



Interactive Music Science Collaborative Activities
 Team Teaching for STEAM Education

Deliverable 4.5

First Version of Music Visualization and Analysis Tools

Date:	05/10/2017
Author(s):	Jean-Marie Laborde (CABRI), Aggelos Gkiokas (ATHENA), Maximos Kaliakatsos-Papakostas (ATHENA), Manuel Bouillon (UNIFRI), Robert Piechaud (IRCAM), Marcus Liwicki (UNIFRI)
Contributor(s):	
Quality Assuror(s):	Robert Piechaud (IRCAM), Carlos Acosta (LEOPOLY)
Dissemination level:	PU
Work package	WP4 – Core enabling technologies modules of iMuSciCA
Version:	1.0
Keywords:	music analysis tools, music visualization tools
Description:	First version of the music analysis and visualization tools.



H2020-ICT-22-2016 Technologies for Learning and Skills
iMuSciCA (Interactive Music Science Collaborative Activities)
 Project No. 731861
 Project Runtime: January 2017 – June 2019
 Copyright © iMuSciCA Consortium 2017-2019

Executive Summary

In this deliverable we present the first version of the music analysis and visualization tools. The demonstrators of these tools are given as standalone web browser applications, loaded from the servers of partners who are mainly responsible for their development (links are given below). These tools can be grouped into the following categories:

- **Basic Core-enabling Musical Technologies:** this category encompasses tools that enable basic musical actions like counting time (Metronome), tuning an instrument (Tuner or Reference Pitch) and record audio (Audio Recorder).
- **Music, Sound and Basic Sound Visualisation:** tools in this category allow students to explore musical ideas by drawing using different timbres (MuSoVis), watch basic visualisations (including spectrogram, and frequency response) and experiment with additive synthesis and visualise the waveform plot along with also the mathematical expressions that produce the sound.
- **Advanced Visualisation tools:** this category includes i) the Snail tool which features a musically-informed audio visualisation ii) the Tempogram, Envelope and 3D Spectrogram tools which employ advanced signal processing for presenting visualisations relevant to rhythm and timbre.
- **Mathematics and Geometry Visualisation Tools:** the set of tools included in Cabri Express that allows graph visualisations using various mathematical modules.

List of available online tools:

Music Tools:

Reference Pitch:

<http://tesla.ilsp.gr:1994/pitcha.1/>

Metronome:

<http://tesla.ilsp.gr:1994/metronome.1/>

All tools (including the Tuner and the Audio Recorder):

<http://tesla.ilsp.gr:1994/musicTools.1/>

Music, Sound and Basic sound Visualization:

UNIFRI Sound Analyzer:

<https://imuscica-platform.unifr.ch/drawme/first/analyser.html>

Sinoids:

<http://tesla.ilsp.gr:1994/sinusoids.1/>

Advanced Visualization Tools:

Envelope:

<http://tesla.ilsp.gr:1994/visualizations.1/envelope.html>

Tempogram:

<http://tesla.ilsp.gr:1994/visualizations.1/tempogram.html>

3D Spectrogram:

<http://tesla.ilsp.gr:1994/visualizations.1/spectrogram.html>

All Visualizations:

<http://tesla.ilsp.gr:1994/visualizations.1/>

Snail:

<http://devtest.leopoly.com/3d-interaction-repository-V1/modalys/public/snailtest.html>

CabriExpress:

<http://cabricloud.com/cabriexpress>

Version Log			
Date	Version No.	Author	Change
04-09-2017	0.1	Laurent Varin (CABRI)	Initial content
23-09-2017	0.2	Maximos Kaliakatsos-Papakostas (ATHENA)	Content update
25-09-2017	0.3	Manuel Bouillon (UNIFRI)	Section 2
28-09-2017	0.5	Marcus Liwicki (UNIFRI)	Updating Summary and Proofreading
01-10-2017	0.6	Maximos Kaliakatsos-Papakostas (ATHENA)	Reviewing contributions
04-10-2017	0.7	Maximos Kaliakatsos-Papakostas (ATHENA)	Contributions of reviewers
05-10-2017	1.0	Vassilis Katsouros (ATHENA)	Submission to EU

Disclaimer

This document contains description of the iMuSciCA project findings, work and products. Certain parts of it might be under partner Intellectual Property Right (IPR) rules so, prior to using its content please contact the consortium head for approval.

In case you believe that this document harms in any way IPR held by you as a person or as a representative of an entity, please do notify us immediately.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of iMuSciCA consortium and can in no way be taken to reflect the views of the European Union.

iMuSciCA is an H2020 project funded by the European Union.

TABLE OF CONTENTS

Executive Summary	1
1. Introduction	6
2. Basic Core-Enabling Musical Technologies: Metronome, Tuner, Audio Recording and Reference Pitch	6
2.1 Installation and technical requirements	6
2.2 Description of demonstrator and user manual	6
3. UNIFRI MuSoVis (Music and Sound Visualization)	9
3.1 Installation and technical requirements	9
3.2 Description of demonstrator and user manual	10
3.2.1 Main view	10
3.2.2 Control buttons and settings	10
3.2.1 Miniature panel	11
4. A Core-Enabling Technology for Studying Timbre: the Additive Synthesis Tool	12
4.1 Installation and technical requirements	12
4.2 Description of demonstrator and user manual	12
5. The Snail: an advanced music visualization tool based on spectral analysis	13
5.1 Installation and technical requirements	13
5.2 Description of the prototype demonstrator	14
5.2.1 Advanced parameters	15
5.2.2 Language selection	16
5.3 Performances	16
5.4 Example of a Snail connection: with Modalys	16
6. Advanced Music Visualisation Core-Enabling Technologies: Tempogram, Envelope and Spectrogram	3D 18
6.1 Installation and technical requirements	18
6.1.1 Importing Visualization Tools as javascript Libraries	18
6.1.1.1 Importing Envelope	19
6.1.1.2 Importing Tempogram	19
6.1.1.3 Importing Spectrogram	19
6.1.2 Importing Visualization Tools as iFrames	20
6.1.2.1 Importing Envelope	20
6.1.2.2 Importing Tempogram	20
6.2 Description of demonstrator and user manual	21
7. Cabri Express, all purpose math exploratory tool	22
7.1 Installation and technical requirements	22
7.2 Description of demonstrator and user manual	23

LIST OF ABBREVIATIONS

Abbreviation	Description
MuSoVis	Music and Sound Visualisation
FFT	Fast Fourier Transform
WebGL	Web Graphic Library
CSS	Cascading Style Sheet
STEAM	Science, Technology, Engineering and Maths combined with Arts
WP	Work Package
PU	Public document
ATHENA	ATHENA RESEARCH AND INNOVATION CENTER IN INFORMATION COMMUNICATION & KNOWLEDGE TECHNOLOGIES
UCLL	UC LIMBURG
EA	ELLINOGERMANIKI AGOGI SCHOLI PANAGEA SAVVA AE
IRCAM	INSTITUT DE RECHERCHE ET DE COORDINATION ACOUSTIQUE MUSIQUE
LEOPOLY	3D FOR ALL SZAMITASTECHNIKAI FEJLESZTO KFT
CABRI	Cabrilog SAS
WIRIS	MATHS FOR MORE SL
UNIFRI	UNIVERSITE DE FRIBOURG

1. Introduction

The iMuSciCA learning environments should allow easy and intuitive access to core-enabling technologies around music and visualisations. To this end, music/audio and mathematical visualisation tools have been developed along with basic music-related tools that will allow students to obtain intuitive information about sound, mathematical equations and geometric shapes through visualisations, while basic tools will facilitate music training (e.g. metronome) and ear training (e.g. pitch reference tool). These core-enabling technologies are in accordance with the purposes of STEAM education and follow a simple and intuitive design approach, as demonstrated by the urls provided in the following section.

2. Basic Core-Enabling Musical Technologies: Metronome, Tuner, Audio Recording and Reference Pitch

2.1 Installation and technical requirements

The core enabling technologies that are incorporated in the basic musical tools are demonstrated in the following link:

<http://tesla.ilsp.gr:1994/musicTools.1/>

The link given above can be opened by any web browser in any operating system, however using Firefox Nightly in Windows is recommended. Since those tools allow the generation and capturing of audio, speakers and a microphone are required for using/testing them. For demonstration purposes, this web page includes all the tools pertaining to the basic musical core enabling technologies. The tools will be available in separate iframes embedded in the workbench and they can be made available to use upon user request. After the description of each tool, a dedicated URL is given for each tool for opening each tool individually.

2.2 Description of demonstrator and user manual

The demonstrator for the basic music core enabling technologies given in the link in the previous subsection includes a metronome, a tool for playing a reference pitch, a tuner, a tool for loading audio files and enabling microphone input and an audio recording tool. Using tool in this list of core-enabling technologies, the student will be able to perform the following tasks:

- Use the metronome for keeping a steady time during musical performance (i.e. during free or gesture-based interaction with the instrument),
- Use the reference pitch tool to listen to a reference pitch for, e.g., tuning the instrument during the instrument design phase.

- Use the tuner to tune the instrument during instrument design in cent-level detail.
- Use the audio input tool to either play a music file (e.g. as a backing track during performance with the virtual instrument) or open the microphone for using audio from sources outside the iMuSciCA system (e.g. voice or physical instrument among others).
- Record the produced audio from all sound-producing tools in the iMuSciCA workbench, except from the metronome, and save it either to audio file or on the cloud.

