Multicraft: A multimodal interface for supporting and studying learning in Minecraft

Marcelo Worsley[0000-0002-2982-0040] , Kevin Mendoza Tudares, Timothy Mwiti, Mitchell Zhen, Marc Jiang

[1] Northwestern University, Evanston, IL, USA
[2] University of California, Berkeley, Berkeley, CA, USA
marcelo.worsley@northwestern.edu,
KevinMendozaTudares2022@u.northwestern.edu,
mwititimothyk@gmail.com, mitchellzhen@berkeley.edu
MarcJiang2021@u.northwestern.edu

Abstract. In this paper, we present work on bringing multimodal interaction to Minecraft. The platform, Multicraft, incorporates speech-based input, eye tracking, and natural language understanding to facilitate more equitable gameplay in Minecraft. We tested the platform with elementary, middle school students and college students through a collection of studies. Students found each of the provided modalities to be a compelling way to play Minecraft. Additionally, we discuss the ways that these different types of multimodal data can be used to identify the meaningful spatial reasoning practices that students demonstrate while playing Minecraft. Collectively, this paper emphasizes the opportunity to bridge a multimodal interface with a means for collecting rich data that can better support diverse learners in non-traditional learning environments.

Keywords: Games, Constructionism, Spatial Reasoning, Data Mining

1 Introduction

Interest and participation in video games continues to grow. Recent reports note that three out of four Americans play video games and an estimated 2.7 billion gamers around the globe [1]. While part of this growth in video games is fueled by the COVID-19 pandemic [2], [3], researchers have long discussed the important role that games can play for learning and socialization [4]–[7]. This opportunity for learning and socialization can have particular positive benefits for students who are disabled by inaccessible, physically collocated, game-based learning experiences. However, to be effective, virtual gaming environments must also be intentional about considering questions of accessibility. Technological developments like the Xbox Adaptive Controller provide an important step towards accessible gaming experiences. Nonetheless, the goals of accessible gaming experiences should also consider equitable play and identify ways that students' game-based practices demonstrate student knowledge development and expertise. Regarding equitable play, it is not sufficient to simply replace the input modality. Additional steps should be taken to develop comparable gaming experiences for all

participants. Furthermore, beyond including novel interfaces for supporting participation, there is an important opportunity to utilize different modalities to chronicle student learning.

In this paper we describe our efforts to combine these ideas in a platform called Multicraft. Multicraft is a collection of multimodal interfaces that allow students to use speech, gaze, text, or any combination of these modalities to play Minecraft. The platform also includes built-in features that can accelerate game play and a method for storing multimodal data that researchers can use to study student in-game computational thinking and spatial reasoning practices.

The next section highlights prior research that informs our work and situates Multicraft relative to this prior work. We then present a quick summary of the design principles and technical architecture for Multicraft. We also highlight some of the core features of the platform. This is followed by a short presentation of user feedback on different elements of the platform. After describing the platform and user feedback, we transition into a high-level presentation of some of the research that we have conducted using multimodal data. We particularly focus on ways that eye-tracking and video data have allowed us to study various complex spatial reasoning practices that students exhibit while playing Minecraft. We conclude with a discussion of future work and suggestions for overarching objectives for this type of work.

2 Prior Literature

2.1 Autcraft

Autcraft is a user community and user-generated modification of Minecraft that was specifically developed for learners with Autism and their families [6], [8], [9]. Across this work Ringland emphasizes how a Minecraft community, when appropriately designed, can be an important space for autistic youth and their families. Ringland[10] specifically describes how families configure and navigate the physical, liminal, and virtual spaces needed to successfully participate in Minecraft. Many of the core features of Autcraft are achieved through the rich community of people, and the custom Minecraft mods that govern how students are permitted to interact within the game. The design of Multicraft takes a similar approach of configuring an open-source server that users can customize and deploy as needed. Additionally, the inclusion of multiple possible input modalities speaks to a recognition of the varying liminal spaces that families configure. Moreover, Multicraft also includes features that try to adapt to the user, as opposed to requiring the user to conform to standardized methods of input. Our adoption of this strategy is an attempt to utilize ability-based design (ABD)[11], which we describe in the following section.

2.2 Ability Based Design

ABD is a set of tenets for guiding computer scientist as they create accessible interfaces. A central tenet of ABD is to embed adaptation into the design of the interface, as

opposed to requiring the user to carry the burden of using their own adaptive technologies and tools. Moreover, interfaces should be designed to be utilized with a variety of input modalities. While Multicraft still has several limitations in terms of the abilities that are supported, our goal is to integrate features that reflect the diverse set of abilities that human possess.

2.3 Multimodal Learning Analytics

The use of multimodal data also provides a means to leverage techniques from Multimodal Learning Analytics (MMLA)[12]. MMLA is a collection of strategies that can support real-time and post-hoc analysis of learners in non-traditional learning environments. Historically MMLA has involved a broad set of modalities that frequently include video, audio, gesture tracking, eye tracking, affect detection, and electro-dermal activation [13], [14]. Multicraft utilizes multimodal fusion of text, speech, and gaze data to provide an accurate and naturalistic input modality. Beyond that, however, the multimodal data provides an opportunity to carefully chronicle student learning and knowledge development within the Minecraft game. In looking at student game play using multimodal data, we will mostly explore work on student spatial reasoning skills, which we quickly summarize in the next section.

2.4 Spatial Reasoning Skills

Spatial reasoning refers to a variety of skills that generally pertain to one's ability to perceive, utilize, and store different types of spatial information [15]. This might include the ability to perform navigation tasks using a map, mentally folding a piece of paper, or rotate an object in one's mind. A variety of spatial reasoning tests have been developed to measure these skills in laboratory contexts, but a growing body of research advocates for researchers to examine spatial reasoning in less restricted contexts[16], [17]. Video games have also been a context where researchers have studied spatial reasoning skills [18]–[20]. Hence, one of the contributions that we explore alongside the development of a multimodal interface is the opportunity to analyze student data, particularly eye-tracking and video data, to better understand and acknowledge the ways that students practice spatial reasoning in Minecraft. This approach follows in a traditional of psychological research that studies mental rotation using eye tracking data during standardized spatial reasoning tests [21]–[23].

