Project Title	REsearch INfrastructures FOR Citizens in Europe
Project Acronym	REINFORCE
Grant Agreement No.	872859
Start Date of Project	01.12.2019
Duration of Project	36 months
Project Website	www.reinforceeu.eu

REINFORCE Sonification Scripts Tutorial

Work Package	WP7 - Increasing the senses, increasing inclusion
Lead Author (Org)	B. Garcia (CONICET)
Contributing Author(s) (Org)	N. Bertaina (IBFI), J. Casado (ITDA, IBFI), P. Chanial (EGO), G. de la Vega (ITDA), W. Diaz-Merced (EGO), G. Hemming (EGO), C. Hettlage (SAAO), S. Katsanevas (EGO), F. Spagnuolo (EGO)
Due Date	31.05.2022
Date	31.05.2022
Version	1.0

Dissemination level	
X	PU: Public
	PP: Restricted to other programme participants (including the Commission)
	RE: Restricted to a group specified by the consortium (including the Commission)
	CO: Confidential, only for members of the consortium (including the Commission)

Document Control

Version	Author/ Reviewer	Status	Date
0.1	B. Garcia (CONICET, WP7 Leader), J. Casado (CONICET, WP7 member)	First version.	19 Jun, 2022

Table of contents

Terminology

Terminology/Acronym	Description
ADSR	Attack, Decay, Sustain and Release
API	Application Programming Interface
CNRS	Centre National De La Recherche Scientifique
CONICET	Consejo Nacional de Investigaciones Científicas y Técnicas
CSA	Coordination and Support Action
CSV	Comma-Separated Values
D0	Project Deliverable + associated number
DoA	Description of Action
EA	Ellinogermanik i Agogi Scholi Panagea
EC	European Commission
EGO	European Gravitational Observatory
ERN	European Researchers' Night
EU	European Union
F2F	Face-to-Face meeting
GA	General Assembly
GWOSC	Gravitational Wave Open Science Center
IASA	Institute of Accelerating Systems and Applications
IBFI	Instituto de Bioingeniería, Facultad de Ingeniería, Universidad de Mendoza
INFN	Istituto Nazionale di Fisica Nucleare
ITDA	Instituto de Tecnologías en Detección y Astropartículas (CNEA, CONICET, UNSAM)
KPI	Key Performance Indicator
LC	The Lisbon Council For Economic Competitiveness

	and Social Renewal
LRI	Large Research Infrastructure
M0	Project month + number
MIDI	Musical Instrument Digital Interface
OU	The Open University
PAB	Project Advisory Board
PDF	Portable Document Format
PNG	Portable Network Graphics
PO	Project Office
PTC	Project Technical Committee
QA	Quality Assurance
QUEST	Quick Unbiased and Efficient Statistical Tree
REA	Research Executive Agency
REINFORCE	REsearch Infrastructure FOR Citizens in Europe
RI	Research Infrastructures
SAAO	South African Astronomical Observatory
UNIPI	Università di Pisa
UOXF	University of Oxford
URI	Uniform Resource Identifier
WAV	Wave Audio File Format
WP	Work Package
WSGI	Web Server Gateway Interface
ZSI	Zentrum für Soziale Innovation

1 Introduction

This document describes the work done in relation to REINFORCE project. Each REINFORCE demonstrator project uses data in a format that is specific to each project. This means that, in terms of sonification, each project has to be treated with a specific, individual approach.

Under request, and exchange with the demonstrators WP teams, a script for each demonstrator was developed to sonify the specific data set, beyond and "ouside" the framework of the sonoUno software.

The data set which can be sonified up to now, with special scrips, are:

- Direct images of glitches, from the WP3;
- New Particle Search at CERN (LHC) data sonification from the WP5;
- Cosmic Muon images data set from the WP6.

2 Pre-requirements to all the sonification scripts

To be able to run the image sonification script it is important:

1. If you have sonoUno installed, continue with the specific script instruction; if not, continue with step 2.

2. If you only want to run REINFORCE scripts follow the next steps (if you also want to run the sonoUno framework, follow the instructions on github, see NOTE 2):
   - Check that you have python 3.x installed on your system running 'python3' or 'python' on a terminal. If you don't have python:
     - For Mac: `brew install python3`
     - For Ubuntu: `sudo apt install --upgrade python3`
     - For Windows download the installer from: https://www.python.org/ftp/python/3.8.1/python-3.8.1-amd64.exe

     During installation in Windows, you must "set the environment path" by clicking in the box that appears on the screen.

NOTE 1: from here we use python3, you have to use python (Windows) or python3 depending on the step before.

NOTE 2: if you also want to run the sonoUno framework, follow the instructions of sonoUno software available on GitHub README (https://github.com/sonoUnoTeam/sonoUno) or sonoUno installation manual. The difference here is that sonoUno framework needs the bridge with octave and octave GUI, and the REINFORCE scripts do not.

- Check that pip is installed with 'python3 -m pip -V'. If not:
    - For Mac, pip is installed with python installation.
    - For Ubuntu: `sudo apt install python3-pip`
    - For Windows, pip is installed with the executable.
- Install the libraries with pip:
    - `python3 -m pip install matplotlib numpy pandas pygame`
    - For Windows: `python -m pip install matplotlib numpy pandas pygame`

NOTE 3: this step not install wxPython and oct2py, because these libraries are a requirement of a sonoUno GUI. If you want to run sonoUno GUI instead of REINFORCE scripts follow the SonoUno installation instructive.