Figure 2.2.1 illustrates the collection of the tools pertaining to the basic musical core-enabling technologies (this figure is a screenshot of the web page given in the link in the previous section). A short description of those tools is given as follows:

1. Metronome: the user can change the tempo and the swing values using the respective sliders, the beat resolution using the dropdown menu in the bottom and start/stop the metronome using the start/stop button. Dedicated URL:
<http://tesla.ilsp.gr:1994/metronome.1/>
2. Reference Pitch: this tool incorporates 12 equally spaced regions within a circle, each of which corresponds to the 12 well-tempered scale pitches in standard (440) tuning of the 5th octave. The user can listen to each pitch by pressing on the respective area. Dedicated URL:
<http://tesla.ilsp.gr:1994/pitcha.1/>
3. Tuner: the fundamental frequency of any sounding element in the iMuSciCA workbench is computed and displayed in the tuner screen as a pitch symbol (using standard 440 tuning), Hz value and deviation in cents from the displayed pitch (in case the tuning is not perfect). No dedicated URL is given for this tool since it is dependent on audio input received from any of the hitherto mentioned tools. Please use the URL demonstrating all the tools for examining the tuner:
<http://tesla.ilsp.gr:1994/musicTools.1/>
4. Audio input: the user can select an audio file in the iMuSciCA cloud and play it, or enable input from the computer microphone using the respective buttons. Dedicated URL:
<http://tesla.ilsp.gr:1994/micFileInput.1/>
5. Audio recording: the user can start recording all sounding elements in the iMuSciCA workbench (except from the microphone) and when the recording stops the user has the option to discard or save the recording either as a file or in the cloud. No dedicated URL is given for this tool since it is dependent on audio input received from any of the hitherto mentioned tools. Please use the URL demonstrating all the tools for examining the audio recorder:
<http://tesla.ilsp.gr:1994/musicTools.1/>



Figure 2.2.1: First version of the basic musical core-enabling technologies.

3. UNIFRI MuSoVis (Music and Sound Visualization)

UniFri MuSoVis is a simple and yet powerful music and sound visualization tool.

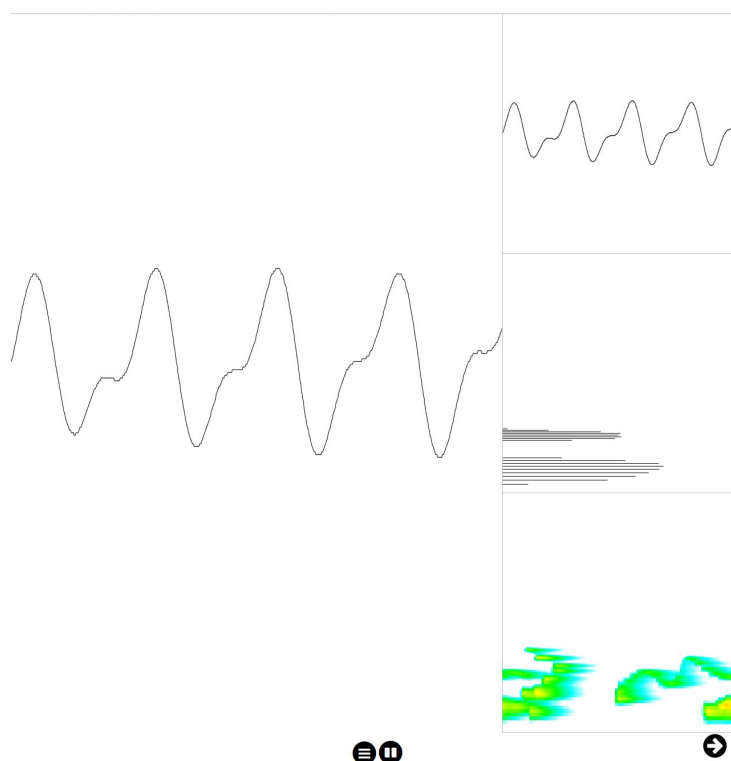


Figure 3.1. UniFri MuSoVis with the main view in the middle, the control buttons at the bottom and the three possible visualization views on the right side.

3.1 Installation and technical requirements

UniFri MuSoVis is based on JavaScript, so it runs on any modern browser (Chrome, Firefox, Edge, etc.) and operating system (Windows, GNU/Linux, macOS) and needs a microphone. It is accessible at the following URL: <https://imuscica-platform.unifr.ch/drawme/first/analyser.html>. You can insert UniFri Movis in an iframe with the following code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html lang=en>
<head>
  <meta charset=utf-8>
  <title>UniFri Movis</title>
</head>
```

```
<body style="margin:0px;padding:0px;overflow:hidden">
  <!-- DAME iframe -->
  <iframe id="tool" src="https://imusica-canvas.herokuapp.com/analyser.html"
frameborder="0" height="100%" width="100%"></iframe>
</body>
</html>
```

3.2 Description of demonstrator and user manual

UniFri MuSoVis is composed of three parts, the drawing canvas in the middle, the control buttons at the bottom and the visualization views on the right.

3.2.1 Main view

The user can visualize sound and music in the main view, which can be either of:

- A waveform view
- A Fourier transform view
- A spectrogram view

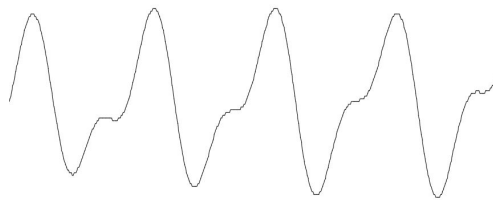


Figure 3.2.1.1. *The main view displaying the waveform view.*

3.2.2 Control buttons and settings

Below the main view, the control buttons enable to select the sound source between:

- UniFri DrAwME (See Deliverable 4.2)
 - For this one, just open DrAwME and click on the right panel
 - Works best open the two side by side
- The user microphone
 - Click on the microphone button and the sound of the microphone will be visualized

The pause/play button enable to freeze the current view and then export it in JSON format.



Figure 3.2.2.1. *The control buttons with the source selection on the left, the pause/play button in the middle and the export option on the right.*

3.2.1 Miniature panel

The miniature panel on the right display small previews of the three different views:

- The waveform view;
- The Fourier transform view;
- The spectrogram view.

Each of the three view can be display in the main view by clicking on it. The miniature panel can be hidden by clicking on the arrow at the bottom right.

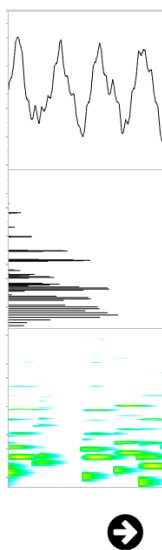


Figure 3.2.1.1. *The visualization panel include three different views: a waveform, a Fourier transform and a spectrogram. The visualization panel can be hidden by clicking on the arrow on the bottom right.*

4. A Core-Enabling Technology for Studying Timbre: the Additive Synthesis Tool

4.1 Installation and technical requirements

The Additive Synthesis tool constitutes a core-enabling technology that allows students to explore timbres as an addition of multiple sinusoidal elements. The students are able to listen to the sound produced by up to eight sinusoidal elements combined (summed), change their parameters (amplitude and frequency) and visualise the produced waveform either of the sum or of each element separately. The x-axis values are in seconds, so the students can perform calculations for estimating the fundamental frequency/periodicity of the produced sound. This tool can be found in the following link:

<http://tesla.ilsp.gr:1994/sinusoids.1/>

The link given above can be opened by any web browser in any operating system, however using Firefox Nightly in Windows is recommended. Since this tool generates audio, speakers are required for using/testing it.

4.2 Description of demonstrator and user manual

The demonstrator of this core-enabling technology can be found in the link provided in the previous section. Figure 4.2.1 shows a screenshot of this tool with numbers describing the utilities that this tool offers:

1. Waveform visualisation: the user can zoom in/out the waveform by scrolling, or move left, right, up and down by click-dragging in the respective directions.
2. Scaling adjustment: by using the X/Y-adjustment sliders, the user can zoom in/out the respective directions, scaling it relative to the other direction.
3. Adjustment of coefficients: by either using the sliders or the textboxes, the student can adjust the values of the sinusoidal elements, i.e. change their amplitude and frequency.
4. Playing/showing options: the student can select to press start/stop sound generation for each element and visualise it on screen in order to understand visually the contribution of each element.
5. Play/stop all and add/remove elements: The student can select to start/stop all elements that are currently used and modify the number of currently used sinusoidal elements by pressing the "+" and "-" buttons in the bottom.

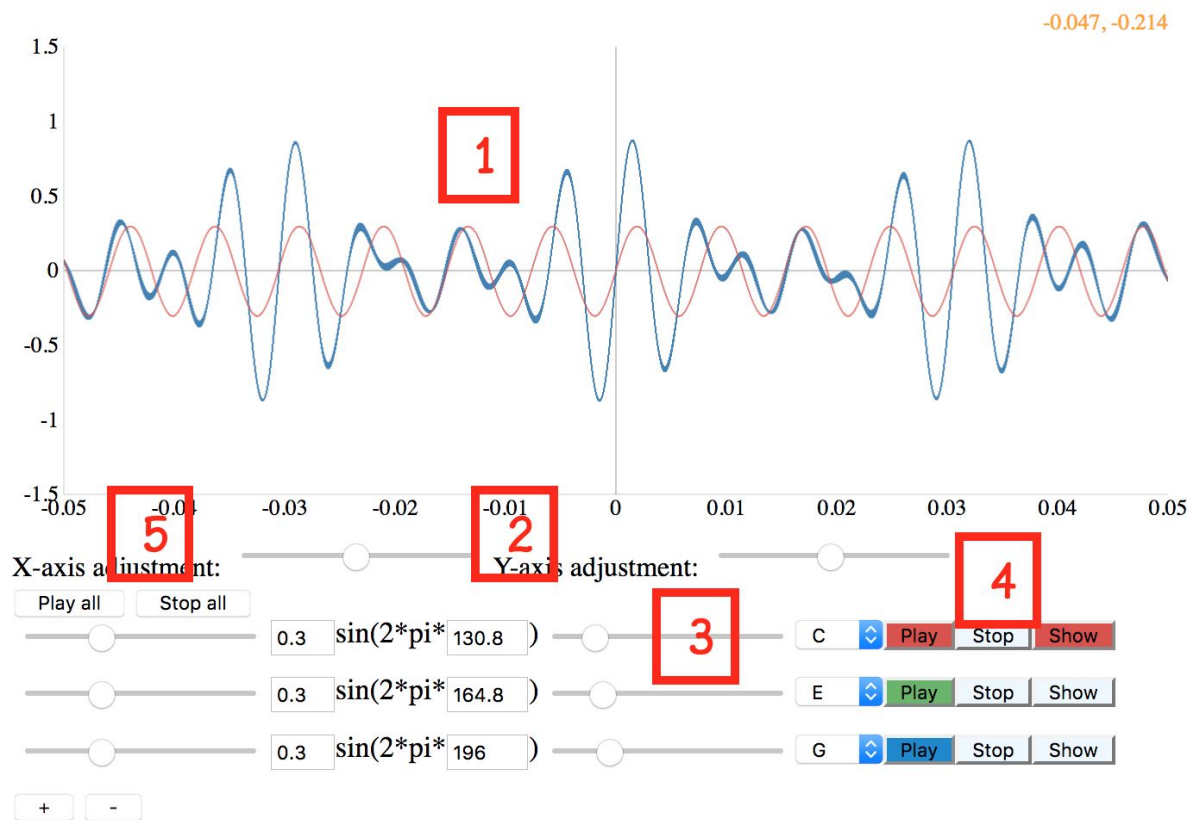


Figure 4.2.1: First version of the Additive Synthesis core-enabling technology.

5. The Snail: an advanced music visualization tool based on spectral analysis

Snail.js is the browser version of the eponymous standalone application developed by Ircam¹. The key concept of the Snail, unlike classical sound representations such as spectrogram or sonogram, is to display the spectrum of a sound in a very musical fashion, that is, on a spiral featuring notes (equally tempered) and octaves. Each component of a sound - or *partial* according to the Fourier theory², is displayed as a “blob” which size and color depend on the energy and phase shift in the Fourier decomposition. With this representation, which evolves very dynamically over time, anyone with a little or no scientific background can immediately “see” the sound and its spectral structure, as it is being produced.

5.1 Installation and technical requirements

The Snail is delivered as a standalone javascript library - **snail.js** - that can be instantiated anywhere inside a html 5 browser page. Very much like modalys.js, snail.js is compiled from C/C++ using Emscripten³ using heavy optimizations. The FFT used behind the hood is pffft⁴, an highly

¹ <http://www.ircamlab.com/products/p2242-The-Snail/>

² https://en.wikipedia.org/wiki/Fourier_transform

³ <http://devtest.leopold.com/3d-interaction-repository-V1/modalys/public/>

⁴ <https://bitbucket.org/jpommier/pffft>

optimized open source fft library written by Julien Pommier. Snail.js will run ideally in Firefox or Edge “nightly”⁵, but also in Chrome or Opera.

A snail object is typically instantiated passing these parameters:

- The div that will embed the snail
- The audio context
- Some optional parameters (as a key/value array)

Finally, an audio source (modalys.js, the microphone, an existing sound file etc.) is connected to the snail for analysis and visualization:

```
var snail = new Snail(document.getElementById("snail"),
                    ctx,
                    {"showControls": true, "size":400});
source.connect(snail);
```

5.2 Description of the prototype demonstrator

The prototype is hosted at this url:

<http://devtest.leopoly.com/3d-interaction-repository-V1/modalys/public/snailtest.html>

The Snail demonstration page features the following:

- The Snail area per se, drawn in pure WebGL
- A button for opening an existing sound file⁶
- The Note Range slider pair
- The FFT window size slider (default is 50 ms)
- The frequency tracking slide (default is 6 hz).
- The note name popup menu (English #/b, French #/b, Greek #/b).
- The Snail global size (in pixels). Everything is sized according to this value.
- The note font size, as in CSS code.

Note that by setting the *showControls* parameter to false, we may hide the controls (as in the next example with Modalys).

⁵ The latest alpha development builds of Firefox and Edge.

⁶ Beware that certain browsers accept certain formats only. For instance, Firefox will accept .ogg or .wav only.

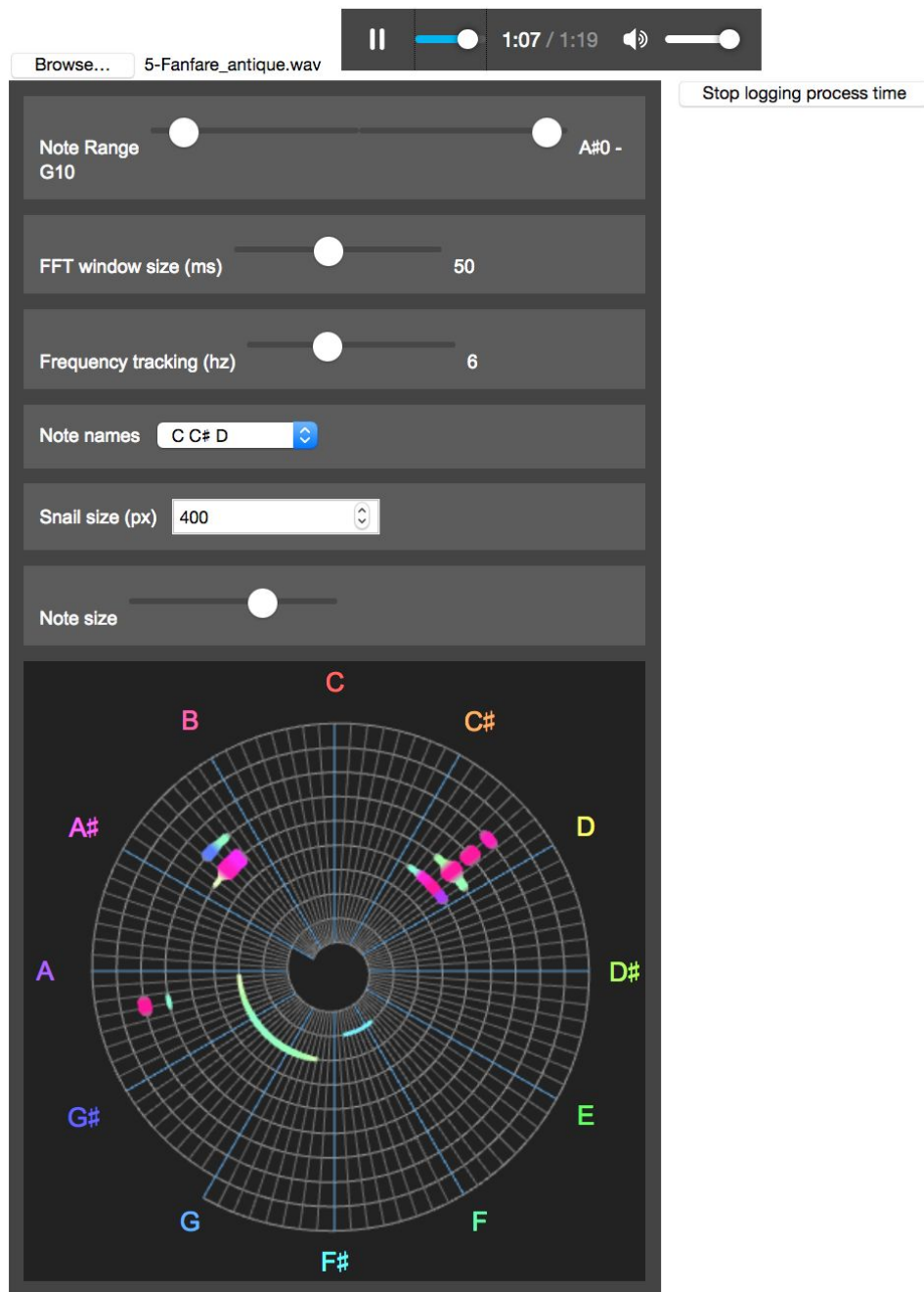


Figure 5.2.1: Interface of the web browser version of the Snail visualisation tool.

5.2.1 Advanced parameters

In this prototype, some advanced parameters, notably FFT window size and frequency tracking bandwidth, are also available through the API and can be changed at run time. We may or may not expose them in the final form of `snail.js`. They can also be passed upon instantiation inside the key-value parameter array:

```
{
  "size": 600,
```



```

"noteSize": 0.68,
"noteNames": "ENGLISH_SHARP",
"interfaceLanguage": "en",
"bufferSize": 1024,
"noteMinMidi": 10,
"noteMaxMidi": 127,
"fftWindowSize": 50,
"frequencyTracking": 6,
"showControls": false
}

```

5.2.2 Language selection

The demonstrator shows a language selection menu. Currently, this concerns the note naming only, featuring French, English and Greek:

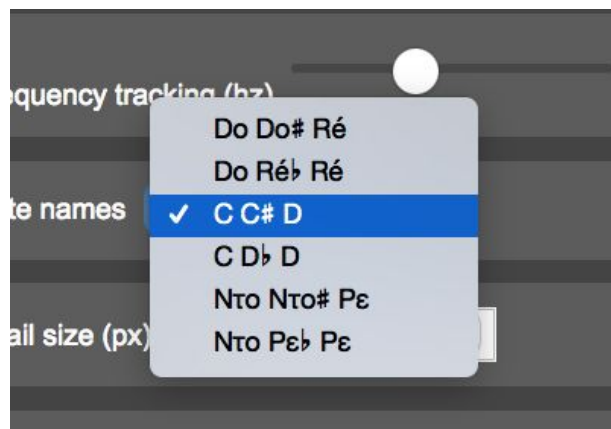


Figure 5.2.2.1: Multi-language selection menu of note naming.

5.3 Performances

Snail.js features pretty decent performances. The average processing time for a buffer of 1024 is 1.5 ms (audio + display) when running Firefox Nightly 57.a1 on a Macbook Pro. We might get even better performances by targeting WebAssembly⁷ during emscripten compilation.

5.4 Example of a Snail connection: with Modalys

In this example, we connect an instance of the Snail to one of Modalys:

