C++ Korea 4th Seminar

C++ 프로젝트 ~처음 만난 세계~

JUCE로 시작하는 GUI 프로그래밍

나의 첫 GUI 리듬게임 hyu 개발 도전기

선린인터넷고등학교

김진영

목차

1. 발표자 소개
2. JUCE 프레임워크 개요
3. JUCE 프레임워크 설치와 API 사용법
4. hyu에 사용한 API 살펴보기
5. Demo
6. Q & A

발표자 소개

- 선린인터넷고등학교 2학년
- Unifox 프로그래밍 동아리 부부장

관심 분야

- C++
- Deep Learning
- Pwnable
- OS

+Block Chain

Email: sdk159147@gmail.com

Blog: cafemocamoca.tistory.com

Github: https://github.com/CafeM0ca

JUCE 프레임워크 개요

- Julian Storer가 DAW(Digital Audio Workstation) 그래픽 및 오디오 기능을 개발하기 위해 C++로 구현
- 2004년부터 시작하여 현재는 JUCE5.3 버전
- 정말 많은 클래스를 제공함.

제공하는 클래스 목록 잠깐 구경하기

Audio

AiffAudioFormat
AudioCDBurner
AudioChannelSet
AudioData::ConverterInstance
AudioDeviceManager
AudioDeviceSelectorComponent

AudioAppComponent
AudioCDBurner::BurnProgressListener
AudioData
AudioData::Pointer
AudioDeviceManager::AudioDeviceSetup
AudioFormat

AudioBuffer
AudioCDReader
AudioData::Converter
AudioDataConverters
AudioDeviceManager::LevelMeter
AudioFormatManager

AudioPluginInstance::Parameter
AudioProcessor::BusProperties
Audi
Audi
Audi
Audi
Audi
Audi
Audi
Audi
Audi
Audi
Buffe

AudioProcessor
AudioProcessor::BusesLayout

AudioProcessor::Bus
AudioProcessor::BusesProperties
ightInfo
nent
ent

CoreAudioFormat
FloatVectorOperations
IIRFilter
KnownPluginList::CustomScanner
LAMEEncoderAudioFormat
MP3AudioFormat
MPEInstrument
MPENot
MPESyn
MPEZon
Memory
MidiFile
MidiKey
MidiMes
MidiMes
MidiRPN
OggVorb

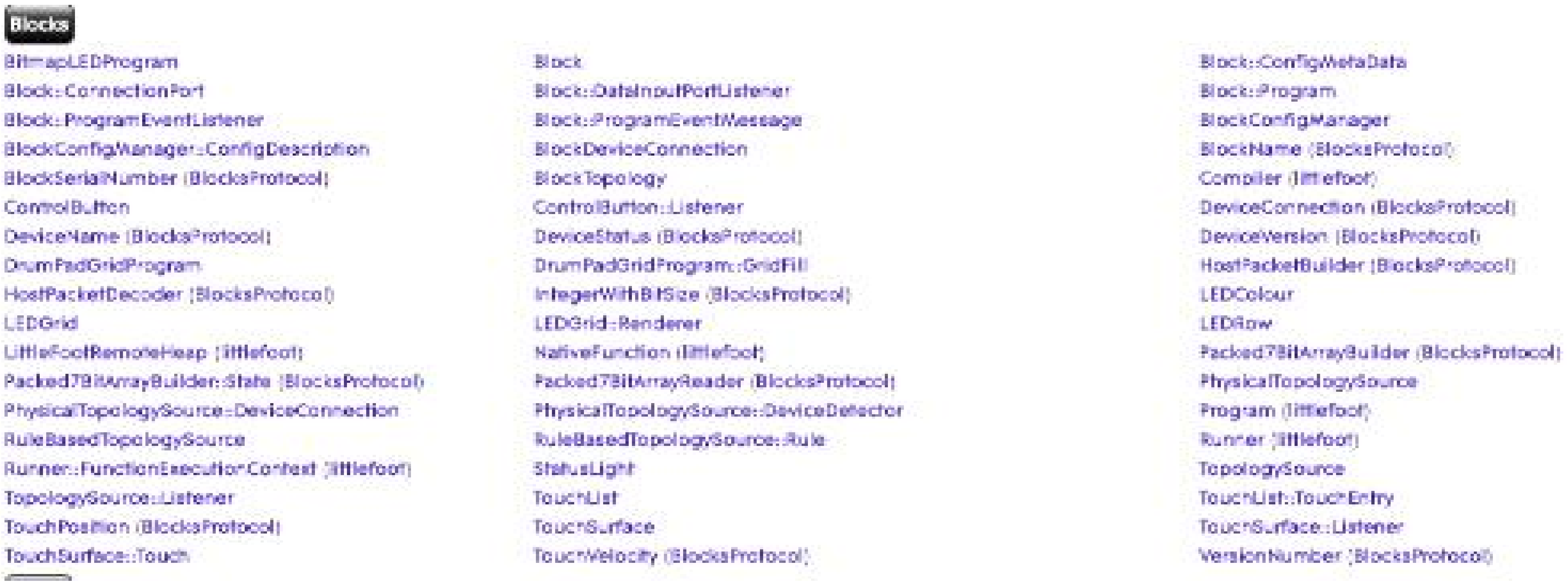
Decibels
GenericAudioProcessorEditor
IIRFilterAudioSource
KnownPluginList::PluginTree
LagrangeInterpolator
MPEChannelAssigner
MPEInstrument::Listener

FlacAudioFormat
IIRCoefficients
KnownPluginList
LADSPAPluginFormat
LinearSmoothedValue
MPEChannelRemapper
MPEMessages

PluginHostType
ResamplingAudioSource
ReverbAudioSource
ScopedNoDenormals
SpeakerMappings::Mapping
StandalonePluginHolder
SynthesiserSound
ToneGeneratorAudioSource
VSTPluginFormat
VstEffectInterface
VstIndividualSpeakerInfo
VstSpeakerConfiguration
WavAudioFormat

PluginListComponent
Reverb
SamplerSound
SoundPlayer
SpeakerMappings::VstSpeakerConfigurationHolder
StandalonePluginHolder::PluginInOuts
SynthesiserVoice
VST3PluginFormat
VSTPluginFormat::ExtraFunctions
VstEvent
VstMidiEvent
VstSysExEvent
WindowsMediaAudioFormat

PositionableAudioSource
Reverb::Parameters
SamplerVoice
SpeakerMappings
StandaloneFilterWindow
Synthesiser
SystemAudioVolume
VSTCallbackHandler
VstEditorBounds
VstEventBlock
VstPinInfo
VstTimingInformation
vst2FxBank



Blocks

BitmapLEDProgram
Block
Block::ConfigMetaData
Block::ConnectionPort
Block::DataInputPortListener
Block::Program
Block::ProgramEventListener
Block::ProgramEventMessage
BlockConfigManager
BlockConfigManager::ConfigDescription
BlockDeviceConnection
BlockName (BlocksProtocol)
BlockSerialNumber (BlocksProtocol)
BlockTopology
Compiler (littlefoot)
ControlButton
ControlButton::Listener
DeviceConnection (BlocksProtocol)
DeviceName (BlocksProtocol)
DeviceStatus (BlocksProtocol)
DeviceVersion (BlocksProtocol)
DrumPadGridProgram
DrumPadGridProgram::GridFill
HostPacketBuilder (BlocksProtocol)
HostPacketDecoder (BlocksProtocol)
IntegerWithBitSize (BlocksProtocol)
LEDColour
LEDGrid
LEDGrid::Renderer
LEDRow
LittleFootRemoteHeap (littlefoot)
NativeFunction (littlefoot)
Packed7BitArrayBuilder (BlocksProtocol)
Packed7BitArrayBuilder::State (BlocksProtocol)
Packed7BitArrayReader (BlocksProtocol)
PhysicalTopologySource
PhysicalTopologySource::DeviceConnection
PhysicalTopologySource::DeviceDetector
Program (littlefoot)
RuleBasedTopologySource
RuleBasedTopologySource::Rule
Runner (littlefoot)
Runner::FunctionExecutionContext (littlefoot)
StatusLight
TopologySource
TopologySource::Listener
TouchList
TouchList::TouchEntry
TouchPosition (BlocksProtocol)
TouchSurface
TouchSurface::Listener
TouchSurface::Touch
TouchVelocity (BlocksProtocol)
VersionNumber (BlocksProtocol)

TouchSurface::Touch

Box2D

Box2DRenderer

Analytics

Analytics
AnalyticsDestination
AnalyticsDestination::AnalyticsEvent
ButtonTracker
ThreadedAnalyticsDestination

Core

AbstractFifo
AbstractFifo::ScopedReadWrite
Array
ArrayAllocationBase
Atomic
Base64
BigInteger
BufferedInputStream
ByteOrder
CharPointer_ASCII
CharPointer_UTF16
CharPointer_UTF32
CharPointer_UTF8
CharacterFunctions
CharacterFunctions::HexParser
ChildProcess
ContainerDeletePolicy
CriticalSection
DatagramSocket
DefaultElementComparator
DefaultHashFunctions
DirectoryIterator
DummyCriticalSection
DummyCriticalSection::ScopedLockType
DynamicLibrary
DynamicObject
Expression
Expression::Scope
Expression::Scope::Visitor
Expression::Symbol
File
FileInputSource
FileOutputStream
GZIPDecompressorInputStream
GenericScopedUnlock
HeapBlock
Identifier
InterProcessLock
JavascriptEngine
LinkedListPointer::Appender
ListenerList::Iterator
MACAddress
MemoryInputStream
NamedPipe
NewLine
OutputStream
PerformanceCounter

PerformanceCounter
PerformanceCounter::Statistics
Process
PropertySet
Random
Range
ReadWriteLock
ReferenceCountedArray
ReferenceCountedObject
ReferenceCountedObjectPtr
RelativeTime
Result
RuntimePermissions
ScopedAutoReleasePool
ScopedPointer
ScopedReadLock
ScopedTimeMeasurement
ScopedValueSetter
ScopedWriteLock
SharedResourcePointer
SingleThreadedReferenceCountedObject
SingletonHolder
SmallestFloatType (TypeHelpers)
SortedSet
SparseSet
SpinLock
StatisticsAccumulator
StreamingSocket
String
StringArray
StringPairArray
StringPool
StringRef
SubregionStream
SystemStats
TemporaryFile
TextDiff
TextDiff::Change
Thread
Thread::Listener
ThreadLocalValue
ThreadPool
ThreadPool::JobSelector
ThreadPoolJob
Time
TimeSliceClient
TimeSliceThread
URL
URL::DownloadTask
URL::DownloadTask::Listener
URLInputSource
UnitTest
UnitTestRunner
UnitTestRunner::TestResult
UnsignedTypeWithSize (TypeHelpers)
Uuid
VariantConverter
WaitableEvent
WeakReference
WeakReference::Master
WeakReference::SharedPointer
WebInputStream
WebInputStream::Listener
WildcardFileFilter
WindowsRegistry
XmlDocument
XmlElement
ZipFile
ZipFile::Builder
ZipFile::ZipEntry
function (std)
var
var::NativeFunctionArgs

Cryptography

BlowFish
MD5
Primes
RSAKey
SHA256
Whirlpool

DSP

AudioBlock (dsp)
Bias (dsp)
Coefficients (dsp::FIR)
Coefficients (dsp::IIR)
Convolution (dsp)
FFT (dsp)
FastMathApproximations (dsp)
Filter (dsp::FIR)
Filter (dsp::IIR)
Filter (dsp::StateVariableFilter)
FilterDesign (dsp)
FilterDesign::IIRPolyphaseAllpassStructure (dsp)
Gain (dsp)
LadderFilter (dsp)
LookupTable (dsp)
LookupTableTransform (dsp)
Matrix (dsp)
Oscillator (dsp)
Oversampling (dsp)
Parameters (dsp::StateVariableFilter)
Phase (dsp)
Polynomial (dsp)
ProcessContextNonReplacing (dsp)
ProcessContextReplacing (dsp)
ProcessSpec (dsp)
ProcessorBase (dsp)
ProcessorDuplicator (dsp)
ProcessorState (dsp)
ProcessorWrapper (dsp)
Reverb (dsp)
SIMDRegister (dsp)
SIMDRegister::ElementAccess (dsp)
SpecialFunctions (dsp)
WaveShaper (dsp)
WindowingFunction (dsp)

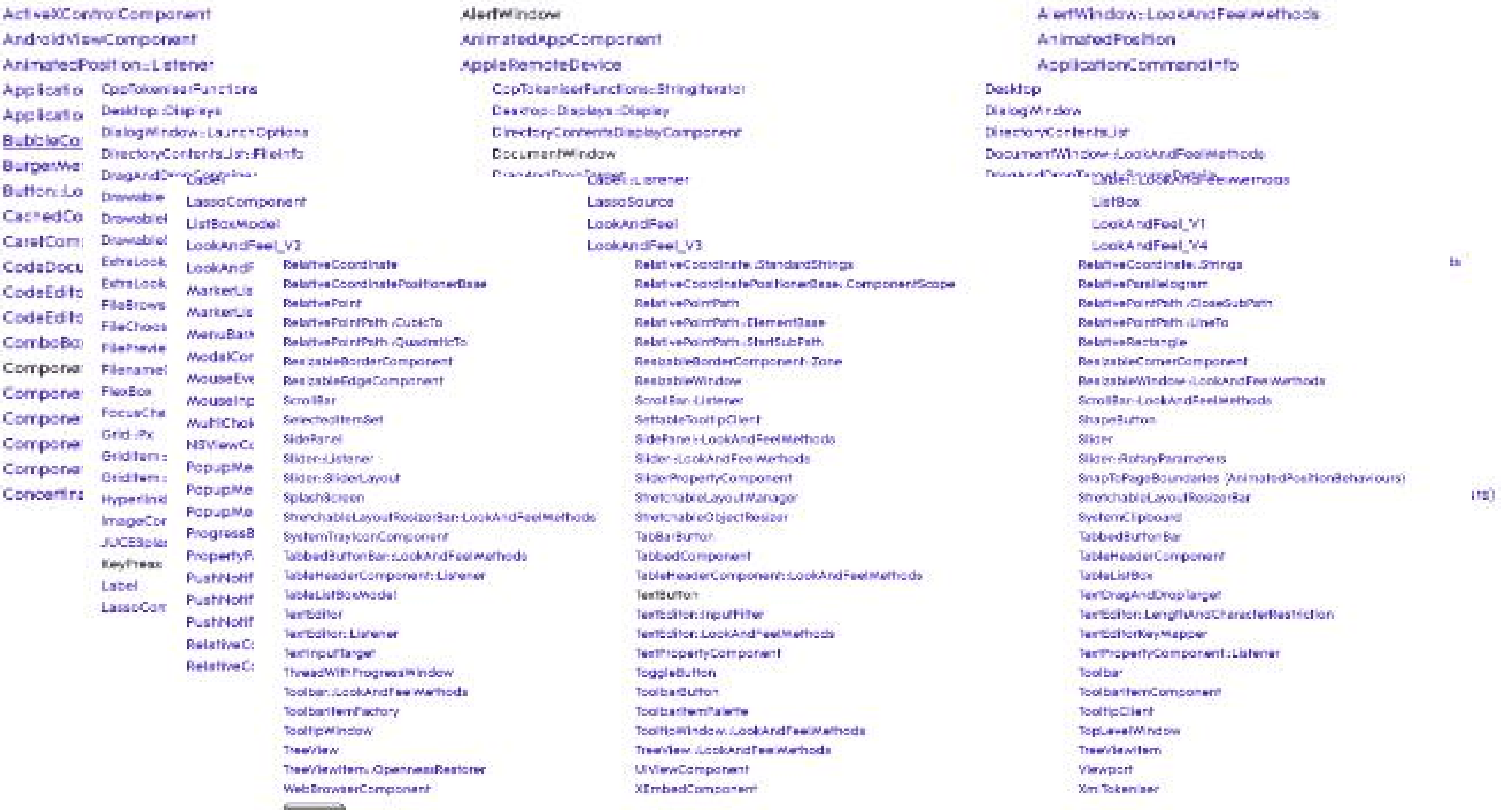
DataStructures

- ApplicationProperties
- CachedValue
- PropertiesFile
- PropertiesFile::Options
- UndoManager
- UndoableAction
- Value
- Value::Listener
- Value::ValueSource
- ValueTree
- ValueTree::Iterator
- ValueTree::Listener
- ValueTreeSynchroniser
- ValueWithDefault

Events

- ActionBroadcaster
- ActionListener
- AsyncUpdater
- CallbackMessage
- ChangeBroadcaster
- ChangeListener
- ChildProcessMaster
- ChildProcessSlave
- DeletedAtShutdown
- InterprocessConnection
- InterprocessConnectionServer
- JUCEApplicationBase
- Message
- MessageListener
- MessageManager
- MessageManager::Lock
- MessageManager::MessageBase
- MessageManagerLock
- MountedVolumeListChangeDetector
- MultiTimer
- ScopedJuceInitialiser_GUI
- Timer

GUI



ActiveXControlComponent
AndroidViewComponent
AnimatedPosition::Listener
Applicatio
Applicatio
BubbleCo
BurgerMe
Button::Lo
CachedCo
CaretCom
CodeDocu
CodeEdito
CodeEdito
ComboBo
Compone
Compone
Compone
Compone
Compone
Concertina

AlertWindow
AnimatedAppComponent
AppleRemoteDevice

AlertWindow::LookAndFeelMethods
AnimatedPosition
ApplicationCommandInfo

CppTokeniserFunctions
Desktop::Displays
DialogWindow::LaunchOptions
DirectoryContentsList::FileInfo
DragAndDropContainer
Drawable
DrawableI
DrawableI
ExtraLook
ExtraLook
FileBrows
FileChoos
FilePrevie
FilenameI
FlexBox
FocusCha
Grid::Px
GridItem::
GridItem::
Hyperlink
ImageCor
JUCESplas
KeyPress
Label
LassoCom

CppTokeniserFunctions::StringIterator
Desktop::Displays::Display
DirectoryContentsDisplayComponent
DocumentWindow
DragAndDropTarget

Desktop
DialogWindow
DirectoryContentsList
DocumentWindow::LookAndFeelMethods
DragAndDropTarget::SourceDetails

Label
LassoComponent
ListBoxModel
LookAndFeel_V2
LookAndF
MarkerLis
MarkerLis
MenuBar
ModalCor
MouseEve
MouseInp
MultiChoi
NSViewCo
PopupMe
PopupMe
PopupMe
ProgressB
PropertyP
PushNotif
PushNotif
PushNotif
RelativeC
RelativeC

Label::Listener
LassoSource
LookAndFeel
LookAndFeel_V3

Label::LookAndFeelMethods
ListBox
LookAndFeel_V1
LookAndFeel_V4

ts

(18)

RelativeCoordinate
RelativeCoordinatePositionerBase
RelativePoint
RelativePointPath::CubicTo
RelativePointPath::QuadraticTo
ResizableBorderComponent
ResizableEdgeComponent
ScrollBar
SelectedItemSet
SidePanel
Slider::Listener
Slider::SliderLayout
SplashScreen
StretchableLayoutResizerBar::LookAndFeelMethods
SystemTrayIconComponent
TabbedButtonBar::LookAndFeelMethods
TableHeaderComponent::Listener
TableListBoxModel
TextEditor
TextEditor::Listener
TextInputTarget
ThreadWithProgressWindow
Toolbar::LookAndFeelMethods
ToolbarItemFactory
TooltipWindow
TreeView
TreeViewItem::OpennessRestorer
WebBrowserComponent

RelativeCoordinate::StandardStrings
RelativeCoordinatePositionerBase::ComponentScope
RelativePointPath
RelativePointPath::ElementBase
RelativePointPath::StartSubPath
ResizableBorderComponent::Zone
ResizableWindow
ScrollBar::Listener
SettableTooltipClient
SidePanel::LookAndFeelMethods
Slider::LookAndFeelMethods
SliderPropertyComponent
StretchableLayoutManager
StretchableObjectResizer
TabBarButton
TabbedComponent
TableHeaderComponent::LookAndFeelMethods
TextButton
TextEditor::InputFilter
TextEditor::LookAndFeelMethods
TextPropertyComponent
ToggleButton
ToolbarButton
ToolbarItemPalette
TooltipWindow::LookAndFeelMethods
TreeView::LookAndFeelMethods
UIViewComponent
XEmbedComponent

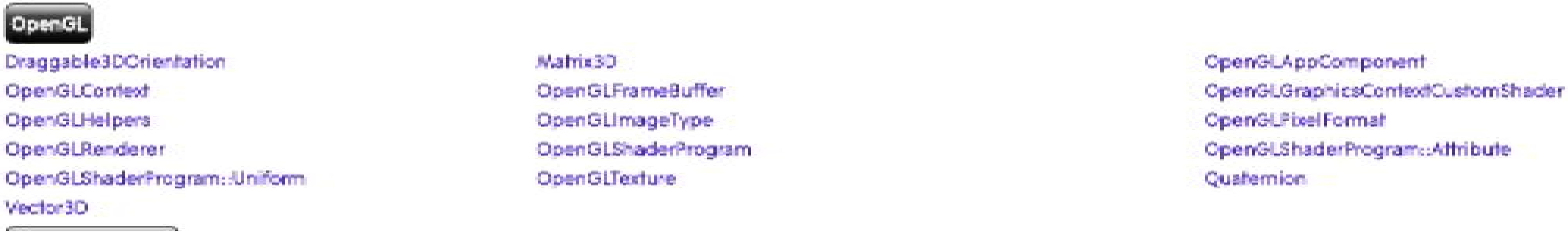
RelativeCoordinate::Strings
RelativeParallelogram
RelativePointPath::CloseSubPath
RelativePointPath::LineTo
RelativeRectangle
ResizableCornerComponent
ResizableWindow::LookAndFeelMethods
ScrollBar::LookAndFeelMethods
ShapeButton
Slider
Slider::RotaryParameters
SnapToPageBoundaries (AnimatedPositionBehaviours)
StretchableLayoutResizerBar
SystemClipboard
TabbedButtonBar
TableHeaderComponent
TableListBox
TextDragAndDropTarget
TextEditor::LengthAndCharacterRestriction
TextEditorKeyMapper
TextPropertyComponent::Listener
Toolbar
ToolbarItemComponent
TooltipClient
TopLevelWindow
TreeViewItem
Viewport
XmlTokeniser

Graphics

- AffineTransform
- AttributedString
- AttributedString::Attribute
- BorderSize
- Colour
- ColourGradient
- Colours
- CustomTypeface
- DropShadow
- DropShadowEffect
- EdgeTable
- FillType
- Font
- GIFImageFormat
- GlowEffect
- GlyphArrangement
- Graphics
- Graphics::ScopedSaveState
- Image
- Image::BitmapData
- Image::BitmapData::BitmapDataReleaser
- ImageCache
- ImageConvolutionKernel
- ImageEffectFilter
- ImageFileFormat
- ImagePixelData
- ImagePixelData::Listener
- ImageType
- JPEGImageFormat
- Justification
- Line
- LowLevelGraphicsContext
- LowLevelGraphicsPostScriptRenderer
- LowLevelGraphicsPostScriptRenderer::SavedState
- LowLevelGraphicsSoftwareRenderer
- NativeImageType
- PNGImageFormat
- Parallelogram
- Path
- Path::Iterator
- PathFlatteningIterator
- PathStrokeType
- PixelARGB
- PixelAlpha
- PixelRGB
- Point
- PositionedGlyph
- Rectangle
- RectangleList
- RectanglePlacement
- SoftwareImageType
- TextLayout
- TextLayout::Glyph
- TextLayout::Line
- TextLayout::Run
- Typeface

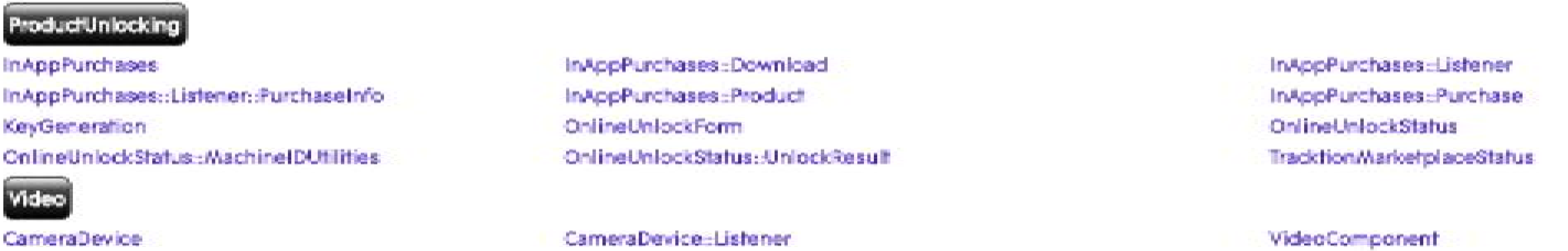
OSC

- OSCAddress
- OSCAddressPattern
- OSCArgument
- OSCBundle
- OSCBundle::Element
- OSCException
- OSCFormatError
- OSCInternalError
- OSCMessage
- OSCReceiver
- OSCReceiver::Listener
- OSCReceiver::ListenerWithOSCAddress
- OSCReceiver::MessageLoopCallback
- OSCReceiver::RealtimeCallback
- OSCSender
- OSCTimeTag
- OSCTypes



OpenGL

Draggable3DOrientation
Matrix3D
OpenGLAppComponent
OpenGLContext
OpenGLFrameBuffer
OpenGLGraphicsContextCustomShader
OpenGLHelpers
OpenGLImageType
OpenGLPixelFormat
OpenGLRenderer
OpenGLShaderProgram
OpenGLShaderProgram::Attribute
OpenGLShaderProgram::Uniform
OpenGLTexture
Quaternion
Vector3D



ProductUnlocking

InAppPurchases
InAppPurchases::Download
InAppPurchases::Listener
InAppPurchases::Listener::PurchaseInfo
InAppPurchases::Product
InAppPurchases::Purchase
KeyGeneration
OnlineUnlockForm
OnlineUnlockStatus
OnlineUnlockStatus::MachineIDUtilities
OnlineUnlockStatus::UnlockResult
TracktionMarketplaceStatus

Video

CameraDevice
CameraDevice::Listener
VideoComponent

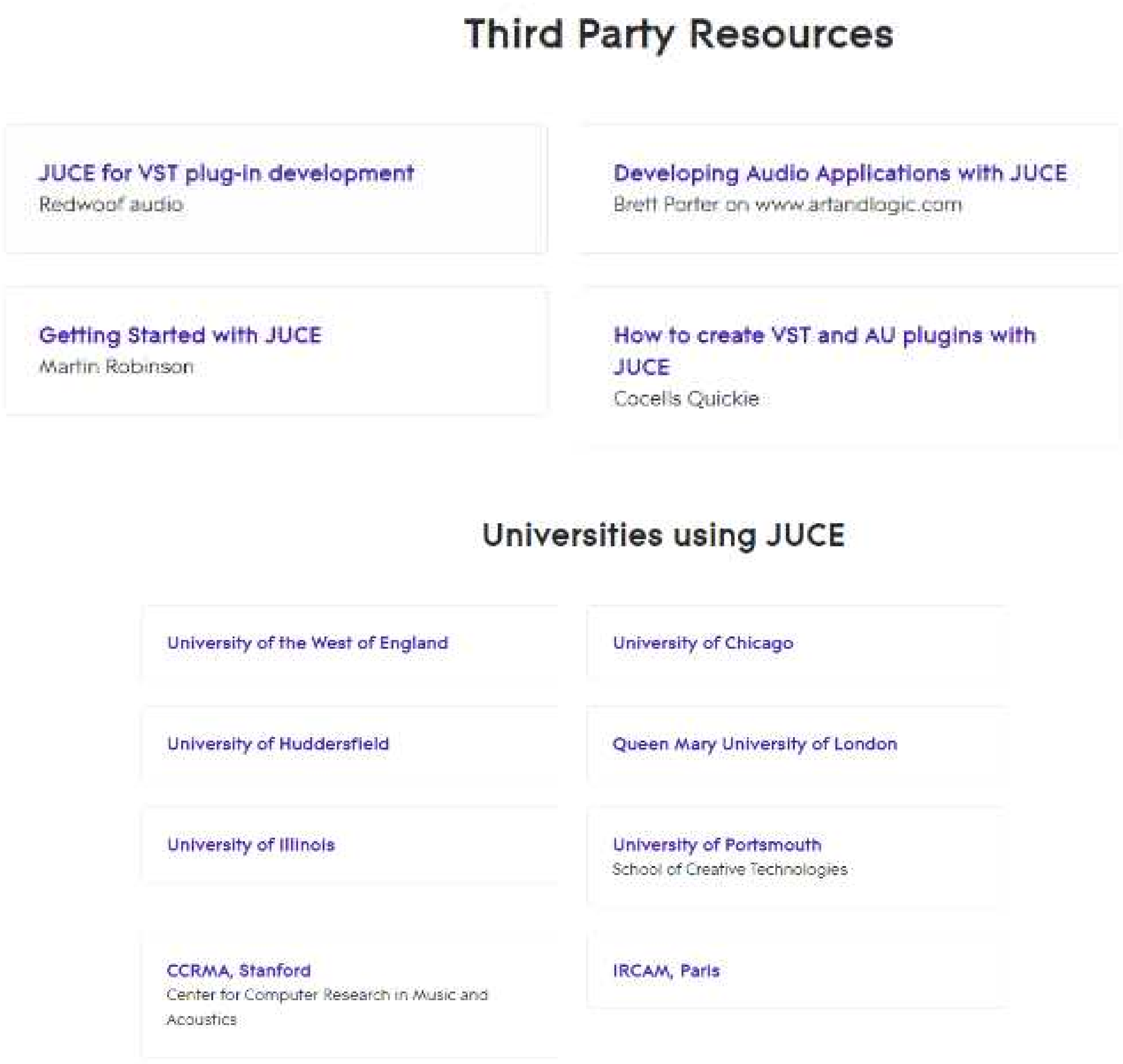
- Cross Platform 지원
  - Windows XP, Vista, 7, 8, 10
  - Mac OS X 10.5 버전 이상
  - Linux kernel 2.6 버전 이상
  - iOS 3 버전 이상
  - Android NDK-v5 이상

