

ComputerCraft: Teaching Programming with Minecraft

Brendon Trombley, Don Miller, Institute of Play
brendon@instituteofplay.org, don@instituteofplay.org

Abstract: Computer programming is an increasingly important skill for young people to learn. However, it remains fairly inaccessible to the majority of learners. ComputerCraft is an add-on that adds working computers and robots into the popular sandbox game Minecraft. These objects are programmable through the Lua scripting language and can be used to complete tasks inside the Minecraft world. Through ComputerCraft, programming becomes an immediate and useful skill, allowing students to accomplish more in Minecraft's virtual space. This workshop models an introduction to ComputerCraft, aimed at educators teaching students. The players' first challenge is to create a "Hello World" program. Next, players encounter robots known as "Turtles", and learn to issue simple commands to them. Finally, players are challenged to program a Turtle to move along a predetermined path. As a result of these exercises, players gain an understanding of ComputerCraft, and by extension, computer programming in general.

Introduction

Computer Programming and Young Learners

Computer programming is an increasingly important skill for young people to learn. Learning to program opens more doors than ever before, and not just traditional computer science jobs like database programming. The Internet grows larger each year, and there is a need for both front-end (user facing) and back-end (server-side) programmers. In-demand creative coders are hired around the world at high-profile advertising agencies. The proliferation of smart phones proves that millions of people are exposed to scores of new apps and games each day. Additionally, the video game industry is not slowing down, and many triple-A games employ a great number of programmers in many different roles, from physics to artificial intelligence programming. Even indie games are breaking into the market as downloadable titles on just about every current generation handheld and home console. All of these jobs require people who know how to program. Many young people play these very same video games each day, with Minecraft being a very popular example. Dedicated players will spend hours creating virtual worlds in Minecraft. However, they may not completely understand how the game works behind the scenes. This is because computer programming remains fairly inaccessible to the majority of young learners.

Learning a programming language, even a simple one, can be a daunting task. There are hundreds of programming languages to choose from, each made for a variety of tasks. Setting up the software needed for the programming environment can be complex. After that, learning the syntax and the ability to debug can be an additional challenge. It requires hours of study even to produce simple results, which may not always prove interesting to young people. The amount of time required to become proficient and create games or apps is quite high, and even then, the results aren't normally as polished as games like Minecraft that so easily captivate their attention.

Minecraft

Minecraft is a indie game created by Markus "Notch" Persson, first released in 2009. Four years later it exists on many platforms and consoles: Windows, OSX, iOS, Android, and Xbox 360. It can be described as a "sandbox" game, allowing players to roam free and interact with the virtual world with little restraint. The world is procedurally generated and players can create constructions out of cubes made from various materials. Players are encouraged to mine to gather resources, as well as craft new cubes and tools. In survival mode players must gather resources and maintain their health. Creative mode removes those restraints and grants players the ability to fly and wield unlimited resources.

User-generated content for Minecraft is extremely popular and is one of the reasons the game continues to be well-liked years after its initial release. The content includes texture packs that change the look of the game, custom maps that mimic real world or famous fictional locations, and more complex modifications that offer new blocks, new enemies, and new ways to craft objects and weapons. ComputerCraft is one of the newer mods, being released in 2011.

ComputerCraft and the Lua programming language

ComputerCraft is a mod created by programmer Daniel Ratcliffe for Minecraft that adds computers, monitors, modems, and turtles to the game. All of these objects are virtual, but can be fully programmed and used to automate tasks that players in Minecraft usually execute manually while playing the game. Turtles can be programmed to

dig, mine, build, and travel large distances automatically. All ComputerCraft objects are programmed with Lua, a real-life programming language that is easy to learn and high-level, meaning it is more abstract and reads like English. Lua is a programming language that was influenced by C++ and first appeared in 1993. As it is a fast-executing scripting language, it can be embedded easily in lower level (less abstract and more complex) C and C++ code. For these reasons, it is often used in video game development.

