wasp, a Grasshopper plug-in for discrete modeling [8]. Wasp enables the creation of descriptions of each discrete building block, combining geometric and topological information, as well as the definition of the assembly rules, used to aggregate structures out of basic units. Building blocks are defined by their geometry as 3D meshes, and by their connections to other blocks as oriented planes on the faces of the part geometry. Assembly rules are defined as syntactic statements, specifying which building blocks and which connections are allowed to connect during aggregation (see Fig. 2). Users are allowed to aggregate structures by applying different algorithms. The main limitation of the approach proposed by Wasp lies in the fact that design is either fully bottom-up and dependent on the aggregation rules (when using stochastic aggregation procedures), or entirely pre-determined in a top-down fashion by a guiding scalar field, based on input geometries or generated density fields. These methods pose limitations to the application of Wasp in a flexible design process, where intuitive modeling is often at odds with rule-based processes [9].

Intending to overcome such limitations, we propose a combination of Wasp's discrete modeling approach with intuitive design in VR, strengthening the possibilities of both systems: this improves the accessibility of modular design approaches, by offering a user-friendly interaction through VR. Concurrently this approach allows for a more direct link between design in VR and fabrication, by embedding the assembly logic directly into the design workflow. The choreographer is enabled to create the discrete system's connection logic in the familiar design-environment of Rhino and Grasshopper® and then export the required information, including both parts geometry and assembly rules, to our VR modeling tool Project DisCo. The export format follows the same conventions of Wasp, using a .json file for storing aggregation information. This allows full bi-directional communication between Wasp and DisCo.

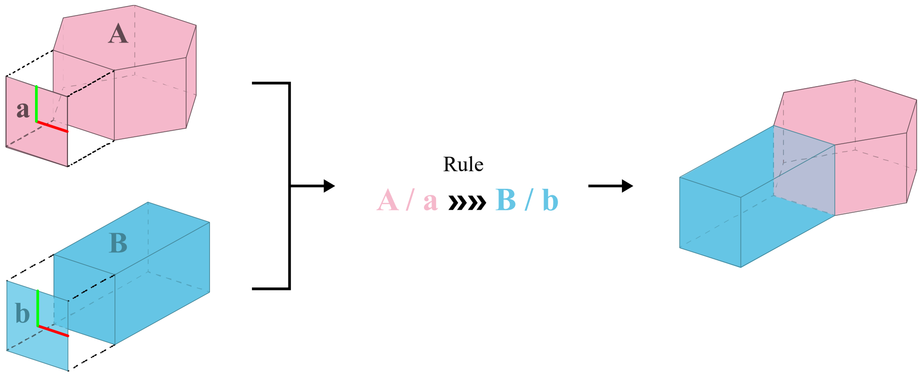

Fig. 2. Wasp Rules, Connections, and Geometry

2.2 Choreographing Digital Material in VR: DisCo

To tightly integrate discrete modeling with a more direct and intuitive design approach, Project DisCo combines the connectivity information provided by Wasp with physics-based interaction gameplay, allowing to generate discrete spatial structures in a VR modeling environment quickly.

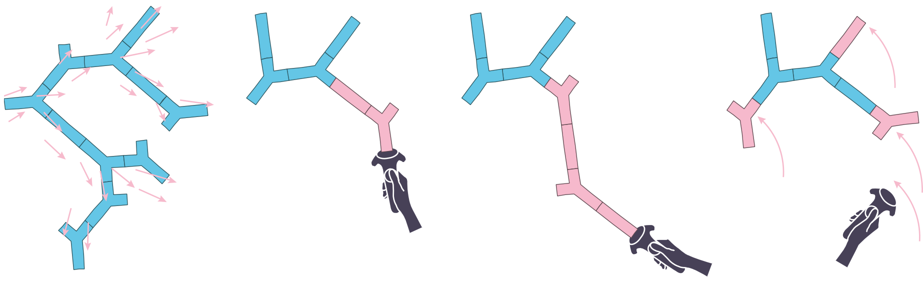

Fig. 3. Vector Field – Direct Placement – Recursive Placement – Choreography

Gameplay. Within DisCo, the choreographer is prompted with a number of building blocks floating around in space freely. The default aggregational tool allows the choreographer to position blocks at specific locations as starting points for an assembly.