- 이중 라이선스
  - GNU General Public License v.3
  - ISC (juce_core, juce_audio_devices, juce_audio_basics,
    juce_event, juce_blocks)

	JUCE Personal Free	JUCE Indie $35	JUCE Pro $65	Education Free
Splash-screen	'made with JUCE' splashscreen	Custom animation or none	Custom animation or none	'made with JUCE' splashscreen
Revenue or funding limit	$50k	$200k	No limit	No limit
Minimum commitment		12 months	12 months	
One-off perpetual price		$700	$1,300	

`made with JUCE` =>

Made with
JUCE



Third Party Resources

JUCE for VST plug-in development
Redwoof audio

Developing Audio Applications with JUCE
Brett Porter on www.artandlogic.com

Getting Started with JUCE
Martin Robinson

How to create VST and AU plugins with JUCE
Cocells Quickie

Universities using JUCE

University of the West of England

University of Chicago

University of Huddersfield

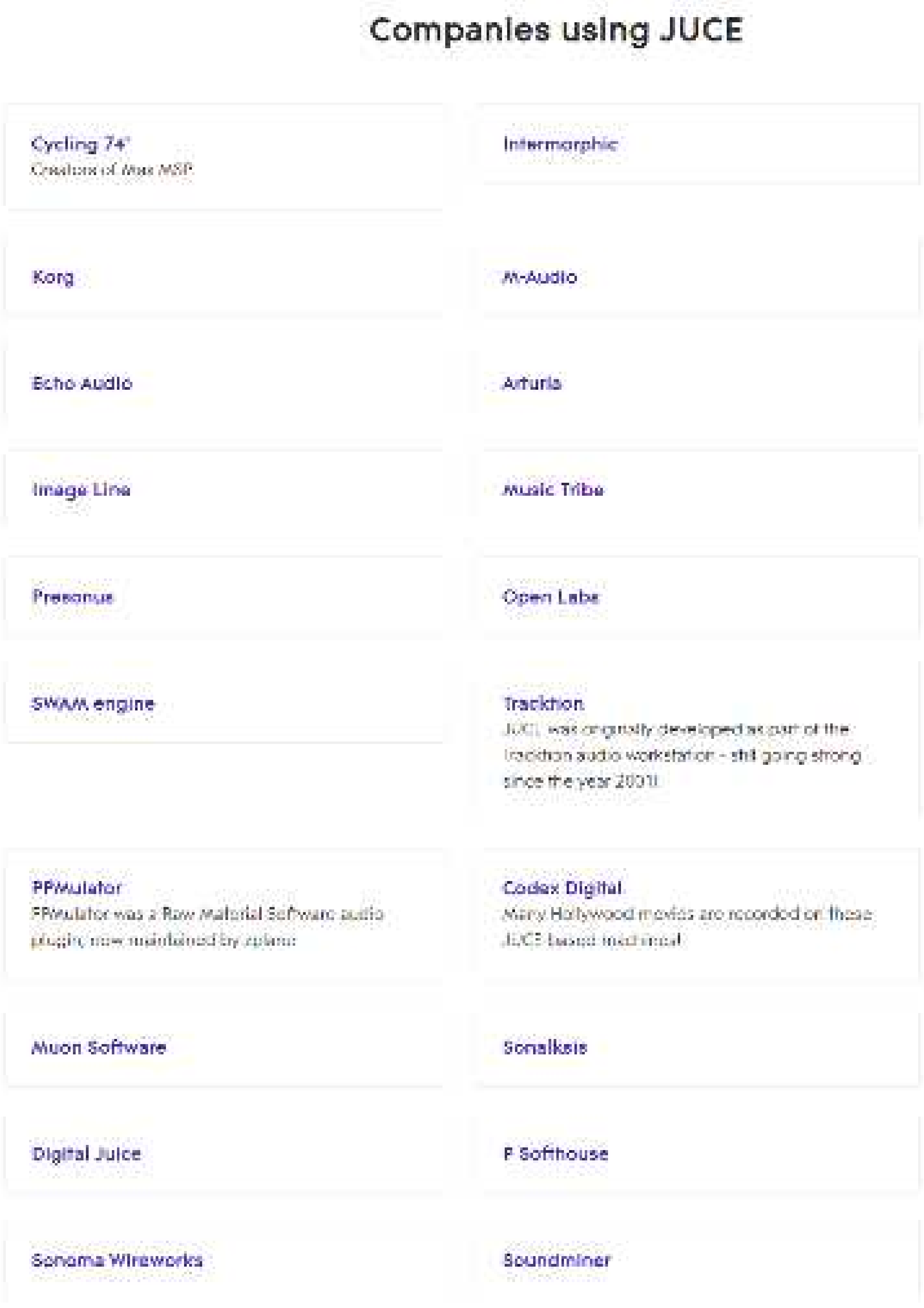
Queen Mary University of London

University of Illinois

University of Portsmouth
School of Creative Technologies

CCRMA, Stanford
Center for Computer Research in Music and Acoustics

IRCAM, Paris



Companies using JUCE

Cycling 74'
Creators of Max MSP

Intermorphic

Korg

M-Audio

Echo Audio

Arturia

Image Line

Music Tribe

Presonus

Open Labs

SWAM engine

Tracktion
JUCE was originally developed as part of the Tracktion audio workstation - still going strong since the year 2001!

PPMulator
PPMulator was a Raw Material Software audio plugin, now maintained by xplane

Codex Digital
Many Hollywood movies are recorded on these JUCE based machines!

Muon Software

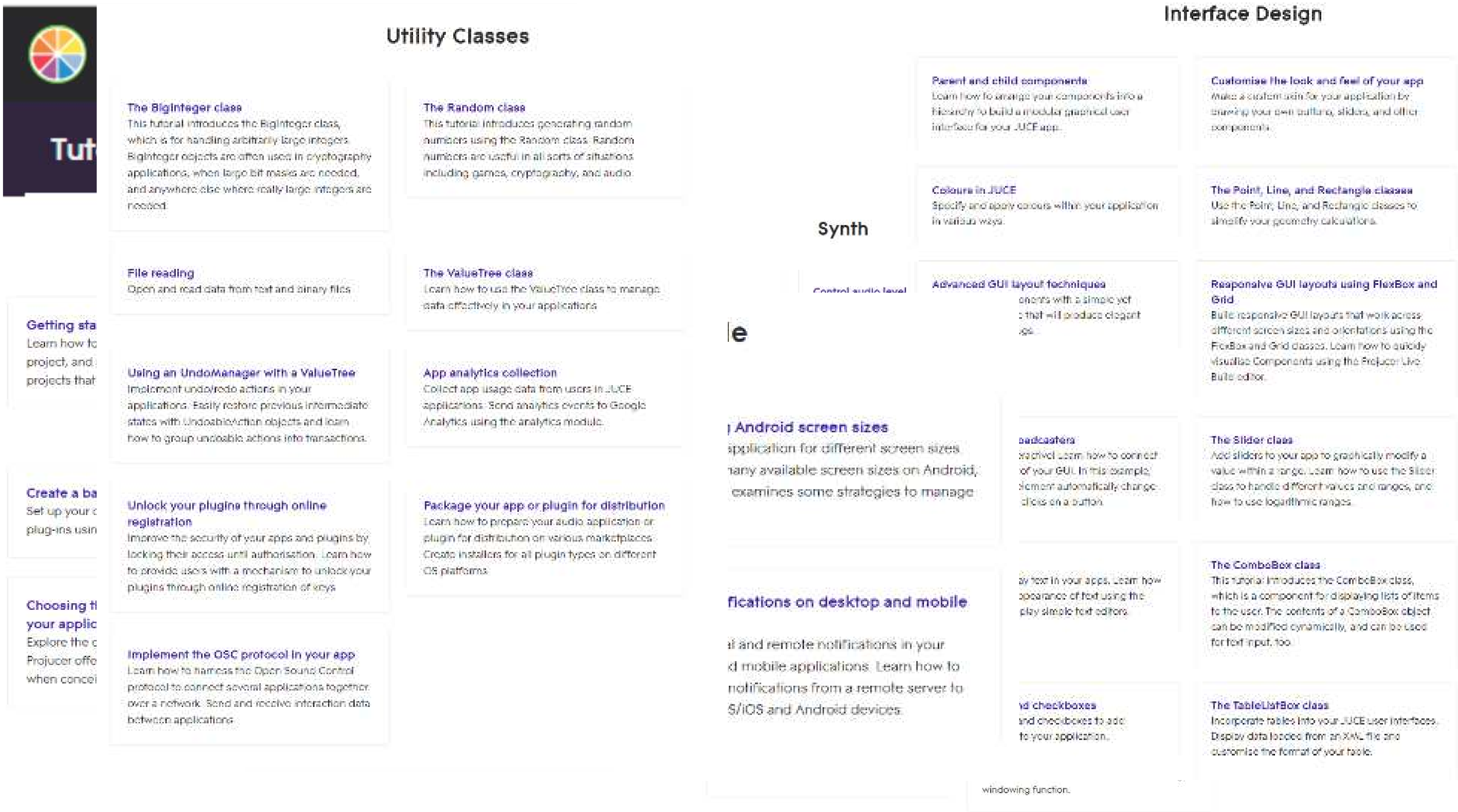
Sonalksis

Digital Juice

F Softhouse

Sonoma Wireworks

Soundminer



Tut

Getting sta
Learn how to
project, and
projects that

Create a ba
Set up your c
plug-ins usin

**Choosing th
your applic**
Explore the c
Projucer offe
when concei

Utility Classes

The BigInteger class
This tutorial introduces the BigInteger class, which is for handling arbitrarily large integers. BigInteger objects are often used in cryptography applications, when large bit masks are needed, and anywhere else where really large integers are needed.

The Random class
This tutorial introduces generating random numbers using the Random class. Random numbers are useful in all sorts of situations including games, cryptography, and audio.

File reading
Open and read data from text and binary files.

The ValueTree class
Learn how to use the ValueTree class to manage data effectively in your applications.

Using an UndoManager with a ValueTree
Implement undo/redo actions in your applications. Easily restore previous intermediate states with UndoableAction objects and learn how to group undoable actions into transactions.

App analytics collection
Collect app usage data from users in JUCE applications. Send analytics events to Google Analytics using the analytics module.

Unlock your plugins through online registration
Improve the security of your apps and plugins by locking their access until authorisation. Learn how to provide users with a mechanism to unlock your plugins through online registration of keys.

Package your app or plugin for distribution
Learn how to prepare your audio application or plugin for distribution on various marketplaces. Create installers for all plugin types on different OS platforms.

Implement the OSC protocol in your app
Learn how to harness the Open Sound Control protocol to connect several applications together over a network. Send and receive interaction data between applications.

Synth

Control audio level

e

Android screen sizes
application for different screen sizes
many available screen sizes on Android,
examines some strategies to manage

fications on desktop and mobile
al and remote notifications in your
d mobile applications. Learn how to
notifications from a remote server to
S/iOS and Android devices.

windowing function.

Interface Design

Parent and child components
Learn how to arrange your components into a hierarchy to build a modular graphical user interface for your JUCE app.

Customise the look and feel of your app
Make a custom skin for your application by drawing your own buttons, sliders, and other components.

Colours in JUCE
Specify and apply colours within your application in various ways.

The Point, Line, and Rectangle classes
Use the Point, Line, and Rectangle classes to simplify your geometry calculations.

Advanced GUI layout techniques
onents with a simple yet
that will produce elegant
gs.

Responsive GUI layouts using FlexBox and Grid
Build responsive GUI layouts that work across different screen sizes and orientations using the FlexBox and Grid classes. Learn how to quickly visualise Components using the Projucer Live Build editor.

oadcasters
eactive! Learn how to connect
of your GUI. In this example,
element automatically change
clicks on a button.

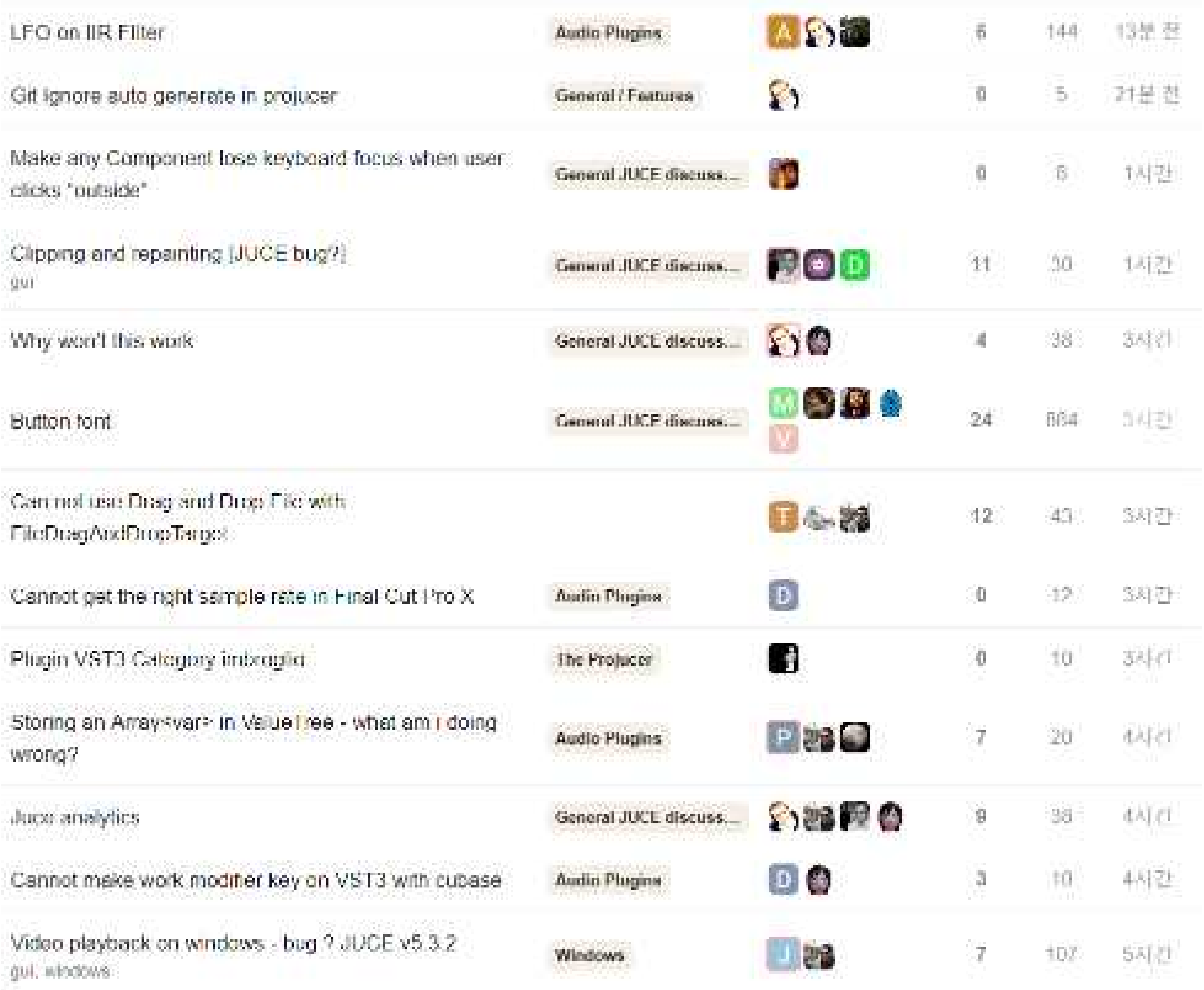
The Slider class
Add sliders to your app to graphically modify a value within a range. Learn how to use the Slider class to handle different values and ranges, and how to use logarithmic ranges.

ay text in your apps. Learn how
ppearance of text using the
play simple text editors.

The ComboBox class
This tutorial introduces the ComboBox class, which is a component for displaying lists of items to the user. The contents of a ComboBox object can be modified dynamically, and can be used for text input, too.

d checkboxes
and checkboxes to add
to your application.

The TableListBox class
Incorporate tables into your JUCE user interfaces. Display data loaded from an XML file and customise the format of your table.



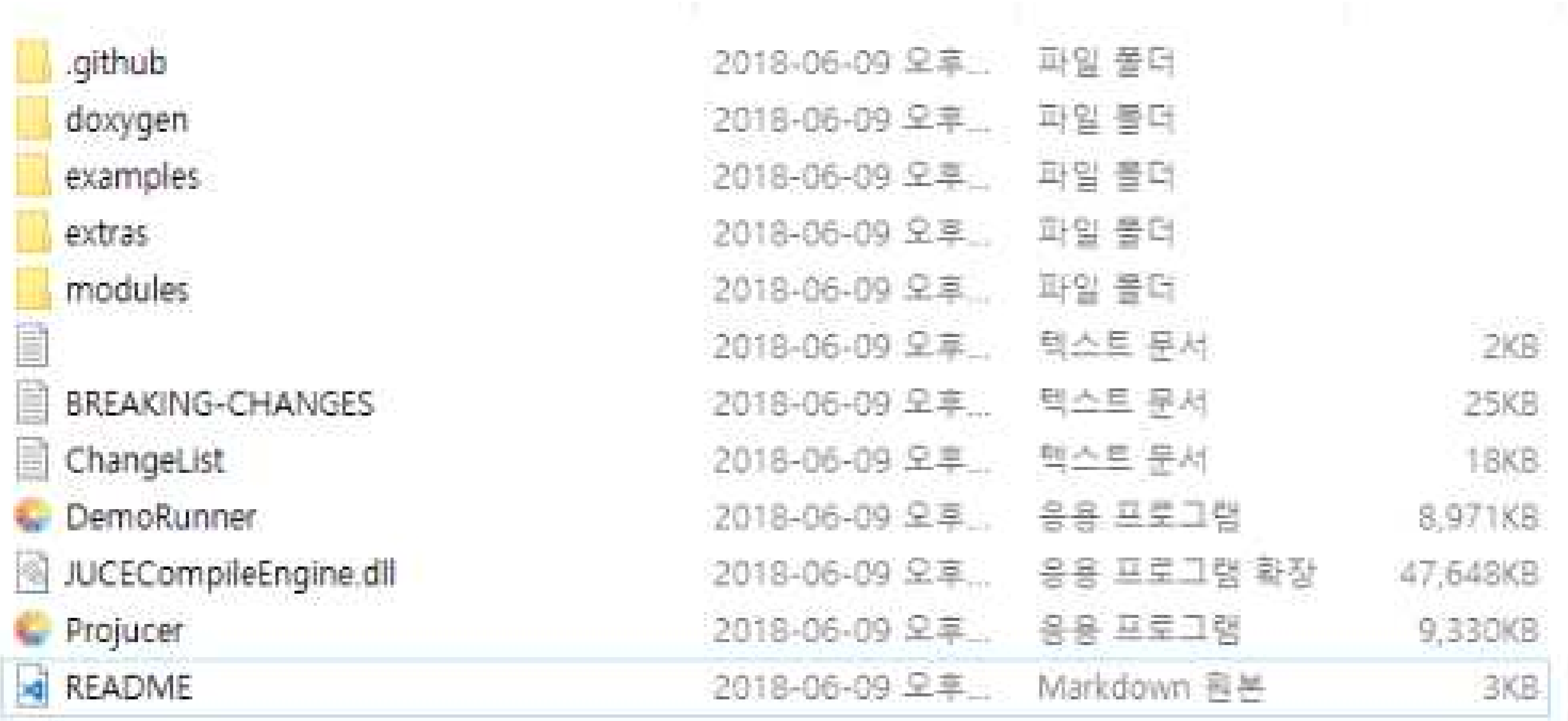
LFO on IIR Filter	Audio Plugins	A	6	144	13분 전
Git ignore auto generate in projucer	General / Features		0	5	21분 전
Make any Component lose keyboard focus when user clicks "outside"	General JUCE discuss...		0	6	1시간
Clipping and repainting [JUCE bug?] gui	General JUCE discuss...	D	11	30	1시간
Why won't this work	General JUCE discuss...		4	38	3시간
Button font	General JUCE discuss...	M V	24	864	3시간
Can not use Drag and Drop File with FileDragAndDropTarget		T	12	43	3시간
Cannot get the right sample rate in Final Cut Pro X	Audio Plugins	D	0	12	3시간
Plugin VST3 Category imbroglio	The Projucer		0	10	3시간
Storing an Array<var> in ValueTree - what am I doing wrong?	Audio Plugins	P	7	20	4시간
Juce analytics	General JUCE discuss...		9	38	4시간
Cannot make work modifier key on VST3 with cubase	Audio Plugins	D	3	10	4시간
Video playback on windows - bug ? JUCE v5.3.2 gui, windows	Windows	J	7	107	5시간

JUCE 프레임워크 설치

Download JUCE => https://shop.juce.com/get-juce

=> https://github.com/WeAreROLI/JUCE.git

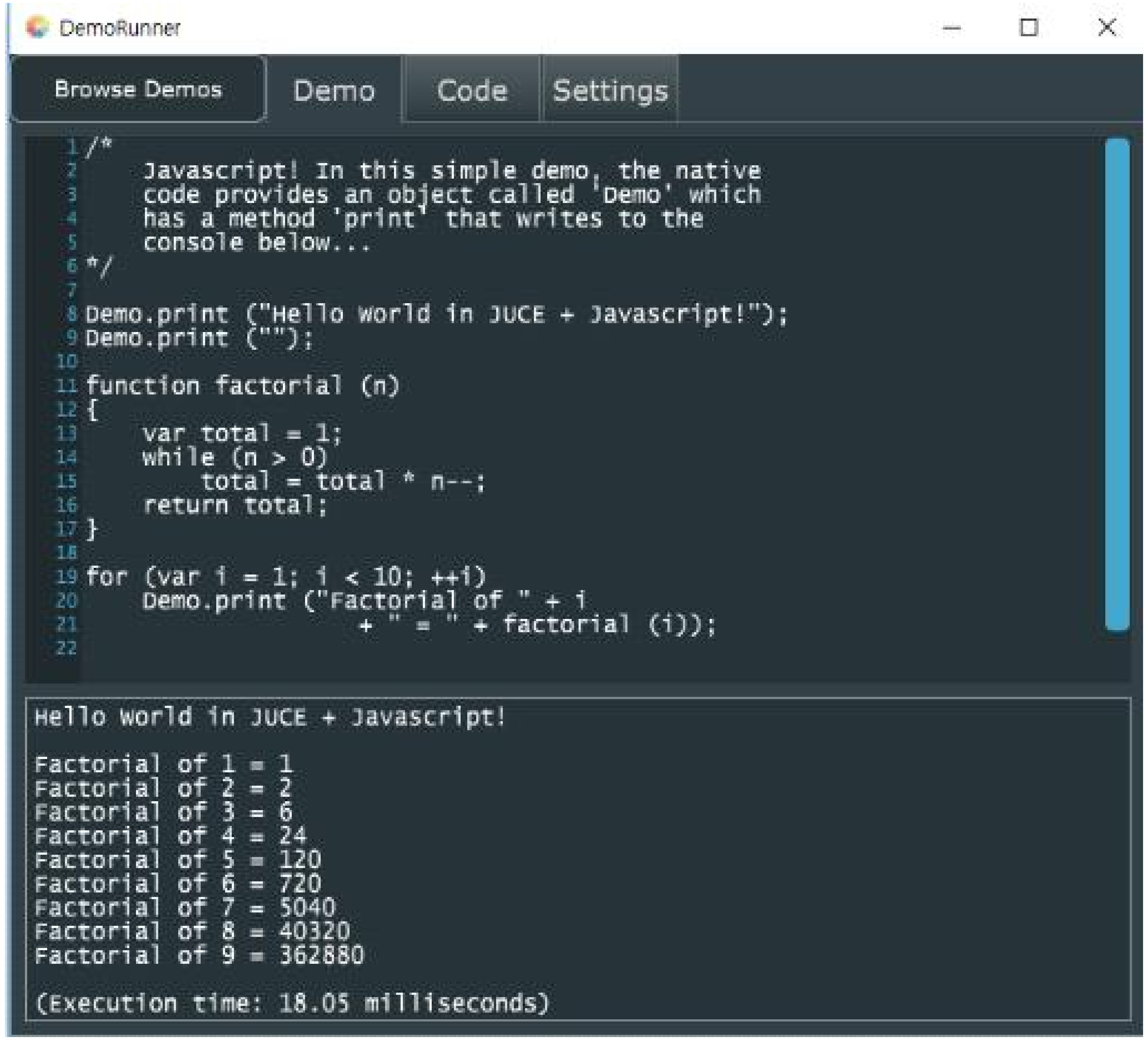
폴더 확인



.github	2018-06-09 오후...	파일 폴더	
doxygen	2018-06-09 오후...	파일 폴더	
examples	2018-06-09 오후...	파일 폴더	
extras	2018-06-09 오후...	파일 폴더	
modules	2018-06-09 오후...	파일 폴더	
	2018-06-09 오후...	텍스트 문서	2KB
BREAKING-CHANGES	2018-06-09 오후...	텍스트 문서	25KB
ChangeList	2018-06-09 오후...	텍스트 문서	18KB
DemoRunner	2018-06-09 오후...	응용 프로그램	8,971KB
JUCECompileEngine.dll	2018-06-09 오후...	응용 프로그램 확장	47,648KB
Projucer	2018-06-09 오후...	응용 프로그램	9,330KB
README	2018-06-09 오후...	Markdown 원본	3KB

DemoRunner?

- JUCE로 할 수 있는 것을 보여주는 프로그램.
- 많은 예제 프로그램이 코드와 함께 DemoRunner에 들어있다.
- API 사용하는 방법을 참고할 때 유용하게 쓰임.



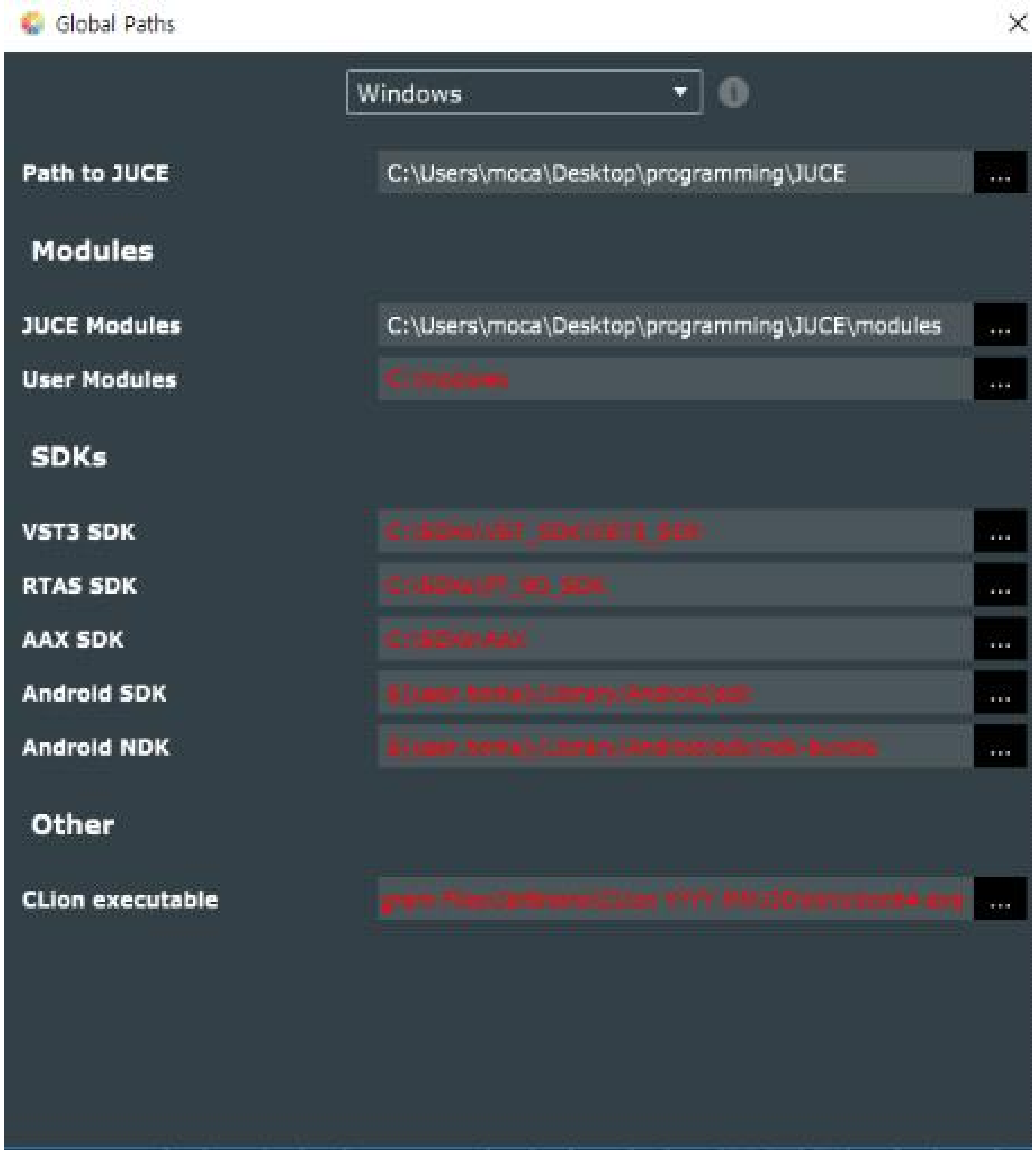
```
/*
    Javascript! In this simple demo, the native
    code provides an object called 'Demo' which
    has a method 'print' that writes to the
    console below...
*/

Demo.print ("Hello World in JUCE + Javascript!");
Demo.print ("");

function factorial (n)
{
    var total = 1;
    while (n > 0)
        total = total * n--;
    return total;
}

for (var i = 1; i < 10; ++i)
    Demo.print ("Factorial of " + i
                + " = " + factorial (i));
```

```
Hello World in JUCE + Javascript!

Factorial of 1 = 1
Factorial of 2 = 2
Factorial of 3 = 6
Factorial of 4 = 24
Factorial of 5 = 120
Factorial of 6 = 720
Factorial of 7 = 5040
Factorial of 8 = 40320
Factorial of 9 = 362880

(Execution time: 18.05 milliseconds)
```

Projucer?

