JUCE

PROJECT

AMBISONIC PLUG-INS W/ JUCE

LAC 2019

DATE MAR 26, 2019

PRESENTER GABRIEL ZALLES

requirements

- Recent installation of JUCE (using v5.4.3)
- Xcode (macOS) or Visual Studio (Windows)
  - Using Xcode (v 10.1)
- Suggested: Reaper DAW
  - Great for multichannel audio
  - Free to try (“unlimited”)
  - Open source and affordable
- Downloadable: image file(s), audio file(s)

requirements (note)

- In the latest release of JUCE there is no need to download the Steinberg VST3 SDK.
- The VST3 SDK has been packaged inside JUCE.
- You can use Spatial Workstation by Facebook for free binaural decoding. Or make your own!

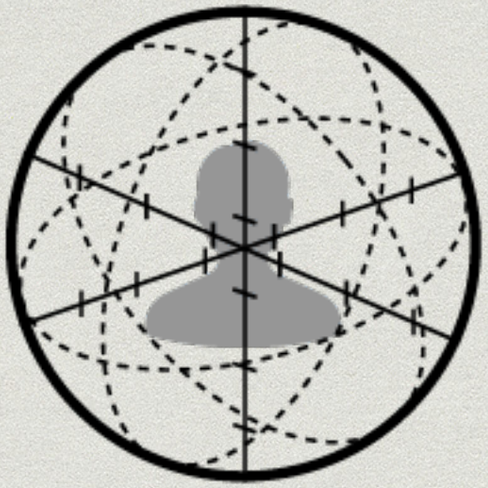
why…

- learn more ambisonics (one of the most popular spatial audio technologies)
- learn about plug-in development using JUCE (one of the most popular VST frameworks)
- acquire skill needed to perform sophisticated experiments
  - no more black boxes

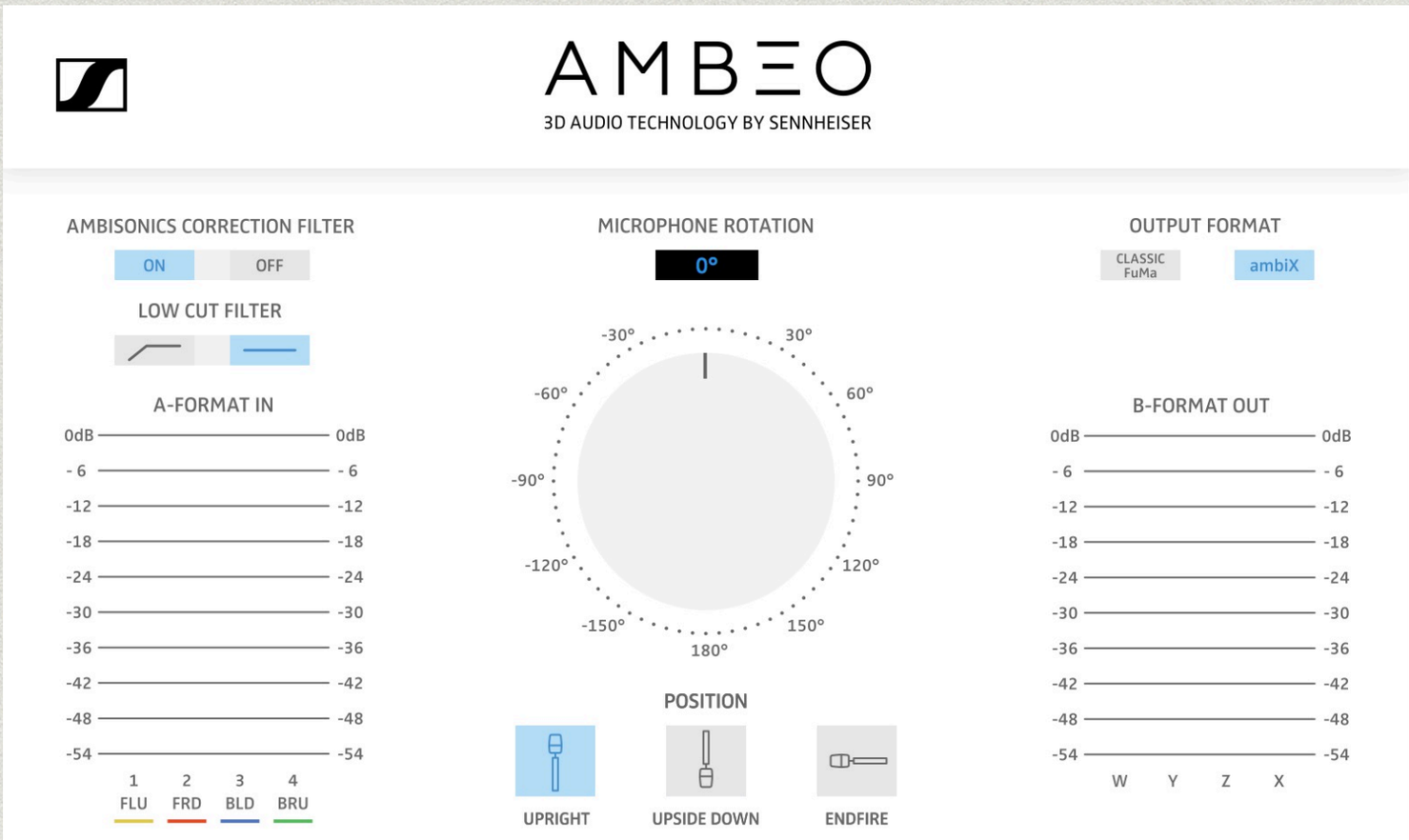
overview

- simple FOA mic encoder (A-to-B format)
- simple FOA panner (with interactive GUI)
- simple FOA decoder

Ambisonics

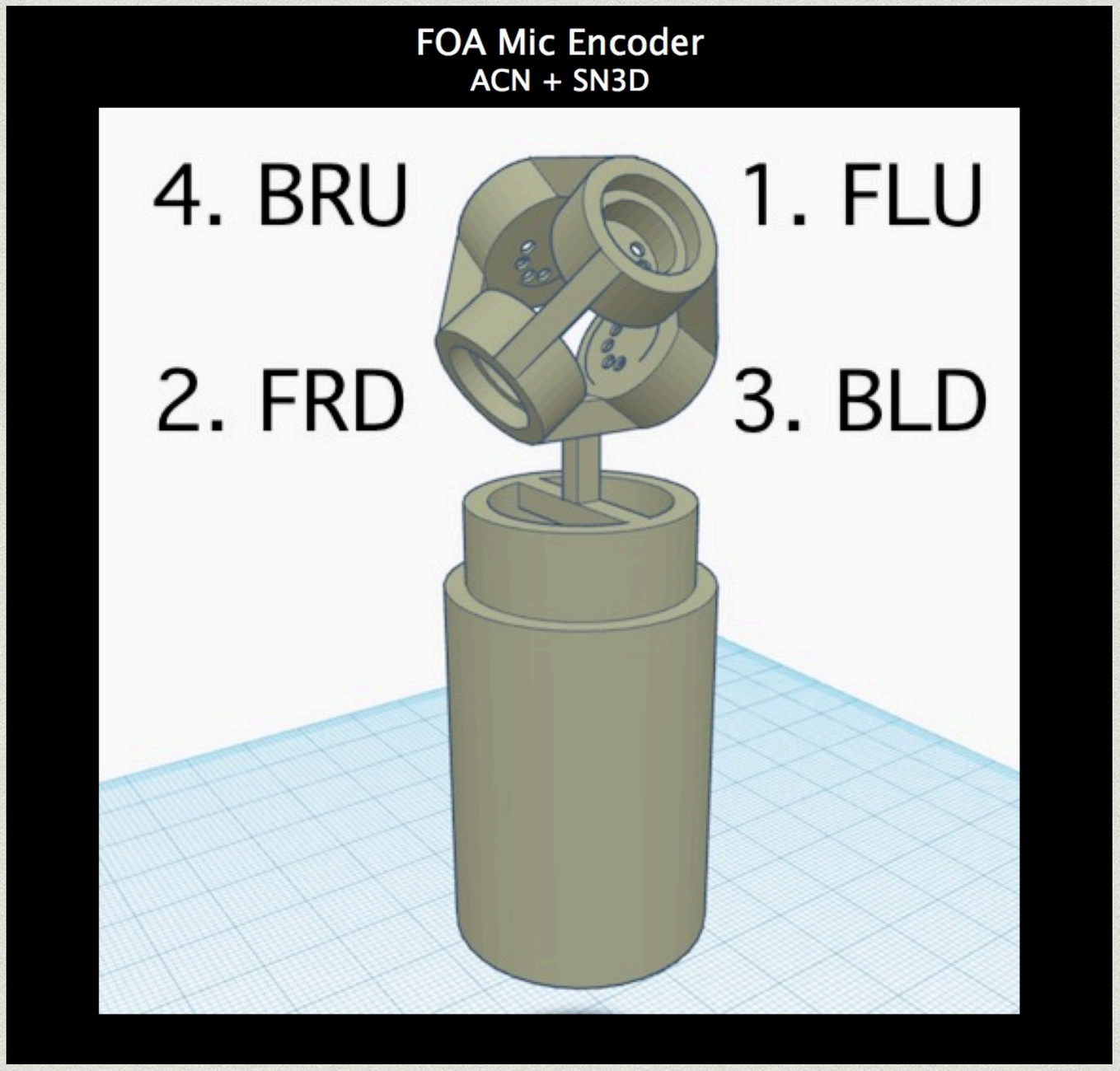


“ideal” mic encoder



- correction filter
- low cut filter
- input monitoring
- rotation (azimuth)
- rotation (elevation)
- output format
- output monitoring

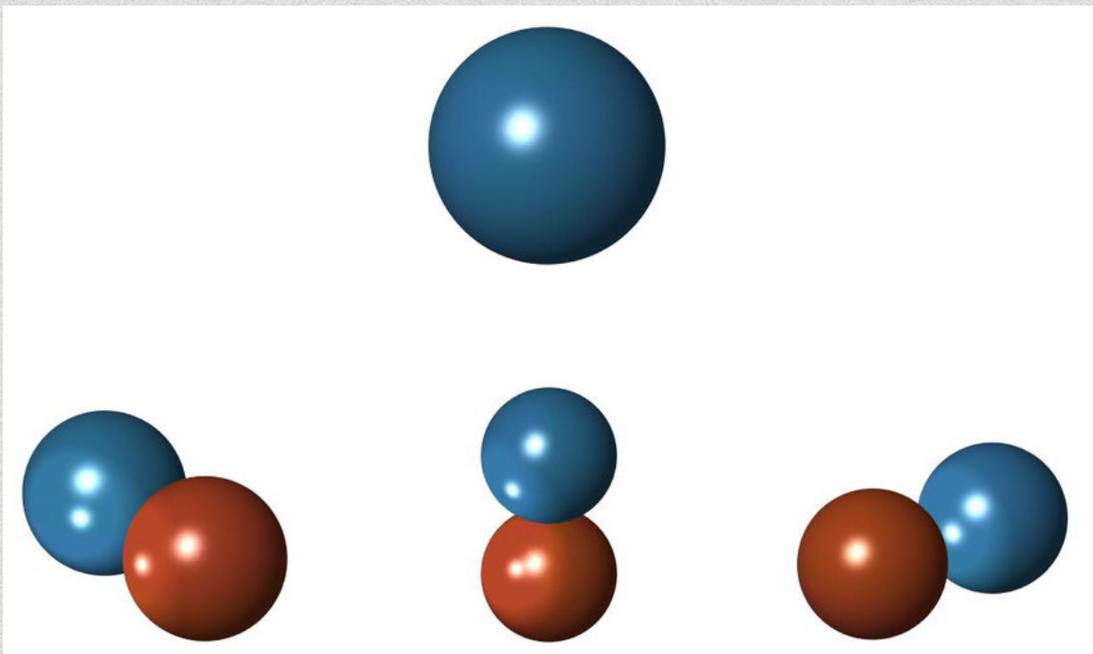
our simple mic encoder



- reference image
- no buttons/sliders
- intended to be a prototype

our mic encoder (overview)