2.5 Summary

There is a broad collection of prior and relevant work that relates to this project. An important distinction that we emphasize with Multicraft is the goal of supporting equitable play and using multimodal data to discern student learning practices. Bringing together these different ideas is novel relative to the prior work in these domains.

3 Multicraft

Multicraft is a platform designed to support multimodal interaction in Minecraft. The current platform integrates speech-based input, eye tracking and natural language understanding and reflects a set of design principles that is informed by hundreds of hours observing elementary and middle school students play Minecraft. These observations include working with students with a variety of physical, visual, and neurological impairments. For example, our team has watched students with cerebral palsy effectively disconnect from the Minecraft experience because of an inaccessible interface. We have seen students on the autism spectrum experience significant anxiety and frustration when they incorrectly execute a command and are unable to easily undo that action. And we have generally seen how novice Minecraft players have struggled to be accepted into a classroom Minecraft community because they have not yet learned the syntax of the platform. These types of observations and others contribute to the design principles that we have incorporated into the Multicraft platform.

3.1 Design Principles

Our design principles center on equitable play, taking a pluralistic approach, allowing for natural language input, facilitating collaboration, and easy version control.

Equitable play is a goal that we believe is sorely missed within prior work. Many of the existing accessible interface look to simply replace the keyboard and mouse with other input modalities but do nothing to ensure that the overall experience is equitable. We enact this principle by seamlessly embedding some computer programming into the platform. For example, students can request to build a house with certain dimensions, instead of having to manually place every block for said house. Students can also easily clone existing objects, create large ravines, and quickly create entire cities.

Pluralistic approach refers to allowing users to complete the same action using a variety of modalities and commands [24], [25]. Given the Constructionist orientation of Minecraft it seems appropriate to also ensure that our platform supports multiple forms of engagement and execution. Concretely, we achieve this by permitting users to complete the same action using a variety of modalities. This goal is also an acknowledgement of the diversity and intersectionality present within disability communities. Users bring many different abilities and preferences, hence, Multicraft aims to support as many of those abilities and preferences as possible.

Allowing natural language input speaks to our desire for Multicraft to adapt to the language and syntax of our users, as opposed to requiring users to learn a specific syntax. The inclusion of a Natural Language Understanding (NLU) Engine advances this principle.

Facilitating collaboration is a central component of the Minecraft platform in that players can collaboratively build, mine, and battle within shared virtual worlds. Players can also share materials with one another. However, Minecraft does not allow players to easily share entire built structures with one another. Multicraft adds this functionality by allowing players to name their built structures which subsequently lets other players to easily re-use them.

Version control refers to the player's ability to easily undo and redo executed commands. Minecraft natively allows students to add and destroy blocks, however, many students have trouble undoing previous actions (e.g., using a command to create a 10 x 10 x 15 house). Not being able to undo or redo command-based build actions inadvertently discourages students from using different commands.

3.2 System Technical Architecture

In the current version, users may speak simple commands to the game, such as "build a ten by twenty-two wall of stone" or "move forward fifteen blocks." The user can also use eye tracking commands like "track my eyes and build a ten by ten by ten building of quartz". This command will start the eye tracker and build the desired structure where the user looks. These commands are executed instantaneously, speeding up the process of building compared to placing blocks individually or using the more complicated built-in Minecraft commands. We achieve these types of interactions by integrating several technical components (**Fig. 1**). The overall technical architecture can be split into three layers: User Devices, the Multicraft Client and the Multicraft Server

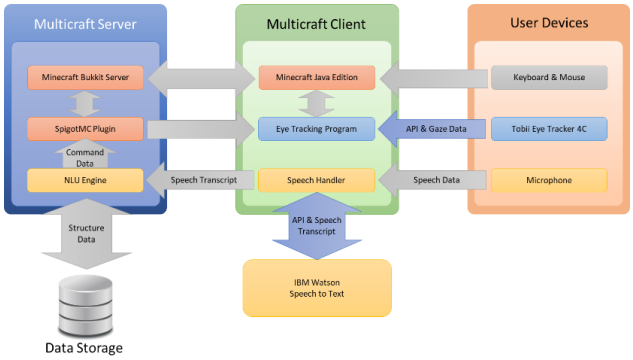


Fig. 1. Technical architecture of Multicraft

User Devices. Users have the option of using any number of input modalities. We emphasize an eye tracker and microphone, because these are the two multimodal inputs that we have explicitly integrated and tested with participants. However, many of the capabilities also work for participants who may be playing Minecraft with a keyboard, mouse, or touch screen.

Multicraft Client. The Multicraft Client interface includes capabilities for communicating between the User Devices and the Multicraft Server. This includes handling text, speech, and eye tracking data, and sending instructions to the Multicraft server.

Audio Processing. For users that elect to use speech-based input, we have included Speech-to-Text capabilities for converting the audio information into transcriptions. This component consists of a Python program that uses web sockets to communicate

with a speech recognition engine. While this piece is customizable, our current implementation uses the IBM Watson Speech-to-Text service. Because of the atypical vocabulary used within Minecraft, and because we wanted to optionally include "Multicraft" as a custom trigger word, we utilize a custom language model and augment the standard dictionary with Minecraft specific words like "redstone" and "ender pearl."

