```
//I did not come up with all of this code on my own, but what I have been able to do successfully
//is continue to add functionality to my LilyPad program.  My main resource can be found at:
http://www.instructables.com/id/Lilypad-Arduino-Christmas-Sweater-with-Blinking-Li/
//Today, after several hours of puttering around and not getting very far, I finally got the motor
//and the temperature sensor to work.  I also got rid of the annoying lights that kept flashing for
//no apparent reason, and was able to modify the rhythm of my song to make it sound a little
//better.  Now that I've gotten this far, I am not quite sure how to assemble the pieces so that
//them match up to the pins.  That will be next task.  Based upon what I've heard, it may drive
me //to drink.

int speakerPin = 7;
int buttonPin = 2;
int led1 = 9;
int led2 = 11;
int led3 = 5;
int led4 = 6;
int motor = 3;
int led5= A3;
int sensorPin = A1;
int sensorValue;

int switchState = LOW;
boolean buttonClear = true;
int songChoice;
int ledPattern = true;

const int beatLength = 25;

void playTone(int tone, int duration) {
for(long i=0; i<duration * 1000L; i+=tone*2){
digitalWrite(speakerPin, HIGH);
delayMicroseconds(tone);
digitalWrite(speakerPin, LOW);
delayMicroseconds(tone);
  }
}
void playNote(char note, int duration, boolean sharp) {
char names[] = {'bflat', 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C', 'D', 'E', 'F', 'G', 'A', 'B' };
int tones[] = { 1073, 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956, 851, 758, 716, 636, 568,
506 };
char names_sharp[] = { 'c', 'd', 'f', 'g', 'a', 'C', 'D', 'F', 'G', 'A' };
int tones_sharp[] = { 1804, 1607, 1351, 1204, 1073, 902, 804, 676, 602, 536 };
```

```
if (sharp == false) {
  for(int i=0;i<14;i++){
if (names[i] == note) {
playTone(tones[i], duration);
  }
  }
} else {
for(int i=0;i<10;i++){
  if (names_sharp[i] == note) {
  playTone(tones_sharp[i],duration);
    }
    }
  }
}
void updateSwitchState(){
  int val=digitalRead(buttonPin);
  if(val==HIGH){
    buttonClear=true;
  }else{
    if(buttonClear==true){
    if(switchState==LOW){
     switchState=HIGH;
    }else{
      switchState=LOW;
    }
    buttonClear=false;
    }
  }
}
void alternateLeds() {
if (ledPattern == true) {
digitalWrite(led1, LOW);
digitalWrite(led2, HIGH);
digitalWrite(led3, LOW);
digitalWrite(led4, HIGH);
ledPattern = false;
} else {
digitalWrite(led1, HIGH);
digitalWrite(led2, LOW);
digitalWrite(led3, HIGH);
digitalWrite(led4, LOW);
ledPattern = true;
}
```

```
}
void parseTune(char notes[], int beatLength, boolean loopSong) {
  boolean play = true;
for(int i=0;notes[i]!='.'&&play==true;i++){
updateSwitchState();
if (switchState == LOW) {
play = false;
} else {
if (notes[i] == ',') {
char len[3];
int count = 0;
while (notes[i+1] >= '0' && notes[i+1] <= '9' && count < 2) {
len[count] = notes[i+1];
count++;
i++;
}
len[count]='0';
int duration = atoi(len);
delay(duration * beatLength);
} else {
alternateLeds();
char note = notes[i];
boolean sharp;
if (notes[i+1] == '#') {
i++;
sharp = true;
} else {
sharp = false;
}
char len[3];
int count = 0;
while (notes[i+1] >= '0' && notes[i+1] <= '9' && count < 2) {
len[count] = notes[i+1];
count++;
i++;
}
len[count] = '0';
int duration = atoi(len);
playNote(note, duration * beatLength, sharp);
}
delay(beatLength / 2);
}
}
```

```
if (loopSong == true) {
switchState = LOW;
}
}
void playTune (int tune) {
if (tune == 1) { // Jingle Bells
char notes[] =
"b4b4b8b4b4b8b4D4g6a2b12,4C4C4C6C2C4b4b4b2b2b4a4a4b4a8D8b4b4b8b4b4b8b4D4g6a
2b12,4,C4C4C6C2C4b4b4b2b2D4D4C4a4g12,8.";
parseTune(notes, beatLength, false);
} else if (tune == 2) { // The Holly and the Ivy
char notes[] =
"g4g2g2g4E4D4b6g2g2g2g4E4D8D2C2b2a2g4b2b2e2e2d4g2a2b2C2b4a4g8,8.";
parseTune(notes, beatLength * 1.50, false);
} else if (tune == 3) { // We Wish You a Merry Christmas
char notes[] =
"d4g4g2a2g2f#2e4c4e4a4a2b2a2g2f#4d4f#4b4b2C2b2a2g4e4d2d2e4a4f#4g8,8.";
parseTune(notes, beatLength * 1.25, false);
} else if (tune == 4) { // Baby It's Cold Outside
char notes[] =
"c2a3g1f2C6,c2a3g1f2a4f2f6,c2bflat3a1g2C6,c2bflat3a1g2bflat4g2g6,c2a3g1f2C6,c2a3g1f2a4f
2f6,C2a4f2g8.";
parseTune(notes, beatLength, false);
}
}
void setup() {
pinMode (led5, OUTPUT);//this is the light that will turn
//on when the temperature goes down
pinMode(sensorPin, INPUT); //set the temp sensor to input
Serial.begin(9600); //initializes the serial port
pinMode(motor, OUTPUT);//makes the motor an output
digitalWrite(led5, HIGH);// turns an extra LED on when it's cold
pinMode(speakerPin, OUTPUT);
pinMode(buttonPin, INPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
pinMode(led4, OUTPUT);
}
void loop() {//runs over and over again
//I added a temperature controlled motor
sensorValue = analogRead(sensorPin);
Serial.println(sensorValue);
```

```
delay(100);
if (sensorValue<=139)
{
  digitalWrite(led5, HIGH);
  digitalWrite(motor, HIGH); //turns motor on
  delay(1000); //delay for 1 second
  digitalWrite(motor, LOW); //turn motor off
  digitalWrite(motor, HIGH); //turns motor on
  delay(1000); //delay for 1 second
  digitalWrite(motor, LOW); //turn motor off

}
else
{
 digitalWrite(led5, LOW);
 digitalWrite(motor, LOW);

int val = analogRead(led1);
if (val < 0) {
songChoice = 1;
digitalWrite(led1, HIGH);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
} else if (val < 0) {
songChoice = 2;
digitalWrite(led1, LOW);
digitalWrite(led2, HIGH);
digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
} else if (val < 0) {
songChoice = 3;
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
digitalWrite(led3, HIGH);
digitalWrite(led4, LOW);
} else {
songChoice = 4;
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
digitalWrite(led4, HIGH);
}
```

```
updateSwitchState();
if (switchState == HIGH) {
playTune(songChoice);
    }
  }
}
```