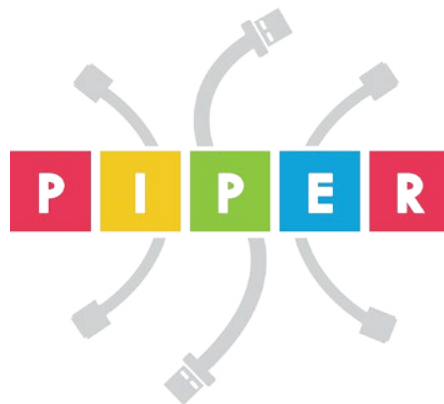
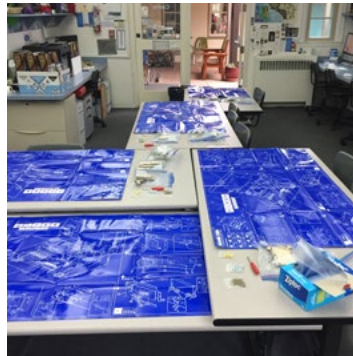
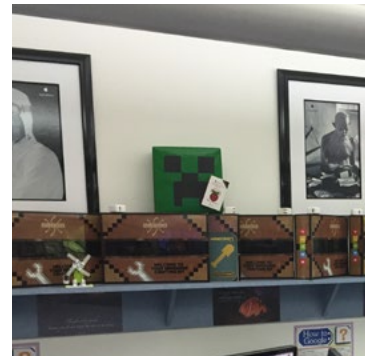
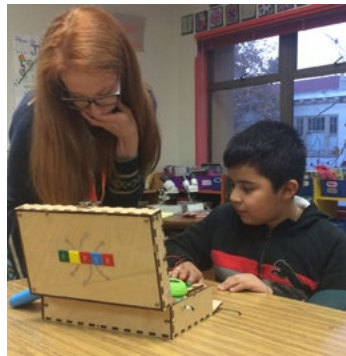


Piper Unit Plan



Introducing the Piper Unit Plan

At Piper, our educational goal is to amplify kids' creative confidence around computers and electronics. The personal computer, the most powerful invention of the 20th century, is yet to be fully implemented in the 21st century curriculum. And with the addition of Minecraft, the engagement level of students to learn engineering theory increased. With Piper, teachers and students collaborate to learn basic engineering skills and build computers and electronics at a historically low cost. Introduced from a young age, this kind of experience can fundamentally shape a child's relationship with the computer and break down the barriers and gates that keep young people (especially women and underrepresented minorities) from passionately pursuing computer science later in life. Piper users will see themselves as more than passive consumers of technology.

The curriculum presented here is designed to be a flexible tool for teachers to make Piper teachable to any age group. Teachers will be able to teach Piper starting as early as 2nd or 3rd grade (ages 7-8) all the way up to high schoolers with little prior hardware experience. For the public education system, we facilitate many options of standards to pick from to meet your specific needs.

Thank you for choosing Piper!

The Piper team

Glossary

BreadBoard: A flexible wiring board that lets us connect components to the RPi.

Buttons: Basic electronic components to momentarily close circuits.

GPIO Pins (General Purpose Input Output): A programmable interface that connects almost anything (such as buttons and LEDs) to RPi.

Jumper Wires: Wires used to connect from the GPIO to the breadboard. Ours are 'socket' to 'pin', but they can have any combination of those two.










LED (Light Emitting Diodes): A diode is a gate that allows current to flow in only one direction. LEDs provide light.

Minecraft Mod: Our software is a modified version of Minecraft called PiperCraft™ for RPi that requires no license.

Piezo Buzzer: Basic audio resistive component. The higher voltage it gets, the higher the pitch it emits.

RPi (Raspberry Pi): A low cost microcomputer. Together with Raspbian, a special operating system, RPi is a full blown computer.

Switches: Basic electronic components that close circuits permanently.