Eye Tracking Module. Gaze interaction is implemented through a C# program that utilizes a Tobii Eye Tracker 4C and the Tobii Interaction Library Beta API. This program collects 2D gaze point data in real time and subsequently uses that data to adjust the player's in-game camera. This eye tracker-based movement results in the current fixation location moving to the center of the computer screen. This capability can be optionally toggled on or off through a voice command (e.g., "start tracking my eyes") or using text. There are three ways of interacting with the eye tracker. The first is for building and allows the user to determine where to build a structure that they request via voice command. The second is for navigation and allows the user to focus and move their character forward. By dwelling at the center of the screen (+/-50 pixels) for three seconds, their avatar begins to walk forward, and continues to walk until they change their gaze. The third is for moving the camera with the user's eye movement. As the player looks to the right or to the left, the screen moves with them, and re-centers on their current gaze location. Voice or text commands can be used to toggle with eye gaze mode the user wishes to utilize.

Text Input. The system also supports typed natural language commands, such as "build a ten by ten by ten gold structure". The commands are sent to the Natural Language Understanding (NLU) Engine and processed accordingly. More rigidly structured commands can also be entered through the in-game chat feature. An example of this would be "/mmbuild 10 10 10 41", where "/mmbuild" corresponds to the custom build command, "10 10 10" to the dimensions of the structure (x, y, and z), and "41" to the Minecraft material ID - in this case 41 indicates a gold block. Versions of the platform have also supported shorthand text-based input. For example, the instruction "build a 10 by 10 by 10 gold structure," could be written in shorthand as "b 10 10 10 41." One reason for including a shorthand notation is because entering "hacks" is already part of the Minecraft culture. Additionally, it eliminates the need for the additional hardware or client-side software used with speech recognition.

Multicraft Server. The main server side component is a Minecraft Bukkit server with a SpigotMC plugin. Bukkit is a free, open-source software for running and extending Minecraft servers. SpigotMC is a high-performance Minecraft server API. The SpigotMC plugin handles executing Multicraft-specific commands and passing those instructions to Minecraft. Multicraft depends on a NLU Engine that we briefly described in the next section.

NLU Engine. The understanding engine produces a semantic representation of the audio transcripts (or text-based commands) using SpaCy [26]. It subsequently operationalizes

those requests into actions within the Minecraft game. Beyond this, the NLU engine incorporates some unique features:

Synonym Detection and Keyword Extraction: We use WordNet [27] to ensure that synonyms are mapped to supported game-based actions. This allows us to support a larger set of commands than simply "build", "move" and "track," for example. Currently, the system can execute commands for building structures, placing blocks, and moving the player along cardinal directions. It also supports activating eye tracking to be used for building or navigating, adjusting the camera, adding items to the user's inventory, and saving, naming, and cloning previously built structures. Finally, as we describe in more detail later, it also supports undo and redo capabilities. In addition to identifying synonyms for actions, the platform can also identify synonyms for cardinal directions and different in-game materials. After identifying the appropriate synonyms, the engine performs keyword extraction. It identifies numbers, material types, cardinal directions, and certain parts of speech. Together, synonym and keyword detection allow players more freedom in how they issue commands.

Error detection and user feedback: Error detection and feedback provide a player with information on why an error has occurred if a command is not executed. For example, if a build command is issued without dimensions, a message is displayed on the screen that informs the player that the command requires dimensions.

Saving and Naming of structures: Building in Minecraft can involve a lot of repetition. Multicraft provides a simple way of accomplishing repeated building by allowing players to name their structures after they have built them. Once a player builds a structure, they can name it (e.g., "home"), move to a new position and issue the command, "/mclone home" and a replica will be built. The system also supports sharing objects with others. The structures that the user names are available for all other users on that Minecraft server to utilize.

Undoing and Redoing: The server implementation also includes the ability to undo and redo items within a user's build history. For example, if a student issues a command to "build a fifteen by eight by nine brick house" and subsequently decides that they no longer want that structure, they can simply say "undo" and the structure will be removed. Similarly, they can say "redo" or type "/mredo" to recreate the structure.

Simplification of existing Minecraft commands: The server-side implementation also includes simplifications of built-in Minecraft commands. For example, Minecraft has existing commands for filling a space with blocks or cloning an existing structure. To use these capabilities, students need to remember the Cartesian coordinates for the bottom front, and upper rear portions of the space to be filled or cloned. Multicraft includes text or speech commands that can be used to store these values for the user and subsequently allow them to issue a command to fill or clone the space.

4 Part 1: User experiences with Multicraft

Throughout the platform development process, we have conducted user studies with different groups of participants. Within this section, we will discuss three groups of

user studies. The first were middle school students participating in one-week long summer Minecraft camps. The second group of participants were middle school student participating through Minecraft clubs. The third group were K-2nd graders.

4.1 Overview

Each group of participants experienced a slightly modified set of tasks. These differences were due to our ongoing development of the Multicraft platform. They were also influenced by the COVID-19 pandemic, which interrupted significant portions of our data collection. The summer camp students tested Multicraft's capabilities, but without eye tracking. Instead, we focused on examining how students would make use of the speech-based natural language input. The middle school Minecraft club participants tested the combined speech and eye-tracking interface. The elementary school students primarily tested the text-based input interface, with a select few also testing speech-based capabilities in Minecraft.

4.2 Participants

The summer camp participants included ten students that identified as boys, and 1 that identified as a girl. None of them identified as frequent Minecraft users, and all were between 12 and 14 years old. The middle school Minecraft club participants included three students that identified as girls and 7 students that identify as boys. All students ranged in age from 12-14 years old. All the students identified as having prior experience with Minecraft. The elementary school participants included four students that identify as girls and five that identify as boys. All students were between kindergarten and 2nd grade. Only two of the students had prior experience with Minecraft. One student in each of the programs was on the autism spectrum, however, our observations will not be based solely on those students.

