MUMT 306 Final Project

Mariano Castillo Barajas

Introduction

The term algorithm can be defined as a set of applied rules in order to solve a specific problem in a finite number of steps. Inside the realm of music, the concept of algorithmic composition is directly related to this definition as it consists of a sequence of predefined tasks, aimed towards creating a finished musical piece. Even though in recent years this musical term has been strongly associated to computer-based creations, algorithmic composition has roots that stretch as far back as the times of the ancient Greeks (Papadopoulos & Wiggins, 1999). Even though there is a wide variety of methods involving computational systems for modern algorithmic composition, in the following work we will focus on a grammar-based (preestablished musical grammar composition) algorithmic creation process.

The main goal of the project was to create a short electronic dance music loop with pitch control capabilities and some level of randomization. In order to accomplish this, two main software elements and two pieces of hardware were required:

- Max/MSP for loop programming
- Arduino IDE for sensor coding
- Arduino Uno hardware for the sensor-computer connection
- HC-SR04 ultrasonic sensor for external pitch control

I. The Loop Section

In order to code the main base of the track, the Max/MSP visual programming language was used. This section was purely focused on event and sound controls where elements such as loop characteristics, instrument timing and chord progression were defined. In Fig.1, the main or "master" section of the loop is defined. The master tempo is used in order to synchronize all the loop elements through the loop manager section. The loop control activates events such as points in time where predefined instruments start and stop playing and keeps track of the elapsed time measures.

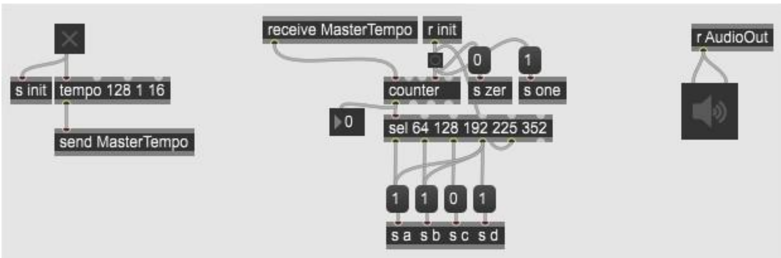

Fig. 1 Master Tempo, Loop Control and Audio Output.

Fig. 2, 3 and 4 show the instrument section of the code. Here, we find the main bass instrument, the arpeggiator and the drum section. The arpeggiator and bass sounds are generated using internal oscillators, while the drum section is triggered through MIDI. The BassGate input seen in both the arpeggiator and the bass section is the one responsible for the pitch control received in the serial to Max conversion shown in Fig. 5. The gate functions serve as triggers that activate the note corresponding to the number received through BassGate.

