PaL workshop documentation

Marcel Kress & Jacob Vigsø

https://github.com/Marcelkress/Deepseek-soundscape.git

Theme and goal

We created a soundscape portraying a scuba diver at the bottom of a mysterious ocean. The goal was to create a feeling of being underwater, with deep noise, strange unknown sounds, in addition to your own heavy breath. To simulate being underwater we used low-pass filters on all of the sounds. Some sounds were recorded and altered, while others were synthesized from oscillators and noise.

The user has access to interactive elements to alter the experience. For example, they can change the chance of some sounds playing. The user also has access to some of the settings of each sound, such as volume, reverb, frequency, time intervals, and stereo pan.

To trigger a sound playing, we used the “metro” object. This is a metronome that sends a bang after a set amount of time. We then used the tabplay~ or tabread4~ objects to play a one-shot of a sound.

Some of the sounds also only play randomly. This is done through a series of objects. Firstly, a metronome sends a bang to the “random” object. This returns a (pseudo) random number between 0-100, which is then evaluated in an “expr” object. This object receives two values: value A which is the random number, and value B which is the threshold value of playing the sound. The expression then checks if value A is larger than value B, and if it is, sends a bang. Otherwise, nothing happens.

The line object is also widely used to create envelopes for the sounds that are played. Here we can adjust things like attack, sustain, and delay, which are all used to adjust how the sound is played.

Sounds

1. **Breath in**

This sound was created to mimic a person inhaling. We wanted to create the feeling of being underwater, but also inside a chunky diver suit. Therefore the breathing should be somewhat heavy. We did this by slowing down the original sound clip and using a lowpass filter on it, cutting all the high frequencies. The original clip was a recording of us hitting a bag with cans, creating a sound with a lot of seemingly random and high-frequency sounds, but it was also a very short sound. We did not intend this to be a breathing sound when we recorded it, but after tweaking the speed at which it got played and cutting out the higher frequency we thought it could be used for that purpose. We also created a method in PD, that would play the sound at random speeds, to create some variance in the pitch.

This was done with the phasor object’s input value being multiplied by a random value. We found that a value between 0.18 and 0.28 created the effect we wanted, but since the random object can't return floats within an interval we had to implement our own solution. This was done by getting a random number between 0 and 10, then adding 18 to ensure a value between 18 and 28. Then multiplying with 0.01 to scale it down, which resulted in random values between 0.18 and 0.28.

2. Breath out

To create the breath-out sound, we recorded a needle being dragged across the floor. We then slowed it down slightly, added a low-pass filter, and faded it in and out. This sound mimics the bubbles that come from exhaling underwater.

This recording was layered with the breath-in sound, but in reverse. This keeps some of the noise from the breathing and makes it sound “opposite” to the breath-in.

We also added a short burst of noise which is played at a lower volume than the bubbles which makes the entire exhale part sound “fuller” with the two sounds layered.

The noise makes use of the line object with .5 second attack, 1 second sustain and 2 second release. This mimics how when exhaling, the volume of the sound produced starts high and gradually lowers.

3. Deep ambient noise

The deep ambient noise was used as just the overall sound of being underwater, using noise in PD, to create a constant but random sound. To not have this sound like wind or rain, we used a low-pass filter, to create a deeper rumbling. As sound travels faster underwater it can appear to be very noisy. The noise is cut off at 244hz, and played through a bob~ object. The bob object creates a resonance at the given cutoff frequency, with the amount of resonance being controllable.

4. Monster growl

This sound was created from a spring being dragged across a metal staircase. The sound was then altered to be deeper and have some reverb, in addition to a low-pass filter. The reverb is important when creating this underwater space as the sound has a lot of “freeroom” to bounce around the environment, also putting a low-pass filter on it created a good way of mimicking how water muffles sound.

This is also one of the specialized and randomly played sounds. The user can adjust if they want to sound to play on the left, right side or both sides. This is done by adjusting the volume of the right and left channels inversely. It plays randomly by a chance specified by the user.

5. Whale sound

The whale sound is an oscillator that is controlled by two line object objects controlling the volume and frequency respectively. The first line object is used to create a gradual increase and decrease in volume. The sound is played in mono so the listener can not figure out which direction the sound comes from.

The second line object changes the frequency as the sound is being played. When whales make sound, they do not keep the exact same pitch/frequency. Therefore, for our whale to sound natural, we had to do this.

Additionally, the sound only has a certain chance of playing

6. **Strange metallic sound**

For this sound, we hit the spring against the stairs railing. Then, to create the metallic sound, we slowed down the sound a lot and only used the resonating part, cutting away the first “hit” sound.

To have the sound sounding more like something coming from underwater, we put a Low-pass filter on it and a lot of reverb. Inside PD, we used a metro into random value into an expr. This way each time the metro bangs we get a random value, we can then using the expr see if that value is higher than something, this way we can create a bit of randomness in how often this sound is played.

7. **Submarine ping sound**

This sound was synthesized from a sine wave, with a decay and attack method, to control the length it got played and how “aggressive” it gets played. attack = 1 10 , Delay = 100 frequency = 770. We did not create a randomness play for this, as we wanted it to be constant, as we already had a whale sound which could sound very similar to this, but having it be constant gives the feeling that this is something “mechanical” or handmade.

8. **Low strange ambience**

For this sound, we strapped a spring tightly between two boards so it sat upright, and then we hit it from the side with a piece of metal to create a sound that resonated a lot.
To achieve the result, we trimmed the clip to remove the initial hit, applied a low-pass filter , stretched the sound, and added some reverb. The sound has a set chance of playing every 10 seconds. The interval is smaller on this particular sound as it is not very different from the base ambience. Therefore, it does not stand out as much but is used to create alterations in the sound.

Discussion

We started by going out and recording all kinds of sounds. We didn’t think much about what exactly they could be used for, but instead let ourselves be inspired as we began processing them. Each sound and its purpose was decided after recording it. Some sounds also do not serve a specific purpose, as in they don’t necessarily represent anything specific, but leaves it up to the listener to imagine what the origin could be. This helped us reach the dark and mysterious tone we were going for.

The method of recording without specifics in mind might not be technically correct, but it helped us to be inspired more by each sound and by trying different things, as we had no clue what types of sounds might be good in specific sound use cases; breathing sounds, monster growl etc.

Overall most of the sounds could have used the pitch variation to create a soundscape that sounds a bit different, even when listening to it for a long time.

The pitch variation was only applied to one sound because it was difficult to implement. Playing back audio files using the phasor object presented some limitations in customizability, but it was achieved and could have been applied to additional sound if we had the time. The difficulties we had were the timing of the breathing sound, as we changed its playback speed but that sometimes would cut it off. But we created a system where it would finish the whole sound, and then turn it off and then instead of starting over immediately again, it would wait for the new metro bang. This method gave us more control of the playback timings

Something that worked very well was the randomness in the soundscape. Many of the sounds only had a certain chance of playing at different intervals which created a dynamic soundscape.

Resources

How to play back audio files in PD:
https://www.youtube.com/watch?v=mvqXk6t7TQ8