Audio Engineering Society

Convention e-Brief 416

Presented at the 144th Convention
2018 May 23–26, Milan, Italy

FXive: A Web Platform for Procedural Sound Synthesis

Parham Bahadoran[1,2], Adán L. Benito[1,2], Thomas Vassallo[1], and Joshua D. Reiss[1]

[1]*Queen Mary University of London, London, UK*   [2]*FXive.com, London, UK*

Correspondence should be addressed to Joshua D. Reiss (joshua.reiss@qmul.ac.uk)

ABSTRACT

FXive is a real-time sound effect synthesis framework in the browser. The system is comprised of a library of synthesis models, audio effects, post-processing tools, temporal and spatial placement functionality for the user to create the scene from scratch. The real-time nature allows the user to manipulate multiple parameters to shape the sound at the point of creation. Semantic descriptors are mapped to low level parameters in order to provide an intuitive means of user manipulation. Post-processing features allow for the auditory, temporal and spatial manipulation of these individual sound effects.

1 Introduction

High quality audio is essential to the creative industries. Sound effects are often defined as non-musical, non-speech sounds used in creative contexts such as film, games or virtual reality. The process of sourcing such sounds is often achieved through Foley [1] or via use of large databases of pre-recorded audio samples. Sound synthesis has remained comparatively unpopular for a number of reasons. Synthesis of high quality effects is often too computationally expensive to be considered a viable alternative, and general purpose synthesisers may result in the design process being arduous for sound designers.

The aim of this paper is to demonstrate that reliance on sample libraries can be avoided by the use of lightweight and versatile sound synthesis models. Such models do not rely on stored samples, and provide a rich range of sounds. Rather than searching libraries and attempting to fit samples to a desired goal, sounds can be shaped at the point of creation.

We introduce an online real-time sound effects synthesis platform, Fxive [2], that demonstrates this concept. We present its different components, including sound source modules and post-processing tools. The platform reduces the technical knowledge required to produce high quality, realistic sound samples through synthesis while alleviating the restrictions imposed by pre-recorded audio.

A large spectrum of sounds are procedurally [3] generated through custom DSP synthesis algorithms. All bespoke synthesis models integrated in the platform have been listed in detail in Table 1. Many of the models have been evaluated in terms of their quality and realism using the framework described in [4]. Two further models outside the standard framework are also provided; a sinusoidal model allowing the user to synthesise sounds based on sample analysis, and a latent force model demonstrating this for selected impact and musical instrument sounds [5].

2 Architecture

FXive's framework is written entirely in JavaScript and relies on a client-side architecture. Sound generation and manipulation is achieved through the employment of the Web Audio API (WAA)[6] and customised JavaScript processors and functions to handle interactions. Every synthesis model and audio effect is encapsulated using the JSAP[7] audio plugin standard and audio chains are handled by JSAP's Plugin Factory. This provides increased flexibility by allowing the creation of complex interconnected audio graphs in a number of configurations.

The framework takes care of sequencing the loading and creation of the different chains and resources and creates automatic connections between interface elements and plugin's parameters. Sounds generated using the provided models and post-processing tools can be recorded by the user and rendered to a stereo track to be downloaded as a .wav file.

Two global chains are differentiated in this architecture: the sound-effect chain, composed by one or more sound effect plugins in parallel, and the post-processing chain, which is comprised of a series of audio processing and spatialisation tools (see Fig 1).
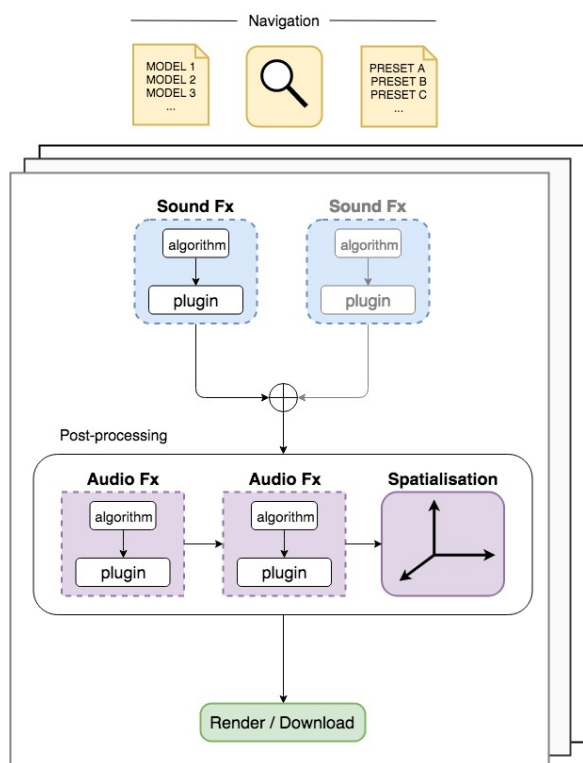
Figure 1. Diagram of the system's architecture.

