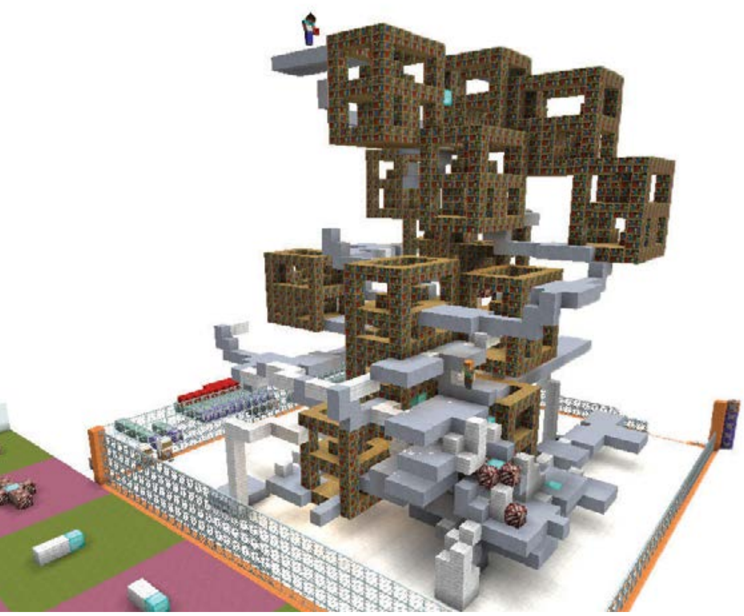
20,000 Blocks

Can gameplay be used to guide non-expert groups in creating architecture?

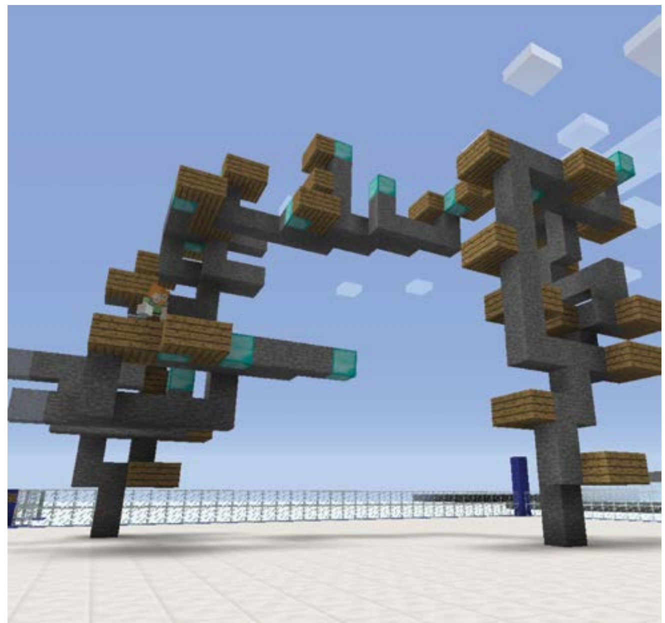
Anton Savov
Digital Design Unit (DDU),
Technical University Darmstadt

Ben Buckton
Invent the World

Oliver Tessmann
Digital Design Unit (DDU),
Technical University Darmstadt



1



2

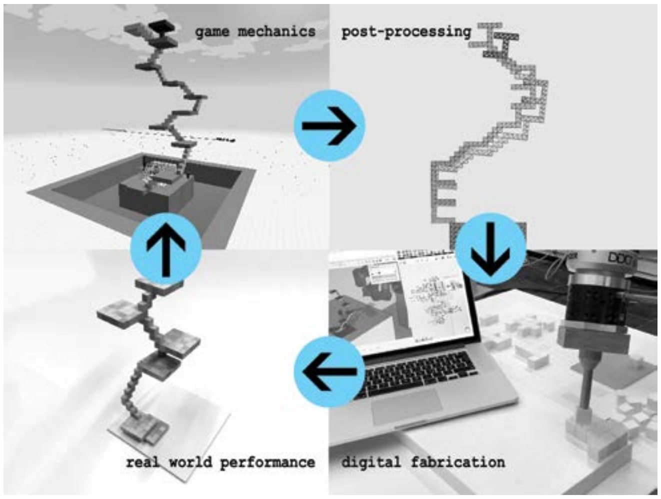
ABSTRACT

The paper follows research in engaging groups of non-trained individuals in the creation of architectural designs using games and crowdsourcing for human-directed problem-solving. With the proposed method, architectural experts can encode their design knowledge into custom-developed multiplayer gameplay in Minecraft. Non-expert players then are constrained by this gameplay which guides them to create unique architectural results. We describe a method with three components: guiding rules, verification routines and fast feedback. The method employs a real-time link between the game and structural analysis in Grasshopper to verify the designs. To prove the viability of these results, we use robotic fabrication, where the digital results are brought to reality at scale. A major finding of the work is the suite of tools for calibrating the balance of influence on the resulting designs between the Experts and the Players. We believe that this process can create designs which are not limited to parametrically optimal solutions but could also solve real-world problems in new and unexpected ways.

1 A 20,000 Blocks game map titled HOW HIGH – created by Andrea Quartara, Marc Emmanuelli and Ulrik Montnemery during our cluster at SmartGeometry 2016.

2 A 20,000 Blocks game map titled WOULD YOU MIND YONA FRIEDMAN? – created by Marios Messios and Max Rudolph during our cluster at SmartGeometry 2016.

3 The Project 20,000 Blocks links participatory design, automated structural analysis and robotic fabrication.



3

PROBLEM STATEMENT

The authors pose the hypothesis that a better built environment could come into being if digital technology is used to include the non-experts, together with the experts, in the architectural design process. Arguments in favor of participatory design methods are not new in architecture and vary from increasing speed of design and construction, through fulfilling the preferences of inhabitants, to building cohesive urban communities (Friedman 1975, Parker 2014).