The primary means of aggregation is the choreography mode, where a set number of free building blocks become responsive to the movements of the controller. Its direction and velocity apply forces to the responsive blocks, where they are affected proportionally to the speed of the controller itself. This responsiveness will result in a flock-like behavior that reacts to the players steering movements, resulting in a globally predictable system, though non-predictable local parts behavior (see Fig. 4).

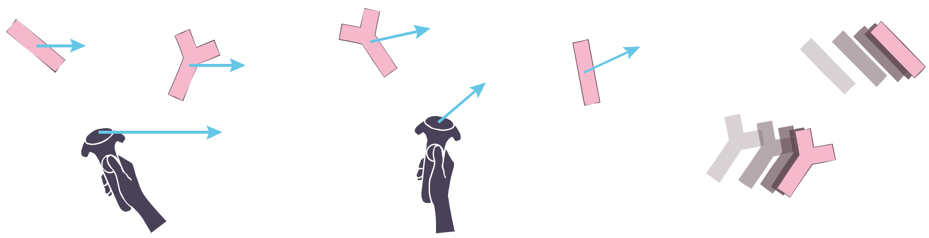

Fig. 4. Controller Direction – Controller Velocity – Drag

Responsive building blocks continuously scan their surroundings for open connections close-by, snapping to them if conditions of proximity and alignment are met (see Fig. 6).

To preserve computer performance, the choreographer can control the number of affected parts as well as disable specific assembled parts for connection, also resulting in a mode for influencing the direction of growth.

Further aggregational tools for guided growth are several filters. The choreographer can disable connection rules following their associated rules grammar, created in Wasp or select types of building blocks to remain unaffected altogether. Aggregational tools can be chosen from a tool menu attached to the controller. (see Fig. 5)

Based on Wasp class libraries, DisCo is programmed in Unity software[10] and adds functionality for real-time behavior and physics, while also defining an exchange pipeline.