4.3 User Testing Tasks

As previously noted, each group completed a different set of tasks. However, consistent across each group was a researcher-led demonstration of the basic capabilities of the platform. The summer camp participants were asked to use the platform while engaging in free play and also given pictures of buildings to recreate. The Minecraft club participants completed three specific tasks. First, they were asked to use the eye tracker to move around the world. This involved adjusting their gaze to the desired location and dwelling in the middle of the screen to move forward. Next, they tested text input commands. Here they could follow the example to construct a building of one-hundred cubic blocks or create something of their own choosing. They then tried a slightly modified way to build the same structure. Finally, they tested out building with the eye tracker activated.

Elementary school students were given two tasks to complete. The first task involved trying two different approaches for building a 100 cubic feet structure in Minecraft. The

second tasks asked them to build that same structure but also create copies of that structure. They completed these two tasks using native Minecraft features and Multicraft features. We also had some of the students test our speech-based input alongside using text-based input.

4.4 Data Collection

Our data collection with the summer camp participants was the most extensive. During the program, the research team collected audio and video data of each student using Open Broadcaster Software (OBS) on the respective participant laptops. OBS also enabled us to capture a video of the screen. In addition to the individual videos, we also collected whole room video, and conducted some informal interviews and surveys with the students about Multicraft. Informal interviews asked students their views on the utility of the platform. We did not explicitly ask them if they did or did not like the platform because the authors had previously interacted with some of the students, and we thought their stated perceptions might be biased. Hence, a large portion of the analysis is based on what we observed students do, and less about what students explicitly said.

For the middle school Minecraft club participants, our data collection included observations and informal interviews. Individual students tested the platform, one-at-a-time. After they finished their tasks, a research team member asked a few questions about the accessibility of the areas of improvement for future iterations, overall enjoyment with Multicraft, and ease of use.

Finally, for the elementary school students, we conducted informal interviews and observed as students used the platform. Specifically, we asked students (and their parents) about the relative ease of use between Minecraft commands, and our custom Multicraft commands.

4.5 Data Analysis

Members of the research team, namely the authors, individually and collectively watched videos of student game play. The research team took note of different observations and discussed these notes with other team members to get their assessment and interpretation of the different episodes. We also consulted our field notes and debrief notes from the different sessions. The surveys from students were also looked at qualitatively to get a general picture of student perceptions. The pieces of data that we selected for this analysis serve as exemplars for some of the design elements that we wish to highlight with this platform. In this way, the analysis is not intended to suggest absolute causality, or universality. Instead, they are indications of potential interpretations of student behaviors and utterances.

4.6 Observations and Findings

Through the observations, surveys, and interviews we were able to identify some important information about Multicraft and its potential to enhance the Minecraft gameplay experience. We also uncovered some challenges with the platform and potential future developments. We will organize the results based on modalities. Specifically, we will begin with observations and student comments associated with text-based input, then speech-based input, and, finally, gaze-based input. We then touch on some key ideas from student interviews.

Text-Based Input. Text-based input proved to be quite difficult for several of the students that we tested with. This was true across the novice middle school students and elementary school students. None of the students knew how to type, with most students trying to type with two or three fingers. This meant that their attempts to enter different commands were hampered by being slow and inaccurate. This resulted in them having to re-enter the same commands a second, and, at times, a third, or fourth time. The shorthand text input, on the other hand, provided a seamless interaction for both middle school and elementary school students. One of the current realities is that many children who are growing up in the age of touch screen and audio assistants, do not have much experience typing. However, only having to type a single letter and a handful of numbers seem to be appropriate for the different users that we observed. The biggest challenge in the case of shorthand text input was the need for students to know the numeric code for the block types that they wanted to use. The numeric codes offered both advantages and disadvantages. Using a number was easier than knowing how to spell words like "acacia," but it also meant that students needed a way to look up the correct material codes. To address this, students found a webpage that includes a full list of block types and their codes, and kept that webpage open so that they could toggle to it as needed. In later versions of this platform, we introduced the ability to use the numbers and material names interchangeably. Apart from this, students seemed to find that the shorthand text input approach worked as expected. Moreover, the young students were particularly appreciative of the ability to name structures and quickly clone them. They also expressed a preference for Multicraft's commands over the Minecraft text commands because the Multicraft required fewer words.

For the middle school students with prior Minecraft experience, using text-based input appeared to be the most natural form of interaction. They already had experience using different Minecraft commands through coding activities in their Minecraft clubs. The students reported that the Multicraft commands made constructing large structures much easier, and many users stated they preferred using the commands over placing the individual blocks. The one deviation from this was that students do not always know exactly what they want to build. In those instances, they did not find much utility in the Multicraft text-based input modality which requires users to state the dimensions of their desired structure.

Speech-Based Input. For many of the students, the prospect of using speech-based input offered a welcome alternative to having to type. They also liked the idea of natural language input because remembering a specific syntax for the various functions that Minecraft makes available was challenging for many of the students. However, in practice, some students found the speech recognition accuracy to be unreliable. Because of this, some students had to repeat their requests several times before getting it to successfully build. We believe that this is due to poor quality acoustic models for adolescents and because many of our participants spoke other languages at home. Once the requests were properly processed, students expressed amazement that the structure was created so quickly. Their amazement and excitement generated social interactions among their peers, as they eagerly shared their creations with other students. Students also appeared to be comfortable talking to the computer and did so using their normal speech cadence and tone. Another challenge was that students expressed uncertainty about what kinds of instructions they could issue, a common challenge within multimodal interfaces [28]. We have since developed a cheat sheet that users can utilize as they interface using Multicraft. We have also improved the speech recognition by executing some additional code-based customizations.

Despite these challenges, the elementary school students that we observed preferred speech-based input to typing. In fact, some of the students even tried to use speech-based input within the Minecraft inventory because they were not sure how to spell the name for several of the materials. At times they would start typing a word incorrectly and subsequently become unable to find the block-type of interest. When facilitators were present, they would ask for spelling assistance, but in the absence of adult involvement, it is unclear how they would be able to build their structures as envisioned. This challenge, on the part of students, exemplifies a primary challenge we want to overcome. Students may have complex and intricate ideas in their minds, but lack the computer knowledge, or Minecraft experience to enact that idea.