- cache image
- resize canvas
- draw text and image
- set IO (4 in 4 out)
  - buses “false”
- use formula
  - define pi

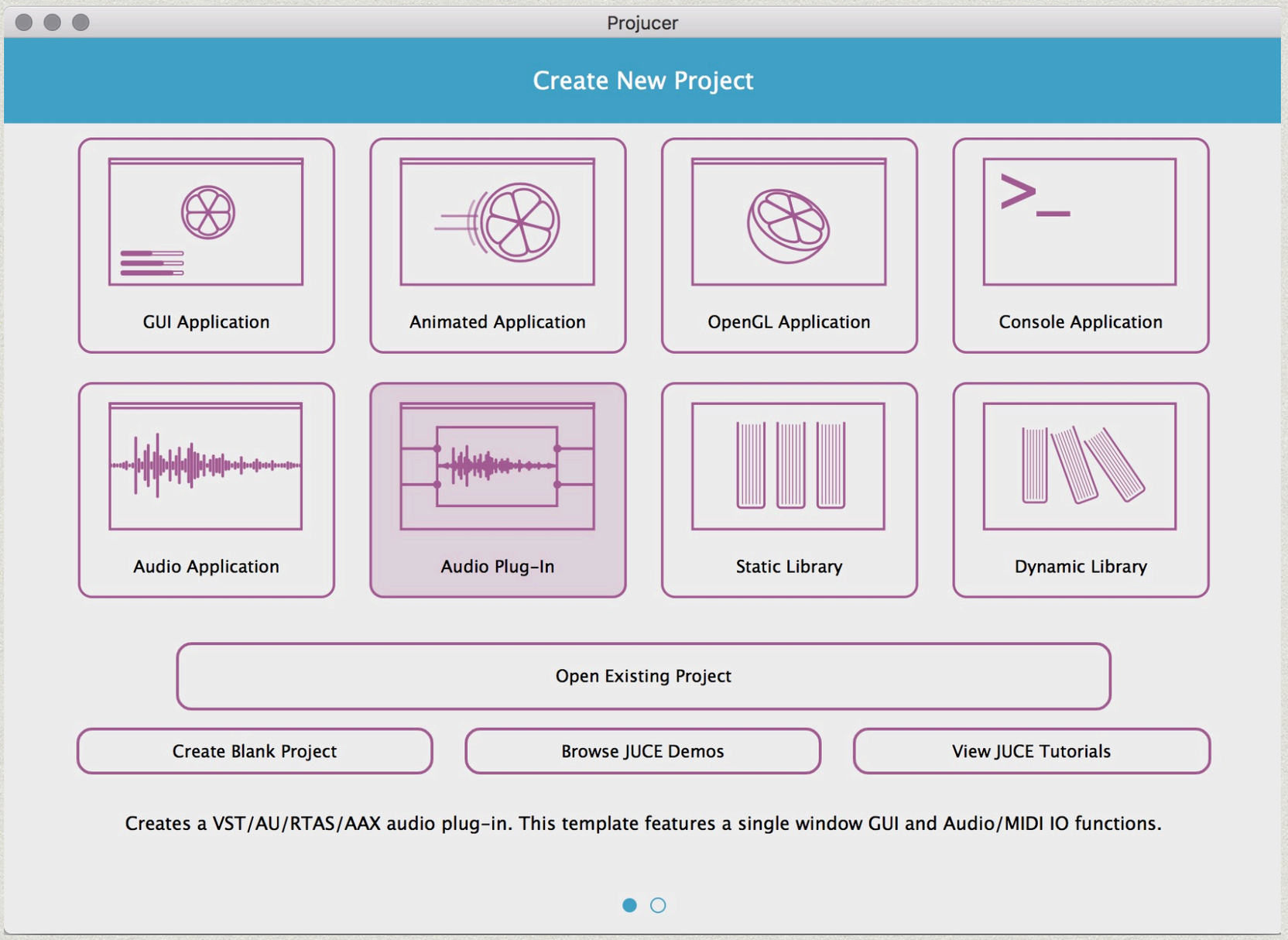


FOA Mic Encoding Formula

$$W = FLU + FRD + BLD + BRU$$
$$X = FLU + FRD - BLD - BRU$$
$$Y = FLU - FRD + BLD - BRU$$
$$Z = FLU - FRD - BLD + BRU$$

JUCE basics

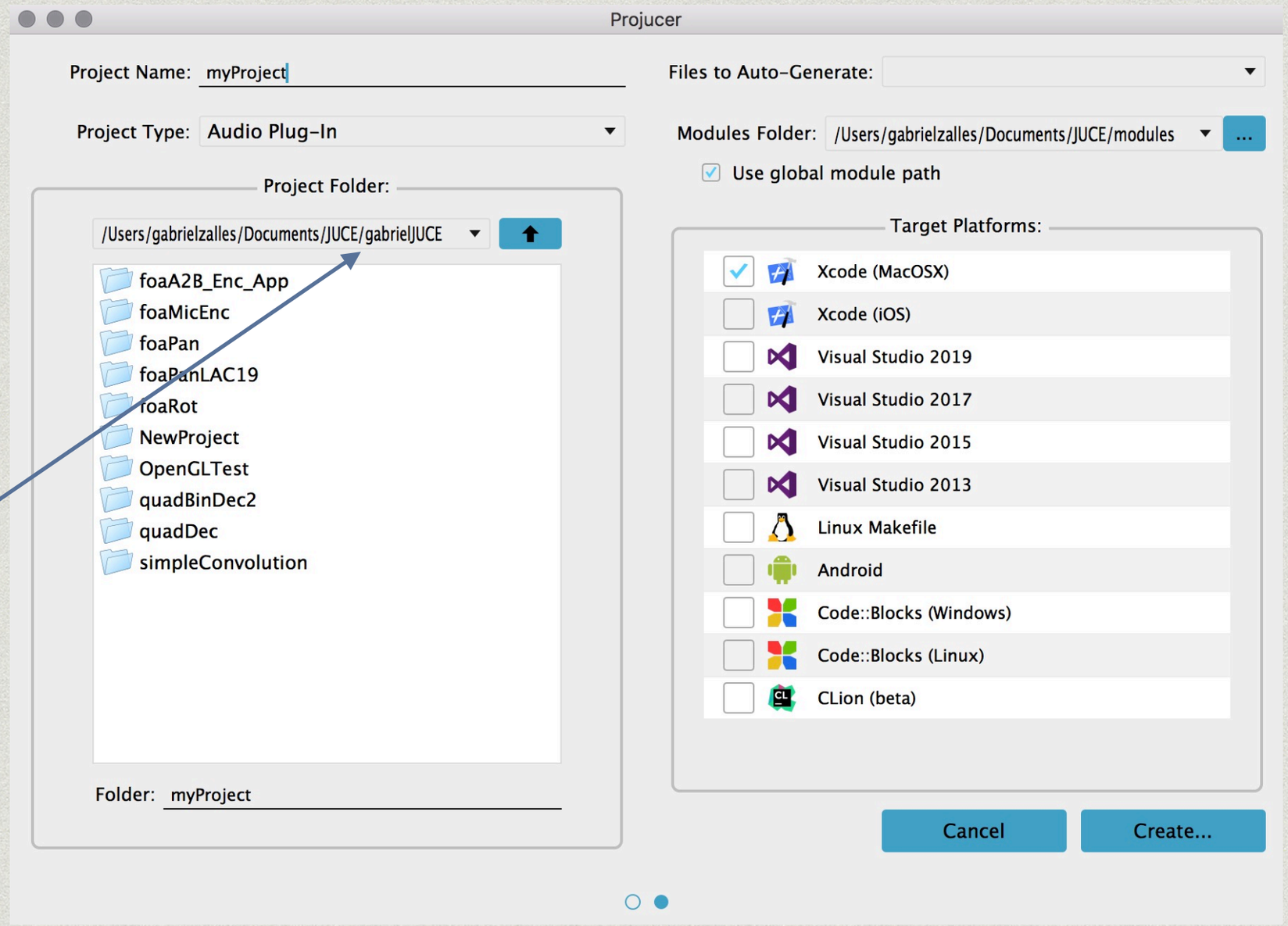
Open Projucer. Select Audio Plug-in.



JUCE basics

Project name: FOAmicEnc

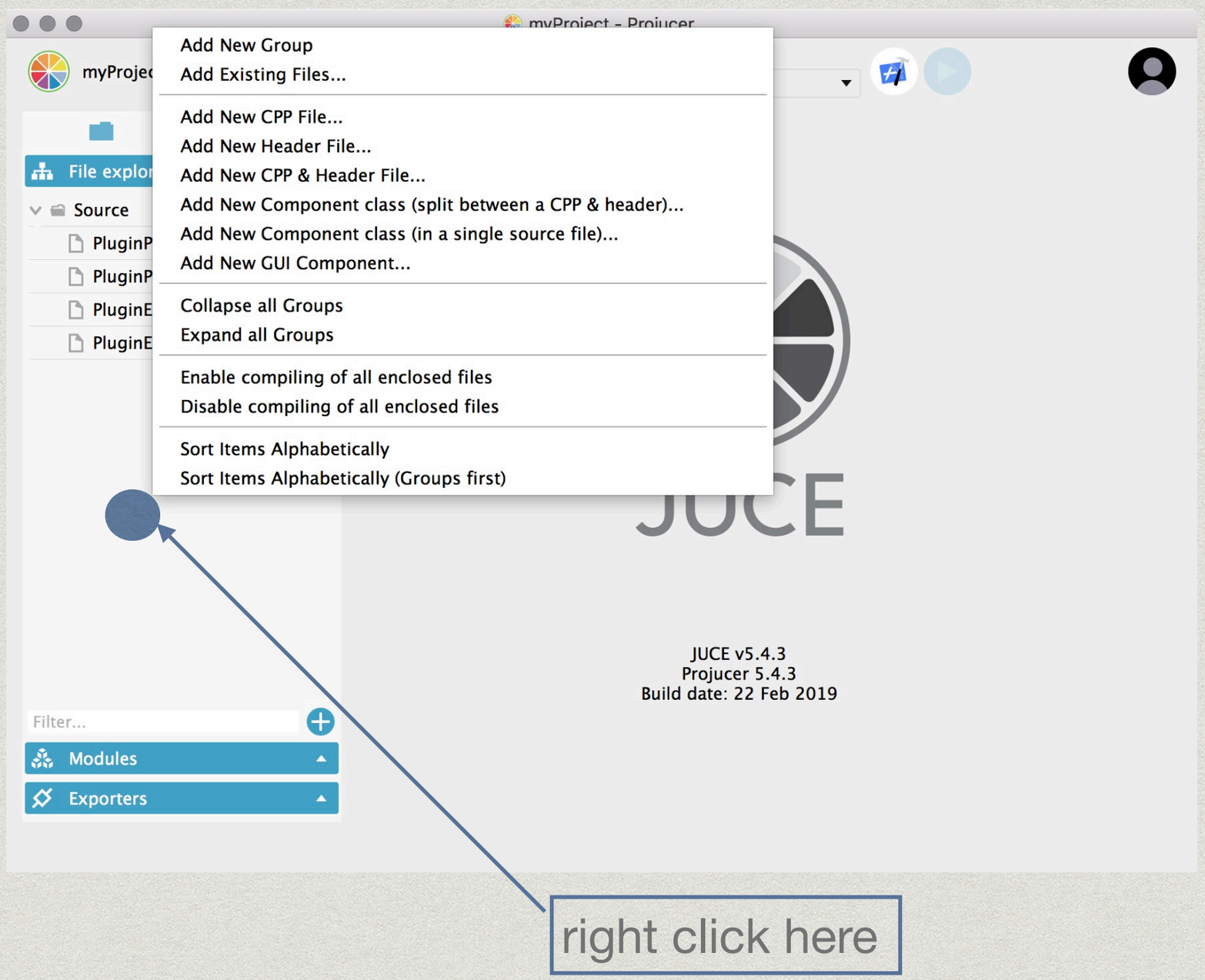
Type in project name. Create.



