From Wooden Blocks and Lego to Minecraft: Designing and Playing with Blocks to Learn in a 3D Virtual World

Anders I. Mørch, Ingvill Thomassen

Department of Education, University of Oslo, Norway
anders.morch@iped.uio.no, ingvt@student.uv.uio.no

Abstract: Minecraft, a virtual 3D world game to fight nighttime monsters and a design environment for building shelters, strikes a balance between "want to" and "have to" participate, that is, students "want to" use it (self-directed learning) and educators can impose constraints by knowledge (curriculum-driven learning). These two goals often seen as contradictory are reconciled by taking advantage of the unlimited number of opportunities block building and role-playing provide, and treating learning as a constrained-based design process. We are in the midst of organizing a pilot study of using Minecraft in a university course in education. We are interested in understanding Minecraft's capacity to engage students in learning and we conjecture there is a connection between building and playing in Minecraft and developing knowledge and skills. We survey related work and use a theoretical framework to guide our efforts. Our findings are preliminary with open issues for further research.

Keywords: 3D virtual world, block-building, empirical research, end-user development, end-user tailoring, learning, Minecraft, modding.

1 Introduction and Motivation

Minecraft is popular video or sandbox game with children and young students because they can create amazing buildings and structures, combat and shield nighttime monsters, and collaborate with peers in one of three different playing modes (creative, survival, hardcore). The users interact in Minecraft by placing and breaking three-dimensional (3D) building blocks (modeled after 1 m^3 physical construction blocks) and they can communicate with each other by chat. The blocks, tools and other items are stored in the inventory. Actions in Minecraft (building and destroying) have a persistent effect, keeping areas in the state that the user leaves them in. The notion "sandbox game" is in analogy to a kid playing in a sandbox; the sand has no instructions or objectives in the outset; it can be formed in an infinite number of ways, enabled by available tools and limited only by imagination. Constraints can be imposed in order to narrow the choices. For example by modeling a specific domain, imposing rules of various kinds (interaction rules, design games, roleplaying), and instructional scaffolding.

We made the following preliminary observation. Designing in the game is tightly coupled with use (playing). It requires coordinating physical action (placing and

Proc. of Fourth International Workshop on Cultures of Participation in the Digital Age - CoPDA 2016
Gothenburg (Sweden), October 23, 2016 (published at http://ceur-ws.org).

destroying blocks) and visual imagery (hand gestures; spatial visualization), whereas playing requires toggling between communication, discussing alternatives, and making decisions. We want to explore the interdependencies of these actions and interactions from a social constructivist and sociocultural perspective (learning as tool mediated, self-directed, and collaborative activity). Therefore, we distinguish an "outer world" of building and playing and an "inner world" of internalization and making sense of external events, which are interdependent (e.g. learning a new concept or an idea and how to use it in interaction with others).

Scholars in educational psychology have addressed these and related issues over many years. For example Piaget (1972) said that knowledge is constructed and that the main function of cognition is adaptive in order to organize and make sense of the experiential world (Piaget, 1972). Vygotsky (1978) suggested internalization and externalization as two complementary processes for concept development in children through interaction with adults (Vygotsky, 1978). By taking block placing as the quintessential design act, can we say that computer-based block building consists of two complementary modes, constructive and argumentative design, as suggested by Fischer and colleagues (1989), or is it better characterized by a co-evolution of internal and external knowledge development (Cress, 2013). Andersen & Mørch (2016) use the term mutual development in a related manner, characterizing the interdependency of two types of software development, general (software house) development and specific (end-user) development.

On that basis we ask the following research questions: 1) What characterizes users' design activity in Minecraft, and 2) how can teachers organize learning as a constraint-based design process?

At this stage the questions are dealt with as part of the analysis of the preliminary findings obtained by observation of Minecraft Realm activity in a pilot study and for presentation at a workshop to stimulate discussions to give us feedback and ideas for further research.