```

Modalys.onready = function()
{
    var audio = Modalys.getAudioContext();
    var snail = new Snail(document.getElementById("snail"), audio,

```

⁷ <https://en.wikipedia.org/wiki/WebAssembly>

```

{"showControls": false, "size":300});
  Modalys.setDestination(snail, audio.destination);
};

```

This example demonstrates how easily the Snail can be connected to an audio node for a deeper understanding of the musical structure of sound (here there are 3 strings pitched around low F, C and G).

modalys.js and snail.js combined: plucked strings

modalys.js version: 0.1.16

String 1

pluck point: listening point: pitchbend: 0

String 2

pluck point: listening point: pitchbend:

String 3

pluck point: listening point: pitchbend:

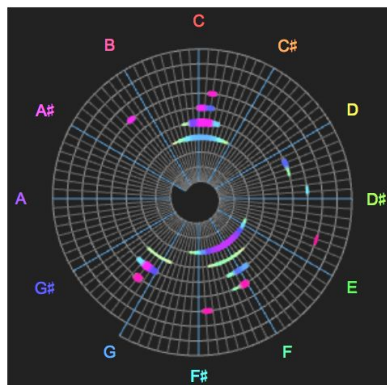


Figure 5.4.1: Demonstration of using Snail for visualising the audio produced by another tool, in this case Modalys.

6. Advanced Music Visualisation Core-Enabling Technologies: Tempogram, Envelope and 3D Spectrogram

6.1 Installation and technical requirements

The Advanced Music Visualization Technologies involve three visualization types:

- **Envelope:** Envelope is a 2D visualization showing the energy of the signal with respect to time.
- **Tempogram:** Tempogram is a 2D visualization showing the salience of rhythmic periodicities.
- **3D Spectrogram:** 3D Spectrogram is a 3D Visualization tool showing the Spectrogram in real-time. The user is able to rotate, zoom in and zoom out the 3D visualization.

The visualization tools can be imported either as iFrames, or as standalone javascript libraries.

6.1.1 Importing Visualization Tools as javascript Libraries

The visualization tools can be imported as javascript libraries from the script found in the following URL:

<http://tesla.ilsp.gr:1994/visualizations.1/scripts/audioVisual.js>

Other javascript dependencies are:

- audioManager.js

<http://tesla.ilsp.gr:1994/visualizations.1/scripts/audioManager.js>

- audioSources.js

<http://tesla.ilsp.gr:1994/visualizations.1/scripts/audioSources.js>

- Buffer.js

<http://tesla.ilsp.gr:1994/visualizations.1/scripts/buffer.js>

- Fft.js

<http://tesla.ilsp.gr:1994/visualizations.1/scripts/fft.js>

- OrbitControls.js

<http://tesla.ilsp.gr:1994/visualizations.1/scripts/OrbitControls.js>

- Three.js

<http://tesla.ilsp.gr:1994/visualizations.1/scripts/Three.js>

- utilsimuscica.js

<http://tesla.ilsp.gr:1994/visualizations.1/scripts/utilsimuscica.js>

- vizClasses.js

<http://tesla.ilsp.gr:1994/visualizations/scripts.1/vizClasses.js>

Each visualization tool is instantiated as a Class Object.

6.1.1.1 Importing Envelope

```
var processingNode = someAudioNode(); // node to receive audio from
var envelopeSize = 512; //size of envelope
var audioBufferSize = 1024; // size of audio analysis window
var audioBufferShift = 512; // size of audio analysis hop size
var myCanvas = newCanvas(); // a canvas object to draw the visualization

var envelopeVisualizer = new EnvelopeVisualizer(processingNode,
envelopeSize, audioBufferSize, audioBufferShift, myCanvas);

envelopeVisualizer.run();
```

6.1.1.2 Importing Tempogram

```
var processingNode = someAudioNode(); // node to receive audio from
var envelopeSize = 512; //size of envelope
var audioBufferSize = 1024; // size of audio analysis window
var audioBufferShift = 512; // size of audio analysis hop size
var myCanvas = newCanvas(); // a canvas object to draw the visualization

var tempogramVisualizer= new TempogramVisualizer(processingNode,
envelopeSize, audioBufferSize, audioBufferShift, myCanvas);

tempogramVisualizer.run();
```

6.1.1.3 Importing Spectrogram

```
var processingNode = someAudioNode(); // node to receive audio from
var envelopeSize = 512; //size of envelope
var audioBufferSize = 1024; // size of audio analysis window
var audioBufferShift = 512; // size of audio analysis hop size
var specLength= 100; // # of 3D spectrogram elements in time
var fftSize= 200; // # of 3D spectrogram elements in frequency

var myDiv = newdiv(); // a div html element to draw the visualization

var spectrogramVisualizer= new SpectrogramVisualizer(processingNode,
audioBufferSize, audioBufferShift, fftSize, specLength, myDiv);
```

```
tempogramVisualizer.run();
```

6.1.2 Importing Visualization Tools as iFrames

All visualization tools can be imported as iFrames. All iFrames should be provided an Script Processing Node from the parent window in order to receive the audio to process it. This is similar to the command of the previous section. The parent window that call the iFrame should provide a function to return the Script Processing Node:

```
var processingNode = someAudioNode(); // node to receive audio from
<script type="text/javascript">
function getScriptProcessingNode(){
    var myNode = new someScriptProcessingNode();
    return myNode;
}
</script>
```

6.1.2.1 Importing Envelope

```
<body style="margin:0px;padding:0px;overflow:hidden">
<!-- DAME iframe -->
<iframe id="envelope"
src="http://tesla.ilsp.gr:1994/iFrames/envelope.html" frameborder="0"
height="100%" width="100%"></iframe>
</body>
</html>
```

6.1.2.2 Importing Tempogram

```
<body style="margin:0px;padding:0px;overflow:hidden">
<!-- DAME iframe -->
<iframe id="envelope"
src="http://tesla.ilsp.gr:1994/iFrames/tempogram.html" frameborder="0"
height="100%" width="100%"></iframe>
</body>
</html>
```

```
<body style="margin:0px;padding:0px;overflow:hidden">
<!-- DAME iframe -->
<iframe id="envelope"
```

