Learn Arduboy

INTRODUCTION TO PROGRAMMING BY CREATING SIMPLE GAMES

Noke Codes | Learn To Program, Roanoke! | Part Two

Eight Ball Magic

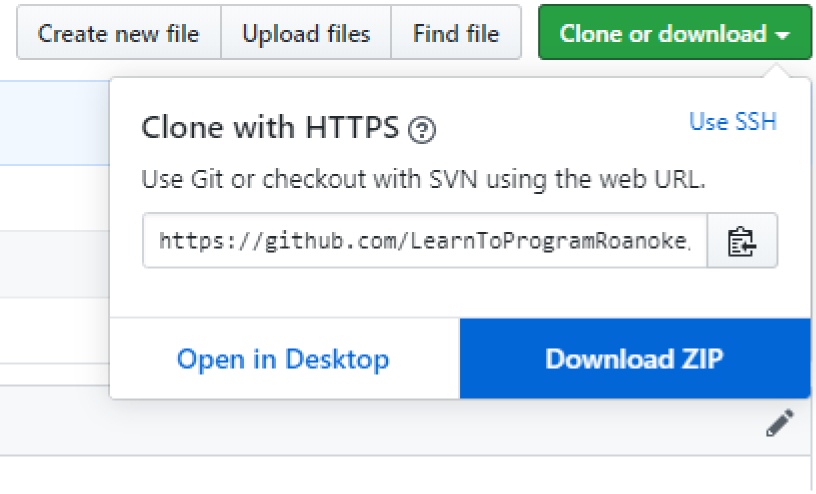
As the next step, this lesson will take a close look at a text game. Based on the classic Magic Eight Ball, this simple program demonstrates using an array to store the responses to display on the screen, code different functions as part of the game logic, capture button presses, and incorporate bitmaps that are displayed on the screen.

Concepts

- Arrays
- initRandomSeed()
- random()
- drawSlowXYBitmap()
- ArduboyTones
- pollButtons()
- justPressed()

This sketch, and any others in this series, will be using the Arduboy2 library. Comprehensive documentation for this library, in both HTML and Acrobat PDF, can be found online.[1]

To begin, the sketch and related files can be downloaded from a git repository hosted on GitHub.[2] There are many free, Open Source games available for the Arduboy that can be downloaded in a similar fashion and imported into the Arduino IDE.[3] Once you navigate to the GitHub page for a specific repository, use the "Clone or Download" button to get the ZIP file.

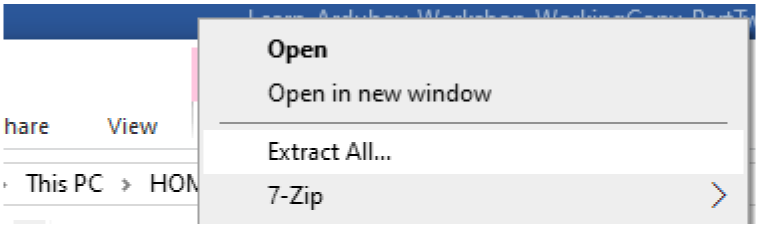


Use Git or checkout with SVN using the web URL.

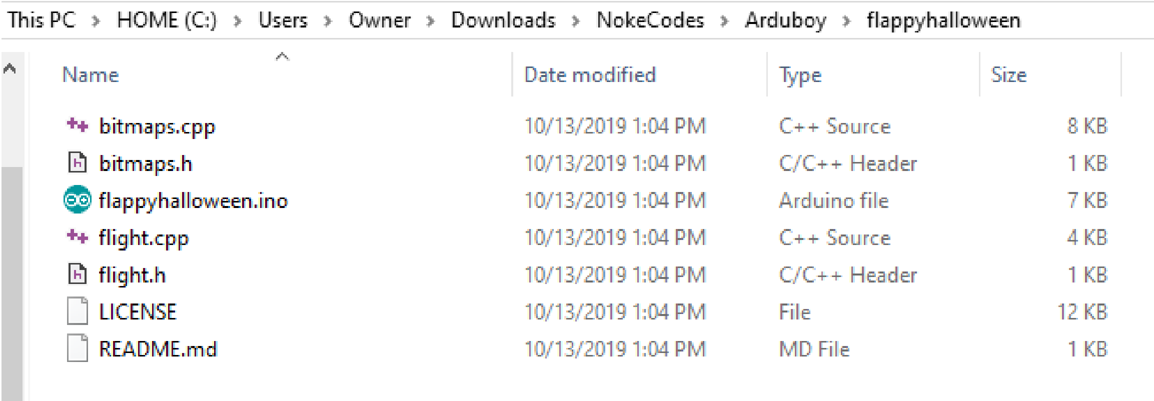
https://github.com/LearnToProgramRoanoke.