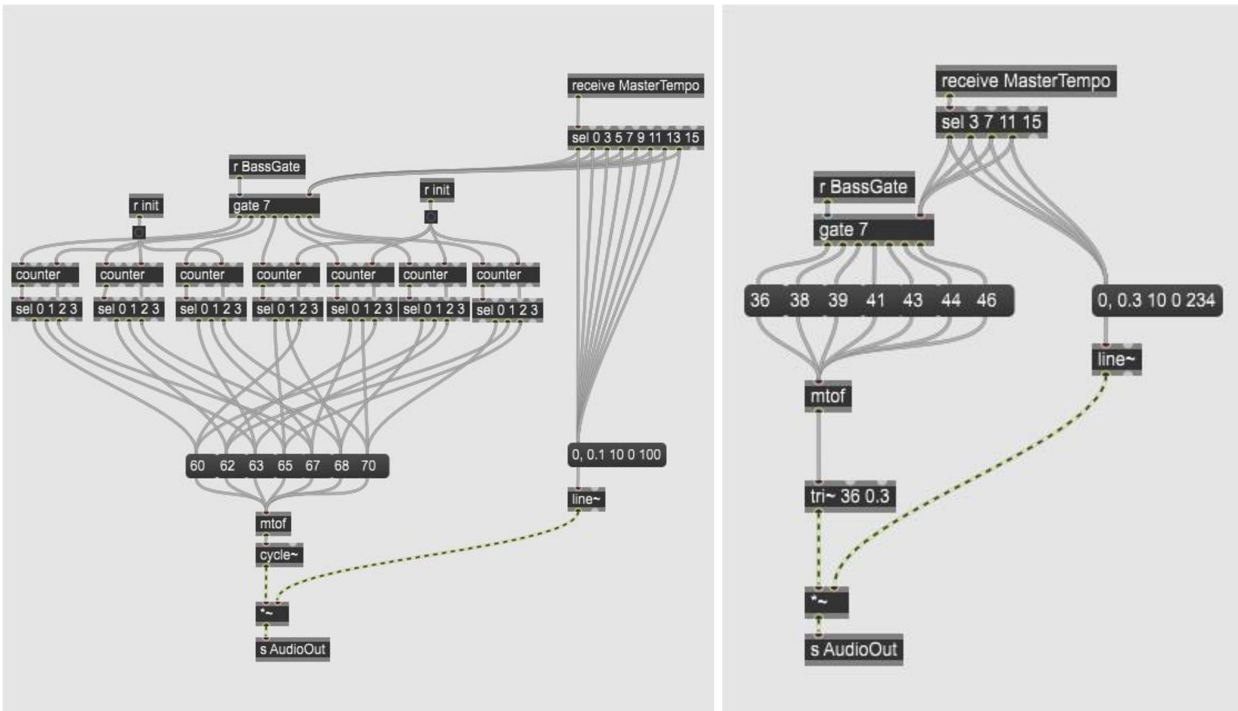

Fig. 2 & 3 Arpeggiator and Bass Section

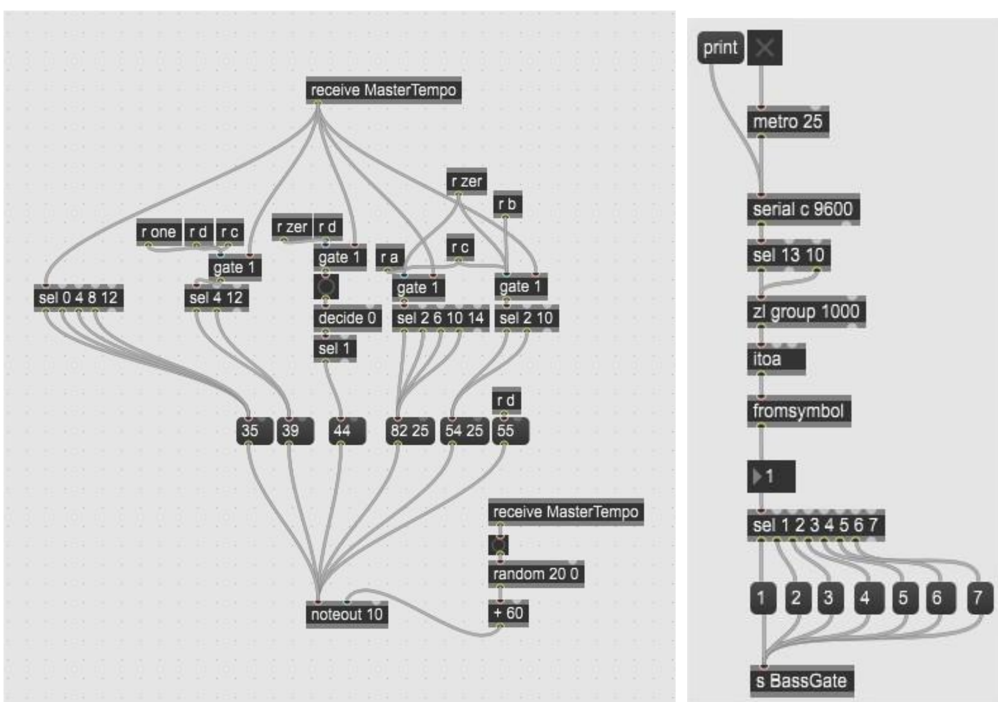

Fig. 4 & 5 Drum Section and Serial to Max Conversion

II. The Arduino Sensor & Code

For the Arduino code, the incoming data from the ultrasonic sensor had to be processed in a way that enabled it to print values from 1 to 7 (the seven harmony degrees inside the tempered occidental scale). For this, the range of response of the ultrasonic sensor had to be limited to a maximum of 1000 milliseconds to a minimum 300 milliseconds. Then, the ranges between these values were mapped to the corresponding chord or harmonic number. Finally, because of the continuous measurement nature of the sensor, a conditional print had to be defined in order to only print once every state change. Then, the sensor was connected to the Arduino through a protoboard in order to succesfully trigger the required elements in Max.

```
const int ultratrigPin = 3;
const int ultraechoPin = 2;

long durTime;
bool canPrint = true;

void setup() {
Serial.begin(9600);

pinMode(ultratrigPin, OUTPUT);
pinMode(ultraechoPin, INPUT);
}

void loop() {