```
src="http://tesla.ilsp.gr:1994/iFrames/spectrogram.html" frameborder="0"
height="100%" width="100%"></iframe>
</body>
</html>
```

6.2 Description of demonstrator and user manual

A demonstration page of all visualizations can be found in the following link:

<http://tesla.ilsp.gr:1994/visualizations.1/>

A screenshot of the page is shown in Figure 6.2.1.

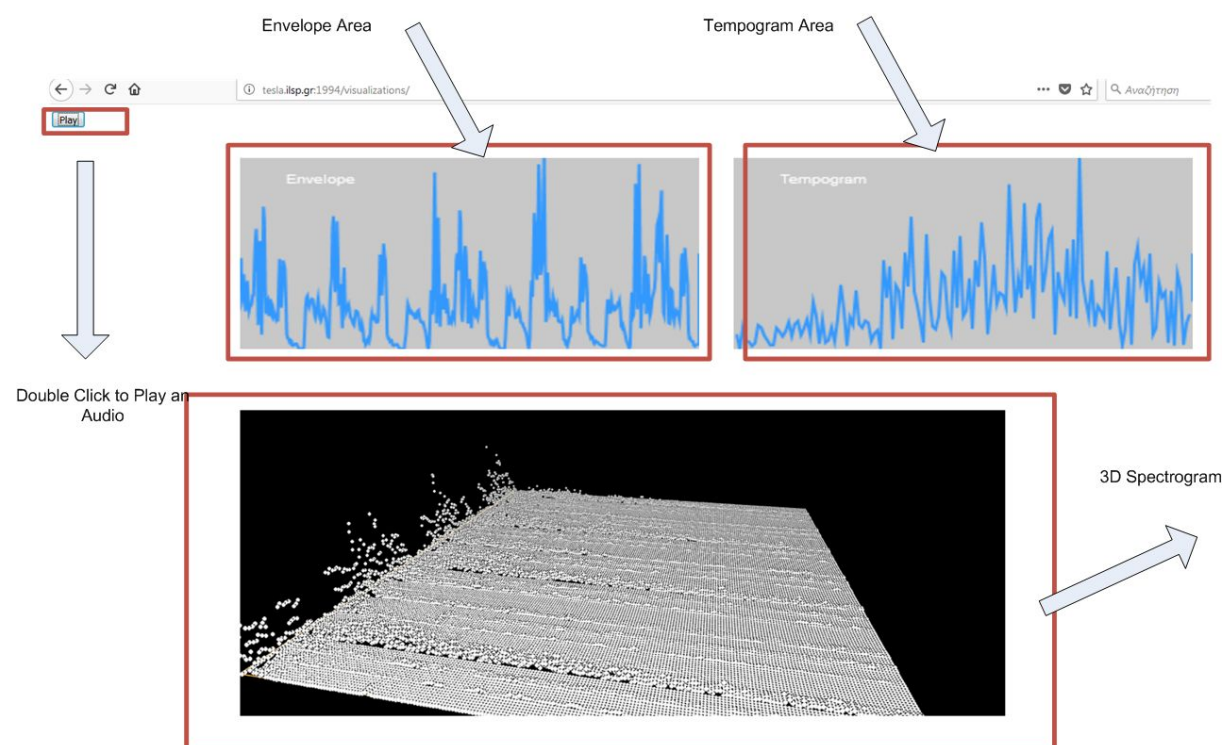


Figure 6.2.1: Screenshot of the demonstration web page that hosts the Envelope, Tempogram and 3D Spectrogram visualisation tools, where the “Play” starts the reproduction of an audio file.

On the top left corner of the window the user can double click on the *Play* button to start playing an audio file. The three visualization types will be displayed in real-time. On the top left and top right corners the envelope and the tempogram are shown respectively. On the bottom part of the window the 3D spectrogram is displayed. The user can navigate in the 3D spectrogram by rotating (holding left mouse button), moving (holding right mouse button) and zoom in and out (mouse scroll wheel).

The individual tools are also demonstrated at

- Envelope:
<http://tesla.ilsp.gr:1994/visualizations.1/envelope.html>

- Tempogram:
<http://tesla.ilsp.gr:1994/visualizations.1/tempogram.html>
- 3D Spectrogram:
<http://tesla.ilsp.gr:1994/visualizations.1/spectrogram.html>

7. Cabri Express, all purpose math exploratory tool

7.1 Installation and technical requirements

The Cabri tool, aka Cabri Express, can run in the workbench as an i-Frame exactly as the other tools.

It runs from following URL <http://cabricloud.com/cabriexpress> which is entered as an iFrame component (to be edited in the Creator version of Cabri)

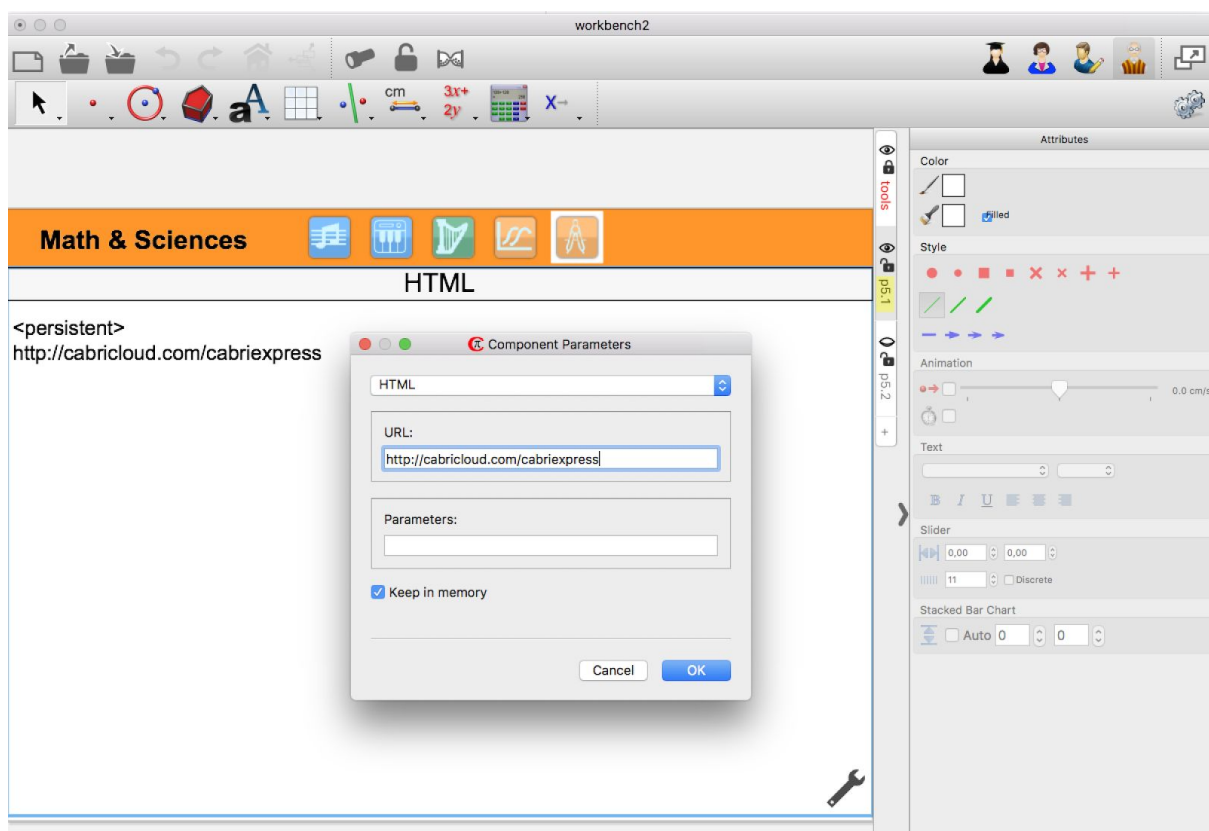


Figure 7.1.1: Illustration of the Cabri Express iFrame loader.

7.2 Description of demonstrator and user manual

The Cabri tool to explore math dynamically is available with pressing the following icon in the workbench:

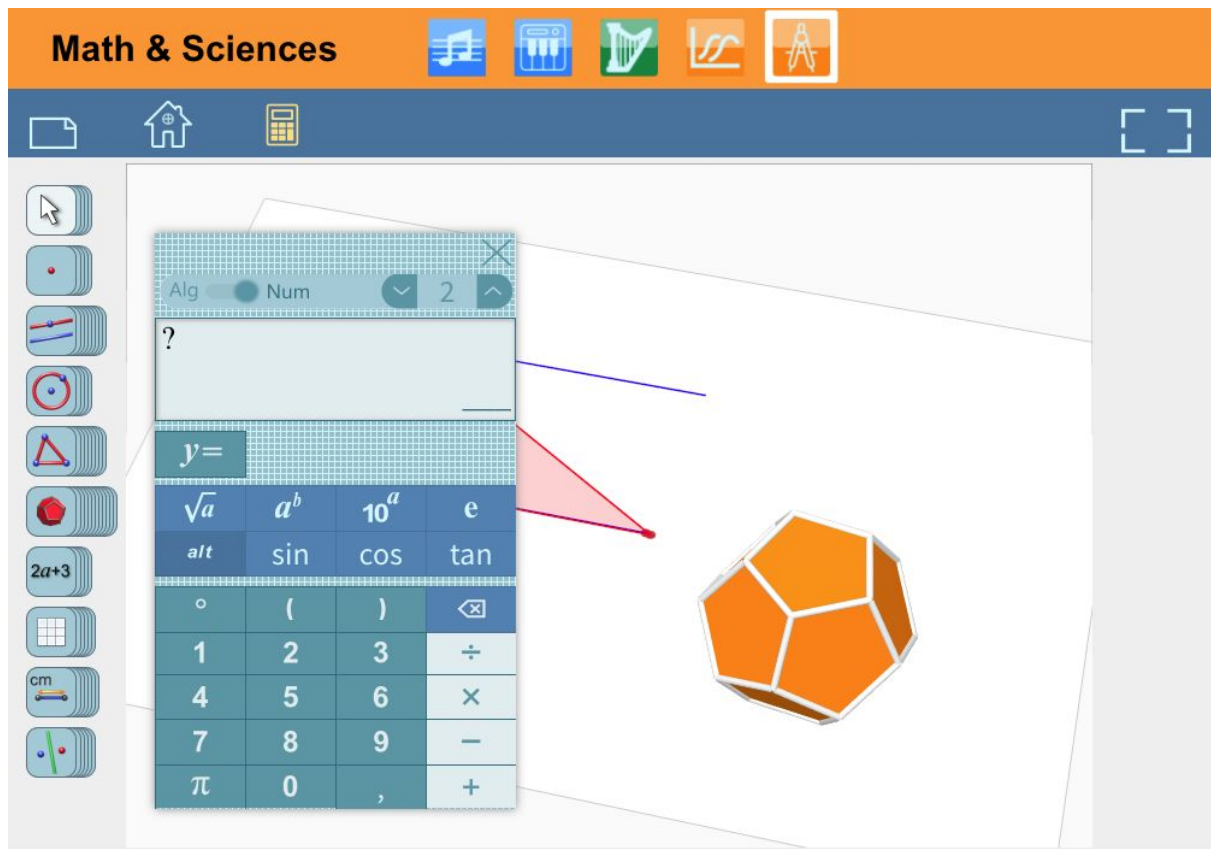


Figure 7.2.1: Screenshot of the Math and Science tab in the Cabri Express environment.

When Cabri Express is first started, it opens its calculator on top of the working area. This is convenient because in many the use cases the learner just needs to perform a calculation without specifically involving other tools as, e.g., geometric construction tools.

The Cabri Express calculator works as most of the so called “scientific pocket” calculators, but with additional features that are actually unique to it (more on those below).

Main constituents of Cabri Express are the **Express Bar**, the **Working Area** and the **Tool Piles** as indicated below:

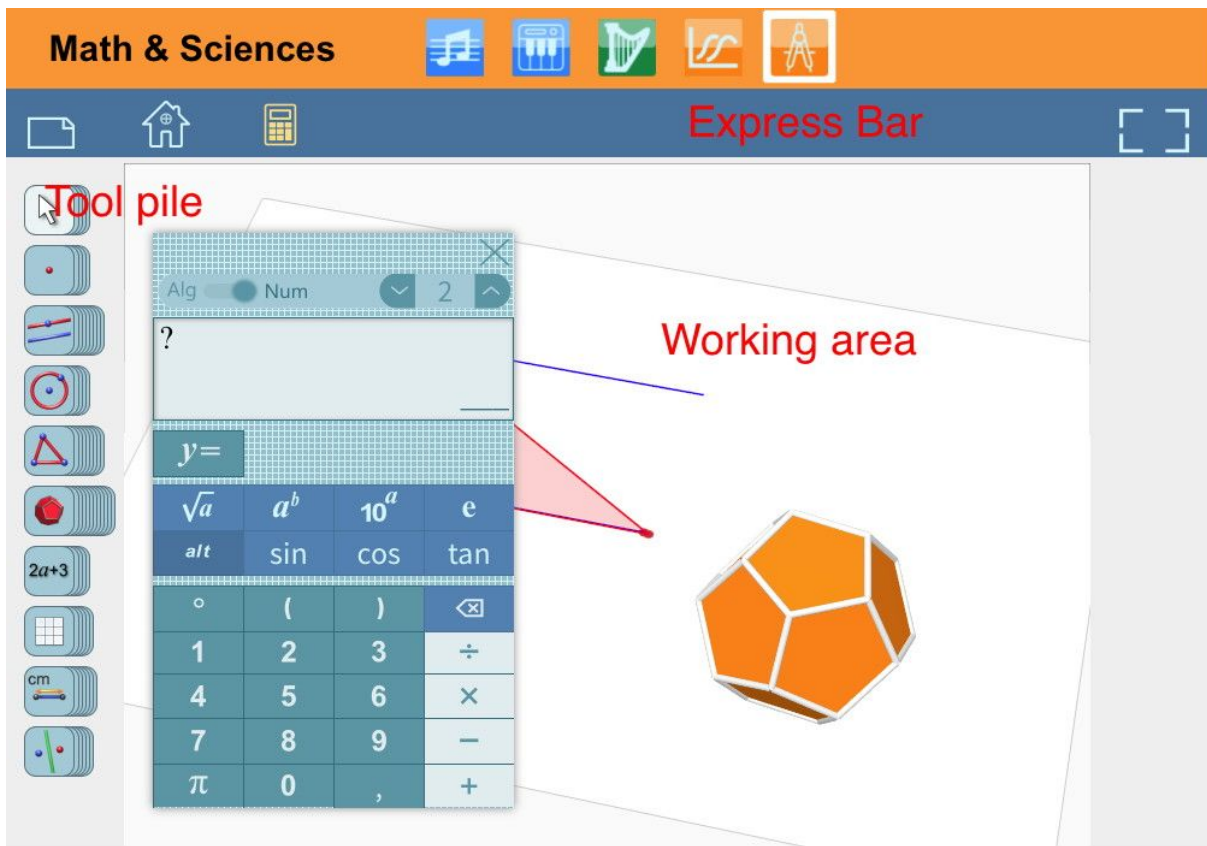


Figure 7.2.2: Illustration of the “Express Bar”, “Tool pile” and “Working area” interaction positions on the Cabri Express environment.

The calculator

The calculator embedded in Cabri Express is an advanced graphing calculator with a simplified UI presenting only few buttons to help students not getting lost by being immersed into an ocean of buttons.