[1] https://mlxxxp.github.io/documents/Arduino/libraries/Arduboy2/Doxygen/html/
[2] https://github.com/LearnToProgramRoanoke/Learn-Arduboy/
[3] https://community.arduboy.com/c/games

Go to the download folder where the ZIP file was saved and right-click on the file. Select "Extract All …"



The un-zipped folder will contain the Arduino sketch and related files. The folder name will have -master appended to the end. Simply right-click to rename the folder without the -master. Then copy and paste the folder inside the default sketch folder for the Arduino IDE. **Note:** the folder and sketch file name need to match for the Arduino IDE to properly recognize it. Here is an example:

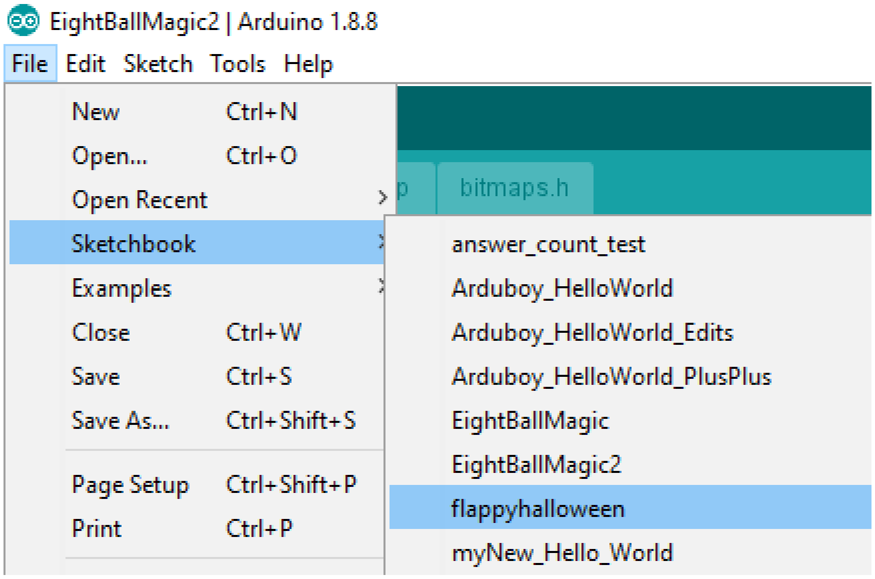


This PC › HOME (C:) › Users › Owner › Downloads › NokeCodes › Arduboy › flappyhalloween

Name	Date modified	Type	Size
bitmaps.cpp	10/13/2019 1:04 PM	C++ Source	8 KB
bitmaps.h	10/13/2019 1:04 PM	C/C++ Header	1 KB
flappyhalloween.ino	10/13/2019 1:04 PM	Arduino file	7 KB
flight.cpp	10/13/2019 1:04 PM	C++ Source	4 KB
flight.h	10/13/2019 1:04 PM	C/C++ Header	1 KB
LICENSE	10/13/2019 1:04 PM	File	12 KB
README.md	10/13/2019 1:04 PM	MD File	1 KB

Notice the folder name (flappyhalloween) and the sketch file name (flappyhalloween.ino) match. Once the folder has been pasted into the default sketch folder, you may need to close and re-open the Arduino IDE for it to be recognized.

The new sketch will now be on the menu of available sketches.



