Download and Login to the NIWeek Mobile App

More Information at ni.com/niweek
As software languages go, LabVIEW is odd and the relationship between the diagrams we make and our brains is interesting and poorly understand. This presentation investigates this, discusses the tools and techniques we use and tries to answer the strangest question of all...Why is LabVIEW Fun?
Prof Watts' Theory on Why Programming in LabVIEW is Fun!

Steve Watts
Childrens Entertainer

Why is LabVIEW Fun? Cos it's like making balloon animals on your computer
19 Years

eCLA 2012 LabVIEW Smells
NIDays 2013 How to Polish Your Software and Development Process to Wow Your End Users
eCLA 2014 Process Smells
NIWeek 2016 TS9456 - ISO 9000 and LabVIEW
NIWeek 2016 PS9044-Shock Testing using Multiple Synchronised Racks
ECLA 2017 Risk and Mitigation
NIWeek 2017 The SSDC Way: Desire Paths to a Simple Software Process
ECLA 2018 Modular Design Deep Dive

Read my blog on ni.com – Random Ramblings on LabVIEW Design
Some places you will find SSDC software
Where you will find SSDC software

In the Expo Hall

Sierra Peaks
#OurGiantsAreFemale

Diana Merry / Adele Goldberg

https://en.m.wikipedia.org/wiki/Diana_Merry
https://www.youtube.com/watch?v=1GNiH85PLVg
https://www.youtube.com/watch?time_continue=105&v=auXcc7WSczM
Goal

- Dig into why LabVIEW is Fun.
- Discuss the psychology of programming.
- Hopefully this will help your design decisions and inform IDE design priorities.
ALL PROGRAMMERS ARE FORCING THEIR BRAINS TO DO THINGS BRAINS ARE NOT MEANT TO DO IN A SITUATION THEY CAN NEVER MAKE BETTER

https://www.youtube.com/watch?v=MticYPfFRp8 Coding sucks
I love LabVIEW

BUT WHY.....
A good tool, like a high-quality wrench, becomes part of your hand, so you feel your hand is turning the bolt directly. The machinery disappears, and you feel connected to the object in question.
Brett Victor Intro

Immediacy

>1 Million views
Demo

LabVIEW and Immediacy
Brain Slots and Complexity

7±2

Millers Law  - George A Miller 1956

http://psychclassics.yorku.ca/Miller/
A programming environment can also convey the experience of immediacy, drawing the programmer closer to the program.

Enabling reflection in action
Backtalk is a necessary condition for reflection in action, just as immediate feedback is a condition for flow performance.

Listen to your tools and materials Dr Carlo Pescio IEEE Software 2006
Flow State

http://www.ted.com/talks/mihaly_csikszentmihalyi_on_flow
Flow State

Task Absorption

Is Flow Really Effortless? The Complex Role of Effortful Attention
David J. Harris, Samuel J. Vine, and Mark R. Wilson
University of Exeter 2016
I'll go farther – your program is a user interface to the solution of a problem

Bret Victor – WorryDream.com
• What if rather than plan out everything we experiment?
• What if rather than remember all the details of everything it's just as easy to test it?
• What if the language we use is dynamic enough that it breaks the traditional programming paradigm?
You have just built and operated your first VI.

If you had problems, carefully review these steps.

1. Did you choose Indicator Only for the $X^Y$ control?
2. Are the X and Y controls in the diagram wired to the proper places on the $X^Y$ icon?
3. Are the wires solid lines?
4. Is the X value in the X control and is the Y value in the Y control?

**Experiment with the New Instrument**

Try changing the values of the base and exponent controls. To change the values on the digital-style numeric controls, use the same technique you used to input the initial values. Then click the Go button and observe the new answer.
Most engineers learn about an instrument by studying its front panel and experimenting with it. With its mouse, menus, scroll bars, and icons, the Macintosh proved that the right interface would also allow someone to learn software by experimentation.


