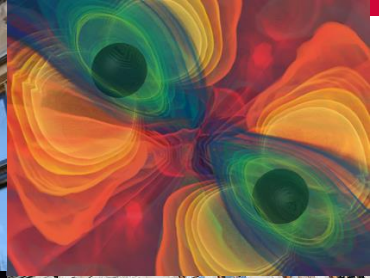
Unifying MSc Physics and MSc Astrophysics Problem-Based Learning with LabVIEW NXG: A Critical Review

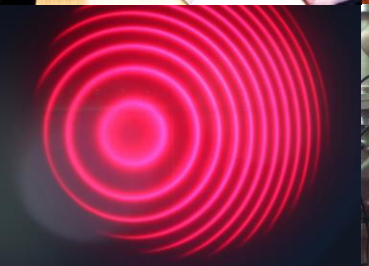
NI Academic Users Forum 2019

Dr Richard James Lewis

Cardiff University School of Physics and Astronomy (PHYSX)




Overview

Plan: transitioning core modules to NXG 2.1

- Summary of MSc activity at Cardiff PHYSX
- Core MSc module structure: then and now

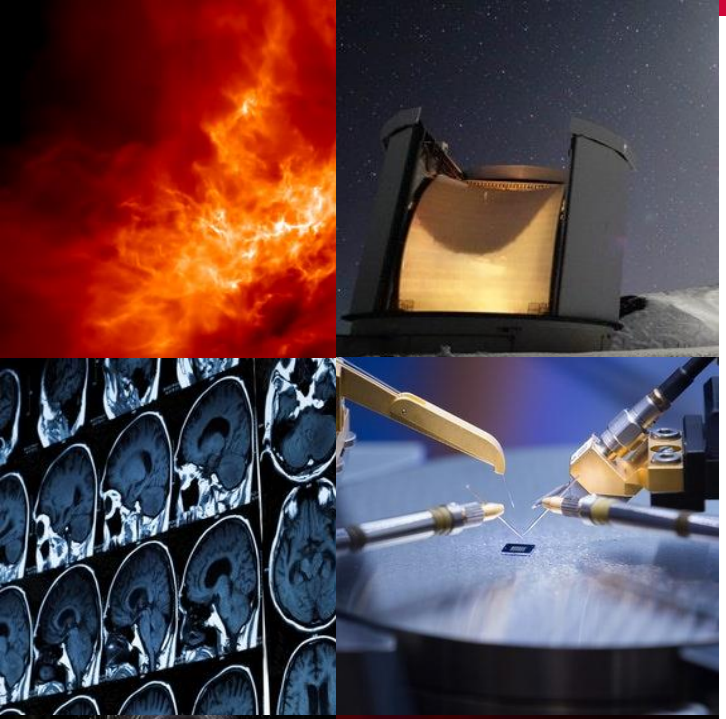
Execution: learning and application

- Impact of NXG on LabVIEW language learning
- Impact of NXG on problem-based learning
- Logistical issues and their solutions

Outcome

- Student reception of new core module
- Future MSc core module design
- Summary of LabVIEW NXG 2.1 critique

Plan: transitioning core modules to NXG 2.1



Summary of MSc activity at Cardiff PHYSX

Programmes

- MSc Physics
- MSc Data-Intensive Physics
- MSc Compound Semiconductor Physics
- MSc Astrophysics
- MSc Data-Intensive Astrophysics
- MSc Gravitational Wave Physics (2019/20)

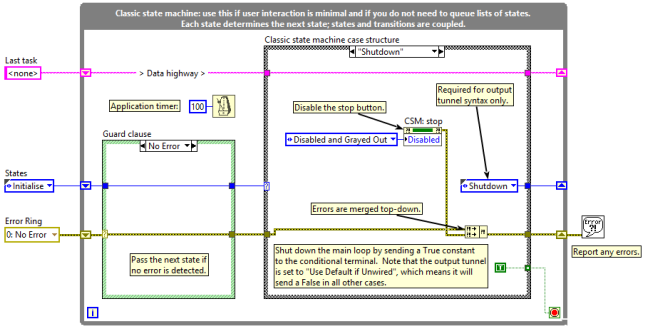
Cohort of 2018/19

- 24 students (11 Physicists, 13 Astrophysicists)
- 2:1 minimum entrance requirement
- Dedicated co-located MSc teaching facilities
- 20cr common core module with LabVIEW NXG

Transitioning core modules to LabVIEW NXG

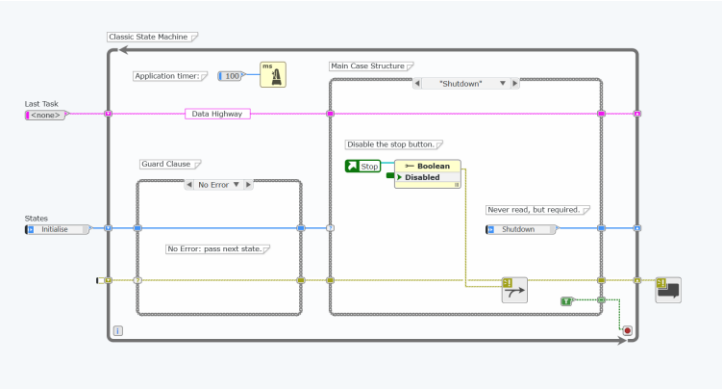
Major restructuring for 2018/19

- Merger of core autumn physics and astrophysics modules
- MSc Physics pattern adopted for this 20cr module:
  - Practical introduction to LabVIEW (10cr)
  - Student-lead micro-projects (10cr)
- Transition from LabVIEW 2015 to LabVIEW NXG 2.1*



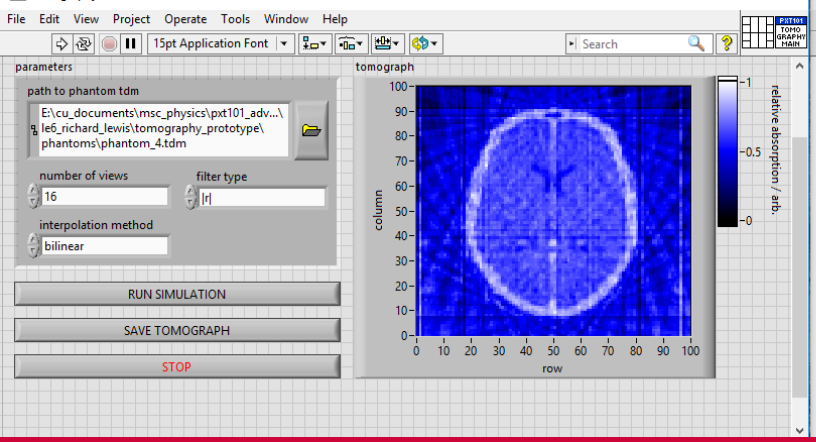
Why LabVIEW NXG?

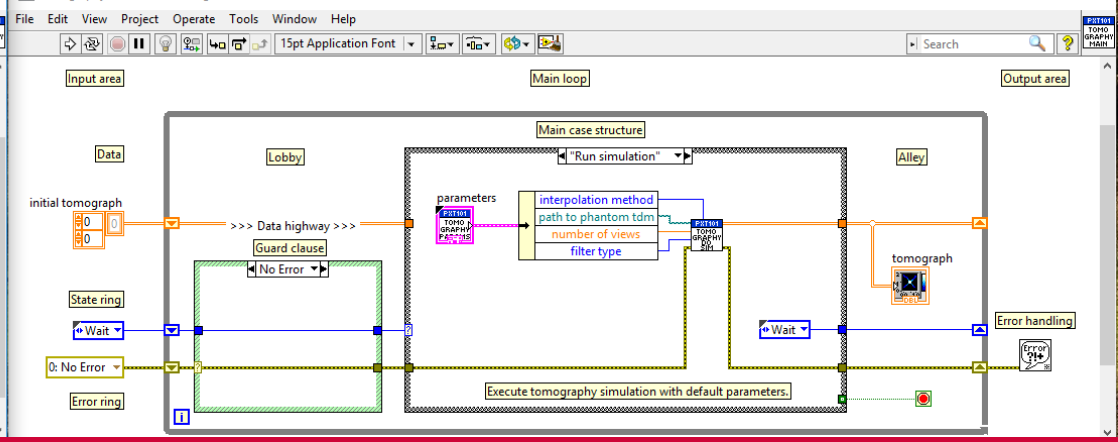
- **Pedagogy:** gentler, more logical learning curve
- **Logistics:** useful from day 1, therefore always relevant



What do we want to maintain / enhance?