Simple direct calculations can be made in pressing digit and/or 4-operation buttons.

As soon an entry can be numerically evaluated, its exact (or in some cases approximate) value is displayed below the entry.

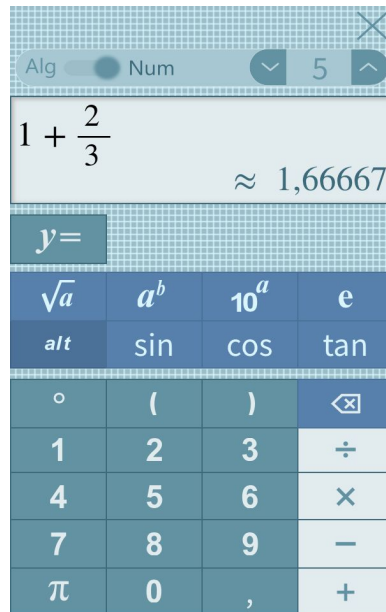


Figure 7.2.3: Illustration of the Cabri calculator in basic arithmetic use.

Actual display precision can be set, as the number of digits after the decimal separator, by pressing the two small arrows at the top of the calc. By default it is 2 (actually 5 in the above picture).

Instead of typing a number from scratch it is possible to click (tap) of existing values if some are available in the working area (e.g. the radius of a circle)

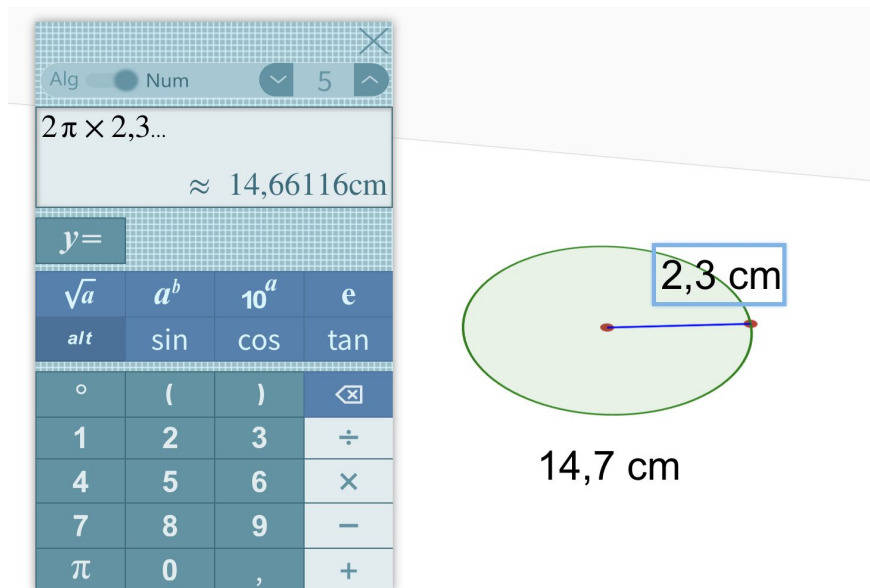


Figure: 7.2.4: Insertion of number in the calculator by clicking/tapping on numerical values as appearing on objects.

As one can see the calculator is able to manage units (e.g. cm, cm², cm³, 1/cm ... or degrees (°), revealing some aspects of the concept of dimension equation (équation aux dimensions).

The calculator has an algebraic mode, useful to create a mental bridge between a numeric computations and the algebraic formula representing it. In the preceding example in switching from numeric to algebraic mode, the entry is rewritten in algebraic mode:

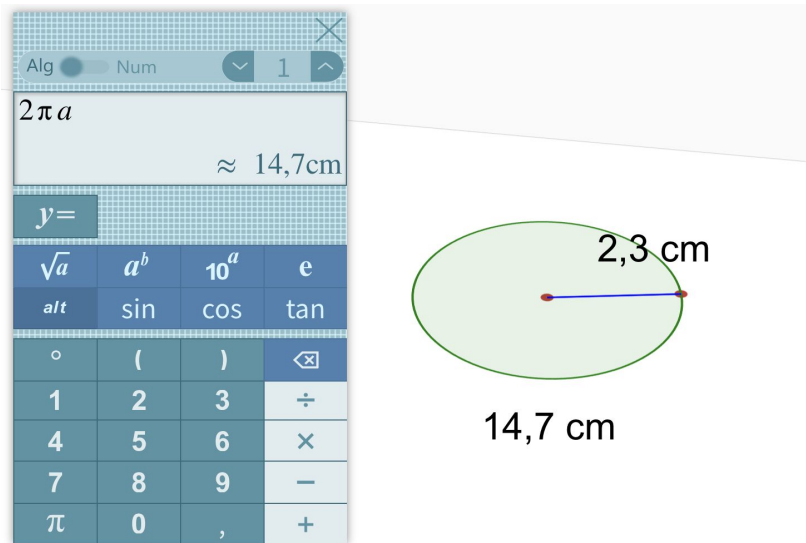
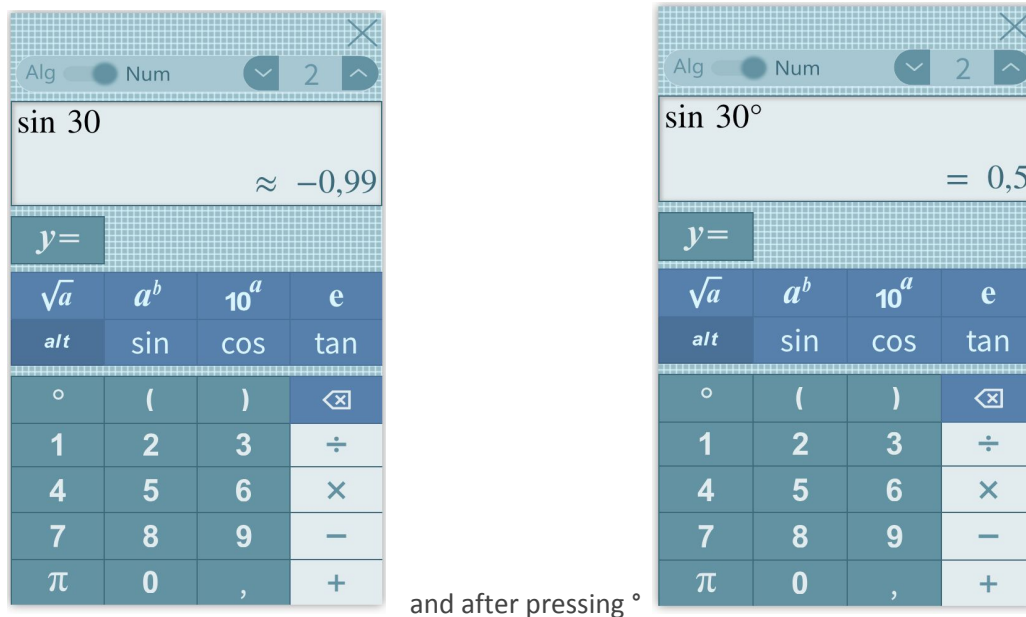


Figure 7.2.5: The algebraic mode equivalent of the arithmetic mode presented in **Figure 7.2.4**.

There is no degree/radian mode but just one mathematical (radian) mode. If the user wants to express an angle in degrees (s)he simply type the symbol $^\circ$ after the number.



Note the change of the approximate sign \approx into the equal sign $=$.

Figure 7.2.6: Transitioning between degree and radian mode.

The $y=$ button can be used to enter functions of variable x . As soon as the $y=$ button is pressed a button x appears to allow for completing an equation of type $y=f(x)$.

As soon as the entry appears as a well formed equation, the graph of the function $x \mapsto f(x)$ is dynamically displayed:

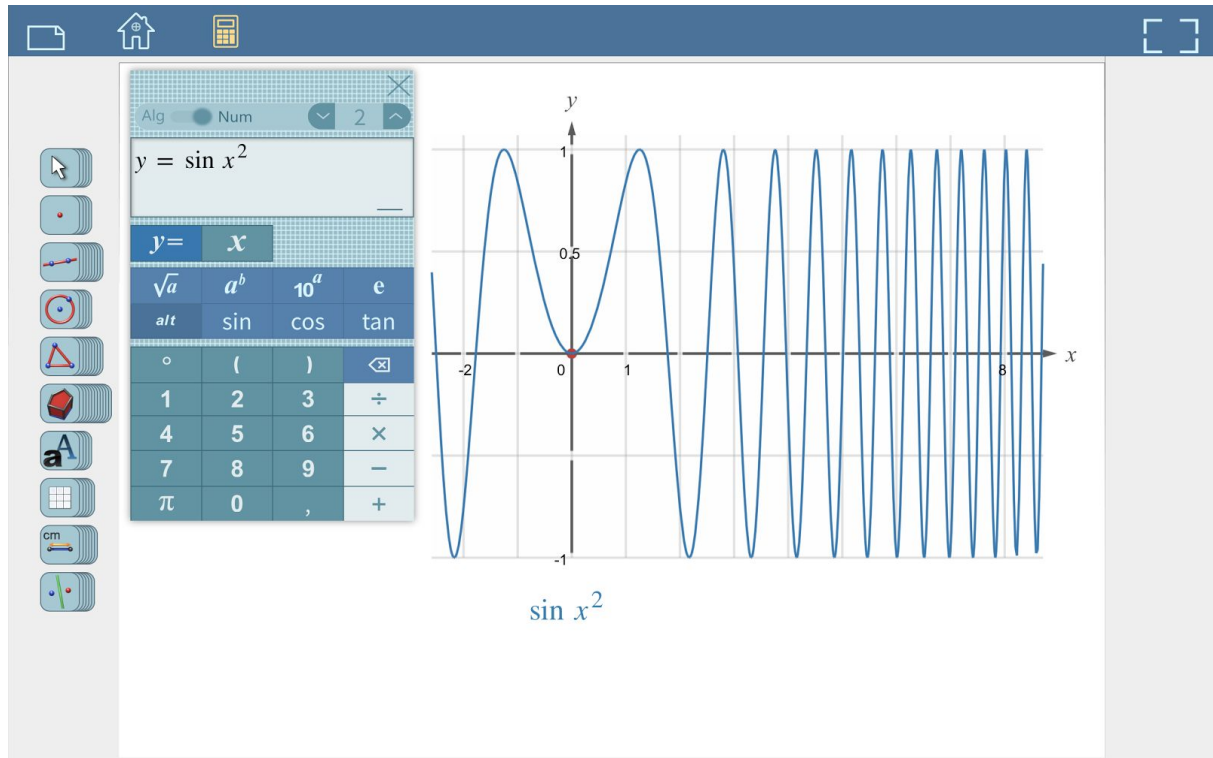


Figure 7.2.7: Demonstration of the equation visualisation in the Cabri Express tool using the calculator in algebraic mode.

Successive functions will be displayed using different colors.

Algebra and Geometry tools

To access geometry and algebraic tools from the piles located on the left, one clicks on any icon at the top of each pile. (A pile is represented as it was made of superposed cardboard icons).

The current (active) tool is highlighted, i.e., its background is white making it more eye catching.

Each pile is made of a set of semantically related tools. Please find below the various tools in their “deployed” state.

To deploy a tool one clicks on the right side of the pile (or even in the free space between the pile and the working area border). For instance, the example could be:



Figure 7.2.8: Illustration of the available tools in the geometry pile of tools.