- If you want to download the software from the repository install git, if not you can download the zip folder from GitHub.

3 Image sonification

The sonification technique proposed is developed in Python, using the *openCV* library for image manipulation and the *sonoUno* sound library for the sonification process. Figure 1 shows a screenshot of the windows displayed by the image-sonification script and the cursor (the blue vertical line) that indicate the column being sonified as it moves along the x-axis of an image used in the GWitchHunters.

The sonification involves sonifying the intensity of the image column with the same pitch variation as the 2D plot in *sonoUno*. The brightness value (white) corresponds to the highest tone and the darkest value (black) to the lowest tone (silence).

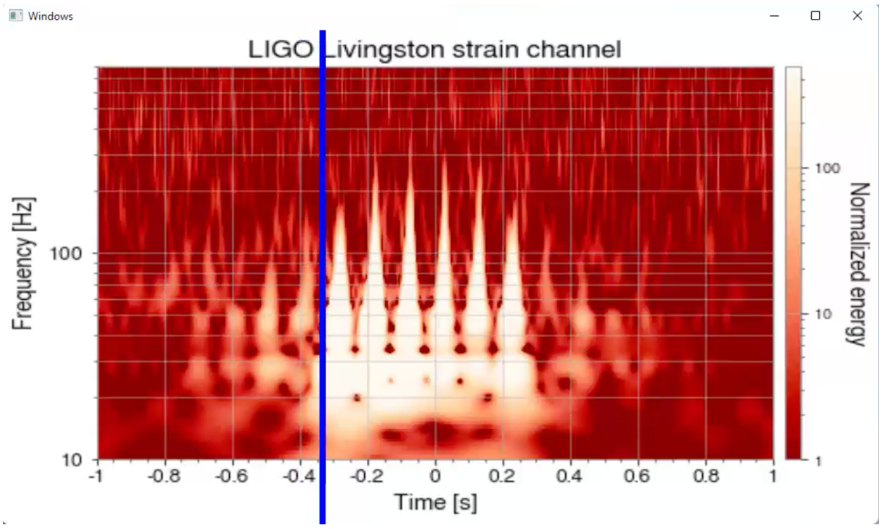

Figure 1 - The image-sonification script at work on an image of a glitch. The blue vertical line shows the position on the image along the x-axis for which the sound is being reproduced[1].

3.1 Additional pre-requirements

Install opencv library for python[2], with pip and the basic installation:

- `python3 -m pip install opencv-python`

Download the code from [sonoUno GitHub](), or clone the repository with git.

3.2 Running image sonification script

To run the image sonification script, it has to be called from the command window and it opens a window displaying the image and showing the position bar during the sonification (Figure 1). From bash, go to the sonoUno folder where the script is located and type:

➔ `python3 img_sonif.py -d "path_to_the_image_to_sonify"`

Once the window with the image appears, *to start the sonification press enter, with the cursor on the image.*

For example, in my computer:

[1] Visit the *sonoUno* gallery to see the video: https://www.sonouno.org.ar/glitch-1126409678-84375/.
[2] https://pypi.org/project/opencv-python/

- ```
  python3 img_sonif.py -d
  "/Users/sonounoteam/Downloads/blip_zooniverse example(to sonify).png"
  ```

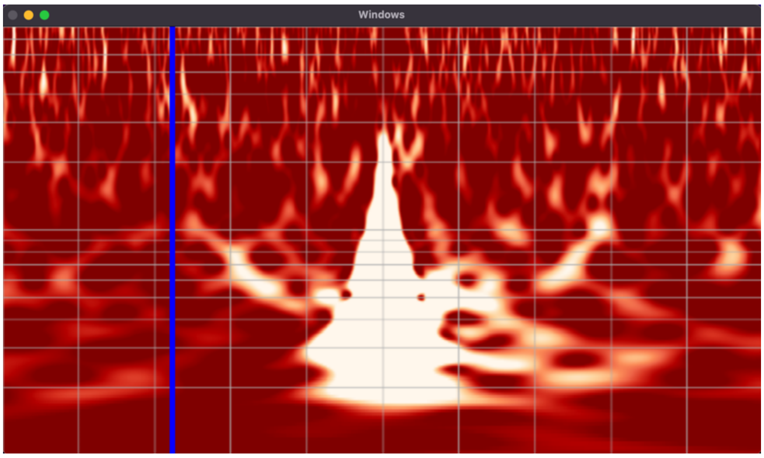

Figure 2 – The window displayed by the image sonification script during the image sonification.

About the command to run the script: the '-d' indicates the path of the image to be sonified. When the sonification display finished, the code saved the sound in the same folder as the image file with the same name, the label '_sound' and in wav format (for example: 'blip_zooniverse example(to sonify)_sound.wav').

4 New Particle Search at CERN data sonification

Sonification of the New Particle Search at CERN demonstrator data required the design of particular sonification parameters. The HYPATIA[3] event display is used to study events and to identify particle tracks and clusters (Figure 3). The sonoUno team began from Figure 3, to try to understand each particle representation and to try to show it with sound.

