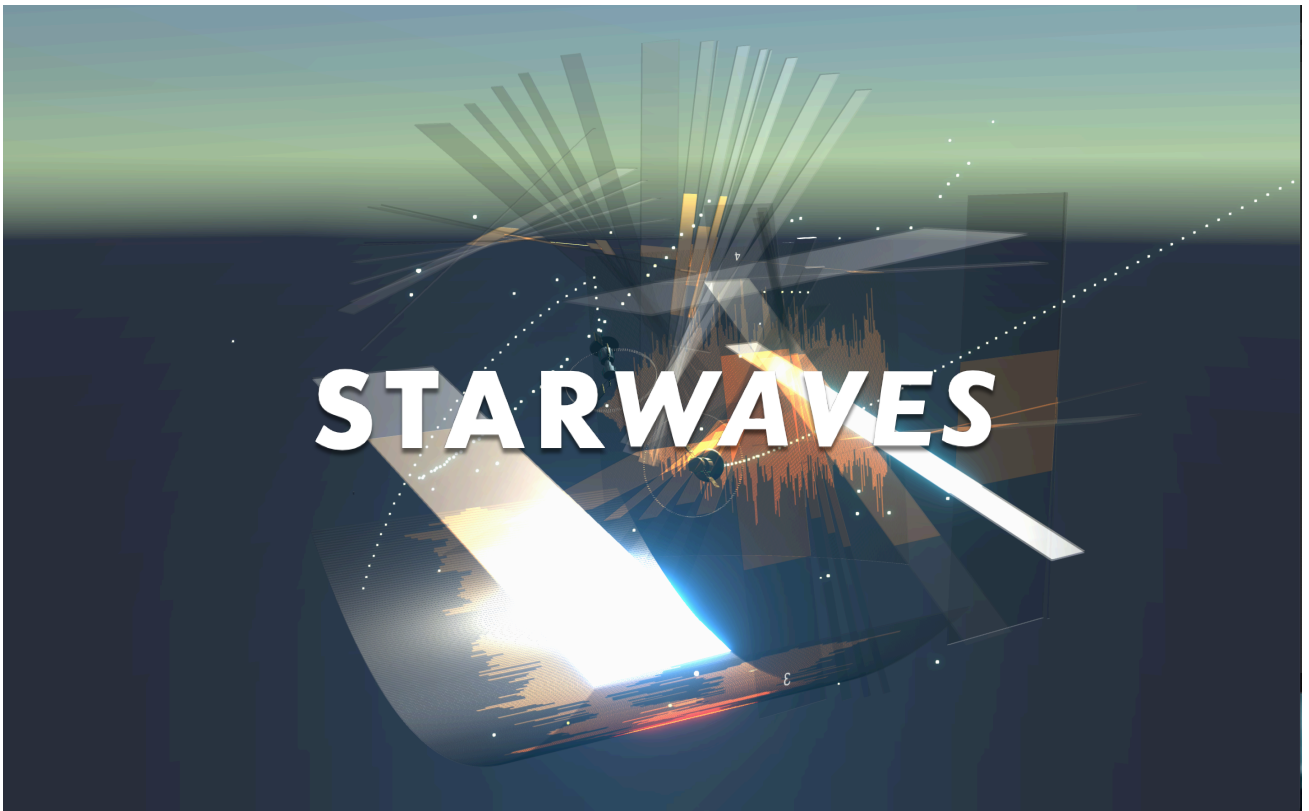


# SONIC PLANET

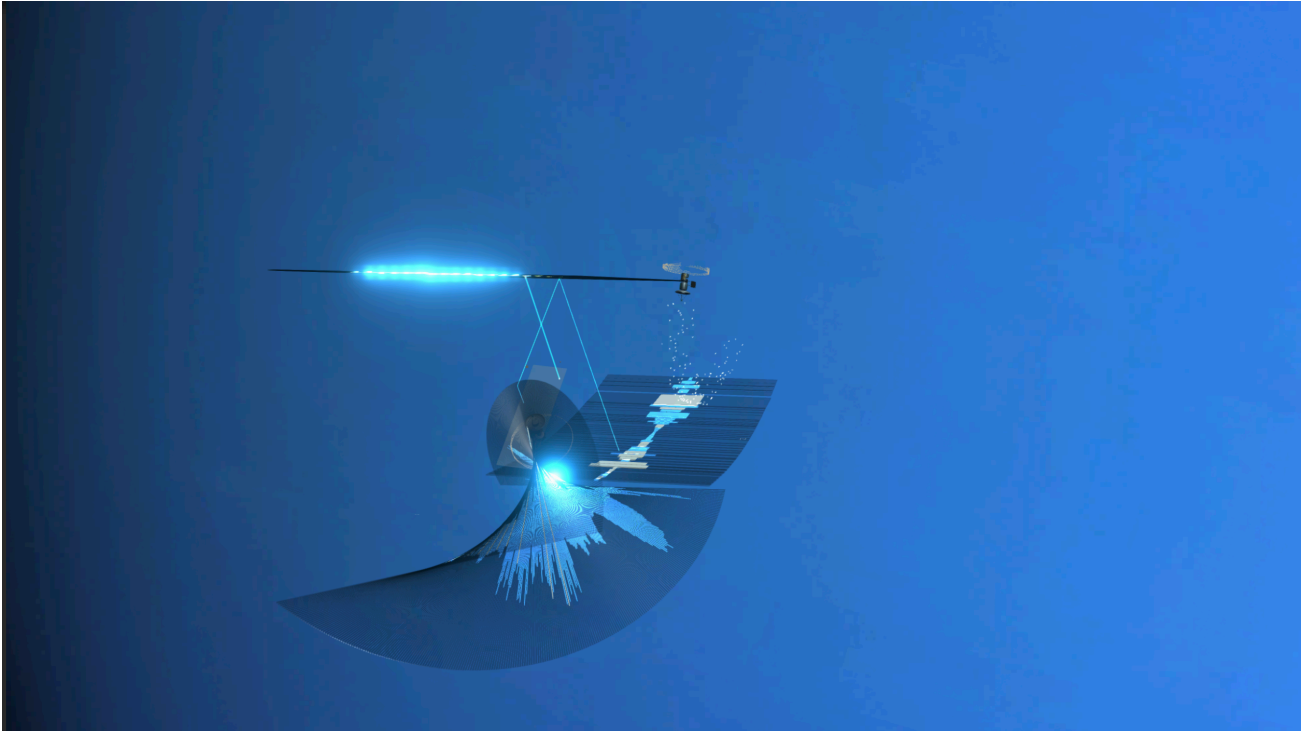


**STARWAVES / STARWAVES VR IS DESIGNED & PROGRAMMED BY SINAN BÖKESÖY.**

**ALL RIGHTS RESERVED**

[WWW.SONICPLANET.COM](http://WWW.SONICPLANET.COM)

# WHAT IS STARWAVES



StarWaves is an audiovisual scene architect, the embodiment of sound, space and visual design rolled into one.

StarWaves gives the sound designer the architect's role by creating possibilities of dialogue between these multiple elements.

## IN THE MAKING OF SONIC ARCHITECTURES

StarWaves can import the sonic material in its concrete form as recorded files. Then this digital data is being represented as 3D rigid physical forms in space. We name them "the platform blocks" distributed spatially, a structure which stems from the sonic material being imported. There can be 100s of them, reaching the granular levels of sonic material. The layout of the platforms can be designed with 3D tools like positioning, rotation; they can be stretched, bent, scattered along their physical dimensions in continuous transformations.

As Goethe wisely stated : the architecture itself is "frozen music". How do we interact with this architecture to create the sonic textures ?

We know that sound travels as waves in an environment. It originates from a source and travels towards a distance, then interacts with the surface material of the environmental objects, gets



reflected, refracted, and absorbed within all the physical reality of its nature, granting a flexible relation to spatiality.

Sound remains as a hidden existence until it reaches our body. StarWaves uses on its scene the transmitters which send particles and beams of light to the space ( reminiscent to particle behavior of light or sound ) They hit the platform blocks ( representing the sonic material ) get reflected and collide with further objects, travel in distance until they die.

The time they collide with these rigid bodies they trigger sonic events and start a flow of synthesis processes. It is this continuous kinematic interaction, this physical realm which composes the sonic being and applies continuous transformations on its texture.

## **VISUAL, CAUSAL AND NARRATIVE FIDELITY OF MOTION**

The poetic expansion of architectural elements comes with the contribution of mathematics, geometry, physical laws ( kinematics ) altogether what defines this environment. The rich motion dynamics constitute a choreographed occupation of the StarWaves space, filled with actions triggering sonic events and with a formal beauty of narrative fidelity.

On StarWaves, we don't interact directly with the sound data as on a standard sound synthesis tool but we design the architecture of the scene which will compose the sonic organization through the spatial interactions and let perform this on a timeline. Here, the sound has no separate existence from space, time and motion.

## **TIME IS THE MIND OF STARWAVES.**

We can create alternate scene designs and store them as individual states. StarWaves makes it possible morphing gradually between these visible and audible alternates. The dimension of time will play us perceptual tricks during these transitions. As the time invades the audience mind, it develops the illusion of a theater of audiovisual forms that move in space.

## **WHY IN "SPACE" ?**

It is merely the very "space" which is an abstraction from our habitat and brings us to the non-gravitational environment of the universe, where there is no sound ! It is this emptiness which provokes the sensual penetration into it, and the sound is a perfect narrative medium to do this.

"NASA's next urgent mission should be to send good poets into space so they can describe what it's really like." Shannon Hale.

As the architects of these virtual scenes and themes of StarWaves, we feel the privilege of describing it with sound design and without doubt there is the poetry within.

## WHY STARWAVES?

The name “StarWaves” comes from the sense of wonder and fantasy which has stimulated us during the project development and by witnessing its immense possibilities waiting us to discover.

As we keep looking for new frontiers in space and get fascinated with its new discoveries and the use of newest technology, we also need to investigate new instruments for sound design and new possibilities for musical creation. Analog synthesizers, keyboards and such equipment of the past were the cradle of electronic music, but one cannot live and progress in a cradle forever. Because I do believe that the future lying ahead is more prolonged than this past.

With current technological advances, we are limited only by our imagination and our will to create. And there are singularities on the timeline of advances in computer hardware and software applications. Today, StarWaves sees the daylight at one of them as it combines multiple disciplines and design elements powerfully and roll them into one high fidelity application.

sonicPlanet will continue to produce at great pace the next level of computer software for sound design, combining science, art and technology.

Dr. Sinan Bökesoy

## PRELIMINARY NOTES

- This version of StarWaves VR is available to Oculus Quest3 and Oculus Quest Pro models.
- If possible, it is recommended to use StarWaves VR with a headphone in addition to the built-in audio system Oculus Quest.
- Sometimes presets need a few seconds to set the buffers and install the parameters. During that period the performance of StarWaves VR might be glitchy.

## WHAT IS NEW ON STARWAVES VR

StarWaves VR is ideantical to StarWaves2 on Windows / OSX in synthesis engine and preset design with the limitation that platform division number being imited to 100 instead of 250.

## FILE ACCESS / OPERATIONS ON STARWAVES VR

StarWaves VR app installs the following directories on Oculus Quest, You can access them with an Android File Access utility from your computer, do backups and install your custom samples.

Once you launch the Android File Access utility you will notice the **com.sonicPlanet.Starwaves** directory.

Presets : All the preset bank files of StarWaves are here, you will see the *Factory Bank* and a *User Bank* folders by default inside this directory.

