

USING CSOUND IN JUPYTER NOTEBOOKS

Joachim Heintz

FMSBW

HMTM Hannover, Germany

joachim.heintz@hmtm-hannover.de

ABSTRACT

What has been planned as live workshop has now become a video presentation. It shows how Csound [1] can be run in a Jupyter Notebook [2] via ctsound [3], a Python binding to the Csound API. In particular the communication between a Graphical User Interface, built in Python, and the ICsound class is demonstrated on simple examples.

WORKSHOP CONTENT

1 ICsound

ICsound is a layer which is built upon ctsound. It runs the Csound instance in a separate thread which allows a flawless communication between the Csound instance and the Python GUI. And it offers some methods which simplify creating and deleting Csound instances, starting and stopping the engine, or sending and receiving values via software channels.

Basic usage of ICsound is shown, including the integrated table display.

Load the csoundmagics which include the ICsound module:

```
%load_ext csoundmagics
```

Create an instance of ICsound:

```
cs = ICsound()
```

Look which methods are defined here, and inherited from ctsound:

```
help(cs)
```

Use the csoundmagics to send code to the running Csound instance,

here a table of 32 points with a Gauss random function:

```
%csound
```

```
i0 = ftgen(1, 0, 32, 21, 6)
```

Plot this table with the integrated Matplotlib view:

```
cs.plotTable(1)
```

Stop the Csound engine:

```
cs.stopEngine()
```

Delete this instance of Csound:

```
del cs
```

2 PySimpleGUI

PySimpleGUI [4] is used here as GUI toolkit. It has two basic code layers: The layout and the event loop. In the layout part, each list represents a horizontal alignment of one or more elements (widgets). In the event loop, incoming events are read, and a break condition is defined (usually closing the window) to leave the loop.

Load the PySimpleGUI module:

```
import PySimpleGUI as sg
```

Create two horizontal lines; the first with a text box, the second with two buttons:

```
layout = [[sg.Input(key='INPUT')],  
          [sg.Button('Read'), sg.Exit()]]
```

Create the window with a title and the layout:

```
window = sg.Window('Please type', layout)
```

Create the event loop and print the events once the Read button has been pressed:

```
while True:  
    event, values = window.read()  
    print(event, values)  
    if event==None or event=='Exit':  
        break
```

Close the window once the loop has been left:

```
window.close()
```

3 GUI to Csound

A basic example is given: A slider from PySimpleGUI is received as frequency input for a Csound oscillator. This is the code on the Csound side:

```
orc = ""  
instr 1  
    kFreq = chnget:k("freq")  
    aOut = poscil:a(.2, kFreq)  
    out(aOut, aOut)  
endin  
schedule(1, 0, -1)  
""  
cs.sendCode(orc)
```

On the PySimpleGUI side, the main job is to send the slider values to the Csound instance. This is done with the `setControlChannel()` method:

```
cs.setControlChannel('freq', values['FREQ'])
```

It is shown how the code should organize the instantiation and deletion of the Csound instance. Another example is added, in which a Browse button on the PySimpleGUI side loads a sound file which is then played back by Csound.

4 Csound to GUI

If a GUI is not used to send control values, but to give visual feedback for anything which happens inside Csound, the main difference on the PySimpleGUI side is the definition of a

```
timeout (in milliseconds) which tells the event loop the update
rate:
while True:
    event, values = window.read(timeout=100)
```

The values from Csound are received via the `cs.channel()` method, and the window elements are updated like this:

```
window['LINE'].update(cs.channel('line')[0])
```

CONCLUSIONS

ICsound and PySimpleGUI allow a flexible communication between a Python GUI toolkit and the Csound audio engine. Based on the concept shown here more complex applications can be built easily.

ACKNOWLEDGMENTS

Many thanks to François Pinot for writing the `ctcsound` module, integrating Andrés Cabrera's ICsound and supporting this workshop in many ways.

LINKS

The video:

<https://vimeo.com/480443886>

The Jupyter notebook:

https://github.com/csound/ctcsound/blob/master/cookbook/drafts/heintz_lac2020.ipynb

REFERENCES

- [1] <https://csound.com/>
- [2] <https://jupyter.org/>
- [3] <https://github.com/csound/ctcsound>
- [4] <https://github.com/PySimpleGUI>