For a meaningful inclusion of the non-architect in the design process, the architectural expert knowledge needs to be partially encoded into an agile, open and generative rule-based system. For this we needed a modeling environment that combines the ease and popularity of computer games with the control and evaluation tools of Computer-Aided Architectural Design (CAAD) software. In this paper, we present such a system with three key components:

1) ***Guiding rules*** employed to direct the participants towards feasible design solutions.
2) ***Verification routines*** to automatically process the resulting designs on key parameters and performances.
3) ***Fast feedback*** to keep the players aware of what is happening, allowing for decision corrections to happen in short cycles.

Our aim is to let online communities use a game to prototype and test various building designs in short cycles. Our approach can be categorized as crowdsourcing. Daren C. Brabham defines crowdsourcing as "an online, distributed problem-solving and production model that leverages the collective intelligence of online communities to serve specific organizational goals. Online communities, also called crowds, are given the opportunity to respond to crowdsourcing activities promoted by the organization, and they are motivated to respond for a variety of reasons." (Brabham 2013).

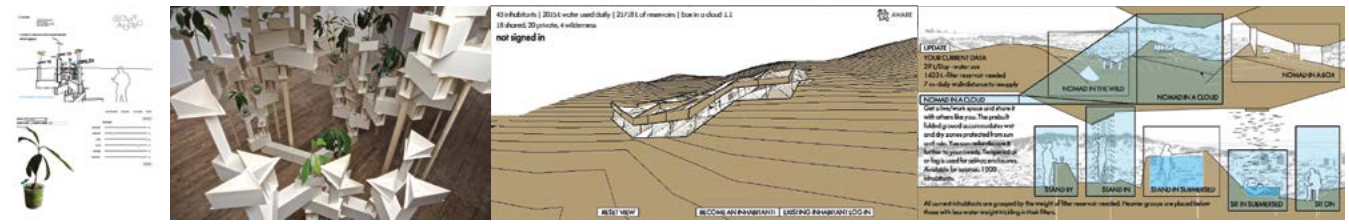
Games are accessible to people of all ages and backgrounds and can process large amounts of data input from many players. Hence our approach seeks to crowdsource within a game environment, particularly focusing on online multiplayer games. CAAD software, on the other hand, takes input from specialists and is capable of advanced performance analysis as well as controlling robotic fabrication. Verification routines based on CAAD software are needed to test the reliability of a multitude of architectural designs created by the non-trained crowd.

Crowdsourcing through gaming, seen as a means of user engagement and problem solving, is gaining ground in the sciences with projects such as FOLD.IT and Eve Online's Project Discovery, and recently in architecture with the work of Jose Sanchez, in particular the game Block'hood (Cooper et al. 2010, Szantner 2016, Sanchez 2015). At the same time, Minecraft (www.minecraft.net) has turned from a game into a phenomenon, being used in initiatives such as "Block by Block" to help the United Nations Habitat involve local communities in the refurbishment of their public spaces (BBC News 2012).

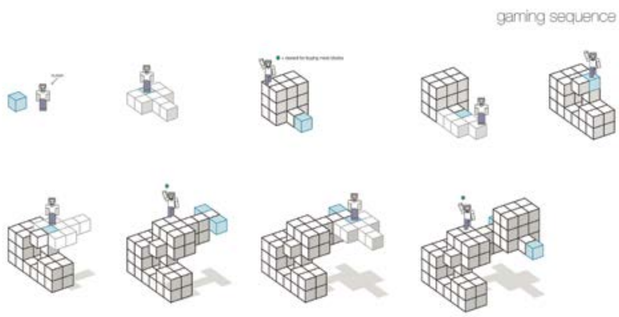
To assess our hypothesis, we set up an experiment, under the project ***20,000 Blocks Above the Ground*** (www.20000blocks.wordpress.com), which links multiplayer gaming, participatory digital design tools and robotic fabrication. (Figure 3)

The research project is run at the ***Digital Design Unit (DDU)*** at Technische Universität Darmstadt and started with the question: "Can gameplay mechanics be used to guide groups of non-experts throughout the collaborative creation of architectural designs?" Anyone can design a building in ***20,000 Blocks Above the Ground*** and have it 3D printed by our robotic arm Ginger. We use a game called ***Minecraft*** that is played with simple graphics and rules. It is a game where you build and demolish cubes of 1x1x1 meters. Currently more than 40 million people play it (Warren 2016).

Minecraft is a sandbox game where players can choose on their own which goals and adventures to pursue. The game consists of a 3D procedurally generated world where materials are spread for the players to mine. Players can combine various materials to "craft" new objects such as pick-axes, wooden planks, buckets, doors etc. Each Minecraft world can be loaded on a server and made available for online access to other players offering multiplayer functionality. Minecraft's rich catalogue of materials and objects plus the ability for in-game scripts allows the creation of custom maps where a goal or an adventure can be defined by



4 Project Avocado (first two from left) and Box in a Cloud.



5

6

7

the map creator. The project ***20,000 Blocks*** uses such a custom-made map to define the game rules leading groups of players through the creation of architectural designs.

Minecraft's abstract, non-photorealistic, voxel world makes it a good choice for our research, based on Fröst and Warren's conclusion that "low-detail of sketch-like real-time 3D models often promotes creativity and discussion"(Fröst and Warren 2000). The make up of a Minecraft world out of 1 meter large blocks offers a suitable resolution for mass-modelling of architectural concepts from the scale of a room up to urban planning. Smaller scale objects such as the details fenestration or furniture are not easily possible in Minecraft but are also irrelevant to our research. Our focus is rather on crowd motivation and guiding techniques for the production of initial, schematic designs.