Eye Tracking Input. Of the three modalities, eye tracking was the one that students found to be most intriguing. Many students had interfaced with systems that used speech recognition, or seen people use them on smartphones. Eye trackers, on the other hand, were a novelty. For the experienced Minecraft users that tested the eye tracking system, there was a noticeable learning curve to navigating the game with their eyes. It generally took students between five and ten minutes to get to the point where they were comfortable using their eyes to navigate and build in Minecraft. Additionally, one of the most frequent comments about the eye tracking feature was that moving the camera with eye movement at first felt unnatural. We also observed that it took time for them to learn the mapping of gaze position to camera movement speed. However, with some practice, students became quite proficient integrating this additional modality into their game play.

Interviews. In addition to observations, we also conducted informal interviews with students about the Multicraft interface. Some of the questions raised were based on

observations that we made. Other questions were focused around how socially acceptable it would be to use this platform when playing with friends, and what additional functionality the students would like to see added to the platform. Here, we focus on questions of social acceptability because that is of primary consideration for our goal of equitable play.

The question around student perceptions of using Multicraft during general multiplayer games was met with mixed reviews. Many students saw no problem with using Multicraft in creative mode (the game mode where students freely build and create). To them, Multicraft fit into their existing schema of Minecraft hacks. In fact, many students came to refer to the shorthand text-based input "Tim [Mwiti]'s Hack" because he had introduced it to them and showed them how it worked. However, students were less keen about using Multicraft in survival mode (a mode where players have limited resources and need to mine and hunt to survive), as they likened it to cheating. When further probed about why it was cheating, a student described that the current implementation does not use any of your inventory items, meaning that when using Multicraft you have an unlimited supply of resources. This would be unfair in survival mode. Other students agreed with this assessment and suggested that while in survival mode, the user should only be able to use items in their inventory. Students were also concerned with the social stigma associated with using Multicraft in multiplayer survival. Because some of the text commands are entered through the chat terminal, some players' Multicraft actions may be visible to others. In our ongoing development, we are working to address these concerns with social stigma and acceptability.

5 Part 2: Multimodal Analyses of Minecraft gameplay

Alongside the utility that Multicraft provides for users, we are also interested in ways that the multimodal data used in the Multicraft platform can be help us better understand and chronicle students' growing competencies. In this section we present data from a laboratory-based study and two Minecraft summer camp-based studies.

5.1 Overview

The summer camp-based studies included students completing build challenges as well as open-builds. Within these camps we were interested in the spatial reasoning practices that students exhibit across different building contexts. The summer camps lasted for approximately 15 hours.

The laboratory-based study included undergraduate and graduate students who completed a mental rotation test and three specific build challenges. This study was undertaken to look at the ways that mental rotation practices that students employ on standardized spatial reasoning tasks might mirror onto the practices students use in Minecraft. It was also an opportunity to explore development of automated techniques for studying spatial reasoning in Minecraft. As students completed the one-hour long build challenges, eye tracking and screen recordings were captured.

5.2 Participants

The summer camp-based studies include the same one that was discussed in section 4. In addition to those students, the data in this section also includes an additional summer camp with 12 middle school students. This group included four females, and eight males. All but one of the students had prior experience with Minecraft.

As previously noted, the laboratory-based study included 19 undergraduate and graduate students. Fifteen students identified as males, while the remaining four identified females. Sixteen of the students were undergraduates, while the remaining three were pursuing graduate degrees. One of the students is on the autism spectrum.

5.3 User Testing Tasks

Within the summer camp studies, students completed various build challenges. One of the structures that students created can be seen in Figure 2.

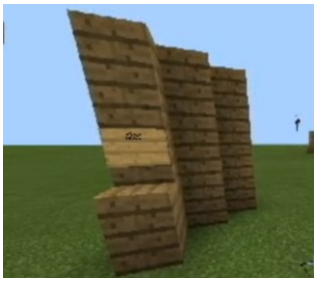


Fig. 2. Sample structure that students recreated in Minecraft

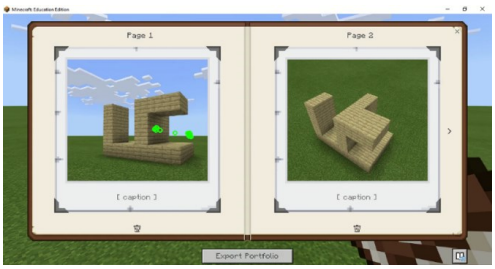


Fig. 3. Picture of in-game reference images with eye tracking data overlay (green dots).

Within the laboratory-based study, students were asked to complete a mental rotation test [29] and then to use in-game reference images (e.g., Figure 3) to recreate the three structures pictured in Figure 4-6.

Fig. 4. Structure A

Fig. 5. Structure B

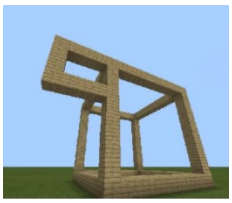


Fig. 6. Structure C

5.4 Data Collection

Across the different studies, we collected eye-tracking and screen recordings of student gameplay. The eye-gaze data was captured using the Tobii 4C eye tracker, which was collecting data at 90Hz. The screen recordings were created using the Social Signal

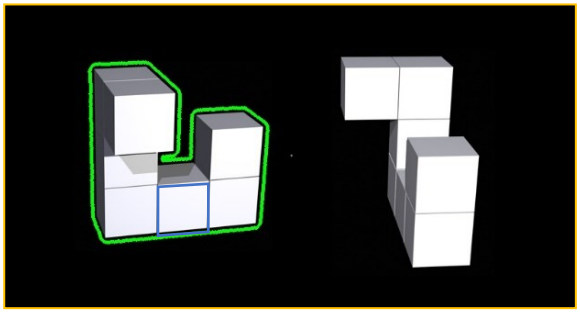
Interpretation (SSI) framework [30] or OBS. We also collected whole room audio, individual audio, and server logs of student game-based actions using the LogBlock plugin. The eye-tracking data and screen recordings are the focal portion of these analyses, though the game-based log data did inform portions of the analyses.

