pi-top

Coding Minecraft: online course with pi-top [4]

- **Course includes pi-top [4] Complete & Sensor Foundation Kit**
- **12 one hour sessions**
- **One-to-one tutoring**
- **For ages 11 to 17, beginner to intermediate level**
- **Tutors from top universities and with CR checks**

This course teaches kids how to program the game they love to play every day.

They'll connect physical components such as buttons and sound & light sensors to the pi-top [4], which they can use to activate and control their code and extend their learning into the real world. We learn by doing!

Students will learn the fundamentals of game development as they invent their own logic-based puzzles, gadgets and devices, and build their own worlds. Once they've mastered coding basics such as variables, if/then statements and constructors using coding blocks, they'll progress to writing their own code in Python.

Standards

Content aligns to CSTA Level 2 Standards:
2-CS-02; 2-CS-03; 2-DA-07; 2-AP-11; 2-AP-12; 2-AP-13; 2-AP-14; 2-AP-16; 2-AP-17; 2-AP-19

Course objectives

Outcome:

- Students will learn how to use computers to make things and change things - not just to consume content. They will learn about how computer programs are made, how they work and how to program them.

Essential Questions:

- What makes a good program/code?
- How can programs make the things I do *easier*?
- How can programs make the things I do *better*?

Students will understand/know:

- What programming and coding mean
- How programs are written and how they run
- How troubleshooting works
- How to diagram/flow-chart processes

Students will be able to:

- Built and prototype physical computing projects
- Write and run Python code

Learning Plan

Day 1: Intro to pi-top, Minecraft Pi

Overview

Getting started: setting up our pi-top and exploring in Minecraft.

Activities

- Set up pi-top [4]
- Update Minecraft via terminal
- Launch Minecraft
- Explore

Day 2: Intro to Python on Minecraft Pi

Review

Yesterday we started up our pi-top and explored in Minecraft. What did you do in Minecraft that was fun? What parts of Minecraft do you wish you could skip, do faster, or change?

Overview

Programming basics. What is a program? What is code? What can programs do for us? What can they do for us in Minecraft?

Activities

- Import libraries, run code
- Hello World
- Teleport
- Build a block

Day 3: Intro to Sensor Foundation Kit + Minecraft + Python

Review

Yesterday we wrote our first basic codes in Minecraft. How can we use the codes we made yesterday to help change or improve the game experience?

Overview
Today we'll add new physical hardware that can help us activate our code. How can this make playing Minecraft easier or better?

Activities

Adding buttons:

- Button teleport
- Button hello world
- Button block

Day 4: Python + Minecraft blocks + buttons

Review
Yesterday we learned to use a physical button to change how we play Minecraft and how we use code. What else could we do with the button? What could we do if we had other components?

Overview
Today we will learn how to do more complex things with the button and how to use more than one button.

Activities

- Different blocks
- Multiple blocks placed at once
- Different buttons doing different things in the same program

Day 5: Beyond buttons

Review
Yesterday we learned to do complex things with our button and how to add more than one button that can both be running at the same time. What could we use to make these things happen besides a button?

Overview
Today we'll replace the button with other inputs, we'll also make new things happen.

Activities

- Replace the button with a light sensor
- Replace the button with a sound sensor
- Light sensor + button torch path (if light is detected and the button is pressed, a trail of torches will be generated behind the player as they move)

Day 6: Potentiometer, variables

Review
Yesterday we replaced buttons with other components such as sound & light sensors.

Overview

Today we'll be learning about potentiometers and using them to control things.

Activities

- Character position x and z are controlled by potentiometers.
- Make a program that places a block when a button is pressed. Program the potentiometer to control what block type is placed when the button is pressed.
- Second potentiometer controls the size of the cube placed. When you have it set to 0 one block is placed. As you turn it to the right, it allows you to build a cube of the given material. In chat it will state how big the cube will be.

Day 7: Ultrasonic sensor

Review

Yesterday we learned about potentiometers and using them to control things.

Overview

Sensor component tutorial.

Activities

- Make a program that teleports a character whenever the button is pressed. The Y (height) is determined by the ultrasonic sensor distance.
- Figure out what the reading is when the sensor is pointed at the ceiling with no obstructions between it and the ceiling. This value should leave their Y at its current value. Anything less than that, but above zero, should change the Y value. This way, if the button is pressed while a hand is above the ultrasonic sensor, the character will teleport to a representative height. While the button is held down, the character can be raised and lowered by moving the hand.

Day 8: Build an elaborate contraption/tool

Overview

Create an object-building mechanism.

Activities

- Object-building mechanism: a Python tool capable of building objects (trees, houses, etc) with triggers and controls coming from the Sensor Foundation Kit and pi-top [4]
- The OLED displays text of what will be generated, the button places the object, the potentiometer controls how far away from the player it is placed

Day 9: Add libraries/modules

Overview

Create a system that references an “object module.”

Activities

- Plug into a pre-made module with a library of objects that can be constructed automatically in minecraft.
- Import the module and call functions from it.
- Write a script that allows them to select objects from the module using the OLED. For example, a script that displays one item at a time from a list of items (tree, house, castle, etc) whichever word is displayed on screen, that object will be built when a button is pressed.

Day 10: Contraption/tool with libraries

Overview

Today we build on our own modules and pull from them.

Activities

- Rebuild the project from yesterday, but using your own modules.
- How many different modules can you build? (One that is just a library of trees, one that is a library of buildings, one that is a library of other structures, etc)

Days 11 & 12: Construction days

Overview

For the last two sessions, we’ll apply what we’ve learned to construct projects following your own interests.