this is bad

tip: juce_projects folder outside main JUCE folder

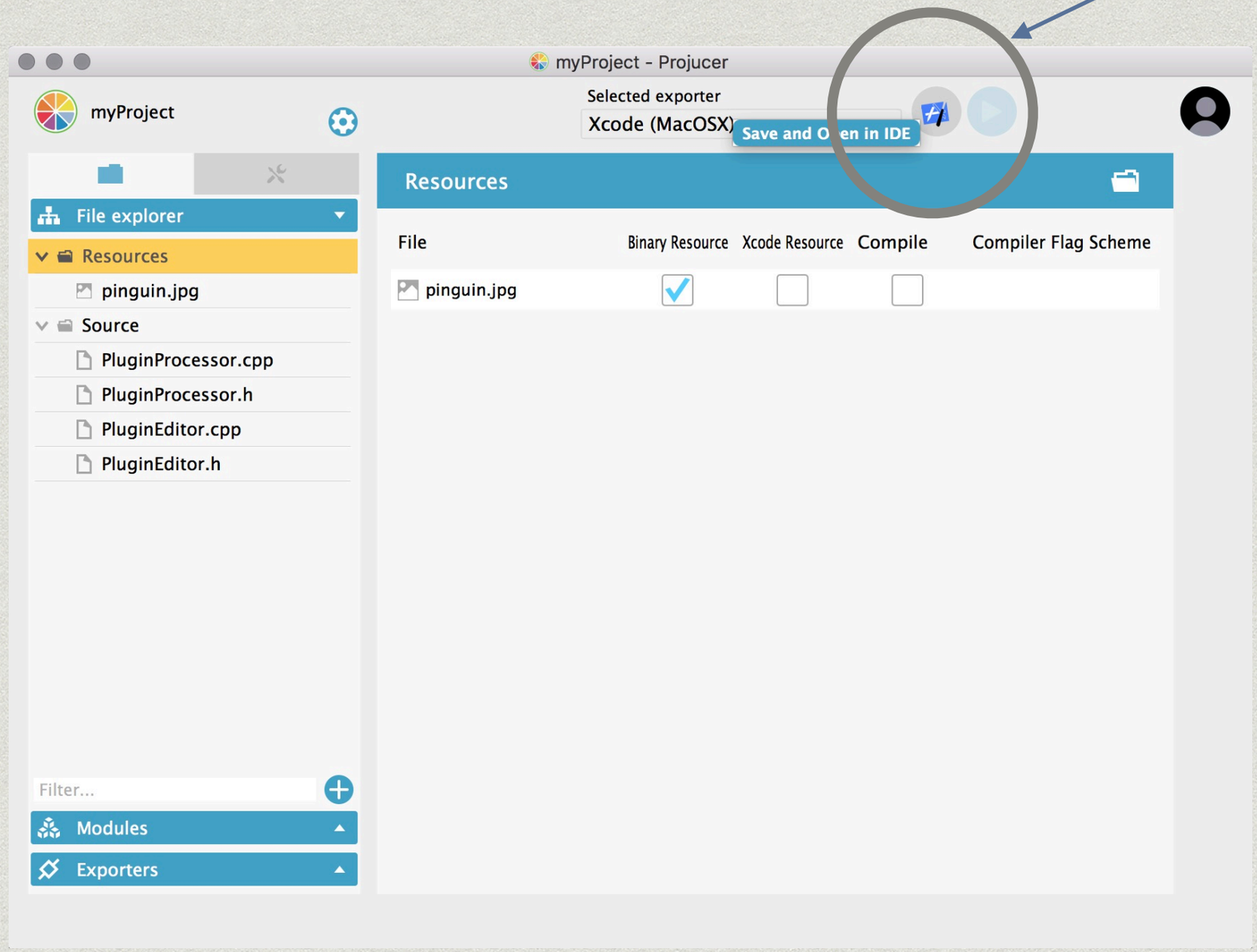
JUCE basics



- Right click under PluginEditor.h
- Add New Group
- I call it “Resources”

JUCE basics

converts resources to binary data



- Add images, audio, 3D models.
- Make sure binary resource is checked.
- Save and open in IDE

JUCE Library Code -> BinaryData.h

JUCE basics

myProject > JUCE Library Code > BinaryData.h > No Selection

```
/*
    ==============================================================================
    ============

    This is an auto-generated file: Any edits you make may be overwritten!

*/

#pragma once

namespace BinaryData
{
    extern const char*   pinguin_jpg;
    const int            pinguin_jpgSize = 90431;

    // Number of elements in the namedResourceList and originalFileNames arrays.
    const int namedResourceListSize = 1;

    // Points to the start of a list of resource names.
    extern const char* namedResourceList[];

    // Points to the start of a list of resource filenames.
    extern const char* originalFilenames[];

    // If you provide the name of one of the binary resource variables above, this function will
    // return the corresponding data and its size (or a null pointer if the name isn't found).
    const char* getNamedResource (const char* resourceNameUTF8, int& dataSizeInBytes);

    // If you provide the name of one of the binary resource variables above, this function will
    // return the corresponding original, non-mangled filename (or a null pointer if the name isn't found).
    const char* getNamedResourceOriginalFilename (const char* resourceNameUTF8);
}
```

Identity and Type
Name: BinaryData.h
Type: Default - C Header
Location: Relative to Project
../../JuceLibraryCode/BinaryData.h
Full Path: /Users/gabrielzalles/Documents/JUCE/gabrielJUCE/myProject/JuceLibraryCode/BinaryData.h

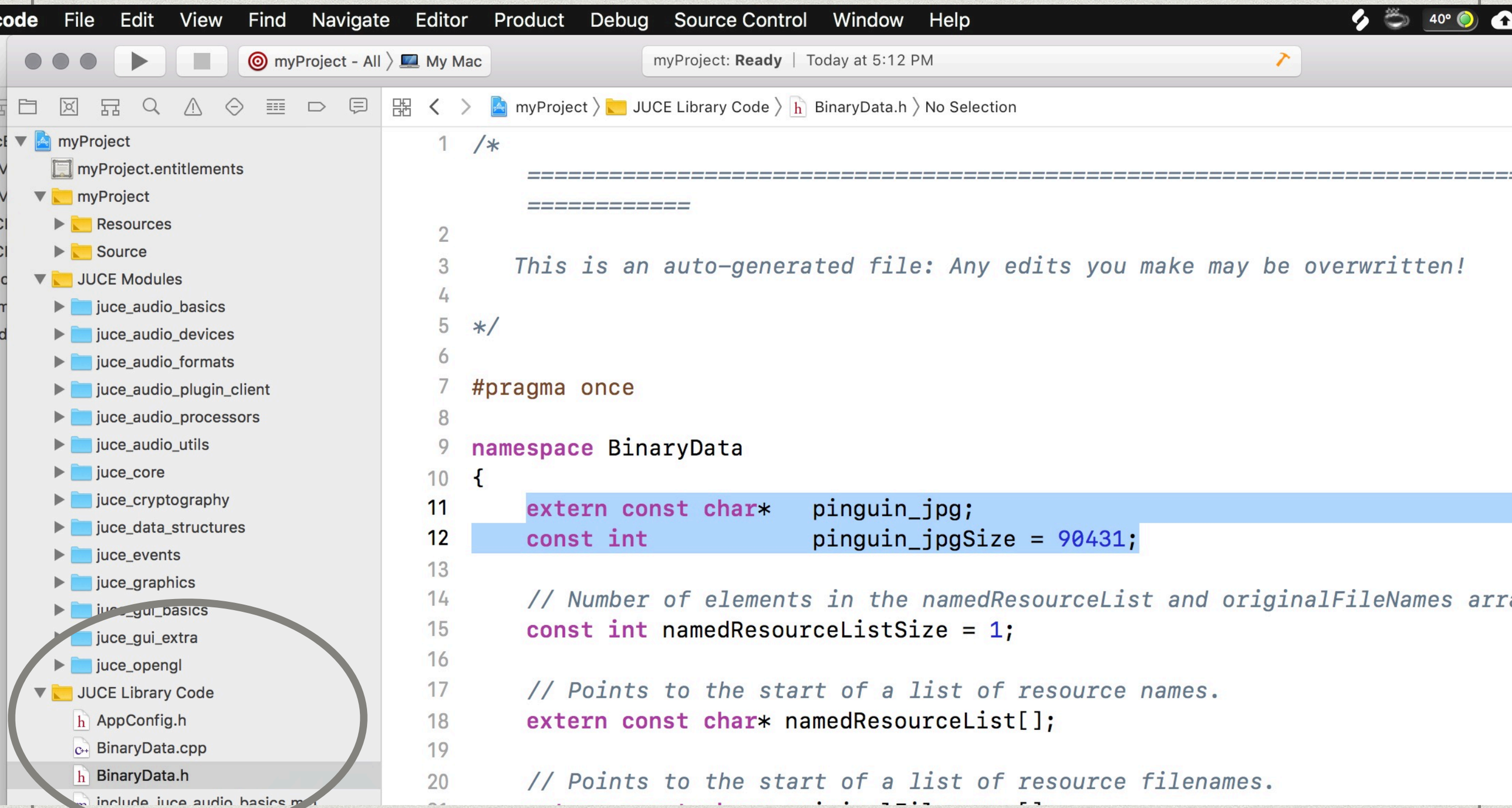
On Demand Resource Tags
Add to a target to enable tagging

Target Membership
myProject - All
myProject - VST3
myProject - AU
myProject - Standalone Plugin
myProject - Shared Code

Text Settings
Text Encoding: No Explicit Encoding
Line Endings: No Explicit Line Endings
Indent Using: Spaces
Widths: 4 Tab, 4 Indent
Wrap lines

JUCE Library Code -> BinaryData.h

JUCE basics



our mic encoder

- cache image ✓
- resize canvas
- draw text and image
- set IO (4 in 4 out)
  - buses “false”
- use formula
  - define pi

Note: no longer working with myProject, moved to foaMicEnc

Source -> PluginEditor.cpp

our mic encoder

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

foaMicEnc - All 〉 My Mac

foaMicEnc | Build foaMicEnc - All: **Succeeded** | Today at 3:22 PM

foaMicEnc 〉 foaMicEnc 〉 Source 〉 PluginEditor.cpp 〉 FoaMicEncAudioProcessorEditor::FoaMicEncAudioProcessorEditor (FoaMicEncAudioProcessor& p)