2.1 Sound effects

All sounds found on FXive are built with real-time sound synthesis models tuned to synthesise an accurate representation of a class of sounds while exposing parameters that control the sound qualities. Though the synthesis model is tailored to generate a parent sound and does not require use of samples, the availability of control parameters allows production of large banks of presets[1]. A portion of the interface for one such synthesis model, the siren, is depicted in Fig 2. Additional benefits over pre-recorded audio files include the ability to manipulate the audio source. This means characteristics such as pitch and duration may be determined by the user without the need to pitch shift and time stretch any audio. The mapping of low level features to perceptually intuitive controls allows for quick manipulation in order to fine-tune sound effects to suit a particular situation.

2.2 Sound Scenes

Sound scenes (or soundscapes) are offered within the framework as a conjunction of parallel synthesis models [2] forming the sound effect chain. These combinations of sounds form more complex textures and illustrate what can be achieved with the platform.

---

[1] Over 100 searchable presets are currently available.

[2] Each with gain compensation, master panning, EQ…

2.3 Audio Effects

A chain of common post-processing effects has been included accompanying each model for further sculpting and manipulation of the source. This chain is comprised of a saturation effect that permits precise shaping of its clipping curve, feedback delay, convolution reverb that offers a selection of impulse responses, 5-band para metric equaliser, master gain and panning. Each audio effect can be bypassed accordingly and the sound effect muted if necessary.

2.4 Navigation

The platform provides several forms of interaction and aims to provide an intuitive user experience and an easy path to access what different users may need. A preset search functionality is built in the homepage. It allows searching for a particular sound which will populate a list of presets from all models matching the searched keyword. A dedicated Presets page also exists for browsing through all presets from different models. For each preset there are options of either downloading a short snippet or visiting the model's page with the preset loaded and ready to manipulate or download. For a more detailed approach, a Models page is provided that can be browsed for a full list of all synthesis models (see Table 1). The models incorporate the full range of sound effects, including sound textures, impact sounds and soundscapes.

Different users may prefer to search for and download a sound effect in a few seconds, manipulate a synthesis model to edit and adjust a preset, or shape a new sound from scratch.

3 Interactions

Due to the nature of the development techniques dis cussed in section 2, models can be regarded as self-contained sound source applications. This provides the flexibility to interact with them in different ways. One or more synthesis model can be chained together and manipulated using an interface. Furthermore, a URL query system can be used on an external host to trigger different actions from the models without the need for a user interface.

3.1 User Interface

3.1.1 Parameter Controls

The low level parameters of sound synthesis algorithms are mapped to semantically meaningful descriptors which are exposed to the user. Each model is given parameters which either describe a physical or real world characteristic of the sound being generated (e.g. density of rain, crackling of fire) or a semantic descriptor commonly used for sound design (e.g. warmth, depth). This allows the user to manipulate the