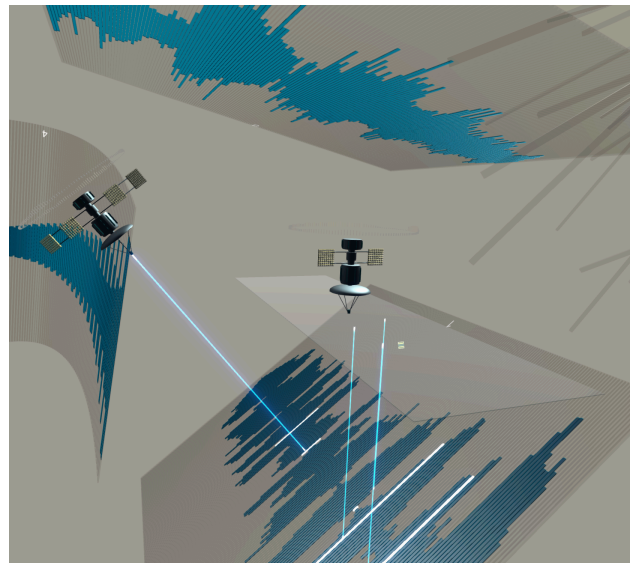
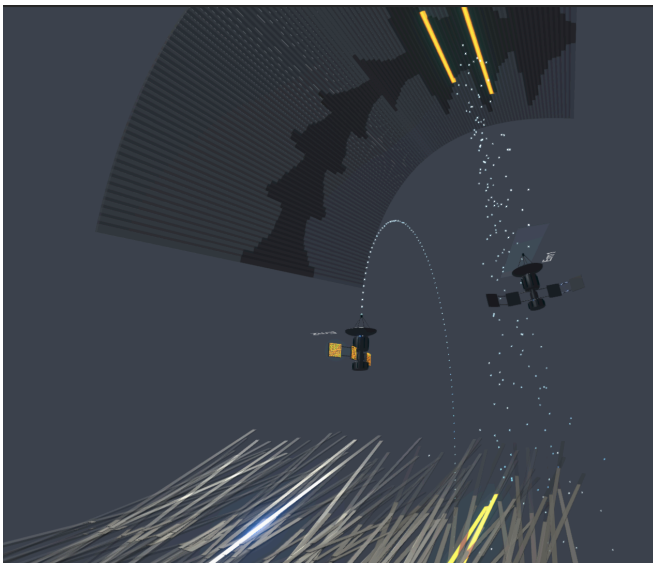
Samples : The sample .wav files are inside this folder, you can put your own samples here as well which will be then listed on StarWaves app's samples menu.

TuningScales : The scale tuning scale files are located inside this folder.

## THE STRUCTURE OF STARWAVES VR

StarWaves includes 2 “**Emitters**” which generates particles or beams and 4 “**Platforms**” which hold the digital sample data and relevant synthesis engine to be triggered by these particles / beams.

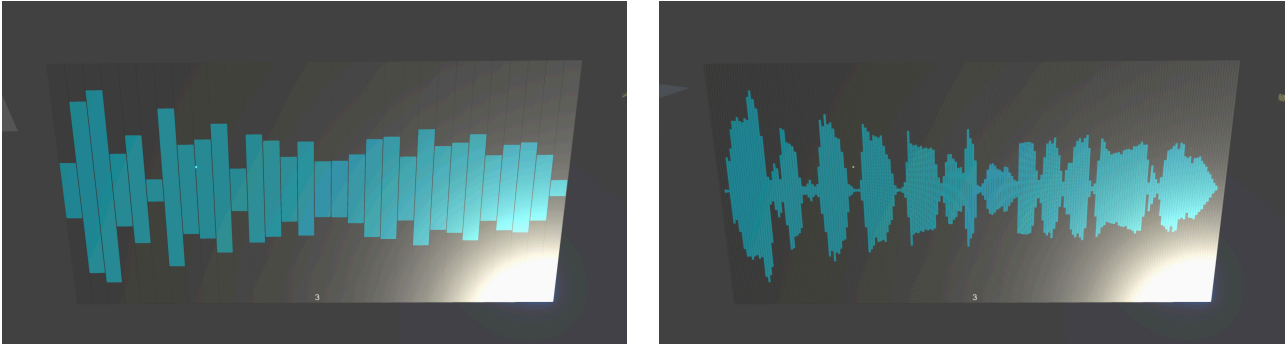
The Emitters emit either particles or a direct laser beam and basically we expect them to hit the platforms triggering their waveform content. Below you can see both cases where on the left, the emitters create particles and on the right, the beams.



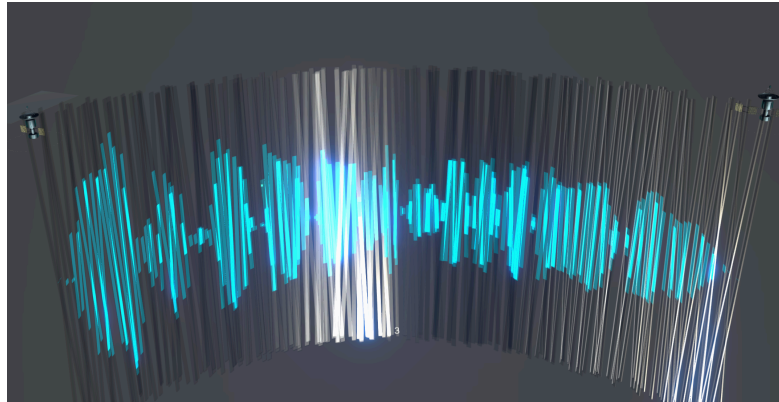
The Platforms contain the digital sample data and they represent this data segmented with 3D rectangular blocks. Once the sample data is loaded to a platform it divides it with a defined quantity of blocks by the user. (It can be between 1 - 100 ).

The time each block receives a particle or a laser beam from an emitter, it will respond by flashing a color and playing the synth voice depending on its sample content and synthesis parameters defined by the hitting emitter’s sound engine. The hit point and its angle ( along the vertical and horizontal axis of each block ) defines also some parameters to be used for the sound synthesis process.

Below you see the same platform holding a sample data. On the left, it is segmented and divided with 30 blocks and on the right, with 99 blocks. Each block object represents the starting and ending point of the sample data on the timeline. The visualization however represents the RMS<sup>1</sup> value of the sample data in that block region.



Below you see again the same platform but bended with a parabolic curve and the blocks are a bit displaced with a random factor each. StarWaves has numerous tools handling the 3D position and rotation of platform blocks and therefore effecting directly the emitter hits which will be used for the sound synthesis process.



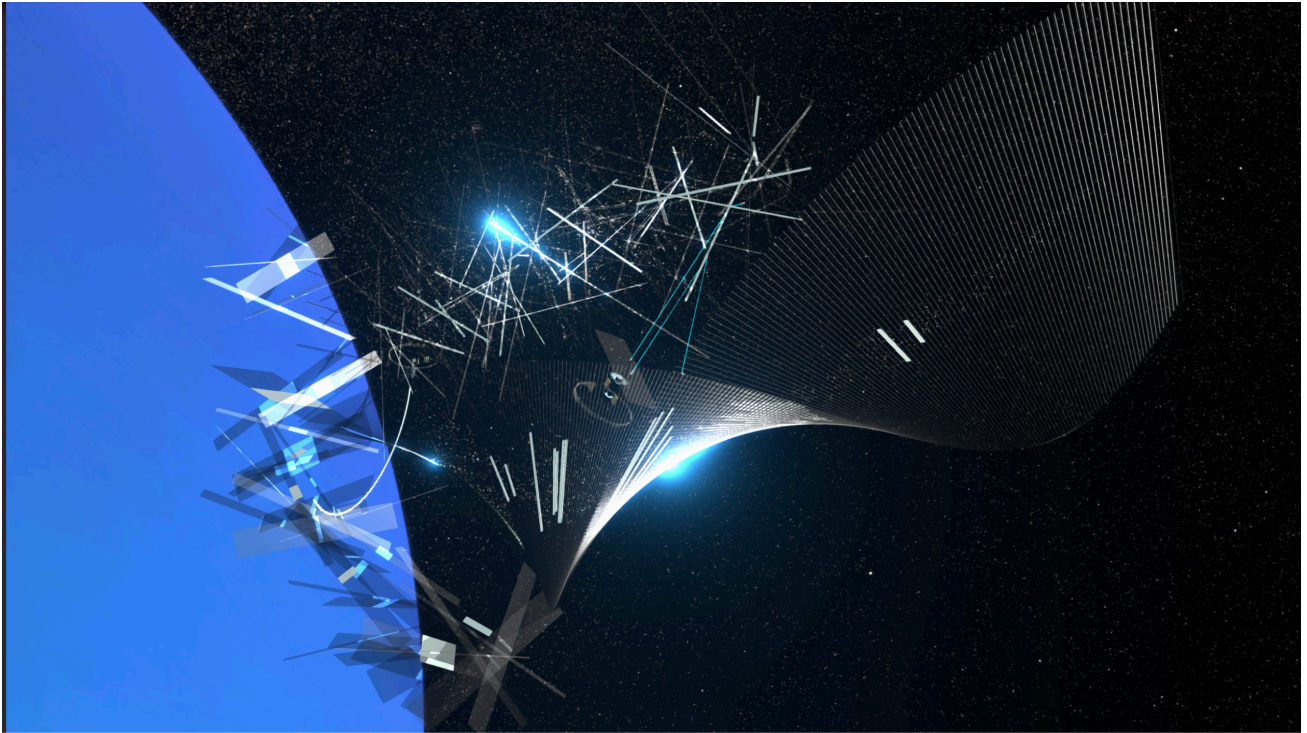
The emitters particles can be made bouncing from the platforms so that they might hit other platforms on their course. Also the laser beam can be reflected from the surface of the platform and hit another platform. ( A laser beam can hit 3 platforms at the same time , 1 direct and 2 reflected. )

The emitter can send particles with variable rate / speed / duration. Also the emitter can send 3 laser beams at the same time with variable starting points.

---

<sup>1</sup> RMS : Root Mean Square of a sample block data gives the average power of the signal in that block time range.

In physics, light can be modeled by particles, moving in light speed and a laser beam is nothing else than a very focused and directed beam of light. So the dynamics of the StarWaves emitters are built with this inspiration.



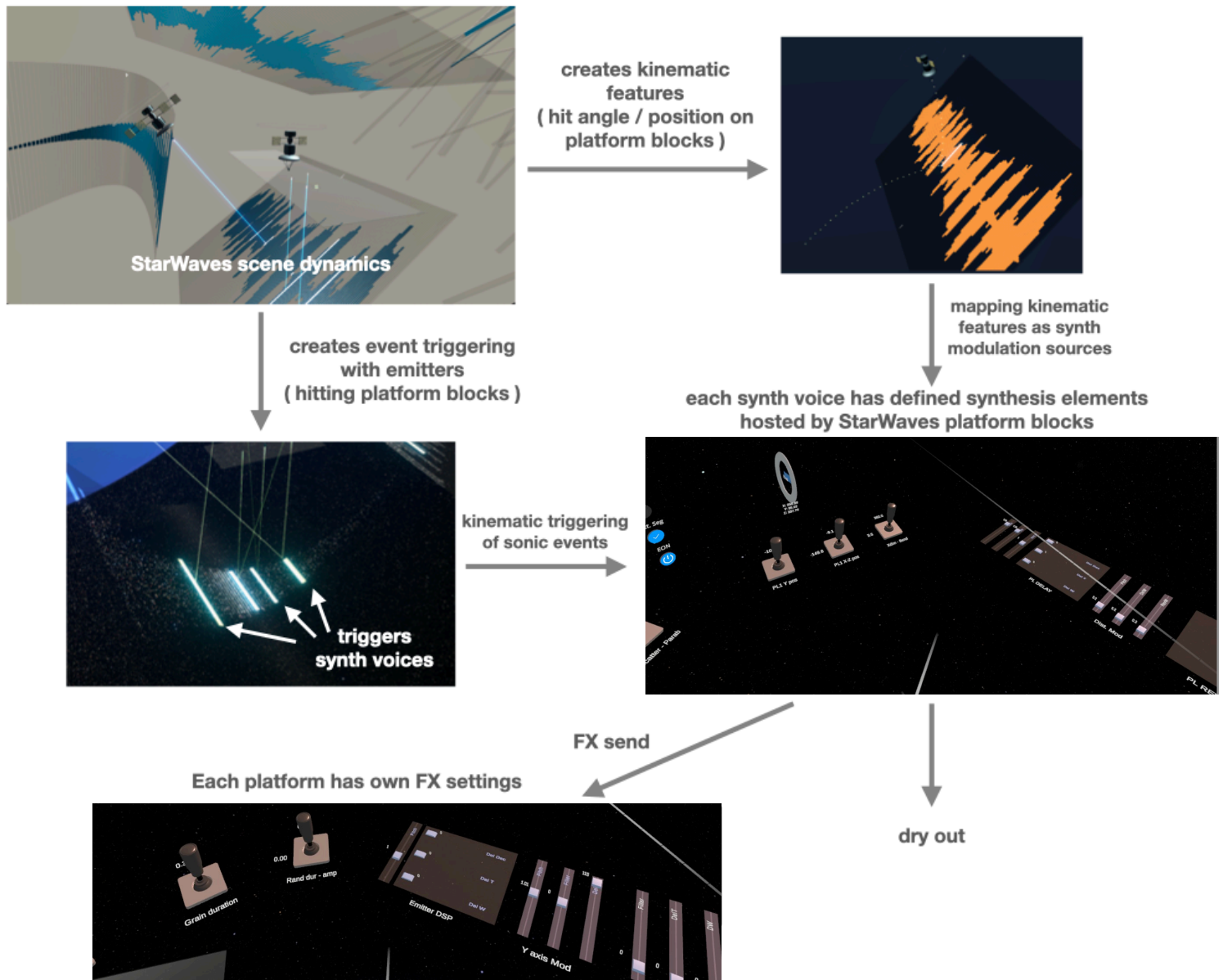
The emitters and platforms constitute the space-atmospheric, non-gravitational 3D scene of the StarWaves. It is an interaction space where the dynamics of physical movement, kinematics which result to an audio-visual experience to be seen and heard in a direct relationship.



