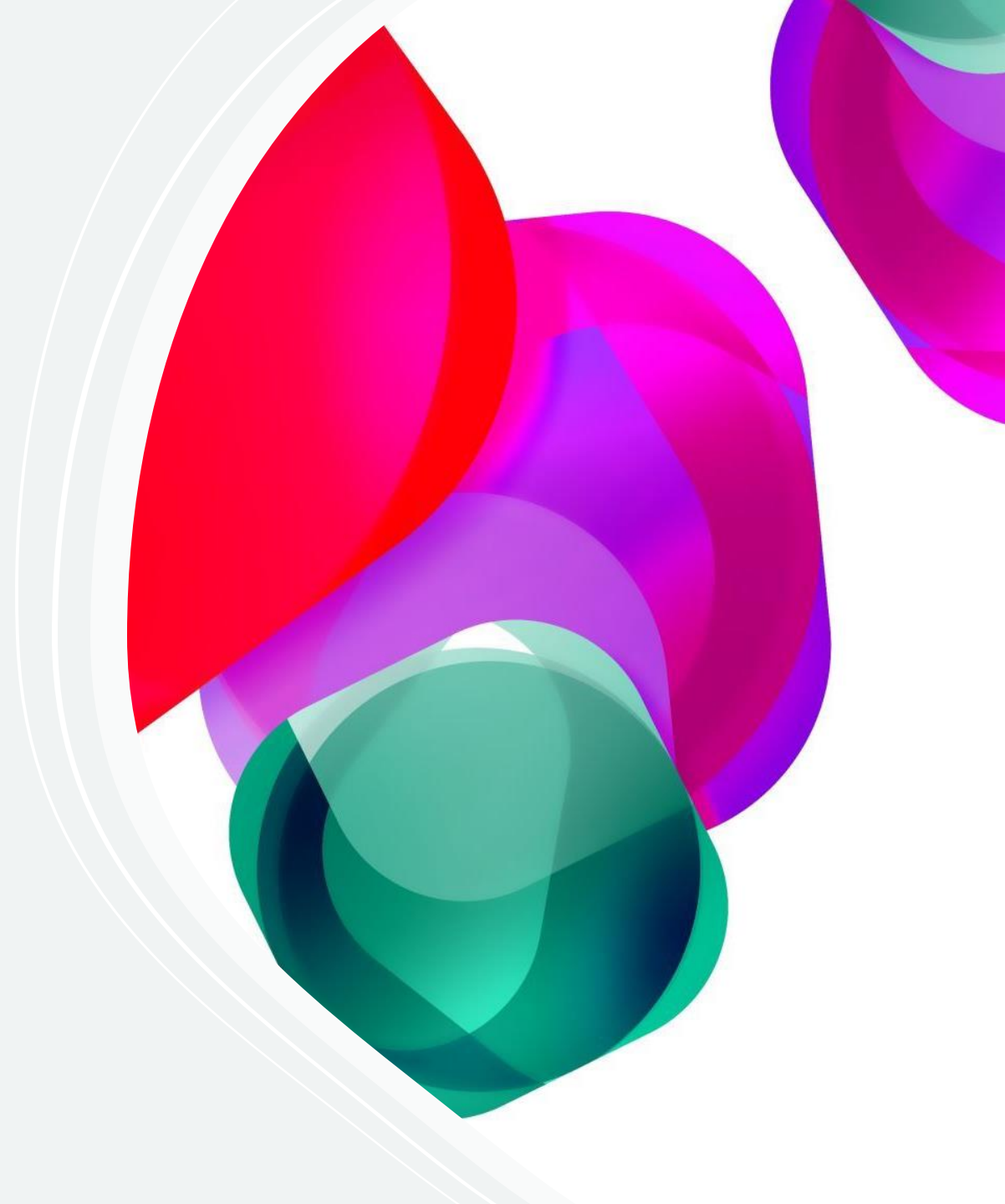


Final JUCE Presentation

By Stephen Davis



!EDIT ACCESS!

**For Prof.
Manzo's and
Davis' to
Edit Only
(Slides 1-12)**

For Student's Edit Access
(Create Any Number of
Slides After Slide 12)

Project Overview

Slides 1-10:

Title Slide

Edit Access

Overview

Intro to JUCE

Literature Review

Methodological Practices

Conclusion to JUCE

Recommended Action

Thanks

References

Key:

Anything **Bold** is an Important Section!