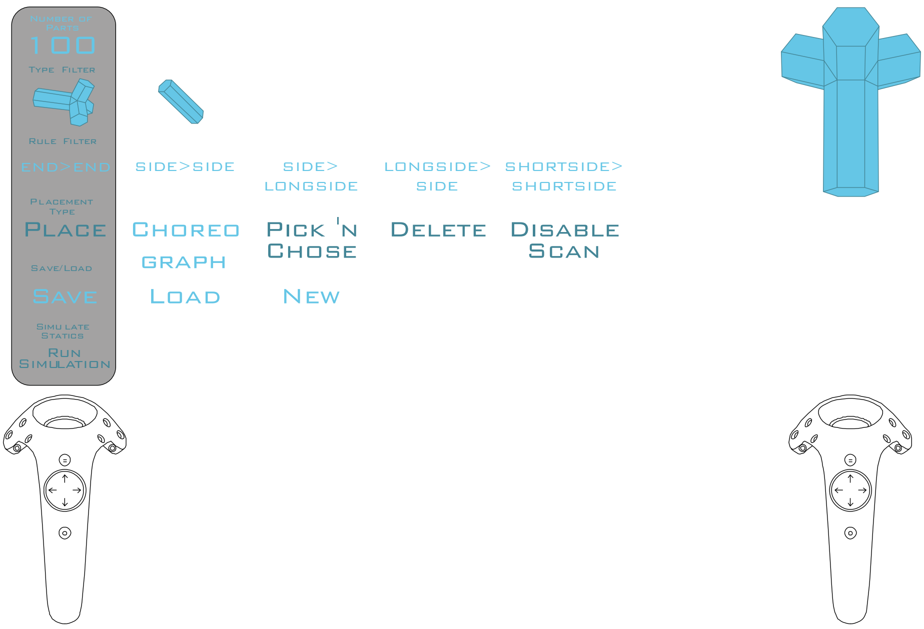

Fig. 5. Controller Menu for HTC Vive®

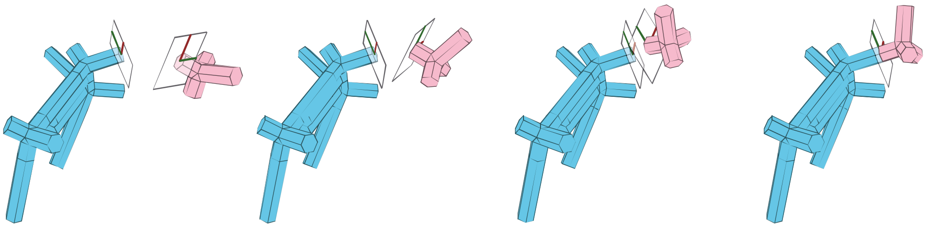

Fig. 6. Process of Connection Snapping

Connection Scanning. Before the start of a session, the choreographer is required to set up the game area as well as a dimension for its division into a voxel grid, to represent this space-partition in a three-dimensional array, where connections are stored according to their voxel position as soon as a part is placed in the aggregation.

An affected part can thus scan for open connections by referencing its position to an item in the array and only get the free connections stored there. (see Fig. 7) The stored connections are then checked against the connections on the affected parts itself, looking for the best fit according to proximity and alignment as well as active rules and rules grammar. (see Fig. 8) If the best valid fit delivers a combined value of proximity and alignment below a predefined threshold, the part attempts to realize the connection. (see Fig. 6) For this, it needs to check against collisions in its new position, only realizing it, when there is no overlap with the adjacent parts. In the case of a collision event taking place, the part is transformed back to its previous position, and the rule is deactivated for future connections.

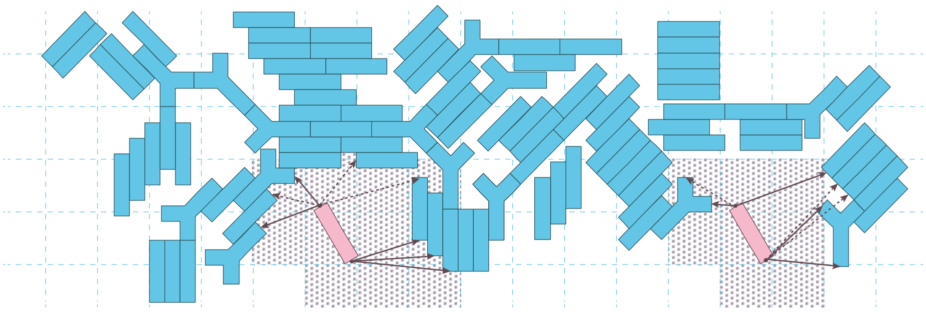

Fig. 7. Space-Partition Grid

Pipeline. The exchange pipeline with Wasp is fully bidirectional. After designing the part geometry in Rhino and the connection syntax in Grasshopper Wasp, the choreographer exports a file from a Grasshopper node to be interpreted by DisCo. This exporter stores all relevant information from Wasp. Contained in this file are both the parts with their names, IDs, and connections, as well as valid assembly rules. Furthermore, it includes mesh geometry information for both representation and collision. DisCo builds the meshes for these purposes during the initialization phase, while interpreting the exchange file.

Conversely, once aggregation is completed, DisCo stores a choreographed assembly in a file that can either be reloaded at a later point or imported into Wasp. This workflow allows the choreographer to work on multiple assemblies that are then loaded into one project, as well as working with a simplified version of the part's geometry, that can be exchanged for a higher resolution model after assembly. An exported assembly can also serve as a starting point for further aggregation in Wasp. Assembled base geometry can thus be thickened and grown with a stochastic algorithm in Wasp, growing a roughly sketched spatial assembly in complexity.

