

ARDUINO WORKSHOP

Version 2
BSc Computing for Games
Arduino

Alcwyn Parker

Introduction

The goals of this workshop is to obtain a good working knowledge of the Arduino platform. This will enable you to break away from traditional forms of hardware used to interact with your games and instead create more playful and tangible experiences for your users to enjoy.

Arduino is not just a micro-controller. It is an entire open source community and ecosystem of hardware and software solutions that are designed from the ground up to be easy to learn. There are a variety of different Arduino boards and shields that will be useful no matter where your ideas take you, from wearables to motion tracking, smart rooms to retro interface. The Arduino community is vibrant, active and full of resources to get you started rapid prototyping with the Arduino board. In this workshop you are encouraged to engage in self directed learning using the resources hosted on the official Arduino site to guide you.

'We find you need to make a game wrong at least two or three times before you find the right path. ... We took a lot of opportunity to design and explore, knowing that a lot of it would be thrown away.'

— Ken Wong, lead designer
Monument Valley

Objectives

- (A) **Identify** the parts of the Arduino and their purpose
- (B) **Write** sketches and upload them to the Arduino board
- (C) **Apply** the basics to create more complex relationships between sensors and actuators

Worksheet Setup

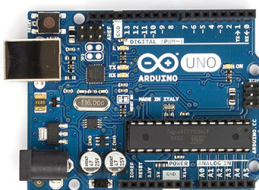
Due to the physical nature of this worksheet there is no need to fork a repo. Everything you need is the Arduino kit supplied. Using the kit supplied and the Arduino IDE, create a hardware/software solution that fulfils each of the activities listed below.

Worksheet Tasks

Begin by completing the first two sections of tutorials on the Arduino.cc website here: <https://www.arduino.cc/en/Tutorial/BuiltInExamples>. You can find the links to each relevant tutorial below.

Before you begin, take a few minutes to familiarise yourself with the Arduino IDE User Interface (UI). Link - <https://www.arduino.cc/en/Guide/Environment>

- Bare minimum sketch - <https://www.arduino.cc/en/Tutorial/BareMinimum>
- Hello World - <https://www.arduino.cc/en/Tutorial/Blink>
- Control an LED using a switch <https://www.arduino.cc/en/Tutorial/Button>
- Pulse width modulation - <https://www.arduino.cc/en/Tutorial/Fade>



Arduino Uno

- Analog to serial - <https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage>
- Analog to LED brightness - <https://www.arduino.cc/en/Tutorial/AnalogInOutSerial>
- Timing - <https://www.arduino.cc/en/Tutorial/BlinkWithoutDelay>
- Debounce - <https://www.arduino.cc/en/Tutorial/Debounce>

Now that you have completed the official tutorials lets use this new knowledge to attempt some challenges.

- **Challenge 1:** Add a potentiometer and LED circuit to the Arduino. Write an Arduino sketch that turns the LED on when the value read from the potentiometer is above 512.
- **Challenge 2:** Now, create a traffic light system by wiring up two more LEDs. These could be the traditional red, amber and green or any colour of your choosing. Write an Arduino sketch that lights a specific LED depending on the value read from the potentiometer. The suggested ranges are:
 - green on when: value < 400
 - amber on when: value > 400 && value < 800
 - red on when: value > 800
- **Challenge 3:** Switch out the potentiometer for a light dependant resistor (LDR). This way the LEDs will change as the sensor receives more or less light.

Additional Guidance

The goal of this worksheet is to get you up to speed with the Arduino platform, most of the work carried out here will be fairly rudimentary and you will be expected to progress quickly and apply these tasks to your hardware interface design.

Additional Resources

- Arduino <http://arduino.cc/>
- Adafruit tutorial <https://learn.adafruit.com/category/learn-arduino>
- Arduino video series https://www.youtube.com/watch?v=fCxzA9_kg6s