synthesis   engine   without   the   need   for   deep   understanding of the algorithm being employed.
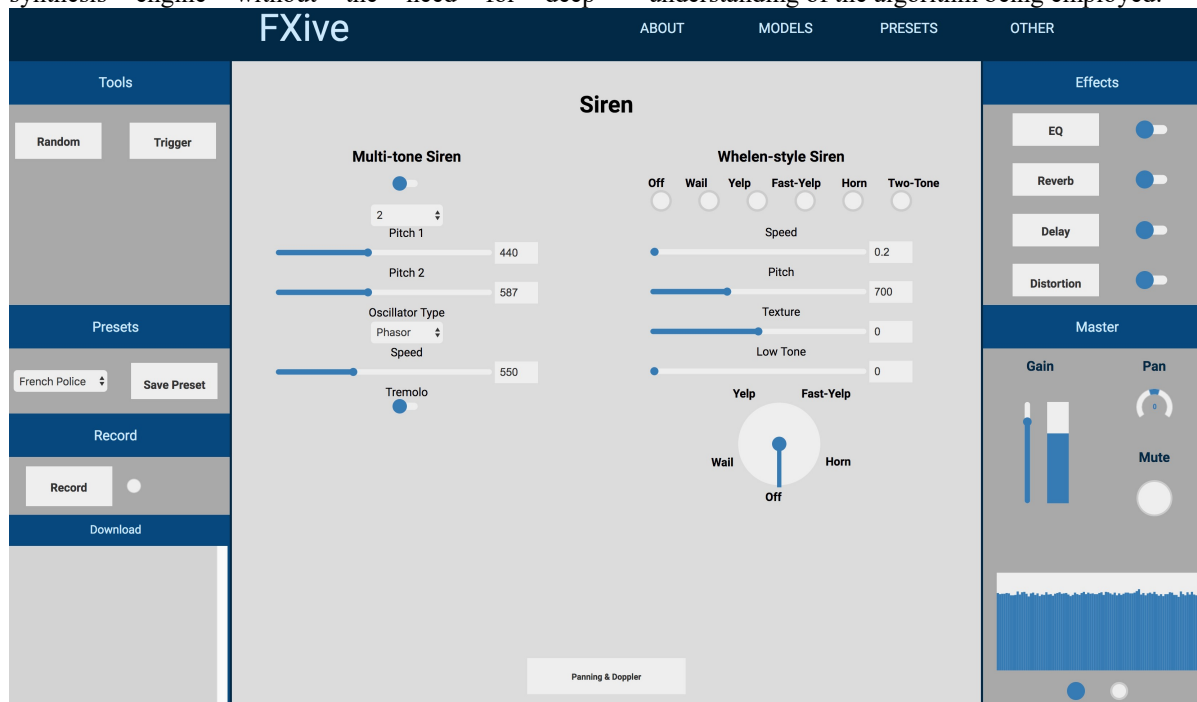

Figure 2. Interface for the siren synthesis model. Not shown are audio effects, randomisation, timeline and spatialization.

A graphical user interface is designed using NexusUI [26] with a variety of control objects for manipulating one or a combination of parameters. The objects are mostly those used in typical audio software, e.g. buttons, sliders and knobs. Other control objects are sometimes uaed to concisely convey elaborate control parameters. Device orientation and motion tracking is also proposed to create more complex combinations of parameters with temporal and spatial attributes. This allows aspects of sound synthesis to be controlled by movement of eligible devices (e.g. a mobile phone).

3.1.2   Randomisation

Values of sliders on the interface can be randomised by the user via a dedicated set of controls. Randomisation takes part within the sliders' range and always employs the last user input as a reference point to   calculate   the   random   variations.   A   control parameter allows the user to select a percentage of the range over which the new values will span. A button permits triggering of randomisation at will.

3.1.3   Spatialisation

A Spatialisation feature is built using the WAA's Pan nerNode which places the sound source around the listener on a 3D Cartesian coordinate. This technique al lows manipulating the location and movement of sound sources and may be used as a powerful sound design tool to further sculpt sound scenes.

The spatialisation model takes into account the sound objects' direction, orientation, velocity and provides parameters for setting the distance model. The only control exposed to the user is a two-part interface to place of the sound source on a 3d coordinate. A 2 dimensional slider is used to place the sound horizon tally (left/right) and vertically (above/below) around the listener and an additional slider controls the third dimension (front/back).

3.1.4   Timeline

A timeline feature is implemented for triggering a model at specific times. Individual timeline tracks can be generated and used to trigger each of the trigger based parameters of the model. The current version of this feature provides discrete sequencer tracks with adjustable step size and sequence length, and an option for looping.

Each trigger-based parameter can be triggered at desired times by marking those points on the timeline of the trigger track. Parameters which can be used include model trigger parameters (e.g. fire, explode, hit, etc), any of the presets and the Randomiser. The Randomiser can be triggered at specific intervals to applying a small percentage of randomisation to the model parameters and create a natural variation in the synthesis over time.

3.2 Online Query System

FXive's framework is equipped with a system that al lows sound effects to be queried by a host via URL without the need for manual setting of their parameters. The framework decodes and interprets the query string and sets the different actions accordingly. The correct sequencing and loading of events related to both audio chains and processing is taken care of by the frame work. All interpretation and sequencing processes are automated by specific JavaScript routines once the URL has been loaded. Audio is still generated real-time on client-side and relies on browser compatibility with the WAA. This query system can be used to set presets auto matically, update and edit plugin states, trigger buttons and even download recorded excerpts.