The particle types are muons (long path, cross all the detector layers), electrons (a track in the inner detector and deposit of energy on the calorimeter), photons (deposit of energy in the calorimeter without a track), converted photons (two close tracks in the inner detector and deposit of energy on the calorimeter) and unknown (only a track). A detailed explanation could be found in section 3.3.

3.1 Pre-requirements

You have two options here:

[3] https://hypatia-app.iasa.gr/Hypatia/
  ```

1. Download the code from sonoUno GitHub, or clone the repository with git.
2. From inside the sonoUno folder in GitHub (here), download the folder named ‘data_lhc’ and the scripts ‘lhc_bash.py’ and ‘lhc_display_bash.py’.

3.2 Running LHC sonification script

More than one event of the LHC could be stored in the same data file, the data file format inside the txt file would be:

```
Name of the event
some_extra_information
track_1 -> its data
track_2 -> its data
...
cluster_1 -> its data
cluster_2 -> its data
...
--------------------------------
Name of the next event
...
```

LHC sonification present two modes:

- a bash script (lhc_bash.py) that took a given data set and produce the images and sounds in the same location as the data file;
- a display script (lhc_display_bash.py), which takes a given data file and displays the sonification opening a new window, the instructions printed in the console must be followed.

3.2.1 Bash script

Inside the command window go to the sonoUno folder or the folder where the downloaded files are placed and type:

➔ `python3 lhc_bash.py -d "path_to_the_lhc_data_file"`

In bash will be displayed a message with the name of the event and one message by track being sonified; after the script finishes, the images and sounds files created will be in the same folder as the data file opened; the names of the files are composed by: the name of the file, the name of the event and the file extension (wav for sound and png for image).

About the command to run the script: the ‘-d’ indicates the path of the file to be opened.

An example in my computer:

➔ `python3 lhc_bash.py -d "C:\Users\Johanna\Documents\lhc\lhc_event.txt"`

The lhc folder where the data file is located has the next files created:
- lhc_event_1073767020.png
- lhc_event_1073767020.wav
- lhc_event_1273292777.png
- lhc_event_1273292777.wav

3.2.2 Display script

Inside the command window go to the sonoUno folder or the folder where the downloaded files are placed and type:

➔ `python3 lhc_display_bash.py -d "path_to_the_lhc_data_file"`

