

```

int curPin = 0; // Initial current pin = 0
int offSet = 2; // Corrects the offset since we started at pin 2
int numberOfLEDs = 7;
int intVel = 100; // Initial velocity = 100
int vel = intVel;
int buttonPin = 13; // Pin 13 connects to the button
int winLED = 10; // Pin 10 is the green "winning" LED
int loseLED = 9; // Pin 9 is the red "losing" LED
int winPin = 5; // Pin 5 is the red LED in the middle of the row
int nol = 3; // The number of lives
int lives = nol;
int level = 1; // Level of the game

void nextPin() {
    curPin = (curPin + 1) % numberOfLEDs; // traverse through the pins and loop
back to the beginning
}

void resetGame() { // set game back to initial status
    lives = nol;
    level = 1;
    vel = intVel;
    curPin = 0;
}

void winSequence() { // display the lights in a "winning" pattern
for( int i = 0; i < numberOfLEDs; i++ ) {
    digitalWrite(i + offSet, LOW); // turn all the LEDs off
}

while( digitalRead(buttonPin) == LOW ) { // while the button is not being
pressed
    for( int i = 0; i < numberOfLEDs; i++ ) {
        if( ( i % 2) == 0 ) {
            digitalWrite(i + offSet, LOW); // turn off all even pins
        } else {
            digitalWrite(i + offSet, HIGH); // turn on all even pins
        }
    }
}

delay(500);

for( int i = 0; i < numberOfLEDs; i++ ) {
    if( ( i % 2) == 1 ) {
        digitalWrite(i + offSet, LOW); // turn off all odd pins
    } else {
        digitalWrite(i + offSet, HIGH); // turn on all odd pins
    }
}

```

```

    }

    delay(500);
}

}

void showLevel() { // lights up LEDs to match which level they lost on
    for( int i = 0; i < numberOfWorkers; i++ ) { // turn off LEDs
        digitalWrite(i + offSet, LOW);
    }

    for( int i = 0; i < level; i++ ) { // turn on LED's singularly until number
matches level
        digitalWrite(i + offSet, HIGH);
        delay(500);
    }

    while( digitalRead(buttonPin) == LOW ) {

    }

    for( int i = 0; i < numberOfWorkers; i++ ) {
        digitalWrite(i + offSet, LOW);
    }

    delay(1000);
}
}

void setup() {
    // Setup Input
    pinMode(buttonPin, INPUT);

    // Setup Win
    pinMode(winLED, OUTPUT);

    // Setup Lose
    pinMode(loseLED, OUTPUT);

    // Setup Output
    for(int i = 0; i < numberOfWorkers; i++ ) {
        pinMode(i + offSet, OUTPUT);
    }
    Serial.begin(9600);
}

}

```

```

void loop() {
    digitalWrite(curPin + offSet, HIGH); // turn the current pin on
    if( lives <= 0 ) { // if there are no more lives
        showLevel();
        resetGame();
    } else {
        if( digitalRead(buttonPin) == HIGH ) { // if the button is being pressed
            if( (curPin + offSet) == winPin ) { // and the LED lit is the winPin
                digitalWrite(winLED, HIGH); // turn the green LED on
                delay(500);
                digitalWrite(winLED, LOW); // turn the green LED off
                vel = vel - 10; // turn up the speed
                level += 1; // add 1 to the level
                digitalWrite(curPin + offSet, LOW); // turn the current LED off
                curPin = numberOfWorkingLEDs - 1; // start the new level at the beginning LED
                if( level == 8 ) { // if level 8 has been reached
                    winSequence();
                    resetGame();
                }
            } else { // the pin pressed was not the winPin
                digitalWrite(loseLED, HIGH); // turn on the red light
                delay(500);
                digitalWrite(loseLED, LOW); // turn off the red light
                lives -= 1; // deduct a life
                digitalWrite(curPin + offSet, LOW);
                curPin = numberOfWorkingLEDs - 1; // start the new level back at the first LED
            }
            while( digitalRead(buttonPin) == HIGH ) { } // prevents players from holding down the button to win
        }
        delay(vel);
        digitalWrite(curPin + offSet, LOW);
        nextPin();
    }
}

```