- Students feel valued as part of our academic community
- Exceptionally high student feedback score average of **93**





Old structure (LabVIEW 2015): drills to application

Week 1: projects, front panel, block diagram, dataflow, Express VIs, AAP

Week 2: arrays, clusters, file I/O, case structures, loops, sub VIs, errors, style

Week 3: functional specifications, efficient VI engineering

Week 4: interfacing with hardware, MAX, VISA, AAP with hardware

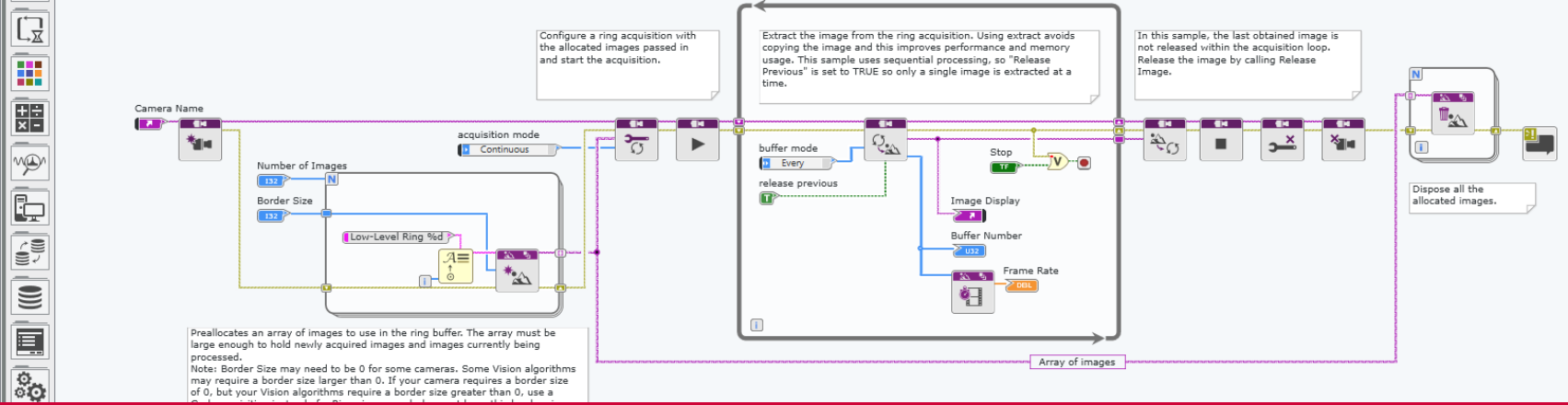
Week 5: development paradigms, type definitions, DAQmx

Week 6: error handling, tunnels, shift registers, classic state machines

Week 7: event structure, event-driven state machines, functional global variables

Week 8: queues, queued state machines

Weeks 9 to 11: applying LabVIEW to micro-projects



Planned (LabVIEW NXG 2.0): DAQ-first, applications throughout

Week 1: a guided tour of LabVIEW NXG, establishing common and linked contexts

Week 2 (AAV and IO I): no coding required, NXG functionality, DAQ on 1 and 2 channels

Week 3 (AAV and IO II): coding drag and drop, DAQ on 1 and 2 channels, images

Week 4 (AAV and IO III): coding from scratch, DAQ on 1 and 2 channels, images

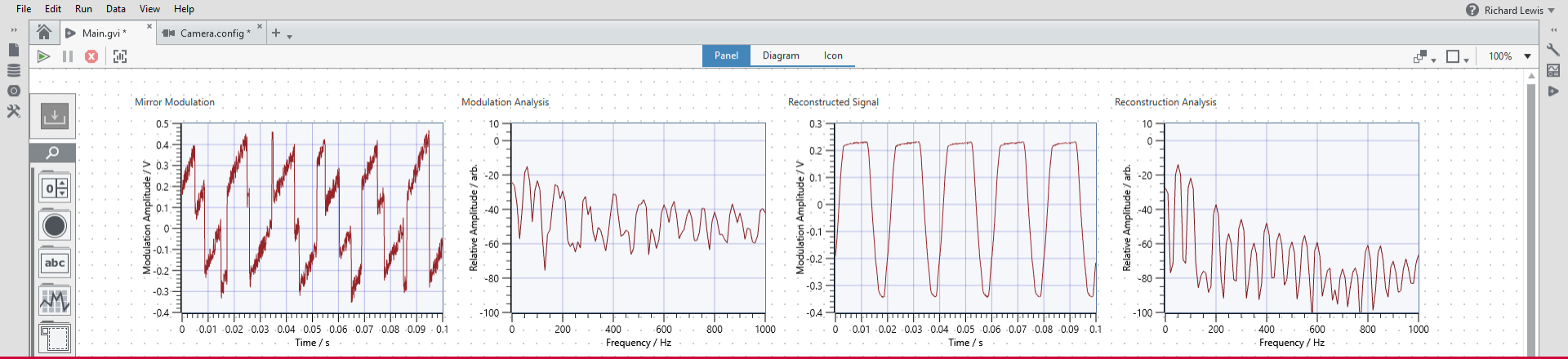
Week 5 (looping and iteration I): looping and iteration, looping AAV code

Week 6 (looping and iteration II): classic state machines

Week 7 (looping and iteration III): event-driven state machines

Week 8 (looping and iteration IV): queued state machines

Weeks 9 to 11: **continuing to apply** LabVIEW to micro-projects



Result (LabVIEW NXG 2.1): DAQ-first, applications throughout

Week 1: a guided tour of LabVIEW NXG, establishing common and linked contexts

Week 2 (AAV and IO I): no coding required, NXG functionality, DAQ on 1 and 2 channels

Week 3 (AAV and IO II): coding drag and drop, DAQ on 1 and 2 channels, images*

Week 4 (AAV and IO III): coding from scratch, DAQ on 1 and 2 channels, images*

Week 5: module recess

Week 6 (looping and iteration I): looping AAV code and classic state machines

Week 7 (looping and iteration II): classic and event-driven state machines

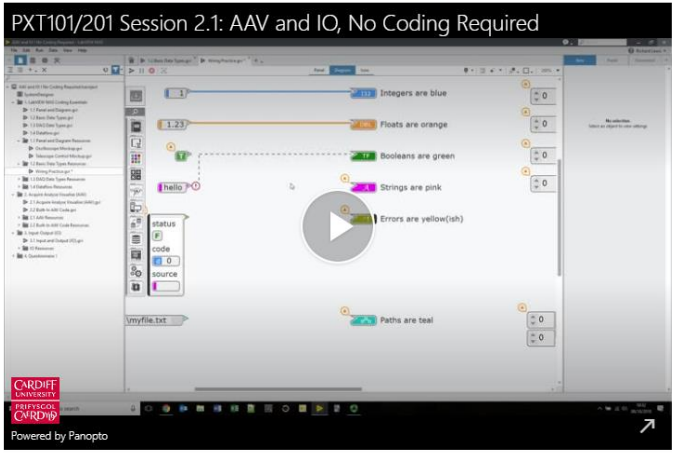
Week 8 (looping and iteration III): : event-driven and queued state machines

Weeks 9 to 11: <u>continuing to apply</u> LabVIEW to micro-projects

Execution: learning and application

Impact of NXG on language learning

Session 2.1: Basic Coding (Dropping in Controls and Indicators, Wiring, AAV)

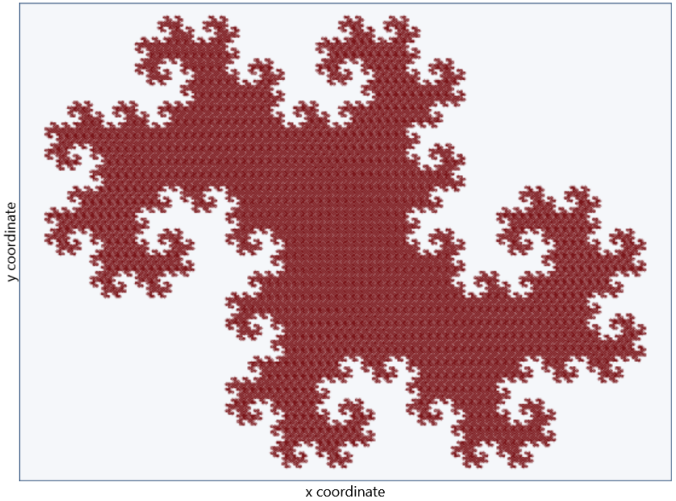


Impact of NXG on language learning

Summary of targets for NXG course

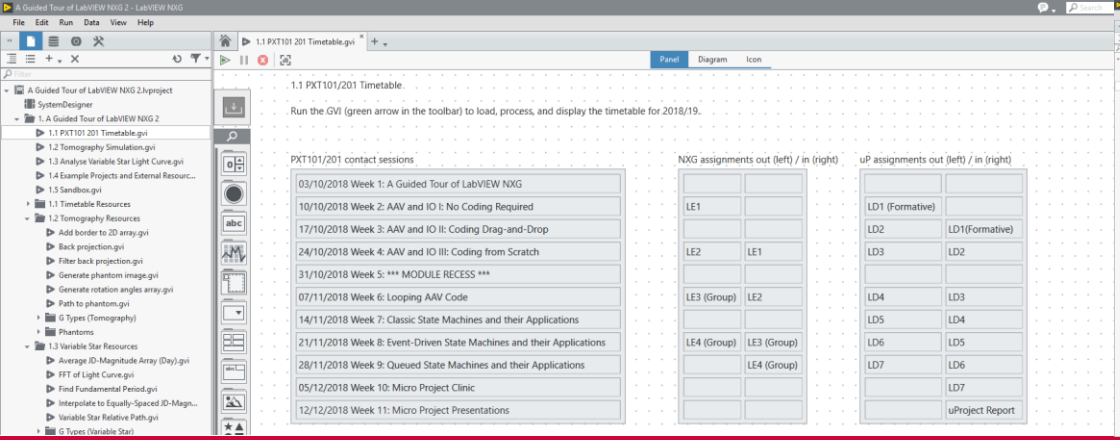
- Everything in projects (no slides)
- Every session recorded on Panopto
- No (nasty) surprises, anticipate student issues
- Physics **and** Astrophysics examples throughout

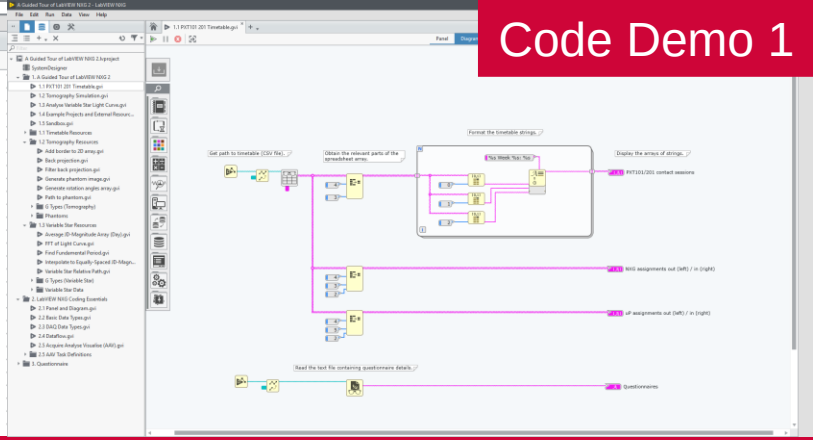
Dragon Curve



Lots to talk about, so have to be selective:

- Experience of keeping everything in projects
- NXG interface advantages and issues
- (Somewhat) nasty surprises, workarounds



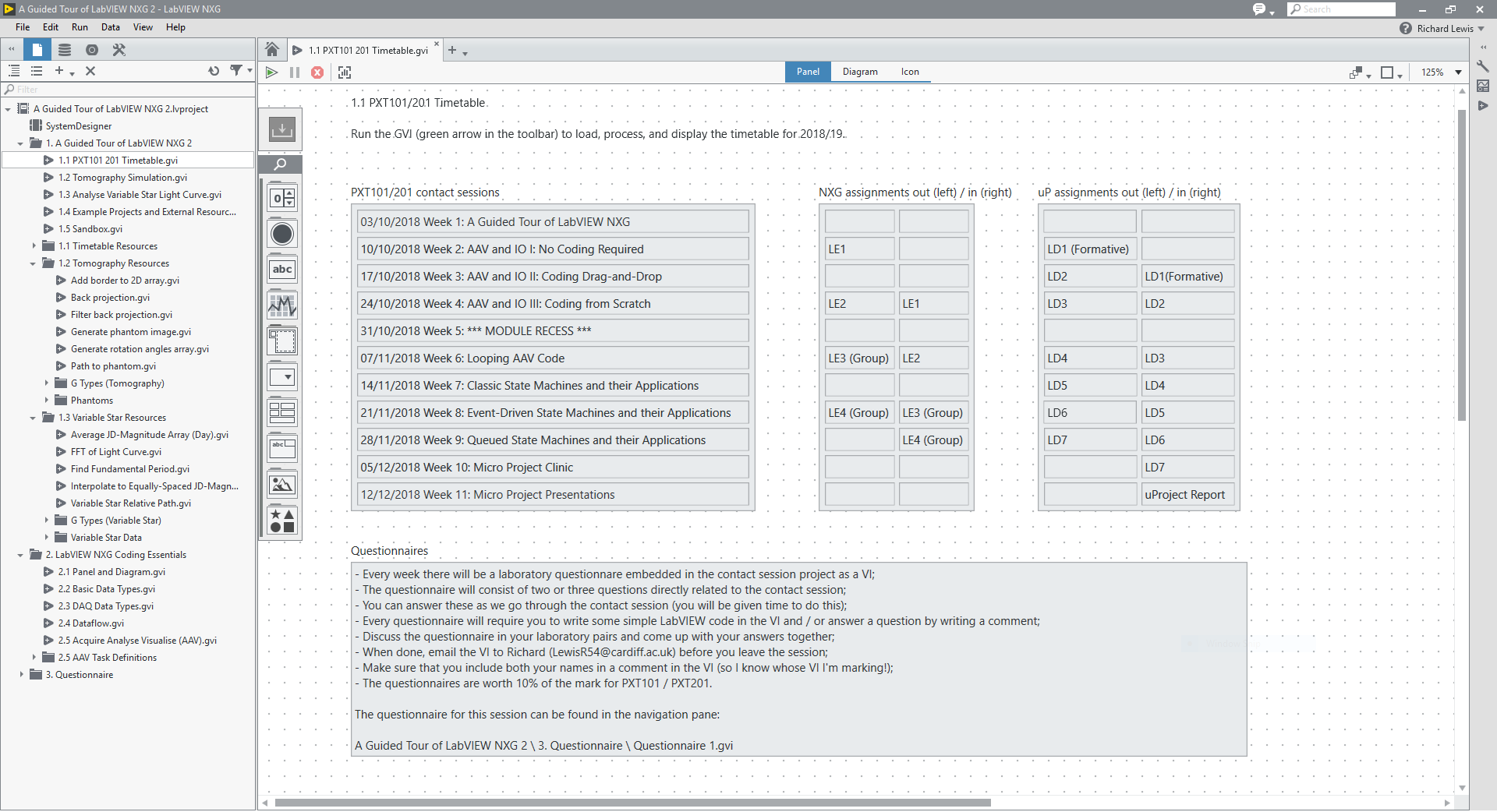


Code Demo 1

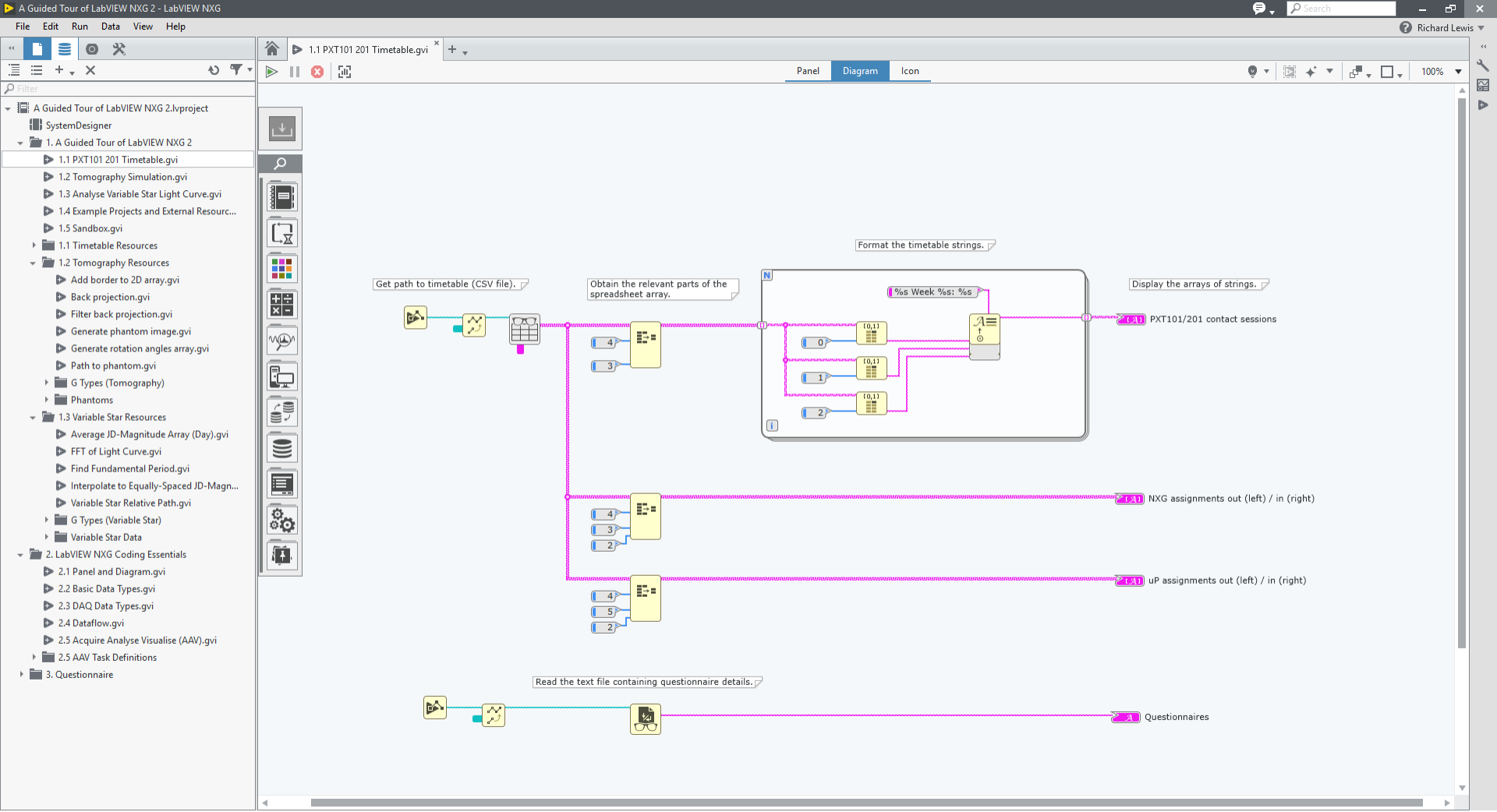
Experience of keeping everything in projects

- Dramatically simplifies distribution logistics, projects enforced, real directories
- Full integration of session questionnaires, exercise scripts, and their solutions
- Live streaming of session capture allows students to review during hands-on
- Sessions much more dynamic than for 2017/18, with far less start-and-stop