Both Peter Fröst and the team of Achille Segard have conducted similar experiments in architecture using the game engine of Half-Life (Fröst 2003, Segard, Moloney, and Moleta 2013). Their work is however targeted mostly at architects and focuses entirely on the design's shape. Claudia Otten has conceptualized a house-making game tool for single player use, targeted at non-architects (Otten 2014).

In our previous work, particularly ***Sensitive Assembly*** (2015), we encountered the power of the players as topology optimization agents (Savov, Tessmann, and Nielsen 2016). ***Project Avocado*** (2012) and ***Box in a Cloud*** (2013), on the other hand, have successfully allowed participants to customize a collectively designed building to their individual preferences (Figure 4) (Savov 2012, 2013).

The project ***20,000 Blocks above the Ground*** builds upon these references and spans the digital and the material worlds. It combines a user-friendly game environment as a front-end with the precise performance feedback from specialized parametric design tools.

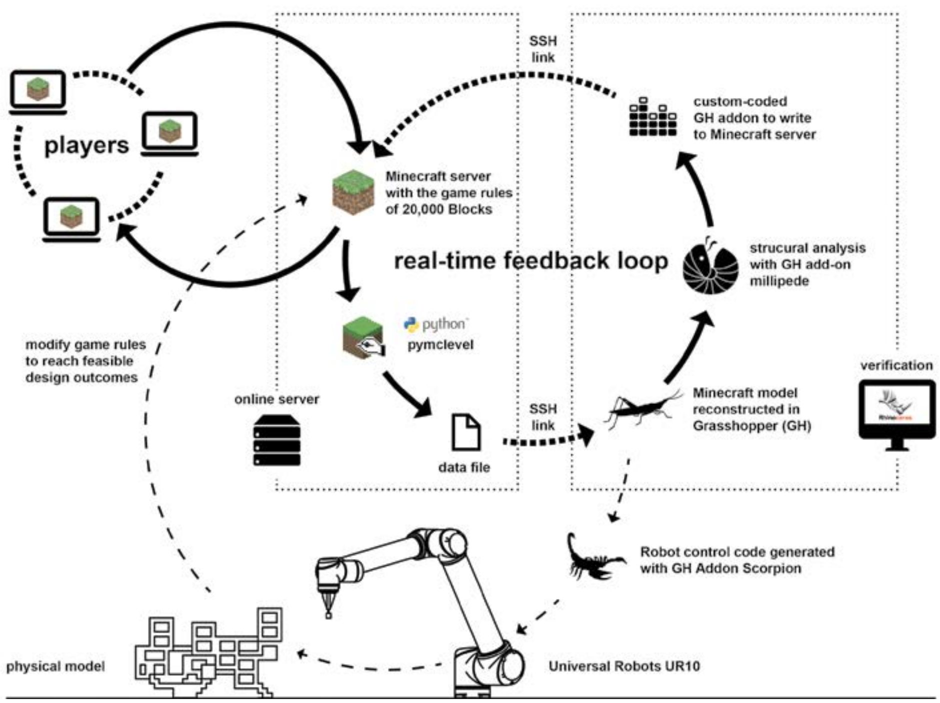
METHOD

Our method places importance on constraining the player choices through game mechanics and immersing players in the gameplay to increase motivation and investment. We consider there to be three key roles involved in this process:

The first role is of project ***Organizers***. As researchers this is the role we take, bringing together the architects, the game designers, the players and the roboticists. We define the technology so others can use what we make to generate designs.

The second role is the ***Experts***, whose knowledge will be encoded. These are the people setting and modifying the vocabulary and defining the goal. The game industry analogy would be of a Level designer, who is in charge of creating the challenges within a game.

Players are the third role. The experience of the players is tightly controlled — the Organizers decide which materials they can



8

5 The vocabulary-based rewarding leads to a player-controlled growth process of the design (drawing by Andrea Quartara).

6 The robotic fabrication process in 20,000 Blocks (drawing by Jörg Hartmann).

7 Players need to build in such a way so they can walk up and down to buy more materials from the villagers. The main goal is modeled in red and when reached the game is saved and reset.

8 Real-time link between the Minecraft game of 20,000 Blocks and structural verification done in Grasshopper/Millipede.

place and break, and where they can walk. The players are given a goal but insufficient resources to achieve it. To progress, players build shapes out of Minecraft blocks, choosing from an architectural vocabulary defined by the Experts. Players are rewarded with resources for building one of the shapes. While players compete to reach the goal, a building emerges out of the shapes that they have built.

To steer the players, we can modify the three components of the method: the guiding rules, the verification routines and the feedback.

Guiding rules

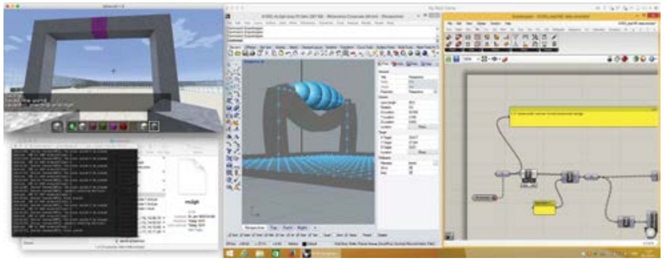
It is of interest to us how we can steer the resulting typology while keeping the player focused only on the game mission and not on the artefact they are creating. The concept of vocabulary was introduced early on in the process and has evolved over several implementations and test iterations (Figure 5).