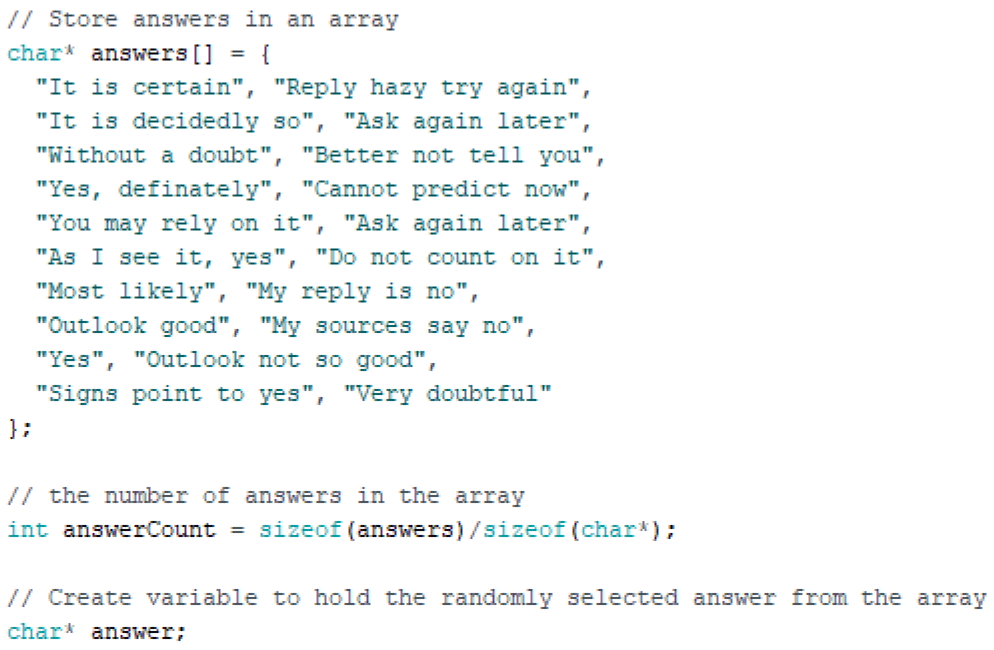
How the Eight Ball Magic Game Works

Open the EightBallMagic2 sketch in the Arduino IDE. We will examine how the program works section by section. First, the required libraries and a reference to the bitmap images file, needed for the graphics displayed in the game, are included. Both Classes, Arduboy2 and ArduboyTones are instantiated by creating an associated object (myArduboy and sound):

```
// Include the Arduboy2 and ArduboyTones libraries
#include <Arduboy2.h>
#include <ArduboyTones.h>
#include "bitmaps.h"

Arduboy2 myArduboy;
ArduboyTones sound(myArduboy.audio.enabled);
```

Before we get to the required *setup()* and *loop()* functions, several items need to be coded that will provide the needed parts of the game play.

The next section of the sketch deals with the Magic Eight Ball responses and the variables needed to store the response. The collection of strings (a group of characters that form a response) are stored in an array named “answers” and the data type is *char*.

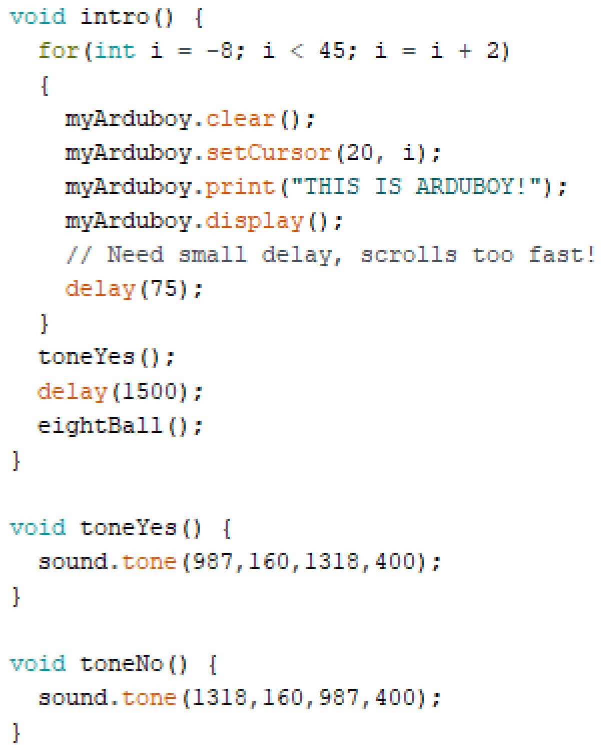


```
// Store answers in an array
char* answers[] = {
  "It is certain", "Reply hazy try again",
  "It is decidedly so", "Ask again later",
  "Without a doubt", "Better not tell you",
  "Yes, definately", "Cannot predict now",
  "You may rely on it", "Ask again later",
  "As I see it, yes", "Do not count on it",
  "Most likely", "My reply is no",
  "Outlook good", "My sources say no",
  "Yes", "Outlook not so good",
  "Signs point to yes", "Very doubtful"
};

// the number of answers in the array
int answerCount = sizeof(answers)/sizeof(char*);

// Create variable to hold the randomly selected answer from the array
char* answer;
```

The asterisk is a “dereference operator” that creates a pointer to the value inside the array rather than the space in memory where the value is stored. More information about how this works can be found online.[4] You can edit or add more responses to the array if you wish (keep it nice though!). A random response will need to be selected from the array, so the number of responses needs to be calculated and stored in an integer variable *answerCount*. Once a random response is selected, it will be stored in a char variable named *answer*.

