

*TECHNICAL INFORMATION**GAMIFIED HUMAN OPERANT RESEARCH: A BRIEF INTRODUCTION TO MINECRAFT EDUCATION*Casey Irwin Helvey^{1,2}, Lucille Gates³, Paige Rountree³, Tom Cariveau³¹RUTGERS ROBERT WOOD JOHNSON MEDICAL SCHOOL,²CHILDREN'S SPECIALIZED HOSPITAL—RUTGERS UNIVERSITY CENTER FOR AUTISM RESEARCH, EDUCATION, AND SERVICES (CSH-RUCARES),³UNIVERSITY OF NORTH CAROLINA WILMINGTON

The notion of boredom as it relates to subject interest in experimental tasks has appeared in discussions about human operant research for decades (e.g., Baron & Perone, 1998; Case et al., 1990; Galizio & Buskist, 1988; Saini & Roane, 2018). Programming dynamic, game-like experimental paradigms may help minimize issues with attrition or boredom associated with otherwise arbitrary human operant tasks (Baron & Perone, 1998; Galizio & Buskist, 1998; Pilgrim 1998; Savastano & Fantino, 1994; Schmitt, 1998). Modern devices and software are particularly suitable for executing gamified research programs due to their accessibility and malleability. Indeed, several games currently exist that could be used by researchers to present complex experimental tasks using in-game building or editing features in lieu of coding.

Establishing a research program that utilizes computer-based games can be a resource-intensive endeavor, demanding significant time and financial investment due to the reliance on individuals skilled in coding or programming. However, employing commercially available games that offer customizable features with minimal or no coding requirements (e.g., Minecraft, Unreal Editor for Fortnite, Roblox) represents a practical solution to these challenges. Many of these games come equipped with integrated tools that allow human operant researchers to create well-controlled experimental paradigms and, in some cases, automate data collection. Although researchers in behavior analysis have

used commercially available games to answer questions relevant to the experimental analysis of human behavior (e.g., Schenk & Reed, 2020), certain game mechanics, particularly those that cannot be manipulated by the experimenter, can introduce experimental confounds that may impact the findings of the study (McMahan et al., 2011; Schenk & Reed, 2020). For example, in Mario Kart, randomly generated items (e.g., bananas, red shells, mushrooms, coins) may have greater effects on responding than any possible contingencies arranged by the experimenter. Alternatively, commercially available games that are highly customizable, such as sandbox games, offer a unique advantage: researchers can design highly controlled but dynamic experimental paradigms for the study of human operant behavior.

The term *sandbox game* describes a class of games that include open-world elements, meaning that the player's movement and interactions within the game are largely unrestricted such that tasks that are player, rather than game or plot, initiated (Rouse, 2022). These games may be contrasted with other games that include progression-based and linear storylines much like the plot of a movie. The customization of sandbox games may be particularly relevant to human operant researchers as the many open-world elements allow for diverse antecedent and consequent conditions to be arranged in video game environments. Although participant's experience with these or similar video games might represent a possible experimental confound, given the versatility of sandbox games, human operant researchers can arrange discrete response-consequence relations that are unique to the experimental paradigm. Relatedly, participants' history with these or related video games might benefit the human operant researcher as participants might be

Author Note: Address correspondence to Casey Irwin Helvey, CSH-RUCARES, 888 Easton Ave., Somerset, NJ 08873. Email: cih18@rutgers.edu

<https://doi.org/10.17605/OSF.IO/VT58F>

familiar with the in-game controls. Such a history might absolve the researcher from providing extensive training or instruction before the experimental conditions can be introduced.