An example of a goal given to the players is to build up to a certain height, let’s say 30 blocks. At the same time, their initial resources allow them to build only 12 blocks high. Each player can create a shape out of the vocabulary for which they get rewarded with additional 12 blocks of material. However, the shape is designed to consume 10 of their blocks yet is just one block high. That means that the maximum height the player can reach now is 15 blocks. Players can combine the shapes from the vocabulary to form the best strategy to reach the height of 30 blocks and win. They can also interact with the other players by building together or stealing each other’s achievements. Another type of goal, for example, could be to build a certain number of a given vocabulary shape with the constraint that, again, the material needed for them is initially insufficient. Completing a mission takes approximately 20 minutes, after which the result is saved and the game world is reset.

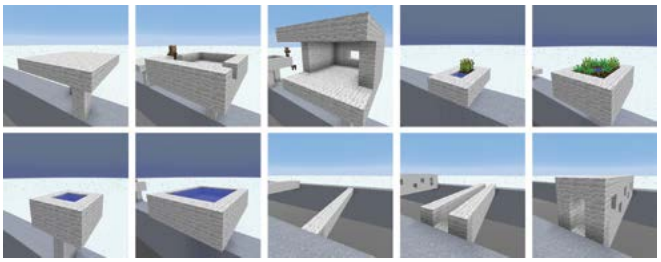
Verification routines

At the heart of the project lies a real-time link between the online Minecraft server and analysis routines in parametric design tool Grasshopper (www.grasshopper3d.com) (Figure 8). We use the open-source Python library ***pymclevel*** by David Vierra to get geometry from Minecraft to Rhino/Grasshopper, which is then reconstructed in a format suitable for structural performance evaluation via the add-on ***Millipede*** (Vierra 2014, Sawako and Panagiotis 2014). The results from this analysis are fed back to the Minecraft server, either by creating new blocks or changing the material/color of existing ones. For this we developed our own Grasshopper components that can send commands to the server over SSH. Along with the limited vocabulary, this information is used to help to guide the players’ actions in the game world so that they create ‘viable’ structures. In section ***Fast Feedback,*** we discuss how this is done.

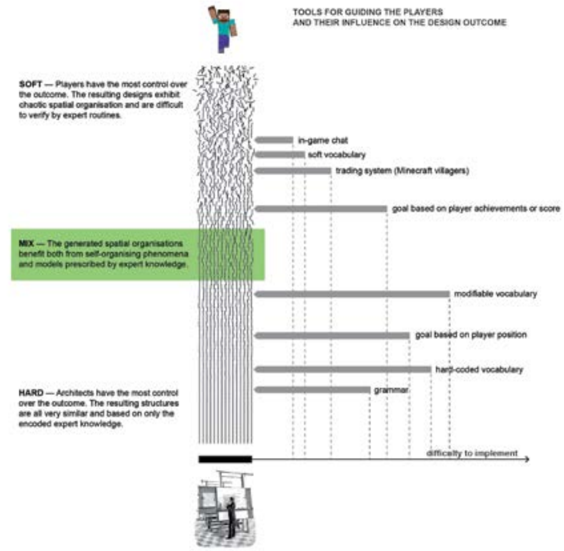
We use a robot fabrication process to verify the stability and constructability of a game result. We feed the point information for each Minecraft block to our robot arm, Ginger, who builds



9



10



11

the model in a scale of 1:100. It grabs wooden cubes with a vacuum gripper, pushes them into the glue and positions them on the correct spot according to the digital model. If a cube is cantilevering, Ginger places free-standing cubes underneath it as a support. These are removed after the model is finished. This additive manufacturing process is similar to selective fusing of materials in a granular bed such as ***binder jetting***, with the difference that the granules are much larger (Figure 6).

The second purpose of the models is to visually and materially represent the player-created designs as a form of feedback for both Players and Experts.

Fast feedback

We provide on-the-fly feedback to the players in several ways: posting messages to the in-game chat, showing particle effects and changing the material of blocks in the game world.

In the chat console, we display messages to inform players what to do when they join the server. While playing, if a player successfully builds a new element out of the vocabulary, we inform everyone. Goal achievements are also broadcast in this fashion.

Particle effects are used to visually attract players to important positions in the game world. The most important of these is used to inform a player that the system is verifying a newly built vocabulary element.

At the game world level, a structurally weak spot computed in Millipede would be marked in Minecraft and turned into an optional, secondary mission for the players. This would award them extra points for resolving the structural problem (Figure 9).

RESULTS AND DISCUSSION

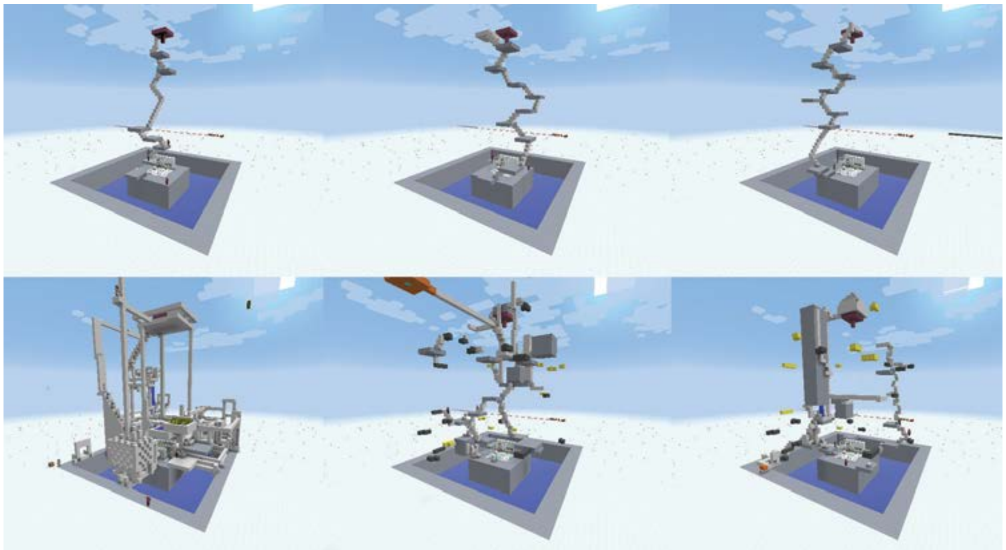
We aimed to learn how to calibrate design sessions of the project ***20,000 Blocks*** so that while playing, groups of non-experts could generate architectural designs with prescribed features. One important design decision which needs to be made is the degree of freedom given to players. In other words, who should have the control over the design: the Expert or the Player.