4 Summary

This paper introduced an online real-time sound effect synthesis service. The platform is built around a database of bespoke sound effect synthesis models, which are based on fine-tuned algorithms, specially developed for generating a large spectrum of sounds. Synthesis models are developed with the sound design process kept in mind, so meaningful control parameters are provided. This library of sound effect synthesis models, post-processing tools and temporal and spatial placement functionality provides the user with enough functionality to produce a wide range sound effects of high quality and flexibility, suitable for use in sound design projects.

FXive can be accessed online at fxive.com .

5 Acknowledgments

References

[1] Ament, V. T., The Foley grail: The art of performing sound for film, games, and animation, CRC Press, 2014.

[2] Bahadoran, P., et al., "A System for Online Sound Effect Synthesis," 2017, UK Patent 1719854.0 (Pending).

[3] Farnell, A., "An introduction to procedural audio and its application in computer games," Audio Mostly, v. 23, 2007.

[4] Moffat, D. and Reiss, J. D., "Perceptual Evaluation of Synthesized Sound Effects," ACM Transactions on Applied Perception (to appear), 2018.

[5] Wilkinson, W., et al., "Latent force models for sound: Learning modal synthesis parameters and excitation functions from audio recordings," 20th International Conference on Digital Audio Effects, 2017.

[6] Adenot, P. and Wilson, C., "Web audio API," W3C Working Draft, December 2015.

[7] Jillings, N., et al., "JSAP: A plugin standard for the web audio API with intelligent functionality," Audio Engineering Society Convention 141, 2016.

[8] Adami, A., et al., "On Similarity and Density of Applause Sounds," Journal of the Audio Engineering Society, 65(11), pp. 897–913, 2017.

[9] Heinrichs, C., et al., "Human performance of computational sound models for immersive environments," The New Soundtrack, 4(2), pp. 139–155, 2014.

[10] Heinrichs, C. and McPherson, A., "Mapping and Interaction Strategies for Performing Environmental Sound," IEEE VR Workshop: Sonic Interaction in Virtual Environments (SIVE), 2014.

[11] Turchet, L., et al., "What do your footsteps sound like? An investigation on interactive footstep sounds adjustment." Applied Acoustics, 111, pp. 77–85, 2016.

[12] Farnell, A., "Marching onwards: procedural synthetic footsteps for video games and animation," Pure Data Convention, 2007.

[13] Cook, P., "Physically informed sonic modeling (phism): Synthesis of percussive sounds," Computer Music Journal, 21(3), pp. 38–49, 1997.

[14] Selfridge, R., et al., "Creating Real-Time Aeroacoustic Sound Effects Using Physically Derived Models," Journal of the Audio Engineering Society (to appear), 2018.

[15] Selfridge, R., et al., "Sound synthesis of objects swinging through air using physical models," Applied Sciences, 7(11), p. 1177, 2017.

[16] Selfridge, R., et al., "Real-time Physical Model for Synthesis of Sword Swing Sounds," 14th Sound and Music Computing Conference, 2017.

[17] Cook, P. R, Real sound synthesis for interactive applications, CRC Press, 2002.

[18] Farnell, Andy, Designing sound, Mit Press, 2010.

[19] Pekonen, J. and Jylhä, A., "3-D Sound Synthesis of a Honeybee Swarm," Audio Eng. Soc. Conv. 127, 2009.

[20] Wilkinson, W. and Reiss, J. D, "A Synthesis Model for Mammalian Vocalization Sound Effects," 61st Audio Engineering Society Conference: Audio for Games, 2016.

[21] Selfridge, R., et al., "Physically derived synthesis model of an Aeolian tone," Audio Engineering Society Convention 141, 2016.

[22] Mengual, L., et al., "Modal Synthesis of Weapon Sounds," 61st Audio Engineering Society Conf.: Audio for Games, 2016.

[23] Rocchesso, D. and Fontana, Federico, The sounding object, Mondo estremo, 2003.

[24] Hendry, S. and Reiss, J. D., "Physical Model- ing and Synthesis of Motor Noise for Replication of a Sound Effects Library," 129th AES Convention, San Francisco, 2010.

