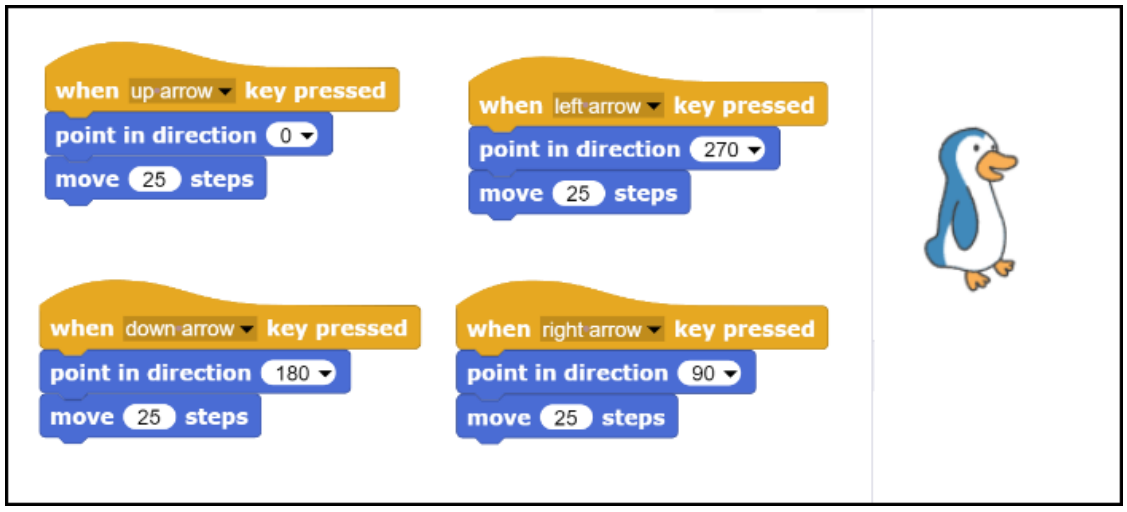
Setting up the Microcontroller

Most arcade cabinets contain at least one joystick and several buttons. Each button and each direction on the joystick connects to a switch that sends a signal to a microcontroller when activated. A microcontroller is a compact and programable microchip designed to perform a specific operation. In the case of an arcade controller, the microcontroller interprets button presses and joystick motions into specific commands that it then sends to the microprocessor running the arcade's program.

Almost all games made to be run on a standard computer rely on the user providing input using a keyboard. In order to create an arcade control bar that will work with a computer, it is necessary to use a microcontroller that can emulate key presses. For example, the following Snap*!* commands use the arrow keys on the computer keyboard to control the movement of the penguin on the screen.



Snap! Arrow Controls

One of the best microcontrollers for emulating key presses in the Raspberry Pico. The Pico is also relatively small, making it ideal for inclusion in compact projects like control mechanisms.

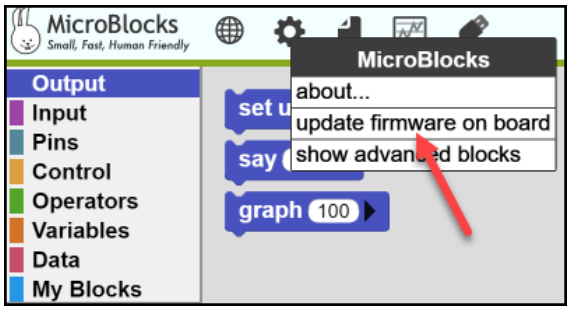
Raspberry Pico Microcontroller

Connecting to the Microcontroller

The Raspberry Pico can be programmed to emulate key presses on a computer keyboard using the block-based coding language MicroBlocks. Microblocks can be accessed at:

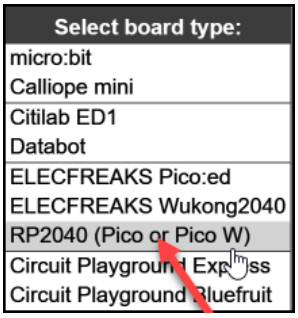
https://microblocks.fun/run-pilot/microblocks.html

To use MicroBlocks, you must first install firmware on the microcontroller. Firmware is a software that provides basic machine instructions that allow the hardware to function and communicate with other software running devices. To install the microcontroller firmware, connect your microcontroller to your computer with a micro-USB cable. From the gear-shaped settings menu, select *Update Firmware on Board*.