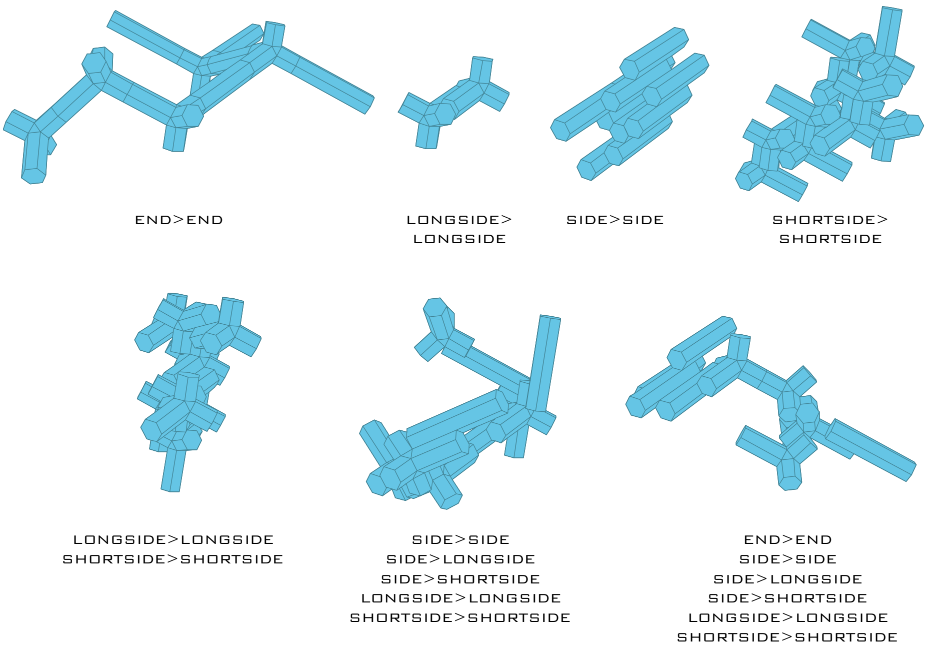

Fig. 8. Connection Filters

Simulation. As a means of testing an assembly for real-world buildability, DisCo features the ability to apply gravity to the aggregated geometries, exposing the complete aggregation to gravitational pull and letting it fall to the ground, to test its stability. (see Fig. 9) The result can be chosen as the new status quo upon which further to aggregate. The ability to quickly simulate rigid-body physics during the aggregation process offers an efficient tool to test the stability of a structure not only when completed, but also during the assembly phase, providing relevant insights regarding the need for scaffolding or counter-balancing blocks during real-world construction.

The physics simulation capabilities inherent to game engines will further be tested in the future. The aim will be to implement structural analysis of assemblies to check forces occurring at the joints.

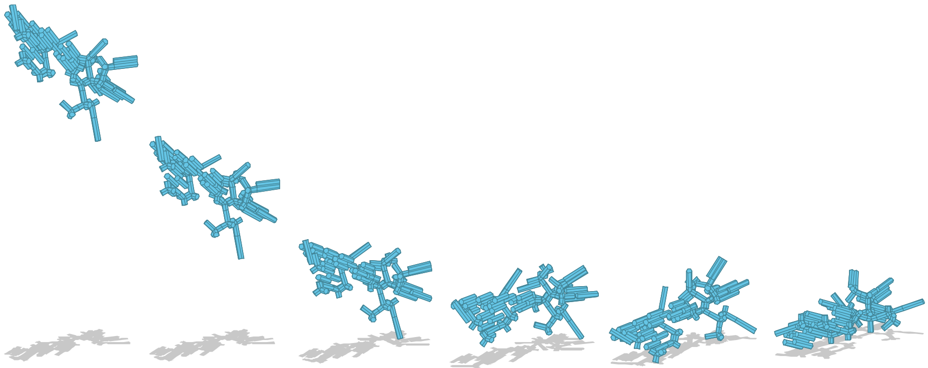

Fig. 9. Simulation Process

Surroundings. In addition to the discrete building blocks and their associated connections and assembly rules, the pipeline also allows for the import of additional geometry. The choreographer can thus create a design space in the otherwise disassociated surroundings inside DisCo.