As Brabham states, for a well-functioning crowdsourcing model it is important that the locus of control regarding the creative production of goods, exists between the organization and the crowd. If the locus of control is closer or larger in the community, such as in the case of open-source software or Wikipedia, or if the control is largely in the organization's hands, such as when a company wants the community to simply vote for the color of a product, we are not seeing a true crowdsourcing model (Brabham 2013).

Therefore, we took into account in our analysis of the results the balance of control that players and experts had. We imagined it on a gradient scale of soft to hard, where the soft end of the spectrum gives more control to ***Players*** and the hard end of the spectrum, more control to ***Experts*** (Figure 11).

Soft vocabulary

We conducted the first test with soft vocabulary. The Players were presented with a catalogue of architectural elements such as rooms, terraces, etc. (Figure 10). A player would build any one of those, anywhere within the confined building site in Minecraft. They would then come to the Organizers, who would visually verify if the built element matches the catalogue and award the player the corresponding points and resources. This happened



12

9 Structural verification in real-time. The block marked in magenta is a new secondary mission for the players.

10 The initial soft vocabulary is a visual catalogue, including three variations of platforms, three bridges, two gardens and two pools.

11 Gradient from soft to hard. Each tool empowers players or transfers an expert's will differently. It also comes with a certain cost in effort to implement.

12 Top: Three game results from a hard vocabulary and one main goal. Bottom left: A game played with soft vocabulary. Bottom middle, right: Two screenshots from gameplay with hard vocabulary and secondary goals marked in gold.

verbally in the game chat. The goal was also soft — get the highest number of points.

The outcomes were rather fuzzy designs as all the aspects of the system — the guiding rules, the verification process and the feedback — were negotiable between the player and the organizers. Another hindrance, we noticed, was that the elements in the vocabulary were too numerous and too complex to be remembered by players easily. This required the players to refer to the vocabulary too often, hence breaking their flow of play. Furthermore, it was also slow — a game was marked as completed when the confined building site was filled up, which took around two hours (Figure 12 bottom left).

Therefore, we looked for ways to make the rules stricter by automating them.

Hard vocabulary

The guiding rules of the hard vocabulary method we tested were based on three automated routines:

1) ***Detection routine***: The elements that players build to gain rewards are recognized automatically by the game engine. We described one element — a platform of 5x5 blocks — in code using Minecraft's programming language and command blocks (Figure 14). The player places a trigger block (blue diamond) and positions their character on top of it to activate the detection routine (Figure 17). Only exact copies of the catalogue structures will be recognized and rewarded, thus reducing the fuzziness of the design solutions.
2) ***Trading***: The resources for building need to be purchased from Non-Player Characters (NPCs) — a Minecraft villager — in exchange for emeralds, gained when successfully building an element from the vocabulary. Villagers can be summoned only at the ground level. This ensures that the structure being created can be walked up and down (Figure 7).
3) ***Goal detection***: We defined a main goal, achievable within 15–20 minutes that gives a clear end to each session (Figure 7). This limits the time for generating a design and allows for many solutions to be created under the same conditions. Reaching the goal automatically triggers a save of the built structure, and the game is reset. When we noticed that game plays resulted in self-similar linear structures (Figure 12 top row) we introduced a set of secondary goals as incentives to break the linearity of the gameplay and create spatially richer designs (Figure 12 bottom middle and right).

At this stage of the project, many structures built by the players had cantilevers which would be not be able to sustain the force of gravity. To unlock the true architectural potential of mass-participation, we need a set of evaluations that relate, and possibly rate, a shape on its performance as an architectural structure in the real world. Verification was done post-factum for most of the game outcomes using Grasshopper/Millipede. The results prompted us to change the elements in the vocabulary until the play results had less problematic overhangs (Figure 15).

The feedback from the game actions was printed automatically to the game chat (Figure 13) and kept the players informed of what to do in the game, who scored a new point by building a 5x5 platform and when the game was over, being saved and reset. This proved very useful and successful in keeping the players aware.

13

14

Modifiable vocabulary

The latest iteration of the guiding gameplay mechanics implements a vocabulary of predefined design elements that follows combinatorial rules (Figure 16). This is somewhat similar to an L-System (Dapper 2003).

It has all the three main principles of the ***Hard Vocabulary*** variation with the difference being that detectable structures are not described in code but are built on dedicated slots next to the building platform as a visual catalogue. This allows us to modify and prototype the vocabulary much faster. It furthermore allowed us to separate the roles of Organizer and Expert without needing to teach Minecraft scripting.

The fact that the vocabulary consisted of more than one element proved to soften the play outcomes. Therefore, we tried a system where the trigger blocks were placed automatically with the vocabulary element and not by the players. We called this new notion Grammar because it meant Experts could define which of the elements could be built upon each other thus opening or limiting choices for the players.

We didn't use the game chat as extensively as in the ***hard-vocabulary*** approach and relied on players orienting themselves in the game world. This proved confusing for most people.