## THE FLOW CHART OF STARWAVES VR

The motor of StarWaves is the kinematic interaction between the emitters and the platforms. The platform blocks hold the sample data.

They are representing the sound object as formed with concrete 3D blocks.



The blocks are triggered with emitter particles or beams when they collide with the platform blocks surface, each representing a synth voice.

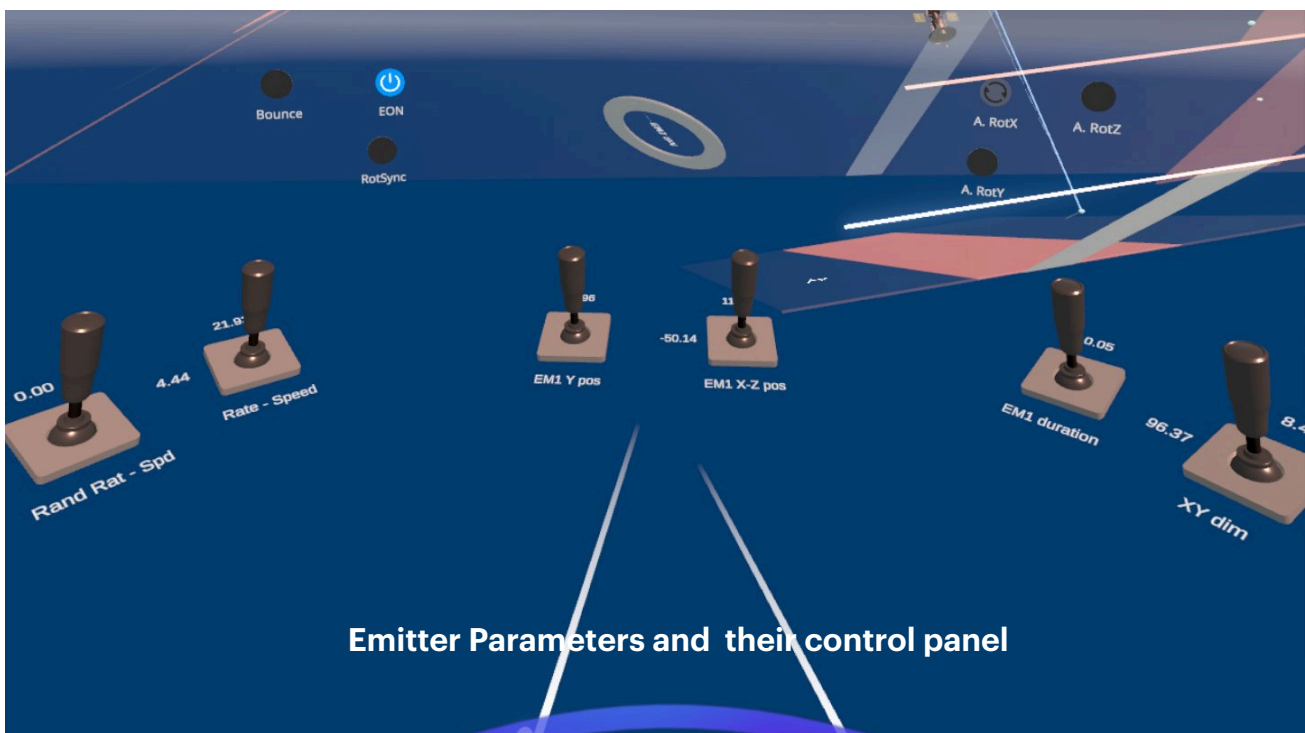
Each collision event creates a synth voice trigger and its kinematic features serve also as modulation data for the synth engine of each voice.

Each platform controls a dedicated FX section ( pitchshifter, delay, reverb ) where the platform block sound output can be sent for further processing.

The platform and emitter control parameters let the user define their behavior, content and spatial orientation.

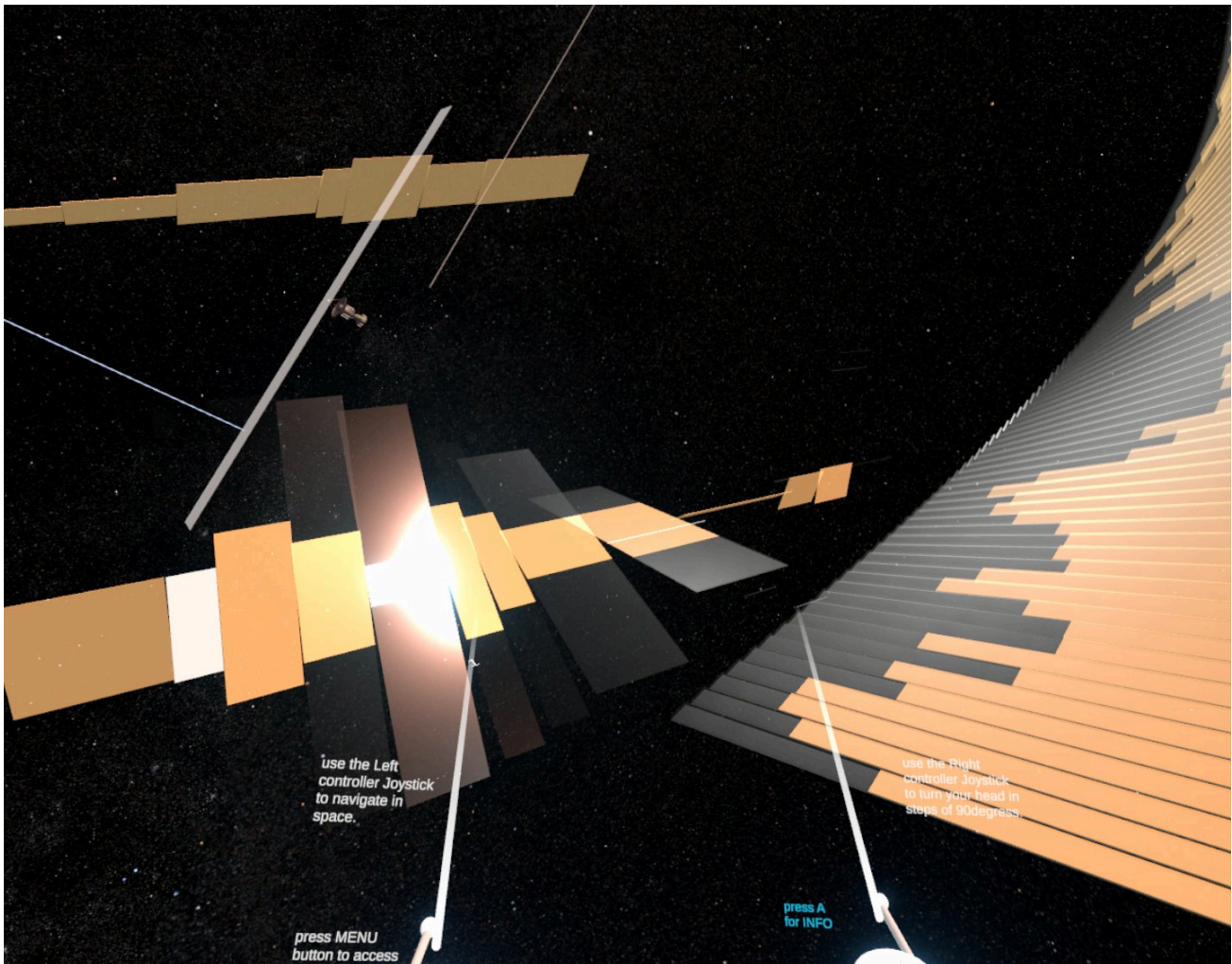
The FX section of each platform can use also some kinematic features to map to effect parameters such as the distance of the emitter from the platform.

The FX section output and Dry output from each platform combines the sonic output of StarWaves.



## THE CONTROL PANELS OF STARWAVES VR

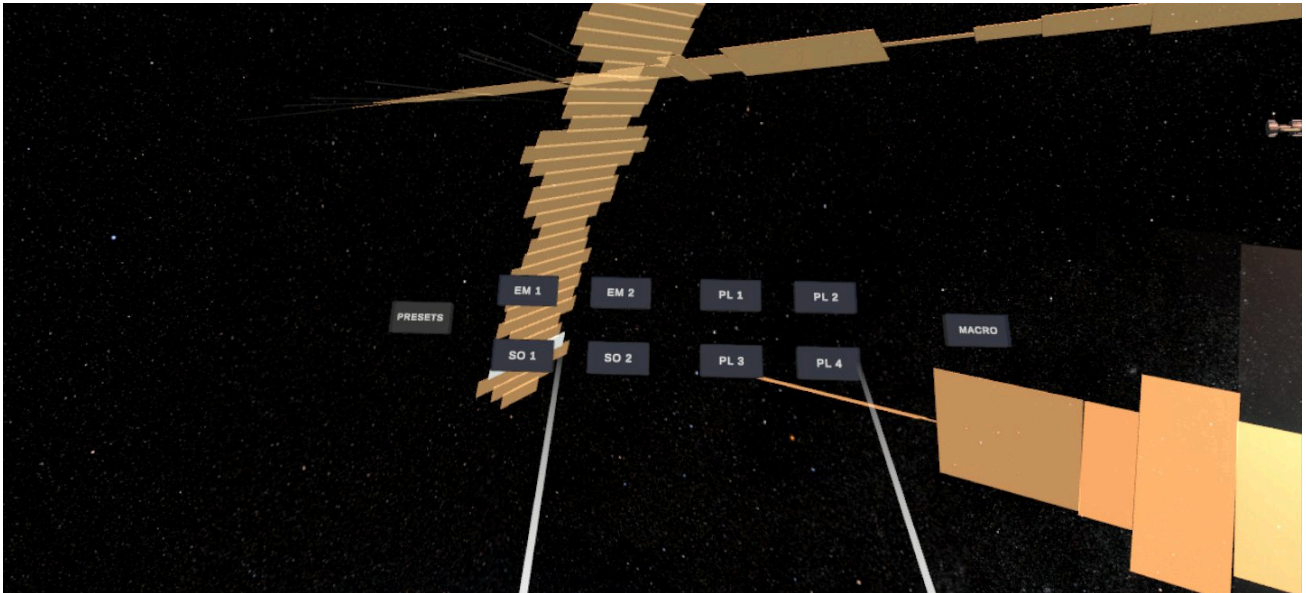
The StarWaves control layout is minimalistic and is made reminiscent of retro-style space ship panels. There is a main panel and sub panels revealed according to the actions on the main panel. Below is the app opening state of StarWaves VR.