- foaMicEnc
  - foaMicEnc.entitlements
  - foaMicEnc
    - Resources
    - Source
      - PluginProcessor.cpp
      - PluginProcessor.h
      - PluginEditor.cpp
      - PluginEditor.h
  - JUCE Modules
  - JUCE Library Code
  - Resources
  - Frameworks
  - Products

```cpp
/*
  ==============================================================================

    This file was auto-generated!

    It contains the basic framework code for a JUCE plugin editor.

  ==============================================================================
*/

#include "PluginProcessor.h"
#include "PluginEditor.h"

//==============================================================================
FoaMicEncAudioProcessorEditor::FoaMicEncAudioProcessorEditor (FoaMicEncAudioProcessor& p)
    : AudioProcessorEditor (&p), processor (p)
{
    //resized from default 400 by 300
    setSize (600, 600);
}

FoaMicEncAudioProcessorEditor::~FoaMicEncAudioProcessorEditor()
{
}

```

our mic encoder

- cache image ✓
- resize canvas ✓
- draw text and image
- set IO (4 in 4 out)
  - buses “false”
- use formula
  - define pi

Note: no longer working with myProject, moved to FOAmicEnc

Source -> PluginEditor.cpp

our mic encoder

- edit **void** FoaMicEncAudioProcessorEditor::paint (Graphics& g)
  - this will determine the look of the plug-in
  - you can copy paste the code from the website
  - well focus on the audio
  - http://gabrielzalles.com/LAC19_workshop/

project name must be foaMicEnc

our mic encoder

- foaMicEnc
  - Resources
  - Source
    - PluginProcessor.cpp
    - PluginProcessor.h
    - PluginEditor.cpp
    - PluginEditor.h
  - JUCE Modules
  - JUCE Library Code
  - Resources
  - Frameworks
  - Products

```cpp
FoaMicEncAudioProcessorEditor::~FoaMicEncAudioProcessorEditor()
{
}

//==============================================================================
void FoaMicEncAudioProcessorEditor::paint (Graphics& g)
{

    // fill the whole window black
    g.fillAll (Colours::black);
    // set the current drawing colour to white
    g.setColour (Colours::white);
    // set the font size and draw text to the screen
    g.setFont (22.0f);

    //set some values for fitted text functions
    int titleHeight = 50;
    int subTitleOffset = 20;

    //draw the title
    g.drawFittedText ("FOA Mic Encoder", 0, 0, getWidth(), titleHeight, Justification::centred, 1);

    //change the font size
    g.setFont (18.0f);

    //draw the subtitle
    g.drawFittedText ("ACN + SN3D", 0, subTitleOffset, getWidth(), titleHeight, Justification::centred, 1);

    //draw the image from cache
    Image background = ImageCache::getFromMemory (BinaryData::mark2_JPG, BinaryData::mark2_JPGSize);

    //set values for image offset
    int imageOffsetX = 56;
    int imageOffsetY = 60;

    //draw the image on the canvas
    g.drawImageAt (background, imageOffsetX, imageOffsetY);

}
```

our mic encoder

- cache image ✓
- resize canvas ✓
- draw text and image ✓
- set IO (4 in 4 out)
  - buses “false”
- use formula
  - define pi

Note: no longer working with myProject, moved to FOAmicEnc

Source -> PluginProcessor.cpp

our mic encoder

foaMicEnc / Source / PluginProcessor.cpp

```
 *  This file was auto-generated!

 *  It contains the basic framework code for a JUCE plugin processor.

 *  ==============================================================================
 */

#include "PluginProcessor.h"
#include "PluginEditor.h"

//==============================================================================
FoaMicEncAudioProcessor::FoaMicEncAudioProcessor()
#ifndef JucePlugin_PreferredChannelConfigurations
: AudioProcessor (BusesProperties()
#if ! JucePlugin_IsMidiEffect
#if ! JucePlugin_IsSynth
                  //change the input to FOA
                  .withInput  ("Input",  AudioChannelSet::ambisonic(1), true)
#endif
                  //change the output to FOA
                  .withOutput ("Output", AudioChannelSet::ambisonic(1), true)
#endif
                  )
#endif
{
    //VTS not needed for this plug-in
}

FoaMicEncAudioProcessor::~FoaMicEncAudioProcessor()
{
}

//==============================================================================
const String FoaMicEncAudioProcessor::getName() const
{
    return JucePlugin_Name;
}

bool FoaMicEncAudioProcessor::acceptsMidi() const
{
#if JucePlugin_WantsMidiInput
    return true;
```

1 = first order

our mic encoder

- cache image ✓
- resize canvas ✓
- draw text and image ✓
- set IO (4 in 4 out) ✓
  - buses “false”
- use formula
  - define pi

Note: no longer working with myProject, moved to FOAmicEnc

our mic encoder

- buses?
  - often used for side chain
  - can also be used to dynamically change IO
    - not supported by many DAWs
  - in our simple VST we will <u>not</u> support **bus layout**

Source -> PluginProcessor.cpp

delete bus layout support

foaMicEnc - All 〉 My Mac | foaMicEnc | Build foaMicEnc - All: **Succeeded** | Today at 6:01 PM

foaMicEnc 〉 foaMicEnc 〉 Source 〉 PluginProcessor.cpp 〉 FoaMicEncAudioProcessor::isBusesLayoutSupported (const BusesLayout& layouts)

- foaMicEnc
  - foaMicEnc.entitlements
  - foaMicEnc
    - Resources
    - Source
      - PluginProcessor.cpp
      - PluginProcessor.h
      - PluginEditor.cpp
      - PluginEditor.h
  - JUCE Modules
  - JUCE Library Code
  - Resources
  - Frameworks
  - Products

```cpp
    pi = 4.0f * atan(1.0f);
}

void FoaMicEncAudioProcessor::releaseResources()
{
    // When playback stops, you can use this as an opportunity to free up any
    // spare memory, etc.
}

#ifndef JucePlugin_PreferredChannelConfigurations
bool FoaMicEncAudioProcessor::isBusesLayoutSupported (const BusesLayout& layouts) const
{

#if JucePlugin_IsMidiEffect
    ignoreUnused (layouts);
    return true;
#else
    // This is the place where you check if the layout is supported.
    // In this template code we only support mono or stereo.
    if (layouts.getMainOutputChannelSet() != AudioChannelSet::mono()
        && layouts.getMainOutputChannelSet() != AudioChannelSet::stereo())
        return false;

    // This checks if the input layout matches the output layout
#if ! JucePlugin_IsSynth
    if (layouts.getMainOutputChannelSet() != layouts.getMainInputChannelSet())
        return false;
#endif

    return true;
#endif


}
#endif

void FoaMicEncAudioProcessor::processBlock (AudioBuffer<float>& buffer, MidiBuffer& midiMessages)
{
    //default JUCE code, leave as is...
    ScopedNoDenormals noDenormals;
    auto totalNumInputChannels  = getTotalNumInputChannels();
    auto totalNumOutputChannels = getTotalNumOutputChannels();

    //clear buffer
    for (auto i = totalNumInputChannels; i < totalNumOutputChannels; ++i)
        buffer.clear (i, 0, buffer.getNumSamples());

    /* DESCRIPTION
```

Identity and Type

Name: PluginProcessor.cpp
Type: Default - C++ Source
Location: Relative to Project
../../Source/PluginProcessor.cpp
Full Path: /Users/gabrielzalles/Documents/JUCE/gabrielJUCE/foaMicEnc/Source/PluginProcessor.cpp

On Demand Resource Tags

Only resources are taggable

Target Membership

- foaMicEnc - All
- foaMicEnc - VST3
- foaMicEnc - AU
- foaMicEnc - Standalone Plugin
- foaMicEnc - Shared Code

Text Settings

Text Encoding: No Explicit Encoding
Line Endings: No Explicit Line Endings
Indent Using: Spaces
Widths: Tab 4, Indent 4
Wrap lines

Source -> PluginProcessor.cpp

return false

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

foaMicEnc - All 〉 My Mac | foaMicEnc | Build foaMicEnc - All: **Succeeded** | Today at 6:06 PM

foaMicEnc 〉 foaMicEnc 〉 Source 〉 PluginProcessor.cpp 〉 FoaMicEncAudioProcessor::isBusesLayoutSupported (const BusesLayout& layouts)

- foaMicEnc
  - foaMicEnc.entitlements
  - foaMicEnc
    - Resources
    - Source
      - PluginProcessor.cpp
      - PluginProcessor.h
      - PluginEditor.cpp
      - PluginEditor.h
  - JUCE Modules
  - JUCE Library Code
  - Resources
  - Frameworks
  - Products

```cpp
void FoaMicEncAudioProcessor::releaseResources()
{
    // When playback stops, you can use this as an opportunity to free up any
    // spare memory, etc.
}

#ifndef JucePlugin_PreferredChannelConfigurations
bool FoaMicEncAudioProcessor::isBusesLayoutSupported (const BusesLayout& layouts) const
{

    return false;

}
#endif

void FoaMicEncAudioProcessor::processBlock (AudioBuffer<float>& buffer, MidiBuffer& midiMessages)
{
    //default JUCE code, leave as is...
    ScopedNoDenormals noDenormals;
    auto totalNumInputChannels  = getTotalNumInputChannels();
    auto totalNumOutputChannels = getTotalNumOutputChannels();

    //clear buffer
    for (auto i = totalNumInputChannels; i < totalNumOutputChannels; ++i)
        buffer.clear (i, 0, buffer.getNumSamples());

    /* DESCRIPTION

     This takes 4 channels and spits back out 4 new channels after encoding.

     Equation:

     W = FLU + FRD + BLD + BRU
     Y = FLU - FRD + BLD - BRU
     Z = FLU - FRD - BLD + BRU
     X = FLU + FRD - BLD - BRU

     ACN.

     */

    //make 4 write pointers (output)
    auto* outDataW = buffer.getWritePointer (0);
    auto* outDataY = buffer.getWritePointer (1);
    auto* outDataZ = buffer.getWritePointer (2);
    auto* outDataX = buffer.getWritePointer (3);
```

Identity and Type

Name: PluginProcessor.cpp
Type: Default - C++ Source
Location: Relative to Project
../../Source/PluginProcessor.cpp
Full Path: /Users/gabrielzalles/Documents/JUCE/gabrielJUCE/foaMicEnc/Source/PluginProcessor.cpp

On Demand Resource Tags

Only resources are taggable

Target Membership

- foaMicEnc - All
- foaMicEnc - VST3
- foaMicEnc - AU
- foaMicEnc - Standalone Plugin
- foaMicEnc - Shared Code