- JUCE 프로젝트를 관리하기 위한 프로젝트 매니져
- Projucer 실행시 JUCE 로그인 및 폴더 Path 설정 요구
- File -> Global Paths에서 Path 변경 가능

- VST3, RTAS, AAX
  - 오디오 관련 플러그인
- Android SDK,NDK는 안드로이드 개발할때 설정



Windows

Path to JUCE: C:\Users\moca\Desktop\programming\JUCE

Modules

JUCE Modules: C:\Users\moca\Desktop\programming\JUCE\modules

User Modules: C:\modules

SDKs

VST3 SDK: C:\SDKs\VST_SDK\VST3_SDK

RTAS SDK: C:\SDKs\PT_90_SDK

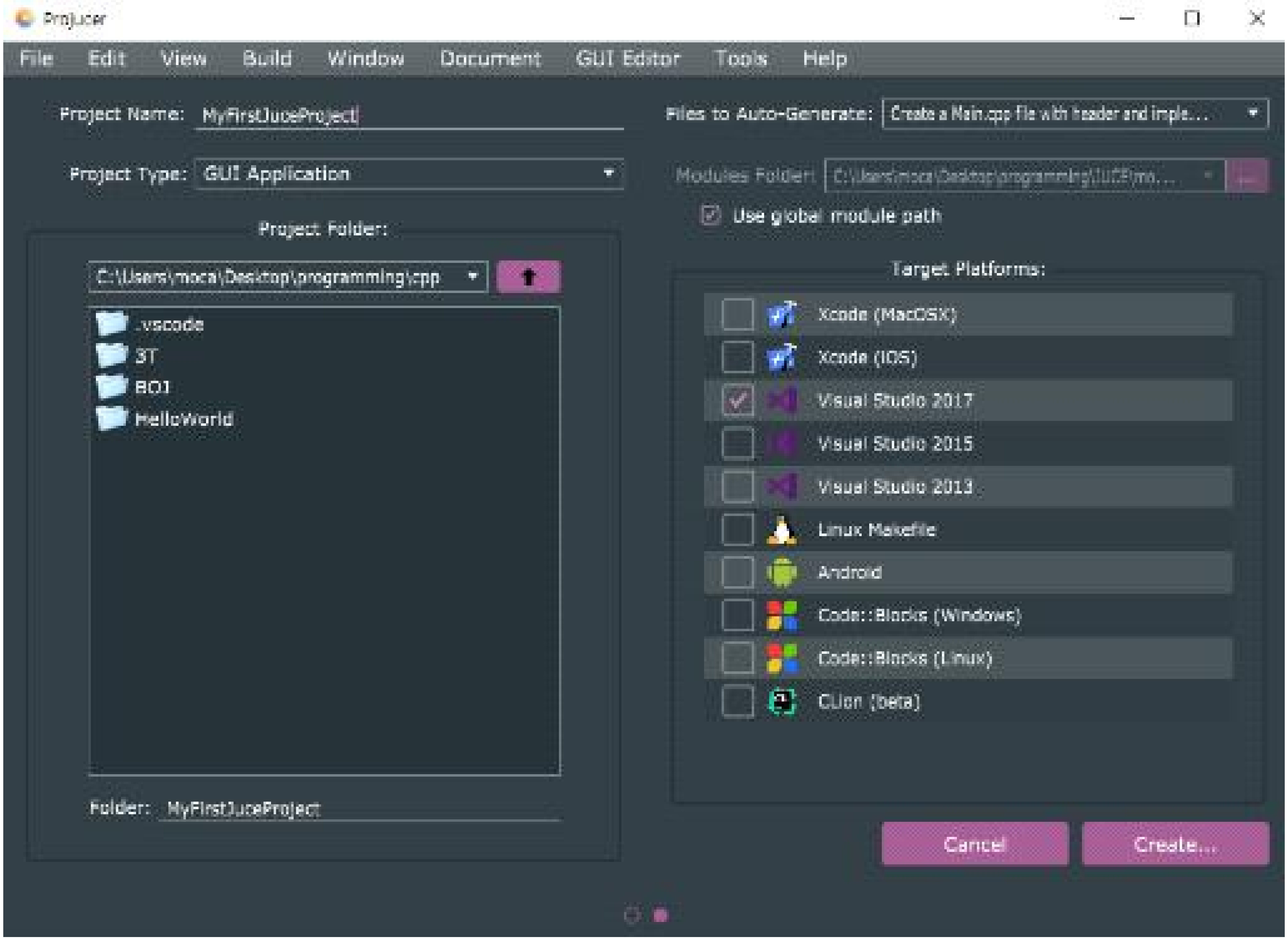
AAX SDK: C:\SDKs\AAX

Android SDK: ${user.home}/Library/Android/sdk

Android NDK: ${user.home}/Library/Android/sdk/ndk-bundle

Other

CLion executable: gram Files\JetBrains\CLion YYYY.MM.DD\bin\clion64.exe



Project Name: MyFirstJuceProject

Project Type: GUI Application

Files to Auto-Generate: Create a Main.cpp file with header and imple...

Modules Folder: C:\Users\moca\Desktop\programming\JUCE\mo...

☑ Use global module path

Project Folder:

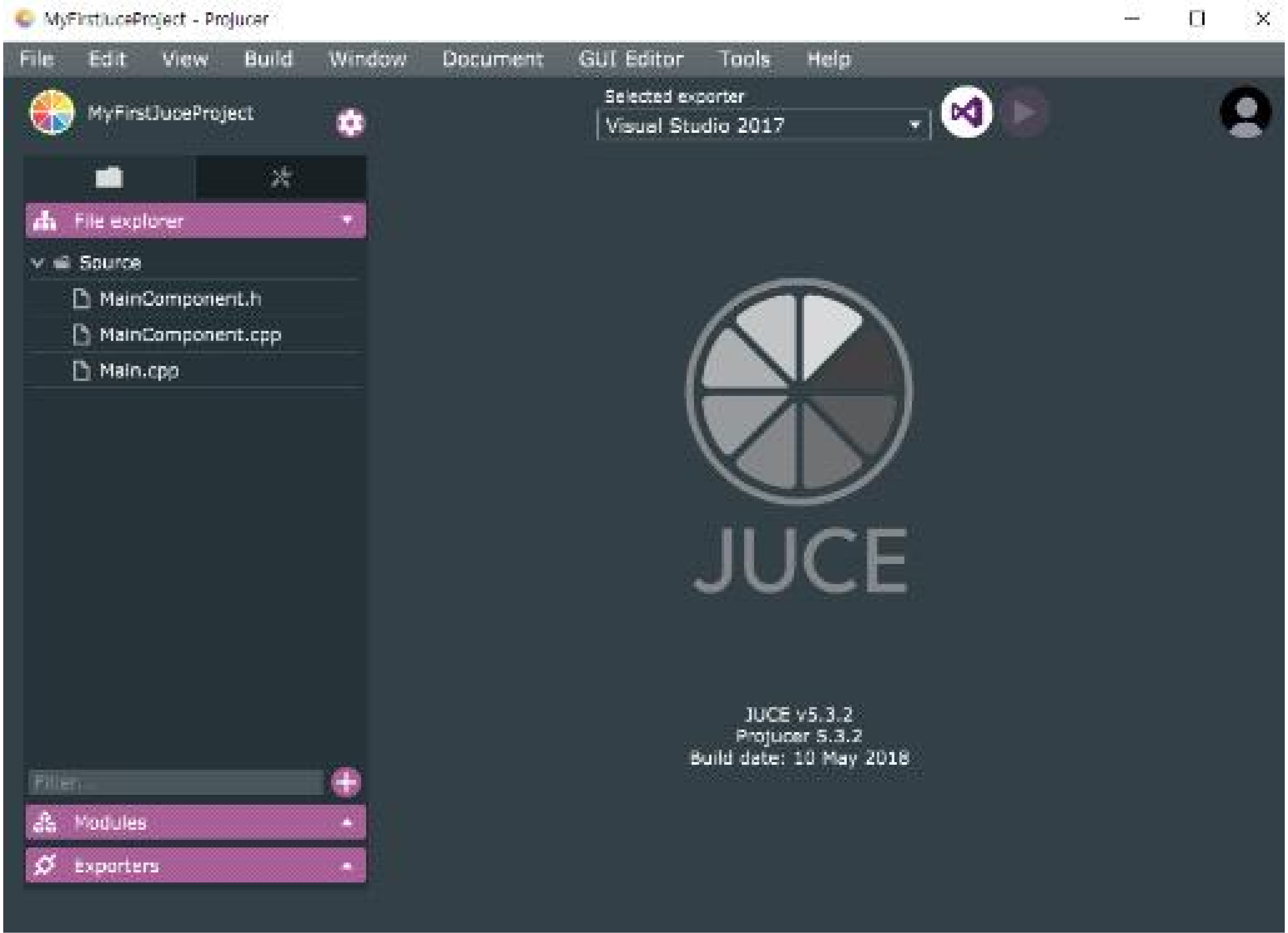
C:\Users\moca\Desktop\programming\cpp

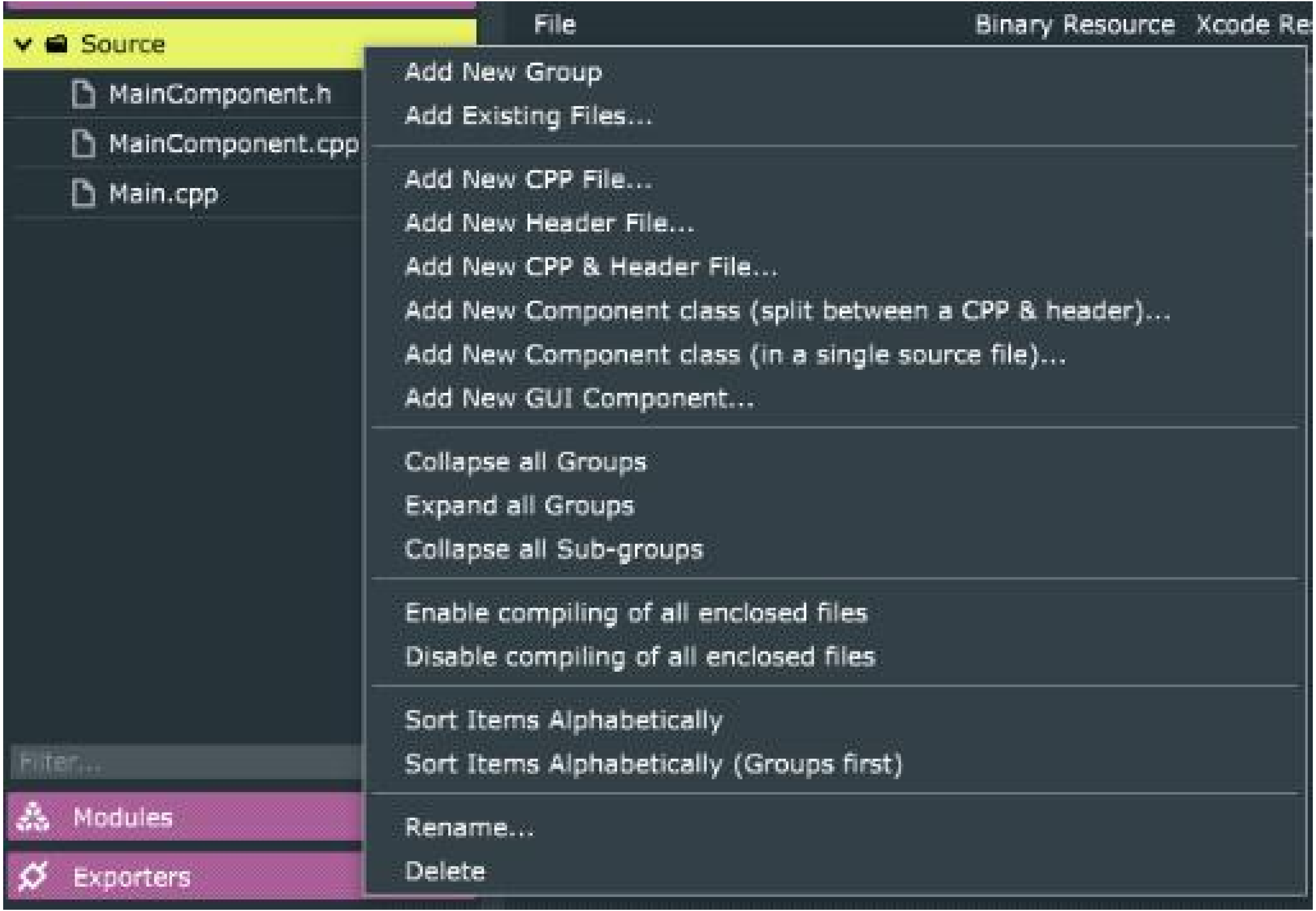
- .vscode
- 3T
- BOJ
- HelloWorld

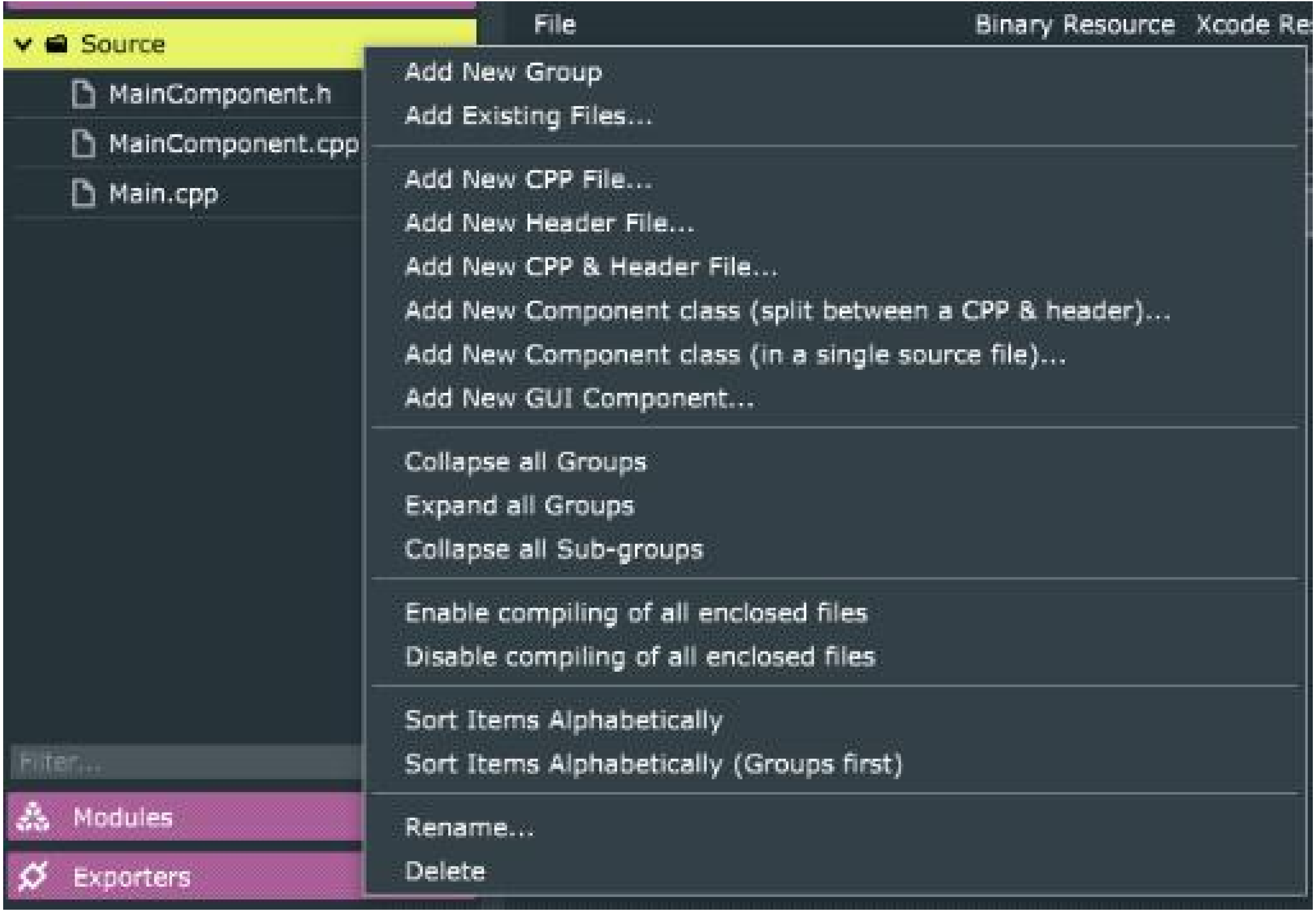
Folder: MyFirstJuceProject

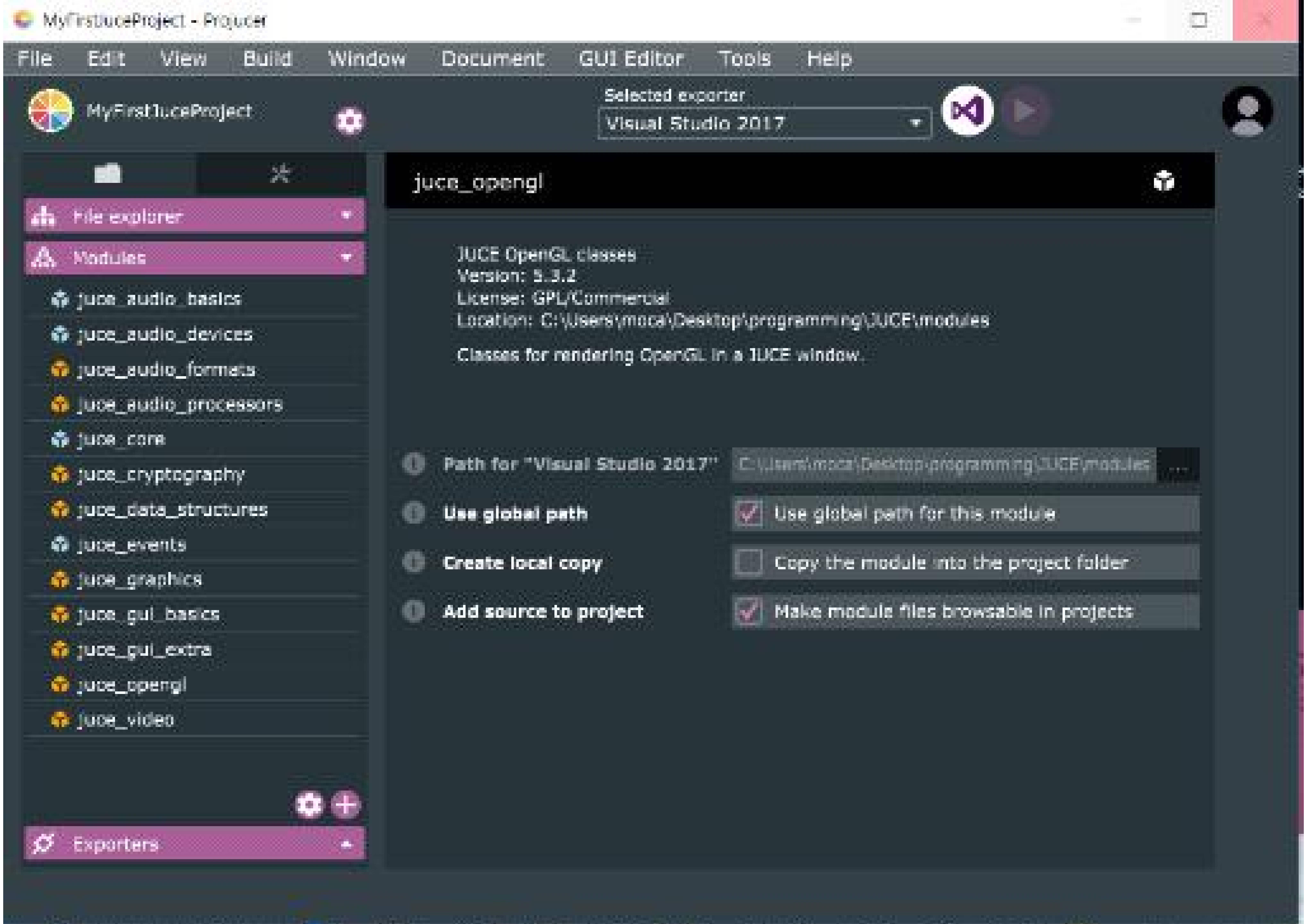
Target Platforms:

- □ Xcode (MacOSX)
- □ Xcode (iOS)
- ☑ Visual Studio 2017
- □ Visual Studio 2015
- □ Visual Studio 2013
- □ Linux Makefile
- □ Android
- □ Code::Blocks (Windows)
- □ Code::Blocks (Linux)
- □ CLion (beta)


MyFirstJuceProject - Projucer
File
Edit
View
Build
Window
Document
GUI Editor
Tools
Help
MyFirstJuceProject
Selected exporter
Visual Studio 2017
File explorer
Source
MainComponent.h
MainComponent.cpp
Main.cpp
Filter...
Modules
Exporters
JUCE
JUCE v5.3.2
Projucer 5.3.2
Build date: 10 May 2018


Source
MainComponent.h
MainComponent.cpp
Main.cpp
Filter...
Modules
Exporters
File
Binary Resource
Xcode Re
Add New Group
Add Existing Files...
Add New CPP File...
Add New Header File...
Add New CPP & Header File...
Add New Component class (split between a CPP & header)...
Add New Component class (in a single source file)...
Add New GUI Component...
Collapse all Groups
Expand all Groups
Collapse all Sub-groups
Enable compiling of all enclosed files
Disable compiling of all enclosed files
Sort Items Alphabetically
Sort Items Alphabetically (Groups first)
Rename...
Delete


File
Binary Resource
Xcode Re
Source
MainComponent.h
MainComponent.cpp
Main.cpp
Filter...
Modules
Exporters
Add New Group
Add Existing Files...
Add New CPP File...
Add New Header File...
Add New CPP & Header File...
Add New Component class (split between a CPP & header)...
Add New Component class (in a single source file)...
Add New GUI Component...
Collapse all Groups
Expand all Groups
Collapse all Sub-groups
Enable compiling of all enclosed files
Disable compiling of all enclosed files
Sort Items Alphabetically
Sort Items Alphabetically (Groups first)
Rename...
Delete



juce_audio_basics
juce_audio_devices
juce_audio_formats
juce_audio_processors
juce_core
juce_cryptography
juce_data_structures
juce_events
juce_graphics
juce_gui_basics
juce_gui_extra
juce_opengl
juce_video

juce_opengl

JUCE OpenGL classes
Version: 5.3.2
License: GPL/Commercial
Location: C:\Users\moca\Desktop\programming\JUCE\modules

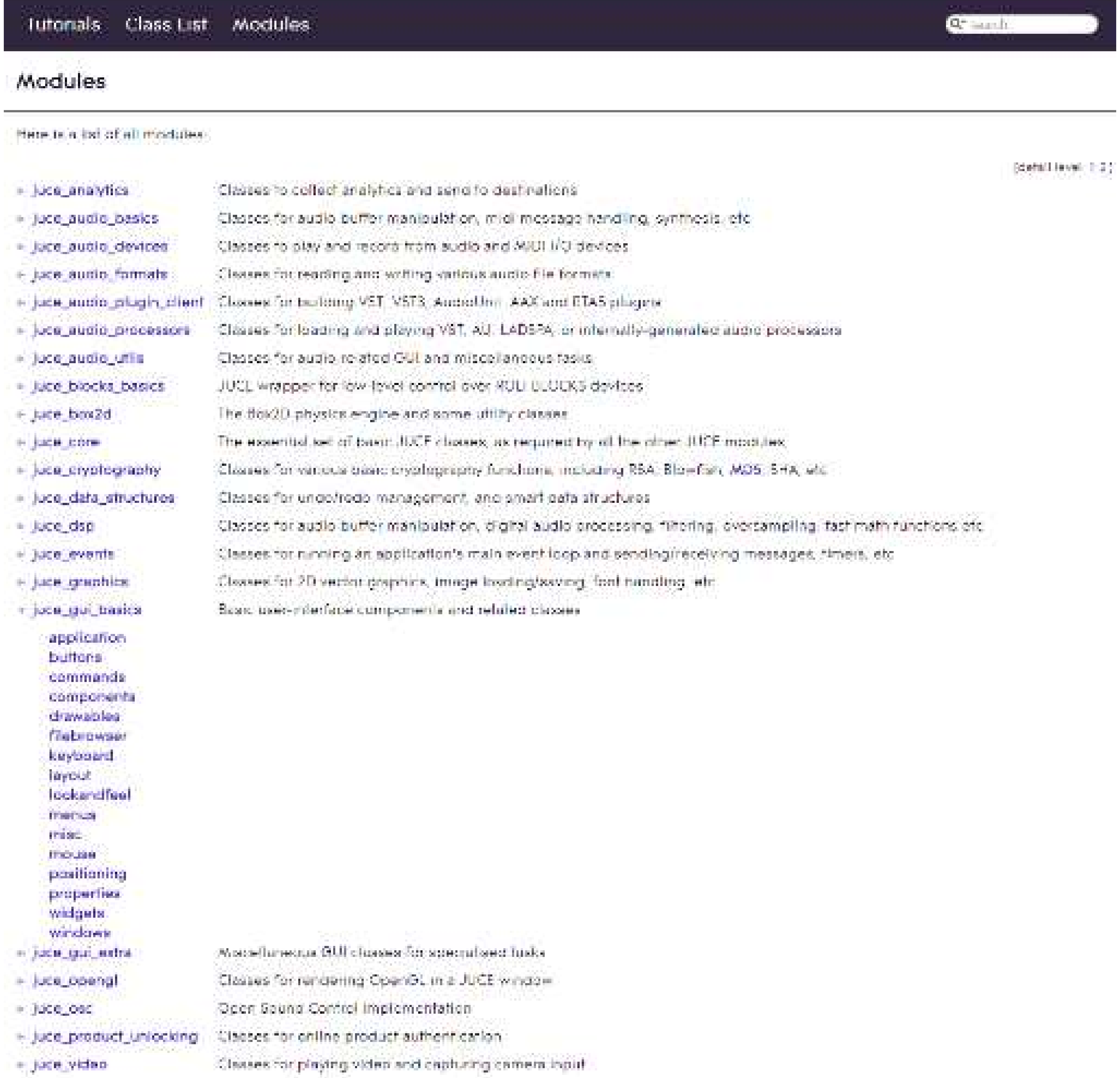
Classes for rendering OpenGL in a JUCE window.

Path for "Visual Studio 2017" C:\Users\moca\Desktop\programming\JUCE\modules ...