Findings

The project has been in continuous development and testing for one year at the time of this writing. Each of the three vocabulary types (soft, hard and modifiable) was put through online playtesting. The hard vocabulary was additionally tested at events at the ***Digital Design Unit (DDU)*** at TU Darmstadt with university students, as well as at ***Invent the World (ITW)*** with 7–10-year-old kids (Figure 18). The modifiable hard vocabulary was tested at the ***SmartGeometry 2016 Conference*** (Figure 19). At sg2016 we also tested how other architects, acting as Experts, could use the modifiable vocabulary to define their own purpose and use-case for the designs (Figure 1 and 2).

With the current iteration, we as Organizers find that the systems in place allow a great range of creativity from both Experts and Players but feel that the results are still too hard in terms of spatial organization.

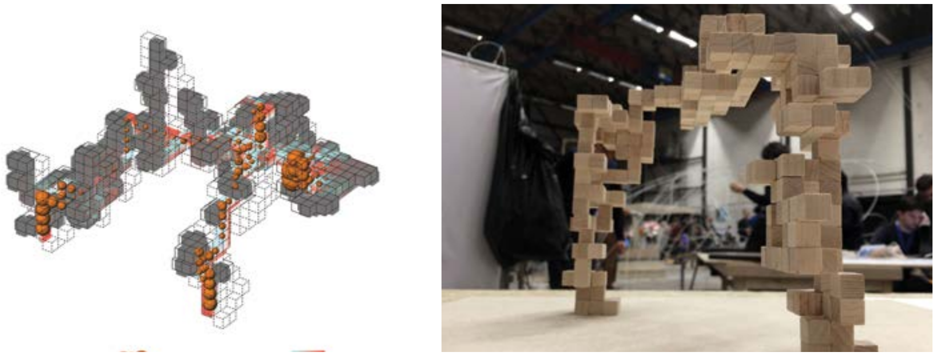
So far, 8 people, split in three teams, have participated in the project as Experts. With the game levels contributed by the Experts and ourselves, we tested a total of 7 game maps, resulting in 57 player-created structures. The Experts were very happy with their experience and were able to both generate unique use cases, as well as the needed gameplay to create suitable results.

Around 70 players have taken part in the various on-site and online play sessions. It takes 5-10 minutes to explain to a new player the basic rules, and then they enter the tutorial part of the game map which we have designed to introduce them to the principle of vocabulary element detection. Currently, the Players are still too overwhelmed when they begin participation. The project requires more guidance and feedback for players, and the skill barrier to entry must be lowered. Much too much time was spent dealing with simple issues such as the players accidentally wasting their rewards or quitting in the middle of a game.

From all 57 game results, we selected, post-processed and robotically manufactured five models. The robotic process is currently too slow to keep up with the digital iterations. The speed is 200 blocks per hour, and an average design requires 1200 to 2000 placed blocks to build, i.e. 6–10 hours per model. This includes both the model blocks as well as the supporting blocks. As the focus of the research is on game design as a guiding instrument, the analysis of the models built is secondary and will be subject to further development and publications.

Key findings:

1) Harder rules are better than soft ones in delivering a feasible architectural design.
2) If the rules become too hard, the players no longer feel part of the design and as such are unmotivated to play.



15

13 The game chat is used to feed current information to players.

14 The interface for coding in Minecraft using command blocks.

15 Millipede screenshot (left) and wooden robotically fabricated model (right) based on a vocabulary designed by Max Rudoph and Marios Messios.

3) If the rules are too soft, the resulting structures become too chaotic. This makes them difficult to verify and construct. In addition, players find those designs confusing to navigate and play through.
4) Robotic fabrication is possible but currently too slow to provide meaningful feedback.
5) An easily modifiable vocabulary allows other architects to participate as Experts and that opens up our method to more possible applications.
6) Clear, non-architectural, time-bound goals make participation easier and more entertaining than tasks where players need to understand the spatial and architectural qualities of the game elements they are building.

The work thus far reveals that the calibration process — the positioning between hard and soft — is an ongoing challenge and main focus of our research. Finding the right balance between hard and soft requires constant testing. With every project iteration and every game played, we are able to calibrate the balance better.

CONCLUSION

The paper explained how the use of Minecraft, in connection with Grasshopper, can allow for non-expert players to generate and evaluate architectural design concepts. We presented a methodology that utilizes a mixture of expert and non-expert participation, crowdsourcing for design creation with real-time structural analysis feedback on design decisions, and robotic fabrication.

The two main hurdles to overcome in our research in the future development of ***20,000 Blocks Above the Ground*** are:

1) ***Fool proofing*** the game world so that players are able to play unsupervised at any time. We can achieve this by making game rounds shorter (10–15 minutes) and with an automated start and end. If a player joins while a game is in progress, they are put in observer mode and added to the queue for the next round.
2) ***Diverse and continuous feedback*** using the game chat console for detailed progress reports as well a continuous structural performance evaluation. We are implementing a new message system able to display full screen large text messages to the players. We also intend to introduce a third type of feedback as an online catalogue of player-created designs. A web-gallery is in the making to display screenshots of all played games immediately after they have been completed.

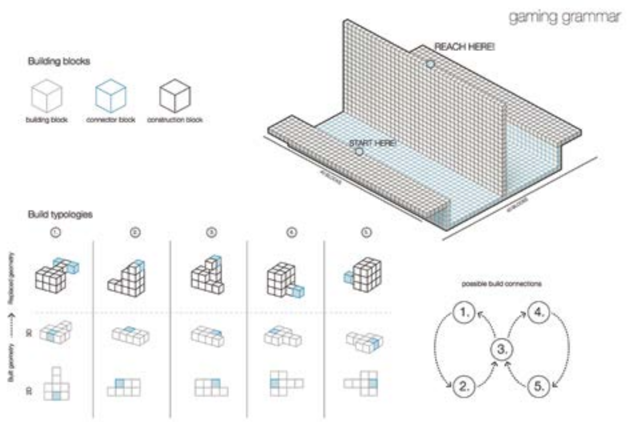
Possible applications we see for our method of game-to-CAD-to-Robot transfer of geometry are:

- In the research field, new forms of architecture could be explored that transgress established typologies. This is helped by engaging the unbiased minds of the non-experts.
- In the practice, a specific design task could be crowdsourced, tapping into the decision-making power of inhabitants, neighbors and investors by defining the corresponding in-game rules and providing the suitable background evaluation algorithms.