  digitalWrite(ultratrigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(ultratrigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(ultratrigPin, LOW);

  durTime = pulseIn (ultraechoPin, HIGH);

   if(durTime >= 1000){
    durTime = 1000;
    }
   else if(durTime <= 300){
    durTime = 300;
    }
   if(durTime < 1000 && canPrint){
    int noteNo = map(durTime, 300, 1000, 1, 8);
    Serial.println(noteNo);
    canPrint = false;
    }
   else if (durTime >= 1000 && !canPrint){
    canPrint = true;
    }

 delay(75);
}
```

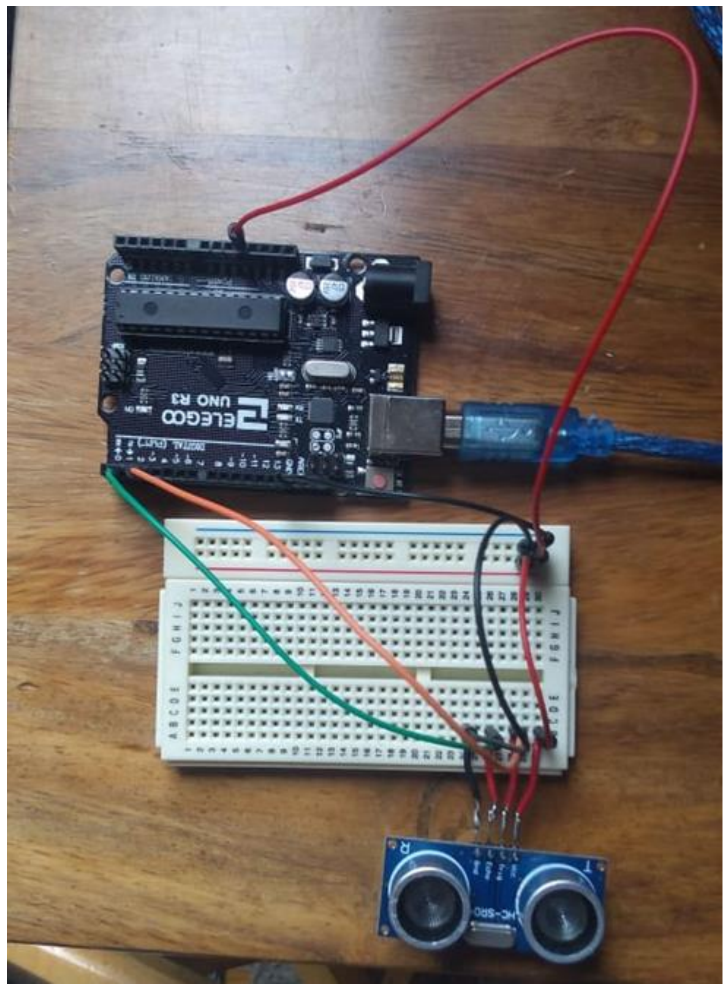

Fig. 6 Sensor Code and Connection

Conclusion

To conclude, one can state that the project goals were accomplished in a satisfactory manner. Some issues regarding the synchronization of the main loop and the sensor's response capability were encountered, but the overall results were positive. In the future, the loop could be extended, and a variety of different sensors could also be tested in order to find the most suitable option for the current version of the project.

References:

Papadopoulos, G. P., & Wiggins, G. W. (1999). AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects. *Computer Music Journal*, *23*(4), 110–117. https://doi.org/10.1162/comj.1999.23.4.79