Use global path ☑ Use global path for this module

Create local copy □ Copy the module into the project folder

Add source to project ☑ Make module files browsable in projects https://docs.juce.com/master/modules.html

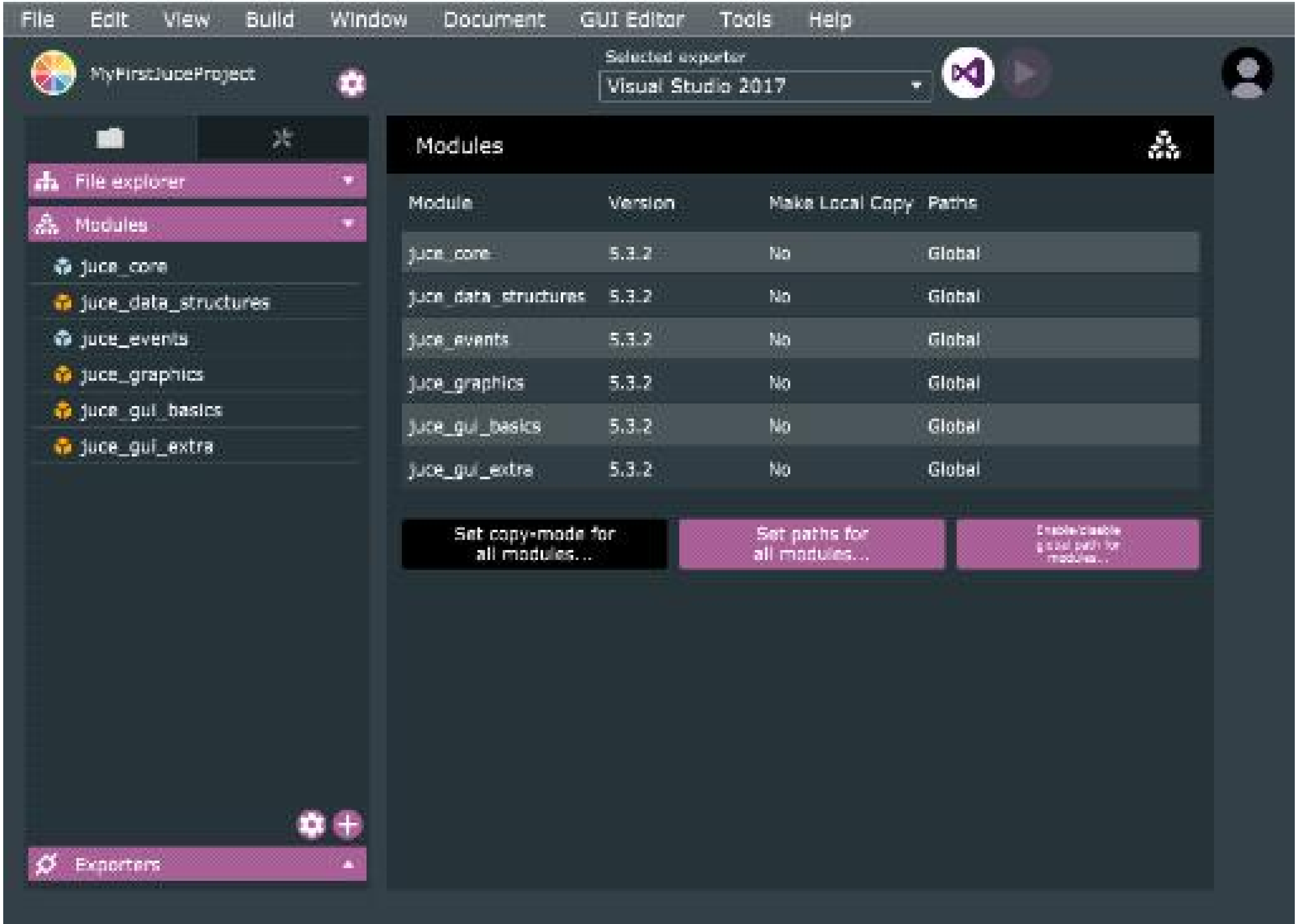


Modules

Here is a list of all modules:

[detail level 1 2]

juce_analytics	Classes to collect analytics and send to destinations
juce_audio_basics	Classes for audio buffer manipulation, midi message handling, synthesis, etc
juce_audio_devices	Classes to play and record from audio and MIDI I/O devices
juce_audio_formats	Classes for reading and writing various audio file formats
juce_audio_plugin_client	Classes for building VST, VST3, AudioUnit, AAX and RTAS plugins
juce_audio_processors	Classes for loading and playing VST, AU, LADSPA, or internally-generated audio processors
juce_audio_utils	Classes for audio-related GUI and miscellaneous tasks
juce_blocks_basics	JUCE wrapper for low-level control over ROLI BLOCKS devices
juce_box2d	The Box2D physics engine and some utility classes
juce_core	The essential set of basic JUCE classes, as required by all the other JUCE modules
juce_cryptography	Classes for various basic cryptography functions, including RSA, Blowfish, MD5, SHA, etc
juce_data_structures	Classes for undo/redo management, and smart data structures
juce_dsp	Classes for audio buffer manipulation, digital audio processing, filtering, oversampling, fast math functions etc
juce_events	Classes for running an application's main event loop and sending/receiving messages, timers, etc
juce_graphics	Classes for 2D vector graphics, image loading/saving, font handling, etc
juce_gui_basics	Basic user-interface components and related classes
application	
buttons	
commands	
components	
drawables	
filebrowser	
keyboard	
layout	
lookandfeel	
menus	
misc	
mouse	
positioning	
properties	
widgets	
windows	
juce_gui_extra	Miscellaneous GUI classes for specialised tasks
juce_opengl	Classes for rendering OpenGL in a JUCE window
juce_osc	Open Sound Control implementation
juce_product_unlocking	Classes for online product authentication
juce_video	Classes for playing video and capturing camera input



Selected exporter: Visual Studio 2017

Modules

Module	Version	Make Local Copy	Paths
juce_core	5.3.2	No	Global
juce_data_structures	5.3.2	No	Global
juce_events	5.3.2	No	Global
juce_graphics	5.3.2	No	Global
juce_gui_basics	5.3.2	No	Global
juce_gui_extra	5.3.2	No	Global

Exporters
Visual Studio 2017
Debug
Release
Licens
Locati
Classe
and se
Path fo
Use glo
Create a new
Xcode (MacO
Xcode (iOS)
Visual Studio
Visual Studio
Visual Studio
Linux Makefil
Android
Code::Blocks
Code::Blocks
CLion (beta)
File
Edit
View
Build
Window
Document
New Project... ctrl + N
New Project From Clipboard... ctrl + shift + N
New PIP... ctrl + P
Open... ctrl + O
Open Recent
Open Example
Close ctrl + alt + W
Save ctrl + S
Save As... ctrl + shift + S
Save All ctrl + alt + S
Close Project
Save Project
Open in IDE...
Save Project and Open in ID
Sign out M0ca...
About Projucer
Application Usage Data
Global Paths...
Quit
GUI Editor
Tools
Help
Selected exporter
Visual Studio 2017
✓ Visual Studio 2017
Linux Makefile
Xcode (MacOSX)
Android
oject Fo

Hello World 출력하기

- Main.cpp

- 어플리케이션이 작동하는 최소한의 코드

```
class MainWindowTutorialApplication  : public JUCEApplication
{
public:
    //...
    class MainWindow    : public DocumentWindow
    {
    public:
        MainWindow (String name)  : DocumentWindow (name,
                                        Colours::lightgrey,
                                        DocumentWindow::allButtons)
        {
            centreWithSize (getWidth(), getHeight());
            setVisible (true);
        }
        void closeButtonPressed() override
        {
            JUCEApplication::getInstance()->systemRequestedQuit();
        }
    private:
        JUCE_DECLARE_NON_COPYABLE_WITH_LEAK_DETECTOR (MainWindow)
    };
private:
    ScopedPointer<MainWindow> mainWindow;
};
```

true : foreground

false : background

C++에서 파악하기 어려운 몇 가지 실수를 예방과 누수를 잡음
메크로를 클래스에서 추가할 수 있으면 사용하는게 좋음

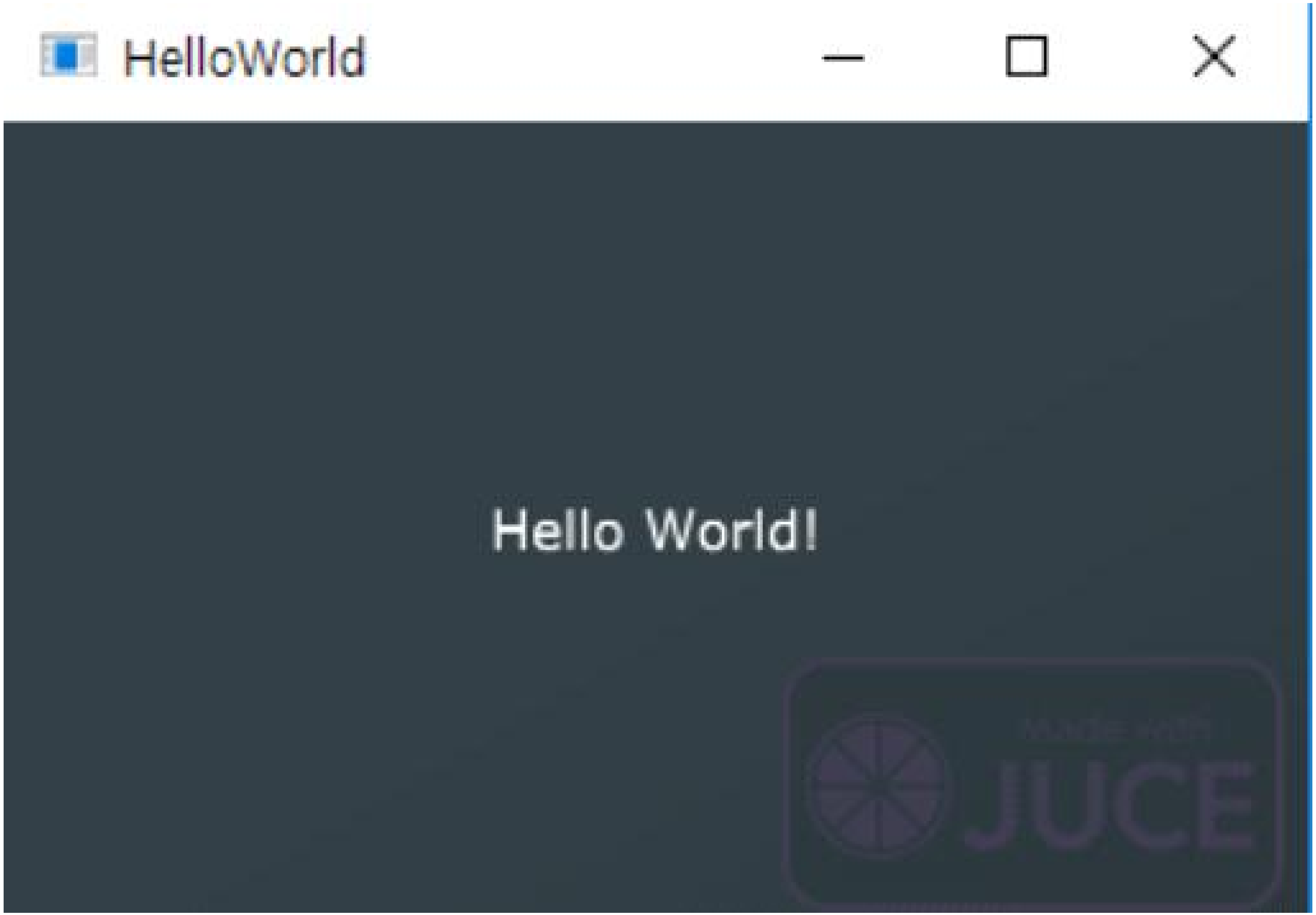
```
class MainWindowTutorialApplication  : public JUCEApplication
{
public:
    //...
    void initialise (const String& commandLine) override
    {
        mainWindow.reset (new MainWindow (getApplicationName()));
    }

   class MainWindow    : public DocumentWindow
   {
   public:
      //...
   };
};
```

- initialise() 함수가 가장 먼저 실행

```
MainComponent::MainComponent()
{
    setSize (600, 400);          // 창 크기 조절
}
void MainComponent::paint (Graphics& g)
{
    g.fillAll (getLookAndFeel().findColour (ResizableWindow::backgroundColourId));
    g.setFont (Font (16.0f));
    g.setColour (Colours::white);
    g.drawText ("Hello World!", getLocalBounds(), Justification::centred, true);
}
```

```
void MainComponent::resized()
{
}
```

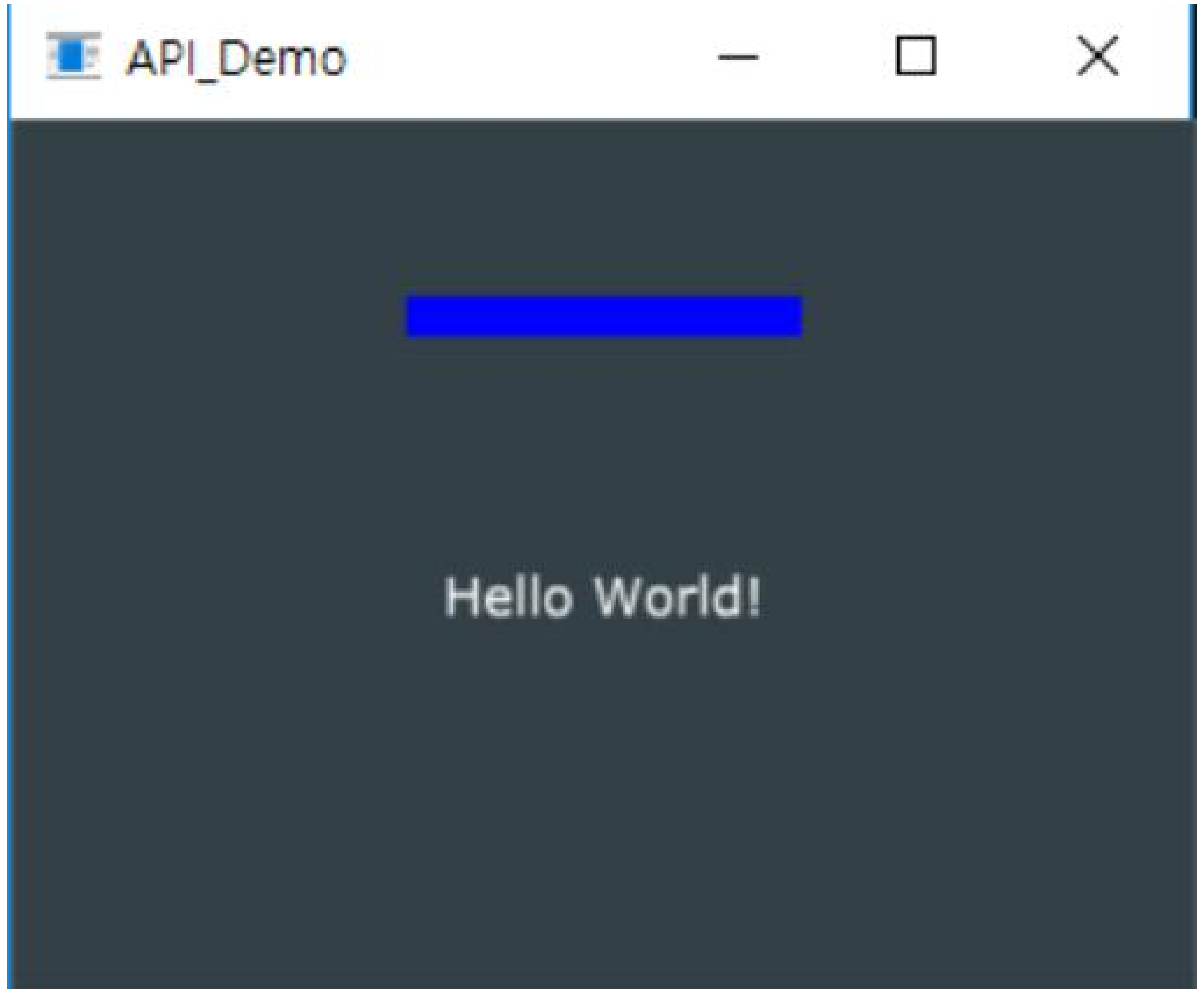
resized()를 임의로 호출하지 말자
GUI 창이 움직일때마다 자동적으로 호출된다



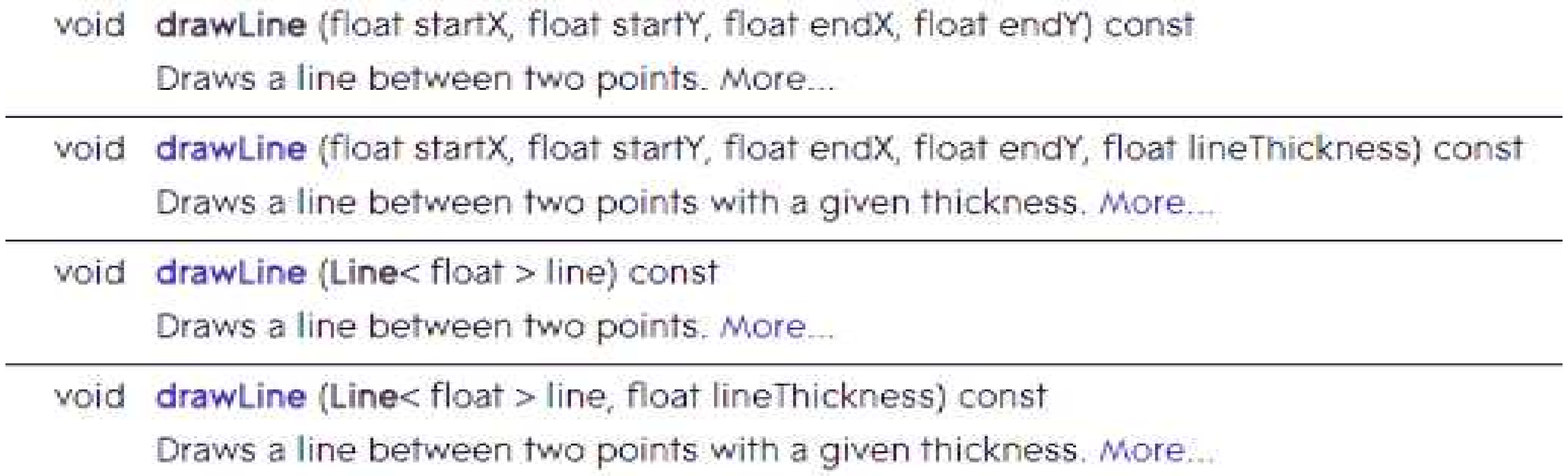
Hello World!

drawLine()

```
void MainComponent::paint (Graphics& g)
{
   // ...
   g.setColour(Colours::blue);
   g.drawLine(100, 50, 200, 50, 10.0f);
}
```



Hello World!



void drawLine (float startX, float startY, float endX, float endY) const
Draws a line between two points. More...

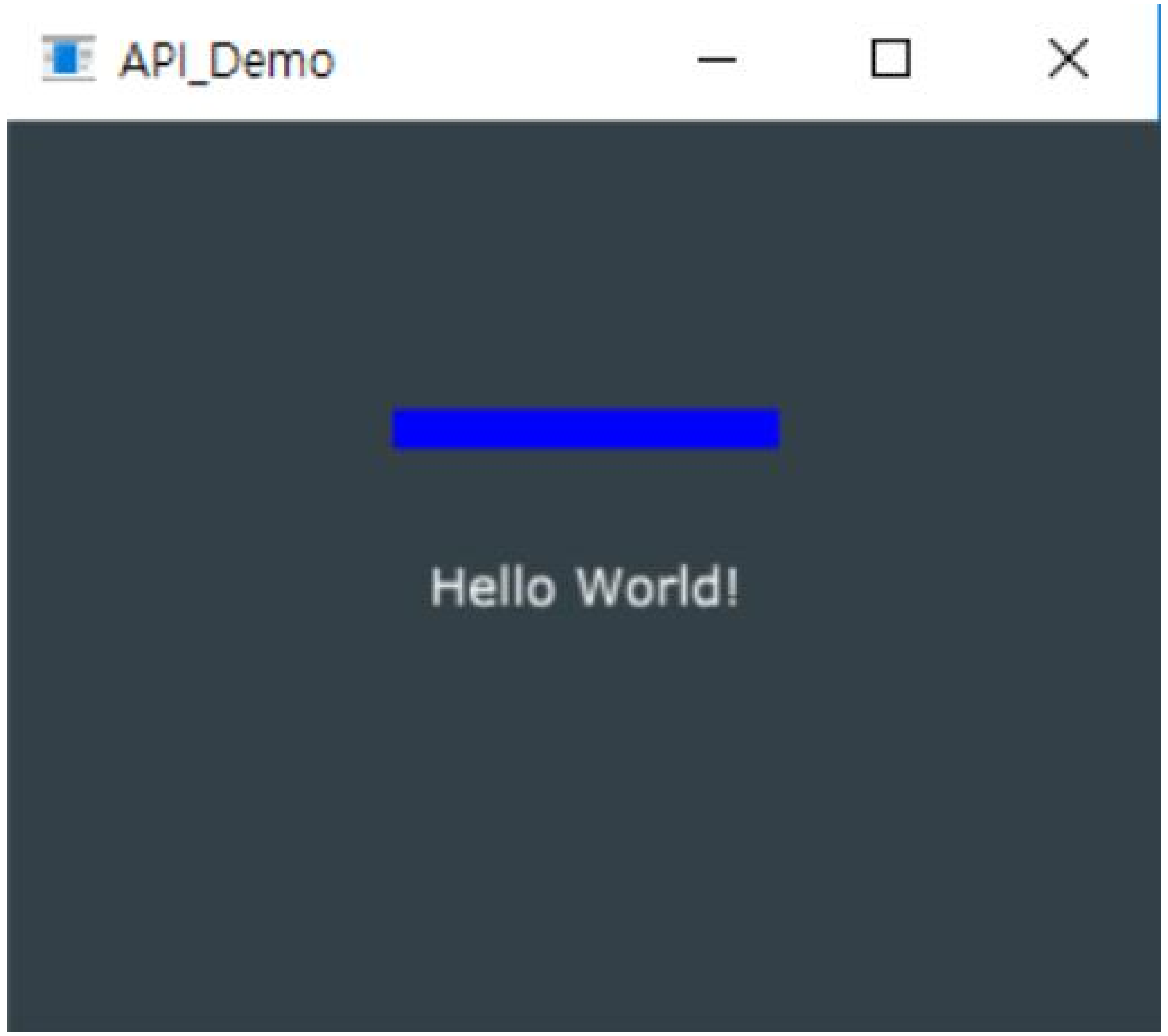
void drawLine (float startX, float startY, float endX, float endY, float lineThickness) const
Draws a line between two points with a given thickness. More...

void drawLine (Line< float > line) const
Draws a line between two points. More...

void drawLine (Line< float > line, float lineThickness) const
Draws a line between two points with a given thickness. More...

상대적으로 접근

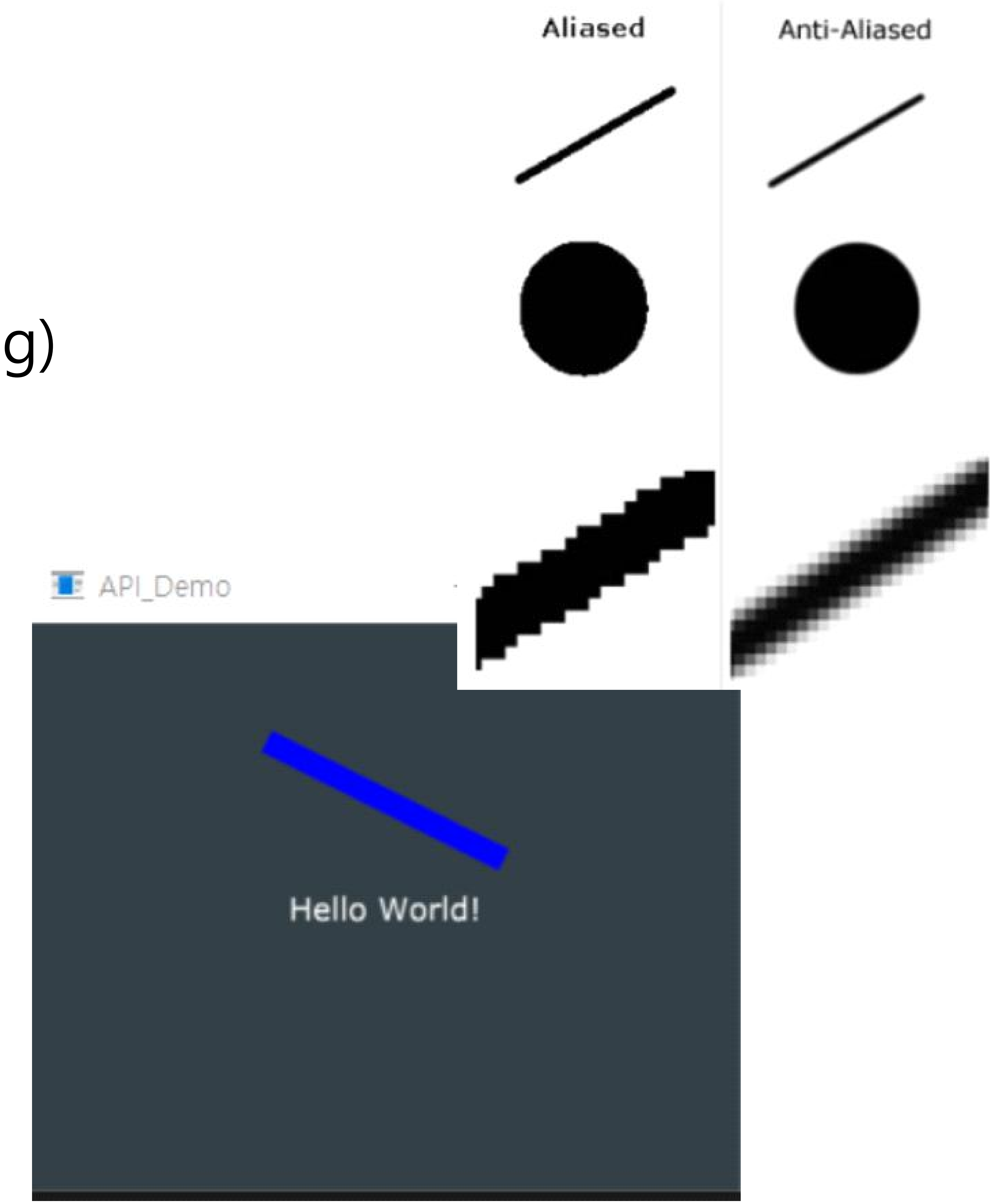
```
void MainComponent::paint (Graphics& g)
{
  // ...
  g.setColour(Colours::blue);

  g.drawLine(getWidth() / 3, getHeight() / 3,
                    getWidth() / 3 + 100 , getHeight() / 3, 10.0f);
}
```



Hello World!