To highlight the potential power of sandbox games as a tool for developing experimental procedures, the current paper will briefly introduce Minecraft Education (hereafter *Minecraft*), which we have used in our lab to conduct a small program of human operant research. Minecraft is played in a three-dimensional world in which players can use different tools (e.g., pickaxe) and items to interact with distinct environmental components (e.g., block types). In 2016, Microsoft, in collaboration with Mojang Studios, released an education edition of the game which was designed to be used as an instructional tool in classroom settings. This version of Minecraft allows users to engage in single- and multi-player game modes across more than 500 user- and developer-built lessons in science, coding, history, and various other subjects (Minecraft, n.d.). Minecraft can be used across various platforms or devices, including Mac, Windows, Chromebook, iPad, iPhone, and Android phones and tablets. This wide accessibility is noteworthy as it allows a diverse range of participants and researchers to engage with the software. Just one year after being released, Minecraft had over 2 million licensed users in over 115 countries worldwide. At the time of this paper, licenses are free to anyone with an active Microsoft Office 365 Education (.edu) account or can be purchased for \$12.99 annually per user for individuals who are not part of an eligible educational institution. By virtue of its expansive and adaptable three-dimensional world and widespread accessibility, Minecraft is an innovative program for developing experimental procedures to study human operant behavior.

USING MINECRAFT

The following sections will describe several features of Minecraft that we have used to develop or manage human operant paradigms. Additional links that include videos or other descriptions of this content are accessible as a Supplemental Material.

User Controls

From the participant's perspective, navigating the Minecraft world requires mastery of only a few controls. These controls differ across devices (e.g., computer or tablet). Because most human operant laboratories continue to use standard desktop computers, computer controls will be described but readers are referred to the Supplemental Material for controls on other platforms. Once a participant joins a Minecraft world using a unique code, they can move their character using the WASD keys on the keyboard. Participants can also alter their view by moving the mouse, left-click to break (i.e., mine) blocks, and right-click to place blocks or interact with items in the world (e.g., press buttons). These controls are assigned by default but can be modified or removed by the experimenter.

Hosting and Player Permissions

A key feature of Minecraft is the ability to host a world that is remotely accessible to other players from any device. The hosting player (i.e., the experimenter) can invite other players to join by sharing a unique code or link. Although participants may join remotely from any location, our lab required participants to join the experimental world from a room in the lab equipped with a desk, chair, and desktop computer. This in-person requirement was used to reduce possible experimental confounds such as access to (a) different peripherals (e.g., computer mouse and keyboard vs. trackpad), (b) internet or graphic capabilities of the device, or (c) distraction-free settings. Moreover, the experimenter was able to monitor the participant during the experimental tasks both in the room and in the game using a one-way mirror and recording software, respectively. Allowing remote participation might enhance the recruitment of diverse and understudied populations in human operant research, but researchers should consider additional safeguards, such as remote monitoring to screen for potential impostor participants (e.g., bots) and to account for extraneous influences on participant responding. In-game observations also allow for periodic fidelity checks to ensure that the player permissions are set properly.

A standard set of player permissions can be programmed by assigning the participant to

one of three in-game roles: member, visitor, or operator. *Members* can place or break blocks and other items in the world, whereas *visitors* can only explore without interacting with blocks or items. *Operators* can select default permissions for each player and execute slash commands (described below). In our lab, participants' roles are set as *members* and other game rules are set (described below) to ensure that the participant cannot alter the environment. Additional settings can be programmed to best suit the experimenters' needs including various game rules and world settings (e.g., visibility, weather conditions). Although these tools were intended to help educators customize instructional lessons and manage students' experiences, they can also be utilized by human operant researchers to maintain controlled experimental conditions.

Two game rules that are important to highlight are (a) world builder and (b) immutable world. Both game rules can be toggled between *true* and *false* by an operator. Only when world builder is set to *true* can operators and members place or break blocks in the Minecraft world. When world builder is set to *false*, players will not be able to place or break blocks; however, the player can interact with other elements of the world, such as pressing buttons or levers, shooting arrows, and picking up items. Setting world builder to *false* is recommended during experimental tasks so that the participant cannot alter other features of the environment. The second game rule is *immutable world*, which can be used to allow members and operators without the world builder ability to place or break blocks. Specifically, if *immutable world* is set to *false*, then players can change the world regardless of their world builder status. However, if *immutable world* is set to *true*, then players can only change the world if their world builder status is also set to *true*.