Together, Minecraft and ComputerCraft create an ideal programming environment for kids in a number of ways:

1. Many students are already familiar with and passionate about Minecraft, unlike traditional IDEs such as Eclipse.
2. Getting “up and running” with code is easy and straightforward after installing the ComputerCraft mod. No additional software is needed.
3. The use of code is simple and powerful with little fear of failure. Students can program robotic turtles to complete tasks with a few lines of code. Unlike traditional programming mistakes will not lead to software locking up or causing problems within the operating system.
4. Iterating the code is fast and easy, and changes are readily apparent. A student can quickly test a program, edit the code, and test it again to see the new results.
5. The “Need to Know” the learning objectives is strong. Students can use their new skills to design and execute solutions to in-game problems.

Minecraft and schools

At Quest to Learn, a NYC public school that incorporates game-based learning into its curriculum, Minecraft is used both formally and informally. In Sports for the Mind, a 6th-8th grade class centered around design and media skills, students play multiplayer Minecraft to develop their skills in collaboration and designing solutions to problems. Students have recreated NYC landmarks such as the Empire State Building, earned badges for working together as a team, and forged alliances with other groups of students. During lunch and after school, students flock to various clubs to continue playing or even learn how to make their own modifications to the game. Some students create new avatars or texture packs, while others delve right into the Java code that makes the game tick. Quest to Learn students and faculty alike find Minecraft to be a flexible, immersive, and highly engaging learning environment.

ComputerCraft Workshop

Workshop Plan

Participants in the workshop will take place in the Minecraft Tekkit Turtle Challenge. This challenge consists has four main learning goals:

1. Navigating the ComputerCraft OS
2. Writing programs in Lua
3. Learning the Turtle API
4. Programming a Turtle to accomplish a task

Our virtual ComputerCraft workshop has three main rooms, each containing several tasks to execute. Below is an overview.

Room 1

1. Using a computer - Players will find the computer block across from the starting point, and can begin using it by walking up to it and right-clicking. This will be present them with the basic ComputerCraft screen.
2. Navigating the OS - Players will type commands such as `cd` and `ls` to navigate directories and check their contents. See Figure 1 below.
3. Finding and running programs - With the help and programs commands, players will discover all the built-in features of the computer and learn a bit about its capabilities.
4. Writing your first program - Players will use the edit command to enter the Lua scripting interface. From there, they will write a basic “Hello world” program and run it.
5. Your program to-go - Players will learn how to use copy to transfer their program from the computer to a ComputerCraft floppy disk. They’ll use this disk to carry their programs with them in the game world.



Figure 1: The TurtleOS

Disco Room

1. Intro to Turtles - Players will travel to a festively-lit room and encounter two ComputerCraft robots, known as Turtles. They'll learn how the Turtles act much like the computer they used in Room 1
2. Using a turtle - Players use a Turtle the same way they used a computer: by walking up to it and right-clicking.
3. Exploring the turtle - By using the help and programs commands, players will learn more about the built-in features of Turtles.
4. Starting the party - Players discover a command called dance built into the turtle. Typing in the command will trigger the Turtle to enter a dancing sequence, sliding and spinning around the Disco Room.
5. It takes two - Players can enter the dance command into the 2nd Turtle.

Challenge Room

1. Intro to final challenge - Players will travel to the final room which contains a Turtle, disk drive, and a glowing white pathway (see Figure 2 below). Their challenge is to program the turtle to walk the pathway.
2. Reminders - Players will be refreshed on the use of disks, cd, ls, and edit to write and save programs.
3. List of Turtle commands - Players will be introduced to the commands for this challenge such as turtle.forward() or turtle.turnLeft().
4. Commence challenge - From here, players will write their programs, testing and fixing until they get their Turtle to navigate the pathway (see Figure 3 below).
5. Extension - Players who complete the challenge early may use a loop to cause their Turtles to enter a looping dance at the end of the pathway.