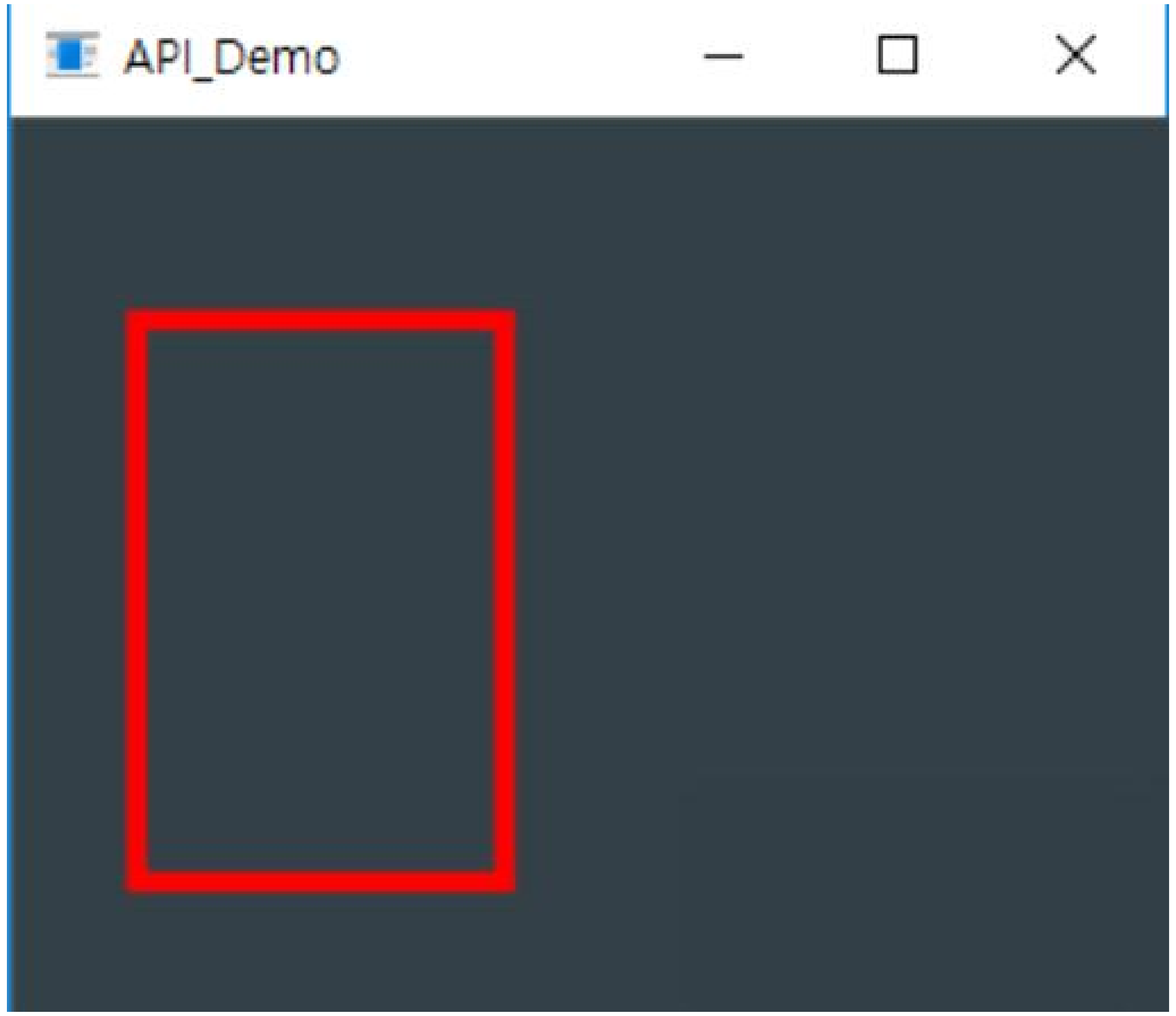
Anti-Aliasing

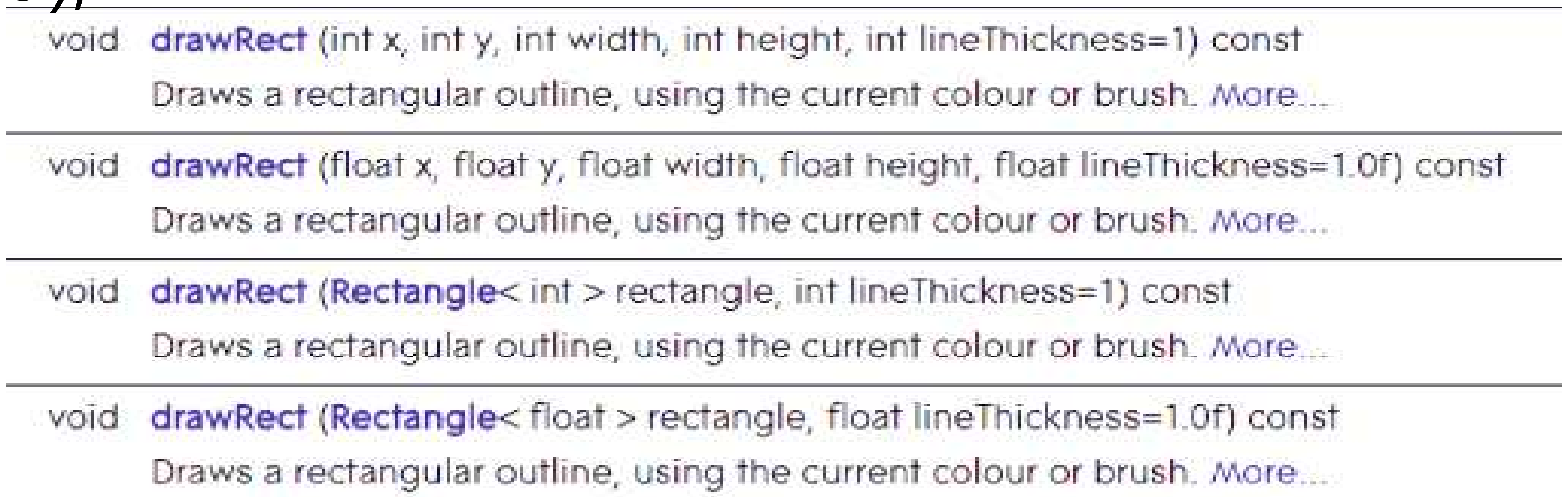
```
void MainComponent::paint (Graphics& g)
{
  // ...
  g.setColour(Colours::blue);
  g.drawLine(100, 50, 200, 100, 10.0f);
}
```



drawRect(Rectangle)

```
void MainComponent::paint (Graphics& g)
{
  // ...
  g.setColour(Colours::Red);
  g.drawRect(30, 50, 100, 150, 5);
}
```





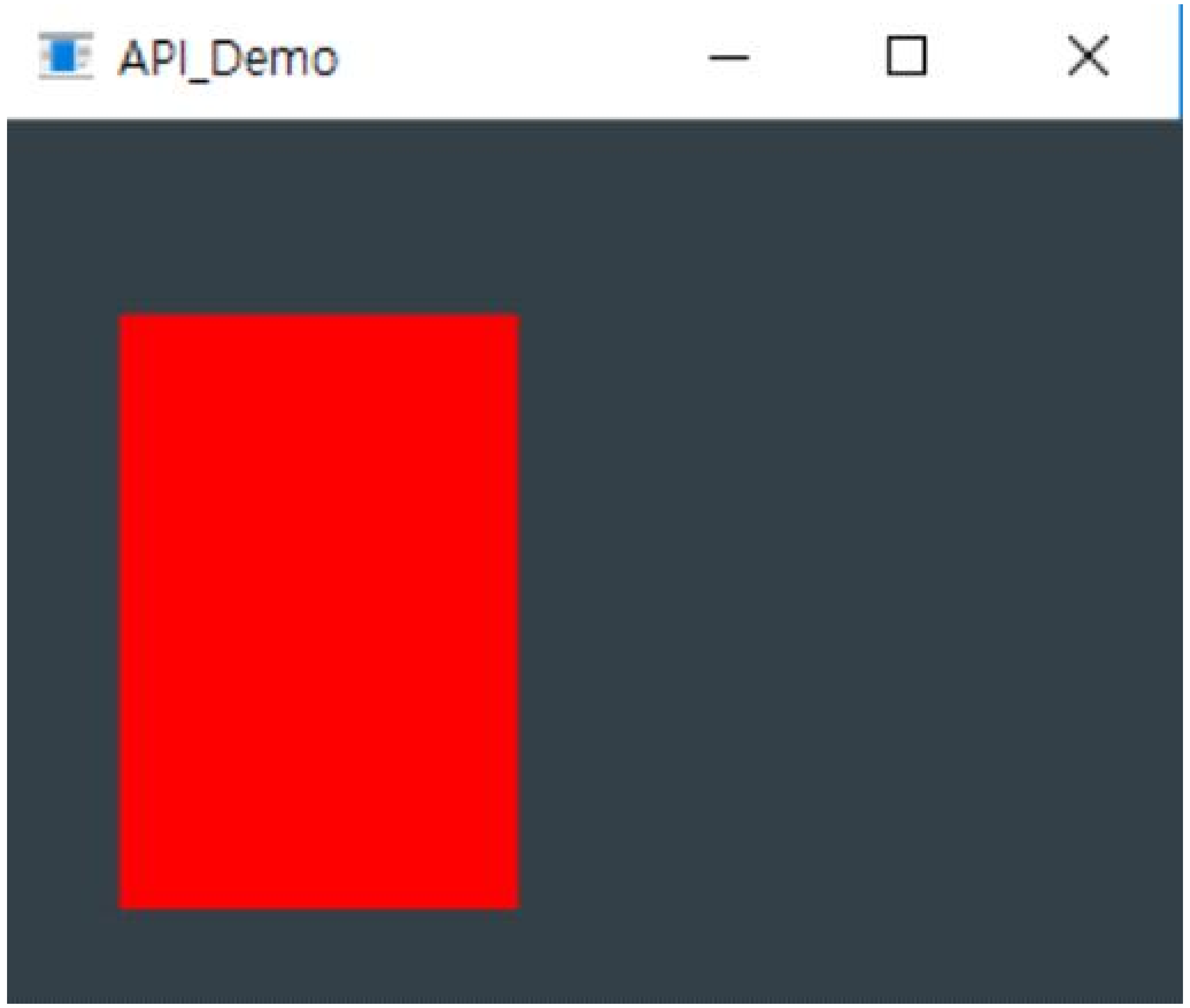
void drawRect (int x, int y, int width, int height, int lineThickness=1) const
Draws a rectangular outline, using the current colour or brush. More...

void drawRect (float x, float y, float width, float height, float lineThickness=1.0f) const
Draws a rectangular outline, using the current colour or brush. More...

void drawRect (Rectangle< int > rectangle, int lineThickness=1) const
Draws a rectangular outline, using the current colour or brush. More...

void drawRect (Rectangle< float > rectangle, float lineThickness=1.0f) const
Draws a rectangular outline, using the current colour or brush. More...

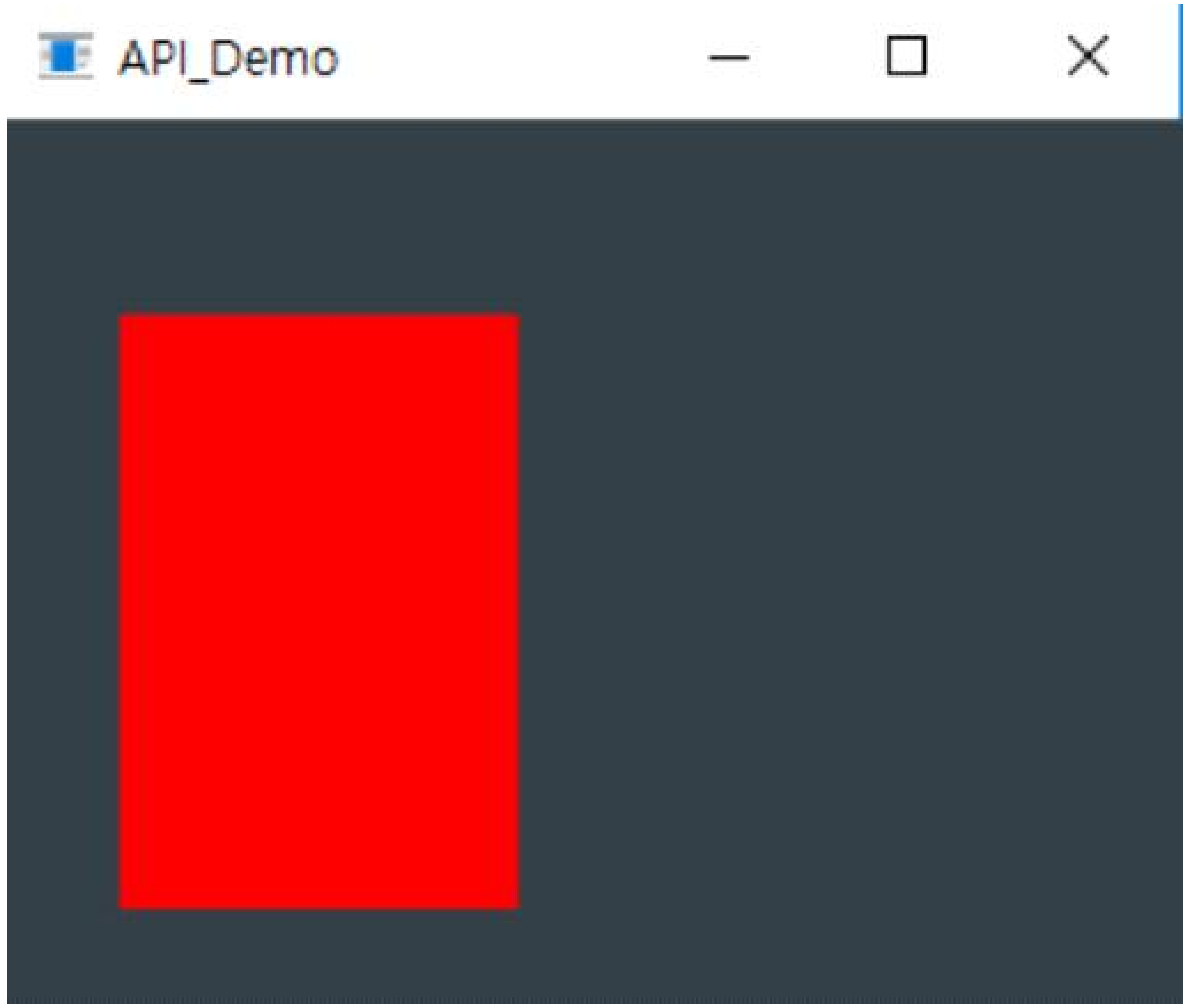
fillRect(Rectangle)



```
void MainComponent::paint (Graphics& g)
{
  // ...
  g.setColour(Colours::red);
  g.drawRect(30, 50, 100, 150, 5);
  g.setColour(Colours::green);
  g.fillRect(30, 50 ,100, 150);
}
```

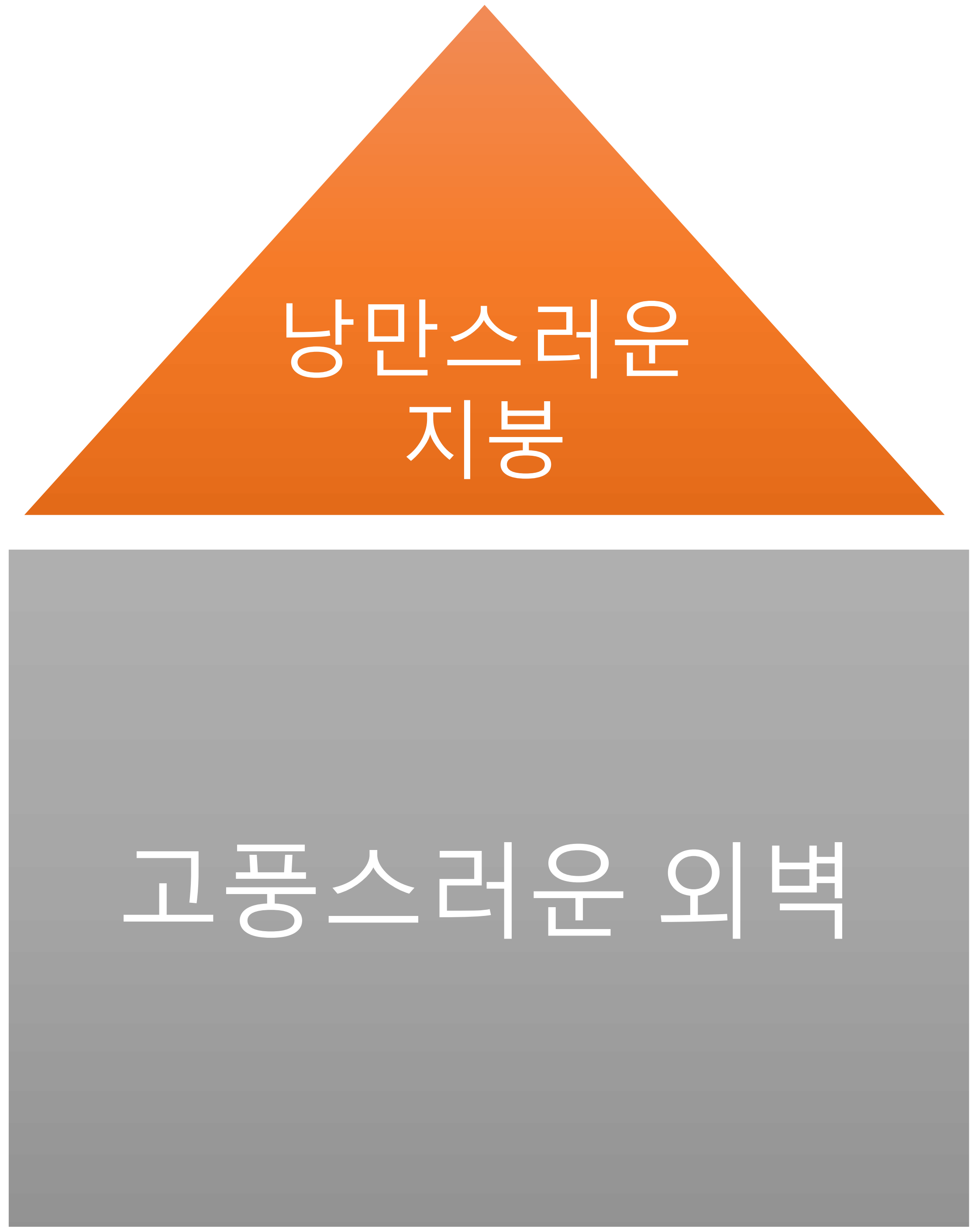
fillRect(Rectangle)

```
void MainComponent::paint (Graphics& g)
{
  // ...
  Rectangle<int> rect{ 30, 50, 100, 150 };
  g.setColour(Colours::red);
  g.drawRect(rect);
  g.fillRect(rect);
}
```



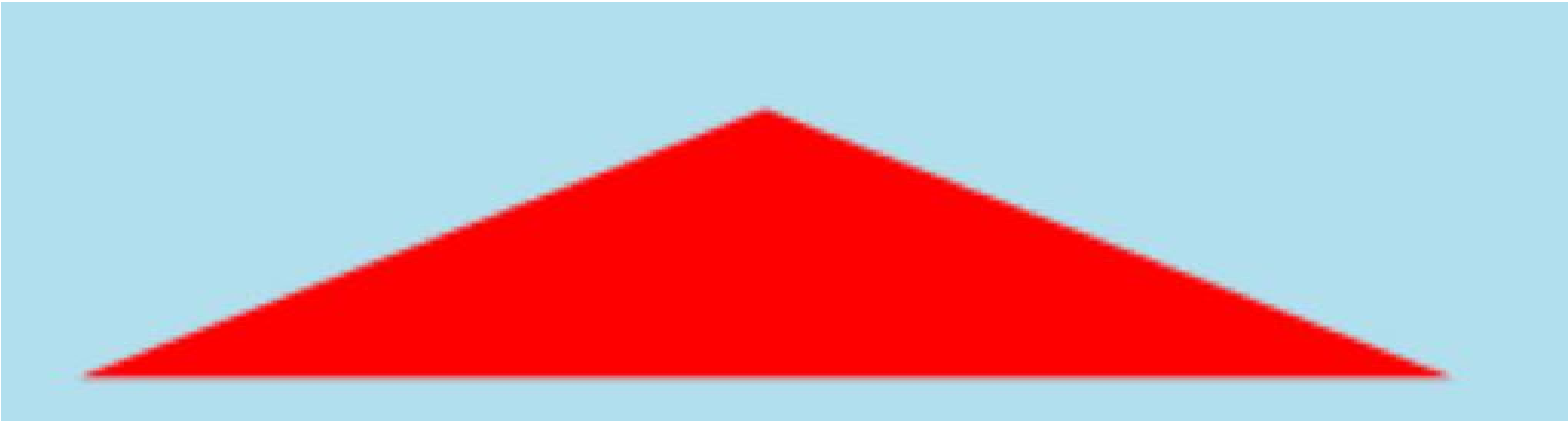
낭만스러운 집 만들기

- 집은 겉보기에 지붕과 외벽으로만 이루어져 있다.
- 컴포넌트로 표현하면 지붕 컴포넌트와 외벽 컴포넌트를 조합하면 집 컴포넌트가 완성된다.



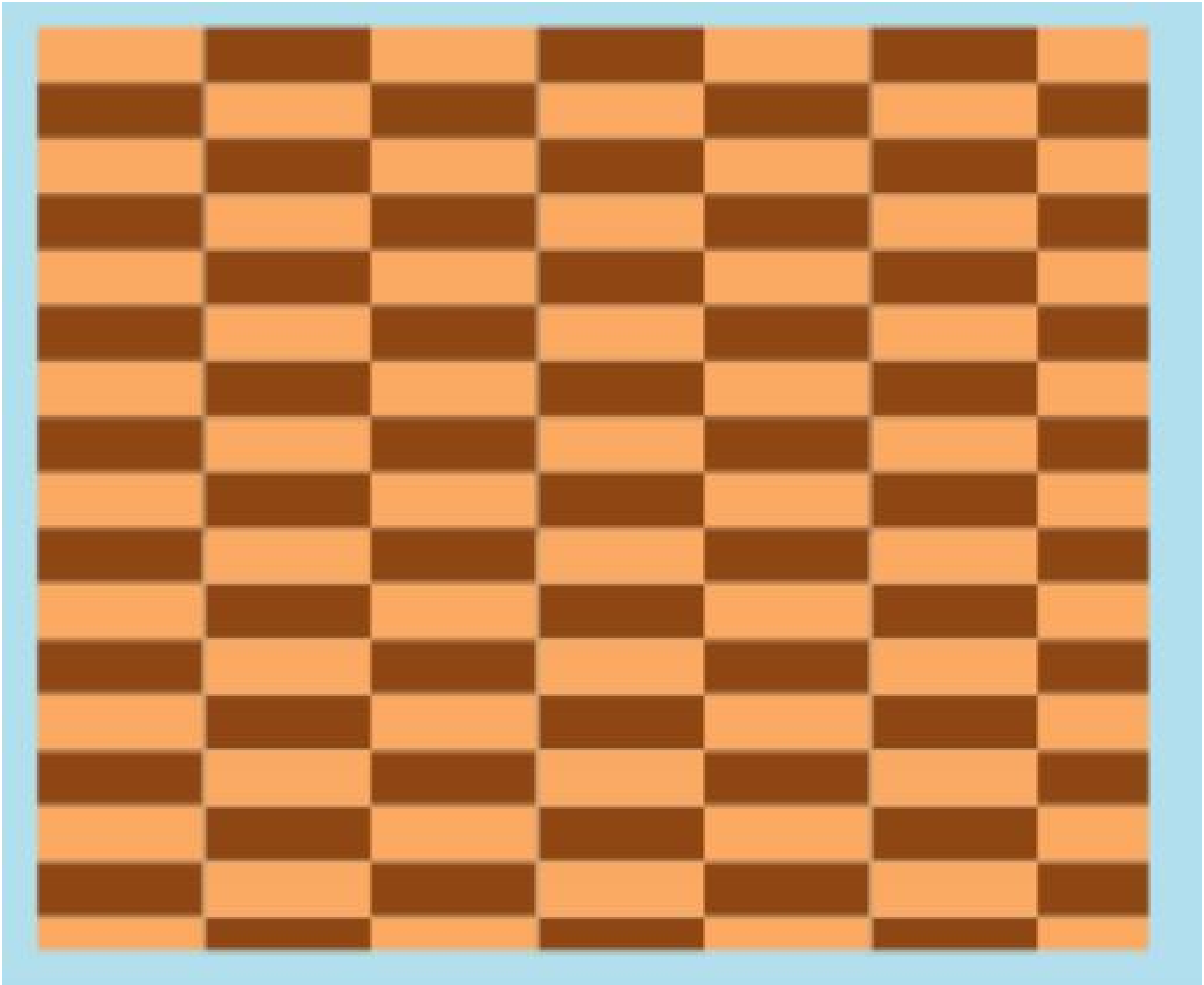
지붕

```
 void RoofComponent::paint (Graphics& g) override
{
    g.setColour (Colours::red);                    // 짓는 김에 도색까지
    Path roof;
    roof.addTriangle (0,                getHeight(),
                      getWidth(),       getHeight(),
                      getWidth() / 2, 0);
    g.fillPath (roof);
}
```

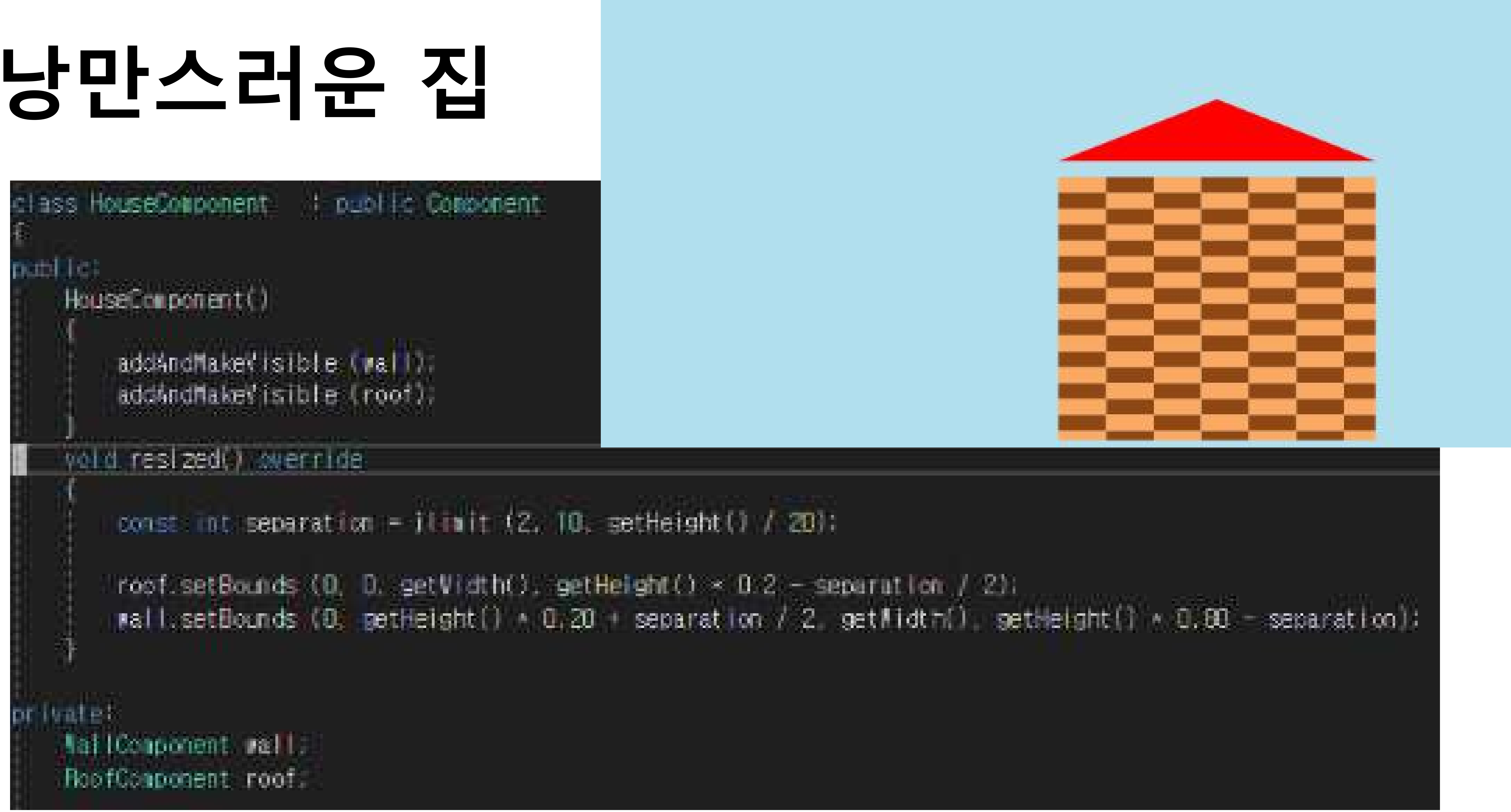


외벽

```
void WallComponent::paint (Graphics& g) override
{
    g.fillCheckerBoard (getLocalBounds().toFloat(), 30, 10,
                                    Colours::sandybrown, Colours::saddlebrown);
}
```