Executing Commands

Human operant researchers might also utilize various in-game commands (known as *slash commands*), which allow the experimenter to present in-game effects (e.g., teleporting or delivering an item to a participant). Several default commands can be used to perform various operations in the world. To enable commands, the experimenter must *activate cheats* in the settings menu. If a command is

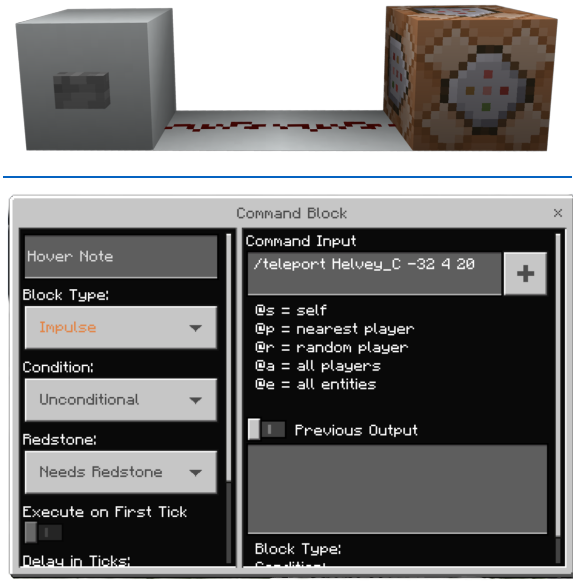


Figure 1. Example Logic Gate (top) and Command Block User Interface (bottom).


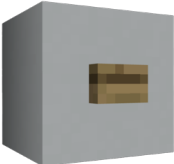
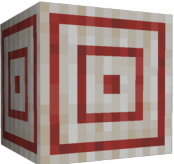

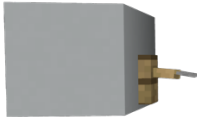
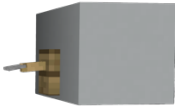
used infrequently, the experimenter can execute any command using the chat window. If a command may be executed repeatedly or triggered by some environmental event (e.g., participant's responding), then the experimenter can instead program the command into a *command block*, which can be placed anywhere in the world. Command blocks are not regularly available in the inventory and must be obtained using a command in the chat window. Once placed, each command block has an interface where the designated command can be input (Figure 1, bottom panel).

Human operant researchers should also be aware of the *clone* command. Using this command, researchers are able to duplicate previously built structures or paradigms. The experimenter must define the area to be cloned as well as the final placement of the cloned structure. In our research, cloning has been used to duplicate chambers which serve as individual trials. Without this function, reproducing identical chambers would represent a nearly insurmountable task.

Circuitry

Minecraft includes powered blocks, known as *redstone*, that can be used as digital logic gates. Logic gates are electronic circuits that process binary information: 0 (i.e., OFF) or 1

Table 1. Examples of In-Game Manipulandum

Name	Item	Effect
Lever		Pressing the lever closes/opens the circuit.
Button		Pressing the wooden button closes the circuit for 2 s.
Target		Shooting the target with a bow and arrow closes the circuit for 2 s.
Pressure plate		Standing on the pressure plate closes the circuit.
Trip wire	 	Walking through the trip wire closes the circuit for 2 s.

(i.e., ON). To build a simple input/output logic gate in Minecraft, the experimenter can connect a lever, or other inputs, to a line of redstone. Right-clicking the lever will power the redstone (i.e., close the circuit) and right-clicking again will remove power (i.e., open the circuit) to the redstone.

Several distinct manipulanda can be used as the input for a simple logic gate (see Table 1), which can then be connected to various in-game mechanics (see Table 2) as outputs. An example logic gate using a button and command block to teleport a participant (Helvey_C) is shown in Figure 1. Pressing the button would power the logic gate and execute the teleport command through the command block. Redstone is not required to power a command block as many of the manipulanda can be placed directly on the block by crouching (Shift) and right clicking on the block. However, experimenters may choose to use redstone to connect the manipulandum






and command block so that it (a) is not visible to the participant, (b) can be connected to other mechanics, or (c) cannot be accessed by the participant.