Ask Fabiola about the Fifth Edition!
Temporal Immediacy

In programming, delay between an effect and observing related events or data in the program puts a strain on programmers’ short-term memory to hold all the relevant information in their heads while waiting for the programming environment to catch up. In Eisenstadt’s article on bug war stories in this issue, 15% of bugs are directly attributable to temporal distance between cause and effect—the largest single source of error in his survey.

David Ungar, Henry Lieberman, Christopher Fry
Communications of ACM Vol 40 Iss 4 1997
Debugging and the experience of immediacy
Spacial Immediacy

The physical distance between causally related events is kept to a minimum. The reason is the same as for temporal immediacy: Events widely separated by space on the screen require users to devote more conscious effort to link them, forcing them to shift attention and putting a strain on their short-term memory.

David Ungar, Henry Lieberman, Christopher Fry
Communications of ACM Vol 40 Iss 4 1997
Debugging and the experience of immediacy
PLATONIC THINKING
Reasoning over Observation
PLATONIC THINKING

REJECTED
Reactive Debugging

Conversational Software

- Make rapid changes in front of the user
- Involve the user in the process*
- Agile <-LabVIEW is the original Agile language
- We (and our customers) learn as we go

*Martin Fowler: Not Just Code Monkeys
https://www.youtube.com/watch?v=4E3xfR6IBII
Debugging While Programming

• When I'm designing a UI I want to be able to quickly see how my changes affect the feel
• I'm a designer not a programmer....
• I like to try things out, therefore the IDE needs to allow me to Play
• Cause and Effect Brain!
Constructionism
Think about how LabVIEW works, if your design diminishes these advantages needlessly you are making your life hard and this risks the success of the projects.
Hindering LabVIEW

- Packed Libraries
- Queues
- User-Events
- Asynchronous Frameworks
- Variants

What about when they are combined?
Context Switching

Moving from Fault Finding to Searching.
Context Switching - Hierarchy

Request →

Discoverability – can you ascertain from the block diagram what the program is doing?

Code to Act on request

Does the source code afford the solution to the problem?
Context Switching

Block Diagram Fundamentalist

I want seamless navigation from visible manifestation of a bug to the code under investigation.
Context Switching – Semantic Zoom example.

Tow Tank State Machine

- Initializing
- Waiting Login
- Exit
- Login Success
- Acquiring
- Acquire
- Idling
- Project Opened
- Calibrating
- Add Channel
- Deleting Channel
- Adding Channel
- Exit

NATIONAL INSTRUMENTS
Simplest Functional VI - Testable
Non-testable VI
Helping LabVIEW

Immediately Testable
Use Non-runnable vis as wrappers/carriers for Immediately Testable vis
Navigation Aids

Put floating controls in with their events, aiding navigation from front panel to block diagram

Signifiers – considering UI rules for your block diagram
Navigation - Searchable
Constant VI is a good example of this.
Why Debugging While Programming is Good

Allows us to explore areas outside of our domain of expertise.
Terms

- Immediacy
- Reactive Debugging
- Experimental Programming
- Conversational Software
- Debugging while programming
- Flow
- Task Absorption
- Explanation in Context
- Context Switching
Key Points

- LabVIEW's #1 advantage is its immediacy
- The iteration of rapid debugging is actually a more natural way to program
- This immediacy in the IDE allows us to explore different domains
- We can make our lives easier by splitting Non-runnable VIs and ITVIIs
- We can leverage this by thinking about navigation and searching
- We can consider UI rules when creating our block diagrams
Summary

- LabVIEW is really good to debug with.
- This is a real advantage, the design decisions we make can affect this advantage.
- Be kind to your block diagram.
- LabVIEW is quite different from other programming languages and it is pretty much ignored by CS.

There is very little research into this aspect of computer science, perhaps we need to do it ourselves!
And make our own rules and processes
If you want to hear me speak about state machines done right
Before you go, take the survey.
Slide Divider Topic Place Here
Stay Connected During and After NIWeek

ni.com/niweekcommunity
facebook.com/NationalInstruments
twitter.com/niglobal
youtube.com/nationalinstruments