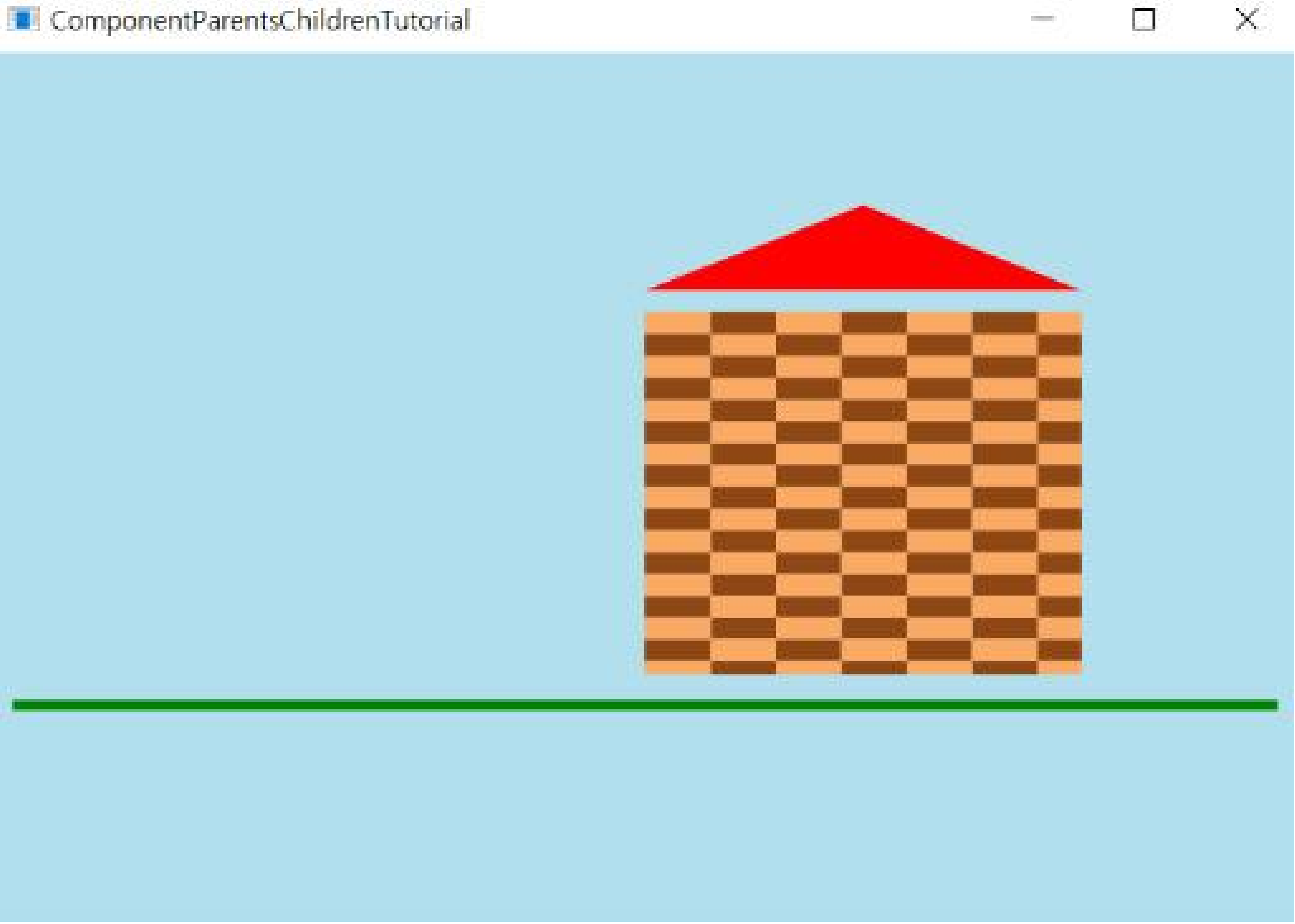
낭만스러운 집



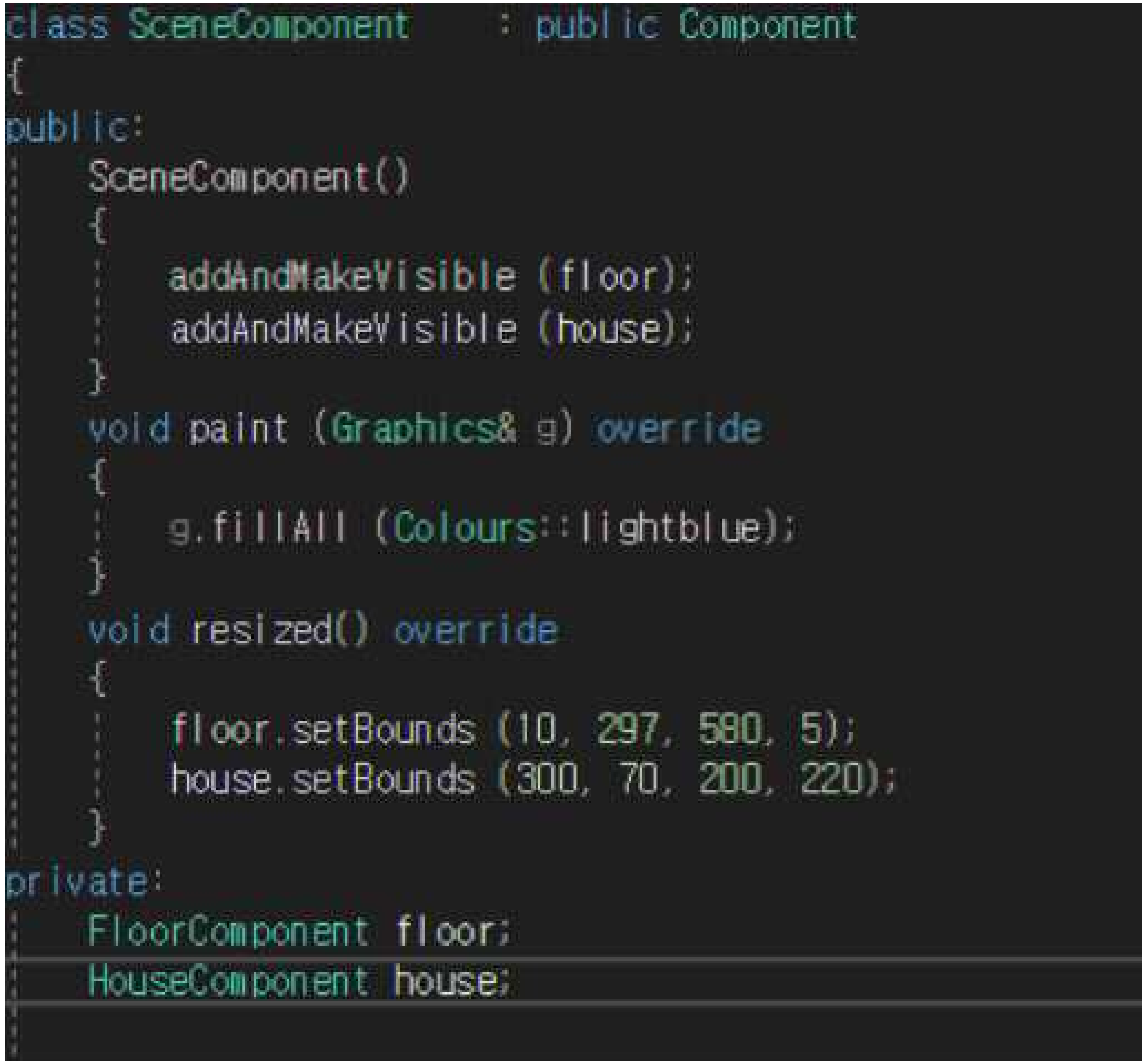
```
class HouseComponent   : public Component
{
public:
    HouseComponent()
    {
        addAndMakeVisible (wall);
        addAndMakeVisible (roof);
    }
    void resized() override
    {
        const int separation = jlimit (2, 10, getHeight() / 20);

        roof.setBounds (0, 0, getWidth(), getHeight() * 0.2 - separation / 2);
        wall.setBounds (0, getHeight() * 0.20 + separation / 2, getWidth(), getHeight() * 0.80 - separation);
    }

private:
    WallComponent wall;
    RoofComponent roof;
```

바닥

```
void FloorComponent::paint (Graphics& g) override
{
    g.setColour (Colours::green);
    g.drawLine (0, getHeight() / 2, getWidth(), getHeight() / 2, 5);
}
```

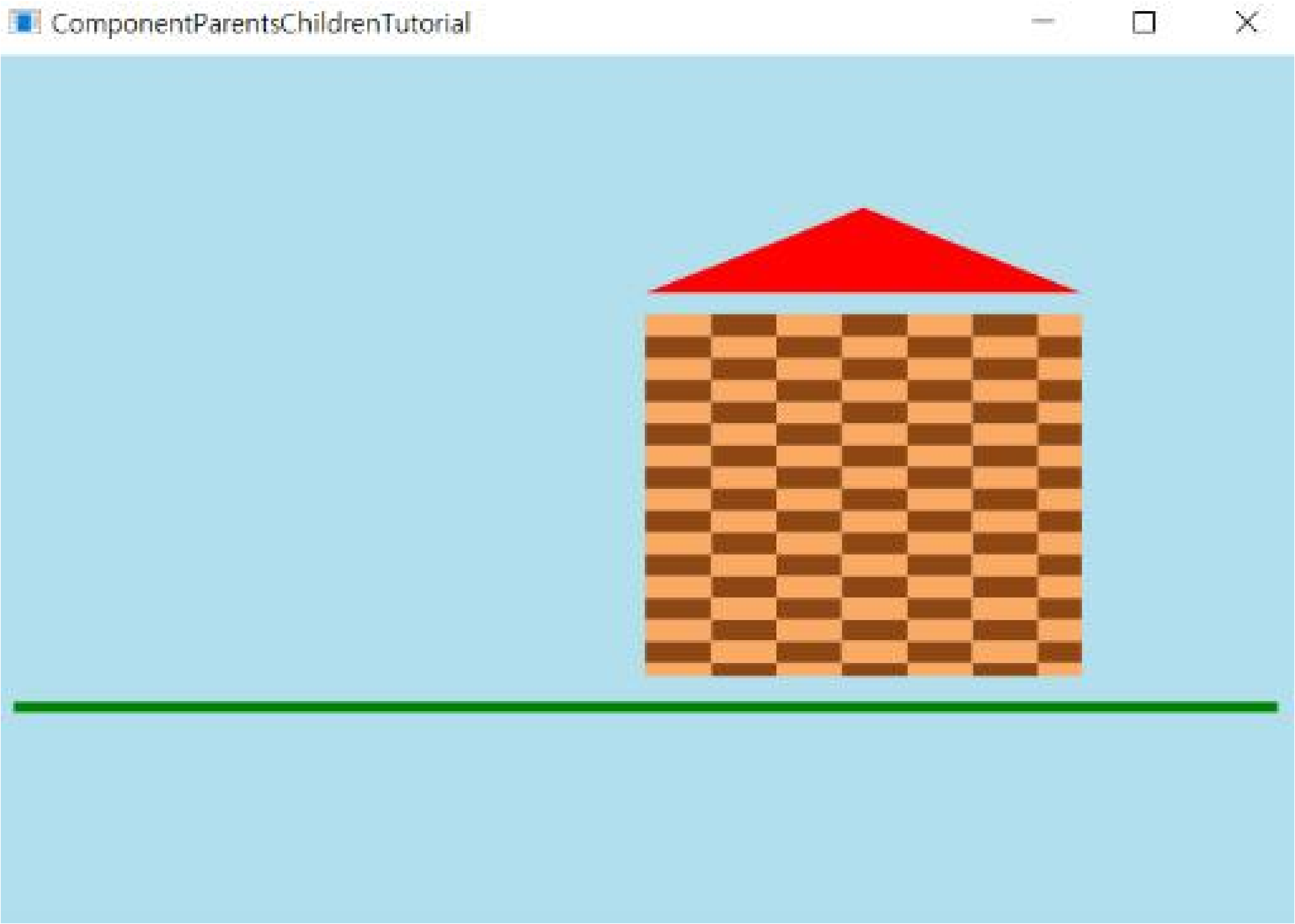


집과 바닥을 하나의 Scene으로



```
class SceneComponent    : public Component
{
public:
    SceneComponent()
    {
        addAndMakeVisible (floor);
        addAndMakeVisible (house);
    }
    void paint (Graphics& g) override
    {
        g.fillAll (Colours::lightblue);
    }
    void resized() override
    {
        floor.setBounds (10, 297, 580, 5);
        house.setBounds (300, 70, 200, 220);
    }
private:
    FloorComponent floor;
    HouseComponent house;
```

addAndMakeVisible():
자식 컴포넌트를 가시화할 필요가 있다. JUCE에서는 일반적으로 addAndMakeVisible을 사용하여 처리



지나가던 요정이...

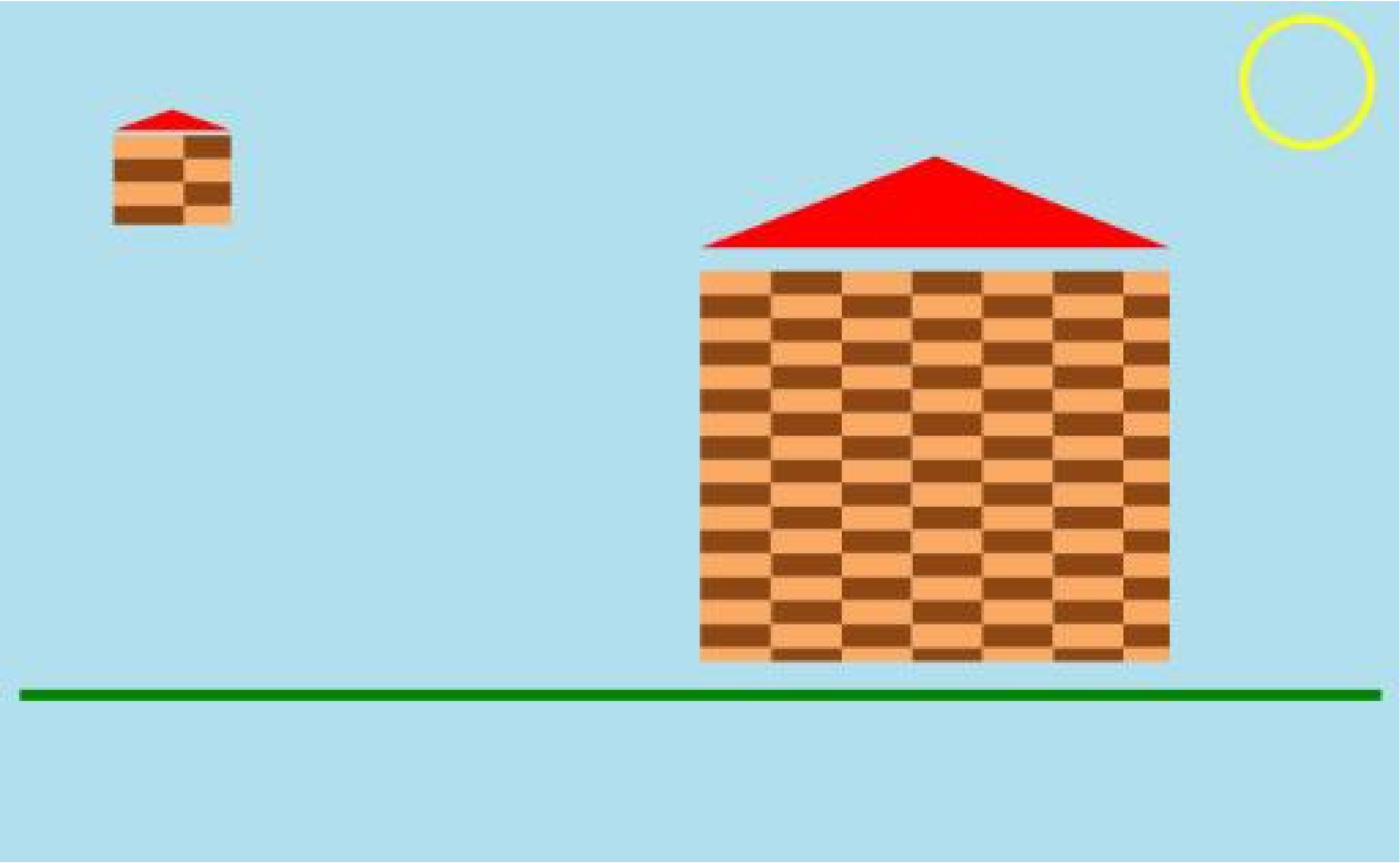
나도

똑같은 집. . .

요정을 위해

```
SceneComponent::SceneComponent()
   {
      addAndMakeVisible (smallHouse);
   }
void SceneComponent::resized() override
   {
      // ...
      smallHouse.setBounds (50, 50, 50, 50); // 요정 집
   }
```

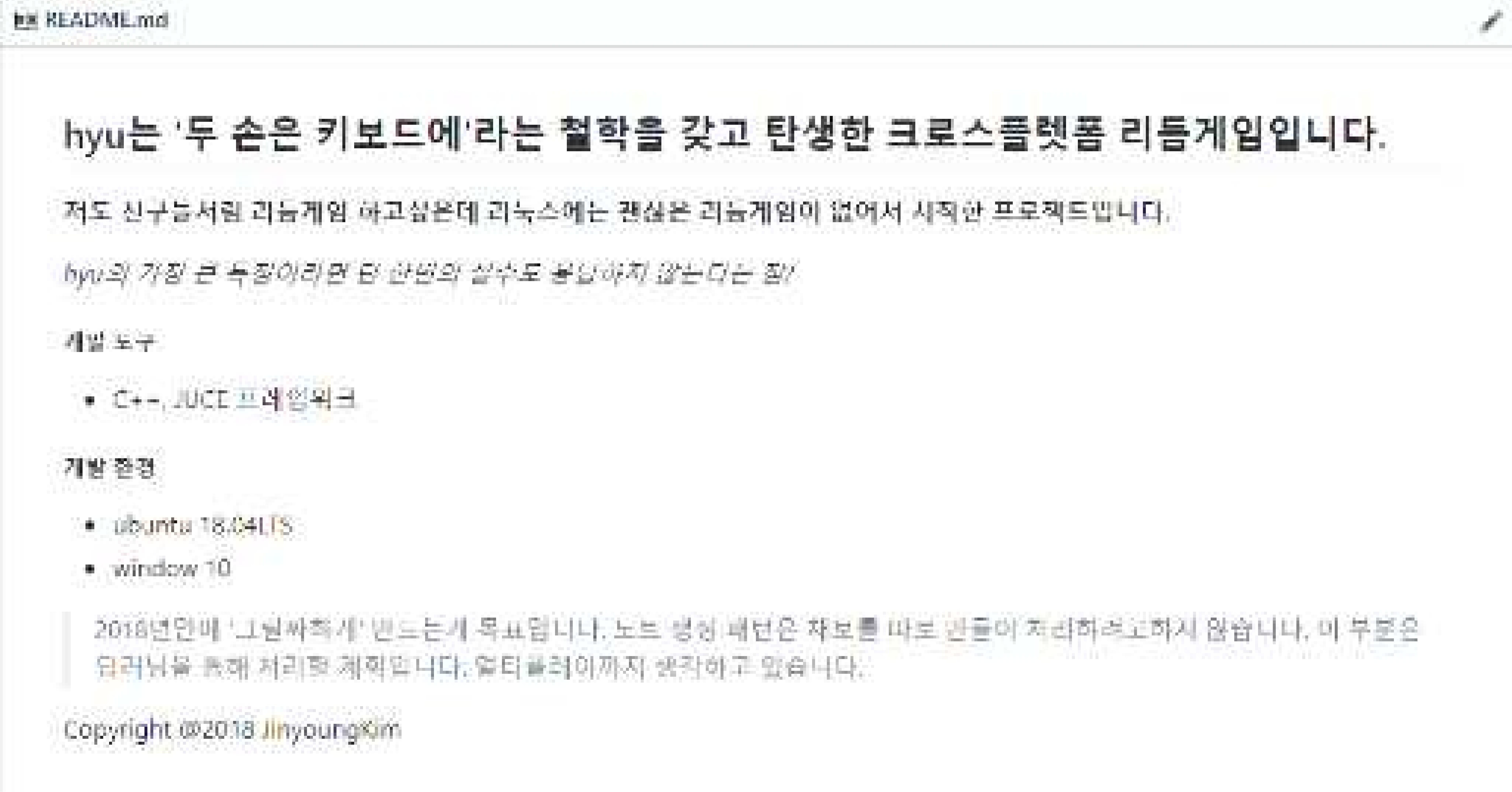
우왕...



hyu에 사용한 API 살펴보기

https://github.com/CafeM0ca/hyu

hyu???



hyu는 '두 손은 키보드에'라는 철학을 갖고 탄생한 크로스플렛폼 리듬게임입니다.

저도 친구들처럼 리듬게임 하고싶은데 리눅스에는 괜찮은 리듬게임이 없어서 시작한 프로젝트입니다.

hyu의 가장 큰 특징이라면 단 한번의 실수도 용납하지 않는다는 점!

개발 도구

- C++, JUCE 프레임워크

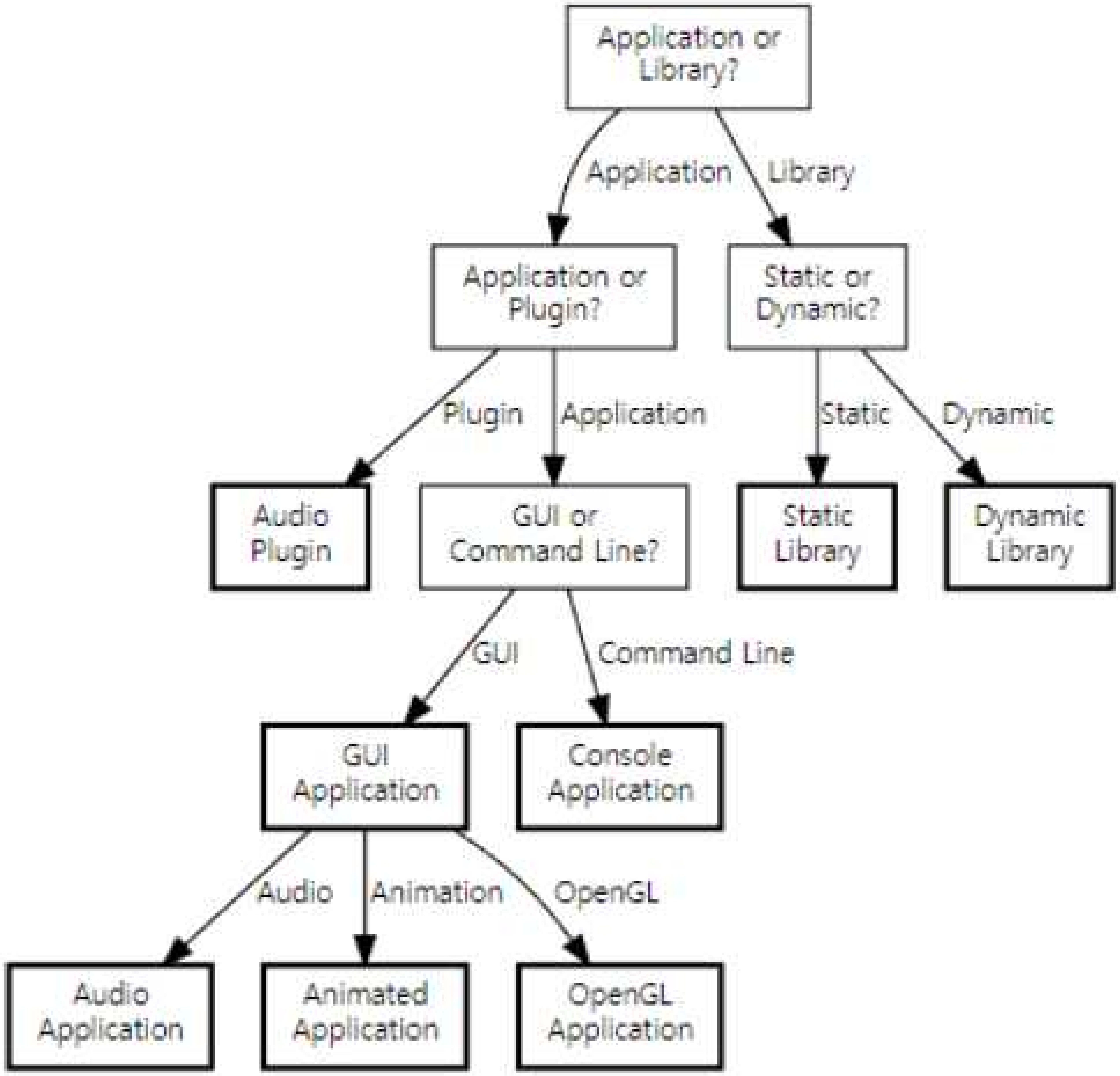
개발 환경

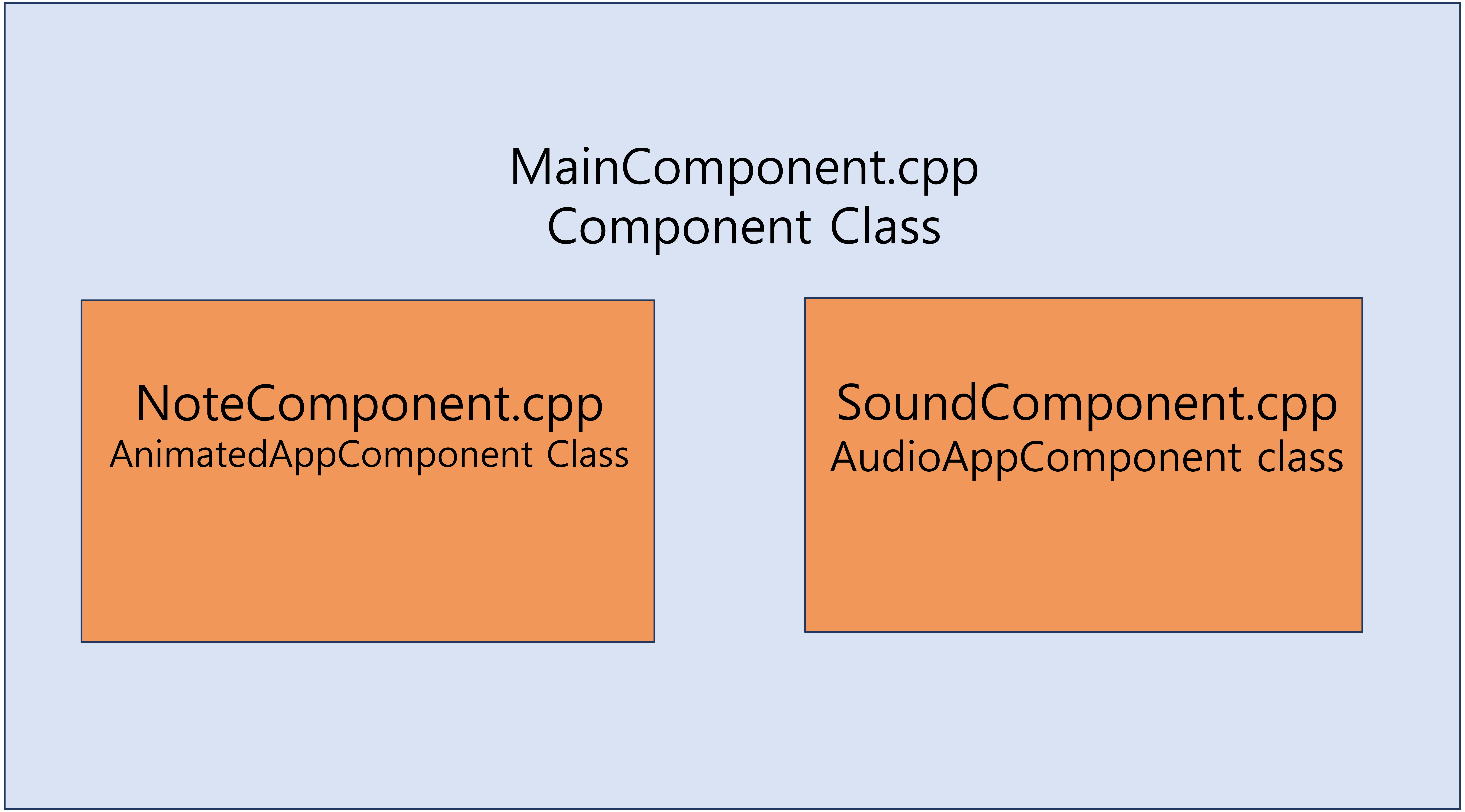
- ubuntu 18.04LTS
- window 10

> 2018년안에 '그럴싸하게' 만드는게 목표입니다. 노트 생성 패턴은 채보를 따로 만들어 처리하려고하지 않습니다. 이 부분은 딥러닝을 통해 처리할 계획입니다. 멀티플레이까지 생각하고 있습니다.