Specialty Blocks and Crafting Features

Minecraft includes a variety of blocks that experimenters can use to create and customize their experimental paradigm. Several distinct types of blocks may be used to build (e.g., grass, dirt, stone, glass), decorate (e.g., flowers, banners, paintings), transport (e.g., rails, minecarts, boats), and provide instructions to participants (e.g., signs).

In addition to these basic blocks, Minecraft also features several specialty blocks that can be placed to constrain participants' activity. First, *border blocks* may be used to create an invisible barrier, which can be placed inconspicuously in

Table 2. Examples of In-Game Effects.

Consequence	Effect	Slash command	In-game effect
Door	A door can be opened given a redstone input.		
Redstone lamp	A lamp can be illuminated given a redstone input.		
Dropper	An item can be dispensed from a dropper given a redstone input.		
Title message	A message can be presented across the participants' screen using a slash command.	<code>/title [participant] title [-1 /+1] Point</code>	 
Nausea	The participant's view is distorted.	<code>/effect [participant] nausea</code>	
Speed	The participant's walking speed can be increased or decreased.	<code>/effect [participant] speed or slowness</code>	
Give/remove	An item can be delivered or removed from the participant's inventory.	<code>/give or clear [participant] [item name] #</code>	
Teleport	The participant can be teleported to a different coordinate.	<code>/teleport [participant] [X Y Z]</code>	

the paradigm (e.g., underground). Participants are unable to pass over or under these blocks. Border blocks can be placed to keep participants in certain areas and prevent them from accessing key components of the experimental paradigm (e.g., command blocks). The second type of specialty block that we have found useful for research is the *allow block*. An allow block permits users to place and break blocks in areas above them, even when other player permissions would not otherwise allow building. These blocks can be placed in areas that the researchers plan to allow participants to build, such as part of a free-operant experimental task or in an area of the world that participants are given contingent access to during reinforcement. The third type of specialty block is the *deny block*, which prevents users from placing or breaking blocks in areas above them, even when other player permissions would otherwise permit building.

Program Elements Relevant to Human Operant Researchers

We began using Minecraft in our research in 2019 as it represented a particularly malleable program that could be used to generate complex experimental tasks for researchers without coding experience. Although our initial paradigms would best be described as replicas of a standard operant chamber, researchers can include more sophisticated arrangements to capitalize on the various game elements that can be programmed in Minecraft. As one example, we recently prepared a dungeon-like map (see Supplemental Materials for video example of one such paradigm) in which participants progressed through dark chambers as they completed experimental tasks (e.g., delayed matching-to-sample, concurrent operants). Doing so allowed for the repeated measurement of the target response, while avoiding the tedious stationary button-pressing of our initial paradigms.

When creating a gamified research paradigm in Minecraft, the human operant researcher may be heartened by the substantial variety of blocks at their disposal. These blocks can be used to arrange various contexts, discriminative conditions, or bolster the general aesthetics of their paradigm. A description of the types of blocks is beyond the scope of this paper and may best be realized by exploring the game. Several manipulanda can also be arranged as can various effects used as consequent events. Some examples of the manipulanda and consequence events are described below, although they represent only a portion of possible mechanics that may be used by human operant researchers.

The manipulanda available in Minecraft can be used to trigger consequent events directly or according to whatever experimental parameter is of interest to the researcher (e.g., schedule, delay). Some of the manipulanda are shown in Table 1, which include a lever, button, target, pressure plate, and trip wire. These represent the majority of the manipulanda used by our lab and each serves a distinct function therein. As an example, the lever can be rapidly pressed using the right-mouse button, which results in the logic gate being powered on or off. In contrast, right clicking a button will power the gate for a brief period (e.g., 2 s) before automatically turning off. The duration of the interval differs across the types of buttons (e.g., wood or stone) and another response cannot be emitted until the interval has elapsed. These manipulanda can then be connected to various outputs.

Table 2 shows a small sample of outputs that may be of interest to the human operant researcher. These outputs are not specific to any given input mechanism. Indeed, the same manipulanda could open a door, produce access to a new area of a chamber, deliver or remove points and items, or spawn enemy players. The human operant researcher can use these outputs to setup whatever conditions are germane to their research program.