As well as being a scale reference or an inspirational space, this entails the possibility of building upon modeled parts of the real surroundings.

In combination with the physics simulation, this makes it possible to include fabrication constraints during aggregation, since it allows to model the physical surroundings of a real construction site and use the simulation to test stability and location of support points.

It also can help the degree of immersion of the choreographer, since it adds the possibility of interaction with the physical space around them. This could also mean directly assembling at the construction site, not only seeing geometrical constraints but also experiencing the space with other senses.

These sensations would, of course, be amplified by the use of mixed reality glasses. Though the existing systems do not have the computational power to run the real-time connection scanning in DisCo, this change of platform can be a goal for the future.

2.3 Classification

While unique in its application, Project DisCo bears similarities to other existing systems. Examples for this can be found in existing software solutions as well as in everyday Objects that follow discrete connection logic.

The LEGO® reference. [11] One of the simplest and most popular systems for discrete building blocks is offered by the LEGO Group. While following detailed instructions towards a finished model is rather easy, depending on the complexity of the build, it is much harder to design a meaningful LEGO assembly. While children do it hands-on, the LEGO Group also offered a piece of software for that very purpose. The LEGO LDD software was a design suite, where users could create LEGO assemblies in a

CAD-like manner, create the instructions and both upload your design and order the necessary parts for your build online [12].

Though indeed a gift for enthusiasts, this approach has two huge disadvantages. One is that it shares with physical LEGO blocks the work-intensive assembly process of looking for the needed piece and then one after the other placing it in the right position. The second is the reduction of geometric complexity to isometric views. This problem is even more relevant for any building blocks leaving the guidance system of an orthogonal/cartesian grid.

In the field of discrete architecture, designers have attempted to overcome such limitations by using different computational ideas to aggregate parts. Methods of growth range from purely stochastic algorithms to more elaborate parametric approaches using vector fields or geometry as directional guides.

DisCo wants to add a further more playful and intuitive approach to this challenge. Basically a system, built around connection syntax, it aims at creating assemblies very much like the LEGO LDD software, without the rigidity of CAD, but as an engaging game where one can immerse oneself in the act of making. The most critical disadvantage of traditional CAD certainly is the lack of three-dimensional representation that VR offers to solve. Furthermore, only the introduction of a system where we can aggregate parts much faster than in serially building with single units makes it a viable option for designing larger structures.

The pace in which a design can be generated in DisCo ranks somewhere between direct placement and generative solutions, though it offers a format where the choreographer can be closer and have increased control over the actual process of aggregation than with conventional computational approaches.