5.5 Data Analysis

The summer camp-based analysis heavily relied on human annotation of the video data. Because the total collection of videos included more than 100 hours of data, we elected to use some simple data mining to help with the video selection process. Computational analysis was used on the log data to look for sessions that showed noticeable differences in the number of blocks that students placed and based on differential performance on the mental rotation test. Based on this information, we were able to select a small collection of videos to human code. The research team collectively watched and coded the videos for different spatial reasoning practices. A subset of these observations is presented in this paper.

The data analysis process for the laboratory-based study used computer vision- based contour detection and synchronous eye tracking data to identify salient features and gaze patterns on the different mental rotation test questions. Contour detection is an approach that allows a computer program to label contiguous shapes within a given image. The contours can be hierarchical, such that a given contour can contain several other contours. Figure 7 contains three contours from a mental rotation test question.

Fig. 7. Picture of hierarchical contours outlined in yellow, green, and blue.



For the eye gaze data, we computed fixations and saccades following research conventions. A fixation was recorded when any set of successive data points was no more than 25 pixels apart from one another, and when the collection of gaze points represented at least 50 milliseconds. The resultant features were used for human observation of common gaze patterns and are also supplied to different machine learning algorithms to highlight correlations between different features and student performance on the mental rotation test.

5.6 Observations and Findings

Summer Minecraft Clubs: The summer camp-based analysis helped surface several ways that student exhibit spatial reasoning practices while playing Minecraft. Broadly speaking, several of the ideas connect to work on using visual anchors to help students make sense of a given design [22], [31]. Some of these practices include choosing a starting point, frequently a corner of a structure, or the middle, and subsequently counting along a single dimension. In some videos students can be heard verbally counting, or moving their mouths as they pass over the different blocks. Viewers can also see the eye gaze trace jump from block to block within the screen recordings.

Another common approach was students looking at a structure from a specific perspective. This perspective was often chosen to match the angle of the reference picture and simplifies their ability to draw a correspondence between the reference image and the structure that they are building.

One specific instance of perspective taking is taking a bird's-eye view of a structure. Frequently, students would fly above their current build so they could see the entire structure. When looking from above, students would scan over the relative dimensions and look for symmetries or other obvious discontinuities.

Perhaps the most intriguing use of the bird's-eye view was in conjunction to students creating their own attentional anchors. We see in example of them when students try to recreate a mushroom tower (Figure 8). While they are working on the bottom part of the mushroom top, they need to create an oval that will go around the center column that they have created. When the students take a bird's-eye view, they see that the surrounding oval is not quite right (Figure 9). To fix this, they voluntarily create a rectangular scaffold (Figure 10), which is used to more easily construct the oval. This represents a fairly complex spatial practice that the student spontaneously uses to fix this build.

Figure 8. Mushroom tower image

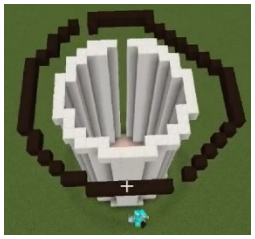


Figure 9. Failed design for mushroom rim (the dark portion)

Figure 10. Scaffold created by students to anchor mushroom rim

Collectively, we see students using several different strategies to spatially reason about structures in Minecraft. The summer camp-based studies helped us surface some of these practices as inferred from computer-informed video selection, and subsequent human analysis. The laboratory study that we conducted was an attempt to explore some of these patterns using more automated techniques and regarding a validated mental rotation test [29].

Laboratory Study: The laboratory-based analysis is still a work in progress. Thus far we have been able to successfully combine computer vision derived features, automated detection of fixation points, and machine learning to discern differential gaze patterns among students that exhibit different mental rotation ability. As a sample output from the mental rotation test portion of the video, see Figure 11.

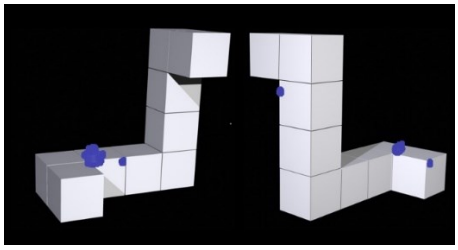


Fig. 11. Aggregated frequent fixation points for laboratory-based study

In Figure 11 we can see the most common gaze points across all participants, as overlaid on a static image of the mental rotation test image (note: detecting this static image in the different frames of the video required using ORB feature detection as students could scroll up and down on the screen, a process that we do not describe here in the interest of brevity). When we aggregate across the fixation points within the reference images that students looked at to inform their builds, we can see what students are and are not paying attention to. We think that this type of analysis can translate to the Minecraft gameplay data by specifically examining which parts of a reference Minecraft building students are and are not paying attention to when trying to replicate a structure.

Our analysis of correlations between different features and student performance on mental rotation tasks has also demonstrated some promise [32]. For example, previous analysis found that a decision tree trained on a broad selection of contour-based, fixation, and saccade features can be used to accurately model student mental rotation performance. Our interest in doing this type of analysis was to explore a good research methodology for studying student spatial reasoning using computer vision and eye tracking data. We see this computational approach being something that we can translate into our analyses of Minecraft video game play and presumably detect some of the complex spatial reasoning practices that we observed in the summer camp-based Minecraft analysis. We might also use these features as a way to look at how individual students' spatial reasoning practices change over time relative to themselves.