Text Settings

Text Encoding: No Explicit Encoding
Line Endings: No Explicit Line Endings
Indent Using: Spaces
Widths: 4 (Tab) 4 (Indent)
Wrap lines

our mic encoder

- cache image ✓
- resize canvas ✓
- draw text and image ✓
- set IO (4 in 4 out) ✓
  - buses “false” ✓
- use formula
  - define pi

this first

Note: no longer working with myProject, moved to FOAmicEnc

Source -> “multiple”

our mic encoder

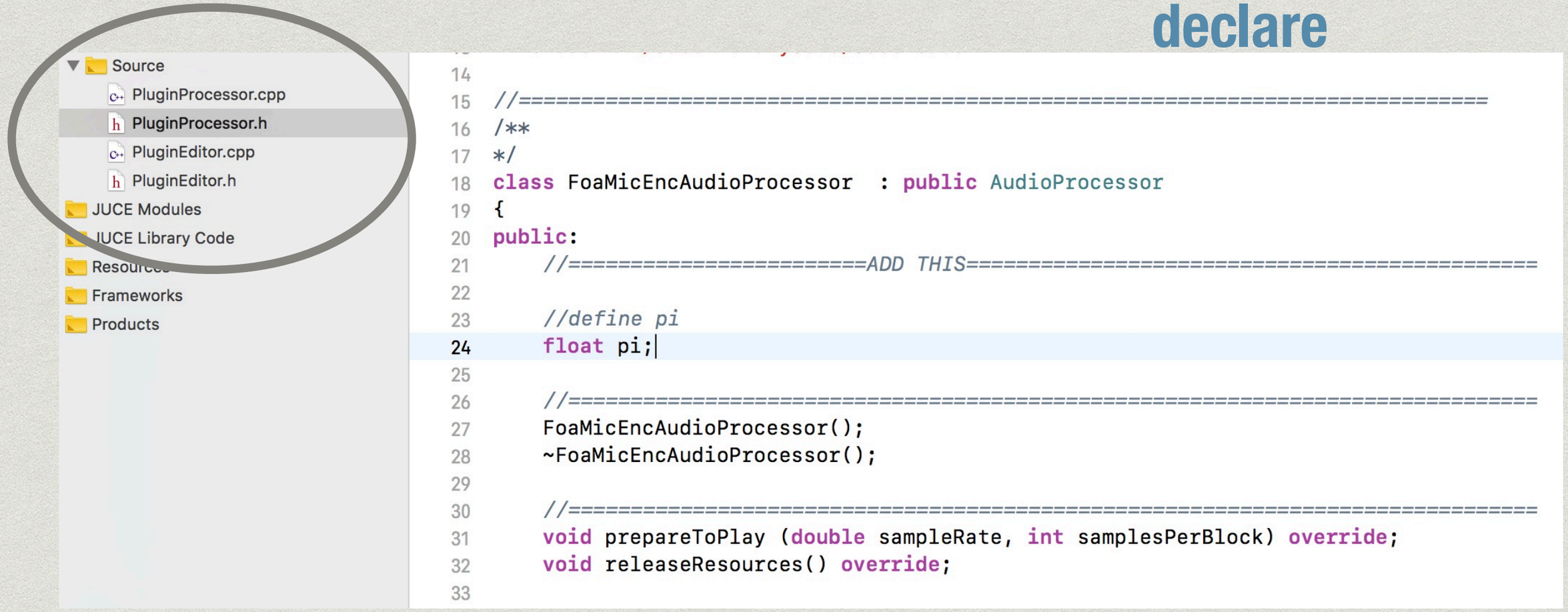
- define pi (used for SN3D normalization)
  - declare it in header -> (PluginProcessor.h)
    - scope = public
  - set equal to 4*atan(1) in prepareToPlay() method
    - prepareToPlay() -> (PluginProcessor.cpp)

project name must be foaMicEnc

Source ->PluginProcessor.h

our mic encoder

declare

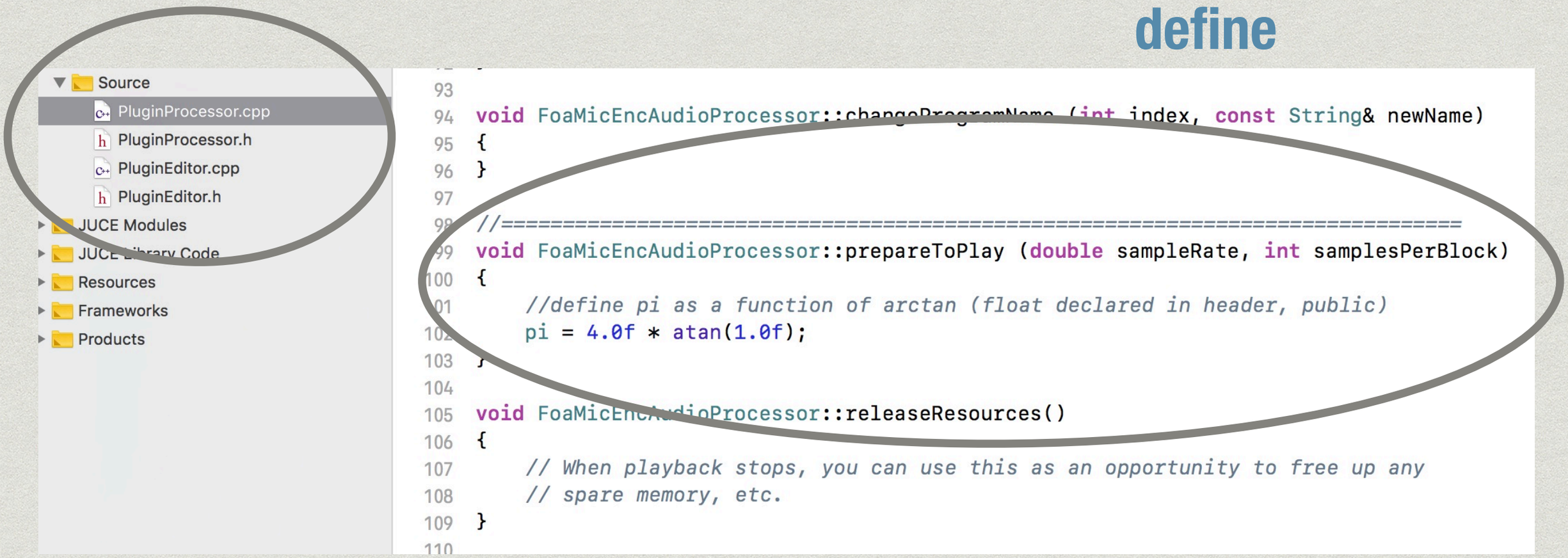


project name must be foaMicEnc

Source ->PluginProcessor.h

our mic encoder

define



project name must be foaMicEnc

our mic encoder

- cache image ✓
- resize canvas ✓
- draw text and image ✓
- set IO (4 in 4 out) ✓
  - buses “false” ✓
- use formula
  - define pi ✓

Note: no longer working with myProject, moved to FOAmicEnc

Source -> PluginProcessor.cpp

our mic encoder

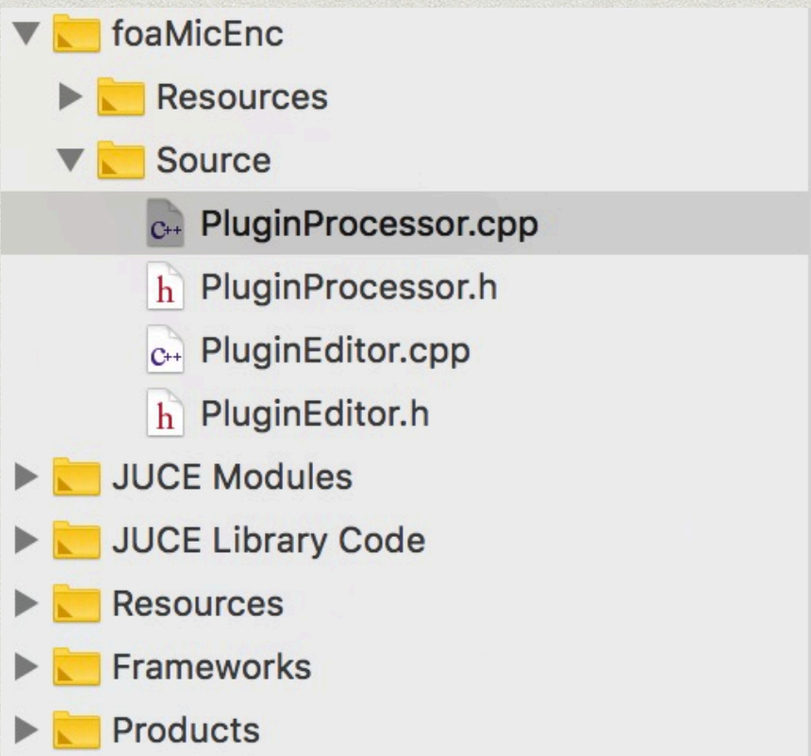
- edit **void** FoaMicEncAudioProcessor::processBlock (AudioBuffer<**float**>& buffer, MidiBuffer& midiMessages)
  - create read and write pointers
  - calculate normalization coefficient
  - traverse samples and apply formula

project name must be foaMicEnc

Source ->PluginProcessor.h

our mic encoder

create read and write pointers



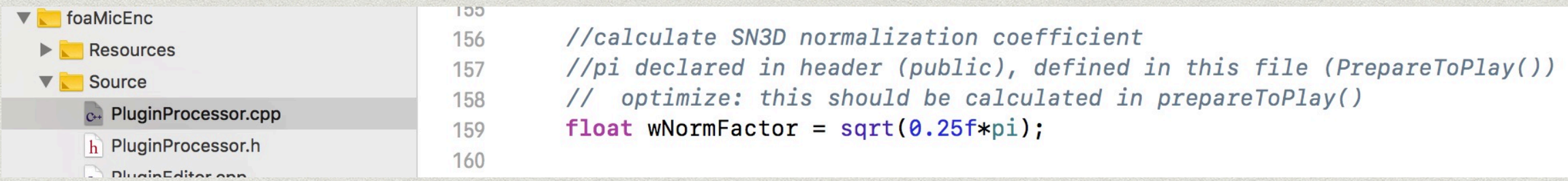

```
143
144     //make 4 write pointers (output)
145     auto* outDataW = buffer.getWritePointer (0);
146     auto* outDataY = buffer.getWritePointer (1);
147     auto* outDataZ = buffer.getWritePointer (2);
148     auto* outDataX = buffer.getWritePointer (3);
149
150     //make 4 read pointers (input)
151     auto* inDataFLU = buffer.getReadPointer (0);
152     auto* inDataFRD = buffer.getReadPointer (1);
153     auto* inDataBLD = buffer.getReadPointer (2);
154     auto* inDataBRU = buffer.getReadPointer (3);
155
```

project name must be foaMicEnc

Source ->PluginProcessor.h

our mic encoder

calculate normalization coefficient



SN3D

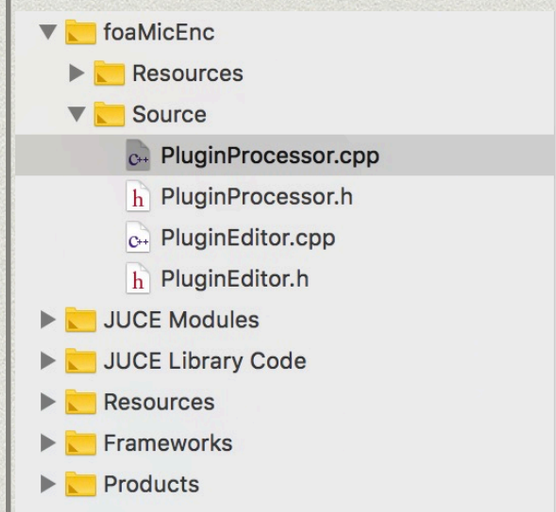
$$N_n^{|m|} = \sqrt{\frac{2-\delta_m}{4\pi}\frac{(n-|m|)!}{(n+|m|)!}}$$

Nachbar, Christian, et al. "Ambix-a suggested ambisonics format." Ambisonics Symposium, Lexington. 2011.