The script opens a new plot window where the tracks will be displayed during the reproduction of the sound. After the reproduction, the sound (wav) and images (png) are stored in the same folder as the data file. The names of the files are composed by: the name of the file, the name of the event and the file extension.

3 Detecting particles in the LHC

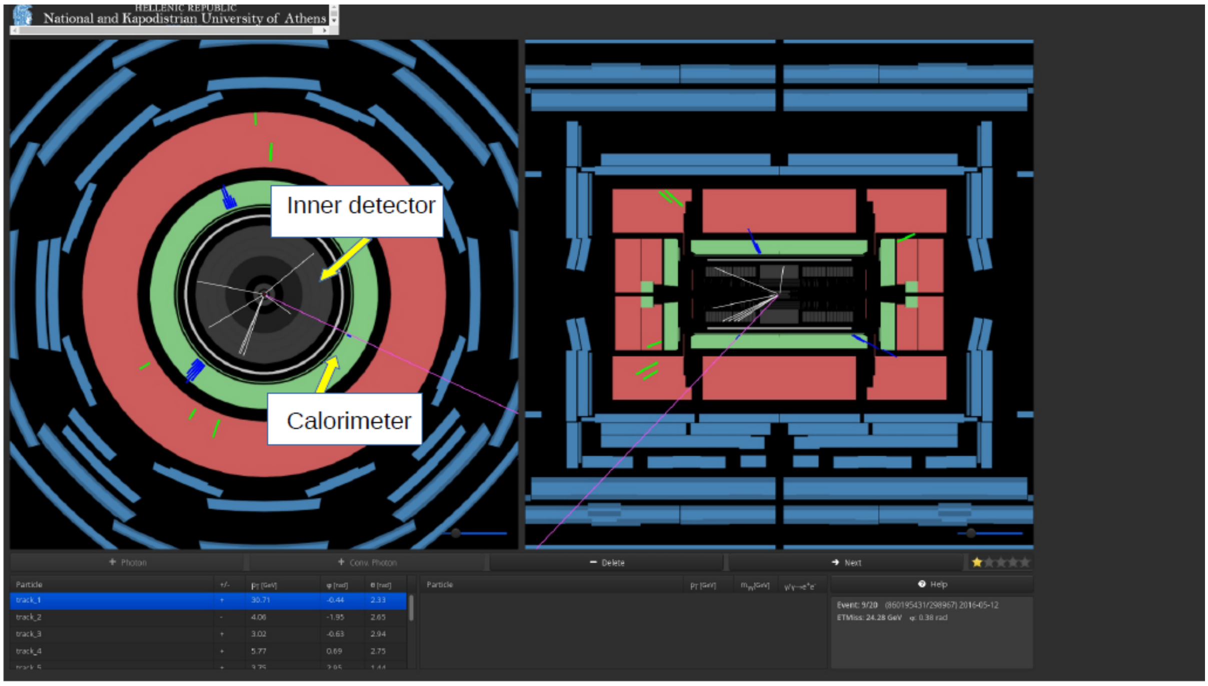

Figure 3 - The HYPATIA event display, showing 860195431 event. On the left, a transversal view; on the right, a longitudinal view, of the full detector; the particles in the event can be seen in each view [(from zooniverse)](#)

The following particles can be found in the detector and displayed through the HYPATIA program:

- an electron - represented by a track in the inner detector (the central grey area, in Fig 3) that points to a cluster in the calorimeter (green area, in Fig. 3);
- a converted photon - represented by two very close tracks in the inner detector, that points to a cluster in the calorimeter;
- a muon - represented by a long track that goes through all of the detector layers and which could, although this is not necessarily the case, be a point to a cluster;
- a photon - represented by a cluster in the calorimeter, but with no track in the inner detector;
- or an unknown particle- which is considered to be any other representation that is not covered by any of those above.

Each particle display is taken into consideration for sonorization. The first step was to extract the common characteristics between the different particle-type representations. These characteristics are:

- the presence or absence of the track in the inner detector;

- the presence or absence of the cluster in the calorimeter;
- and, specifically, considering whether the muon presents a long track.

The next question is how to produce a sound that represented each situation in sufficiently a general way as to be able to use it with each particle, but assuring that it is possible to discriminate between them. In this sense, several meetings between the LHC and sonoUno were necessary.

The final sonification proposal was to sonify each particle trajectory of an event, using the following sound configuration:

1. The center of the detector and the beginning of the particle track is indicated by a tick mark[4], which one is represented by a beep sound (this sound is present in each particle sonification).

2. The different tracks or the absence of track in the inner detector, are represented by:
    a. A continuous sound with a specific frequency (piano note 'D6', frequency 1174.66Hz) to represent each single track in the inner detector (duration of a simple track in the inner detector: 2s); in the case of a muon, the continuous sound is of longer duration, representing a longer track that exceeds the inner detector (duration of a muon total track sound: 4s).
    b. In the case of two tracks, two continuous sounds at two different frequencies are sonified: frequency 1 with piano note 'D6', 1174,66Hz; frequency 2 with piano note 'C6', 1046.50Hz. This represents the two closer tracks in the inner detector (duration of a simple track in the inner detector: 2s).
    c. In the case of the absence of a track, a silence of 2s is sonified.

3. The end of the inner detector and the beginning of the calorimeter is represented by a second tick mark, a sound of a short 'F7' piano note, of 2793.82Hz in frequency, with a duration 1ms.
    a. In the case of muons, this tick mark is sonified after 2 seconds of the beginning of the continuous sound, indicating that the muon passes from the inner detector to the calorimeter.

4. When a cluster exists in the calorimeter, a specific compilation of short sounds is reproduced to represent it, and its volume is related to the energy of the cluster (less energy-lower volume; more energy-higher volume).

[4] A tick-mark is a specific sound that add context to the sonification process, this sound represent something predefined during the sonification. For example, in this case, a bip sound represent the center of the calorimeter.

a. In the case of muons, the cluster is sonified during the track sound in the corresponding time (just after the tick mark which indicates the transition to the calorimeter).

The generated script opens a file with some events and the tracks and clusters associated with each event. It separates each event and then identifies the tracks and clusters within them, in order to plot and sonify the data. With the tracks and clusters identified, the script continues with the sonification and plot process, which can be displayed particle by particle[5] (Figure 4) or the script can be run in *bash*, saving the plots and sound files per event.

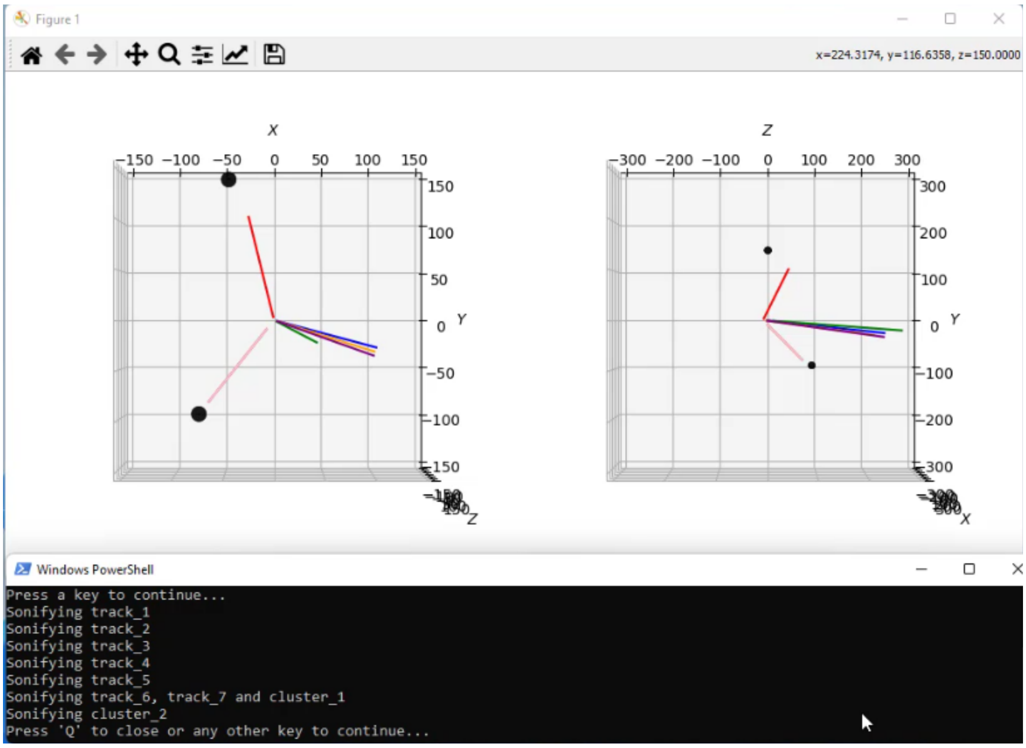

Figure 4 - The plot generated with the developed script for Event 326146241. On the left, a transversal view, and on the right, a longitudinal view, of the full detector. The clusters are represented by black circles. The sonification action-log is displayed in the area at the bottom.

Figure 5, below, provides a comparison of how an event displayed in HYPATIA appears when sonified in *sonoUno*.