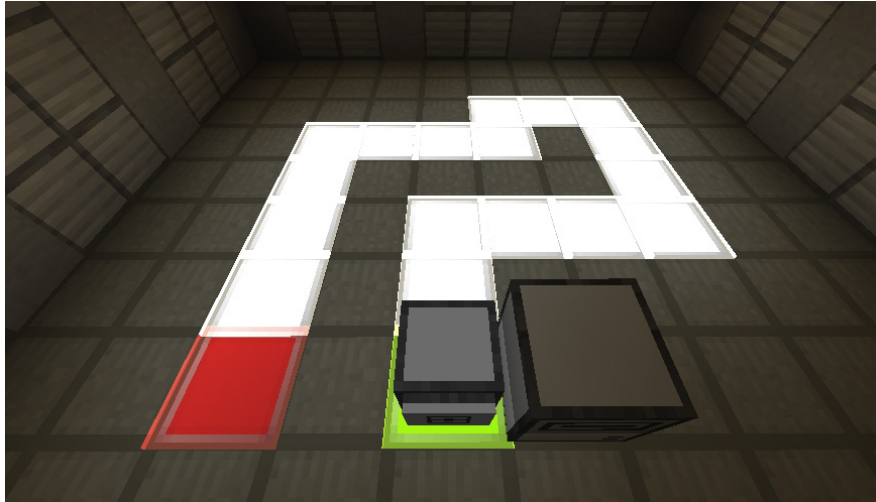


Figure 2: The Glowing Path



Figure 3: Turtle Commands

Playtesting Feedback

We playtested the demo with Ariel Kitch, a Sports for the Mind (design and media class) teacher at Quest to Learn. Here are some followup questions we asked, followed by her responses.

What did you learn during the ComputerCraft demo with Brendon?

I learned how programming turtles worked in Minecraft. I learned the basics of using computers and programming the turtles to do specific tasks.

How could you imagine this demo and/or ComputerCraft being used in your classroom?

I think the ComputerCraft mod would make a great immersive environment for students to learn basic programming once they know Minecraft essentials. I was really impressed with how the world made it easy to get involved in basic programming.

Why might you teach programming inside of a game world like Minecraft?

It gives students the ability to learn programming without worrying about text editors - thus it's a lot more accessible to students.

We also playtested the demo with some Quest to Learn 7th and 8th graders. Here is some notable feedback from those students.

What can your classmates learn from playing this demo?

Student 1: They can learn about coding while having fun controlling the turtle

Student 2: They can program easily

What was your favorite thing about the demo?

Student 1: Getting my turtle through the maze on my first try

Student 2: Making the robot spin

What was your least favorite thing about the demo?

Student 1: Having to type the same code over and over

Student 2: NOTHING

It's interesting to note that Student 1's comment about typing the same code over and over shows a potential readiness to learn new techniques that make writing code more efficient and reusable.

Lastly, we ran the ComputerCraft workshop at GLS 2013 with positive results. A large variety of people attended, from novice to expert levels of skill in Minecraft and programming. By the end of the workshop, the majority of attendees were at least partially through the final challenge. Afterward, number of people expressed the desire to use ComputerCraft with their own students, and several complimented the interactivity of the workshop itself. We received some constructive criticism regarding the somewhat quick pacing and high cognitive load for novice users.

Workshop Needs:

1. Attendees should have enough experience with Minecraft to know the controls and basics of navigating the environment.
2. They should download the zip file containing the workshop files. The location of file is forthcoming.
3. An internet connection for attendees during the session may be necessary

Acknowledgments

Thank you to Markus Persson and Mojang, the creators of Minecraft. Thank you to Daniel Ratcliffe, the creator of ComputerCraft. Thank you to the staff and students of Quest to Learn who helped with the playtests. Finally, thank you to Joel Levin, John Murphy, and Grant Tumen for their assistance during the GLS workshop.