

```

/*
Exercise 5.2
Binary Quiz Game
Chapter Five @ http://www.tronixstuff.com/tutorials
*/

int seglatch = 13;
int segclock = 12;
int segdata = 11;
int ledlatch = 7;
int ledclock = 6;
int leddata = 5;
int speaker = 4;
int button = 2;
int possanswer=0;
int currentscore = 0;
int currentnumber = 0;
int scrollldelay = 150;
int greeting1[] = {
    0,0,0,118,121,56,56,92,0,70,118,121,80,121,0,0,0,0,0,0}; // byte values for "Hello there"
int greeting2[] = {
    0,0,0,115,80,121,109,109,0,124,28,70,70,92,84,0,70,92,0,109,70,119,80,70,0,0,0,0,0,0}; // = Press
button to start
int numbers[] = {
    63,6,91,79,102,109,125,7,127,111}; // bytes to display 0~9 on seven segment display
int greeting3[] = {
    0,0,0,109,88,92,80,121,0,0,0,0,0,0,0,0}; // "Score"
int greeting4[] = {
    0,0,0,70,92,115,0,109,88,92,80,121,0,0,0,0,0,0,0,0}; // "Top Score"
int rawreading = 0;
float rawdivided = 0;
int ldig = 0;
int mdig = 0;
int rdig = 0;
int finalread = 0;
float d = 0;
int c = 0;
float b = 0;
int e = 0;

void setup()
{
    pinMode(13, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(2, INPUT);
}

void startup() // how to make it look at button
{
    digitalWrite(ledlatch, LOW); // blank out LEDs
    shiftOut(leddata, ledclock, MSBFIRST, 0);
    shiftOut(leddata, ledclock, MSBFIRST, 0);
    shiftOut(leddata, ledclock, MSBFIRST, 0);
    digitalWrite(ledlatch, HIGH);
    for (int q = 0; q<15; q++)
    {
        segdisplay(greeting1[q+2],greeting1[q+1],greeting1[q]);
        delay(scrollldelay);
    }
    delay(600);
    for (int q = 0; q<25; q++)
    {
        segdisplay(greeting2[q+2],greeting2[q+1],greeting2[q]);
        delay(scrollldelay);
    }
    waitforpress(button);
}

```

```

void segdisplay(int lastchar, int midchar, int firstchar)
{
  digitalWrite(seg latch, LOW);
  shiftOut(segdata, segclock, MSBFIRST, lastchar);
  shiftOut(segdata, segclock, MSBFIRST, midchar);
  shiftOut(segdata, segclock, MSBFIRST, firstchar);
  digitalWrite(seg latch, HIGH);
}

void waitforpress(int inputpin)
// function to stop and wait for user to press button on input
// pin number 'inputpin'. Ensure this is set to input using
// pinMode(input, pin number)
{
  while (digitalRead(inputpin)==0)
  {
    delay(10); //
  }
  delay(1000);
  ; // debounce
}

void readdial()
// function to read the potentiometer, display the value between 0~255 on the display
// and once user has pressed the button, return the selected number as an integer
// pot analogue 5
{
  do
  {
    rawreading = analogRead(5);
    rawdivided = rawreading/4;
    finalread = int(rawdivided); // get the value of the potentiometer, convert it into 0~255
    possanswer=finalread;
    if (finalread<10)
    {
      segdisplay(numbers[finalread],0,0);
    }
    else if (finalread>=10 && finalread<100)
    {
      d=finalread%10; // find the remainder of dividing d by 10, this will be the right-hand digit
      c=int(d); // make it an integer, c is the right hand digit
      b=finalread/10; // divide z by 10 - the whole number value will be the left-hand digit
      e = int(b); // e is the left hand digit
      segdisplay(numbers[c],numbers[e],0);
    }
    else if (finalread>=100)
    {
      b=finalread/100;
      ldig=int(b); // left-hand digit
      b=finalread % 100;
      d=b/10;
      mdig=int(d); // middle digit
      d=finalread%10;
      c=int(d);
      b=finalread/10;
      rdig=int(d); // right digit
      segdisplay(numbers[rdig],numbers[mdig],numbers[ldig]);
    }
  }
  while (digitalRead(2) == 0);
}

void correctanswer()
// function to run if the user enters a correct answer
{
  currentscore++;
  digitalWrite(ledlatch, LOW); // turn off LEDs
  shiftOut(leddata, ledclock, MSBFIRST, 0);
  digitalWrite(ledlatch, HIGH);
  segdisplay(109,121,110); // blink "YES" twice
  delay(500);
}