[4] http://www.cplusplus.com/doc/tutorial/pointers/

Next in the sketch is the opening sequence *intro()*, which is very similar to the "Hello World" sketch we worked on earlier, and two functions to create the sounds used in the game. More details about the ArduboyTones library can be found in the related GitHub repository.[5]

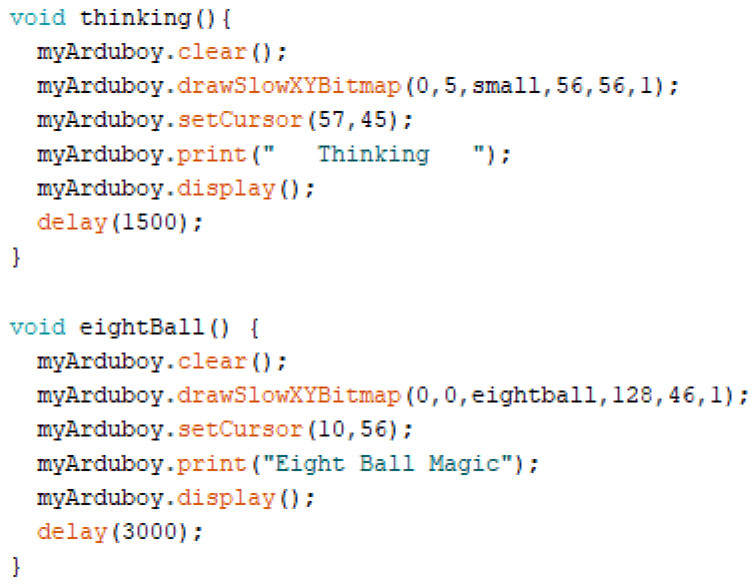


```
void intro() {
  for(int i = -8; i < 45; i = i + 2)
  {
    myArduboy.clear();
    myArduboy.setCursor(20, i);
    myArduboy.print("THIS IS ARDUBOY!");
    myArduboy.display();
    // Need small delay, scrolls too fast!
    delay(75);
  }
  toneYes();
  delay(1500);
  eightBall();
}

void toneYes() {
  sound.tone(987,160,1318,400);
}

void toneNo() {
  sound.tone(1318,160,987,400);
}
```

There are two functions that draw a bitmap image on the screen, *eightball()* and *thinking()* that both use a function built into the Arduboy2 library called drawSlowXYBitmap(). Details of the parameters passed to the function are in the reference document.[6]

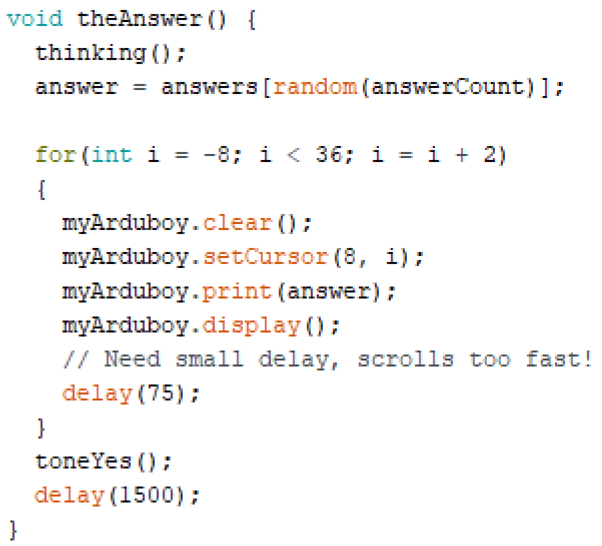
[5] https://github.com/MLXXXp/ArduboyTones

[6] https://mlxxxp.github.io/documents/Arduino/libraries/Arduboy2/Doxygen/html/

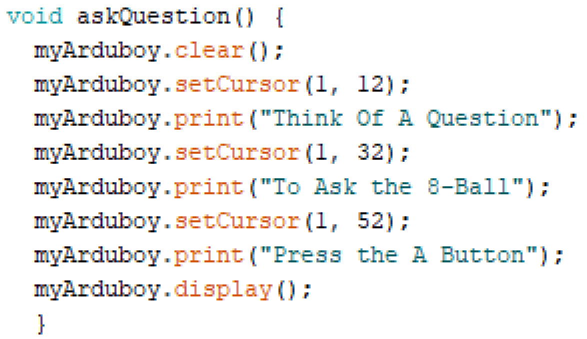


```
void thinking(){
  myArduboy.clear();
  myArduboy.drawSlowXYBitmap(0,5,small,56,56,1);
  myArduboy.setCursor(57,45);
  myArduboy.print("   Thinking   ");
  myArduboy.display();
  delay(1500);
}

void eightBall() {
  myArduboy.clear();
  myArduboy.drawSlowXYBitmap(0,0,eightball,128,46,1);
  myArduboy.setCursor(10,56);
  myArduboy.print("Eight Ball Magic");
  myArduboy.display();
  delay(3000);
}
```