One can then click on any buttons to activate the tool just chosen. The pile retracts automatically and the user can then performed in the working area the desired task. As a reminder the top icon of the pile is now the icon of the tool in use which will remain active and highlighted until the current action is completed (e.g. clicking on existing points or creating them “on the fly” to construct a quadrilateral.

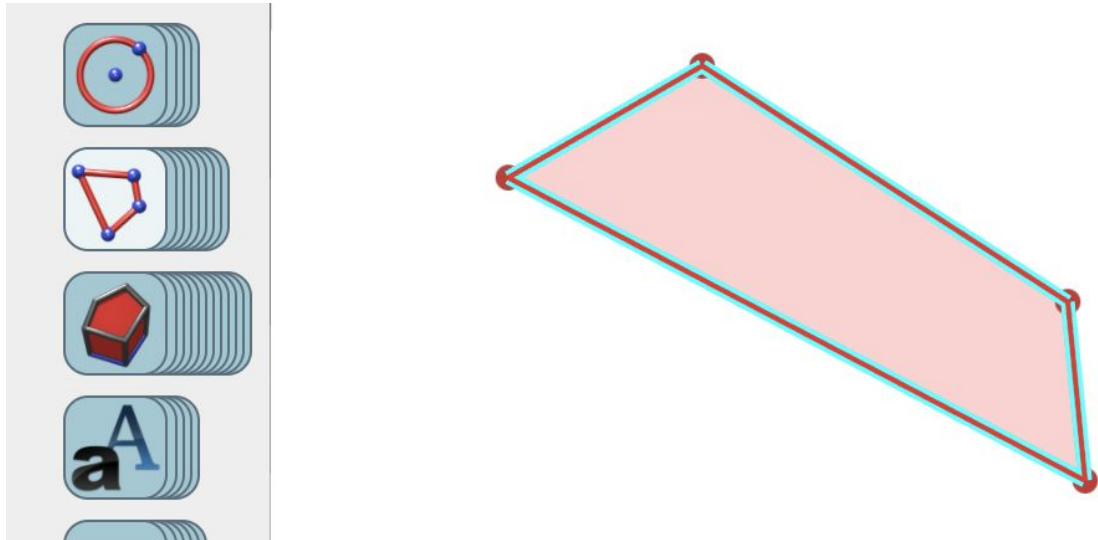


Figure 7.2.9: Demonstration of constructing a quadrilateral “on the fly”.

If one would like to create in a row several quadrilaterals, to avoid having to choose again the tool one could have double-clicked (double tap on a tablet) the the quadrilateral tool. Having done so the tool will appear within a yellow background:

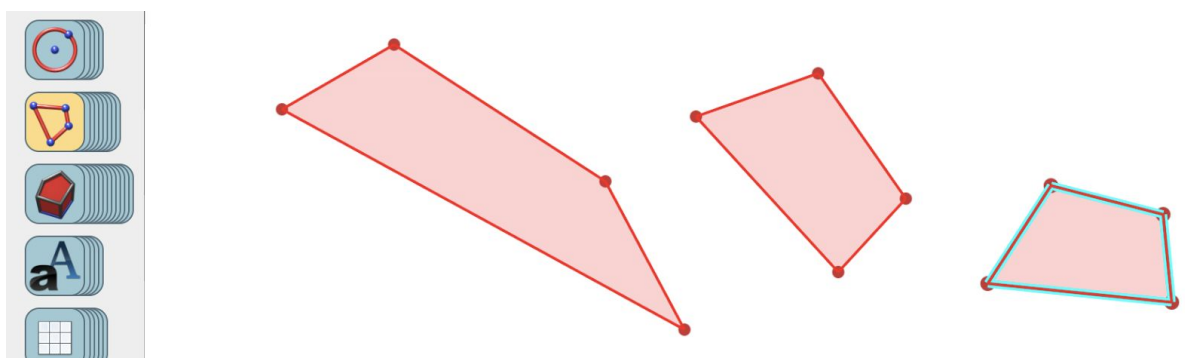


Figure 7.2.10: Demonstration of constructing several quadrilateral by double clicking the respective tool from the pile of tools.

In such a case the tool continues to be active until the user decides to choose another tool from the piles or to quit the construction state by clicking (tapping) outside the working areas.

It is possible that each label under a tool in a deployed state is hardly readable by student in the back. To overcome this difficulty it is possible to click (tap) on any label and have it displayed with a bigger font.

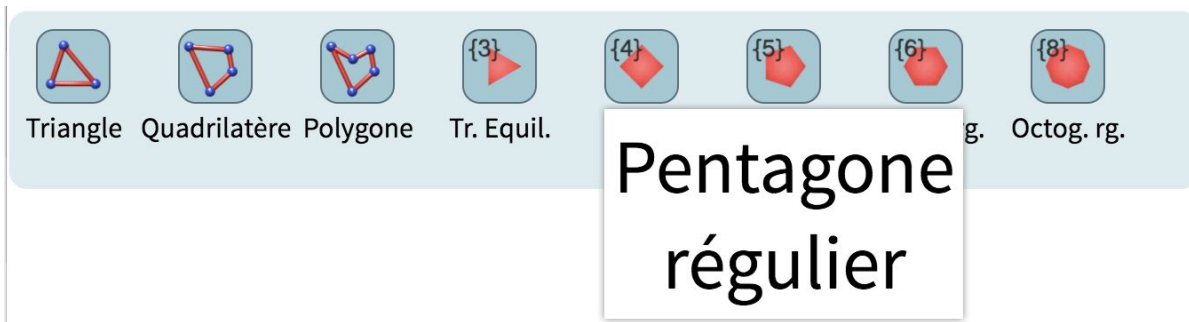


Figure 7.2.11: Illustration of magnifying the font size for assisting student in reading the short description of the selected tool.

In such a case in addition the name of the tool is rewritten in full, avoiding any abbreviation. Finally note that a long click (or tap) on the top icon of any pile, “opens” the corresponding pile.

The way to make use of the various tools is more or less self-explanatory. Actually modern (good) UIs care about making possible to use an environment without reading any manual. BTW Cabri since the 90s pioneered most of the UI aspects of standard Dynamic Geometry or Dynamic Mathematics. A large part of the work here was done during the development of both GSP (Geometer’s Sketchpad) in the US and Cabri in France. Nowadays, following the important DGS and DM UI evolution in DGS which was necessary to cope with tactile devices, practically all DM environments follow the same design at the level of individual tools. This is the reason I will simply note below what is special about Cabri tools.

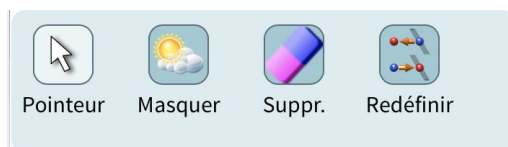


Figure 7.2.12: Content of Pile 1 - General Purpose tools.

The Mask tool can be used to hide or to make visible again any object from the working area. When this tool is active, the objects currently hidden (not visible) blink. Clicking on any of them will turn it back to visible.



Figure 7.2.13: Content of Pile 2 - Point tools.

Last tool in the pile is the Measurement Transfer tool. It make possible to construct a point at a given distance (in cm) on a linear object away from one of its points.



Figure 7.2.14: Content of Pile 3 - Straight Linear tools.