2 Related Work

Previous studies have shown that using different forms of building blocks in constructive play, such as wooden blocks, jigsaw puzzles and Lego bricks, can improve different types of abilities in children, one of which is spatial abilities, such as spatial visualization, solving mazes and reading maps (Caldera et al., 1999). The development of spatial abilities can also lead to an improvement in mathematical problem solving performance in children (Oostermeijer, Boonen & Jolles, 2014).

Minecraft shares some similarities with traditional block building, in that it has a Lego-like feel and can be used in co-construction tasks with peers (Canossa, Martinez & Togelius, 2013). Minecraft is already being used as an educational tool in a number of different topics such as language and literacy, teaching about sustainable planning, digital storytelling and social skills. The environment can also be adapted to teach about scientific topics such as mathematics and chemistry (Nebel, Sascha & Günter, 2016; Bos et al. 2014). Similar to building with blocks, the open-ended quality of a

Proc. of Fourth International Workshop on Cultures of Participation in the Digital Age - CoPDA 2016
Gothenburg (Sweden), October 23, 2016 (published at http://ceur-ws.org).

sandbox environment such as Minecraft allows the students great freedom while exploring the game, which makes it suited for scenario-based learning and role-playing activities. Bos and colleagues (2014) used Minecraft to teach third-grade students about mathematics. The students were given a scenario in which they were instructed to build a copy of a coastal town where the visual imagery of the connected cubes was used to teach the students the difference between the concepts perimeter and area. The students had a variety of possible dimensions they could choose from to shape the buildings, which prompted discussions on how to solve the task, and helped them to differentiate between the concepts (Bos et al., 2014).

Modding or modifying a game means to alter it to make it behave differently from the original version, and this might be initiated by identifying an opportunity for improving the game experience (Christiansen, 2014). Adding a new building block to the Minecraft inventory to render a wall using a specific pattern is one example, but there are many others as well. El-Nasr and Smith (2006) discussed modding as a pedagogical activity. They found that modding could engage students in learning computer science concepts related to game development, but modding has the potential to engage the learner in other topics as well such as 3D geometry and vector calculus (El-Nasr & Smith, 2006).

3 Evolving Artefacts Framework

Fischer (1998) proposed a model of software development called Seeding-Evolutionary-growth-Reseeding (SER). Seeding is the stage when a new system is ready to be used (initialized). Evolutionary growth is when end-user developers incrementally add new functionality to the system to adapt it to unanticipated needs, whereas reseeding occurs when developers integrate new and existing functionality to create a new version of a system. The stages are interdependent and interact (Fischer, 1998). The interaction of evolutionary growth and reseeding was the object of two case studies conducted by Andersen & Mørch (2009; 2016) and referred to as mutual development.

Mørch (1997) suggested tools and techniques for evolutionary growth by three levels of tailoring based on a review of tailoring functionality found in ordinary applications and modification techniques made possible by object-oriented programming, graphical user interfaces and direct invocation of tailoring functionality. The three levels are referred to as customization, integration and extension (Mørch, 1997), and provide gradual access to application functionality. *Customization* means to modify the user interface and to choose among predefined configuration options. *Integration* means to add new functionality to the application, which could be functionality created by someone else (e.g. a mod in a video game), and *extension* means to add new functionality by creating a program that extends reusable (inherited) functionality.

More recently g*eneralization* has been suggested as a developer activity that follows extension and evolutionary growth (Mørch, 2011). Generalization can be described in two different ways: 1) reorganize application functionality to more easily

Proc. of Fourth International Workshop on Cultures of Participation in the Digital Age - CoPDA 2016
Gothenburg (Sweden), October 23, 2016 (published at http://ceur-ws.org).

integrate with new functionality (Fischer, 1998) and to 2) spread an end-user developed solution to a large community of users with or without involvement of professional developers (Mørch, Nygård & Ludvigsen, 2009; von Hippel, 2001).