Source ->PluginProcessor.h

our mic encoder

traverse samples and use our formula




```
    //traverse samples
    for(int index = 0; index < buffer.getNumSamples(); index++) {

        //ACN = W Y Z X
        outDataW[index] = wNormFactor * (inDataFLU[index] + inDataFRD[index] +
                                         inDataBLD[index] + inDataBRU[index]);

        outDataY[index] = inDataFLU[index] - inDataFRD[index] + inDataBLD[index] - inDataBRU[index];
        outDataZ[index] = inDataFLU[index] - inDataFRD[index] - inDataBLD[index] + inDataBRU[index];
        outDataX[index] = inDataFLU[index] + inDataFRD[index] - inDataBLD[index] - inDataBRU[index];
    }
```

code on workshop site

FoaMicEncAudioProcessor

http://gabrielzalles.com/LAC19_workshop/

our mic encoder

- cache image ✓
- resize canvas ✓
- draw text and image ✓
- set IO (4 in 4 out) ✓
  - buses “false” ✓
- use formula ✓
  - define pi ✓

Note: no longer working with myProject, moved to FOAmicEnc

our mic encoder (testing)

- Find/record A-format audio
- Modify Facebook Template
  - A-format track(s)
  - Encoder track (send)
- WigWare VST for monitoring (optional but useful)
- Pass to binaural decoder (optional but useful)

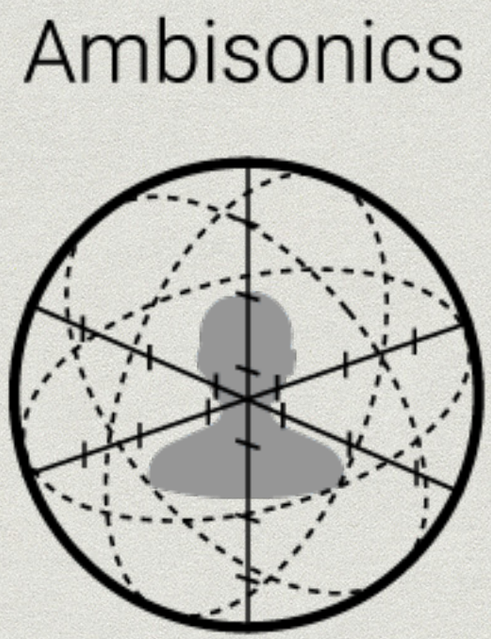
download Reaper Project

our mic encoder (issues)

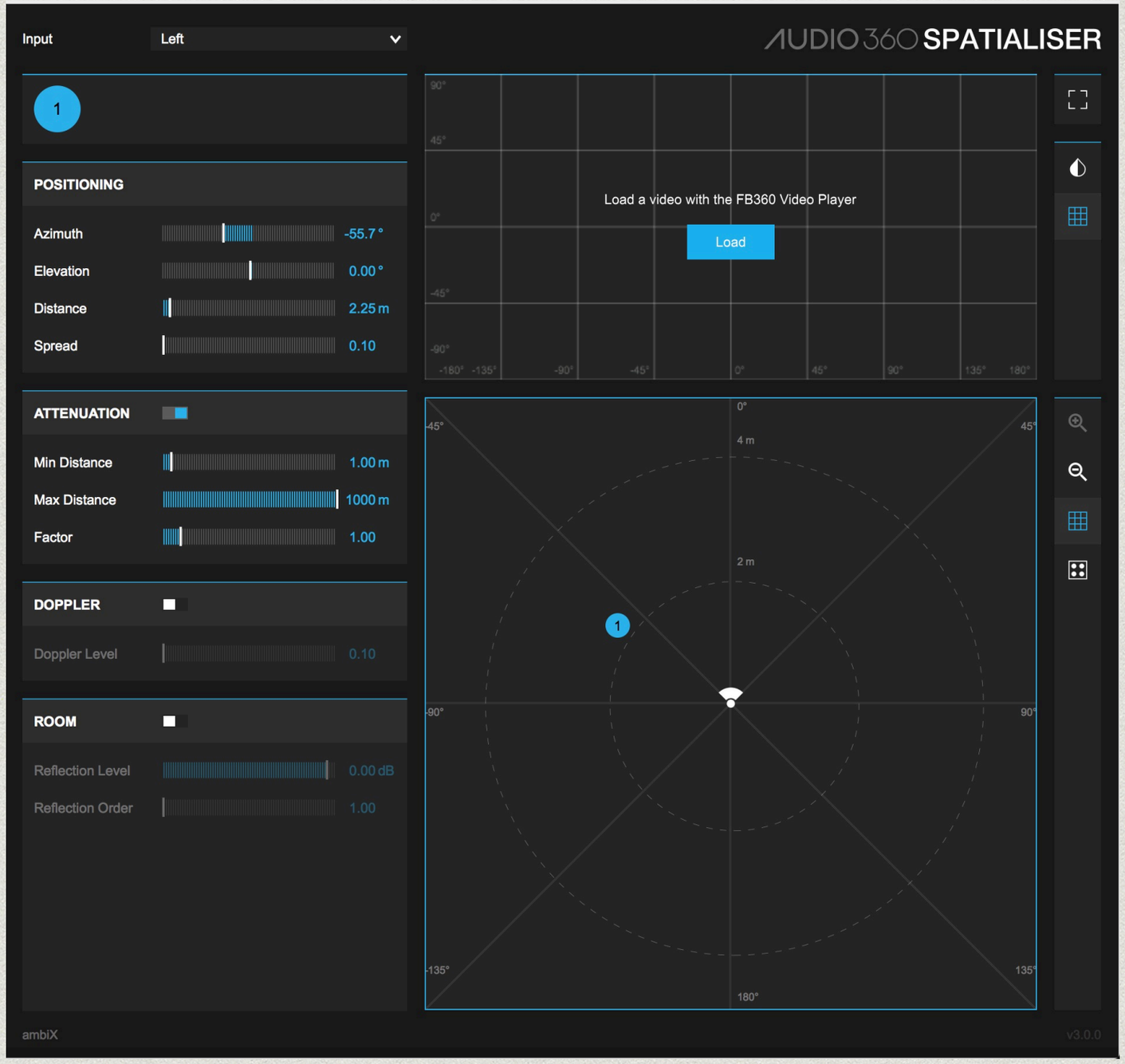
- Louder than original
  - Implement gain matching.
- Not a lot of features
  - Compared to Ambeo VST
- Not very attractive looking yet…
- Attendee can explore.

overview

- simple FOA mic encoder (A-to-B format)
- simple FOA panner (with interactive GUI)
- simple FOA decoder

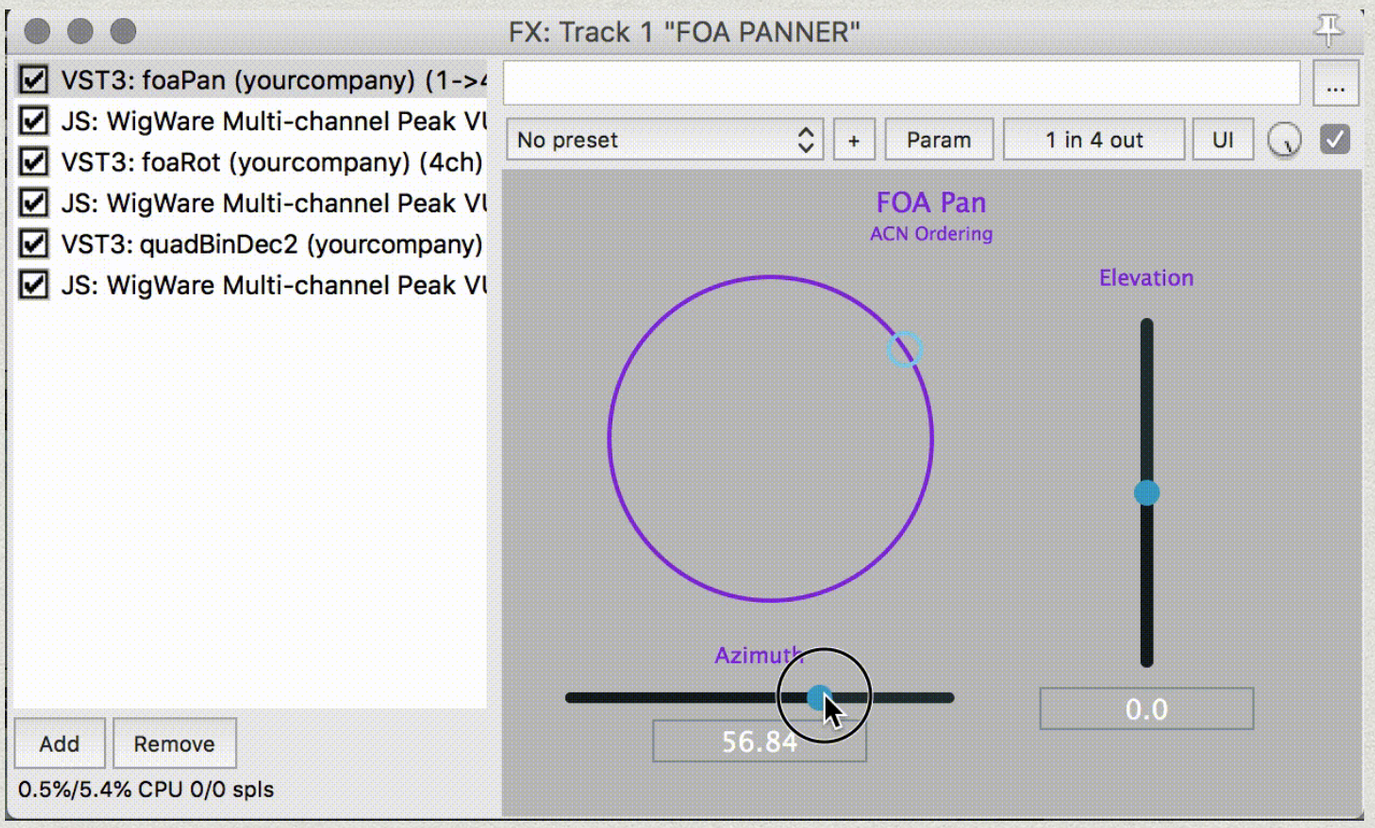


“ideal” FOA panner



- doppler
- change distance
- change spread
- room settings
- position interaction

our FOA panner



- minimal
- animated
- intuitive

our FOA panner (overview)

- create the appearance and animation
- add sliders to VTS
  - enabling communication between processor and editor
  - enabling automation in DAW
- use the formula