Tools here are really standard, the only thing to note is that “lines” in Cabri have representations close to the way one represents them on an actual physical notebook or page in a printed book. It means that lines do not cross all the available space on the screen (or working area) like in most of the other SW. At will it is possible to extend or reduce the size of the “drawing” on any lines, in contrast to what happens with line segment. BTW such line representation can be viewed as a reification the concept present in Euclid and later expressed in saying that a geometric line is potentially infinite.



Figure 7.2.15: Content of Pile 4 - Curved Linear tools.

It can be noted that the Compass tool is a semi physically-realistic tool, with some augmented reality features. It works best in 3D. The idea is to offer to learners a way to construct circle matching to some extent embodied cognition about circles (a circle is related to the idea of turning, an aspect entirely muted when using the “ordinary” circle tool).

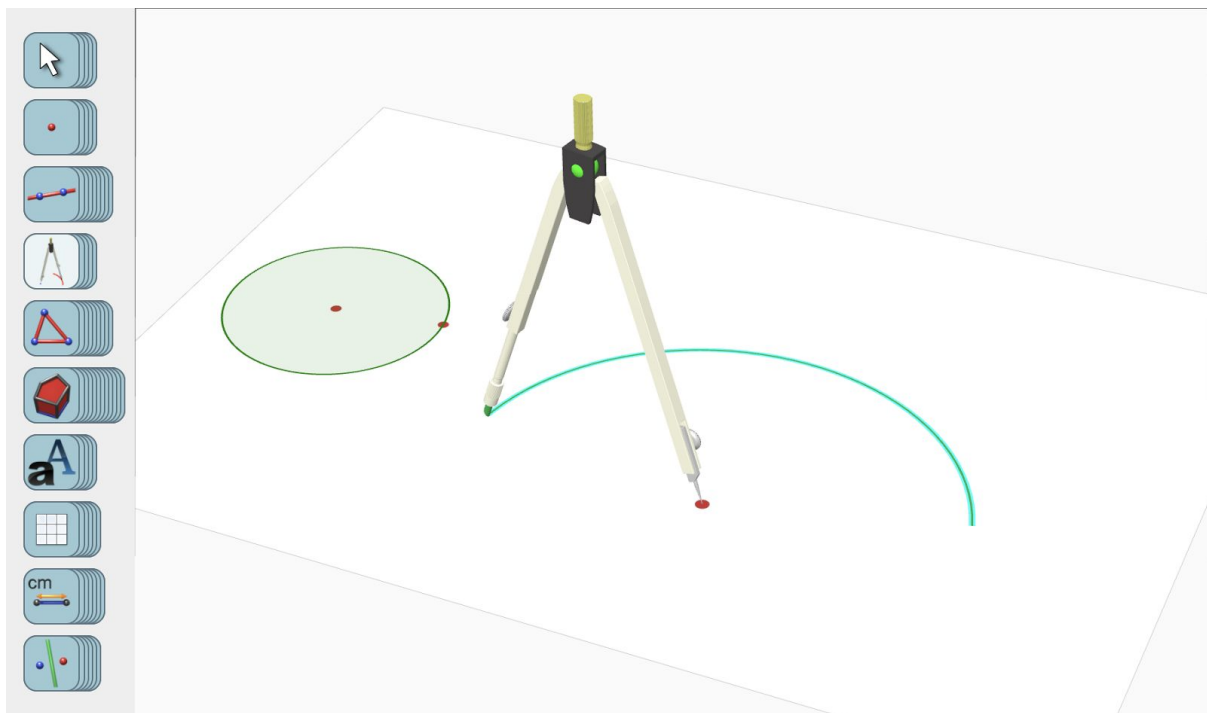


Figure 7.2.16: Illustration of using the compass tool.

To construct standard polygons and most common regular polygons the Polygon tools can be used.



Figure 7.2.17: Content of Pile 5 - Polygons tools.



Figure 7.2.18: Content of Pile 6 - Polyedra tools.

Standard 3D bodies can be obtained. To see them better the user has to tilt the plane to perceive explicitly the third dimension.

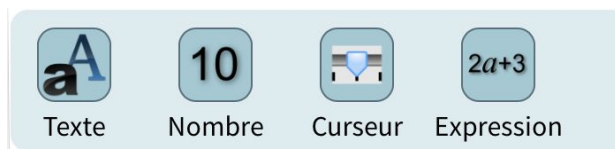


Figure 7.2.19: Content of Pile 7 - Number and Text tools.

The expression tool allows for entering mathematical formula using a straightforward syntax. As a simple rule of thumb just type on a line the formula the way you would read it. For example to obtain a fraction with $\sin(x)$ as numerator and $\cos(x)$ as denominator, type $\sin x/\cos x$ (no parenthesis are need, although it is possible to put them)

$\sin x/\cos x$

will give

$$\frac{\sin x}{\cos x}$$

Similarly

$\alpha \sin (\omega t + \phi)$

will give

$$\alpha \sin (\omega t + \phi)$$



Figure 7.2.20: Content of Pile 8 - Grid and Graphs tools.

The same page can contain several systems of axes at different scale and possibly slanted axes:

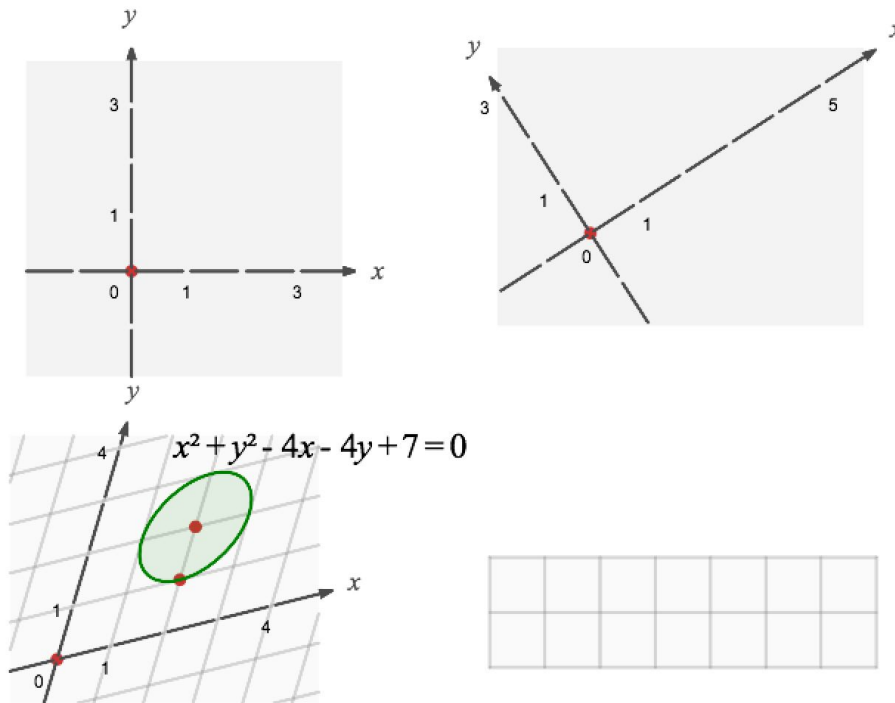


Figure 7.2.21: Illustration of several axis systems and equation visualisation.



Figure 7.2.22: Content of Pile 9 - Measurements tools.

Standard measurements tools can also be used for displaying on the visualisation axes.

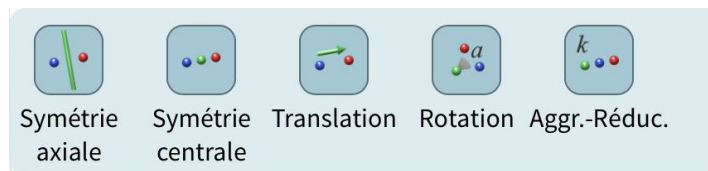


Figure 7.2.23: Content of Pile 10 - Transformation tools.

To perform a transformation, e.g. to translate of polygon, one has first to indicate, in any order making sense, the given polygon. In case of ambiguity the last object to be clicked is the object to be transformed (other objects can be viewed as parameters of the transformation). In the case above, the polygon can be clicked at any time, because the polygon itself cannot be a “parameter” of the translation. Parameters of a translation can be just a vector or possibly the ordered pair of a first and a second point. So the action could be performed, among other meaningful possibilities, when clicking first on a point as first point of a pair, a second point and finally any object, here a polygon.