4 Evolutionary Growth in Minecraft: A Pilot Study

Our main goal is to understand the relationship of end-user tailoring (including modding) and learning, e.g., what type of learning can be achieved or associated with modding in Minecraft. However, modding turned out to be too advanced use of the game for our non-technically oriented users, and stepping-stones toward modification were needed. We have organized and studied a course in technology-enhanced learning in general education at our University where students created a learning environment in Minecraft intended for use in skills practice in primary and secondary schools in Norway. The assignment required the use of Minecraft's creative mode (without monsters) and imposed only two constraints: 1) connecting the learning environment with a subject specific learning goal, and 2) creating three use cases in of increased complexity and named as follows: 1) building, 2) collaborating, and 3) tailoring, respectively. The use cases should have the following connections to each other: 1) the usefulness of building should be tested by roleplay (e.g. instructions for building by remote collaboration, carry out a domain specific roleplay), and 2) improving collaboration by end-user tailoring (e.g. adding new items, blocks, and scenes to the game in order to more authentically model a subject domain). Figure 1 shows some screen images from Minecraft.

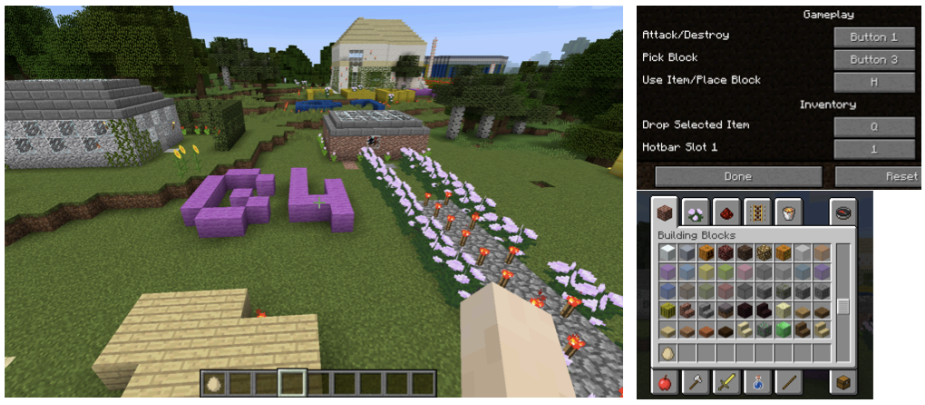

Fig. 1. Leftmost picture shows one of the group's constructions; top-right picture shows part of the controls editor (for customizing button actions); bottom-right: building block palette

The students were divided into groups of 4 and they could pick a subject domain of interest, which were English as a foreign language; religious buildings; tool use in the Stone Age; 21st century skills.

All the students were able to successfully accomplish the first two parts (building and collaborating). The building activity was demonstrated by screen images of steps

Proc. of Fourth International Workshop on Cultures of Participation in the Digital Age - CoPDA 2016
Gothenburg (Sweden), October 23, 2016 (published at http://ceur-ws.org).

in the process (e.g. Figure 1, leftmost image) and chat and audio recording captured students' collaboration. All groups struggled with understanding the concept of modding, in particular the distinction between making modifications in the shared environment (Realm) and one's installed copy of Minecraft, as mods apply directly to the latter and indirectly in the former. One group was able to articulate the need for a mod, and another group was able to search and find a relevant mod (TerraFirmaCraft), but none of the groups were able to install a mod in their learning environment (a current limitation of Minecraft Realms is that it does not support modding). None attempted to create a mod from scratch (requires Java programming).