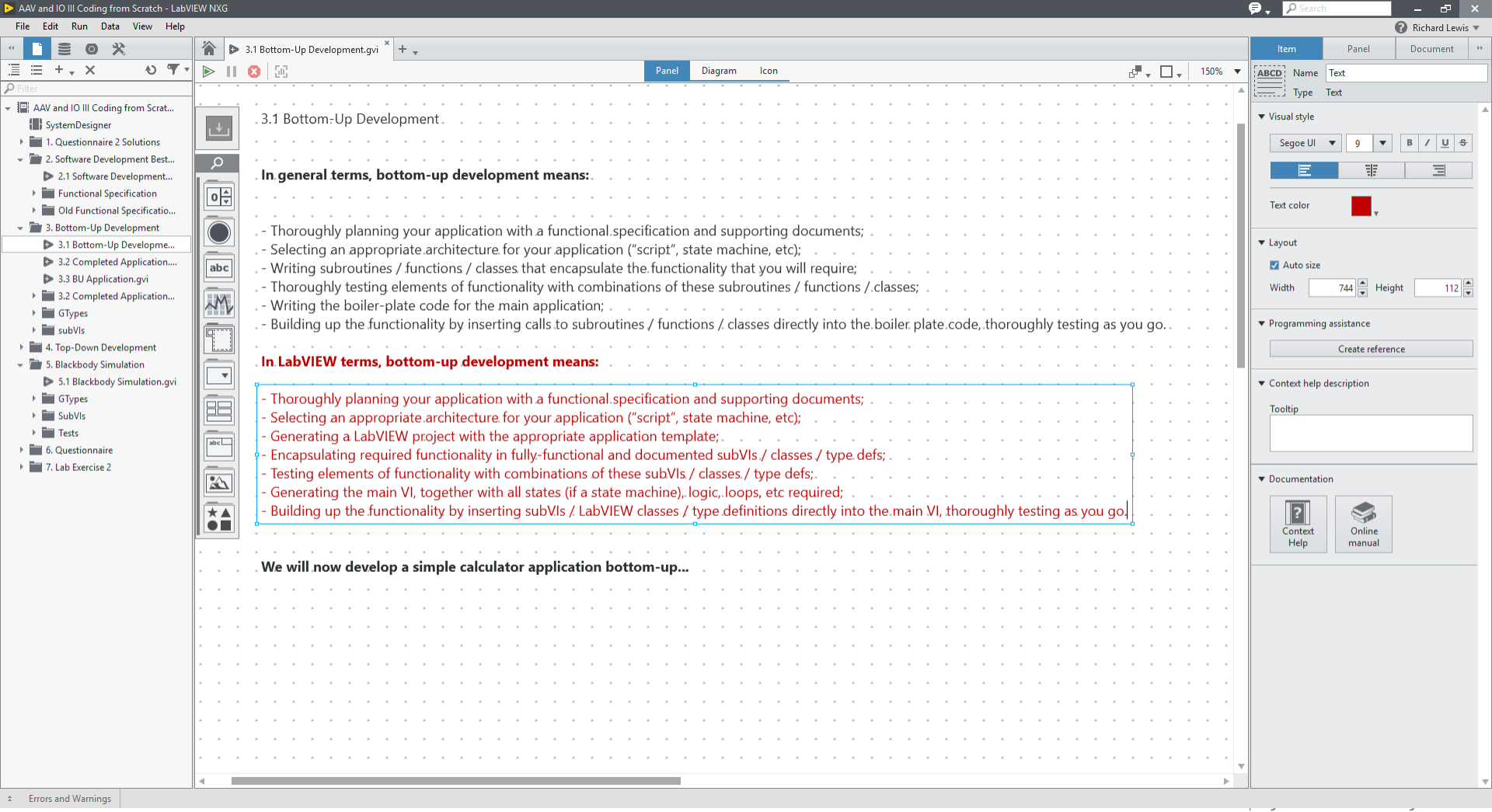
- Formatting tools in NXG for free text comments are very simplistic
- Workbooks are much more sophisticated, but not available to end user



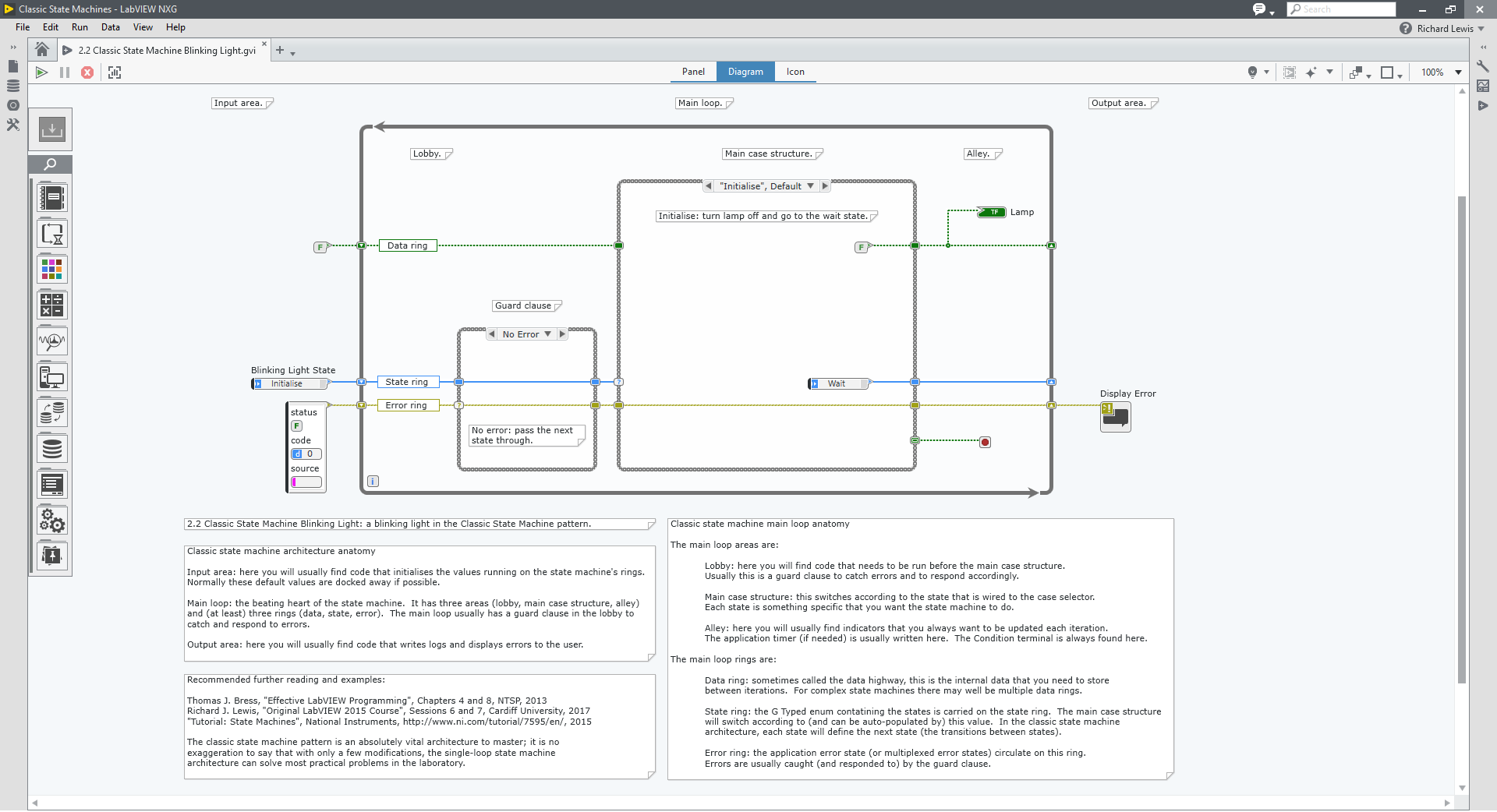
A contact session can be contained entirely within a LabVIEW NXG project. The use of external materials and other applications has been minimised to maintain session flow.



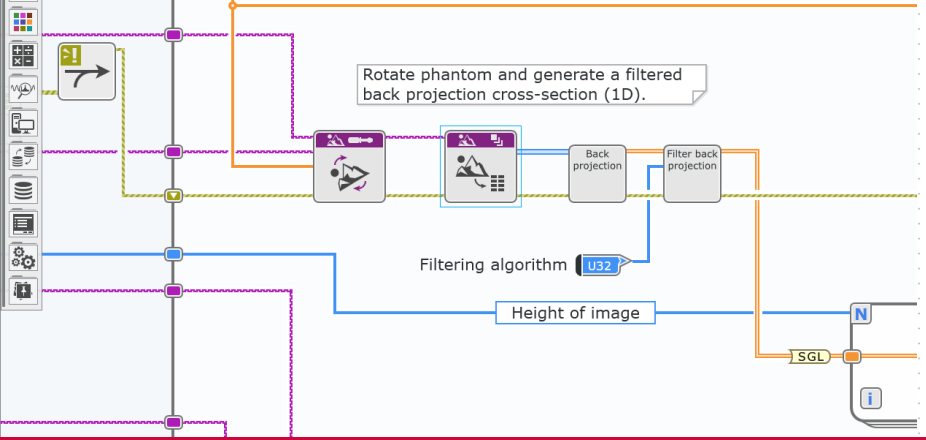
Embracing the “everything in a project” concept allows for unexpected teaching and learning opportunities. Here we can use the course timetable to demonstrate dataflow!

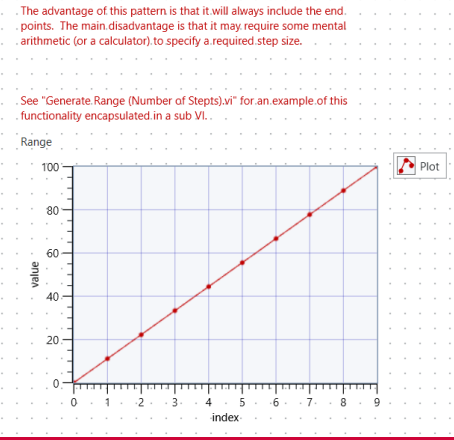


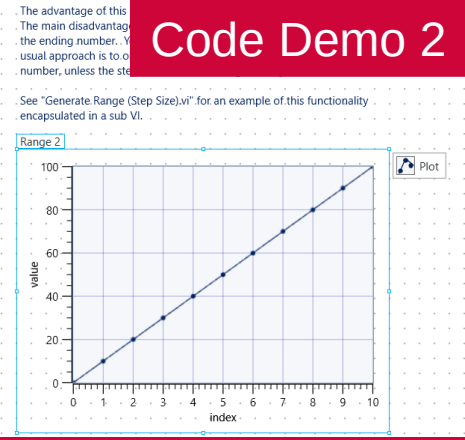
The text formatting tools in NXG 2.1 apply to entire text blocks. In the Panel shown here are no fewer than six text blocks.



The text formatting tools in NXG 2.1 apply to entire text blocks. In this Diagram we would like to be able to shade the boxes and bold headings, but this is not possible.