In addition to the mechanical outputs that may be arranged in Minecraft, experimenters can also arrange in-game status effects. Like the outputs described above, these participant-directed effects can be added or removed following some environmental event or at any point by the experimenter. One example is the nausea effect, which alters the participant's

field-of-view by creating a wobble effect. Experimenters can also alter the participants' speed across a range of values (i.e., 1=slowest, 255=fastest). These effects can also be applied for a set duration, can be terminated, or can be reinstated based on an environmental event. Overall, the abundance of manipulanda and effects that can be programmed in Minecraft offers incredible flexibility to human operant researchers, which can be leveraged to study a myriad of behavioral phenomena.

Another program element that can enhance the utility of Minecraft in human operant research is automated data collection. Experimenters can automatically record the participant's responding on the manipulanda by adding a *scoreboard* using a slash command (see Supplemental Materials). Once a scoreboard is added, the experimenter can arrange for any response by the participant to be logged in the scoreboard. This scoreboard can be made visible to the participants or only to the experimenter. The placement of the scoreboard can also be set by the experimenter. Specifically, the scoreboard can be made to appear on the right side of the participant's screen or accessible in the player menu, which is accessed by pressing the Escape key. The scoreboard can also be reset to allow for trial- or session-specific data collection.

ADVANTAGES AND POTENTIAL USES

Minecraft is a software program with many advantages for human operant researchers interested in gamifying their research paradigms. Some of the most notable advantages include: (a) the ability to host and join an experiment from a wide range of devices, (b) user-friendly interfaces and controls, and (c) the considerable number of aesthetic and functional resources that are available in the game. These advantages may serve the human operant researcher by allowing for: (a) greater accessibility by understudied populations, (b) reduced instructional or training requirements, and (c) nearly endless customization of experimental paradigms.

Given the abundance of aesthetic and functional designs available in Minecraft, we anticipate that this program would not be limited to any particular research question. Nevertheless, we have found it to be a

particularly helpful paradigm for exploring our research interests in social behavior, observing behavior, and choice. As one example, we have conducted several experiments on dyadic competition, which sought to extend the limited research on competitive and cooperative contingencies (e.g., Schmitt, 1976; 1998). Previous research in this area has reported instances of participant attrition (Schmitt, 1976), which were likely related to features of the study (e.g., schedule of reinforcement), but also the monotony of the task and associated operanda (e.g., Lindsley knob, audit counters, toggles, and switches). Given that most video games include a cooperative or competitive contingency, we felt that similar gamified arrangements, developed in Minecraft, would be a meaningful approach to extend the previous work of David Schmitt and Don Hake (e.g., Hake & Vukelich, 1972; Schmitt, 1984; 1986). Further, some of the challenges faced by these researchers were related to the delivery of conditioned reinforcers (Schmitt & Marwell, 1972), which might also be diminished in video game paradigms. Indeed, a large amount of behavior and money is allocated to the accumulation of in-game resources with no tangible benefit to the player (Marder et al., 2019).

We also recently began to explore ways to arrange in-game conditioned reinforcers, which has proven to be a fruitful area of exploration particularly as it relates to the financial sustainability of our lab. For example, experimenters may program the delivery of in-game items (e.g., gems) as tokens, which may be exchanged for other in-game resources (e.g., apparel, weaponry) instead of traditional monetary compensation that is delivered contingent on performance or general participation. We hope that similar gamified arrangements might be utilized by other research labs and, in doing so, might provoke the human operant researcher to consider alternatives to trinkets, candies, or raffle tickets in their own research program. While our research using Minecraft has predominantly employed discrete-trial or performance-based arrangements, the nature of Minecraft as a sandbox game with minimal character or environment restrictions makes it an ideal program for researchers interested in studying free operant behavior.

We hope that we have sufficiently described the potential utility and malleability of

Minecraft in human operant research and also goaded our fellow researchers into considering how their own research questions may be pursued in a gamified format. Doing so might help sustain otherwise costly research programs, but also increase participation, enjoyment, and acceptability of our procedures.