Right away, you will see some helper texts on the sides of both controllers, which lead to enter the navigations panels of StarWavesVR.

By pressing the MENU button on the Left controller, you will see the menu navigation panel, which will always open on your front and follow your camera perspective.

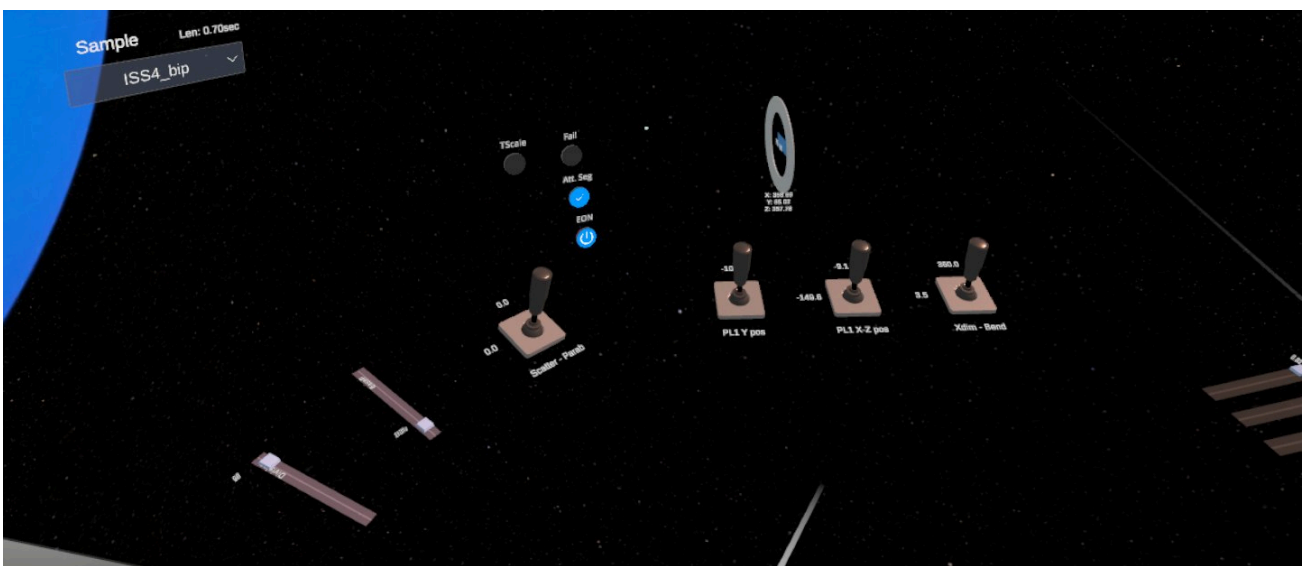




Here you can access to the Preset panel, Emitter panel, Sound Object Panel and the 4 Platform panels and also the Macro parameters panel with controller ray interaction.

As indicated on the initial helper texts, the Left controller joystick will navigate and move you in space in the direction where you do look.

The “A button” on the Right controller will reveal simple instructions on each user interface element.

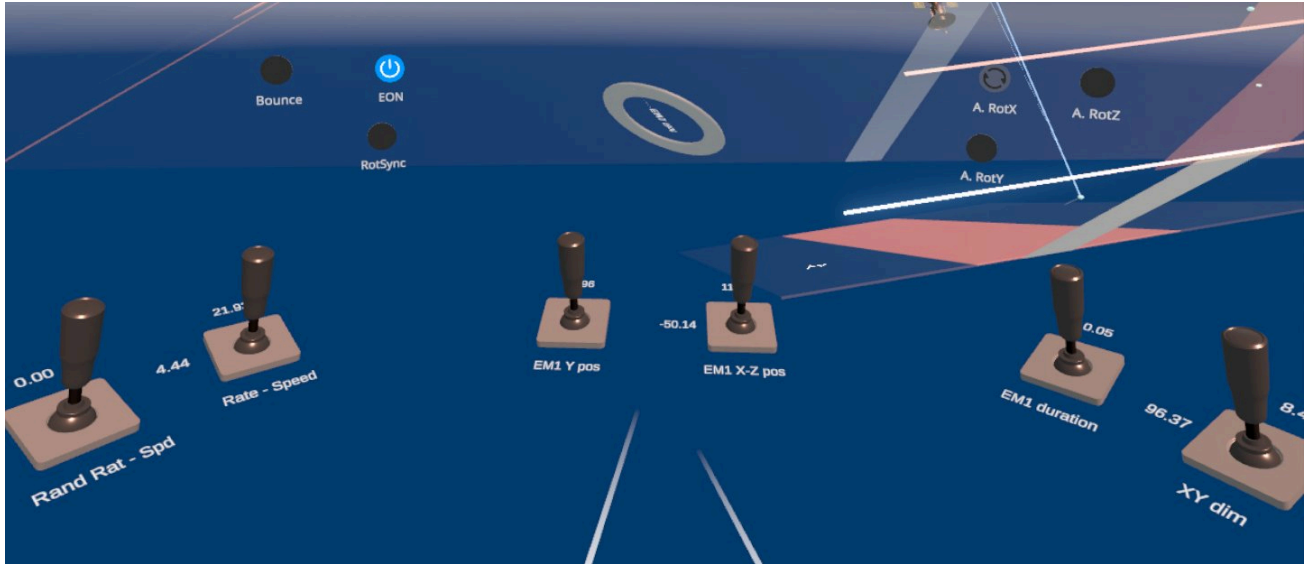


The user interface elements of StarWaves VR are 3D inter-actable objects, such as mono or bi-directional joysticks, button switches, rotation disks and sliders.

The joystick levers function as incremental parameter value changes, which work in precision and the sliders offer a frictionless smooth control environment. The panel layout is surrounding you spatially and reopening each panel automatically re-centers it in front of you.

The joystick lever and the sliders need to be grabbed. The buttons are poke-able elements and the 2D menu panels, menu switches etc. can be interacted with controller rays.

## THE EMITTER PANEL



The Emitter panel defines the behavior of the particle emitter and also the 3D spatial definition of the emitter itself. On the scene, Emitter 1 has always two solar panels for you to easily distinguish it and the second one has 4 solar panels.

**EM duration** : This joystick defines duration of the particle life span , if too short they might be dead even without hitting any object.

**EM Rate** : Emitter rate ( relevant joystick x-axis value ) defines the density of the particles emitted in one second.

**EM Speed** : Emitter speed ( relevant joystick y-axis value ) defines the speed / how fast the particles travel in space.

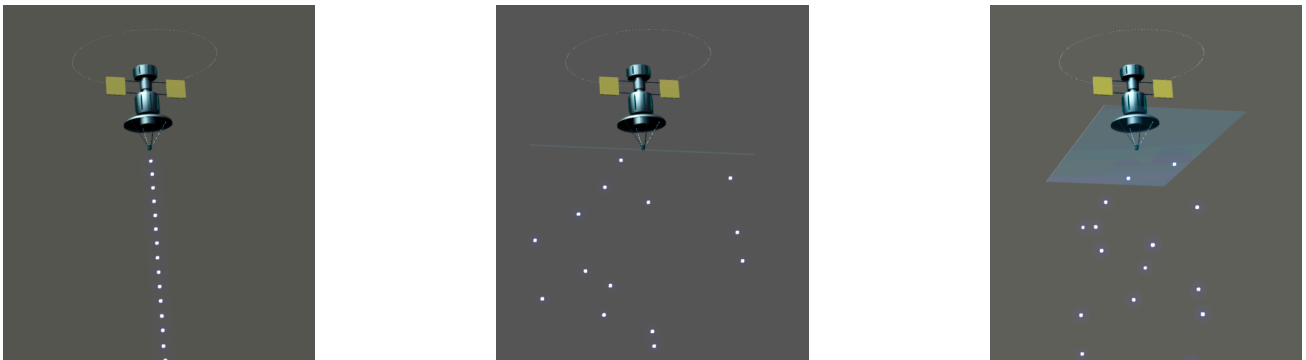
**Rand rate**: Emitter rate ( relevant joystick x-axis value ) randomization factor.

**Rand speed**: Emitter speed ( relevant joystick y-axis value ) randomization factor.

**X dim** : Normally the particles are emitted from a single point ( the tip of the emitter satellite ). With this setting ( relevant joystick x-axis value ), we can distribute the particle starting points along the emitter x-axis line randomly ( the length of the line is set basically with the X dim value ).

**Y dim** : Normally the particles are emitted from a single point ( the tip of the emitter satellite ). With this setting ( relevant joystick y-axis value ), we can distribute the particle starting points along the emitter Y-axis line randomly ( the length of the line is set basically with the Y dim value ).

When using both the X dim and Y dim settings together, we can define a 2D plane which distributes the particle start at random points on this plane. Below you can see the example use of these settings.



To rotate the Emitter grab the rotation disk on the controller panel and apply the desired rotation around x, y and z asks.





**EM Y pos** : This joystick moves the emitter on its Y axis.

**EM X Z pos** : This joystick moves the emitter on its X and Z axis.

As you see, there are 6 switch buttons on the top line of the EM control panel. Let's go through their functions.

**Eon**: Starts and stops the emitter activity.

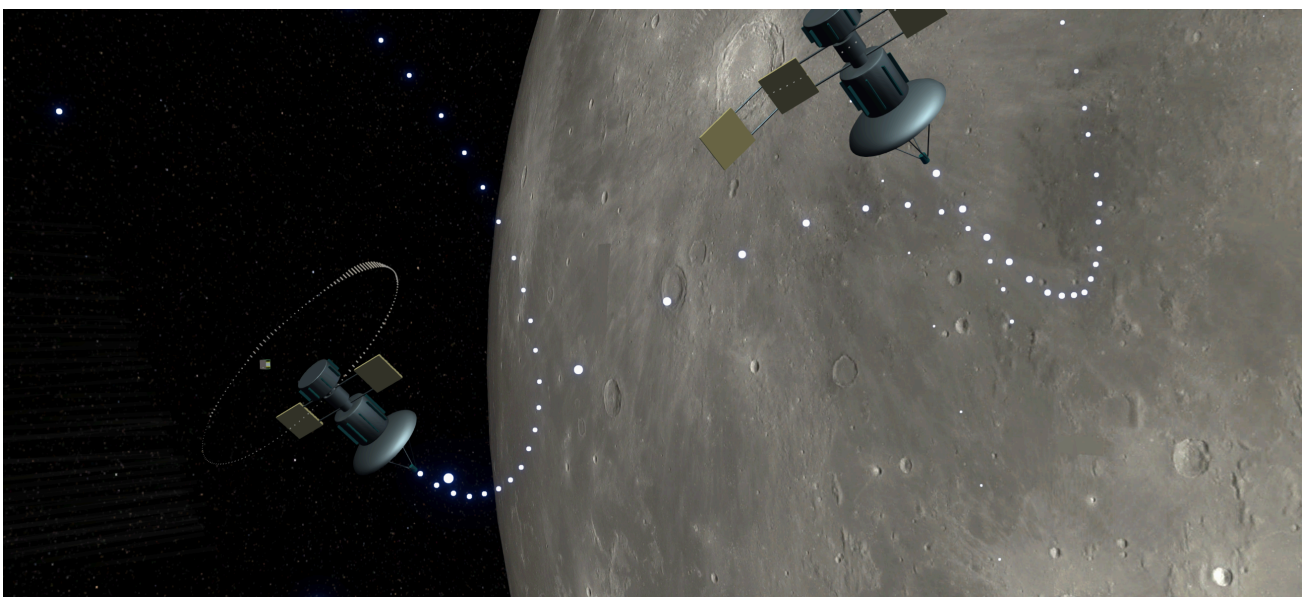
**ARotX** : Continuously rotates the emitter around its x-axis. The speed of this rotation will be set now with *Rot X* slider.