Using the terminology of the evolving artefacts framework we can say that customizing Minecraft means to configure the environment without adding new functionality (choosing a mode; customizing your avatar; setting hotkeys). The students customized their environment by setting hot keys and modifying their avatar. Integration means to add new functionality to the environment (installing a mod), created by other Minecraft end user developers and downloaded as a .jar (Java Archive) file, but not supported by our version of Minecraft Realms. Extension means to add new functionality by writing code in Java, reusing an existing Java class and overriding its class methods by object-oriented programming. Finally, generalization can be accomplished in one of two ways: 1) share a new mod by uploading it to a mod repository, and 2) course organizers reorganizing the student space for next year's course (or move the location to a new set of x-z coordinates). Generalization by sharing (spreading an innovation) is one of the strong features of Minecraft and one that separates it from predecessor block building games. For example, spreading an innovation in the Lego community means to suggest to the (Lego) company that it should offer a new set of building blocks or a new game, which requires signatures of at least 10.000 users. Generalization in Minecraft is organized according to the user-manufacturing model (self-organization), which bypasses the scrutiny and quality assurance of professional developers (von Hippel, 2001). The risk associated with user manufacturing is that you might create or install a mod that makes your Minecraft unusable, which in worse case would require installing a new version of Minecraft and/or purchasing a new Mojang account.

5 Discussion and Open Issues

Minecraft provides a range of online activities, some open ended and some fixed (e.g. crafting tools), and we have explored three open-ended activities: building, collaborating (roleplay) and tailoring (customizing and modding). We have studied a group of graduate students in education doing this as part of a course project. Informed by the evolving artefacts framework, our long-term goal is to understand the relationship of end-user tailoring (such as modding) and learning. Towards that end we asked the following research questions of the pilot study: 1) What characterizes users' design activity in Minecraft, and 2) how can teachers organize learning as a constraint-based design process?

Proc. of Fourth International Workshop on Cultures of Participation in the Digital Age - CoPDA 2016
Gothenburg (Sweden), October 23, 2016 (published at http://ceur-ws.org).

5.1 What Characterizes Users' Design Activity in Minecraft

The students we observed were deeply engaged in the tasks of the assignment. However, the assignment required discussions up front for proper understanding, e.g. that the three scenarios or use cases were meant to model increasingly complex design activity, moving from individual design (building; customizing) to collaboration, first collaboration in small groups (building and giving instructions, roleplay within a building, etc.), and then mass collaboration (reusing and sharing user-developed mods on the internet). Furthermore, it was a threshold for students to understand the difference between the effects of modification on an individual level (e.g. a new item, building block or tool in the Inventory) and the indirect effect or appearance of new materials in the buildings created in the shared environment as a result of using new building blocks installed as mods by individual players. This was explained at a theoretical level.

The successful participation in the three design activities varied accordingly. All students were able to start building structures and customize their personal environment within a few hours; and creating and recording collaboration or roleplay scenarios took a couple of weeks. All students were able to understand the idea of modding (at the theoretical level), but no group was able to accomplish it technically by reusing and/or creating mods. The students themselves were able to identify the sources of the problem. Further work is necessary for the course organizers to make sure it is technically possible to install mods in Minecraft Realms or to use another shared Minecraft environment (e.g. Education Edition).

5.2 How can Teachers Organize Learning as a Constraint-Based Design Process

Teachers organize the learning activities by creating an assignment for the students, which defines the constraints for the game. Within these constraints the students have the freedom to explore the environment with their own creative ideas. To introduce students to modding, intermediate stages might be necessary, and we suggested three stages, each providing progressively more complexity, starting with ordinary (individual) use and ending with modifying the game. The stages allow learners to engage in and access both subject specific and generic skills. Further work ought to develop these ideas into more detail, including defining the role of the teacher as a designer of the learning environment (Mørch, et al. 2015), what skills are more appropriate to teach in this environment (domain specific or domain general), and what are the appropriate methods for data collection and analysis in research design.

References

1. Andersen, R., Mørch, A. I.: Mutual Development: A Case Study in Customer-Initiated Software Product Development. In: Proceedings 2nd Int'l Symposium on End User Development, pp. 31-49. Springer, Berlin Heidelberg (2009)

Proc. of Fourth International Workshop on Cultures of Participation in the Digital Age - CoPDA 2016
Gothenburg (Sweden), October 23, 2016 (published at http://ceur-ws.org).