 Breadboard	 Buttons	 GPIO Pins
 LED Lights	 Piezo Buzzer	 Battery
 Jumper Wires	 Raspberry Pi Computer	 Switches

How Piper Works

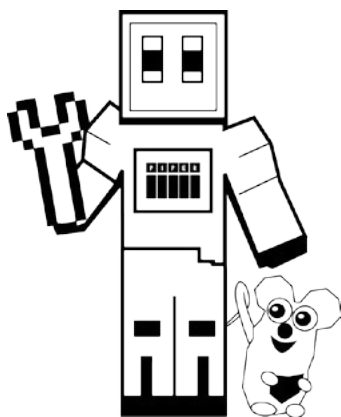
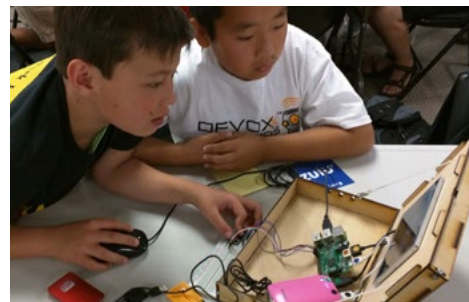


Piper Computer Kit

Piper is a fully functioning computer with WiFi, USB ports, memory, internet browser, an operating system and many pre-loaded apps like Scratch, Python and Code Club on the computer's desktop. Piper's Minecraft Mod is software on the MicroSD card and may be updated to the latest version at any time for free by clicking the "update" button on the main menu. Even though the Piper kit does not come with a keyboard, most keyboards will work with Piper via the USB ports.

Build. Play. Learn.

Students and teachers using Piper learn about electronics while playing our Minecraft Mod game. Once they complete the digital challenge, they must apply their learning by building a specific electronic gadget that will "power up" the game. It is amazing to see how quickly students understand complex engineering diagrams when presented in the PiperCraft world!



PiperCraft™ Minecraft Mod

PiperBot, our main character, is tasked to solve a world-ending catastrophe. PiperBot's trusty mouse, Pip, is always by PiperBot's side. Our experienced leader, Commander, communicates instructions from Piper base station. The story begins when a cheesteroid is detected approaching earth and PiperBot is reactivated to prevent a collision. The only problem is PiperBot has been inactive and all its functions need to be repaired. Students are tasked with building PiperBot a controller to navigate through the game. To do so, the student engages in PiperCraft, where each "crafting table" shows students how

to wire the real-world electronics that PiperBot needs.

PiperCraft is broken into two sections: story mode and creative mode. Within story mode, there are "worlds" that Piper has created, each supported with curriculum. Creative mode is an open design area for students to program within.

Desired Results

The big idea is to instill creative confidence with electronics. Students will learn engineering by hooking them with a fun low barrier experience and smoothly building and transitioning into a high ceiling creative experience. Specifically, students will know and be able to do the following:

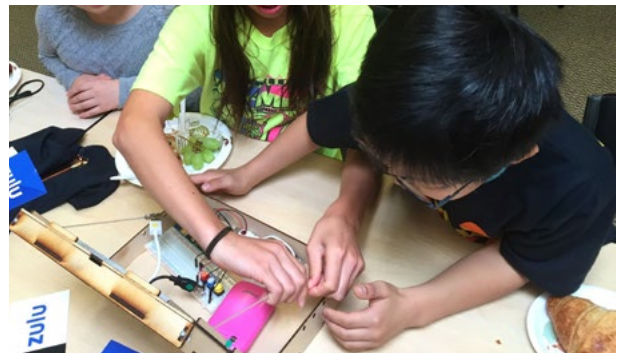
- Build electronics with Raspberry Pi, a breadboard, and jumper wires
- Complete circuits
- Construct inputs and outputs (buttons, switches, LED lights, and buzzers)
- Understand binary states and parallel circuits

Best Practices

We want you to be successful using Piper in the classroom. To do that, we suggest using these practices to help ensure maximum learning and engagement for your students.

Collaboration

While Piper works great playing solo, we encourage collaboration. Two or three students may all use one Piper kit collaboratively to play the digital game and build electronic gadgets. Additionally, in a classroom environment, we suggest to have students who are struggling to get help from students that are moving through the chapter quickly. This empowers the students to collaborate and solve problems without teacher guidance. We have found success stopping the class every 10 minutes to have a student talk about a challenge in Piper that they overcame.



Facilitation vs Direct Instruction

With Piper, instructions are presented in the game within the PiperCraft world. We recommend to facilitate the class versus giving the students the answer. We understand that it is hard to turn down students when they get stuck, but we firmly believe this is part of the learning process, and what ultimately helps build

creative confidence. When reading through our lessons, look for the “essential questions” to ask during a lesson. When students ask questions, try answering with leading questions!