**ARotY** : Continuously rotates the emitter around its y-axis. The speed of this rotation will be set now with *Rot Y* slider.

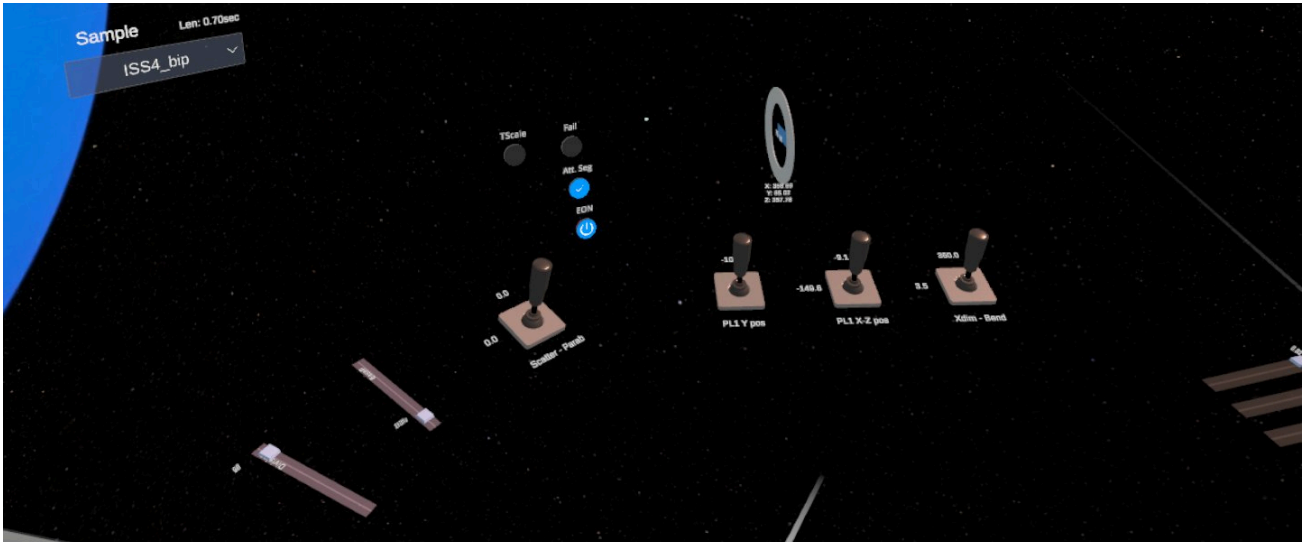
**ARotZ** : Continuously rotates the emitter around its z-axis. The speed of this rotation will be set now with *Rot Z* slider.

**Bounce**: When turned on, the particles will bounce on the surface which they hit, otherwise they will terminate there.

**RotSync** : When turned on, the second emitter satellite will sync to the first emitter so that it becomes its twin. Beautiful motion choreography can be established on this mode and playing with the rotation dynamics.



## THE PLATFORM PANEL



A platform needs a sound sample content. In order to load your sample, click on the pop-up menu on the left and a full list of the StarWaves sample folder will be ready to choose from.

By using the [Android File Transfer Manager](#) software, you can easily put your custom samples inside that folder, and only mono **.wav** files format is supported ( ensuring cross-platform compatibility ) Please use sample files with less then 22sec. duration.

When you load a sample successfully, its name and length will be updated. Also you will see the platform visual updated projecting your sample data.

**Tuning Scale :** The Emitters perform the sample content of each platform by using particles or beams scanning over the platform surface. We can apply the pitch modulation by using the Y-axis of the platform, so depending where a particle or beam hit the platform surface along the Y-axis , the pitch can change proportionally.

By activating the “TScale” button, we can also choose a tuning scale to quantize these pitch values so that the distributed pitch values can be part of a chosen scale on its pop-up menu. The relevant tuning scale can be selected on the pop-up menu panel.



**Base Pitch Slider** : Here we define in the base frequency value of the sample. For instance if it is a piano sample on middle A key, then the frequency would be 440hz. As soon as you change this value, a new scale distribution will be calculated by taking this new base freq. value.

**Division** : This defines the number of blocks which represent the sample data on the platform. As explained at the beginning section of the manual, each block holds a portion of sample data with a duration and specific starting and ending times on the sample waveform time line. The more blocks, the shorter their duration. The visual on the platform / each block represents the average RMS value of this block sample data. When a particle or laser beam hits a block, it will get triggered and play its content as a synth voice in accordance with the settings of the emitter sound engine.

**Scatter** : This setting will distort the platform by changing each blocks rotation and position randomly. ( relevant joystick x-axis value ) The value defines the degree of this distortion.

**Parabolic** : This setting bends the platform with a parabolic function. Likewise it behaves like a lens when used with emitter beams and is a lot of fun to experiment with. ( relevant joystick y-axis value ),

**Rotation disc** : By grabbing and rotating this disc, you can rotate the platform around the x,y and z axis.

**X dim** : The value of this joystick streches the length of the platform. ( relevant joystick x-axis value )

**Bend** : This spins each block along the x-axis of the platform gradually depending the value of this slider. ( relevant joystick y-axis value )

Now about the 3 switch buttons on the top-left of the Platform control panel;

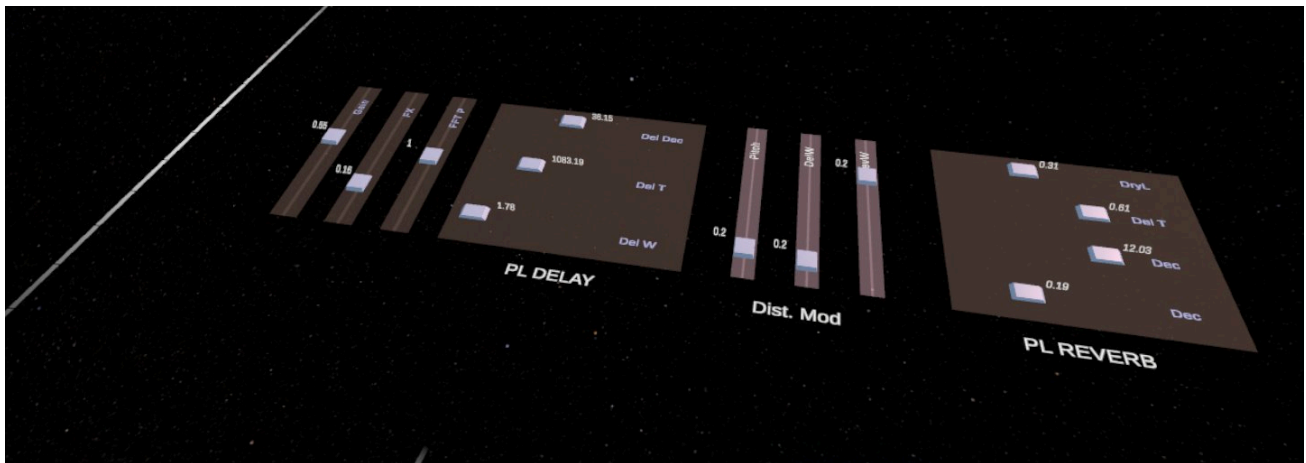
**EOn** : This makes the platform visible or not on the scene view.

**Att Seg** : This changes the block segmentation from linear to attack sensitive mode. Likewise, the sound sample of the platform will be analyzed and segmented according the attack content. This is great for using rhythmic groove-style sonic content where platform block start and stop points can be matched at beat points by using this mode.

**Fail**: If you think the platform blocks as light bulbs, this mode applies a stochastic fail event for them. They will disappear and come back like a flickering light bulb. Each time you turn off / on this mode, a different distribution of the effected platform blocks will be applied.

The sound objects of each emitter can send its audio output to Platform FX section for further processing. Each of the 4 platforms on StarWaves offer their dedicated effects section. For example, an emitter can distribute in space its particles and they can hit all 4 platforms at different times. Each platform block they hit will be processed by that platform's effect section.

You can see the Platform FX on the right side of the Platform control panel. It consists of dedicated number of sliders.



The **Gain** and **FX** vertical sliders do control the dry signal out and processed FX signal level of the relevant platform effects section.

The platform effects are the following and established in a serial order, and each effect has its dedicated parameters.

*Pitch Shifter -> Delay processor -> Reverb processor.*

It is designed by keeping in mind adding beautiful shimmer effects on the platforms of StarWaves.

The **FFT Pitch** sliders control the pitch shift amount. From the Pitch shifter output the signal goes to the delay effect. The **DelTime**, **Decay** and **WetMix** sliders are dedicated to the Delay processor. They set the delay time, the delay decay time ( feedback ) and the wet delay signal level of this delay effect.

From the delay processor, the signal goes to the Reverb processor. The **DryL**, **RevL**, **Dec** and **Room** sliders are dedicated to the Reverb processor. They control the dry level entering to the reverb, the reverb level of reverb processor, the decay time of the reverb effect and the room reverb density of the reverb effect.

All these settings are unique to each StarWaves platform and you can morph these parameters between StarWaves scene states.

Like on the sound object engine, we can map a kinematic parameter for modulating these effect parameters as you see on the Dist. Mod slider set.

The feature we do focus here is the distance between the emitter and the platform which it's hitting with its particles or beams.

Each slider on the Dist. Mod panel will set the depth of this mapping for each of the parameters **Pitch**, **DelW**, **RevW**. They are the Pitch shift amount, delay wet signal amount and the reverb signal amount.

For an example, the distance between the platforms and the emitters can increase the reverb signal amount , when you push the distance slider for the **RevW**. This will wash out the distant hit events.



## THE SOUND OBJECT PANEL



Each emitter has its own sound object panel as their synth engines are independent.

On the upper part of the panel, you can choose the emitter behavior. As mentioned before the emitter can send particles ( of which parameters are set mainly on the Emitter Panel itself ) or the emitter can produce beams. Whichever you choose, it will share the same signal processing chain.

When a particle or beam hits a platform block, it triggers a sonic event which starts playing the sample data hold by that block and its duration is defined by the **Grain duration** of the Synth Panel. It can be in the range from 1ms to a second.



Additionally you can randomize this duration with the **Rand dur** value and also randomize the event amplitude with the **Rand amp** value with the relevant joystick lever.

**Gain** : This controls the gain of the sonic event.

**AD panel** : This applies to an attack ( fade in ) of a gain envelope a decay ( fade out ) as of a gain envelope. The values are related to time. ( The beam mode has a fixed attack and decay internal setting. )

**Filter panel** : There you can use the relevant sliders to set the cutoff frequency of the filter and its resonance value.

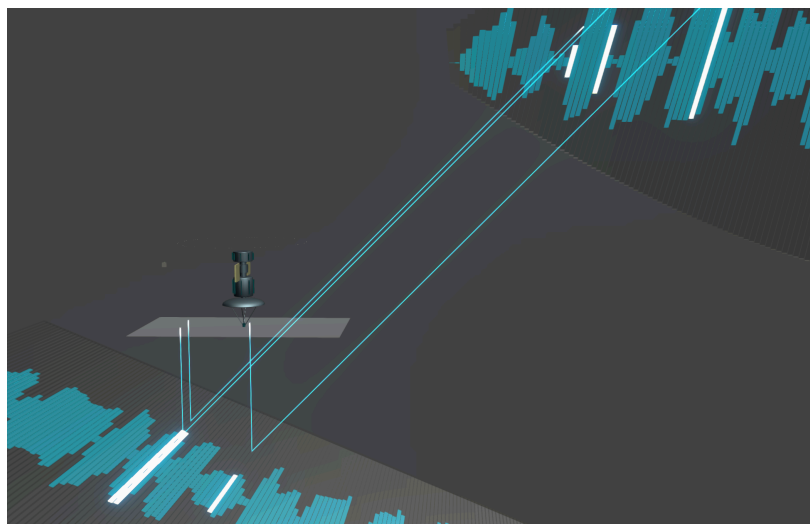
**Lopass / Hipass** : With that button we switch to low pass or high pass filter behavior.