Slides Past 12:

Unknown for Now

(However, Please Update
This Defacto Table of
Contents in the Future)

Intro to JUCE

- JUCE was originally created in 2004, as one of the first ways for "user" plug-ins to be created.
- This was the beginning of a "producer's revolution", as producers now had the power to create their own DAW related-programs using a C++ framework to design any number of plug-ins of their design.
- JUCE is still being updated to this day (by its original co-founders Julian Storer (created the physical program) and Rufus Lewis (made it open-source to coders around the world through the creation of the ADC (audio developers conference series)).

Literature Review (See References)

- The main pieces of literature that were reviewed for the development of this capstone (and, its subsequent programs) were as follows:
 - Professor VJM's IQP on Plug-In Creation
 - JUCE Website Tutorials
 - JUCE Youtube Tutorials
 - C++ Youtube Tutorials
 - C++ StackOverflow Forum
 - Additional Resources:
 - Citation Machine
 - Google Scholar
 - Zotero

Methodologic Practices (See Excel Ratings)

Essential Steps:

- Learn Computer Programming:
 - C++ (the Program JUCE Runs In)
 - Learn JUCE's Basics:
 - Using JUCE's Provided Website Tutorials

Recommended Steps:

- Learn Computer Programming's Basics:
 - BASIC, Python, and JAVA are starter programs that you may want to consider learning the basics of before fully jumping into C++
 - Learn JUCE's Complexities:
 - Through the YouTube's Tutorials

Conclusion to JUCE

- JUCE is overall an excellent starter program to help on create their first initial plug-in.
- There are, however, other programs that are out there once the basics of plug-in mastery have been developed.
- See the following recommendation slide, where will go through those.

Recommended Action

- As previously stated in our conclusion slide, there are other programs that one may want to consider using once they are fully developed their overall knowledgeability of JUCE, these programs are as follows:

- XCode:
- XCode is an essential program to run JUCE. However, it can create plug-ins on its own when used properly.
 - With that knowledge, one should note that XCode should only be used in lieu of JUCE when one has gained completely competency in C++.
 - This is since running XCode without the use of JUCE forces you to create all JUCE's open-source files that are already primarily initialized and created if you were to just use the JUCE program.
 - For some more context, we tried to create a plug-in using only XCode after having spent a whole summer on this project, and I (Stephen Davis) got stuck on step 3 of 10.
 - With that said, I did not have prior knowledge of C, C#, or C++ (and, had only ever coded in BASIC, Python, and Java before. But, even with that knowledge, you should know that I would recommend you take at least 2-4 semesters worth of time to get full understanding of JUCE before trying to move on to just XCode.

- SynthEdit:
- SynthEdit is a tad newer than JUCE (having been created two years afterwards when Windows was inspired enough by JUCE's first ADC in 2006 to make a similar program themselves).
 - Although one may think that a program being built right into the Windows interface would be easier to use than a program that isn't, that isn't the case. The reasons for this are:
 - Unless you are running a DAW that runs on Windows, you can't use SynthEdit, as it isn't a "cross-OS" platform (sorry, MAC users, or Logic Pro users).
 - Additionally, as indicated by the name, SynthEdit can basically only be used for the creation of Synth VST Plug-Ins (this means no sampler plug-ins, no audio plug-ins, and no FX plug-ins that could be created by JUCE).
 - I tested this program myself when trying to create by own synth plug-in (as, I have heard that JUCE is better for all either types of plug-ins, but SynthEdit is where it's at for Synths). However, I found that the Windows format was outdated (it looks like it prefers to run on Windows 7, or Windows 8, not Windows 10, or Windows 11).
 - And the problem with this is that would mean that your computer would have to an un-updated refurbished Windows PC from the years 2016 – 2018 exclusively to properly run both JUCE (as it requires a newer computer) and SynthEdit (as it requires an older computer), which is as highly unlikely, as it is illogical. However, if you do happen to fall into this extremely small case of computer users and DAW users, then you should definitely at least try using both JUCE and SynthEdit.

!THANK YOU!

A Big Thanks Goes Out to the
Wonderful Prof. Manzo for
Getting Me Started in This
Project and Assisting Me
Along the Way!

References

This Picture Shows Our
Sources Used