2. Andersen, R., Mørch, A.I.: Mutual Development in Mass Collaboration: Identifying Interaction Patterns in Customer-Initiated Software Product Development. Computers in Human Behavior 65, 77–91 (2016)
3. Bos, B., Wilder, L., Cook, M., O'Donnell, R.: Learning Mathematics through Minecraft. Teaching Children Mathematics 21(1), 56-59 (2014)
4. Caldera, Y.M-, Culp, A.M., O'Brien, M, Truglio, R.T., Alvarez, M, Huston, A.C.: Children's Play Preferences, Construction Play with Blocks, and Visual-spatial Skills: Are they Related? International Journal of Behavioral Development 23(4), 855-872 (1999)
5. Canossa, A., Martinez, J., Togelius, J.: Give Me a Reason to Dig: Minecraft and Psychology of Motivation. In: IEEE Conference on Computational Intelligence in Games (2013) DOI: 10.1109/CIG.2013.6633612
6. Christiansen, P.: Players, Modders, and Hackers. In: Understanding Minecraft: Essays on Play, Community and Possibilities, pp. 23-29. McFarland, Jefferson NC (2014)
7. Cress, U.: Mass Collaboration and Learning. In: Handbook on Design in Educational Technology, pp. 416–425. Taylor and Francis, London (2013)
8. El-Nasr, M.S., Smith, B.K.: Learning Through Game Modding. Computers in Entertainment 4(1) Article 7 (2006) DOI=http://dx.doi.org/10.1145/1111293.1111301
9. Fischer, G.: Seeding, Evolutionary Growth and Reseeding: Constructing, Capturing and Evolving Knowledge in Domain-Oriented Design Environments. Automated Software Engineering 5(4), 447-464 (1998)
10. Fischer, G., McCall, R. and Morch, A.: Design Environments for Constructive and Argumentative Design. In: Proceedings CHI'89, pp. 269-275. ACM Press, New York (1989)
11. Mørch, A.: Three Levels of End-User Tailoring: Customization, Integration, and Extension. In: Computers and Design in Context, pp. 51-76. The MIT Press, Cambridge MA, (1997)
12. Mørch A. I.: Evolutionary Application Development: Tools to Make Tools and Boundary Crossing. In: Reframing Humans in Information Systems Development, pp. 151-171. Springer, London (2011)
13. Mørch, A.I., Nygård, K.A., Ludvigsen, S.R.: Adaptation and Generalisation in Software Product Development. In: Activity Theory in Practice: Promoting Learning Across Boundaries, pp. 184-205. Taylor & Francis Books, London UK (2009)
14. Mørch A.I., Hartley M.D., Ludlow B.L., Caruso V., Thomassen I.: The Teacher as Designer: Preparations for Teaching in a Second Life Distance Education Course. In: Proceedings ICALT-2014, pp. 691-693. IEEE Computer Society, Washington DC (2014)
15. Oostermeijer, M., Boonen J.H., Jolles J.: The Relation Between Children's Constructive Play Activities, Spatial Ability, and Mathematical Word Problem-Solving Performance: A Mediation Analysis in Sixth-Grade Students. Frontiers in Psychology 5(782), (2014) DOI= http://dx.doi.org/10.3389/fpsyg.2014.00782
16. Nebel, S., Schneider, S., Rey, G.D.: Mining Learning and Crafting Scientific Experiments: A Literature Review on the use of Minecraft in Education and Research. Educational Technology & Society 19(2), 355-366 (2016)
17. Piaget, J.: The Origins of Intelligence in Children. International University Press, New York (1952).
18. Von Hippel, E.: Innovation by User Communities: Learning from Open-Source Software, MIT Sloan Management Review 42(4), 82-86 (2001)
19. Vygotsky, L.: Mind in Society: The Development of Higher Psychological Processes. Harvard University Press, Cambridge MA (1978/1934).

Proc. of Fourth International Workshop on Cultures of Participation in the Digital Age - CoPDA 2016
Gothenburg (Sweden), October 23, 2016 (published at http://ceur-ws.org).