My parrot patch has these features:

- average room noise level detector
- maximum buffer size of 10000ms
- buffer is cropped to recording length
- playback ramps from -400 cents to 400 cents and plays the recording twice
- simple bandpass filter before output
- level meter and waveform display

The average room noise level detector was one of the first features I implemented to automatically set the recording threshold since some rooms are naturally louder than others. The average value is calculated by looking at all the values sent from the levelmeter over 5 seconds and using the mean object to get the final number. If there was ever any reason not to use this, I also added a simple number object which lets the user set the threshold themselves, with the default being -45 dB. I added 15 dB to the calculated average value because, in my testing, it was very easy for any slight sound to cross the threshold value, but with the extra 15 dB, it required direct speaking into the microphone.

Once the patch is turned on, it will start listening for any sounds louder than the threshold, and it will record until either nothing is detected above the threshold or until 10 seconds are over. One issue I was facing at the start was with the buffer size. I wanted to record for any length of time, but as soon as I started thinking about maximum buffer size, then my patch started working smoothly. I set my maximum buffer size to 10 seconds, but the problem was that if a sound continued for longer, it still recorded it but didn't store it in the buffer and therefore didn't playback the extra sound. Additionally, for shorter recordings, the entire 10-second buffer was played back even if the recording didn't fill the whole buffer. I solved these issues using the timer object, which received a bang when the recording started and received another bang when the recording ended to calculate the length of the recording on the fly and send it to a crop message that would shorten the buffer.

Also, a different timer would receive a bang at the beginning of the recording and receive another bang 10 seconds later from a metronome if the recording had continued for that long so that the recording would be stopped and would be played back.

My playback feature includes a pitchshift~ which receives the length of the recording from the first timer and generates a ramp from -400 cents to 400 cents over that period of time. Since it also plays the recording twice, you're essentially hearing one from -400 to 0 and one from 0 to 400. There is also a reson~ right before the final output, which differentiates the recording from the sound source a little more.

Overall, this was a challenging but rewarding project, and it was interesting to work with buffers and recording because they require some attention to detail which might mess up the whole patch if not cared for. Since I actually owned a similar parrot toy when I was younger, it was fun to implement that functionality in Max.