[5] https://drive.google.com/file/d/1UrSdt6wqCehqcoeHILivnm8feHOt1vh7/view?usp=sharing

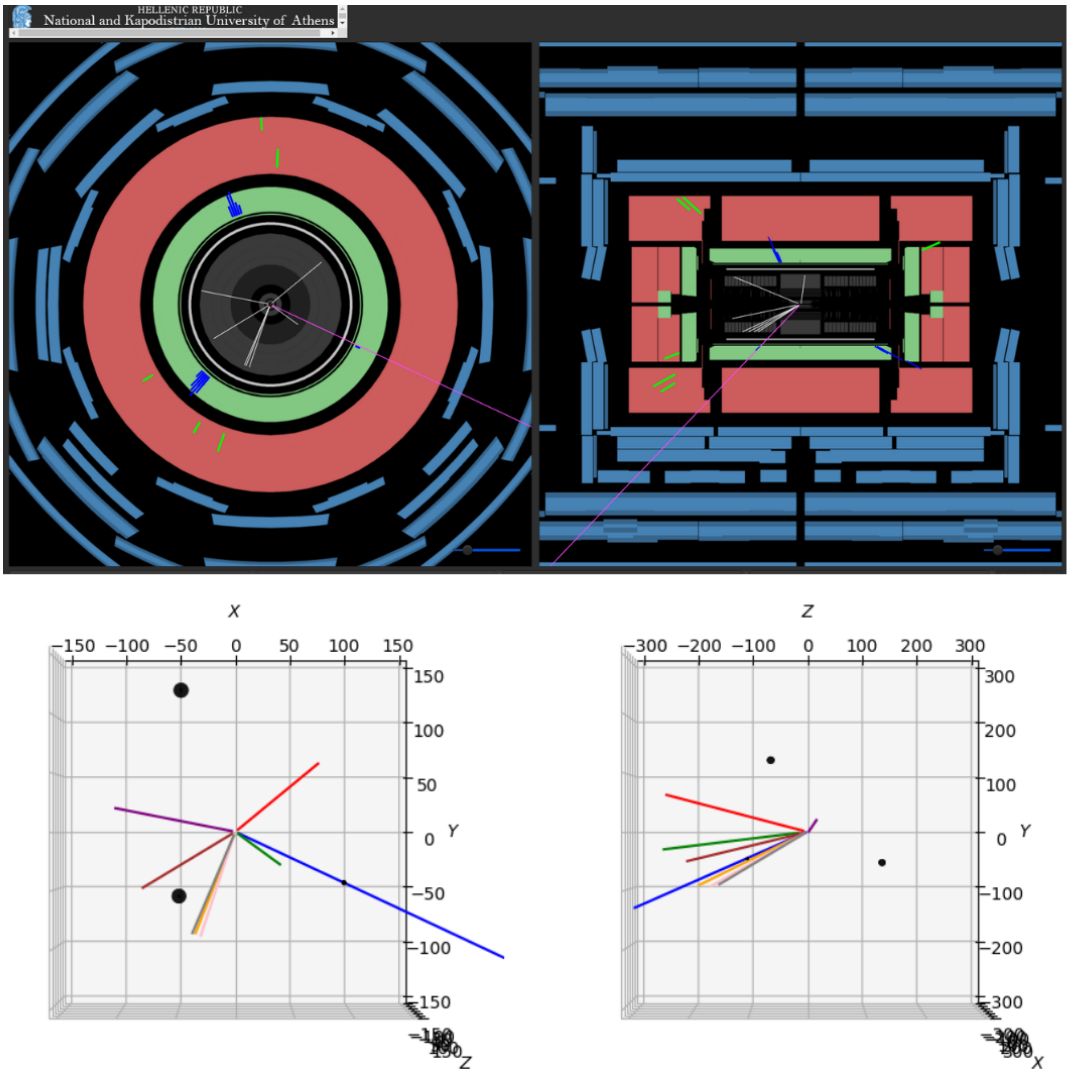

Figure 5 - A comparison of the event displayed in Figure 3 generated in HYPATIA, with its representation in sonoUno.

5 Cosmic Muon data sonification

The muon datasets used in the Cosmic Muon demonstrator project are similar in structure to the data used in the New Particle Search at CERN demonstrator. In the case of muon sonification, the objective is to be able to identify if there are any deposits of energy in the three layers of the detector (Figure 6) and to understand if these deposits of energy are aligned. If these two requirements are met, we are in the presence of a muon.

The sonification proposal here consists in sonifying the possible track of the muon by matching the deposits of energy. Assigning a note to each channel plotted in Figure 7 and sonifying the three layers of the detector one after another, if the sound is consistent and the notes goes up or down in frequency, a line could be drawn between each layer of the detector. This sonification is made from each view of the detector, so the line must exist in the two views (Figure 7 shows an example of muon presence).

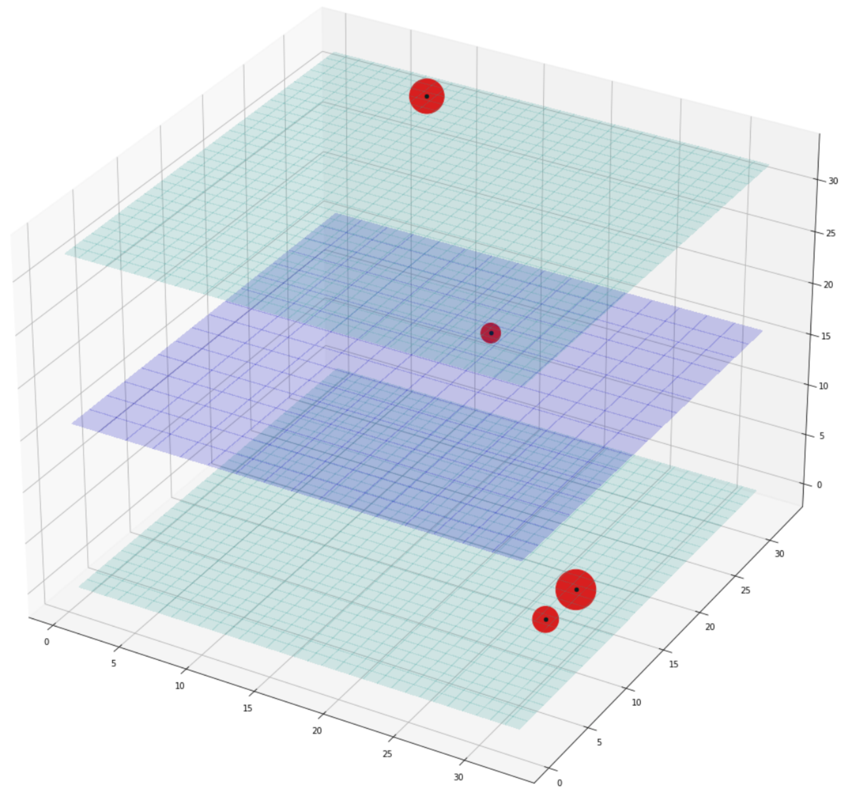

Figure 6. An example of a graphical representation of the Cosmic Muon Image data, showing the three layers of the detector [(from zooniverse)]()

When a muon is not present in the data, the holes couldn't be aligned between the three layers of the detector, and in general, the sound doesn't seem to go up or down in frequency following a line during the same sonification of the three layers (Figure 8).

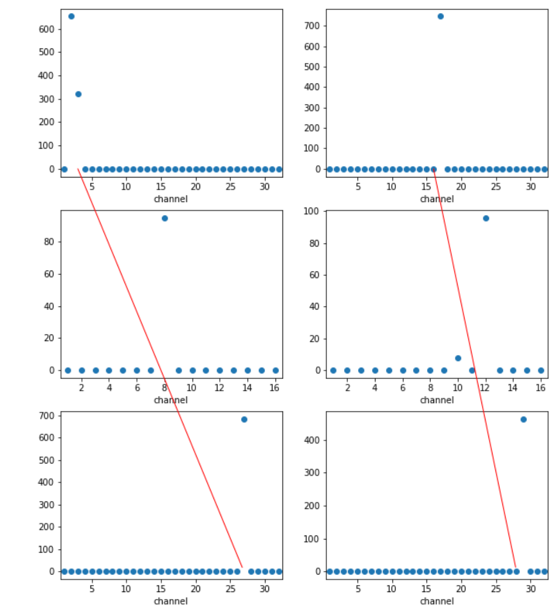

Figure 7. An example of the presence of a muon in the data, the sound could be heard [here](#).

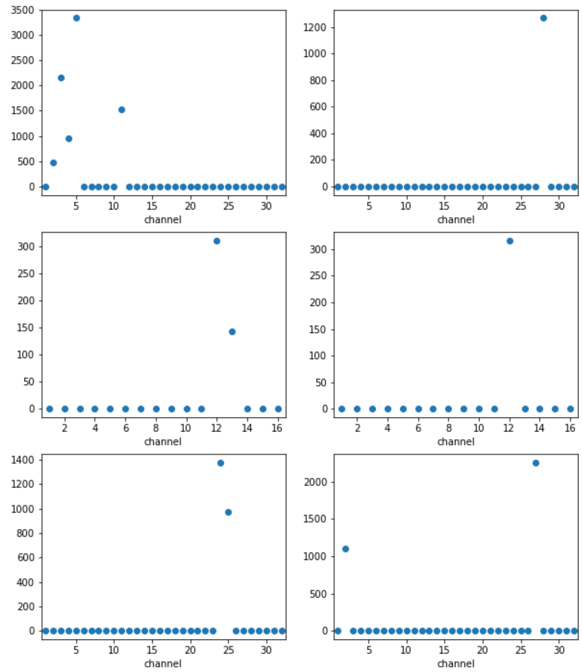

Figure 8. An example of the non-existence of muon, the sound could be heard [here](#).

3.1 Pre-requirements

Download the code from [sonoUno GitHub](#), or clone the repository with git.

3.2 Running muon sonification script

To run the muon sonification scripts, inside the command window go to the sonoUno folder where the script (muon_bash.py) is located and type:

➔ `python3 muon_bash.py -t "csv" -d "path_to_the_folder_where_muon_data_are" -p True`

NOTE: *be careful here, under the '-d' parameter we write the folder location, not the file location. This script sonify all the data in csv or txt format inside the path given.*