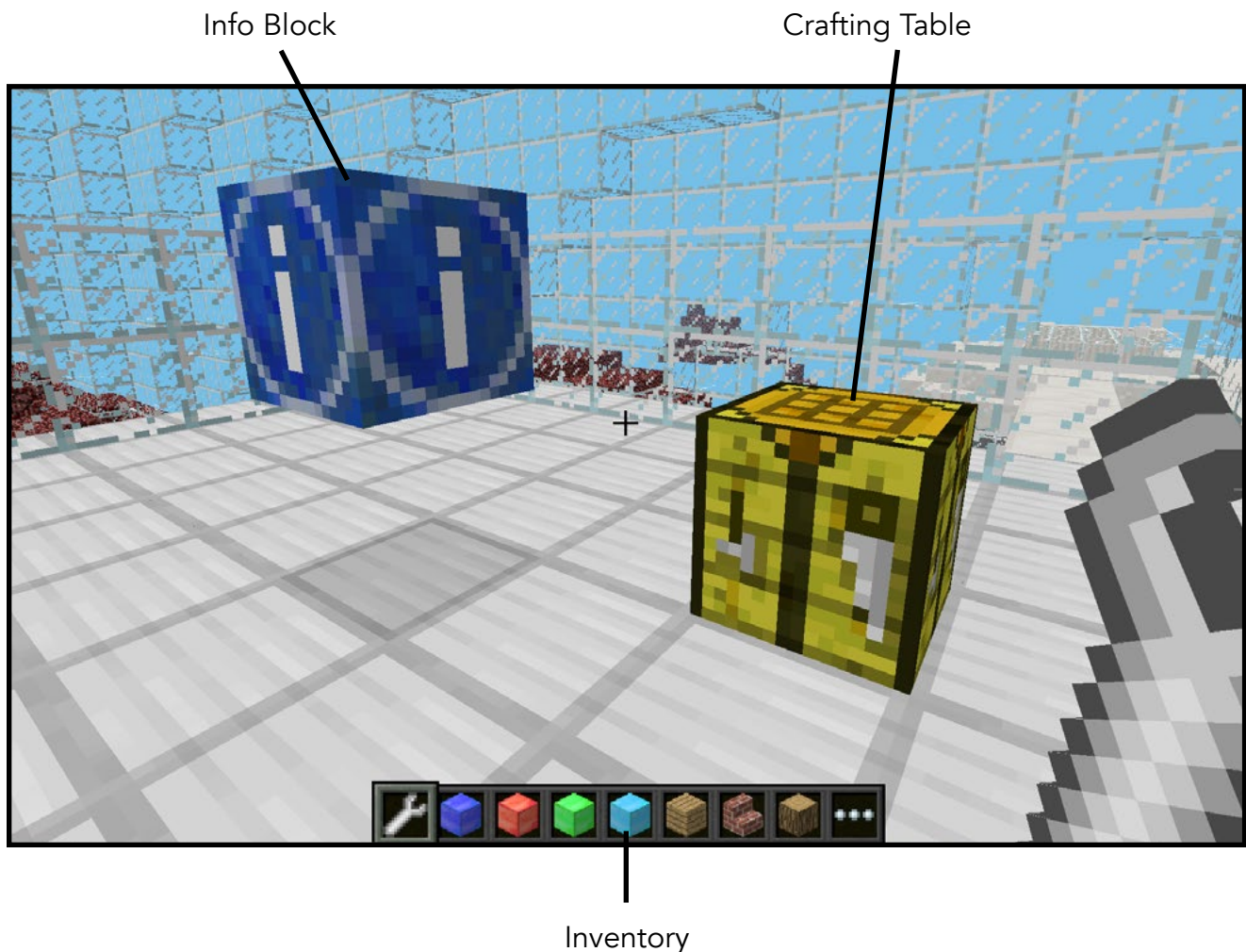
Set Expectations and Build Confidence

We have found success starting each session with a discussion and reinforcing expectations that Piper will be challenging, failure will happen (that is OK), and details are more important than speed, etc. Have students talk about some things they have made themselves to give them confidence that they are now ready to create more!