6 Discussion

The overarching objective of the featured studies and analyses was to describe our current efforts to couple a multimodal interface that promotes equitable play with opportunities to use MMLA to delve into the complex spatial reasoning practices that students demonstrate while playing Minecraft. The user feedback from elementary and

middle school students suggests that there are several aspects of the Multicraft platform that they find to be compelling and useful for different groups of users. Several students found the speech and gaze-based input modalities to be a welcomed change from the standard approach to building in Minecraft. These different modalities were adopted based on our observations from working with hundreds of elementary and middle school students play Minecraft, and particularly informed by observed experiences of students with disabilities. The results that we have gathered so far suggest that many of the objectives around using multimodal input were achieved. However, as we noted earlier in this paper, simply providing alternate modalities for input is insufficient. Instead, we want to afford a more equitable gaming experience where students feel equipped to participate alongside their peers regardless of abilities. In one respect, the student feedback that using Multicraft during multiplayer survival seemed unfair is an acknowledgement that Multicraft can offer noticeable benefits in executing different commands and actions faster. At the same time, however, this feedback also points to potentially larger challenges about the social stigma of accessible interfaces that aim for equitable experiences. The students' primary concerns were about players having access to resources outside of their inventory, which is something that we can easily correct in future iterations of the platform, but some social stigma may persist.

Our ongoing analyses using multimodal data also hint at some promising opportunities to chronicle the knowledge and reasoning strategies that students evidence while playing Minecraft. Our human analysis of video data keyed in on several common spatial reasoning practices, while also noting the ways that students might be intentionally creating visual anchors to help them better recognize symmetries and other visual elements. Additionally, the computational analysis that we briefly described that combines computer vision, fixation detection, and machine learning during mental rotation tests is a first step in automatically mining student game play data for different spatial reasoning-relevant practices. We suggest, however, that this is just one example of what we can accomplish using techniques from MMLA in game-based learning contexts. Though we did not describe it in detail, our work also involves looking at student computational thinking in Minecraft using student game-play videos and eye tracking data. To date, we have successfully used computer vision to detect how much time students spend using the coding interface in Minecraft Education Edition. Detecting these video clips has helped us focus our human video analysis process by automatically selecting clips where students are actively programming. It also can help elucidate the design prompts and activity structures that successfully lead to students doing more programming in Minecraft. Moving forward, we intend to build out more of the techniques from laboratory studies to utilize on data derived from more ecological settings.

7 Limitations

A major limitation of this work is that it was conducted with small groups of students who self-selected into the programs. Additionally, these studies were completed with multiple groups of participants who had interacted with one of the authors on previous occasions. This may have made the students less likely to share their true opinions.

Another potential limitation is that we tested different elements of the platform with different populations. While we had planned for a more systematic study during the first quarter of 2020, the COVID-19 pandemic made this infeasible. Finally, while some students with disabilities were present in our different groups of users, neither the data collection nor the analysis identified them. On the one hand, that these students were able to participate alongside other students seems to be a positive observation. However, we recognize that this work should be tested with more students with disabilities, especially in recognition of the diversity and intersectionality that exists across the different disability communities. Nonetheless, we believe that the insights gathered thus far are still beneficial for considering the design of multimodal interfaces for equitable play. We intend to address these limitations within our on-going studies and as we continue to develop the platform. As we think more about the future development of this work, we also want to speak to an important consideration in thinking about using MMLA. Prior research in MMLA has involved the use of various multimodal sensors that can proxy for everything from arousal, to cognitive load, to mind wandering, to fine motor gesticulations. We must be careful not to use the analytics in ways that are normative and overlook the diversity that exists among and within different populations. One approach for addressing this is to look at ways that students' data deviates from their typical behaviors. Hence, even in thinking about ways that we look at student eye tracking data and examining the visual spatial anchors that they may be references, simply looking at aggregate behaviors across groups should be conducted with caution.

8 Conclusion

This project began because of our motivation to make Minecraft more accessible for students with disabilities. However, more important than simply making Minecraft more accessible, we wanted to promote a game play and social experience that would be equitable. Through our user studies, we found that the platform helps fulfill some of those goals by providing capabilities that can spur on amazement and excitement among traditional Minecraft users and novices. We also find that many of the multimodal components, while not immediately intuitive for users, proved to be preferred modes of game play. In this sense, we feel that this tool is moving in the right direction in terms of the system capabilities that it provides. Our analyses also point to the meaningful ways that multimodal data can be used to study student learning in these game-based environments, and free students from standardized testing and learning experiences. As we iterate on this platform, we look forward to creating a more robust solution that we will test among students with disabilities, and among mixed ability groups, since our goal is to support inclusive learning experiences for all students.

References

[1] NPD, "Across All Age Groups, U.S. Consumers are Investing More of Their Entertainment Participation, Time and Money on Video Games, Reports The NPD Group," 2020. [Online]. Available:

https://www.npd.com/wps/portal/npd/us/news/press-releases/2020/across-all-age-groups-us-consumers-are-investing-more-of-their-entertainment-participation/.

[2] Nielsen Media, "3, 2, 1 GO! VIDEO GAMING IS AT AN ALL-TIME HIGH DURING COVID-19," 03-Jun-2020. [Online]. Available: https://www.nielsen.com/us/en/insights/article/2020/3-2-1-go-video-gaming-is-at-an-all-time-high-during-covid-19/. [Accessed: 19-Dec-2020].

[3] J. Clement, "Increase in video game sales during the coronavirus (COVID-19) pandemic worldwide as of March 2020.," 2021. [Online]. Available: https://www.statista.com/statistics/1109977/video-game-sales-covid/. [Accessed: 02-Feb-2021].

[4] J. L. Plass, B. D. Homer, and C. K. Kinzer, "Foundations of game-based learning," *Educ. Psychol.*, vol. 50, no. 4, pp. 258–283, 2015.

[5] R. Stevens, T. Satwicz, and L. McCarthy, "In-game, in-room, in-world: Reconnecting video game play to the rest of kids' lives," *Ecol. Games Connect. Youth, Games Learn.*, 2008.