The Settings Menu

Then, select *RP204 (Pico or Pico W)* from the drop-down menu and follow the on-screen instructions to complete the firmware installation process.



Board Selection

Click on the icon of a USB cable in MicroBlocks and select *Connect*.

Connecting the Board

A dialog box should appear that says “Microblocks.fun wants to connect to a serial port”. Select the microcontroller and click the *Connect* button in the dialog box. A green circle should appear behind the USB icon to show that the microcontroller is connected.

Board Connected Icon

Programming the Microcontroller

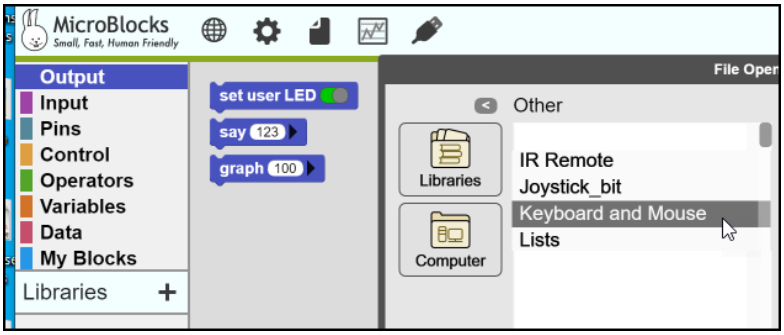
Arcade controls such as buttons and joysticks often make use of microswitches. A microswitch is an electric switch that can be operated rapidly by a small motion. In the image below, pressing the red button closes the contact between the two metal tabs, allowing electrical current to flow through them.

In an arcade button such as the one shown below, pressing the button depresses the microswitch.

Some arcade joysticks are constructed by combining four microswitches, one for each direction (up, down, left, and right) of the joystick.

In all of these examples, each microswitch must be connected to the microcontroller, with one pin on the microcontroller assigned to each individual microswitch.

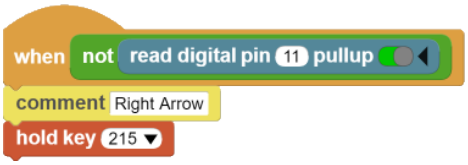
Since this arcade controller will emulate key presses on a keyboard, you will need to load the *Keyboard and Mouse* library in MicroBlocks. To do this, click on the + symbol next to *Libraries* in the menu on the left side of the screen. On the window that opens, select *Other* and then select *Keyboard and Mouse*. Once you load the library, the new category of blocks will appear in the menu on the left side of the screen.



In keyboard emulation, when the joystick moves up, down, left, or right, MicroBlocks sends the appropriate keypress to the program running the arcade game. In the game, it appears as though one of the arrow keys was pressed on the computer keyboard.

For most of the keys on the computer keyboard, pressing and holding a key can be emulated in the Hold Key code block by entering the letter associated with the key. For example, in the example below, when the designated button on the arcade control is pressed, the letter “S” is emulated:

However, in the case of a key such as an arrow key, a numeric code is used instead:



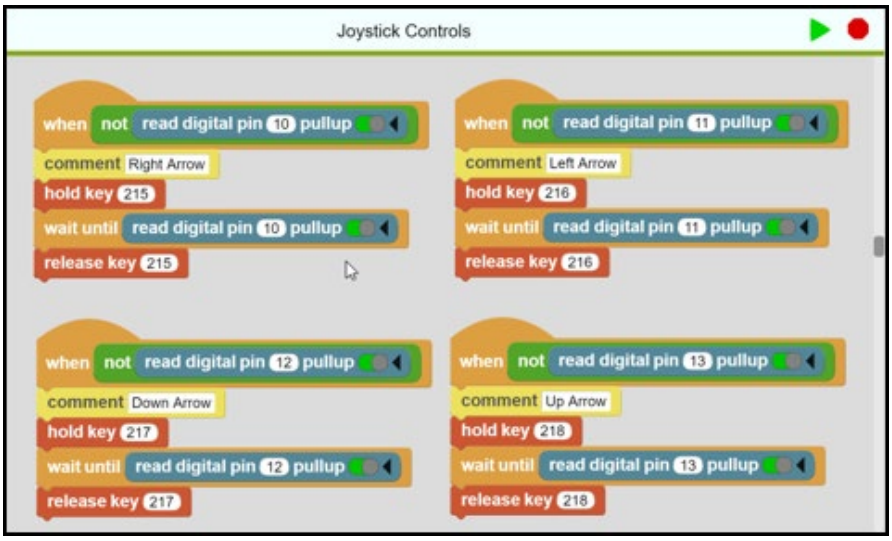
The numeric codes for special keys for microcontrollers in the Arduino family of microcontrollers can be found at the following address:

https://www.arduino.cc/reference/en/language/functions/usb/keyboard/keyboardmodifiers/

For example, the table lists the following numeric values that would be entered into the Hold Key code block to emulate the arrow keys on the computer keyboard. Additionally, the decimal code for the “Space Bar” key is 32.

KEY	DECIMAL VALUE
KEY_UP_ARROW	218
KEY_DOWN_ARROW	217
KEY_LEFT_ARROW	216
KEY_RIGHT_ARROW	215

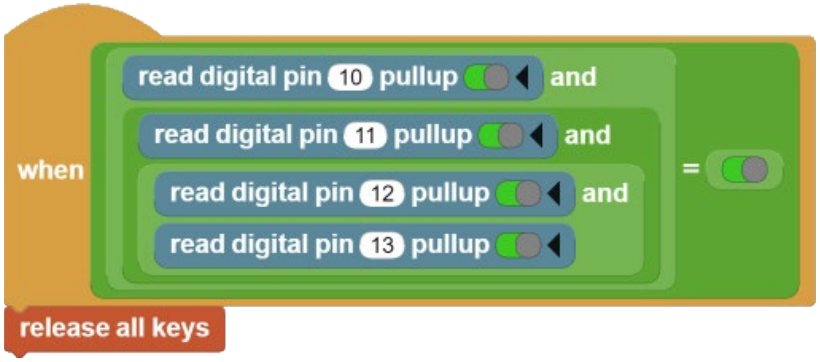
Assuming that the four microswitches of the joystick arcade controls have been connectioned to Digital Pins 10, 11, 12, and 13 of the microcontroller, the MicroBlocks code required to detect movement of the joystick would look like this:



The **Hold Key** code blocks in the illustration above emulate the process of holding down key on the omputer keyboard when movement of the joystick is detected. For example, when the joystick is moved to the right, the microswitch connected to Digital Pin 10 on the microcontroller closes. When that closure is detected, the digital code for a right arrow keypress (the number 215 in this example) is sent to the operating system of the computer. The Snap*!* game would receive the *up arrow* keypress, which would appear that it was being sent from a keypress on the computer keyboard, and take the appropriate action.

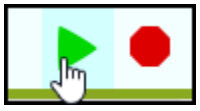
The **Hold Key** code block emulates the process of holding a key on the computer keyboard until another code block releases the key (i.e., the equivalent of removing a finger from a key that has been pressed so that it is released). For example, after the microswitch connected to a Digital Pin is activated, the code waits until the microswitch connected to that pin is deactivated. When the microswitch is deactivated, the **Release Key** block releases the key corresponding to that Digital Pin.

To ensure that keys do not get stuck, the following MicroBlocks code detects when none of the microswitches on the joystick are depressed. This occurs when the joystick is in the centered position. When this condition occurs, the Release All Keys code block emulates the condition in which a previously depressed key has been released.



To add arcade buttons, this code would be replicated and extended by assigning key presses (e.g., A, S, D, Space Bar) to additional digital pins that will be connected to those buttons.

Once you have finished writing the code, press the green *Play* button in the top right corner of the screen to load it onto the microcontroller.



You can now disconnect the microcontroller from your computer and plug it into your arcade.