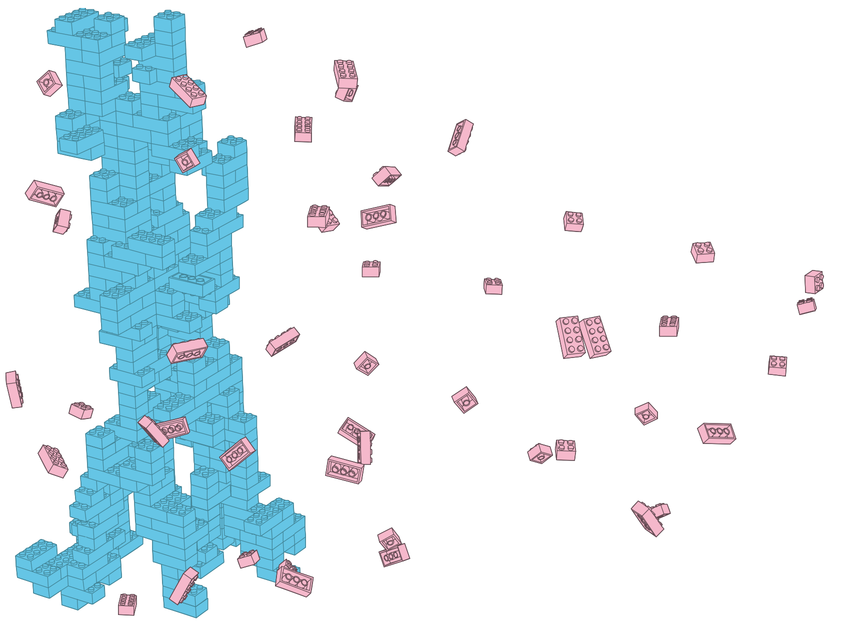

Fig. 10. LEGO® block aggregation

The multibody CAD reference. Another reference to the workings of DisCo would be 3D CAD with multibody systems, like the ones used in mechanical engineering. Mainly an assembly is composed of several rigid bodies constrained to one another via rigid joints. In mechanical engineering, a big part of the design process is utilizing off the shelf parts to cut cost. Assembling a system out of given parts by constraining them to one another at assigned connections bears quite some similarities to DisCo. If we consider the LEGO® approach of placing one building block after another, any multibody CAD tool could be used to achieve an assembly like with DisCo more laboriously.

Assigning degrees of freedom to a connection could be a feature for a future release of DisCo. This feature would make the modeling of kinematic systems possible, either for the choreographer to be able to change an aggregation or as a means of creating animated systems.

Towards an open platform. To be a DisCo choreographer, users do not necessarily need to be proficient with Rhino and Grasshopper. Project DisCo will be provided with multiple example files, ready to be used to choreograph an aggregation. In the future, it could well be possible to exchange aggregational systems directly via an associated online platform and crowdsource design ideas from an audience that is not concerned with the system's creation, mirroring José Sanchez's concept of "platform design" [13].

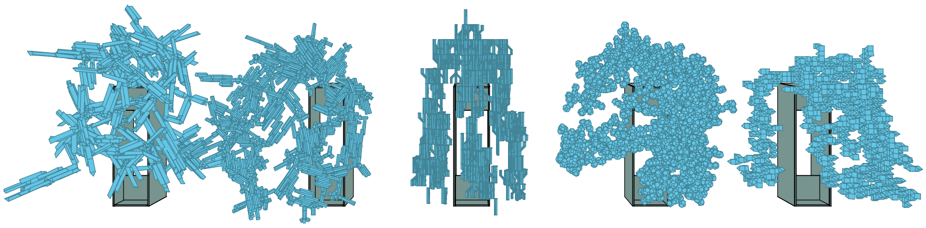

Fig. 11. Example Aggregations

2.4 Application

The discreteness of the system makes it possible to have analog equivalents of the digital building blocks with the connection logic embedded into their geometry. Through mass production, many of these building blocks for use in different architectural contexts, could be produced in automated manufacturing processes and be brought to the construction site for assembly.

Since the accuracy of the system is embedded into the discreteness of the parts' logic, all manners of different scales are imaginable. From the assembly of small structures by hand with only a digitally controlled guidance system, all the way to the manufacture of large structures by swarms of autonomous robots, different means of construction is imaginable [14]. Since the discreteness of the aggregation can act as a guidance system through its geometry, small robots could assemble it without expensive sensing mechanisms.

This process not only has the possibility to achieve a fully automated construction site but could also lead to a democratization of construction, since groups of nonprofessionals could erect even complex structures out of mass-produced parts with embedded assembly logic.

3 Outlook

Project DisCo, in its current version, will be tested in depth for its design applications in the following semester at our M.Sc. program. It is to be understood as a first alpha version that will be further developed. Improvements in both performance and interaction are already put into a roadmap, and a first public release is available online [15]. The roadmap also includes additional functionality such as the physics simulations mentioned above, an exchange platform for discrete systems and their associated Rhino and Grasshopper files and tools for guided construction. The DisCo export node is also embedded directly in the current Wasp releases to enable users to migrate designs to VR quickly. [16]

Future alterations will bring the application from the virtual into reality by incorporating structural analysis methods as well as fabrication-aware methodologies.