When the emitter is on Beam mode, you will see the additional beam activation switches on the upper left part. As mentioned before, there can be 3 Beams sent at the same time in parallel from an emitter. The switches **B1**, **B2** and **B3** do activate those beams.

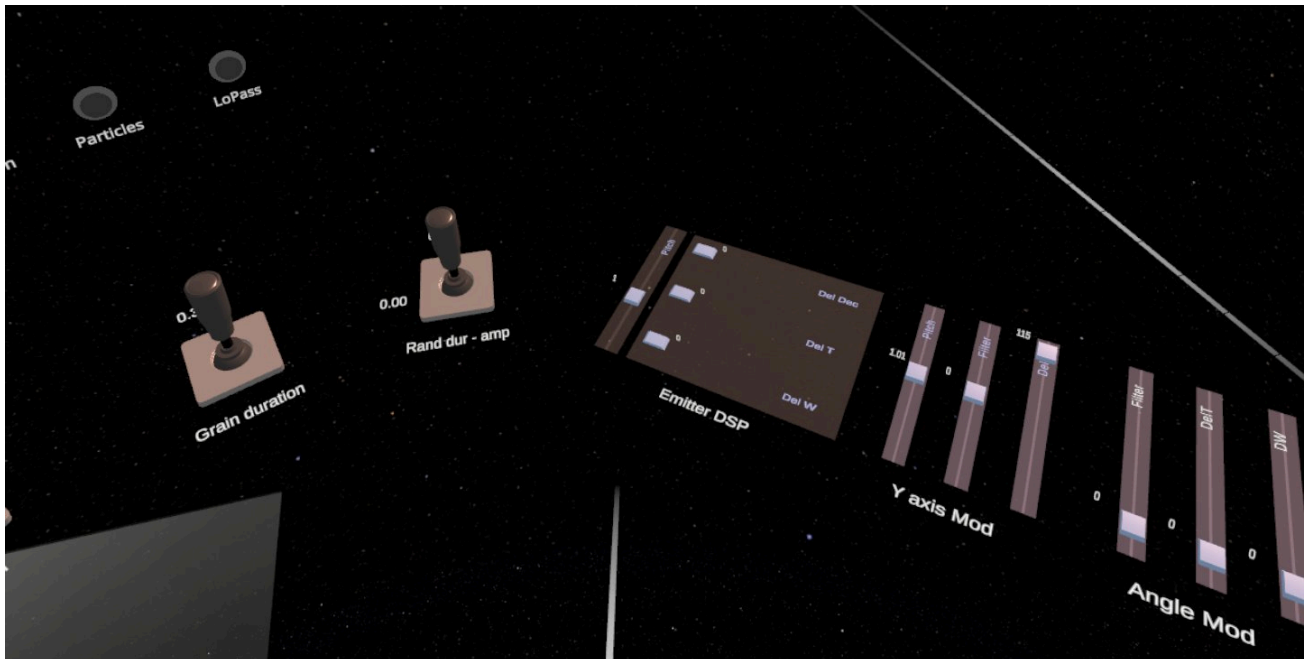
**Beam pos. panel** : On this panel you can set each active beam position relative to the emitter panel dimensions.

The **B. Move** joystick can define the degree of randomness of the beam position change among the emitter panel X / Y dimension ranges. Hence, the beams will continuously move and reveal a fantastic motion element.

Below you see an example case where an emitter is sending 3 beams, they are each hitting relevant platform blocks underneath and then also do bounce with an angle and hitting the other platform blocks on the top.



On the right side of the Sound Object panel you will see a set of dedicated sliders. First set is the Emitter DSP which applies a mono-delay processor to the signal.



**DelT :** This sets the delay time for a mono delay processing which is part of the emitter synthesis engine and applied to all triggered sound events by the emitter. Very low settings of the delay time with proper decay setting can create comb filter effects.

**DelW :** This is the wet balance for the delay effect.

**Dec :** This is the delay decay setting. Higher values will cause more feedback.

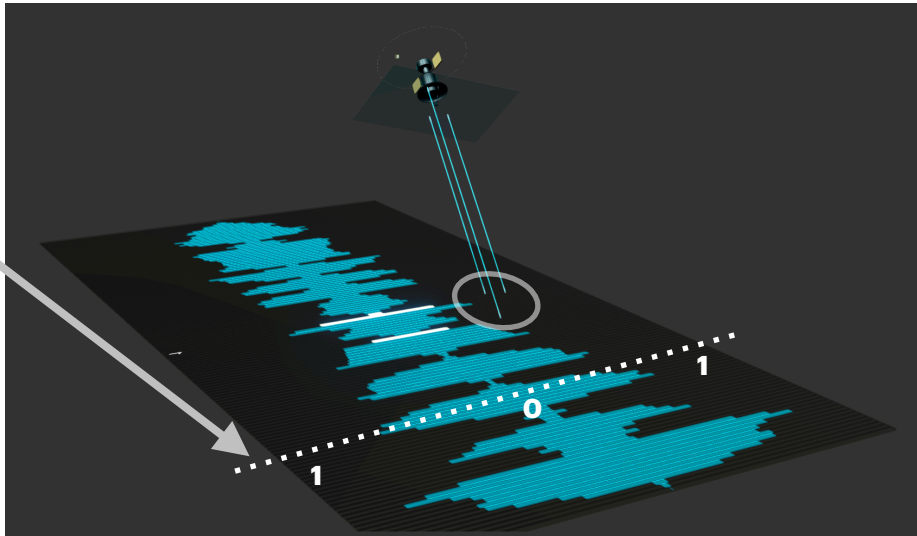
This next set of sliders offer us a new paradigm in modulation design for sound synthesis, by extracting the kinematic aspects of the StarWaves scene and map them to signal processing tools. These kinematic features are derived solely from the interaction of the emitter and the platform which it is hitting.

We do focus on two kinematic features of the emitter and platform interaction.

1. The Y-axis position of the hitting particle / beam on the platform surface.
2. The hit angle between the particle / beam and the surface.

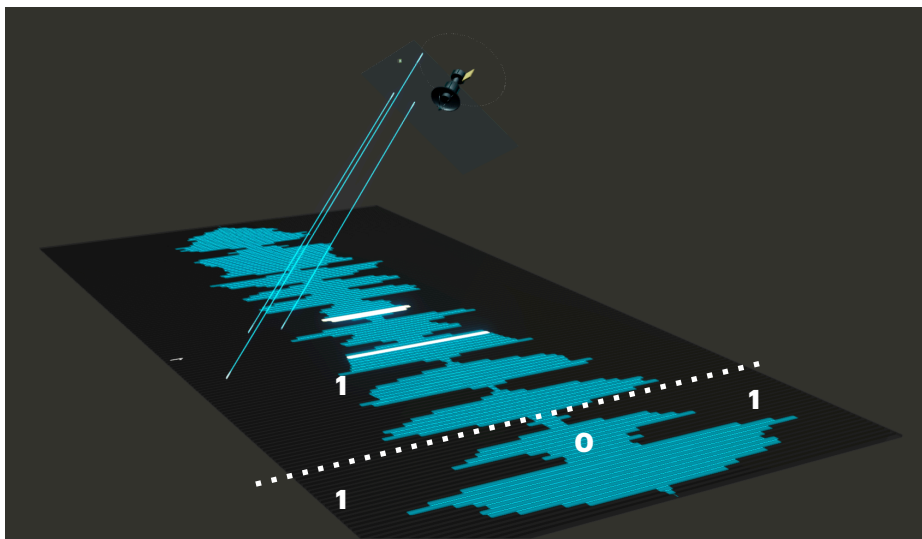
Below, the relevant shots will clarify these aspects.

The Y-axis of the platform



On the scene shot above, an emitter is rotating and entering the platform area and hitting it with 3 beams, and as you see the relevant blocks are lit up. Also the y-axis of this platform has been drawn so that you can see it clearly.

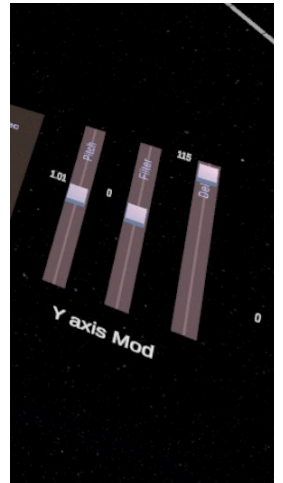
We have designed the interaction so that when the hit happens towards the center of the y-axis of the platform, it will create minimum effect but towards the top and bottom edges of the platform it will produce the maximum effect. And above, the emitter beams hit the platform surface near to the top edge so that the modulation value will be high.



Since the emitter rotation is continuous, the beams will traverse along the Y-axis and on the shot above we see that one beam is close to the center and another one is closer to the edge producing different modulation effects.

To map this Y-axis hit modulation, you can set with the **Y-axis Mod** for of the following sonic processing destinations; Pitch, Filter Cutoff and Delay Time.

For instance consider the Pitch slider on the **Yaxis Mod** panel, which will serve to set the modulation depth effecting the *pitch* parameter. When it is at the center, there will be no modulation. When the slider is set to zero, then when the particles / beams hit the surface close to the edges , the pitch will decrease one octave.

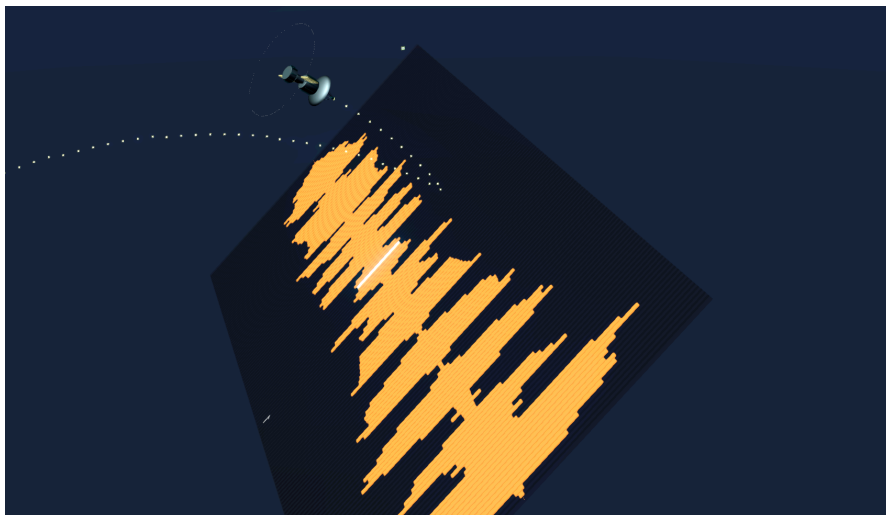


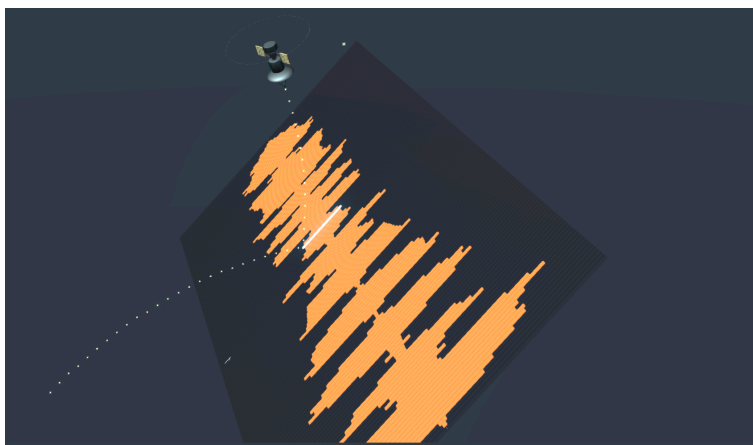
And when they hit close to the center of the platform, the pitch value won't change. Alternately, if you set the slider value at maximum, the pitch value will be one octave higher.

All these pitch changes will happen continuously depending on the hit position along the y-axis of the platform.

A similar logic works for the Filter Cutoff and Delay Time sliders as well. You can modulate these destinations equally with the particle/beam hit dynamics along the Y-axis of the platform surface.