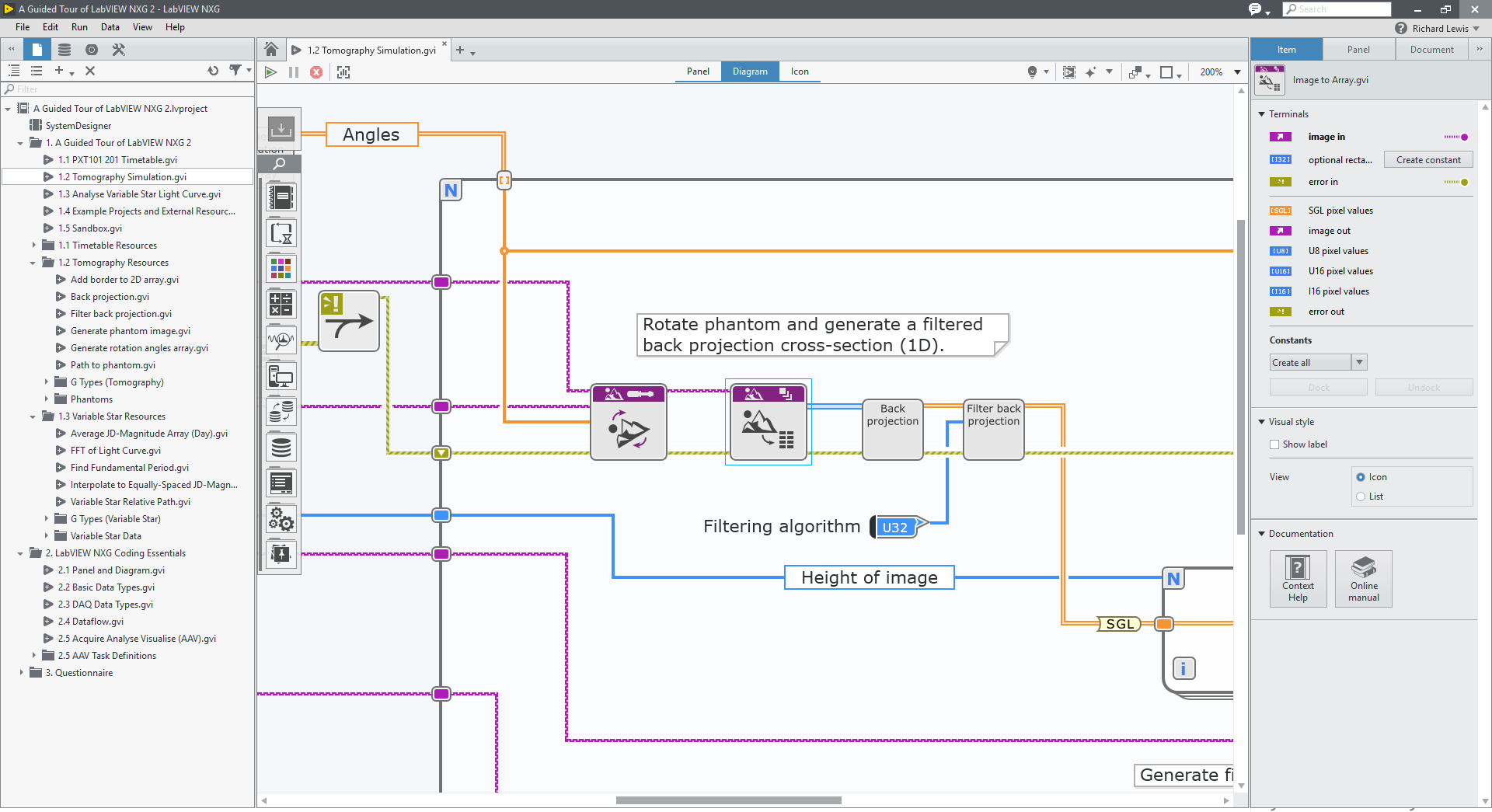


Code Demo 2

NXG interface advantages and issues

- NXG UI is vastly superior to LabVIEW UI in almost every usability aspect
- Panel and Diagram zoom enhances delivery, recording, and development
- Docking constants and the properties pane really help with clarity

- The ability to zoom necessitated a rethink of what “good style” means
- NXG is noticeably slower and less stable compared to LabVIEW
- Some consistency issues: analysis panel only available for waveforms, e.g.



The zoom function is a killer feature on the Diagram. It is particularly useful in session recordings to remove visual clutter.

Looping AAV Code - LabVIEW NXG

File Edit Run Data View Help

Richard Lewis

3.1 Generating Ranges.gvi

Panel Diagram Icon 125%

i. The "start, end, number of steps" pattern

Start value 0

Number of steps 10

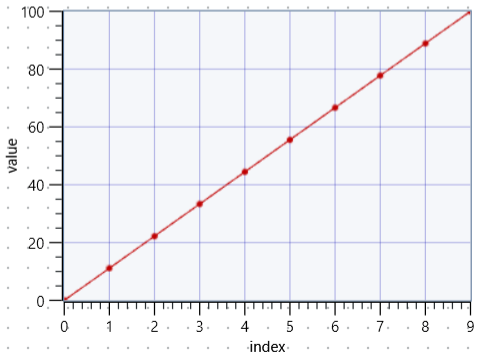
End value 100

The advantage of this pattern is that it will always include the end points. The main disadvantage is that it may require some mental arithmetic (or a calculator) to specify a required step size.

See "Generate Range (Number of Steps).vi" for an example of this functionality encapsulated in a sub VI.

Range

Plot



ii. The "start, end, step size" pattern

Start value 2 0

Step size 10

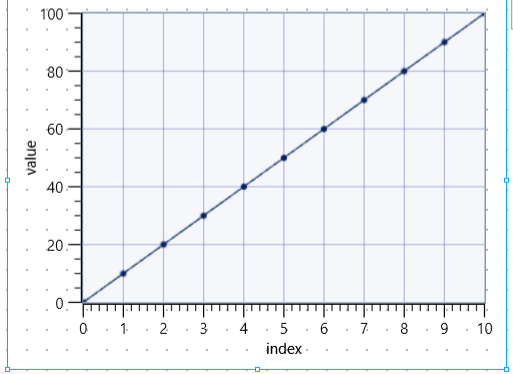
End value 2 100

The advantage of this pattern is that you can specify a required step size. The main disadvantage is that the step size you specify may not include the ending number. You have to decide how to handle such cases; the usual approach is to only go as far as the largest number before the ending number, unless the step size divides the range exactly.

See "Generate Range (Step Size).vi" for an example of this functionality encapsulated in a sub VI.

