Make Your Own Arduboy Game: Part 4 - Questions & Button Input

crait (Holmes) #1 June 21, 2019, 6:46am

This is Part 4 in a series on learning how to program your own Arduboy game. If you have skipped the previous parts, please read over **Part 1**, **Part 2**, and **Part 3**!

An Important Question

I used to teach programming to little kids and would always ask them a very important question. Think about this for a second: **What is a computer's favorite color?**

I've heard so many different answers to this. One kid told me, " *Green! Because hackers use green text on their screen!* " 😂 That was wrong. 🙂 Of course, the correct answer is that computers don't have a favorite color! That's because computers don't have opinions. You can't just *ask* a computer what it's favorite color is.

The only questions that you can ask a computer is to evaluate numbers, check the value of a variable or to count something.

Stopping Robots From Washing Their Hair

Remember in the previous part of this tutorial, I told you about the back of the shampoo bottles? The instructions say:

```
1. Lather
2. Rinse
3. Repeat
```

The problem with this is that if you gave these instructions to a robot, it will repeat these steps forever. We need to implement some way to stop the robot from doing that. In computer programming, we can use **conditional statements** . Conditional statements are basically questions that we ask the computer and if the answer is true, then follow different instructions

```
1. Lather
2. Rinse
3. If you need to wash your hair again, then Repeat
```

The first kind of conditional statements you'll come into contact with are `if` statements. Like the example above, you can basically ask the computer `if something is true, then do something` . A robot following the above instructions won't wash their hair forever since it will eventually be washed.

Buttons

We can also use `if` statements to test buttons, too! That's how video games know *if* you're pushing buttons. Moving a character may look like this:

```
1. If the Right Button is pressed, move character right
2. If the Left Button is pressed, move character left
3. Print character picture to screen
4. Repeat
```

Format

The format for `if` statements in C/C++ is:

```
if (something is true) {
    DoSomething();
}
```

Only `if` the `something is true` , then `DoSomething();` command will run.

To check `if` the A Button is pressed, we need to use the `arduboy.pressed(A_BUTTON)` function. Putting it in the correct format, it will look like this:

```
if( arduboy.pressed(A_BUTTON) == true ) {
    DoSomething();
}
```

Notice that we use a double equal sign == instead of just a single equal sign = . You'll mostly see this inside of conditional statements. Don't let it scare you! A single equal sign = means that you are assigning a value. A double equal sign == is checking the value. I'll talk more about this later!

The above code can also be written as:

```
if( arduboy.pressed(A_BUTTON)) {
    DoSomething();
}
```

Why? The `arduboy.pressed()` function returns a true or a false value and these can be evaluated directly. In my opinion, the inclusion of the `== true` in the first example actually makes it harder to read.

Above, we checked to see if the A Button was being pressed, but we can also check if the A Button is not being pressed!

```
if( arduboy.pressed(A_BUTTON) == false ) {
    DoSomething();
}
```

The `Arduboy2` library includes a function called `notPressed()` which allows us to simplify the code above to:

```
if( arduboy.notPressed(A_BUTTON)) {
    DoSomething();
}
```

Cool, right?!

Modifying The Code

Let's go grab that code from Part 3 and modify it:

```
//initialize our counter variable
int counter;
//The setup() function runs once when you turn your Arduboy on
void setup() {
  //Start the Arduboy properly and display the Arduboy logo

  arduboy.begin();
  //Get rid of the Arduboy logo and clear the screen
  arduboy.clear();
  //Assign our counter variable to be equal to 0
  counter = 0;
}
//The loop() function repeats forever after setup() is done
void loop() {
  //Clear whatever is printed on the screen
  arduboy.clear();
  //Move the cursor back to the top-left of the screen
  arduboy.setCursor(0, 0);
  //Increase counter's value by 1
  counter = counter + 1;
  //Print out the value of counter
  arduboy.print(counter);
  //Refresh the screen to show whatever's printed to it
  arduboy.display();
}
```

Remember that `counter` is our variable that increases forever in our previous code. What we're going to do is change this code to increase or decrease `counter` depending on if you're holding the Up Button or Down Button.

We need to find the line that increases `counter` and remove it. After we clear the screen with `arduboy.clear();` , let's add the following code:

```
if( arduboy.pressed(UP_BUTTON) ) {
    //Increase counter
}
if( arduboy.pressed(DOWN_BUTTON) ) {
    //Decrease counter
}
```

Our code to increase `counter` was `counter = counter + 1;` . Using that code, we can decrease it by switching the plus sign for minus sign. To decrease `counter` , our code will be `counter = counter - 1;` . Try to plug those in and when you're done, check to see if it is the same as the completed code, below.

Testing Numbers & Variables

I was going to end the tutorial now, but let's extend it a little more and tell you about testing numbers!

I told you before that inside of an `if` statement, you can check to see if something is true or false. But, you can also check the values of variables or numbers.

```
if( counter == 36 ) {
  arduboy.setCursor(30, 30);
  arduboy.print("Yay!");
}
```

What do you think would happen if you ran the above code? `if` `counter` is equal to 36, then the Arduboy will set the cursor and print out "Yay!" to the screen. Notice that you can put multiple lines of code inside of the `if` statements.

Add that right after the other `if` statements that check for the Up and Down Buttons.

The Completed Code

😃

```
//Jonathan Holmes (crait)
//October 21st, 2016
//Button Test
//Include the Arduboy Library
#include <Arduboy2.h>
//Initialize the arduboy object
Arduboy2 arduboy;
//Initialize our counter variable
int counter;
//The setup() function runs once when you turn your Arduboy on
void setup() {
  //Start the Arduboy properly and display the Arduboy logo
  arduboy.begin();
  //Get rid of the Arduboy logo and clear the screen
  arduboy.clear();
  //Assign our counter variable to be equal to 0
  counter = 0;
}
//The loop() function repeats forever after setup() is done
void loop() {
  //Clear whatever is printed on the screen
  arduboy.clear();
  //Check if the UP_BUTTON is being pressed
  if( arduboy.pressed(UP_BUTTON) ) {
```

Running The Code

Run the code and see what happens when you push the Up and Down Button!