REFERENCES

- Baron, A., Perone, M. (1998). Experimental design and analysis in the laboratory study of human operant behavior. In K. A. Lattal & M. Perone (Eds.), *Handbook of research methods in human operant behavior* (pp. 45-91). Springer. https://doi.org/10.1007/978-1-4899-1947-2_3
- Case, D. A., Ploog, B. O., & Fantino, E. (1990). Observing behavior in a computer game. *Journal of the Experimental Analysis of Behavior*, 54(3), 185-199. <https://doi.org/10.1901/jeab.1990.54-185>
- Galizio, M., & Buskist, W. (1988). Laboratory lore and research practices in the experimental analysis of human behavior: Selecting reinforcers and arranging contingencies. *The Behavior Analyst*, 11(1), 65-69. <https://doi.org/10.1007/BF03392457>
- Hake, D. F., & Vukelich, R. (1972). A classification and review of cooperation procedures. *Journal of the Experimental Analysis of Behavior*, 18(2), 333-343. <https://doi.org/10.1901/jeab.1972.18-333>
- Marder, B., Gattig, D., Collins, E., Pitt, L., Kietzmann, J., & Erz, A. (2019). The avatar's new clothes: Understanding why players purchase non-functional items in free-to-play games. *Computers in Human Behavior*, 91, 72-83. <https://doi.org/10.1016/j.chb.2018.09.006>
- McMahan, R. P., Ragan, E. D., Leal, A., Beaton, R. J., & Bowman, D. A. (2011). Considerations for the use of commercial video games in controlled experiments. *Entertainment Computing*, 2(1), 3-9. <https://doi.org/10.1016/j.entcom.2011.03.002>
- Minecraft Education*. (n.d.) Retrieved March 27, 2023, from <https://education.minecraft.net/en-us>
- Pilgrim, C. (1998). The human subject. In K. A. Lattal & M. Perone (Eds.), *Handbook of research methods in human operant behavior* (pp. 15-44). Springer. https://doi.org/10.1007/978-1-4899-1947-2_2

- Rouse, M. (2022, June 3). *Sandbox*. Techopedia. <https://www.techopedia.com/definition/3952/sandbox-gaming>
- Saini, V., & Roane, H. S. (2018). Technological advances in the experimental analysis of human behavior. *Behavior Analysis: Research and Practice*, 18(3), 288-304. <https://doi.org/10.1037/bar0000124>
- Savastano, H. I., & Fantino, E. (1994). Human choice in concurrent ratio-interval schedules of reinforcement. *Journal of the Experimental Analysis of Behavior*, 61(3), 453-463. <https://doi.org/10.1901/jeab.1994.61-453>
- Schenk, M. J., & Reed, D. D. (2020). Experimental evaluation of matching via a commercially available basketball video game. *Journal of Applied Behavior Analysis*, 53(1), 209-221. <https://doi.org/10.1002/jaba.551>
- Schmitt, D. R. (1976). Some conditions affecting the choice to cooperate or compete. *Journal of the Experimental Analysis of Behavior*, 25(2), 165-178. <https://doi.org/10.1901/jeab.1976.25-165>
- Schmitt, D. (1984). Interpersonal relations: Cooperation and competition. *Journal of the Experimental Analysis of Behavior*, 42(3), 377-383. <https://doi.org/10.1901/jeab.1984.42-377>
- Schmitt, D. (1986). Competition: Some behavioral issues. *The Behavior Analyst*, 9(1), 27-34. <https://doi.org/10.1007/BF03391927>
- Schmitt, D.R. (1998). Social behavior. In K. A. Lattal & M. Perone (Eds.), *Handbook of research methods in human operant behavior* (pp. 471-505). Springer. https://doi.org/10.1007/978-1-4899-1947-2_15
- Schmitt, D. R., & Marwell, G. (1972). Experimental use of points or money as reinforcers: Does what is risked make a difference? *Psychological Reports*, 31(2), 425-426. <https://doi.org/10.2466/pr0.1972.31.2.42>