Next are the functions that select the random answer from the array and another creates the screen with instructions on how to ask a question.

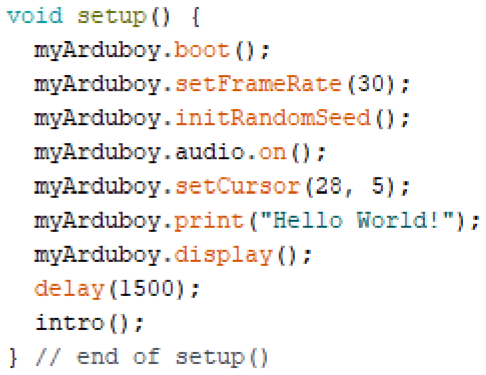


```
void theAnswer() {
  thinking();
  answer = answers[random(answerCount)];

  for(int i = -8; i < 36; i = i + 2)
  {
    myArduboy.clear();
    myArduboy.setCursor(8, i);
    myArduboy.print(answer);
    myArduboy.display();
    // Need small delay, scrolls too fast!
    delay(75);
  }
  toneYes();
  delay(1500);
}
```



```
void askQuestion() {
  myArduboy.clear();
  myArduboy.setCursor(1, 12);
  myArduboy.print("Think Of A Question");
  myArduboy.setCursor(1, 32);
  myArduboy.print("To Ask the 8-Ball");
  myArduboy.setCursor(1, 52);
  myArduboy.print("Press the A Button");
  myArduboy.display();
  }
```

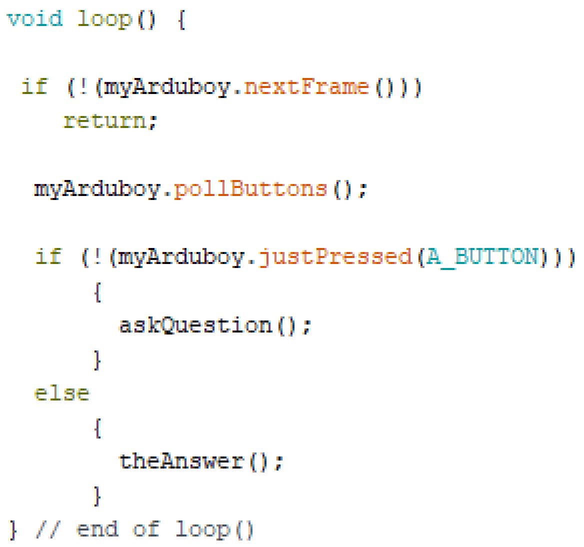
Now we're ready for the *setup()* function, which initiates the myArduboy object and other required steps we have seen before.



```
void setup() {
  myArduboy.boot();
  myArduboy.setFrameRate(30);
  myArduboy.initRandomSeed();
  myArduboy.audio.on();
  myArduboy.setCursor(28, 5);
  myArduboy.print("Hello World!");
  myArduboy.display();
  delay(1500);
  intro();
} // end of setup()
```

What is new is the *initRandomSeed()* function. This helps keep the random selection from becoming predictable.[7]

Finally, the *loop()* function with the *pollButtons()* and an *if/else* statement that calls either *theAnswer()* or *askQuestion()* functions.



```
void loop() {

 if (!(myArduboy.nextFrame()))
    return;

  myArduboy.pollButtons();

  if (!(myArduboy.justPressed(A_BUTTON)))
      {
        askQuestion();
      }
  else
      {
        theAnswer();
      }
} // end of loop()
```

The logic (using the ! "not" operator) goes like this: "If the A Button is not pressed, display the Question. Else, display the Answer."

Upload the sketch to the Arduboy and give it a try. Make some changes to the responses array if you'd like. **Have fun!**

This concludes part two of the **Learn Arduboy** series.

[7] https://www.arduino.cc/reference/en/language/functions/random-numbers/random/