Copyright @2018 JinyoungKim

- ***GUI Application***
- ***Animated Application***
- OpenGL Applictaion
- Console Application
- ***Audio Applictaion***
- Audio Plug-In
- Static Library
- Dynamic Labrary





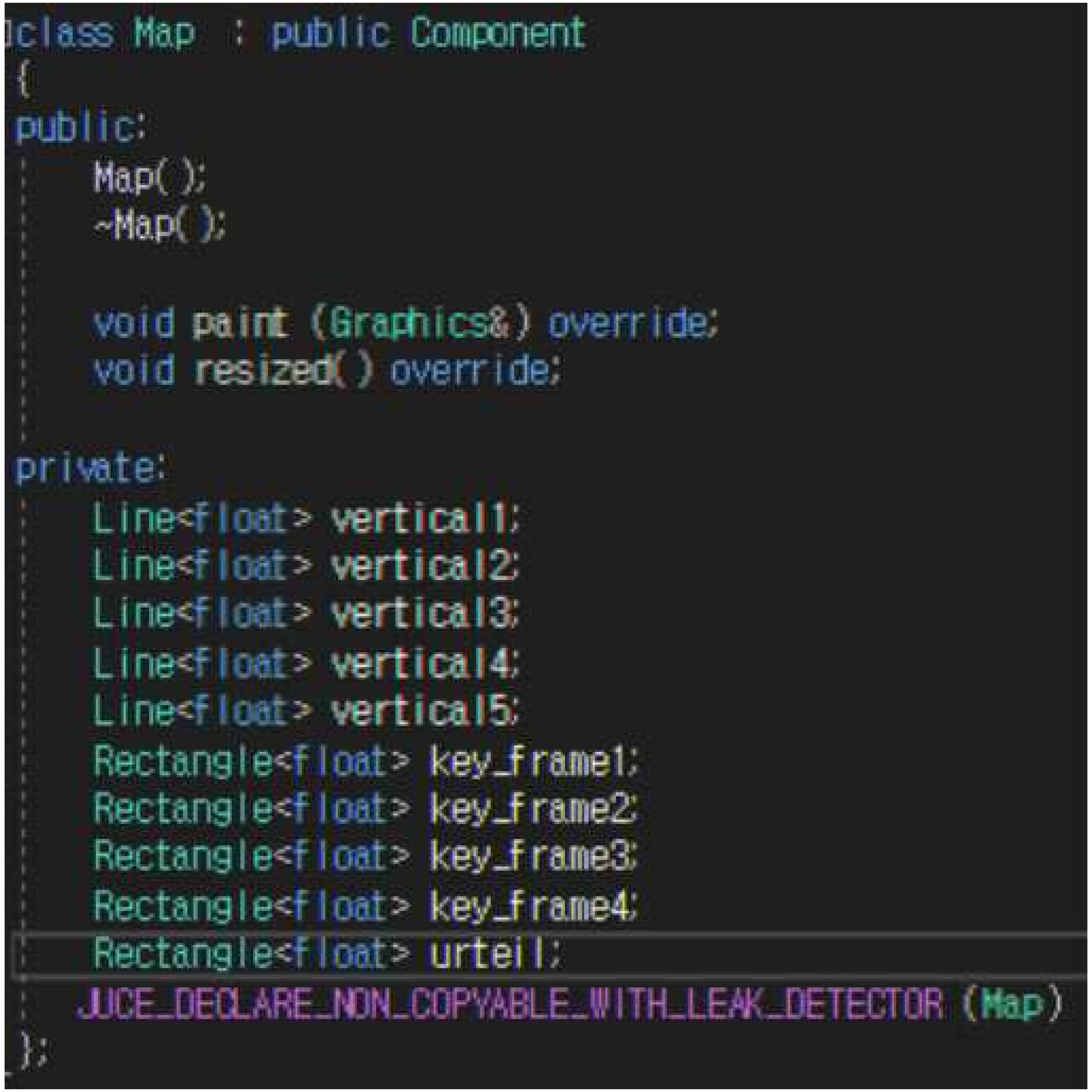
MainComponent.cpp
Component Class

NoteComponent.cpp
AnimatedAppComponent Class

SoundComponent.cpp
AudioAppComponent class

맵 구현

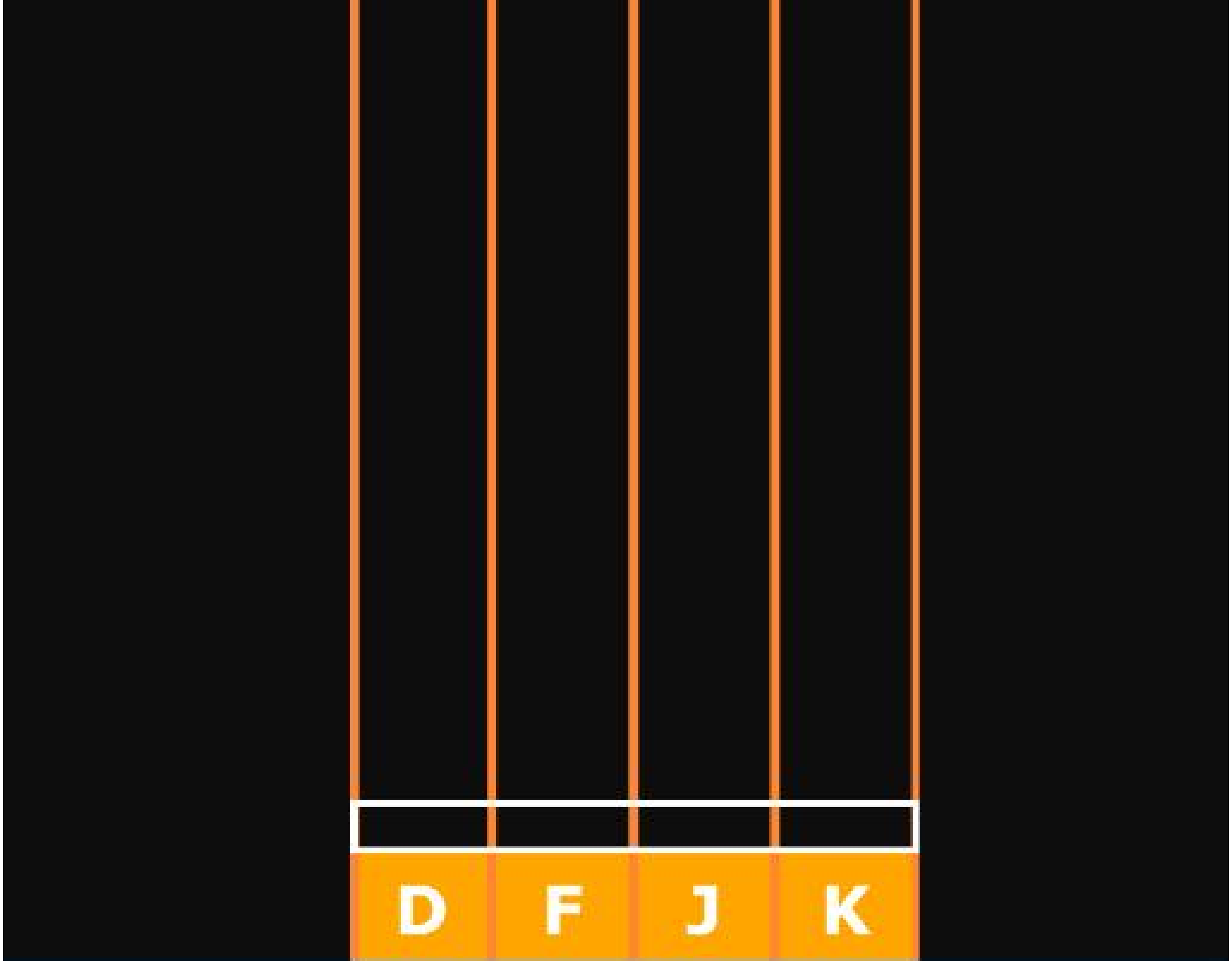
```
void Map::paint (Graphics& g)
{

  // D F J K 부분 덮기
  g.setColour(Colours::orange);
  g.fillRect(key_frame1);
  g.fillRect(key_frame2);
  g.fillRect(key_frame3);
  g.fillRect(key_frame4);

}
```



```
class Map  : public Component
{
public:
    Map();
    ~Map();

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

private:
    Line<float> vertical1;
    Line<float> vertical2;
    Line<float> vertical3;
    Line<float> vertical4;
    Line<float> vertical5;
    Rectangle<float> key_frame1;
    Rectangle<float> key_frame2;
    Rectangle<float> key_frame3;
    Rectangle<float> key_frame4;
    Rectangle<float> urteil;
   JUCE_DECLARE_NON_COPYABLE_WITH_LEAK_DETECTOR (Map)
};
```

```
void Map::paint (Graphics& g)
{
  // ...
  // Map 골격
  g.setColour(Colour(255,133,51));
  g.drawLine(vertical1, 7.0f);
  g.drawLine(vertical2, 7.0f);
  g.drawLine(vertical3, 7.0);
  g.drawLine(vertical4, 7.0f);
  g.drawLine(vertical5, 7.0f);
}
```

```
void Map::paint (Graphics& g)
{
  // judgement rectangle
  g.setColour(Colours::white);
  g.drawRect(urteil, 5.0f);

  // text in rectangle
  g.setFont(Font(60.0f,Font::bold));
  g.setColour(Colours::white);
  g.drawText("D", key_frame1,Justification::centred, true);
  g.drawText("F", key_frame2,Justification::centred, true);
  g.drawText("J", key_frame3,Justification::centred, true);
  g.drawText("K", key_frame4,Justification::centred, true);
}
```


D
F
J
K

노트 애니메이션 구현1

노트 애니메이션 부자연스럽

CafeM0ca committed on 6 May

첫번째 접근 방법

- Note Class는 애니메이션을 담당
- NoteController Class는 생성과 판정, 속도 담당
- 두 클래스 모두 Component를 상속. NoteController는 Timer 상속

=> 당시에는 AnimatedAppComponent를 사용할 생각을 못하였다.
결국 두번째 방법으로

노트 애니메이션 구현2

두번째 접근 방법

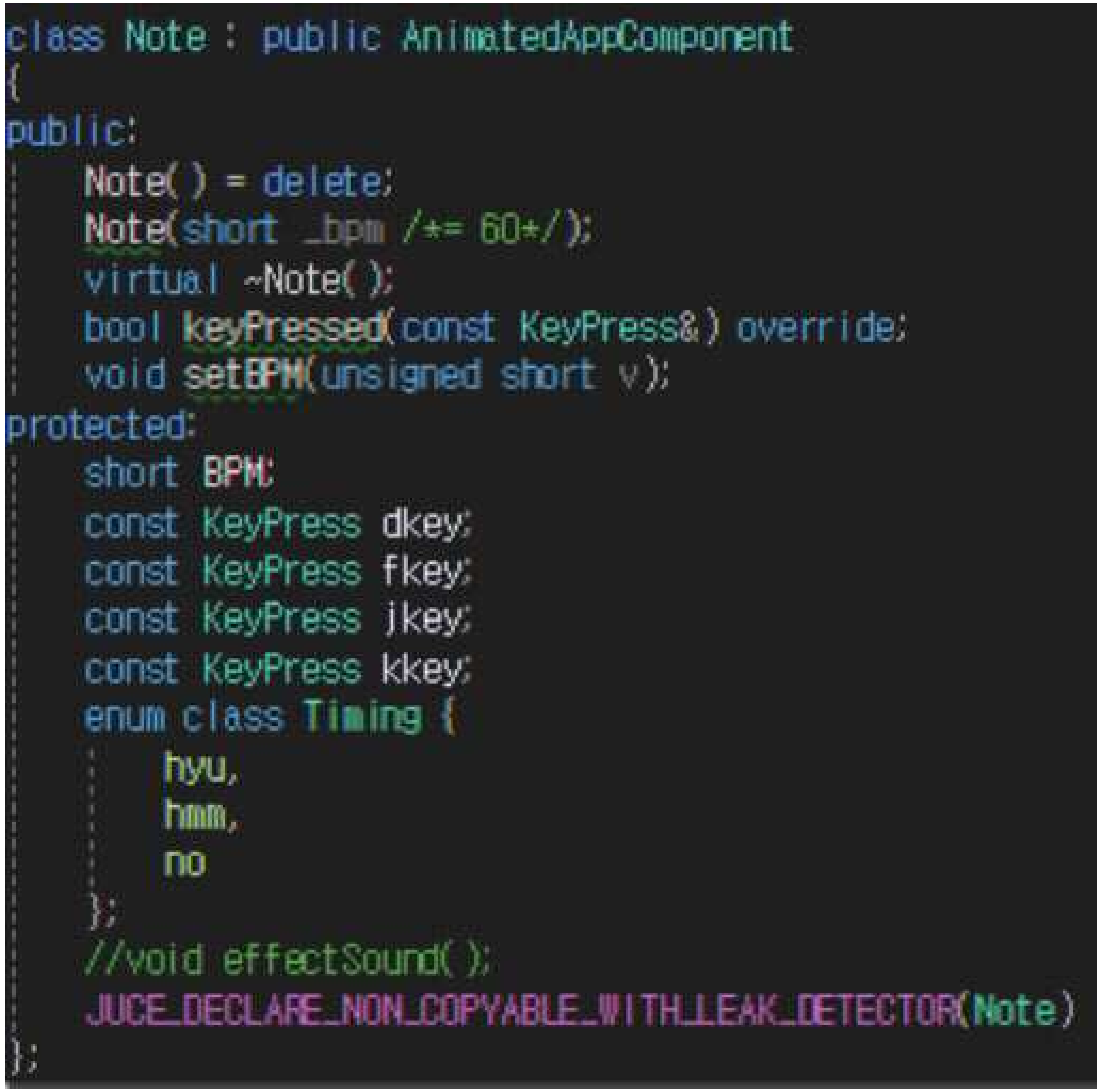
- Note Class가 AnimatedAppComponent를 상속
- 생성, 애니메이션, 속도, 판정 담당

-> 확장성 및 인터페이스 BAD..

결과 =>

전부 갈아 엎어야한다j

CafeM0ca committed on 3 Jul



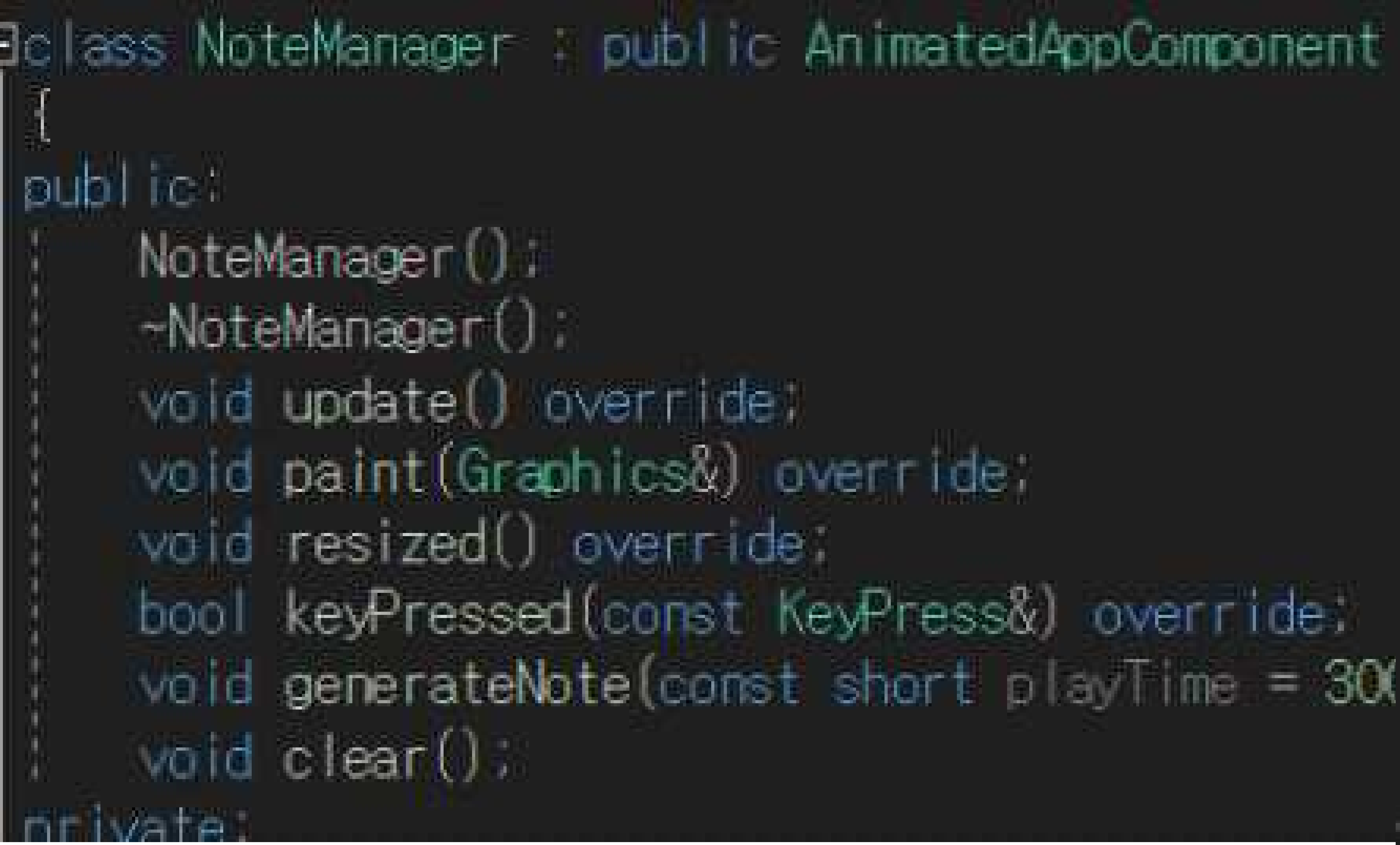
```
class Note : public AnimatedAppComponent
{
public:
    Note( ) = delete;
    Note(short _bpm /*= 60*/);
    virtual ~Note( );
    bool keyPressed(const KeyPress&) override;
    void setBPM(unsigned short v);
protected:
    short BPM;
    const KeyPress dkey;
    const KeyPress fkey;
    const KeyPress jkey;
    const KeyPress kkey;
    enum class Timing {
        hyu,
        hmm,
        no
    };
    //void effectSound( );
    JUCE_DECLARE_NON_COPYABLE_WITH_LEAK_DETECTOR(Note)
};
```

노트 애니메이션 구현3

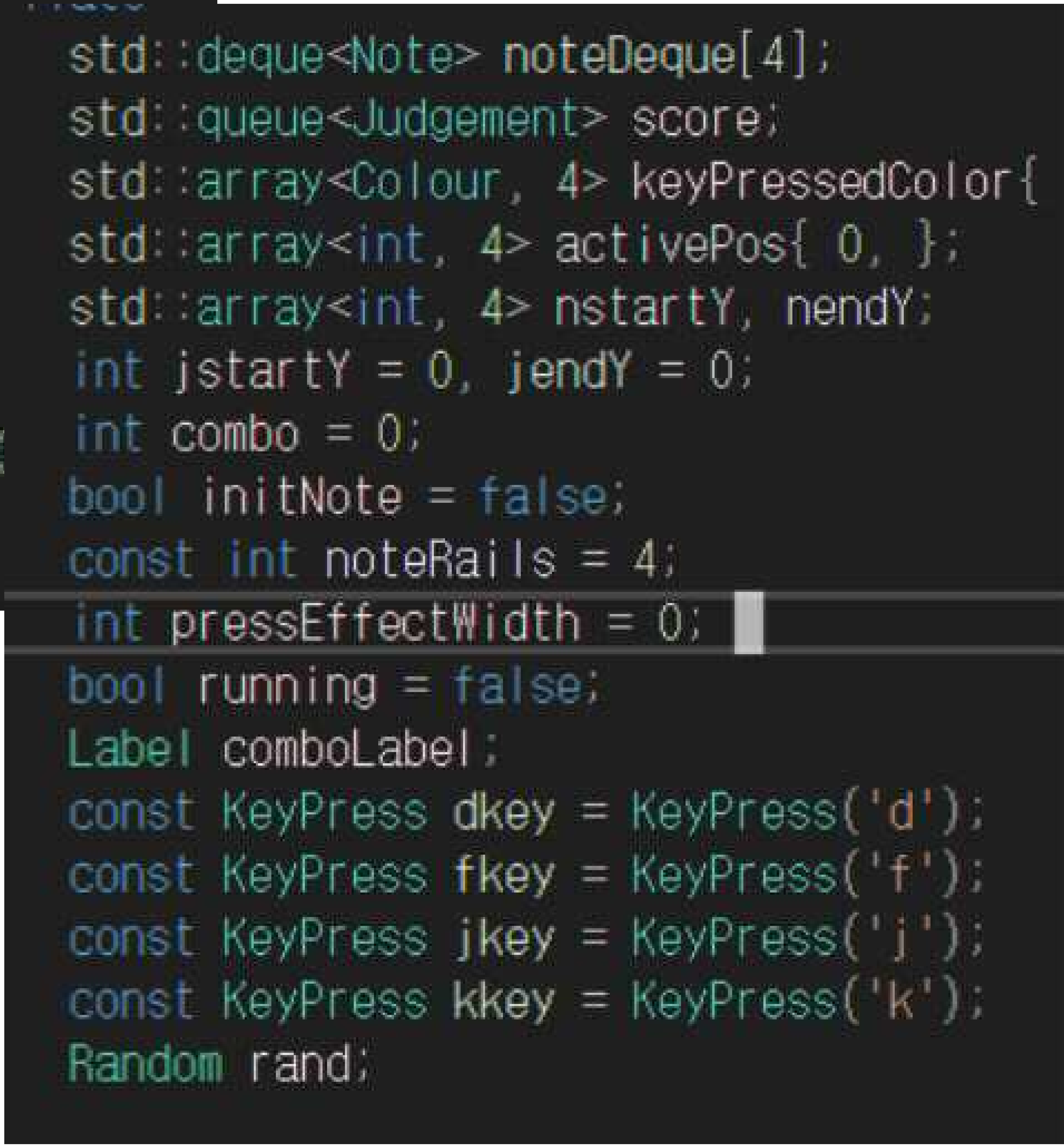
세번째 접근 방법

- 첫번째 접근법 + 두번째 접근법
- Note 데이터를 관리하는 노트 매니져 클래스

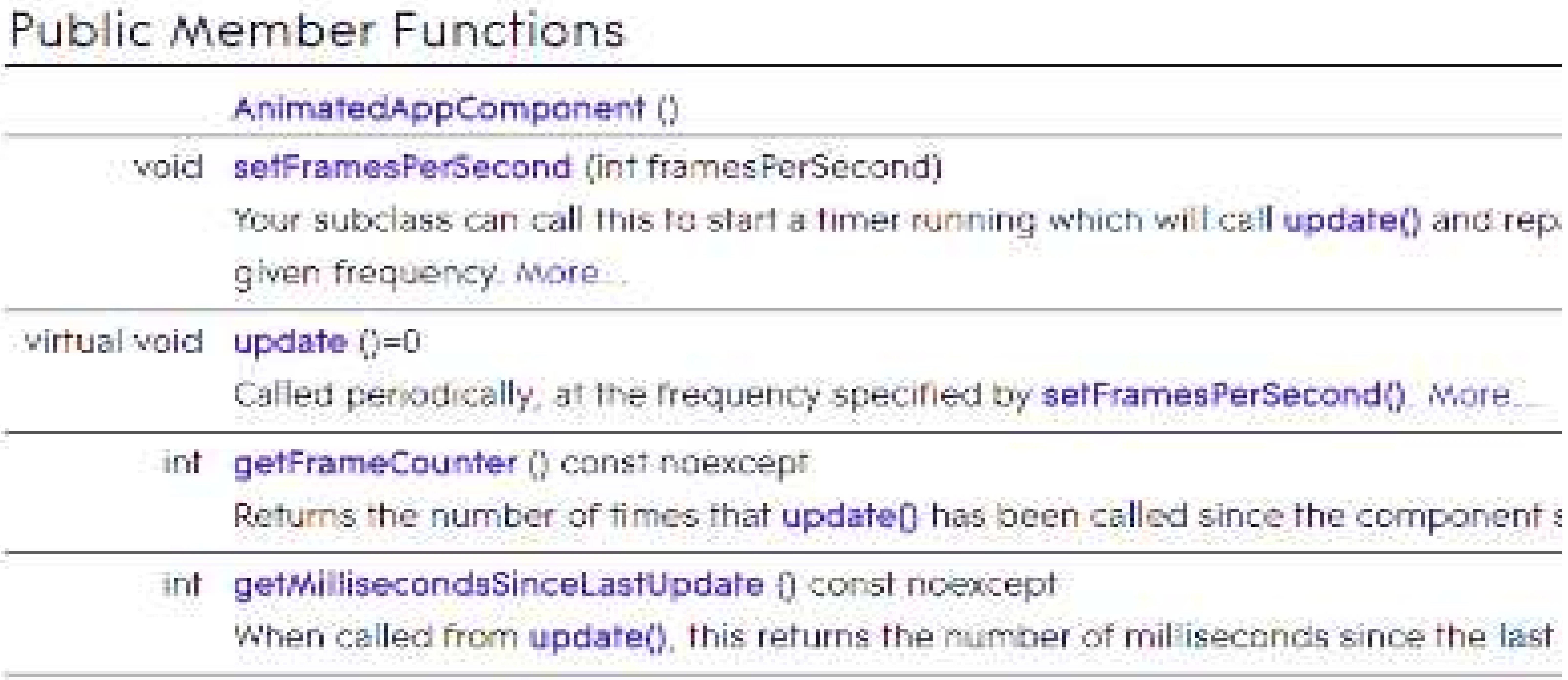
결과 => 확장성 및 인터페이스 매우 좋아짐



```
class NoteManager : public AnimatedAppComponent
{
public:
    NoteManager();
    ~NoteManager();
    void update() override;
    void paint(Graphics&) override;
    void resized() override;
    bool keyPressed(const KeyPress&) override;
    void generateNote(const short playTime = 30
    void clear();
private:
```

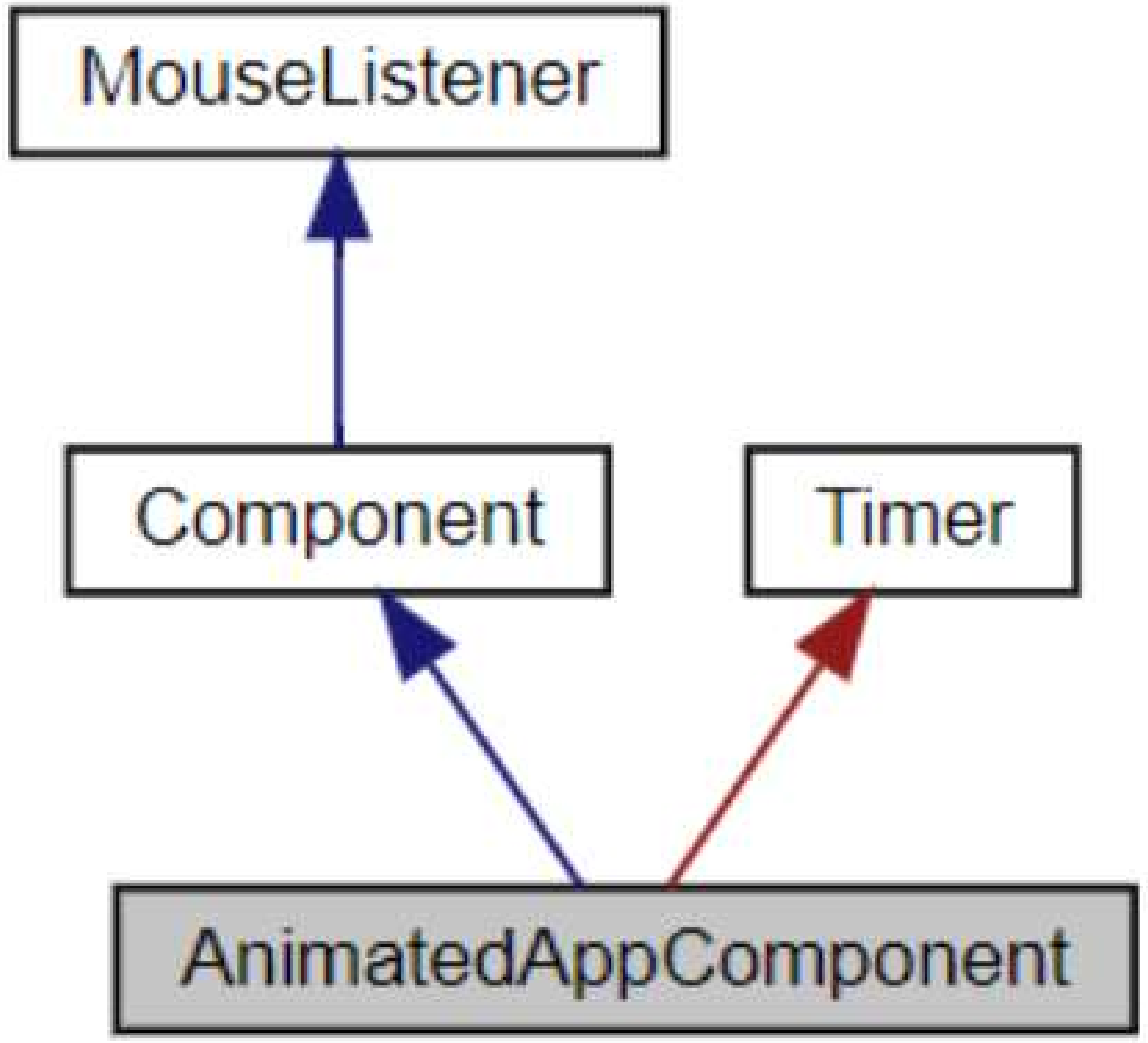


```
    std::deque<Note> noteDeque[4];
    std::queue<Judgement> score;
    std::array<Colour, 4> keyPressedColor{
    std::array<int, 4> activePos{ 0, };
    std::array<int, 4> nstartY, nendY;
    int jstartY = 0, jendY = 0;
    int combo = 0;
    bool initNote = false;
    const int noteRails = 4;
    int pressEffectWidth = 0;
    bool running = false;
    Label comboLabel;
    const KeyPress dkey = KeyPress('d');
    const KeyPress fkey = KeyPress('f');
    const KeyPress jkey = KeyPress('j');
    const KeyPress kkey = KeyPress('k');
    Random rand;
```



Public Member Functions

	AnimatedAppComponent ()
void	setFramesPerSecond (int framesPerSecond) Your subclass can call this to start a timer running which will call update() and rep given frequency. More...
virtual void	update ()=0 Called periodically, at the frequency specified by setFramesPerSecond(). More...
int	getFrameCounter () const noexcept Returns the number of times that update() has been called since the component s
int	getMillisecondsSinceLastUpdate () const noexcept When called from update(), this returns the number of milliseconds since the last