New to Minecraft? 5 Functions to Know

1. Left click the mouse to “break” blocks
2. Right click the mouse to “place” blocks
3. Use the mouse scroll to go through inventory (found at the bottom of the game screen)
4. Right click on the “crafting table” to get instructions on what to build
5. Click on the blue “info block” to get hints on the challenge





PiperCraft is full of fun, interactive worlds that marry gaming with computer engineering principles. Our staff has worked hard to make learning fun and engaging for your students. Listed below are all of the worlds (levels) that the Piper team has created, and that are used in this curriculum.

Mars

Introduction:

PiperBot is immobile because his interior wiring is not working. To allow PiperBot to move, the student must connect the red wires, first together, and then to the controller board. The green and blue buttons and wires will follow.

GPIO setup: Green:1,2; Red: 5,6; Blue: 9,10.



Concept:

The concept at work in this lesson is electrical currents and circuits. The student should understand that after connecting the red wires, there is an electric current flowing through both pins on the Pi that sends a message (INPUT) for PiperBot to move forward or strafe sideways. Only when the circuit is completed can the current flow properly. When the wires are placed into the breadboard, pushing the button, rather than touching the metal ends, allows the current to flow.

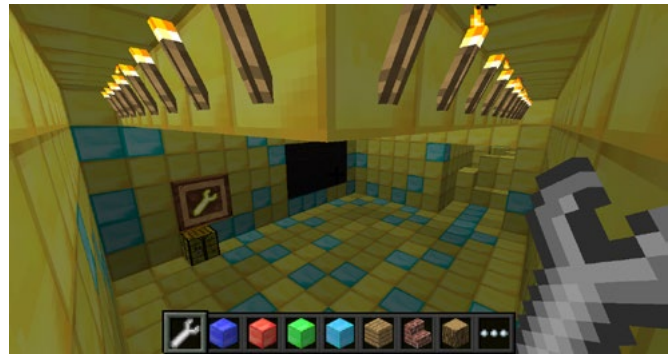
Troubleshooting:

A common point of failure is the specific placement of the wires on the GPIO pins on the Raspberry Pi. The necessary attention to detail will take some students by surprise but will set the precedent for the level of precision expected through the game. If there are two or more students working on a single Piper, moving PiperBot with the wires while directing him with the mouse is a good way to incorporate both players. Additionally, be sure that students are firmly placing the buttons on the breadboard. If they are loose or wobbly, the connection won't be good enough to complete the circuit!

Cheeseteroid

Introduction:

PiperBot is inside the Cheeseteroid maze. PiperBot needs a new jumping function to get through the maze. This will be accomplished by continuing with the wiring scheme presented earlier. The main difference is that the yellow wires will go into pins 13 and 14. Once able to jump, it's just a matter of getting through the maze.



Concept:

This the same as in Mars: electrical currents, circuits and buttons. This is a good time to introduce the additional concept of directional flow of circuits. Draw upon an analogy between the R-Pi and a battery to teach about how current flows from positive to negative. Ask students to observe the

animation to determine which are the positive and negative poles. Students should understand that one pin is positive and one is negative.

Troubleshooting:

In rare cases, a point of frustration is disorientation that can happen from trying to follow the trail of red blocks. A tip for kids is to hold the forward and jump buttons down and put pressure forward on the buttons to get good electrical contact from all components. This helps students move and jump continuously over the rugged terrain.



Treasure Hunt

Introduction:

PiperBot's task is to find the golden wrench. To do so, Piperbot will need to find a series of clues from info boxes which will eventually lead to a workbench. Once there, PiperBot will wire an LED to a small breadboard and Pi. The LED has been coded as a "treasure detector". When PiperBot gets closer to the X, marking the hidden wrench, the LED will flash increasingly faster.



Concept:

Building on the concepts of current and circuits practiced on the previous functions, the student should understand that after connecting the wires to the LED, there is an electric current flowing from the Raspberry-Pi (R-Pi), through the Light Emitting Diode (LED), which sends a visual message (OUTPUT) to the player.

Troubleshooting:

No issues anticipated. Note the LED output has been coded to work without care of polarity. The LED can be put in either way and it should work. Bent LEDs may be an issue.



Chain Reaction

Introduction:

Having found the golden wrench, PiperBot now has to find the next portal. PiperBot will need to blow its way through sand and castle walls to reach the next portal. Along the way, momentary switches (buttons) and switches are compared and contrasted.



Troubleshooting:

For young people, the biggest pain point is having the manual dexterity and hand strength to properly seat the buttons on the mini breadboards. This is a good opportunity to revise attention to precision when wiring components.