Range 2

Plot



Item Panel Document

Name Range 2

Type Cartesian Graph

Parts

Plot legend
Cursor legend
Scale legend
Graph tools

Plots

Metadata always overrides plot names

Visual style

Anti-aliased

Label

Show label Label color

Segoe UI 9

Label placement Top left

Content font

Segoe UI 9

Layout

Width 392 Height 296

Plot area

Auto size

Left 50 Top 8

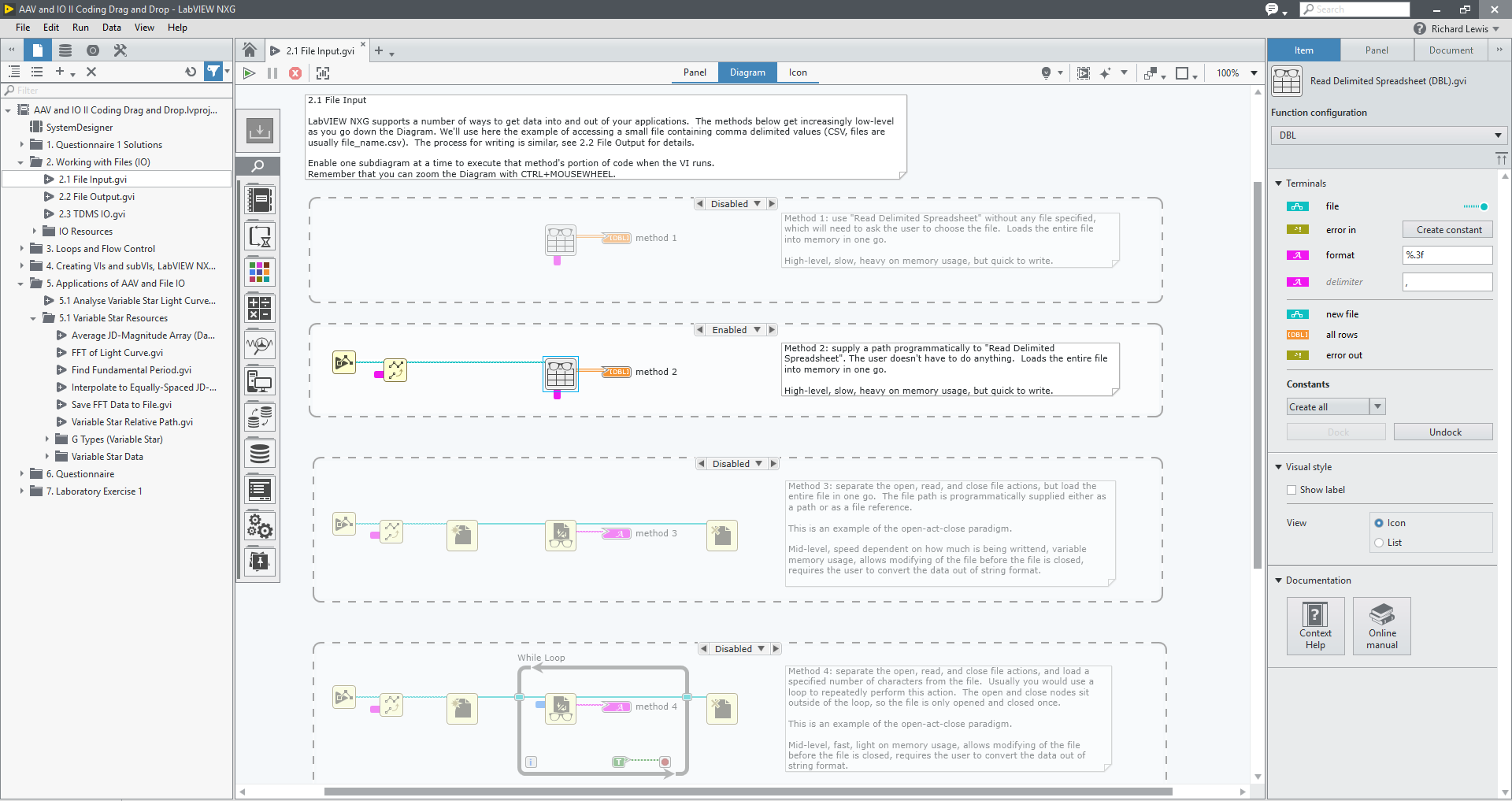
Right 7 Bottom 43

Behavior

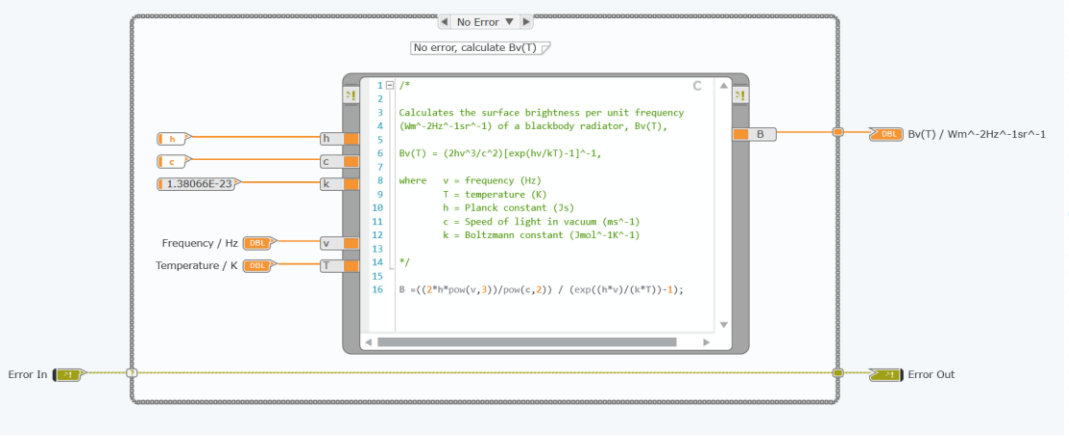
Object Control Indicator

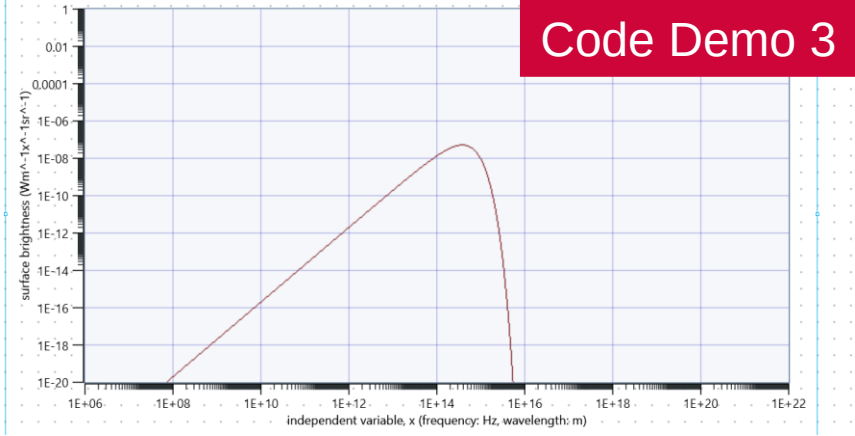
Errors and Warnings

The zoom function works equally well on the Panel. It is particularly useful in session recordings to remove visual clutter.



The zoom function aids exploration from the student's perspective, allowing clearer focus on individual areas of code.



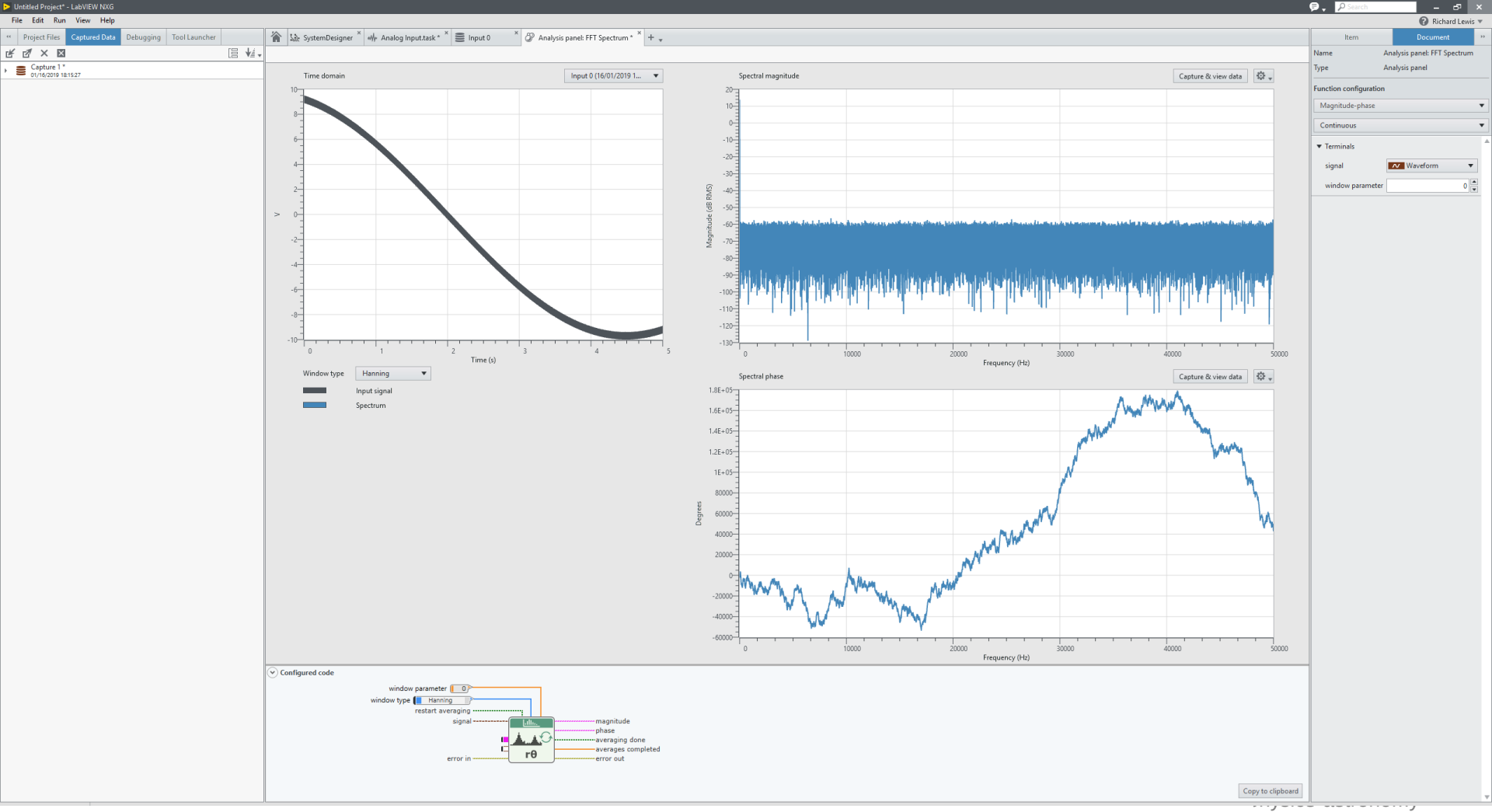


Code Demo 3

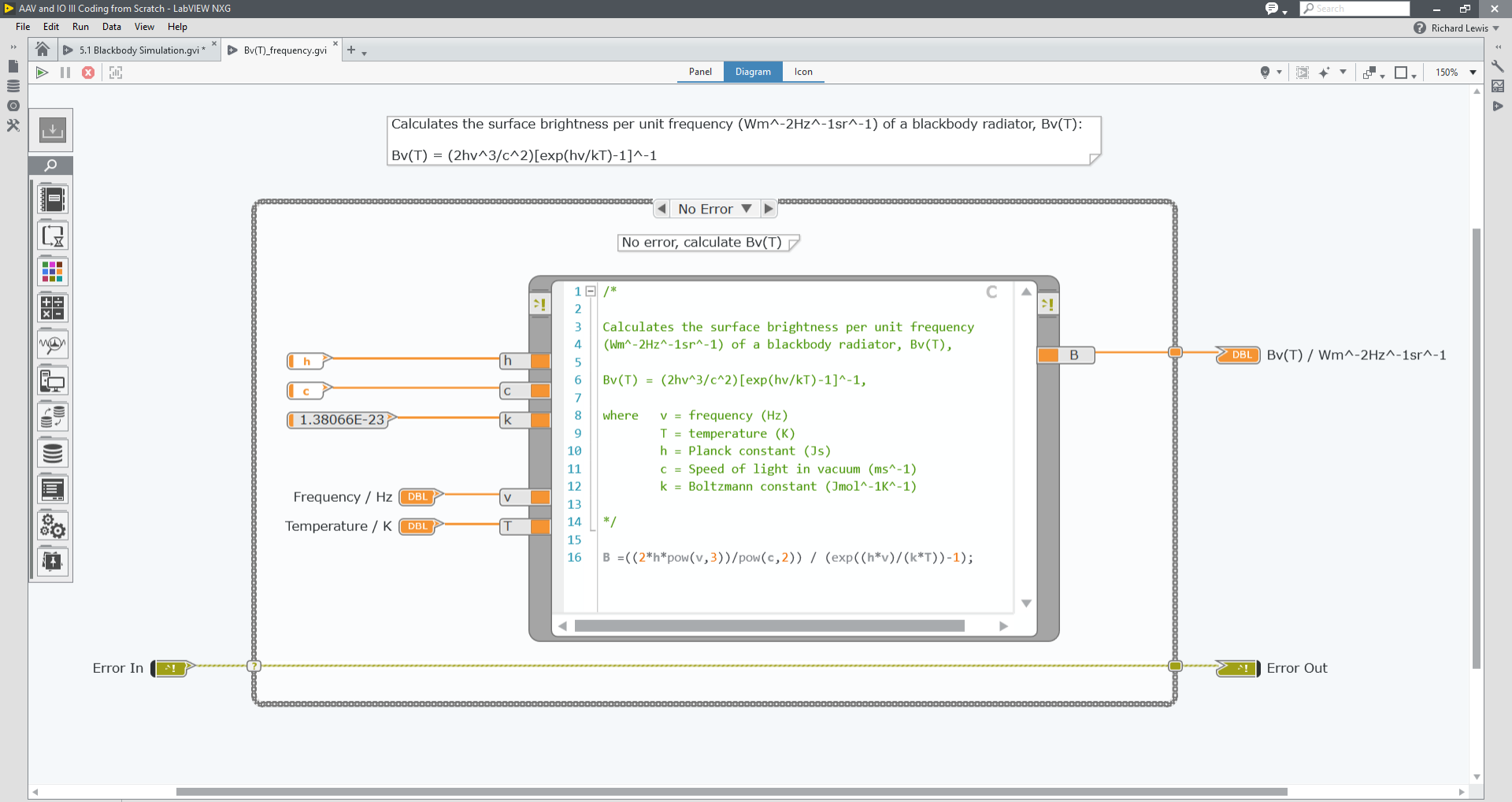
(Somewhat) nasty surprises: some workarounds required