FOA Source Encoding Formula

$$W : 1/k * s(1/sqrt(2))$$

$$Y : 1/k * s(sin(a)cos(e))$$

$$Z : 1/k * s(sin(e))$$

$$X : 1/k * s(cos(a)cos(e))$$

our FOA panner

comma syntax

editor header

```
class FoaPanAudioProcessorEditor  : public AudioProcessorEditor,
private Slider::Listener //add listener to initialization list!
{
public:
    FoaPanAudioProcessorEditor (FoaPanAudioProcessor&);
    ~FoaPanAudioProcessorEditor();

    //
        ==========================================================
        =======
    void paint (Graphics&) override;
    void resized() override;

    //declare our slider function
    void sliderValueChanged (Slider* slider) override;

    Slider azimuthSlider;   // declare slider for azi
    Slider elevationSlider; // declare slider for elev

    //declare scoped pointers to add sliders to VTS
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        azimuthSliderAttachment;
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        elevationSliderAttachment;

    //declare some float variables
    float aziPosX, aziPosY; //use to draw smaller circle
    float aziVal, elevVal;  //used to store slider value
```

our FOA panner

editor header

```
class FoaPanAudioProcessorEditor  : public AudioProcessorEditor,
private Slider::Listener //add listener to initialization list!
{
public:
    FoaPanAudioProcessorEditor (FoaPanAudioProcessor&);
    ~FoaPanAudioProcessorEditor();

    //
        ==========================================================
        =======
    void paint (Graphics&) override;
    void resized() override;

    //declare our slider function
    void sliderValueChanged (Slider* slider) override;

    Slider azimuthSlider;   // declare slider for azi
    Slider elevationSlider; // declare slider for elev

    //declare scoped pointers to add sliders to VTS
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        azimuthSliderAttachment;
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        elevationSliderAttachment;

    //declare some float variables
    float aziPosX, aziPosY; //use to draw smaller circle
    float aziVal, elevVal;  //used to store slider value
```

we will define this function in the .cpp file

our FOA panner

editor header

```
class FoaPanAudioProcessorEditor  : public AudioProcessorEditor,
private Slider::Listener //add listener to initialization list!
{
public:
    FoaPanAudioProcessorEditor (FoaPanAudioProcessor&);
    ~FoaPanAudioProcessorEditor();

    //
        ==========================================================
        =======
    void paint (Graphics&) override;
    void resized() override;

    //declare our slider function
    void sliderValueChanged (Slider* slider) override;

    Slider azimuthSlider;   // declare slider for azi
    Slider elevationSlider; // declare slider for elev

    //declare scoped pointers to add sliders to VTS
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        azimuthSliderAttachment;
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        elevationSliderAttachment;

    //declare some float variables
    float aziPosX, aziPosY; //use to draw smaller circle
    float aziVal, elevVal;  //used to store slider value
```

self explanatory

our FOA panner

editor header

```
class FoaPanAudioProcessorEditor  : public AudioProcessorEditor,
private Slider::Listener //add listener to initialization list!
{
public:
    FoaPanAudioProcessorEditor (FoaPanAudioProcessor&);
    ~FoaPanAudioProcessorEditor();

    //
        ==========================================================
        =======
    void paint (Graphics&) override;
    void resized() override;

    //declare our slider function
    void sliderValueChanged (Slider* slider) override;

    Slider azimuthSlider;   // declare slider for azi
    Slider elevationSlider; // declare slider for elev

    //declare scoped pointers to add sliders to VTS
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        azimuthSliderAttachment;
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        elevationSliderAttachment;

    //declare some float variables
    float aziPosX, aziPosY; //use to draw smaller circle
    float aziVal, elevVal;  //used to store slider value
```

weird code for VTS

our FOA panner

```
class FoaPanAudioProcessorEditor  : public AudioProcessorEditor,
private Slider::Listener //add listener to initialization list!
{
public:
    FoaPanAudioProcessorEditor (FoaPanAudioProcessor&);
    ~FoaPanAudioProcessorEditor();

    //
        ==========================================================
        =======
    void paint (Graphics&) override;
    void resized() override;

    //declare our slider function
    void sliderValueChanged (Slider* slider) override;

    Slider azimuthSlider;   // declare slider for azi
    Slider elevationSlider; // declare slider for elev

    //declare scoped pointers to add sliders to VTS
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        azimuthSliderAttachment;
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        elevationSliderAttachment;

    //declare some float variables
    float aziPosX, aziPosY; //use to draw smaller circle
    float aziVal, elevVal;  //used to store slider value
```

editor header

smaller circle depicts source position

our FOA panner

```
class FoaPanAudioProcessorEditor  : public AudioProcessorEditor,
private Slider::Listener //add listener to initialization list!
{
public:
    FoaPanAudioProcessorEditor (FoaPanAudioProcessor&);
    ~FoaPanAudioProcessorEditor();

    //
        ==========================================================
        =======
    void paint (Graphics&) override;
    void resized() override;

    //declare our slider function
    void sliderValueChanged (Slider* slider) override;

    Slider azimuthSlider;   // declare slider for azi
    Slider elevationSlider; // declare slider for elev

    //declare scoped pointers to add sliders to VTS
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        azimuthSliderAttachment;
    ScopedPointer<AudioProcessorValueTreeState::SliderAttachment>
        elevationSliderAttachment;

    //declare some float variables
    float aziPosX, aziPosY; //use to draw smaller circle
    float aziVal, elevVal;  //used to store slider value
```

editor header

self explanatory

our FOA panner

download or copy paste snippet, edit as needed.

our FOA panner

define constructor in editor .cpp

```
FoaPanAudioProcessorEditor::FoaPanAudioProcessorEditor (FoaPanAudioProcessor&
    p)
: AudioProcessorEditor (&p), processor (p)
{
    // Make sure that before the constructor has finished, you've set the
    // editor's size to whatever you need it to be.
    setSize (400, 300);

    int textEntryBoxWidth = 100;
    int textEntryBoxHeight = 20;
    float rangeMin = -180.0;
    float rangeMax = +180.0;
    bool readOnly = false;

    // these define the parameters of our azimuth slider object
    azimuthSlider.setRange(rangeMin, rangeMax);
    azimuthSlider.setSliderStyle (Slider::LinearHorizontal);
    azimuthSlider.setTextBoxStyle (Slider::TextBoxBelow, readOnly,
        textEntryBoxWidth, textEntryBoxHeight);
    azimuthSlider.addListener(this);//this = Slider::Listener
    addAndMakeVisible (&azimuthSlider);

    // these define the parameters of our elevation slider object
    elevationSlider.setRange(rangeMin, rangeMax);
    elevationSlider.setSliderStyle (Slider::LinearVertical);
    elevationSlider.setTextBoxStyle (Slider::TextBoxBelow, readOnly,
        textEntryBoxWidth, textEntryBoxHeight);
    elevationSlider.addListener(this);//this = Slider::Listener
    addAndMakeVisible (&elevationSlider);

    //add the sliders to the VTS
    azimuthSliderAttachment = new
        AudioProcessorValueTreeState::SliderAttachment (processor.parameters,
        "azimuth", azimuthSlider);
    elevationSliderAttachment = new
        AudioProcessorValueTreeState::SliderAttachment (processor.parameters,
        "elevation", elevationSlider);
}
```

Slider::Listener from init list

our FOA panner

define constructor in editor .cpp

```
FoaPanAudioProcessorEditor::FoaPanAudioProcessorEditor (FoaPanAudioProcessor&
    p)
: AudioProcessorEditor (&p), processor (p)
{
    // Make sure that before the constructor has finished, you've set the
    // editor's size to whatever you need it to be.
    setSize (400, 300);

    int textEntryBoxWidth = 100;
    int textEntryBoxHeight = 20;
    float rangeMin = -180.0;
    float rangeMax = +180.0;
    bool readOnly = false;

    // these define the parameters of our azimuth slider object
    azimuthSlider.setRange(rangeMin, rangeMax);
    azimuthSlider.setSliderStyle (Slider::LinearHorizontal);
    azimuthSlider.setTextBoxStyle (Slider::TextBoxBelow, readOnly,
        textEntryBoxWidth, textEntryBoxHeight);
    azimuthSlider.addListener(this);//this = Slider::Listener
    addAndMakeVisible (&azimuthSlider);

    // these define the parameters of our elevation slider object
    elevationSlider.setRange(rangeMin, rangeMax);
    elevationSlider.setSliderStyle (Slider::LinearVertical);
    elevationSlider.setTextBoxStyle (Slider::TextBoxBelow, readOnly,
        textEntryBoxWidth, textEntryBoxHeight);
    elevationSlider.addListener(this);//this = Slider::Listener
    addAndMakeVisible (&elevationSlider);

    //add the sliders to the VTS
    azimuthSliderAttachment = new
        AudioProcessorValueTreeState::SliderAttachment (processor.parameters,
        "azimuth", azimuthSlider);
    elevationSliderAttachment = new
        AudioProcessorValueTreeState::SliderAttachment (processor.parameters,
        "elevation", elevationSlider);
}
```

same stuff but vertical slider

our FOA panner

define constructor in editor .cpp

```
    azimuthSlider.addListener(this);//this = Slider::Listener
    addAndMakeVisible (&azimuthSlider);

    // these define the parameters of our elevation slider object
    elevationSlider.setRange(rangeMin, rangeMax);
    elevationSlider.setSliderStyle (Slider::LinearVertical);
    elevationSlider.setTextBoxStyle (Slider::TextBoxBelow, readOnly,
        textEntryBoxWidth, textEntryBoxHeight);
    elevationSlider.addListener(this);//this = Slider::Listener
    addAndMakeVisible (&elevationSlider);

    //add the sliders to the VTS
    azimuthSliderAttachment = new
        AudioProcessorValueTreeState::SliderAttachment (processor.parameters,
        "azimuth", azimuthSlider);
    elevationSliderAttachment = new
        AudioProcessorValueTreeState::SliderAttachment (processor.parameters,
        "elevation", elevationSlider);
}
```

identifier!
*

match header declaration

code online!

our FOA panner

define paint method (animation)

```cpp
void FoaPanAudioProcessorEditor::paint (Graphics& g)
{
    float titleHeight = 15.0;//same as font size
    float subTitleHeight = 10.0;//same as font size

    // fill the whole window silver
    g.fillAll (Colours::silver);
    // set the current drawing colour to blueviolet
    g.setColour (Colours::blueviolet);
    // set the font size and draw text to the screen for title and subtitle
    g.setFont (15.0f);
    g.drawFittedText ("FOA Pan", 0, 0, getWidth(), titleHeight, Justification::centred, 1);//1=maxNumLines
    g.setFont (10.0f);
    g.drawFittedText ("ACN Ordering [WYZX]",
                      0, titleHeight, getWidth(), subTitleHeight, Justification::centred, 1);

    //set font size for slider text and draw text to the screen for both sliders
    g.setFont (12.0f);
    g.drawFittedText ("Azimuth", 20, 215, 200, 20, Justification::centred, 1);//trial and error
    g.drawFittedText ("Elevation", 200, 40, 200, 20, Justification::centred, 1);

    //define large ellipsis params
    int largeDiameter = 150;
    float largeRadius = largeDiameter/2;
    int lineThickness = 2;

    //not "grabbable" [to do]
    //draw the circle/"sphere" [to do]
    //ellipses depict position of source on azimuth plane only
    g.setColour (Colours::blueviolet);
    g.drawEllipse(50, 50, largeDiameter, largeDiameter, lineThickness);//main big circle

    //define small ellipsis params
    int smallDiameter = 15;
    float smallRadius = smallDiameter/2;

    //go to corner of large circle, then its origin, then to the position, multiply it by radius.
    //minus 90 degrees (pi/2) justifies the position so that 0 is above rather than on the right
    aziPosX = 50 + largeRadius + cos(aziVal * (M_PI / 180) - M_PI/2) * largeRadius;
    aziPosY = 50 + largeRadius + sin(aziVal * (M_PI / 180) - M_PI/2) * largeRadius;

    g.setColour (Colours::skyblue);
    //subtract the w and h of circle to have origin of mini circle at the circumference
    //posX, posY, w, h, thickness.
    g.drawEllipse(aziPosX - smallRadius, aziPosY - smallRadius, smallDiameter, smallDiameter, lineThickness);

    repaint();//important, redraws every frame
}
```

cpp

online

our FOA panner

define paint method (animation)

```
void FoaPanAudioProcessorEditor::paint (Graphics& g)
{
    float titleHeight = 15.0;//same as font size
    float subTitleHeight = 10.0;//same as font size

    // fill the whole window silver
    g.fillAll (Colours::silver);
    // set the current drawing colour to blueviolet
    g.setColour (Colours::blueviolet);
    // set the font size and draw text to the screen for title and subtitle
    g.setFont (15.0f);
    g.drawFittedText ("FOA Pan", 0, 0, getWidth(), titleHeight, Justification::centred, 1);//1=maxNumLines
    g.setFont (10.0f);
    g.drawFittedText ("ACN Ordering [WYZX]",
                      0, titleHeight, getWidth(), subTitleHeight, Justification::centred, 1);

    //set font size for slider text and draw text to the screen for both sliders
    g.setFont (12.0f);
    g.drawFittedText ("Azimuth", 20, 215, 200, 20, Justification::centred, 1);//trial and error
    g.drawFittedText ("Elevation", 200, 40, 200, 20, Justification::centred, 1);
```

basic stuff

our FOA panner

define paint method (animation)

```
    //define large ellipsis params
    int largeDiameter = 150;
    float largeRadius = largeDiameter/2;
    int lineThickness = 2;

    //not "grabbable" [to do]
    //draw the circle/"sphere" [to do]
    //ellipses depict position of source on azimuth plane only
    g.setColour (Colours::blueviolet);
    g.drawEllipse(50, 50, largeDiameter, largeDiameter, lineThickness);//main big circle

    //define small ellipsis params
    int smallDiameter = 15;
    float smallRadius = smallDiameter/2;

    //go to corner of large circle, then its origin, then to the position, multiply it by radius.
    //minus 90 degrees (pi/2) justifies the position so that 0 is above rather than on the right
    aziPosX = 50 + largeRadius + cos(aziVal * (M_PI / 180) - M_PI/2) * largeRadius;
    aziPosY = 50 + largeRadius + sin(aziVal * (M_PI / 180) - M_PI/2) * largeRadius;

    g.setColour (Colours::skyblue);
    //subtract the w and h of circle to have origin of mini circle at the circumference
    //posX, posY, w, h, thickness.
    g.drawEllipse(aziPosX - smallRadius, aziPosY - smallRadius, smallDiameter, smallDiameter, lineThickness);

    repaint();//important, redraws every frame
}
```

our FOA panner

define paint method (animation)

