# Quick Draw Arduino Game
Don't get faked out, beat your opponent to become the fastest gunslinger!
Josiah Medina & Gavin Hageman

Displaying the Game

To display the main game visuals, we used an Adafruit TFT LCD 1.8" Screen. Connected to an Arduino Uno and soldered with extra connections for external gameplay mechanics. The images used are 24-bit Bitmap files, 128x160 and displayed in portrait orientation.
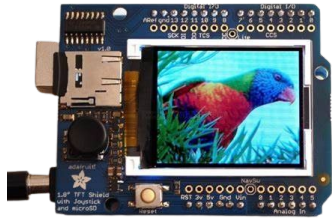
Scrapped Components

The game was originally going to feature usage of a speaker and 2 RGB LEDs controlled by player input for aesthetics. Simply put, this cannot happen because of the need for extra power through the Arduino (any Arduino attempting to initialize an SD card will use a lot of power into that task, resulting in failure to initialize without power focused on the SD Card constantly.) and the need for a program that is no longer updated.

The speaker, in retrospect, uses a program to encode audio (typically .Wav) files. This program is no longer supported and cannot be run because of missing files from the original developers, high-low tech. The example from high-low tech using only a speaker, ground, and any PCM pin does work however, but simply put, the encoding of audio files, if possible, would use a lot of bytes because of the sheer size of text needed to produce audio.
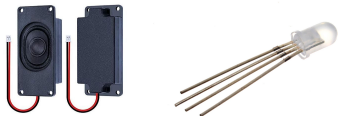
Original Design Aspects

When it came to designing the project, it first was thought to be a larger scale-tabletop game. When time came, we decided to model a smaller build because of the smaller scale of the TFT LCD. We decided to go for a look inspired by the Nintendo 3DS, a handheld, small, and portable device. You just need to connect it to power and you can play.

The simplest form of the design is a power supply into the Arduino, and the Arduino powers the TFT LCD and the buttons to play.
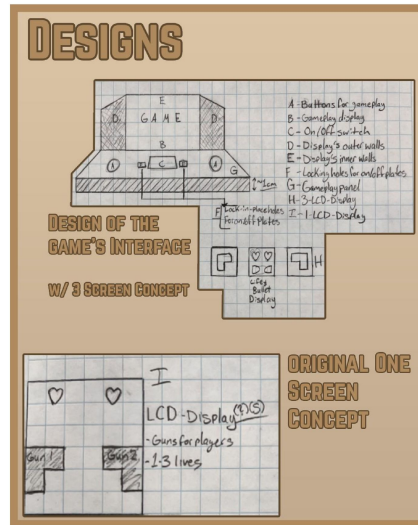
Drawbacks / Challenges Faced

Starting from very early of April when the code for the project was initially built by Josiah, the code had a constant error that had to do with the SD card listing off a file, opening it, then failing to open said file when attempting to display the image. About 4 re-builds of the code later, the project has not had this error anymore, and the initial problem was found to be a minor conflict in the rendering method.

On the topic of initial knowledge, wiring was initially correct, but the code was a new issue since it was unknown to us how to program something like an Adafruit TFT LCD. A larger part of the project's start was making workarounds for blocked resources we could not normally view, such as project resources being blocked on certain computers, Arduino's cloud editor not working on certain computers, and the serial monitor not being able to connect. A few cable switches and a different software, the project started working as needed.

```
void drawBitmap(const char *filename, int16_t x, uint16_t y) {
  Serial.print("Attempting to open file: ");
  Serial.println(filename);

  uint8_t sdbuffer[3 * BUFFPIXEL];
  uint8_t buffidx = 0;

  File bmpFile = SD.open(filename);
  int bmpWidth, bmpHeight;
  uint8_t bmpDepth;
  uint32_t bmpImageoffset;
  uint32_t rowSize;
  boolean goodBmp = false;
  boolean flip = true;
  int w, h, row, col;
  uint8_t r, g, b;
  uint32_t pos = 0, startTime = millis();

  if ((x >= tft.width()) || (y >= tft.height())) return;

  if (!bmpFile) {
    Serial.print("File not found: ");
    Serial.println(filename);
    return;
  }

  Serial.println();
```

Overall Display and Simple Wiring