Concept:

This is mostly review, however instead of enabling motion, holding the switch down enables a different wrench function. This is the basic concept behind digital binary bit states (ON/OFF is equivalent to 1/0). Note: Unlike a button which has to be held down to hold a bit state, switches hold the state depending on the position of the switch.



Power Plant

Introduction:

PiperBot is now inside a new game world and it's a power plant! Turn the power ON to open gates, release water, pump water, raise elevators, and make your way to the next portal.



Concept:

This is review of buttons and switches. The concept of hydroelectric power generation can be touched upon. As water falls down, it releases kinetic energy which is transformed by the generator and stored in the battery. The energy is then released to activate gates and elevators.

Troubleshooting:

For young people, the biggest pain point is having the manual dexterity to properly seat the buttons on the mini breadboards. Bent components may start to be an issue. This is a good opportunity to check for attention to precision when wiring components.



Rainbow Bridge

Introduction:

To reach the next portal PiperBot has to cross a constantly changing rainbow bridge. One random color of the rainbow bridge disappears periodically. To be able to cross, PiperBot will have to enable its buzzer, which hints at which color will disappear next. The number of clicks indicates which color will disappear.



only lets current flow one way.

Troubleshooting:

Polarity is most likely the one issue that may come up. Point to the plus sign on the clicker, it needs line up with the red wire and the leftmost pin of the pin pair.

Concept:

This is a review of outputs. A new type of output is introduced with a clicker/buzzer. Another concept to touch is component polarity. The positive (red wire) current must go into the long leg of the clicker and exit through the short leg, through the blue wire and back to the R-Pi. You can draw an analogy to a one way gateway: the clicker



Funky Fungi

Introduction:

PiperBot is in a magical world of funky fungi and chasms searching for the next portal. This is review of buttons and switches, however they are combined now in a new wiring scheme. Two buttons allow you each to call a different structure from memory that will allow you to cross the chasm.

Eventually, you'll plug in a switch as well, which will let you 'pave' a platform as you move or jump.



Concept:

In a parallel setup, the buttons allow you to call different memory states and the buttons to share what is called a ground. A third state set by a switch allows you to call from memory continuously instead of on demand.

Troubleshooting:

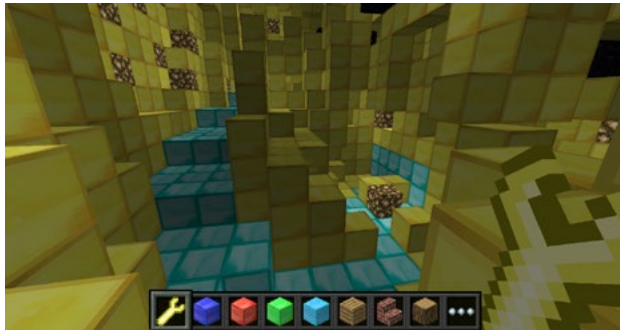
It can be easy to get lost in this level. After the second crafting bench keep going up to reach the red platform with the portal.



Cheeseteroid return

Introduction:

This is the longest and hardest level in the game. Once beaten, PiperBot saves earth and the student unlocks creative mode.

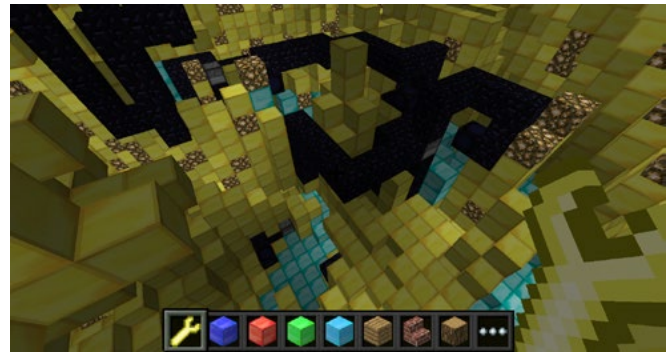


Concept:

A similar button and switch setup as the one in Funky Fungi is used in this level. Piperbot will have to alternate between constructing and destroying the cheese world to get through. Instead of memory states, the switch allows the player to switch between on demand construction/destruction and constant construction/destruction.