The second kinematic feature we do use on the sound object engine is the hit angle between the particle / beam and the surface. On the below example, you see that the particles are hitting the surface nearly with 90 degrees.



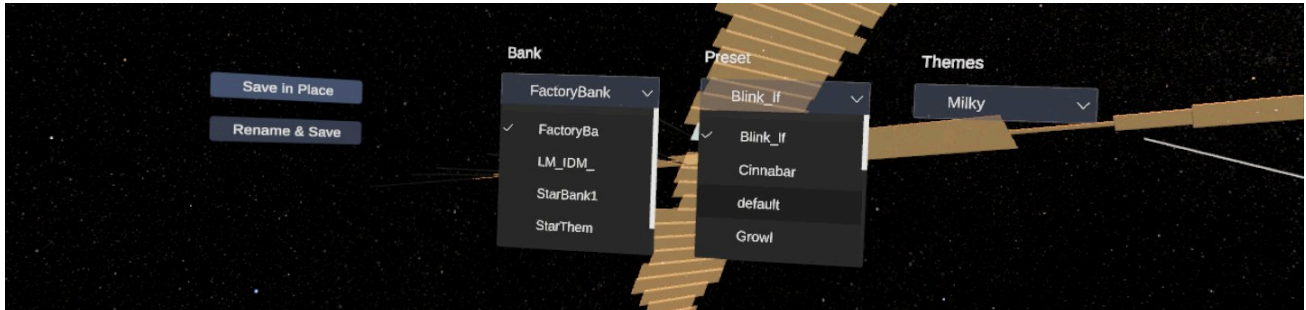


And here the particles are hitting the surface with angle close to 45 degrees. The less the angle, the more the modulation effect will be.

In order to map the angle modulation, you can use the Angle Mod slider set for of the following sonic processing destinations : Filter Cutoff, Delay Time, Wet balance for the Delay effect.

StarWave scenes can exhibit a complex and continuous kinematic interaction. One event can trigger other happenings as a network of connected events. The above modulation mapping will help to perceive these dynamics as sonic transformations.

## THE PRESET PANEL



The preset panel can be reached directly by pressing the *Preset* button on the main navigation panel.

StarWaves presets are grouped in banks and each bank contains 12 presets. Banks are merely folders in your StarWaves directory. You can re-organize them at your will by accessing them with the Android File Transfer Utility. Each preset comes with 5 scene states stored on the sequence panel. They are scene variations and can be called by pressing the relevant state recall buttons on your hand menu.

In order to load a preset, first select a bank by using the dropdown menu. As soon as you select a bank, its preset list will be uploaded to the preset dropdown menu on its right. Here you can select a preset, and its content will be loaded automatically.

If you just want to re-save the last loaded preset with your edit, you can use the **Save In Place** button.

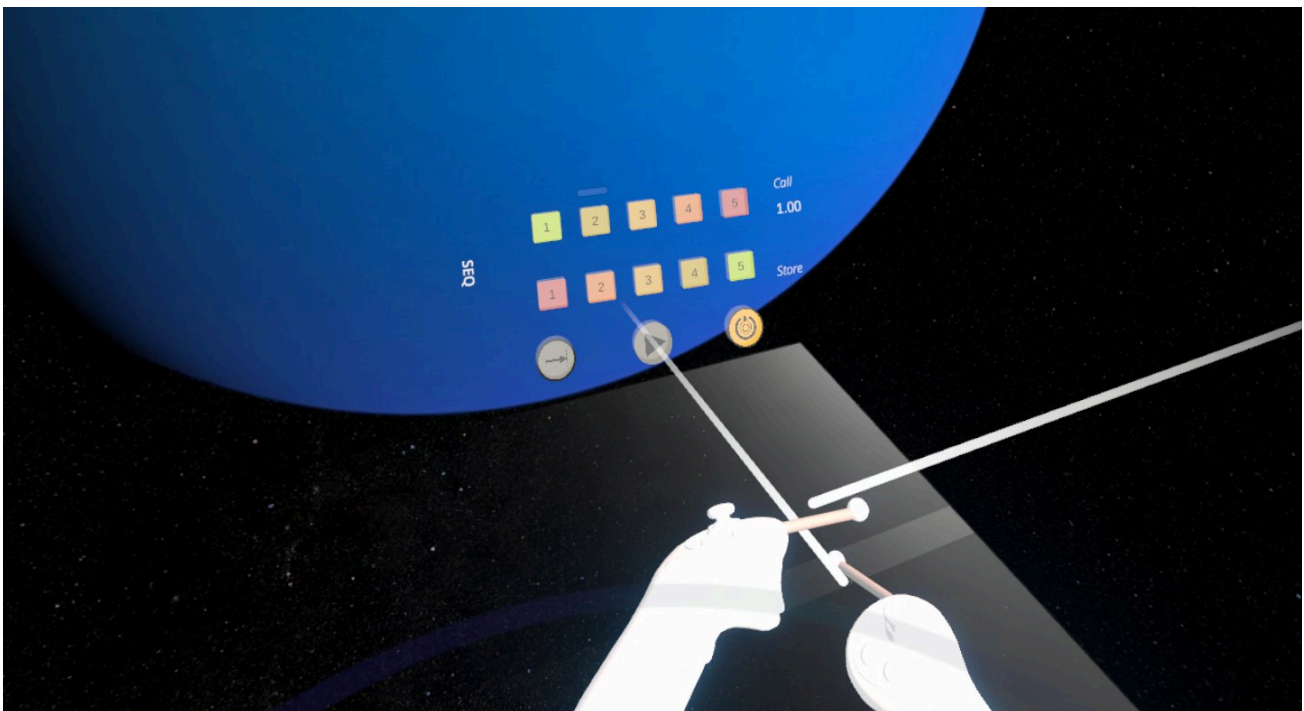
If you want to give an another name to your preset and save it likewise, you the Rename & Save button, which will open a dedicated keyboard pad to type in the name of the preset.

StarWaves VR comes by default with 10 scene **Themes** which are meticulously crafted theme designs.

The Theme park pop-up menu can be reached on the Presets panel .You can choose your preferred theme for your preset ( this choice will be saved within the preset ).

## THE HAND MENU

The hand menu panel can be displayed any time by bending your elbow towards your right side and your palm a bit upwards. Here you can store all scene parameters / panel parameters as individual states and call these states back with smooth transitions.



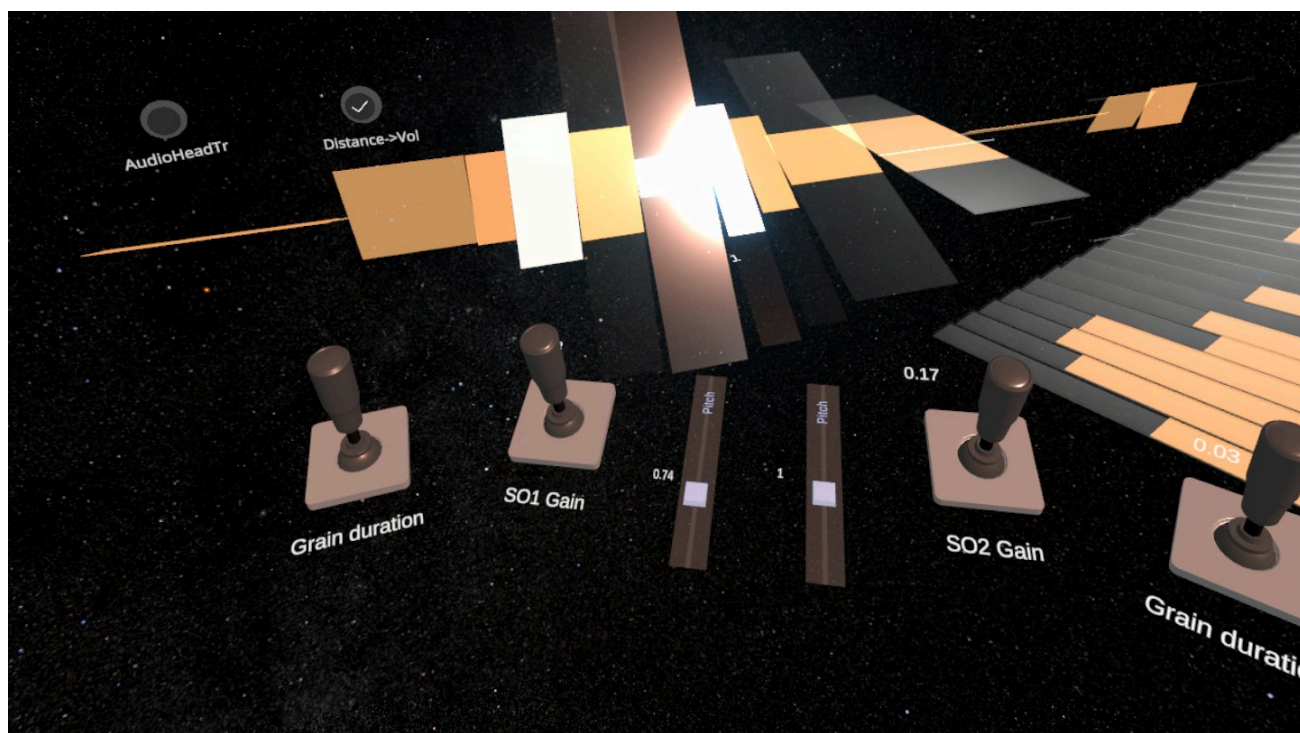
StarWaves VR offers 5 state slots where you can save 5 unique states of the scene. There are 5 numbered buttons dedicated to each of them, you can use the ones on the upper row to call back the saved states. And In order to store a scene state, just poke click on a button on the bottom row.

In order to perform all state transition automatically we can use the Play button. When active, it will be on the **play mode** and when not, it will remain on the **pause mode**. This will perform a defined state sequence. This can be a one shot performance or a looping one depending the state of the Loop button ( which is on the left ).

All these scene states and sequence settings are part of a StarWaves preset.



## THE MACRO PANEL



StarWaves VR presents a macro panel, where you can find easy access to some synthesis parameters.

Also you will see on the upper left part two switch buttons.

AudioHeadTR : When turned on, head-tracking will apply to the spatialization of the sounds.

Distance->VOL : When turned on, the amplitude of each sound source ( platform block ) will also depend on your distance to them. The further you are, the less the amplitude.

## ABOUT THE PRESETS OF STARWAVES

Here we present some comments of **Laurent Mialon** about his preset creations on StarWaves.

### FACTORY BANK

#### Blink\_If

Two rather rhythmical samples here have been loaded using the **Att Segment** feature, which divides the platforms according to their transients (vs. equally). Since they were not classic drum loops, during their making of, a transient enhancer has been used (many DAWs have such a tool, using a compressor with a slow attack can also help). A noise gate also can be useful, to further define the sample parts to be isolated by Starwaves.

When switching back and forth from the first to the second state, you can hear some drone produced by the second emitter synth engine. Its beam, unlike the other emitter's, is not moving, which contributes to giving a steady drone-ish sound. It is only audible during the state 1 to state 2 transition, as its volume is turned down for the first snapshot (yet you can see the trapezoidal shape made by the reflecting beam).

Even with non rotating emitters, the combination of a randomized area for the EM1 beam with some oddly placed platforms can produce some generative soundscape, with endless changes.

From state 3, the emitters are rotating, unveiling sounds that were not audible until then.

#### Cinnabar

This preset was inspired by the sulfide mineral which is the most common source ore for elemental mercury.

By morphing between states which have different delay rates, these being often modulated by the **Yaxis** and the **Angle** settings, it's possible to achieve some sounds that feel at the same time metallic and liquid. It can be interesting to try out long sequences of snapshots with short morphing times to come up with organic structures.

#### Growl

As for all the other presets, it's worth exploring the states one by one, as well as playing them sequentially (by pressing **Play** and even **Loop** - since both emitters will randomly scan the platforms, the probability to have the same repeating sequence is close to null).