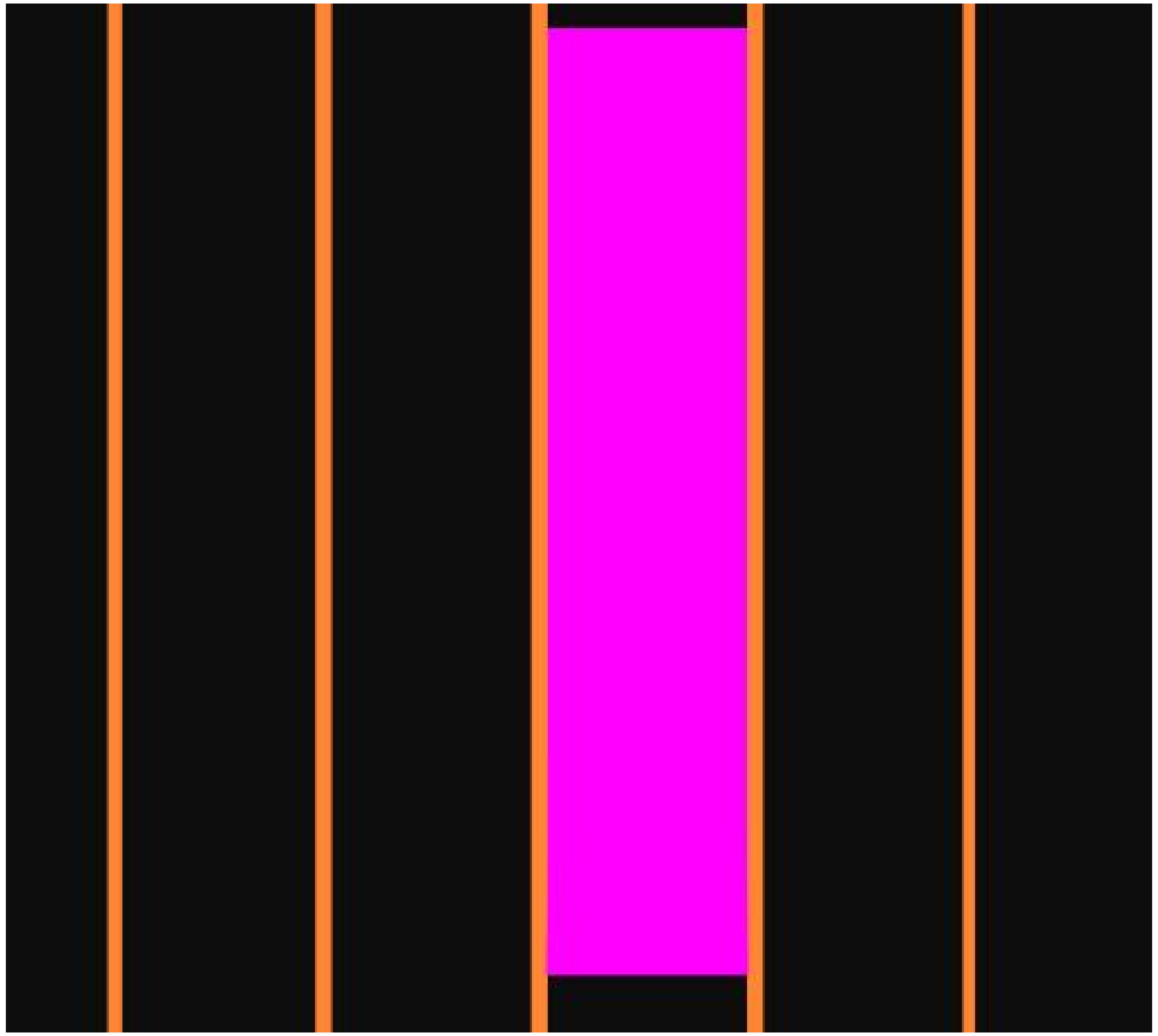
update()는 setFramesPerSecond()에 설정한 값만큼 1초에 n번 호출됨

deque<Note>에 활성화된 노트의 위치 조절과 수명이 끝난 노트 제거에 유용

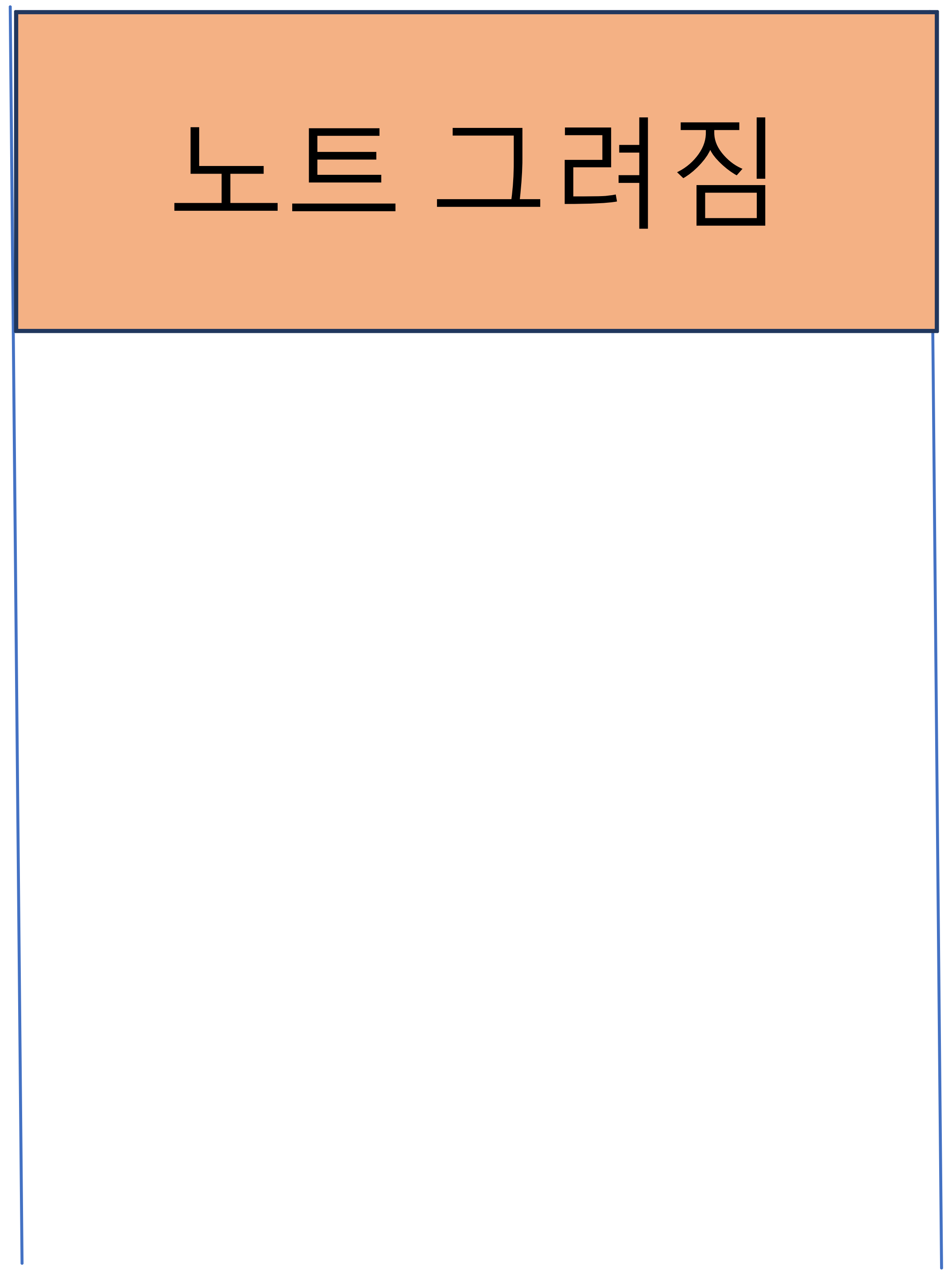
```
NoteManager::NoteManager()
{
  setFramesPerSecond(60);                      // update()를 1초에 60번 호출
  addAndMakeVisible(comboLabel);               // 콤보 Label이 화면에 보이게
  comboLabel.setJustificationType(Justification::centred);
  comboLabel.setFont(Font(50.0f));
}

void NoteManager::update()
{
  // 생성, 활성화된 노트, 수명이 끝난 노트 로직 처리
}
```

```
void NoteManager::paint(Graphics& g)
{
  // draw Note
  g.setColour(Colours::fuchsia);
  for (int i = 0; i < noteRails; i++)
  {
    // 노트 그려줌
  }
  // ...
}
```

노트가 가래떡처럼 길어지는 문제 발생

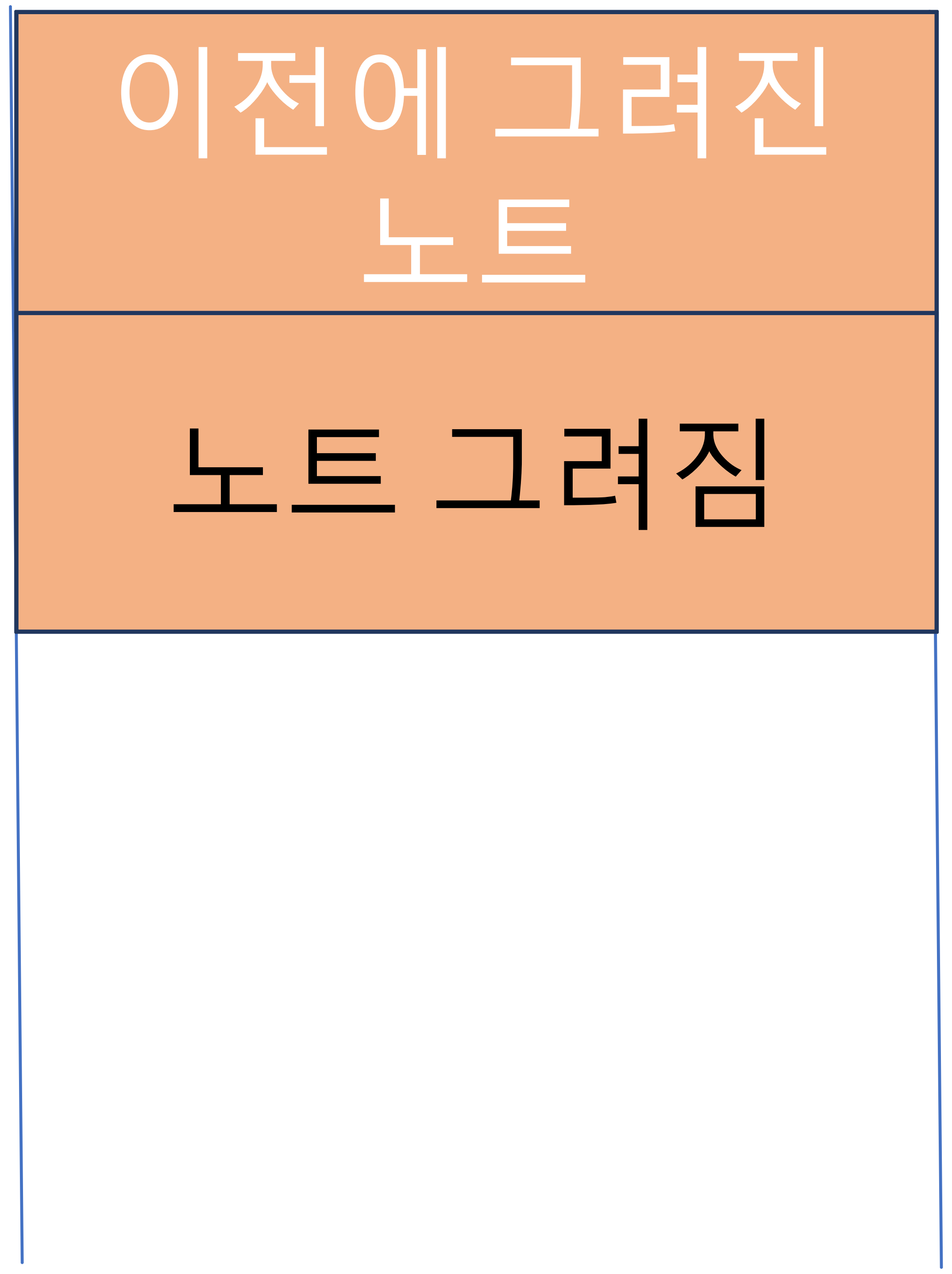


Why???



노트 그려짐

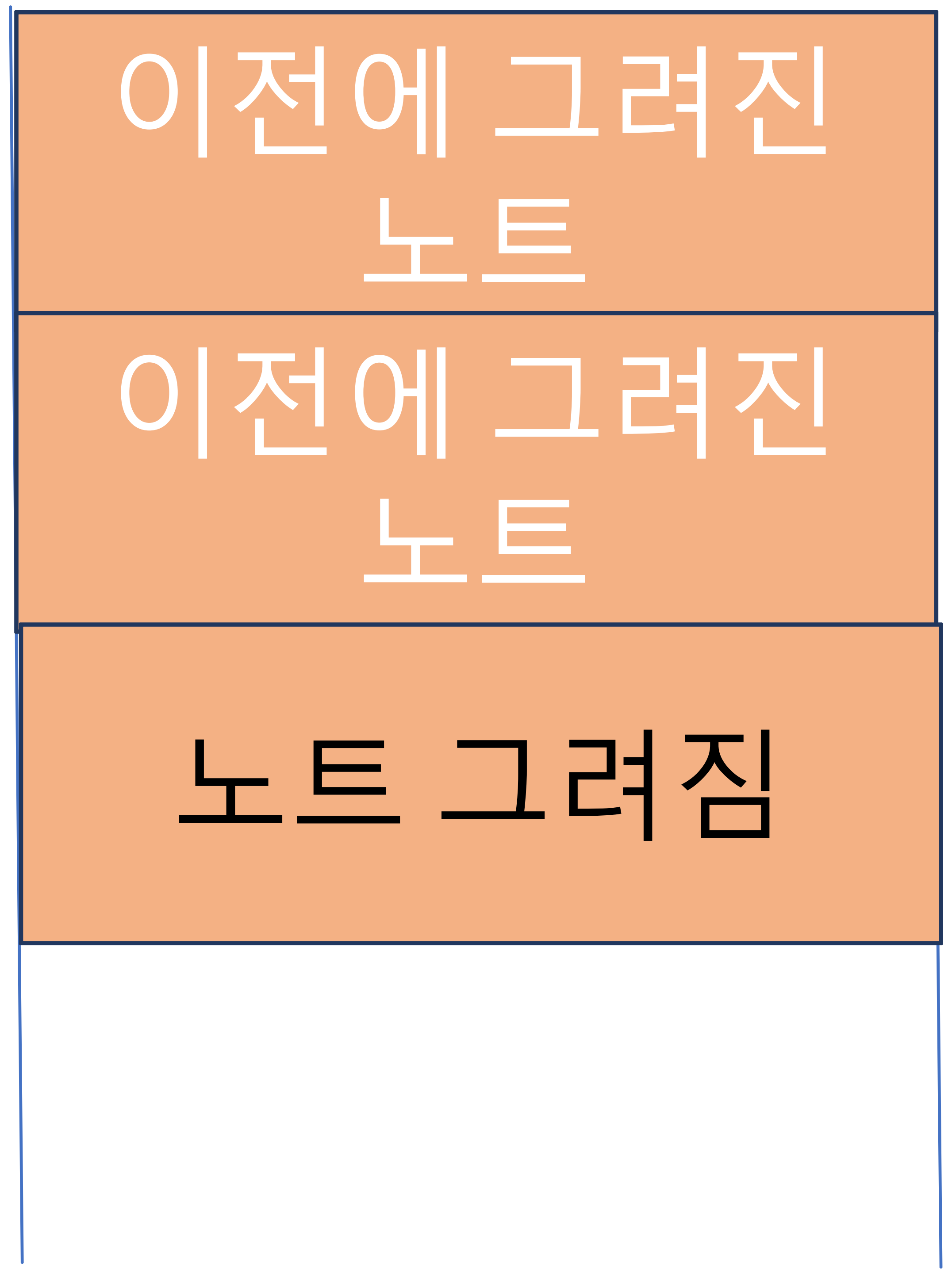
paint() 첫번째 호출



이전에 그려진 노트

노트 그려짐

paint() 두번째 호출



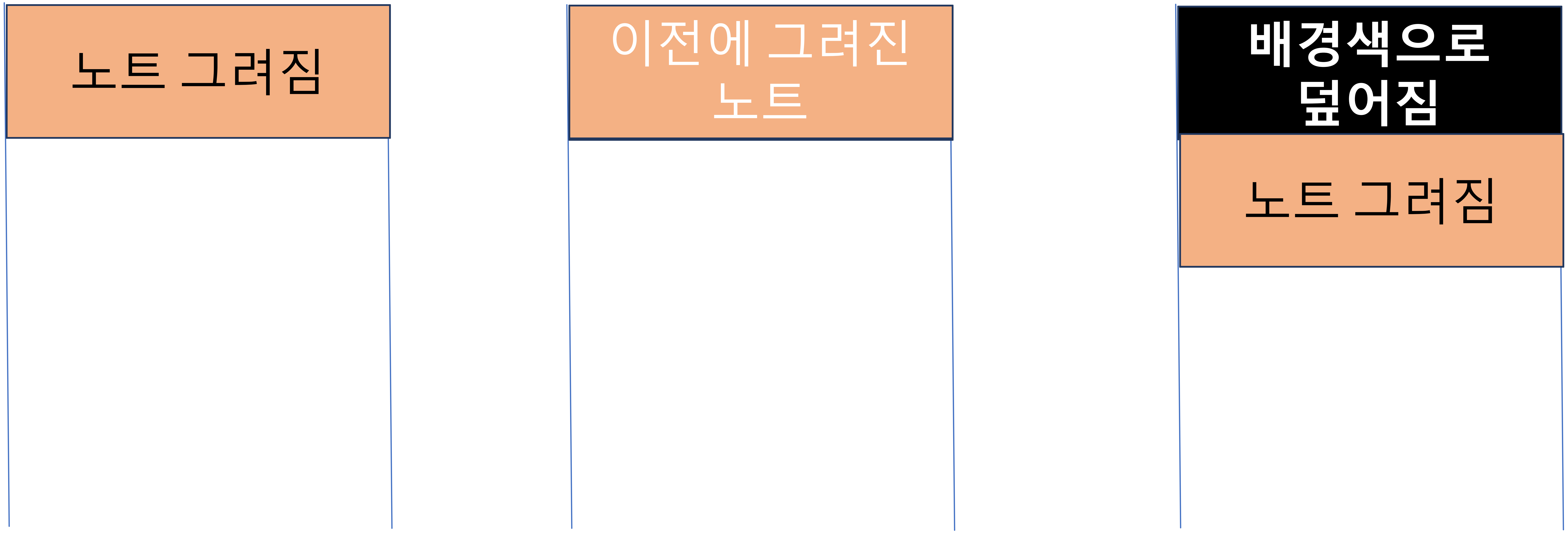
이전에 그려진 노트

이전에 그려진 노트

노트 그려짐

paint() 두번째 호출

Solution!!!



노트 그려짐

이전에 그려진 노트

배경색으로 덮어짐

노트 그려짐

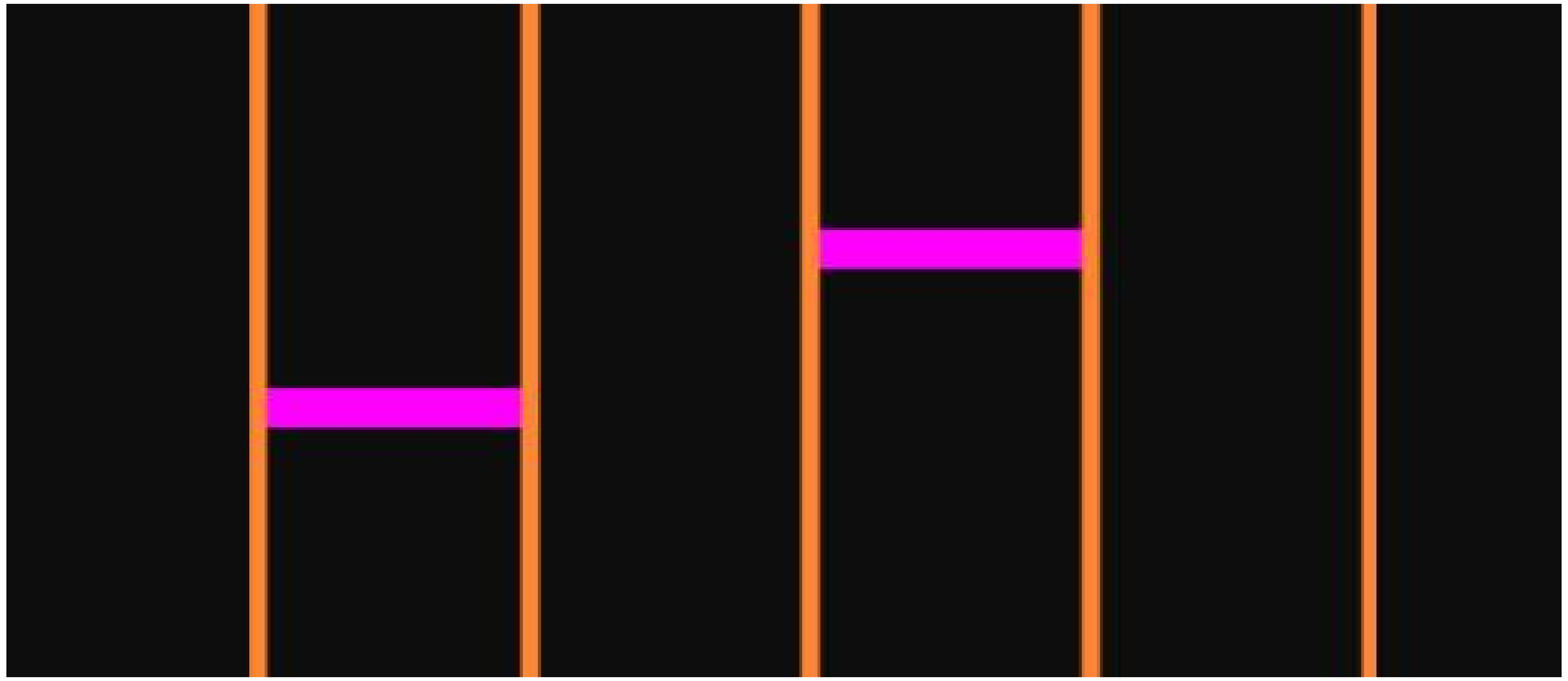
paint() 첫번째 호출

paint() 두번째 호출

paint() 두번째 호출

```
void NoteManager::paint(Graphics& g)
{
  // 기존에 그려진걸 다 덮는다
  g.fillAll(Colour(13, 13, 13));

  // draw Note
  g.setColour(Colours::fuchsia);
  for (int i = 0; i < noteRails; i++)
  {
    // 노트 그려줌
  }
  // ...
}
```

해결



입력 효과 구현

```
void NoteManager::paint(Graphics& g)
{
  // …
  // When key pressed, paint judgement line
  if (dkey.isCurrentlyDown()) {
     // 키가 눌렸으니 판정 처리
    g.setColour(keyPressedColor[0]);
    g.fillRect(Rectangle<float>(0, jstartY, pressEffectWidth, jendY));
  }
  // . . .
}
```

한 번 눌렀지만 paint() 가 여러번 호출되어
여러번 판정 처리

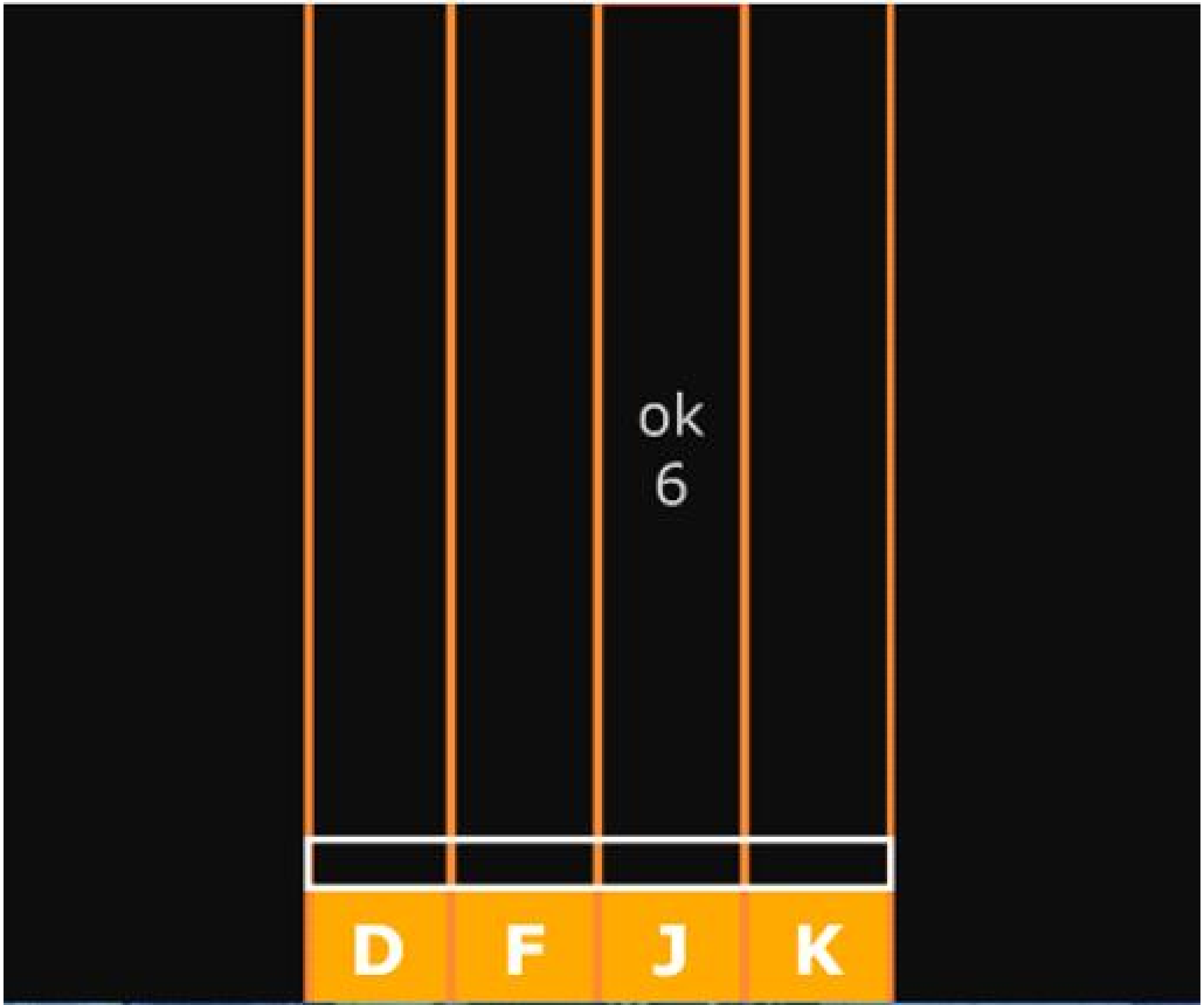
=> 뒤의 노트도 덩달아 판정처리

keyPressed()로 해결

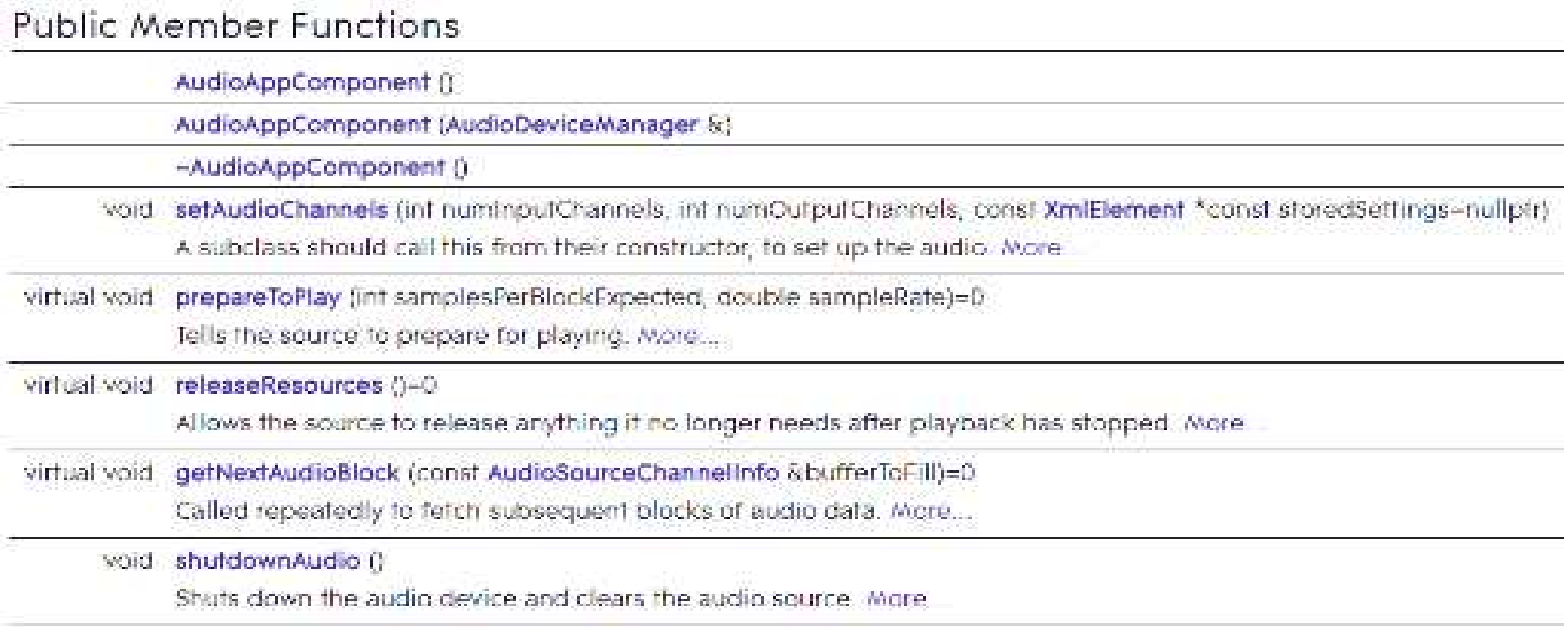
```
bool NoteManager::keyPressed(const KeyPress& key)
{
  if (const int index = [this, key]()->auto {
        if      (key == dkey) return 0;
        else if (key == fkey) return 1;
        else if (key == jkey) return 2;
        else if (key == kkey) return 3;
        else           return -1;
      }(); index >= 0 && index <= 3) {
    if(!noteDeque[index].empty() && nstartY[index] > getHeight() / 2)
      judgeNote(index, nstartY[index], nendY[index]);
    return true;
  } else return false;
}
```

키가 눌린 순간 호출.
맨 앞 노트 판정처리를 보장.

자매품으로 keyStateChanged(bool)
true: 눌린 순간
false: 땐 순간


ok
6
D
F
J
K

AudioAppComponent



Public Member Functions

	AudioAppComponent ()
	AudioAppComponent (AudioDeviceManager &)
	~AudioAppComponent ()
void	setAudioChannels (int numInputChannels, int numOutputChannels, const XmlElement *const storedSettings=nullptr) A subclass should call this from their constructor, to set up the audio. More...
virtual void	prepareToPlay (int samplesPerBlockExpected, double sampleRate)=0 Tells the source to prepare for playing. More...
virtual void	releaseResources ()=0 Allows the source to release anything it no longer needs after playback has stopped. More...
virtual void	getNextAudioBlock (const AudioSourceChannelInfo &bufferToFill)=0 Called repeatedly to fetch subsequent blocks of audio data. More...
void	shutdownAudio () Shuts down the audio device and clears the audio source. More...

prepareToPlay

getNextAudioBlock

releaseResource

Demo

Q & A?

감사합니다.

Reference

https://juce.com/

https://en.wikipedia.org/wiki/JUCE

http://cafemocamoca.tistory.com/category/Programming/JUCE