Troubleshooting:

Having reached the part where you have to see what both power blocks are doing, there is a crack in the geometry that is easy to miss. Otherwise keep building destroying and alternating from constant and on demand mode until you get through.



Learning Plan Overview

Duration:

Each day is equivalent to a 45 - 65 minute period for a total of 9 periods.

Preparation:

Have your students complete [this Google survey](#) before the unit is started, please!

Days 1-4:

LP01 - Parts, Purposes, Complexities of the Piper + Kit build and computer literacy.

Days 5-9:

LP02 - Buttons and Breadboard

LP03 - Basic I/O (LEDs, buttons and switches)

LP04 - Buttons, switches, and power generation. Polarity and audio output.

LP05 - Buttons and switches in parallel configurations.

LP06 - Take apart and unit reflection. Please have your students fill out this [post survey](#).

Expansion Options:

- Single or multi-player world creation through Piper API.
- GPIO and creative mode.

Standards Addressed In This Unit

K-12 Computer Science

- [K-12CS Concepts](#): Computing Systems - Devices, Hardware & Software
- [K-12CS Crosscutting Concepts](#): Invention/Innovation, Patterns/Abstraction, Systems
- [K-12CS Practices](#): Fostering an Inclusive and Diverse Computing Culture, Collaborating, Communicating About Computing, Computational Thinking (Recognizing and Defining Computational Problems, Developing and Using Abstractions, Creating Computational Artifacts, Testing and Refining)

NGSS (Next Generation Science Standards)

- [NGSS Science and Engineering practices](#): Developing and using (electronic) models, Asking questions
- [NGSS Crosscutting Concepts](#): Patterns, Cause and effect: Mechanism and explanation, Using mathematics and computational thinking, Systems and system models, Structure and function, Stability and change
- [NGSS ETS Disciplinary Core Ideas](#): Engineering, Technology and Applications of Science

ISTE 2016 - [Global Collaborator](#)

- Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.

Lesson 1 - Building the Kit



Established Goals

The goal of this lesson is for students to begin learning the literacy of the Piper hardware components in the kit. Instead of front loading a sheet of terms, students will experience the [Parts, Purposes, Complexities thinking routine](#) (described below) developed by [Agency by Design](#). The goal of the routine is for students to engage in close looking, inspire question making and natural curiosity, and have student observations and questions recorded on a graphic artifact (poster with labeled drawings of the parts). Students will also experiment with the components to learn how they work together.

Learning Objectives:

- Students will understand the purpose of every Piper component.

Suggested Sequence

- Days 1 - 2: Parts, Purposes, Complexities of the basic microcomputer components
- Days 3 - 4: Introduce case wooden parts and brass hardware, kit build, SD card

Preparation

- Suggested student to kit ratio is 2:1 up to 3:1
- Prepare enough medium or large format drawing paper (or other surfaces such as dry erase tables, poster and panels) and colored pencils or markers.
- Make sure each kit contains the following pieces: R-Pi, SD card*, mouse, display, cables (power, USB, HDMI), speaker, wires, breadboard, colored buttons, power pack*.
 - *Note: If you don't want students to go into the software yet, hold on to the SD card or hold on to the power pack.
- Hold on to the blueprint, laser cut pieces, and brass hardware.

Day 1

Introduction (5 minutes):

- **Announce new unit:** By the end of this unit, you'll know how to build a computer, its parts and how to create basic electronics while playing a Minecraft mod.
- **Hook:** show [video](#).
- **Hold:** Announce expectation that before we can play, we need to learn to build a computer.

Main Activity (80% of class time):

- Prompt students to address the prompts below through a labeled illustration of the system. Students should be aware they will be sharing their work with the class later.
 - What are its **parts**? What are its various pieces or components?

- What are its **purposes?** What are the purposes for each of these parts?
- What are its **complexities?** How is it complicated in its parts and purposes, the relationship between the two, or in other ways?
- During this time, the teacher should be roaming around the room, asking provocative questions such as:
 - What do you think that is?
 - Where do you think it connects to?
 - What would you call that?
 - See that label or number on it? Look it up on the internet and tell me what it is.
 - How does it work?

Closing Activity (15-20% of class time):

- **Group discussion:** Have one group volunteer to share their work with the class. Visibility for whole class will be key.
- Address misconceptions as students share.
- Collect work, allow time for students to clean up after themselves.

Days 2-3

Introduction (5-15 minutes):

- Distribute work.
- Have a different group share their work.