The script opens a new plot window where the muon data plot will be displayed (Figure 9) and the points being sonified will change its color during the reproduction of the sound (see section 3.3). After it finishes, the plot (png file) and sound (wav file) is saved in the same folder with the added plot or sound label to the data file name.

About the command to run the script: the '-t' indicates the file type (in the example, csv); the '-d' indicates the directory where the data files are located; and the '-p' indicates if the plot has to be displayed or not.

3.3 Detecting the existence of muons

About the data file, each line represents a channel and each column represents each view (x-y) of the 3 layers of the detector (these are each one of the six plots in the 1D representation (Figure 9)).

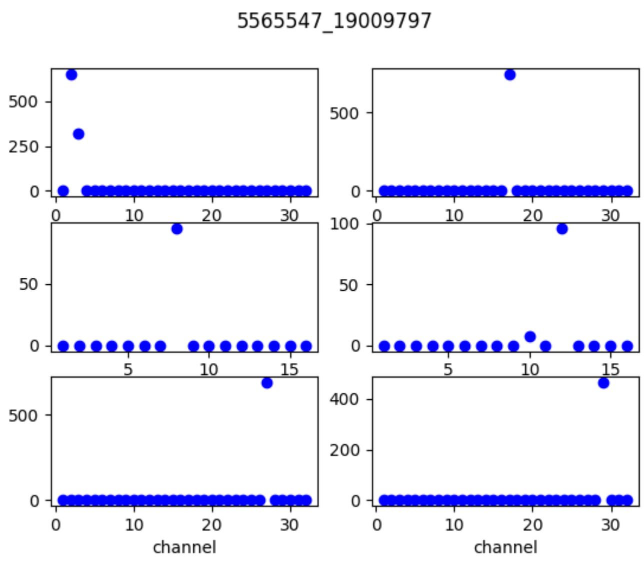

Figure 9. An example of the existence of muon plotted with the script, the sound could be heard here.

From this 1D representation, to be able to sonify it and could distinguish when the holes of the right part of the plot (top-center-button) are aligned, the next sound configuration was adopted:

- ➔ Each of the 32 or 16 channels of each plot on the right/left was assigned to specific piano notes. For simplicity, 16 piano notes where used, so in the case of 32 channels, the piano note change by 2 channels, this is:

Piano key 'A3'	correspond to channel 1 and 2 of top and bottom plots; correspond to channel 1 of center plot.
Piano key 'B3'	correspond to channel 3 and 4 of top and bottom plots; correspond to channel 2 of center plot.
Piano key 'C4'	correspond to channel 5 and 6 of top and bottom plots; correspond to channel 3 of center plot.
Piano key 'D4'	correspond to channel 7 and 8 of top and bottom plots; correspond to channel 4 of center plot.
Piano key 'E4'	correspond to channel 9 and 10 of top and bottom plots; correspond to channel 5 of center plot.
Piano key 'F4'	correspond to channel 11 and 12 of top and bottom plots; correspond to channel 6 of center plot.
Piano key 'G4'	correspond to channel 13 and 14 of top and bottom plots; correspond to channel 7 of center plot.
Piano key 'A4'	correspond to channel 15 and 16 of top and bottom plots; correspond to channel 8 of center plot.
Piano key 'B4'	correspond to channel 17 and 18 of top and bottom plots; correspond to channel 9 of center plot.
Piano key 'C5'	correspond to channel 19 and 20 of top and bottom plots; correspond to channel 10 of center plot.
Piano key 'D5'	correspond to channel 21 and 22 of top and bottom plots; correspond to channel 11 of center plot.
Piano key 'E5'	correspond to channel 23 and 24 of top and bottom plots; correspond to channel 12 of center plot.