- Analysis panels (from direct acquisition / capture) unreliable across PCs
- Cannot distribute compiled C Node code (crashes target PCs)
- G Types do not always work as expected (i.e. as strict type definitions)
- LabVIEW NXG 2.0 code will not load at all on LabVIEW NXG 2.1 machines

- Have reported the bugs to NI, will provide code to reproduce the errors



The analysis panel is not available for all data types. Attempting to invoke the analysis panel occasionally crashes NXG 2.1



C Nodes in NXG 2.1 have a tendency to crash when projects are run on PCs other than the original machine.

Execution: learning and application

Impact of NXG on problem-based learning

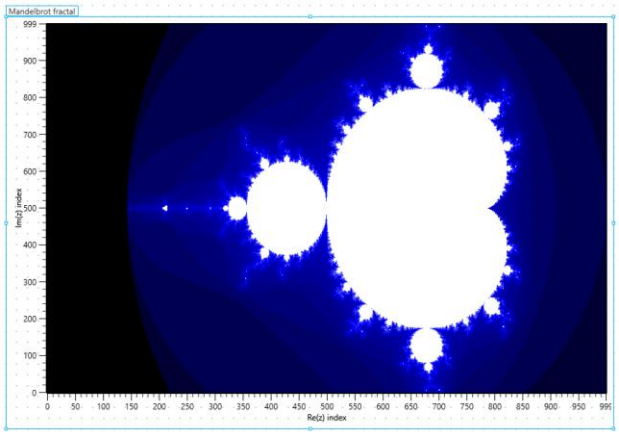
Impact of NXG on problem-based learning

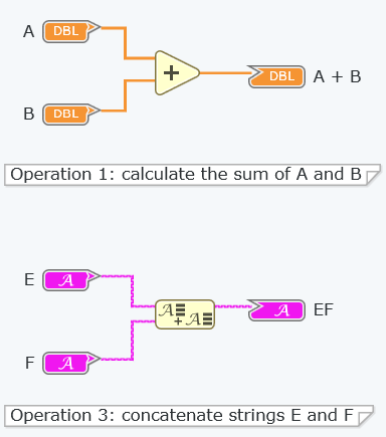
Aims: LabVIEW NXG course contact sessions

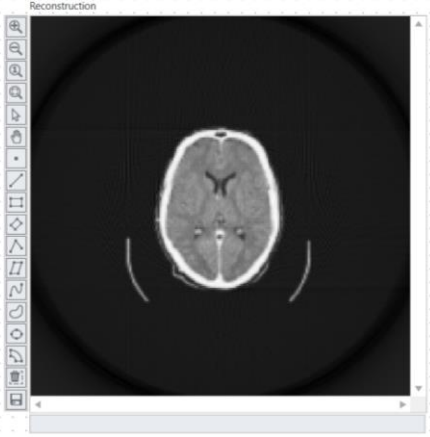
- Start useful, stay useful
- Flatten the learning curve
- Remain relevant to Physics and Astrophysics
- Leverage NXG's strengths (UI, projects, etc.)
- Retain old module's USPs (funcspecs, etc.)

Aims: Student-lead micro-projects

- Start useful, stay useful
- Allow more sophisticated applications earlier
- Provide a common language for related projects to encourage collaboration
- Retain old module's USPs (community, etc.)





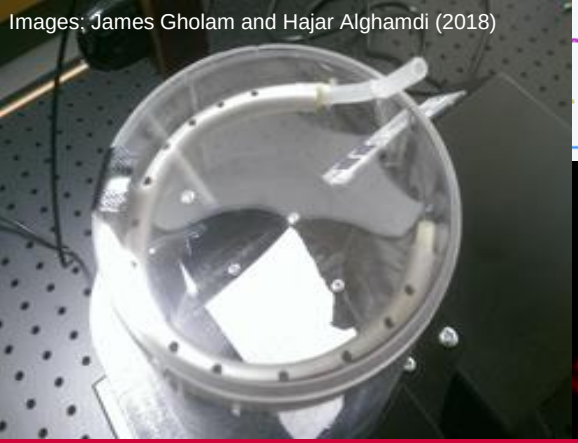


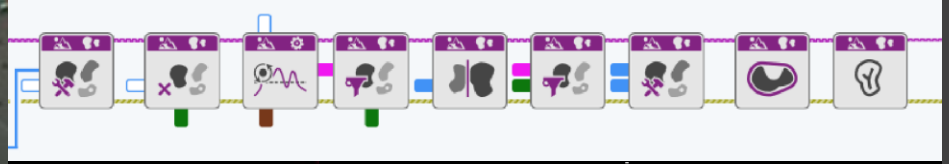
Impact of NXG in contact session problem-based learning

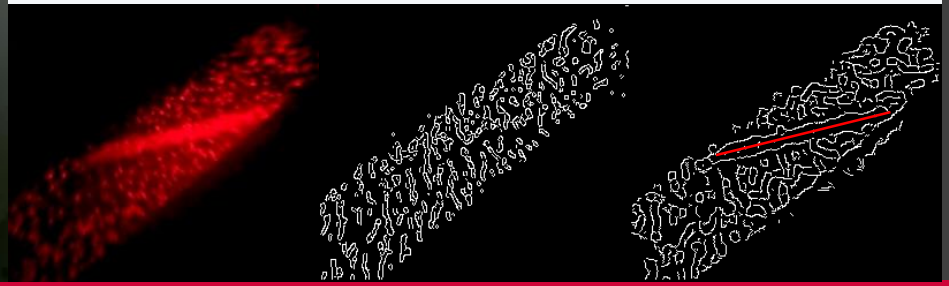
- Flashes of brilliance when NXG's strengths all came together
- Freely-flowing sessions dramatically accelerated learning, made teaching more efficient
- Staying "in project" allowed students to concentrate more effectively on their tasks
- Full retention of original module's USPs, group work in particular enhanced

- Instability and bugs introduced interruptions in otherwise freely-flowing, dynamic sessions
- DAQ-first concept partially thwarted by crashes, but the concept itself is sound

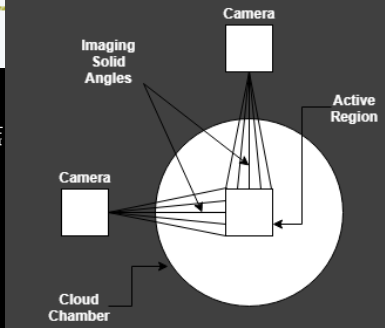
Images: James Gholam and Hajar Alghamdi (2018)