Main Activity (80% of class time):

- Have students put case and electronic components together.
- If you haven't already handed out SD cards, do so as students complete their kit.

Closing Activity (10-15% of class time):

- Have students record any new findings on their graphic artifact.
- Collect work.

Days 3-4

Introduction (2-5 minutes):

- Announce expectation: Kit should be built by end of day 4. Students should work together and use the blueprint as a building guide.

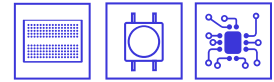
Main Activity (50-75% of class time):

- Students finish building kit and graphic artifact. Try not to micromanage here... students will make mistakes and find ways to solve the problem themselves.

Closing Activity (25% of class time):

- Teacher led discussion: [LP01 Component review slides](#). Students should take notes from the slides on their graphic artifacts.
- Clarify misconceptions and answer questions as needed.

Lesson 2 - Buttons & Breadboards



Established Goals

Having built the kit and learned the physical structure and functions/purposes of the all parts, students will now go through the fundamental concept of wiring a circuit and understanding how circuits on the breadboard work.

Learning Objectives:

- Practice breadboarding and wiring.
- Understand wire and pin positions for specific buttons.
- Understand that circuits flow from positive to negative, through a component.
- Understand the difference between closed and open circuits.

The Lesson (1-Day Plan)

Introduction (2-5 minutes):

- Introduce game and tell students that they will be learning how to make Piperbot move using wires, buttons and breadboards.

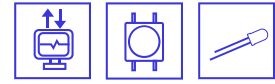
Main Activity (30-40 minutes):

- Allow students to go through **Mars** and **Cheeseteroid** worlds.
- During this time, the teacher should be roaming around the room, asking the **essential questions** of this lesson:
 - What happens when you press the button (both in the physical and electrical worlds and inside the computer)?
 - Why do we need two wires per component?
- **Note:** Students may have trouble getting the buttons to work. Remember that good contact is the key to good electric flow. Buttons and wires need to be properly aligned and seated. Students may need assistance getting the wiring just right.
- Students put kit away to avoid distractions during teacher led discussion.

Closing Activity (10-15 minutes):

- Teacher led discussion: [LP02 Slides - How do breadboards work?](#)
- Have students perform the [Buttons and wiring task](#) (formative assessment using Google Draw). [Here is the solution](#) to the wiring task.
- Quick reflection [exit slip](#) (optional). Use this to get feedback from students of their understanding.

Lesson 3 - Basic Inputs & Outputs



Established Goals

Having learned the basics of circuits, breadboards and the different electrical properties of different materials, students will understand the difference between an output and input by comparing the LED behavior to button and switch behavior.

Learning Objectives

- Understand there are many different kinds of inputs and outputs.
- Understand electricity can be controlled to behave like binary (most basic computer language) and either have a high or low voltage: high = 1 = ON, low = 0 = OFF
- Understand that electric flow is sensed by the computer hardware and programmed to have an effect in software.
- Understand that the computer is programmed to detect the state of the electricity going into the pin.
- Understand that the computer is programmed to send a high voltage to the pin when light is desired.

The Lesson (1-Day Plan)

Introduction (2-5 minutes):

- Tell students they will be working through 2 more levels of the game today, which will introduce using LED lights and switches.

Main Activity (30-40 minutes):

- Allow students to go through **Treasure hunt** and **Chain Reaction** worlds.
- During this time, the teacher should be roaming around the room, asking the **essential questions** of this lesson:
 - What happens when you press the button in the electrical and software worlds? What happens when the LED is lighting up?
 - Why do we need two wires per component?
 - What is the difference between a switch and a button?
- **Note:** Students may have trouble getting the buttons to work. Remember that good contact is the key to good electric flow. Buttons and wires need to be properly aligned and seated. Students may need assistance getting the wiring just right.
- Students put kit away to avoid distractions during teacher led discussion.

Closing Activity (10-15 minutes):

- Teacher led discussion: [slides](#)
 - Students participate in discussion, take notes on graphic artifact.
- Have students perform the [placeholder] (formative assessment) and solution [placeholder]
- Quick reflection [exit slip](#) (optional). Use this to get feedback from students of their understanding.

Lesson 4 - Polarity & Audio Outputs



Established Goals

Having understood the difference between an output and input, students will review using buttons and switches and learn the new concept of polarity.