Piano key 'F5'	correspond to channel 25 and 26 of top and bottom plots;

	correspond to channel 13 of center plot.
Piano key 'G5'	correspond to channel 27 and 28 of top and bottom plots; correspond to channel 14 of center plot.
Piano key 'A5'	correspond to channel 29 and 30 of top and bottom plots; correspond to channel 15 of center plot.
Piano key 'B5'	correspond to channel 31 and 32 of top and bottom plots; correspond to channel 16 of center plot.

In the case of more than one channel presenting a hole (Figure 8 at left-top) a sound composed by all the piano keys of the channels that present a hole is produced. Figure 10 shows an example of non-existence of a muon and in the composition of the final sound, this case is very evident.

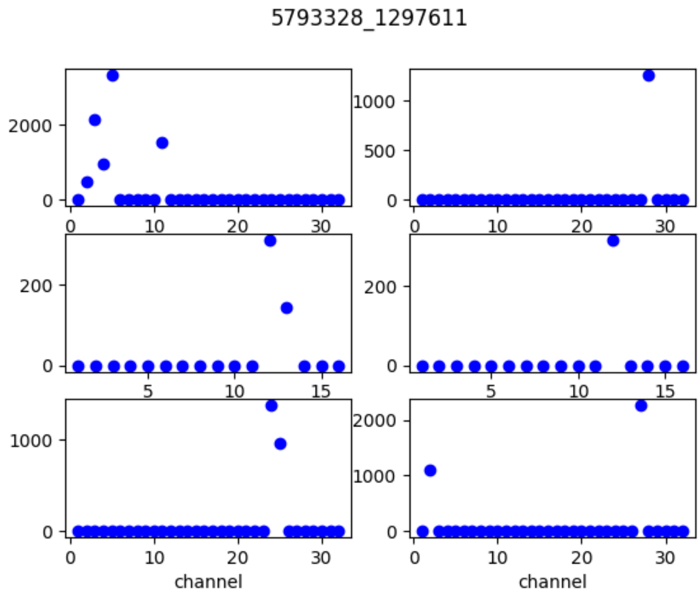

Figure 10. An example of the non-existence of muon plotted with the script, the sound could be heard here.

With this sound configuration, a change on the image during the reproduction was made. Each point being sonifyed change its color and adopt a color from red to violet corresponding the sound frequency, this is: red for low frequency and violet for high frequency. The color arrange is:

Red	correspond to channel 1 to 4 of top and bottom plots; correspond to channel 1 and 2 of center plot.

Orange	correspond to channel 5 to 8 of top and bottom plots; correspond to channel 3 and 4 of center plot.
Yellow	correspond to channel 9 to 12 of top and bottom plots; correspond to channel 5 and 6 of center plot.
Olive	correspond to channel 13 to 16 of top and bottom plots; correspond to channel 7 and 8 of center plot.
Green	correspond to channel 17 to 20 of top and bottom plots; correspond to channel 9 and 10 of center plot.
Cyan	correspond to channel 21 to 24 of top and bottom plots; correspond to channel 11 and 12 of center plot.
Blue	correspond to channel 25 to 28 of top and bottom plots; correspond to channel 13 and 14 of center plot.
Purple	correspond to channel 29 to 32 of top and bottom plots; correspond to channel 15 and 16 of center plot.

In addition, there are some cases where detecting the existence or not of a muon is very difficult with sound. In those cases, there are a lot of holes producing a sound with harmonics but visually the possibility to draw a line between the different layers of each view is possible. In these difficult detection cases, visual and auditory approaches are needed. One example is shown in Figure 11.

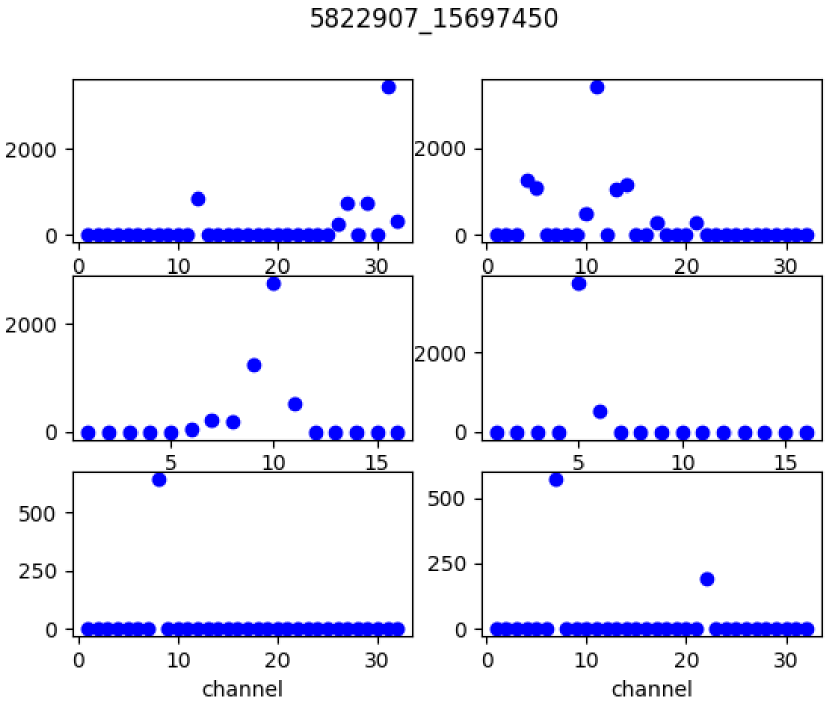

Figure 11. An example of the existence of muon plotted with the script, the sound could be heard [here](#).