We consider the ideal scenario to test the method in reality is the massing out of an architectural concept, such as defining the rough placement and orientation of rooms in a building or determining the location and infrastructure of buildings in an urban design scheme (Figure 20). The ability to involve a team of experts in the making and testing of the architectural vocabulary holds the potential for feasible and well-performing solutions. To make this viable, a main hurdle of the architectural field to overcome is developing post-processing routines that can quickly turn player-generated designs into CAD models for further specification.

ACKNOWLEDGEMENTS

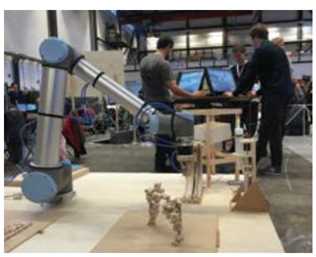
The authors would like to thank: all the Players of 20,000 Blocks; Jörg Hartmann and Sebastian Kotterer for assistance with the robotic fabrication process; the participants at our workshop at SmartGeometry 2016 who entered into the role of Experts — Andrea Quartara, Marc Emmanuelli, Ulrik Montnemery, Marios



16

17

18



19

16 The vocabulary shown here, designed by Alexander Gösta, Samuel Eliasson and Ashris Choudhury results in bridge-like building designs.

17 Screenshot from the game play at Invent the World. The trigger blocks are in blue.

18 Play session at Invent the World, Australia.

19 Our setup with 4-way game terminal and a robot at SmartGeometry 2016.

Messios, Max Rudolph, Alexander Gösta, Samuel Eliasson, Ashris Choudhury; the organizers of SmartGeometry 2016; Oskar Gerspach for shooting and cutting the project's video; Christian Schwamborn from TU Darmstadt's IT department for setting up our server; and our colleagues at the Digital Design Unit (DDU) and Invent the World (ITW).

REFERENCES

BBC. 2012. "Minecraft to Aid UN Regeneration Projects." BBC News, November 26. Accessed June 22, 2016. http://www.bbc.com/news/technology-20492908.

Brabham, Daren C. 2013. Crowdsourcing. Cambridge, MA: MIT Press.

Cooper, Seth, Adrien Treuille, Janos Barbero, Andrew Leaver-Fay, Kathleen Tuite, Firas Khatib, Alex Cho Snyder, Michael Beenen, David Salesin, David Baker, Zoran Popovic, and >57000 Foldit players. 2010. "The Challenge of Designing Scientific Discovery Games." In Proceedings of the Fifth International Conference on the Foundations of Digital Games. Monterey, CA: FDG. 40–47.

Dapper, Timm. 2003. "Practical Procedural Modeling of Plants." Accessed March 25, 2016. http://www.td-grafik.de/artic/talk20030122/overview.html.

Friedman, Yona. 1975. Toward a Scientific Architecture. Translated by Cynthia Lang. Cambridge MA: MIT Press.

Fröst, Peter, and Peter Warren. 2000. "Virtual Reality Used in a Collaborative Architectural Design Process." In Proceedings of the IEEE International Conference on Information Visualization. London: IV. 568–573.

Fröst, Peter. 2003. "A Real Time 3D Environment for Collaborative Design." In Proceedings of the 10th International Conference on Computer Aided Architectural Design Futures, edited by Mao-Lin Chiu, Jin-Yeu Tsou, Thomas Kvan, Mitsuo Morozumi, and Tay-Sheng Jeng. Tainan, Taiwan: CAAD. 203–212.

Otten, Claudia W. 2014. "Everyone is an Architect." In ACADIA 2014: Design Agency—Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture, edited by David Gerber, Alvin Huang, and Jose Sanchez. Los Angeles: ACADIA. 81–90.

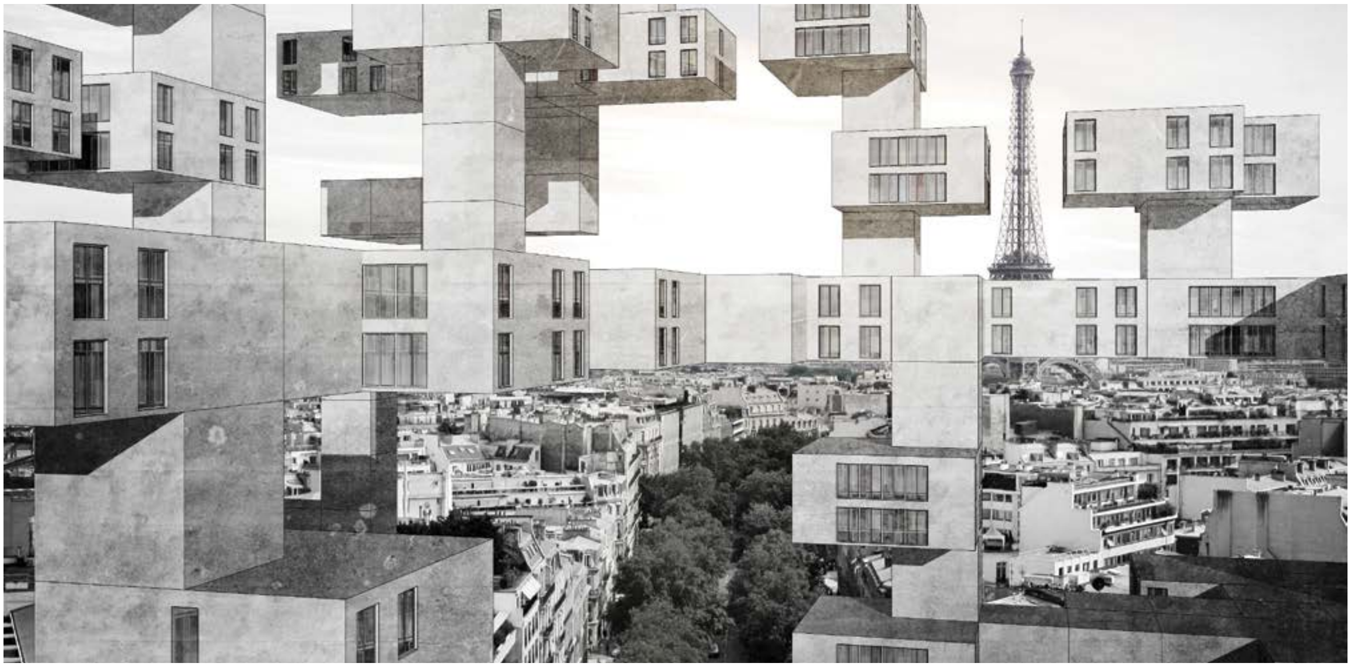
Parker, Laura. 2014. "Not Just Playing Around Anymore: Games for Change Uses Video Games for Social Projects." New York Times, April 21, 2014. Accessed May 10, 2016. http://www.nytimes.com/2014/04/22/arts/video-games/games-for-change-uses-video-games-for-social-projects.html?_r=2.