Fig. 12. Assembly Rendering

References

1. Popescu, G. A., Kunzler, P., Gershenfeld, N.: Digital Printing of Digital Materials. In: Digital Fabrication 2006 Final Program and Proceedings, pp. 55-57. Society for Imaging Science and Technology, Denver, Colorado (2006).
2. Jenett, B., Cellucci, D., Gregg, C., Cheung, K.: Meso-Scale Digital Material: Modular, Reconfigurable, Lattice-Based Structures. In: Proceedings of the 2016 Manufacturing Science and Engineering Conference MSEC2016, ASME, Blacksburg, Virginia (2016)
3. Retsin, G.: Discrete Assembly and Digital Materials in Architecture. In: Herneoja, A., Österlund, T., Markkanen, P. (eds.) eCAADe 34 – COMPLEXITY & SIMPLICITY, vol. 1, pp. 143-151. eCAADe, Brussels (2016).
4. Sanchez, J.: Polyomino – “The Missing Topology Mechanic”. In: Kretzer, M., Hovestadt, L. (eds.) ALIVE – ADVANCES IN ADAPTIVE ARCHITECTURE 2013, Applied Virtuality Book Series, vol. 8, p 8. Birkhäuser Verlag GmbH, Basel (2014).
5. Quinn, G., Galeazzi, A., Schneider, F., Gengnagel, C.: StructVR Virtual Reality Structures. In: Mueller, C., Adriaenssen, S. (eds.) IASS 2018 BOSTON SYMPOSIUM: APPROACHES FOR CONCEPTUAL STRUCTURAL DESIGN, Proceedings of IASS Annual Symposia, vol. 2018, pp. 1-8. IASS, Boston (2018)
6. Jovanovic, D.: Platform Sandbox – A pedagogical design software. In: Lamere, J., Parreño Alonso, C. (eds.) ACADIA 2017 – DISCIPLINES DISRUPTION, Projects Catalog of the 37th Annual Conference of the Association for Computer Aided Design in Architecture, pp. Acadia Publishing Company, Cambridge, Massachusetts (2017).
7. Lynn, G.: Animate Form. 1st edn. Princeton Architectural Press, New York City (1999).
8. Rossi, A., Tessmann, O.: From Voxels to Parts: Hierarchical Discrete Modeling for Design and Assembly. In: Cocchiarella, L. (eds.) ICGG 2018 – PROCEEDINGS OF THE 18th INTERNATIONAL CONFERENCE ON GEOMETRY AND GRAPHICS, AISC vol. 809, pp. 1001-1012. Springer, Milan (2018)
9. Johns, R.L.: Augmented materiality: Modelling with material indeterminacy. In: Gramazio, F., Kohler, M., Langenberg, S. (eds.) FABRICATE 2014: NEGOTIATING DESIGN AND MAKING, pp. 216-223. Gta Verlag, Zurich (2014).
10. These materials are not sponsored by or affiliated with Unity Technologies or its affiliates. “Unity” is a trademark or registered trademark of Unity Technologies or its affiliates in the U.S. and elsewhere.
11. LEGO® is a trademark of the LEGO Group of companies which does not sponsor, authorize, or endorse this material.
12. LEGO Group: LEGO Digital Designer (2012), http://ldd.lego.com/, last accessed 2019/03/29.
13. Sanches, J.: Architecture for the commons. Architectural Design 89(2), 22-29 (2019).
14. Gershenfeld, N.: Programming a new reality. In: TEDxCERN (2015), min. 8:00. https://www.youtube.com/watch?v=EA-wcFtUBE4, last accessed 2019/04/02.
15. Project DisCo is available at http://www.project-disco.com
16. Wasp is available at https://www.food4rhino.com/app/wasp