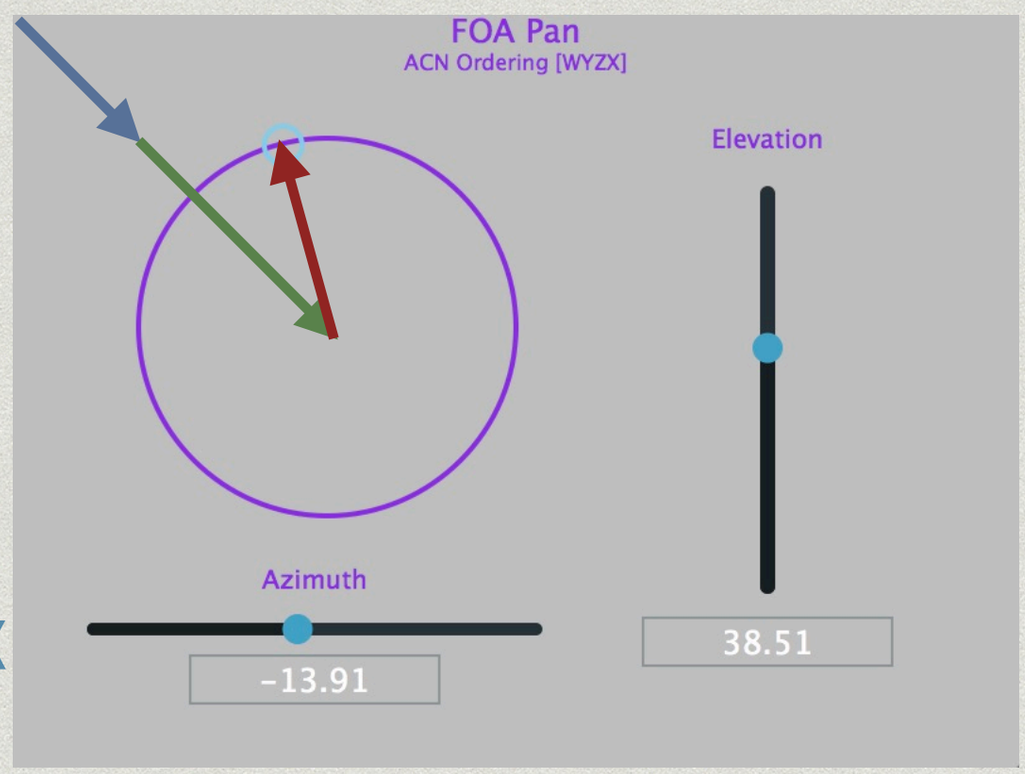
```
float smallRadius = smallDiameter/2;

//go to corner of large circle, then its origin, then to the position, multiply it by radius.
//minus 90 degrees (pi/2) justifies the position so that 0 is above rather than on the right
aziPosX = 50 + largeRadius + cos(aziVal * (M_PI / 180) - M_PI/2) * largeRadius;
aziPosY = 50 + largeRadius + sin(aziVal * (M_PI / 180) - M_PI/2) * largeRadius;

g.setColour (Colours::skyblue);
```

- circle position defined from top left corner
- multiplying by pi/180 (deg2rad)
- phase offset fixes trig identity
- times radius scales proportionally

our FOA panner

define paint method (animation)

- circle position defined from top left corner
- multiplying by pi/180 (deg2rad)
- phase offset fixes trig identity
- times radius scales proportionally



horizontal = cos = x
vertical = sin = y

our FOA panner

define paint method (animation)

```
    //go to corner of large circle, then its origin, then to the position, multiply it by radius.
    //minus 90 degrees (pi/2) justifies the position so that 0 is above rather than on the right
    aziPosX = 50 + largeRadius + cos(aziVal * (M_PI / 180) - M_PI/2) * largeRadius;
    aziPosY = 50 + largeRadius + sin(aziVal * (M_PI / 180) - M_PI/2) * largeRadius;

    g.setColour (Colours::skyblue);
    //subtract the w and h of circle to have origin of mini circle at the circumference
    //posX, posY, w, h, thickness.
    g.drawEllipse(aziPosX - smallRadius, aziPosY - smallRadius, smallDiameter, smallDiameter, lineThickness);

    repaint();//important, redraws every frame
}
```

offset smaller circle from origin to edge of invisible box

code online

our FOA panner (overview)

- create the appearance and animation ✓
- add sliders to VTS ✓
  - enabling communication between processor and editor
  - enabling automation in DAW
- use the formula

FOA Source Encoding Formula

$$W : 1/k * s(1/sqrt(2))$$

$$Y : 1/k * s(sin(a)cos(e))$$

$$Z : 1/k * s(sin(e))$$

$$X : 1/k * s(cos(a)cos(e))$$

s = signal

k = 1, one source

our FOA panner

audio stuff…

```
class FoaPanAudioProcessor  : public AudioProcessor
{
public:
    //==============================================================================
    float twopi;
    float azimuth;
    float elevation;
    AudioProcessorValueTreeState parameters;

    //==============================================================================
    FoaPanAudioProcessor();
    ~FoaPanAudioProcessor();
```

…

header

our FOA panner

constructor definition AudioProcessor

```
#include "PluginProcessor.h"
#include "PluginEditor.h"

//==============================================================================
FoaPanAudioProcessor::FoaPanAudioProcessor() : parameters (*this, nullptr)
#ifndef JucePlugin_PreferredChannelConfigurations
, AudioProcessor (BusesProperties()
#if ! JucePlugin_IsMidiEffect
#if ! JucePlugin_IsSynth
                  .withInput  ("Input",  AudioChannelSet::mono(), true)
#endif
                  .withOutput ("Output", AudioChannelSet::ambisonic(1), true)
#endif
                  )
#endif
{
    parameters.createAndAddParameter(std::make_unique<AudioParameterFloat>("azimuth",// parameter ID
                                                                           "azimuth",// parameter name
                                                                           NormalisableRange<float>
                                                                           (-180.0f, 180.0f),// range
                                                                           0.0f,// default value
                                                                           "degrees"));

    parameters.createAndAddParameter(std::make_unique<AudioParameterFloat>("elevation",// parameter ID
                                                                           "elevation",// parameter name
                                                                           NormalisableRange<float> (-180.0f, 180.0f),// range
                                                                           0.0f,// default value
                                                                           "degrees"));

    parameters.state = ValueTree (Identifier ("FoaPanVT"));
}
```

Field 'parameters' will be i

one in four out

ID *

online

our FOA panner

define vars

```
void FoaPanAudioProcessor::prepareToPlay (double sampleRate, int samplesPerBlock)
{
    // Use this method as the place to do any pre-playback
    // initialisation that you need..
    azimuth = 0;
    elevation = 0;
    twopi = 8.0f * atan(1.0f);
}
```

disable buses

```
#ifndef JucePlugin_PreferredChannelConfigurations
bool FoaPanAudioProcessor::isBusesLayoutSupported (const BusesLayout& layouts) const
{

    return false;//force plug in to take one input and spit out 4
}
#endif
```

our FOA panner

get slider values

```
//get the azi val from the slider
azimuth = *parameters.getRawParameterValue("azimuth");
azimuth = azimuth * (M_PI / 180.0f);//convert degrees to radians

//get the elev val from the slider
elevation = *parameters.getRawParameterValue("elevation");
elevation = elevation * (M_PI / 180.0f);//convert degrees to radians

//the code we wrote wants azimuth values increasing counterclockwise from 0
    (at the center) to 359 (1 degree to the right). In order to compensate
    for that (and to keep our sliders) we multiply negative values by -1
    and subtract 360 (and take the absolute value) for positive values.
if (azimuth < 0) {
    azimuth = azimuth * -1.0f;
}

if (azimuth > 0) {
    azimuth = abs(azimuth - 360.0f);
}
//if 0 leave as is. got all cases covered.
```

scale

hollerweger, 2008 “intro”

our FOA panner

```
    //traverse the output channels
    for (int channel = 0; channel < totalNumOutputChannels; ++channel)
    {
        //get write pointers (output)
        auto* outData = buffer.getWritePointer(channel);
        //always reading from mono source (input)
        auto* inData = buffer.getReadPointer (0);

        //traverse samples
        for(int index = 0; index < buffer.getNumSamples(); index++)
        {
            switch (channel) { //switch based on channel count
                case 0:
                    outData[index] = inData[index]*(1/(sqrt(2)));
                        //W : ACN channel 0
                    break;
                case 1:
                    outData[index] = inData[index]*(sin(azimuth)*cos(elevation));
                        //Y : ACN channel 1
                    break;
                case 2:
                    outData[index] = inData[index]*(sin(elevation));
                        //Z : ACN channel 2
                    break;
                case 3:
                    outData[index] = inData[index]*(cos(azimuth)*cos(elevation));
                        //X : ACN channel 3
                    break;
                default:
                    break;
            }
        }
    }
}
```

same as before, basically…

quiz

can you write the decoder without any help (easy)

how about the rotator? (a bit trickier)

there is an image on repo for the decoder.

first order “in-phase”

quiz

can you write the decoder without any help (easy)

logic:

read the four channels of B-format audio

apply the decoding matrix

blueRippleSound

Table 1: *Quad Decoding Coefficients (FOA)*

ACN	W	Y	X
FL	0.25	0.177	0.177
BL	0.25	0.177	-0.177
BR	0.25	-0.177	-0.177
FR	0.25	-0.177	0.177

ie. FL = W*0.25 + Y*0.177+x*0.177

first order “in-phase”

quiz

can you write the decoder without any help (easy)

Table 1: *Quad Decoding Coefficients (FOA)*

ACN	W	Y	X
FL	0.25	0.177	0.177
BL	0.25	0.177	-0.177
BR	0.25	-0.177	-0.177
FR	0.25	-0.177	0.177

ie. FL = W*0.25 + Y*0.177+x*0.177

why 0.177?

traditional 1/sqrt(2) divided by 4 channels

quiz

how about the rotator? (a bit trickier)