[6] K. E. Ringland, L. Boyd, H. Faucett, A. L. L. Cullen, and G. R. Hayes, "Making in Minecraft: A Means of Self-Expression for Youth with Autism," *Proc. 2017 Conf. Interact. Des. Child.*, pp. 340–345, 2017.

[7] I. Granic, A. Lobel, and R. C. M. E. Engels, "The benefits of playing video games.," *American Psychologist*, vol. 69, no. 1. American Psychological Association, Granic, Isabela: Developmental Psychopathology Department, Behavioural Science Institute, Radboud University Nijmegen, Montessorilaan 3, Nijmegen, Netherlands, 6525 HR, i.granic@pwo.ru.nl, pp. 66–78, 2014.

[8] K. E. Ringland, C. T. Wolf, L. E. Boyd, M. S. Baldwin, and G. R. Hayes, "Would You Be Mine: Appropriating Minecraft as an Assistive Technology for Youth with Autism," *Proc. 18th Int. ACM SIGACCESS Conf. Comput. Access. - ASSETS '16*, pp. 33–41, 2016.

[9] K. E. Ringland, C. T. Wolf, L. Dombrowski, and G. R. Hayes, "Making" Safe" Community-Centered Practices in a Virtual World Dedicated to Children with Autism," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 2015, pp. 1788–1800.

[10] K. E. Ringland, "A Place to Play: The (Dis)Abled Embodied Experience for Autistic Children in Online Spaces," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–14.

[11] J. O. Wobbrock, S. K. Kane, K. Z. Gajos, S. Harada, and J. Froehlich, "Ability-Based Design: Concept, Principles and Examples," *ACM Trans.*, vol. 3, no. 3, pp. 1–36, 2011.

[12] P. Blikstein and M. Worsley, "Multimodal Learning Analytics: a methodological framework for research in constructivist learning," *J. Learn. Anal.*, vol. 3, no. 2, pp. 220–238, 2016.

[13] K. Sharma and M. Giannakos, "Multimodal data capabilities for learning: What can multimodal data tell us about learning?," *Br. J. Educ. Technol.*, p. e13280, 2020.

[14] M. Worsley, "Multimodal Learning Analytics ' Past , Present , and , Potential Futures," in *Companion Proceedings of the 8th International Conference on learning Analytics & Knowledge*, 2018.

[15] D. Uttal and C. A. Cohen, "Spatial Thinking and STEM Education. When, Why, and

How?," *Psychol. Learn. Motiv. - Adv. Res. Theory*, vol. 57, pp. 147–181, 2012.

[16] K. E. Ramey, R. Stevens, and D. H. Uttal, "Steam learning in an in-school makerspace: the role of distributed spatial sensemaking," in *Proceedings of International Conference of the Learning Sciences, ICLS* , 2018, vol. 1, no. 2018-June, pp. 168–175.

[17] K. E. Ramey and D. Uttal, "Making Sense of Space: Distributed Spatial Sensemaking in a Middle School Summer Engineering Camp," *J. Learn. Sci.*, vol. 26, no. 2, pp. 277–319, 2017.

[18] C. S. Green and D. Bavelier, "Action-video-game experience alters the spatial resolution of vision: Research article," *Psychol. Sci.*, vol. 18, no. 1, pp. 88–94, 2007.

[19] H. Wauck, Z. Xiao, P.-T. Chiu, and W.-T. Fu, "Untangling the Relationship Between Spatial Skills, Game Features, and Gender in a Video Game," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces - IUI '17*, 2017.

[20] A. Nguyen and S. Rank, "Spatial Involvement in Training Mental Rotation with Minecraft," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, 2016, pp. 245–252.

[21] S. Fitzhugh, T. F. Shipley, N. Newcombe, K. McKenna, and D. Dumay, "Mental rotation of real word Shepard-Metzler figures: An eye tracking study," *J. Vis.*, vol. 8, no. 6, p. 648, 2008.

[22] M. A. Just and P. A. Carpenter, "Eye fixations and cognitive processes.," *Cogn. Psychol.*, vol. 8, no. 4, pp. 441–480, 1975.

[23] J. Xue, C. Li, C. Quan, Y. Lu, J. Yue, and C. Zhang, "Uncovering the cognitive processes underlying mental rotation: an eye-movement study," *Sci. Rep.*, vol. 7, no. 1, p. 10076, 2017.

[24] M. Resnick *et al.*, "Design Principles for Tools to Support Creative Thinking," *Science (80-.).*, vol. 20, no. 2, pp. 25–35, 2005.

[25] S. Turkle and S. Papert, "Epistemological Pluralism and the Revaluation of the Concrete," *J. Math. Behav.*, vol. 11, no. 1, pp. 1–30, 1992.

[26] "Spacy." [Online]. Available: http://spacy.io.

[27] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *Int. J. Lexicogr.*, vol. 3, no. 4, pp. 235–244, 1990.

[28] E. Tse, S. Greenberg, C. Shen, and C. Forlines, "Multimodal multiplayer tabletop gaming," *Comput. Entertain.*, vol. 5, no. 2, p. 12, 2007.

[29] G. Ganis and R. Kievit, "A new set of three-dimensional shapes for investigating mental rotation processes: validation data and stimulus set." figshare, Jun-2014.

[30] J. Wagner, F. Lingenfelser, T. Baur, I. Damian, F. Kistler, and E. André, "The social signal interpretation (SSI) framework: multimodal signal processing and recognition in real-time," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013, pp. 831–834.

[31] D. Abrahamson, S. Shayan, A. Bakker, and M. Van Der Schaaf, "Eye-Tracking Piaget: Capturing the Emergence of Attentional Anchors in the Coordination of Proportional Motor Action," *Hum. Dev.*, 2016.

[32] M. Worsley, "Seeing Spatial Reasoning," in *Companion Proceedings of 11th International Conferenece on Learning Analytics and Knowledge (LAK)*, 2021.