[25] Selfridge, R., et al., "Physically Derived Sound Synthesis Model of a Propeller," 12th Audio Mostly Conf., p. 16, ACM, 2017.

[26] Taylor, B. et al., "Simplified Expressive Mobile Development with NexusUI, NexusUp, and NexusDrop." NIME, pp. 257–262, 2014.

Category	Model	Basic Description	Based on
Action	Applause	Distributed and synchronised hand-claps based on filtered, shaped noise bursts	[8]
	Creaking door	Creaky sound of a door opening/closing, controllable door movement	[9, 10]
	Footsteps	Quasi-periodic sequence of shaped impacts on various surfaces	[11, 12]
	Shaker	Pseudorandom overlapping and adding of small grains of sound	[13]
	Swinging object	Aeroacoustic model of objects swung in the air based on compact sources	[14-16]
	Whistle	Noise, modulated and passed through resonant filters	[17]
Animal	Birds	FM and AM synthetic bird call sounds	[18]
	Insects	A composite model made of several families of insect sounds	[18, 19]
	Mammal sounds	Wolf howl and lion roaring based on physical models of vocal chords	[20]]
Artificial	Alert	Frequency sweep modulated by fast rising and falling amplitude envelope, delay based resonators and filters	[18]
	Beep	Oscillators multiplied by control signal that alternates between 0 and 1 at fixed intervals	O
	Droid	Shaped oscillators imitating droids talking sound	[18]
	Lightsabre	Sheath and unsheath noise made of delayed noise for phase interaction. Hum and swing sounds from swept oscillator frequencies	[18]
	Ringtones	Four-tone frequency-modulated ring sounds	[18]
	Siren	Two Siren models with tunable parameters for many types of emergency vehicle siren sounds	O
	Telephone	Modern (DTMF) and old dialing and ringing phone sounds	[18]
	Teleport	Frequency modulated sawtooth sent to flanger and filterbank	[18]
	Whoosh	Bandpass-filtered noise with amplitude and frequency envelopes	O
Environmental	Aeolian tones	Aeroacoustic model of air passing through a string	[14, 21]
	Bubbles	Pseudorandom overlapping and adding of small grains of sound	[20]
	Electricity	Electric hum and spark noises with phasing	[18]
	Fire	Sound of fire, controllable crackle, hissing and lapping components	[18]
	Pouring water	Sound of pouring liquid into a container with bubbles and waterflow components	[18]
	Rain	Sound of rain, controllable density, rumble and ambience	[18]
	Stream	Sound of running water, controllable bubble density and sound texture	[18]
	Thunder	Combination of noise sources, staggered delays, waveshaping, and delay-based echo.	[18]
	Wind	Howling wind, gusts and wails based on air passing by different objects	[18]
Impact	Bouncing	Filtered noise mimics impact material. Bounce intervals calculated using coefficient of restitution according to object and initial height	O
	Explosion	Filtered white and pink noise modulated using exponential ADSR envelopes	O
	Gun	AK47 with shell detonation and gas explosion, generates variety of other gun sounds	[18]
	Gunshot	Modal synthesis with incorporated residual	[22]
	Metal impact	Modal synthesis of metal impacts	O
	Ricochet	Fast filter sweep of noise source to mimic a doppler effect	
	Rocket	Sound of complete rocket launch, chamber, exit noise and rocket components	[18]
	Rolling	Physical model of rapid sequence of impact sounds	[23]
Mechanical	Clock	Filtered noise with delay-line based resonators	[18]
	Electric motor	Physical model of a DC motor	[24]
	Fan	Blade, Motor and noise components, produces a range of propeller and motor sounds	[18]
	Jet	Turbine component comprised of clipped oscillators, burn component comprised of filtered and clipped noise. Both modulated according to engine speed	[18]
	Propeller	Aeroacoustic model of propeller-powered aircraft sounds	[14, 25]
	Squeaky toy	Sequence of filtered, modulated noise bursts	O
	Switch	Rapid sequence of short click sounds	[18]
Soundscape	Factory	Adjustable sound scene of various machinery and artificial sounds	O
	Night scene	An adjustable sound scene comprised of light wind, fire and cricket sounds	O
	Stormy day	Sound scene created combining thunder, wind, rain and stream sounds	[18]

Table 1: Summary of the models offered on the platform, their classification, and the original works from which they are originally informed. 'O' indicates that it is based on an original design.