Sanchez, Jose. 2015. "Block'hood, Developing an Architectural Simulation Video Game." In Real Time: Proceedings of the 33rd eCAADe Conference, vol. 1, edited by B. Martens, G. Wurzer, T. Grasl, W. E. Lorenz, and R. Scharanek. Vienna, Austria: eCAADe. 89–97.

Savov, Anton, Olivier Tessmann, and Stig A. Nielsen. 2016. "Sensitive Assembly: Gamifying the Design and Assembly of Façade Wall Prototypes." International Journal of Architectural Computing 14 (1): 30–48.

Savov, Anton. 2012. "Project Avocado." Accessed March 3, 2016. http://www.aware-studio.com/progeny/project-avocado/.

———. 2013. "Box in a Cloud." Accessed March 3, 2016. http://www.aware-studio.com/progeny/box-in-a-cloud/.



20 A rendering showing an architect's interpretation of a schematic design created by players in the 20,000 Blocks' game map WOULD YOU MIND YONA FRIEDMAN? by Marios Messios and Max Rudolph (rendering Max Rudolph).

Sawako, Kaijima, and Michalatos Panagiotis. 2014. "Millipede." Accessed September 14, 2015. http://sawapan.eu/.

Szantner, Attila. 2016. "What Did the Easter Bunny Bring Us." MMOS, March 28. Accessed April 15, 2016. http://mmos.ch/news/2016/03/28/what-did-the-easter-bunny-bring-us.html.

Segard, Achille, Jules Moloney, and Tane Moleta. 2013. "Open Communitition: Competitive Design in a Collaborative Virtual Environment." In Open Systems: Proceedings of the 18th International Conference on Computer-Aided Architectural Design Research in Asia, edited by R. Stouffs, P. Janssen, S. Roudavski, and B. Tunçer. Singapore: CADRIA. 231–240.

Vierra, D. 2014. "pymclevel — Minecraft Levels for Python." Accessed November 10, 2015. https://github.com/mcedit/pymclevel.

Warren, Tom. 2016. "Minecraft Sales Top 100 Million." The Verge, June 2, 2016. Accessed July 2, 2016. http://www.theverge.com/2016/6/2/11838036/minecraft-sales-100-million.

IMAGE CREDITS

Figures 1, 5: Quartara, 2016, ©

Figure 2: Messios, 2016, ©

Figures 3, 8, 9, 10, 11, 12, 13, 14, 17, 19: Savov, 2016, ©

Figure 4: Savov, 2013, ©

Figure 6: Hartmann, 2016, ©

Figure 7: Savov/Messios, 2016, ©

Figure 15: Rudolph/Messios, 2016, ©

Figure 16: Eliasson, 2016, ©

Figure 18: Buckton, 2016, ©

Figure 20: Rudoplh, 2016, ©

Anton Savov works as a research associate at the Digital Design Unit (DDU), Faculty of Architecture at the Technical University Darmstadt and is the founder of AWARE — an architectural design studio using crowdsourcing and game design to engage the non-experts in the creative process. Previously, Anton has worked at Bollinger+Grohmann Engineers and taught at the Städelschule Architecture Class, Frankfurt. In 2012 Anton received the Artists and Architects in Residence Program Scholarship of the MAK Center in Los Angeles.

Ben Buckton is a Freelance Game designer, consultant and educator. As a completely self-taught developer he has a somewhat unorthodox way of approaching problems which has gotten him involved in many interesting projects. As a partner at Invent the World, Ben designs and runs Holiday programs for 'nerdy' children in Australia. This sees him traveling back and forth across the globe with great regularity. Ben is also the event organizer for the Frankfurt Independent Developer scene, helping to bring together unique and talented people.

Oliver Tessmann is an architect and Professor for Digital Design at TU Darmstadt. His research and teaching is located in the field of computational design and digital fabrication. After receiving his doctorate from the University of Kassel in 2008, he served as the director of the performativeBuildingGroup at Bollinger+Grohmann. From 2010 to 2012 he was Visiting Professor at the Städelschule Architecture Class in Frankfurt and, from 2012 to 2015, held the position of Assistant Professor at KTH Stockholm.