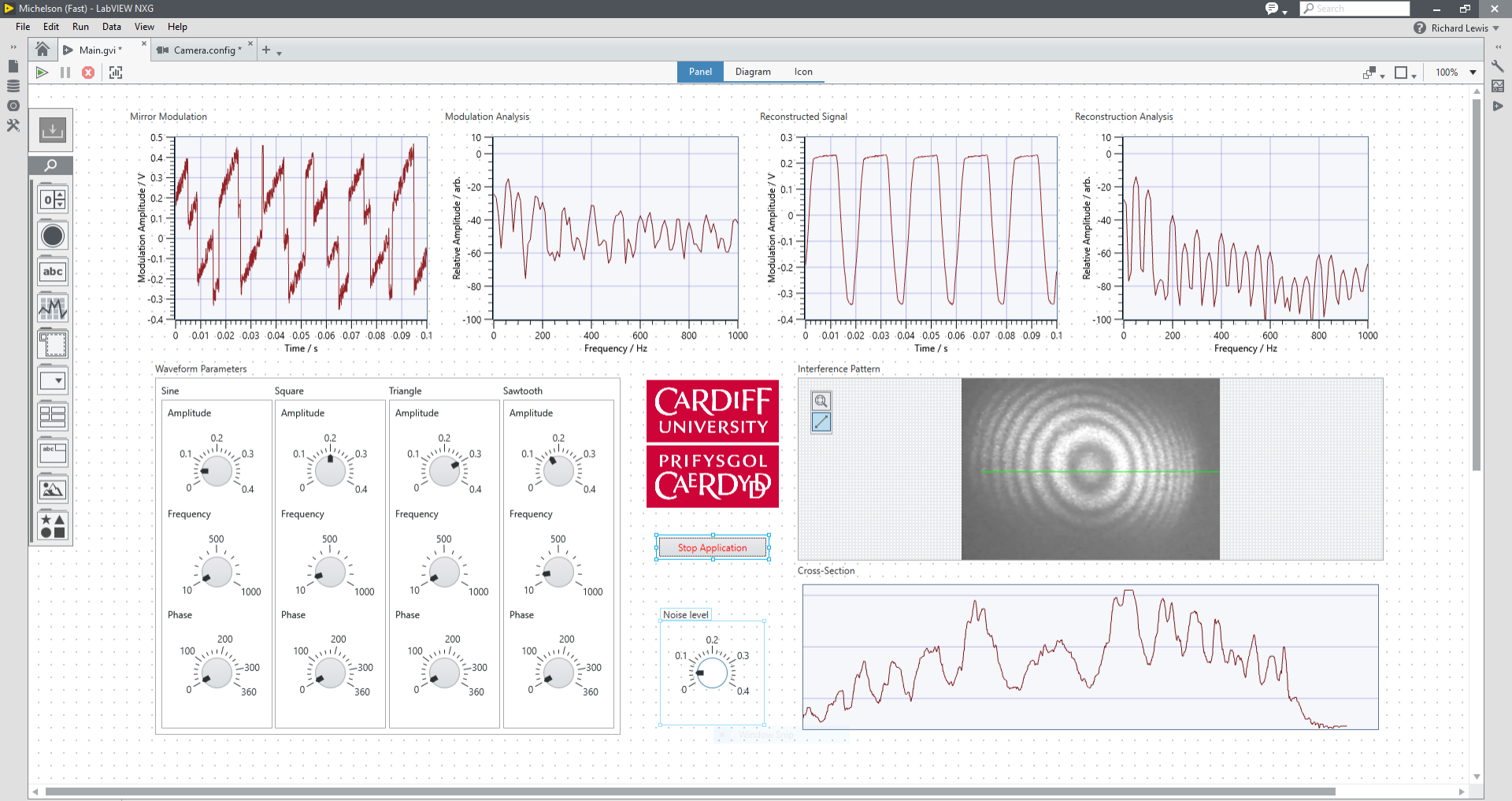
Code Demo 4



Impact of NXG in student-lead micro-projects

- This is where NXG really shone: a noticeable increase in efficiency
- Image analysis in particular was found to be much more accessible

- Students explored the language in much more depth than in previous years
- Students progressed much further in the micro-projects than in previous years
- One micro-project formed the basis of an exhibition at a Royal Society Event…



Michelson Interferometer as a LIGO Analogue: prototyped in an MSc Physics micro-project 2018

Execution: learning and application

(Other) logistical issues and their solutions

(Other) logistical issues and their solutions

Point release 2.0 to 2.1 broke compatibility(!)

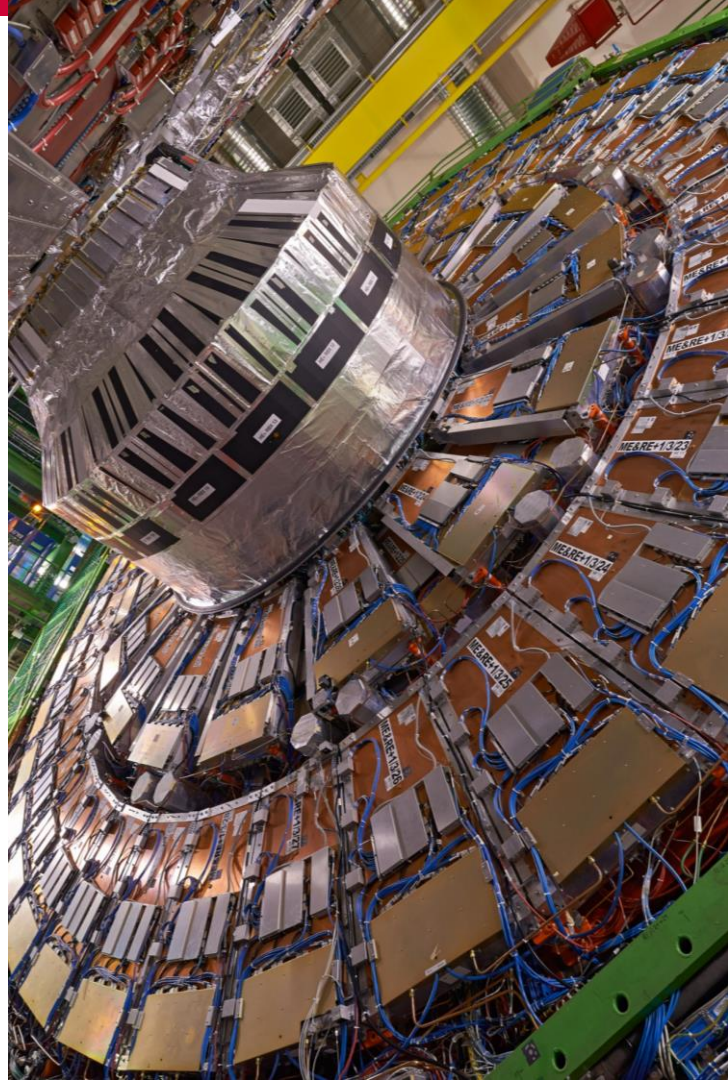
- No easy way around this; code was re-written
- PCs in PHYSX migrated to 2.1

Some code elements crashed on lab PCs

- Hard to pin down; some PCs worse than other
- Drive towards “safe” implementations
- Avoided these elements in assessments

Freezes, crashes, and bugs

- Very rarely show-stopping in practice
- Bugs reported to NI



Outcome

Student reception of new core module

Student module evaluation scores

Academic Year	LabVIEW Version	Score / 100
2018/19	**NXG 2.1**	**86**
2017/18	2015	94
2016/17	2015	90
2015/16	2015	94

Student feedback

- Overall very positive, all students felt they got something positive from the module (projects)
- Astrophysicists see the value of LabVIEW, but some would have preferred to develop their Python instead
- LabVIEW generally valued, especially on projects

Future MSc core module design (2019/20)

Scaling (>30) and broadening of MSc student cohort

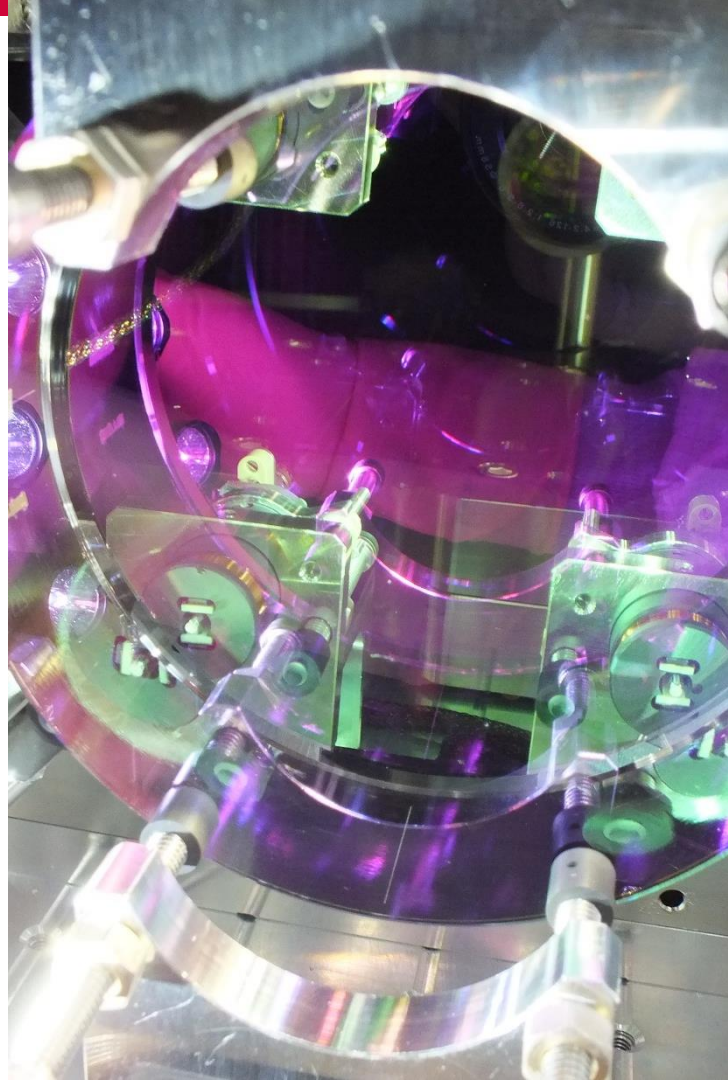
- MSc Gravitational Wave Physics comes online for 2019/20
- Additional micro-projects and relevant examples required
- Currently reviewing language teaching structure for 2019/20

Will we use LabVIEW NXG for 2019/20 or revert to 2015?

- Contingent on outcome of module review
- Will evaluate NXG 3.0 for stability improvements
- Now have full resources (inc. videos) for 2015 and NXG

Will we use NXG for the advanced programming module?

- Main problem is the lack of RIO support in NXG
- More difficult to migrate advanced materials to NXG
- Retain LabVIEW 2015 for now (considering 2018 SP1)



Summary: critique of LabVIEW NXG 2.1 for teaching and micro-projects

- **Usability:** excellent for micro-projects, more variable for teaching, but good.
- **Reliability:** NXG can be slow, crashy, and buggy. Often masks true potential.
- **Materials:** can stay in-project, so more efficient. Really need workbooks!
- **Student reception:** excellent for micro-projects, overall positive for teaching.

- **Overall:** NXG 2.1 has its issues, but the potential for brilliance is there.

Contact details

Dr Richard James Lewis

Director of Postgraduate Studies
School of Physics and Astronomy
Cardiff University

Tel: +44(0)29 2087 5433

Email: LewisR54@cardiff.ac.uk

URL: https://tinyurl.com/y94cfmk5

Case studies, presentations

Transitioning MSc Physics Teaching to LabVIEW NXG 2.0: From Drills to DAQ-First (NI AUF 2018)

Reflections on LabVIEW as a Common Language for Community and Skill Building (NIDays 2017)

LabVIEW as a Common Language for Community and Skill Building (NI AUF 2016, NIWeek 2017)

MSc Physics Students Take Ownership of their Learning with LabVIEW (NI EIA 2016)

Bringing the Research Group Ethos into Taught Masters Learning (VICE/PHEC 2016)