```

```

segdisplay(0,0,0);
delay(500);
segdisplay(109,121,110);
delay(500);
segdisplay(0,0,0);
for (int q = 0; q<9; q++) // Shows "Score" then score. run twice
{
    segdisplay(greeting3[q+2],greeting3[q+1],greeting3[q]);
    delay(scrolldelay);
}
delay(100);

if (currentscore<10)
{
    segdisplay(numbers[currentscore],0,0);
}
else if (currentscore>=10)
{
    d=currentscore%10; // find the remainder of dividing d by 10, this will be the right-hand digit
    c=int(d); // make it an integer, c is the right hand digit
    b=finalread/10; // divide z by 10 - the whole number value will be the left-hand digit
    e = int(b); // e is the left hand digit
    segdisplay(numbers[c],numbers[e],0);
}
delay(500);
segdisplay(0,0,0);
}

void incorrectanswer()
// function to run if the user enters an incorrect answer
{
    segdisplay(0,92,84); // blink "no" twice
    delay(500);
    segdisplay(0,0,0);
    delay(500);
    segdisplay(0,92,84);
    delay(500);
    segdisplay(0,0,0);
    delay(500);
    segdisplay(0,109,6); // "is"
    delay(1000);
    b=currentnumber/100;
    ldig=int(b); // left-hand digit
    b=currentnumber % 100;
    d=b/10;
    mdig=int(d); // middle digit
    d=currentnumber%10;
    c=int(d);
    b=currentnumber/10;
    rdig=int(d); // right digit
    segdisplay(numbers[rdig],numbers[mdig],numbers[ldig]);
    waitforpress(button);
    for (int q = 0; q<9; q++) // Shows "Score" then score
    {
        segdisplay(greeting3[q+2],greeting3[q+1],greeting3[q]);
        delay(scrolldelay);
    }
    delay(100);
    if (currentscore<10)
    {
        segdisplay(numbers[currentscore],0,0);
    }
    else if (currentscore>=10)
    {
        d=currentscore%10; // find the remainder of dividing d by 10, this will be the right-hand digit
        c=int(d); // make it an integer, c is the right hand digit
        b=finalread/10; // divide z by 10 - the whole number value will be the left-hand digit
        e = int(b); // e is the left hand digit
        segdisplay(numbers[c],numbers[e],0);
    }
    delay(1000);
    segdisplay(0,0,0);
}

```

```
}

void endgame()
{
  for (int q = 0; q<9; q++)
  {
    segdisplay(greeting3[q+2],greeting3[q+1],greeting3[q]);
    delay(scrolldelay);
  }
  delay(1000);
  d=currentscore%10; // find the remainder of dividing d by 10, this will be the right-hand digit
  c=int(d); // make it an integer, c is the right hand digit
  b=currentscore/10; // divide z by 10 - the whole number value will be the left-hand digit
  e = int(b); // e is the left hand digit
  segdisplay(numbers[c],numbers[e],0);
  delay(1000);
  waitforpress(2);
}

void loop()
{
  startup();
  randomSeed(analogRead(3)); // seed the random number generator // put this within the loop of the
main game
  currentscore=0;
  for (int loopy=0; loopy<3; loopy++)
  {
    currentnumber=random(256);
    digitalWrite(ledlatch, LOW); // send the digits down to the shift registers!
    shiftOut(leddata, ledclock, MSBFIRST, currentnumber);
    digitalWrite(ledlatch, HIGH);
    readdial();
    if (possanswer == currentnumber)
    {
      correctanswer();
    }
    else if (possanswer != currentnumber)
    {
      incorrectanswer();
    }
  }
  endgame();
}
```