Learning Objectives

- Understand buttons and switches are used in most complex electric and electronic devices, no matter the size.
- Understand switches are different to buttons in that they allow a sustained effect. Buttons are momentary.
- Understand components with polarity have to be placed a certain way in the circuit.

The Lesson (1-Day Plan)

Introduction (2-5 minutes):

- Tell students they will be playing through 2 more levels of the game today, exploring the many ways to use buttons and switches in Piper, and learning about polarity.

Main Activity (30-40 minutes):

- Allow students to go through the **Power Plant** and **Rainbow Bridge** worlds.
- During this time, the teacher should be roaming around the room, asking the **essential questions** of this lesson:
 - In Power Plant: Where have you seen buttons and switches used before?
 - In Rainbow Bridge: Look closely at the buzzer, what special markers are used on it to indicate polarity? What happens if you put in an LED instead of the buzzer?
- Students put kit away to avoid distractions during teacher led discussion.

Closing Activity (10-15 minutes):

- Teacher led discussion: [slides](#)
 - Use these slides to unpack uses of buttons and switches in the outside world and polarity and audio output.
- Quick reflection [exit slip](#) (optional). Use this to get feedback from students of their understanding.

Lesson 5 - Parallel Circuits



Established Goals

Having reviewed buttons and switches and learned about polarity, students will now look at parallel circuits with buttons and switches.

Learning Objectives

- Understand that parallel circuits, together with buttons and switches can be used to simplify the button setup. Many buttons can share a ground pin.
- Understand that by using switches in combo with buttons in parallel setups, effects can be stacked. Different memory states can be called or different functions can be activated

The Lesson (1-Day Plan)

Introduction (2-5 minutes):

- Address misconceptions from previous lesson and tell students they will be playing through 2 more levels of the game today, exploring the many ways to use buttons and switches in Piper, and learning about parallel circuits.

Main Activity (30-40 minutes):

- Allow students to go through the **Funky Fungi** and **Return to Cheesteroid** worlds.
- During this time, the teacher should be roaming around the room, asking the **essential questions** of this lesson:
 - In Funky Fungi: What is the benefit of the two buttons sharing a pin? How does adding the switch affect the buttons and behavior in the game?
 - In Return to Cheesteroid: How does adding the switch affect the buttons and behaviour in the game? How is this setup different than in Funky Fungi?
- Students put kit away to avoid distractions during teacher led discussion.

Closing Activity (10-15 minutes):

- Teacher led discussion: [slides](#)
 - Use these slides to unpack Parallel circuits. Allow time for students to ask questions.
- Quick reflection [exit slip](#) (optional). Use this to get feedback from students of their understanding.

Lesson 6 - Take Apart and Reflection



Established Goals

Throughout this unit students have explored and learned many basic concept of electronics. To tie the unit up, students learn to take apart their Piper kits, evaluate their effort and collaboration and reflect on their learning. These activities serve as a summative assessment.

The Lesson (1-Day Plan)

Introduction (2-5 minutes):

- Address misconceptions from previous lesson and announce that today they will be taking apart the kits and evaluating their work on the unit.

Main Activity (30-40 minutes):

- To understand the best practices to take apart the kit, take students through [these slides](#) or let them go through them.
- Have teams evaluate themselves for effort and participation using this [form](#) (make a copy and use with students).

Closing Activity (10-15 minutes):

- Have students complete end of unit reflection using this [form](#) (make a copy and use with students).
- Unless there is something else planned for your class, give students free time or have students share some of their reflection points.

Tell us what you think!

We hope this unit plan is useful to you, and we can't wait to hear your stories about your Piper classroom experience. Was the unit fun and engaging for your students? What did you notice about your students using Piper that surprised you? Did you end up making your own lesson plans to better fit your teaching methods? We want to hear it all!

Join the Piper community

There are a lot of Piper enthusiasts just like you that are dying to hear about your experiences using Piper in the classroom. Join the forum and share your story!

<http://community.playpiper.com/>

Get support

Something going wrong with one of your Pipers? Have any questions you need answered? We've got you covered.

<http://support.playpiper.com>

Send us a message!

Want to send us a personal message instead and tell us all the juicy details of your experience using Piper? Simply want to chat with the creators? Send us an email. We would love to hear from you.

educators@playpiper.com