One little trick here to create pseudo exponential rhythms was to have a beam hitting the same segment, with short duration sounds on one state, and increase that value for a nearby snapshot.

## Her Majesty's Sheep

A minimal preset, which is better played one state after the other, with manual changing of the state (**shift + state no**, if you are in full screen mode, double-click anywhere on the scene screen for this). By reversing the platforms, tilting or rotating them alongside their x axis, modifying the pitch shift amount between states by meaningful intervals, it is possible to get different atmospheres, which played slowly one after the other will tell a full comprehensive story. It's good practise by the way to load samples that are within the same scale, though experimenting is totally worth it too.

### In Pace

This preset is also better played slowly, one snapshot (state) after the other.

Let's focus on the first state to showcase a little trick. The camera on that snapshot is focused on the only audible events (others are coming in the other states, and are visually shown). When the beam touches platform 1 for the second time (after it's been reflected by P2 on top), it will touch the platform at a different (non perpendicular) angle than initially. With the **Angle** to Frequency slider pushed to the max, with a pretty high resonance, you can get some nice overtones - which are also randomized, as the beam will hit the platform surface on a random area (within the limits defined by emitter's **X dim**, in conjunction with the effective length of the platform, defined by the platform's X dim). It's thus possible, with a minimal set-up to make never-ending melodies.

## Iskander

For the first two states, the emitters are static, sending their particles on a chosen segment. It can be a visually rewarding experience to then play with the particles view, with some pseudo stroboscopic effects.

You can also note, that for the second preset, the original sound of P1 has been enriched with the platform's reverb, harmonically **Pitch** shifted. It's worth trying out different FFT sizes every time you use the pitch shifter.

From the third state, the emitters are rotating. In state 4, some particles are bouncing whilst the slowest rate emitter's aren't : the same sounds are being scanned in parallel with a slight delay related to the emitters relative position to each other's, but these synched emitters are doing way more than a simple echo, it's in turn like layering one sound with a totally different texture - which happens nevertheless to be directly related to the platform's content, initially unveiled by the first emitter to hit it.

## Oxycomptine

A rhythmic preset, which basically consists of three platforms hit by the same emitter, whilst the second emitter is playing the background pad.

The colour of this last sonic event is changing from one preset to the other, thanks to notably some **Pitch** shifter / **FFTsize** adjustments.

Also, to give the impression the pad is not static, the particles hit the platform in different spots on both axis, the **Yaxis** being a modulator routed to the platform's LPF (alongside the angle).

As far as the other discrete events (vs. the background pad) are concerned, the variety between states is a consequence of the platforms' position, enabling the particles to bounce on specific areas of them, at specific times - yet with a randomization factor, related to the area chosen for the particles emission, alongside with the degree of randomness per axis (**Mov** slider).

## Station F2

Only two emitters and two platforms for this preset.

The 3 beams of EM2 are scanning the second platform, where a sound made of the 4 microtonal (Bohlen-Pierce scale) distinct tones of plucked synth has been loaded. Sometimes you will hear a 2-notes loops, sometimes a 3-notes one, and eventually the full loop will play.

In the background a highly modulated sample made on Novation Bass Station 2 is being scanned slowly, for some classic granularization textures. The richer the texture of the loaded sample, the more variety you will eventually get once the platform is played within StarWaves.

When they are meant to overlap it can be recommended to load sounds with a different frequency spectra on different platforms, so that their frequencies stay audible.

Please note, notably on State 3, the comb effects on the Bass Station sample, here achieved with a short delay time on the EM1 synth engine, the modulation being provided by both the **Angle** and **Yaxis** position.

## STAR BANK 1

### AlaFX

This is a variation of the preset named Oxycomptine, present on the Factory Bank

### Being Palmer Eldritch

Here, 3 platforms have sounds with defined transients, hence the choice to use the **Att Segment** dividing mode. They are all together triggered by the same slow rate Emitter, whilst the tonal part is provided by a Tibetan bell loaded on P4.

Things get more complex from state 4. Putting the platforms closer to each other's and using the **Scatter**, **RotAnim** and **Parabolic** function establishes a topology that allows the formation of reticular sounding patterns that always change. By adjusting the width of **X dim**, we can effectively control the amount of randomness.

Be sure to try out slow and fast morphing times when you play the sequences.

## CDG Nord

A simple setting that works well by default is to have the platforms in vis-à-vis, with rotating emitters reading their content - one emitter for each pair, emitters being 90° apart.

Between states 2 and 3, the division of P3 and P4 has been increased, transforming a discrete arpeggio into a more continuous sound (still with transients actually, but closer to what we could refer as a single iterative sonic object). Using **RotAnim** alongside the assignation of **Yaxis** to the sound engine's delay time combined with the modulation of the LPF cutoff by the **Angle** of collision transforms what could be a static iterative sound object into a more organic iridescent, swell.

## Cinq Cents

This preset was conceived thinking of what StarWaves could also be about (the fascinating relationship between the void and silence, space and sound), with arguably Pascal's most famous quote (« le silence éternel de ces espaces infinis m'effraie ») loaded on P1.

Let's jump to state 5, and focus on that classic psychedelic effect on the voice : as the synth engine delay time is increasing the perceived pitch of the sample is, as both the platform **Pitch** and EM1 **Delay** time have a positive modulation originating from the same source, **Yaxis**.

Let's go back to snapshot 4, and consider one of the advantages of using a twisted platform. Besides some possible variations in color (in the event the position on the Y axis and the angle of the collision are used as modulation sources), we can use the odd shape in conjunction with a scanning beam to have irregular reflections on further platforms. Here, what's being reflected on P3 reaches P4. Try changing P4's **Parabolic**, **RotAnim** and **X Dim** parameters to observe how the beams originating with EM2 end up on either platform.

P1 is being initially scanned by the other emitter; the occasional playback that is a consequence of the multiple reflections of the EM2 beams contributes to the repetitive hypnotic effect.

## Crevasses

Let's focus on the action of EM1 (rotating alongside the vertical axis) in relation to platforms 1 and 2 (the horizontal ones).

Recall state 1. Both platforms have been divided in seven segments. Rare particles (it's a slow , irregular rate - note the high value of **S rand**) are triggering specific piano chords.

Now recall state 2. The platforms' segmentation has been pushed to the max, transforming the loud piano chords into accelerating or decelerating sequences of the same chords - the change in speed being controlled by the curvature of the platforms and distance related to the emitter. Also, by reversing the platform orientation, the beam will pick various amplitudes of the decaying chords, thus creating, depending on this orientation, either a crescendo either a decrescendo.

## Dub Huit

A simple preset, where a rhythmic loop is being read by the first emitter, whilst the second is responsible for playing a background pad. Now and then EM1 also triggers the playback of the other platforms' blocks.

This is particularly noticeable in state 4, where the twisted shape of platform 2 allows the occasional triggering of some of P4' blocks, creating some harmonically rich background.

## Kinzhal

A melancholic ambient preset, with little changes between the states. Please note though, the difference in tone that is a direct consequence of the FFTsize adjustments between states.

## Mandragora Blooming

One emitter is sending beams to 2 platforms, that have been segmented with **Att Segment**, given their nature, and more important, the sonic intention (remember that you can always edit the sounds to be loaded so that StarWaves reads a specific number of blocks, either accentuating transients, either making some perfect divisions)

## Morgen

A few things can be spotted here.

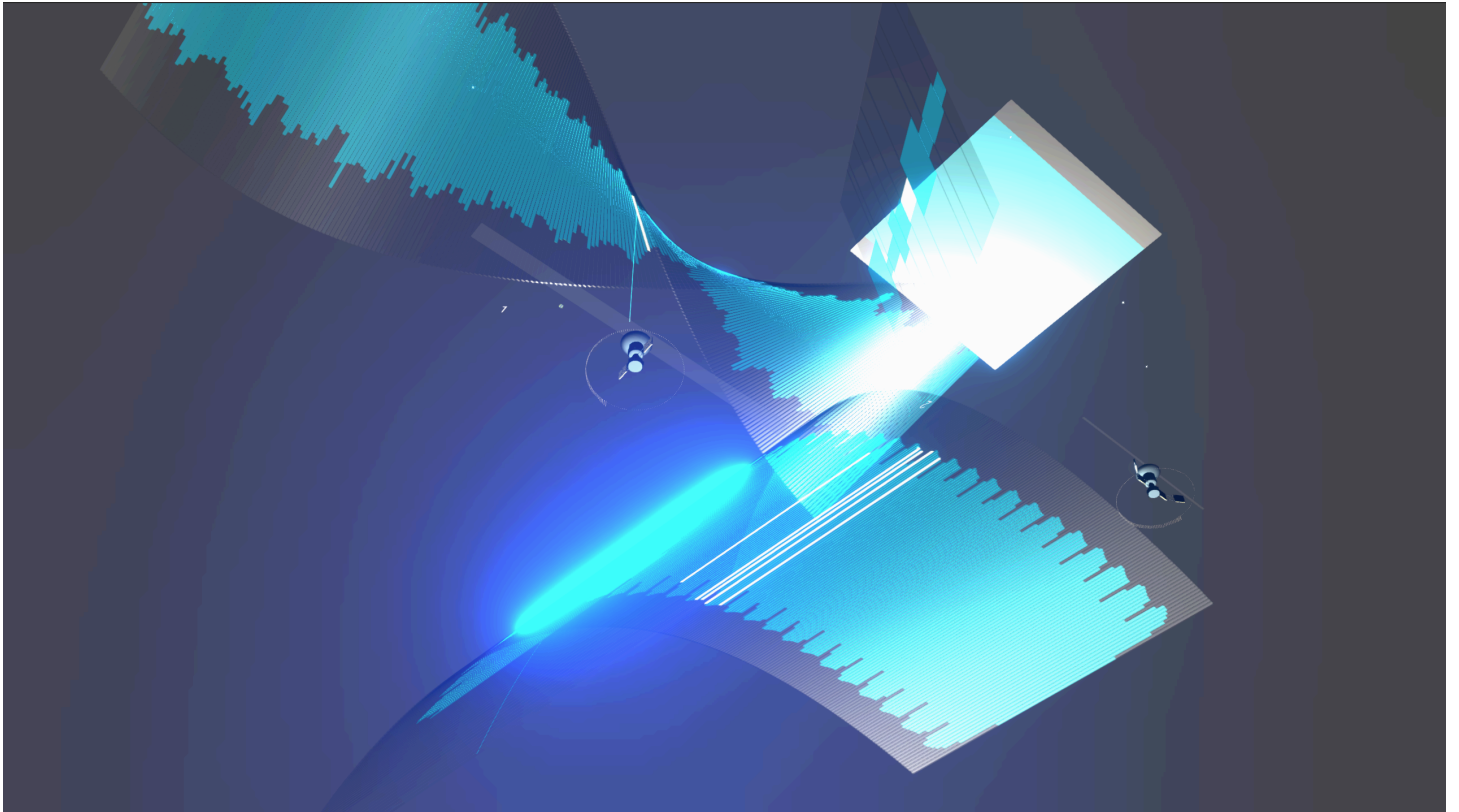
The State 1 to 2 transition : for EM1, check the sliders of **EM dur**, **EM rat**, **Em Spd**, **Y dim**. For the platforms' FX, check the following **PL1 FX** parameters : dry and wet gains, pitch shifter, FFT size. By changing multiple parameters at both the collided platform and the emitter synth engine levels, you can use StarWaves to create the most unusual transitions and with the added advantage of having a powerful visual feedback.

On State 5, the platform blocks have been fractured and made fitting the same small space, with the first emitter being in the core of that topological chaos. Since the emitter is rotating, with big values for both **X dim** and **Y dim**, you have now a totally self-generative abstract mini piece, which sounds totally different from the straight loop you could hear in states 2,3,4.

STARWAVES



Finally, thank you for using StarWaves. Independent of your profession, and what you are doing, we believe that it will spark some creativity for you and deliver valuable experience for the time you spend on it.



[WWW.SONICPLANET.COM](http://WWW.SONICPLANET.COM)