VVVV

Specialization Seminar – 99 Bottles of Beer

Institute of Computer Science
University of Innsbruck

Author: Stefan Spiss
Supervisor: Maria Schett

Overview

- Visual programming
- Data-flow programming
- History
- General Information
- Environment
- Capabilities & Features
- Comparison to Max and Pure Data
- Example programs
- Conclusion

Visual Languages

- Programming languages
  - at the highest level of abstraction
  - using pictorial information
- Two categories:
  - Program visualization
  - Visual programming languages (VPL)
- Examples: Max, Pure Data, vvvv

Data-flow Programming

- Data modeled along the data-flow graph:
  - Nodes are operations
  - Edges describe dataflow
- High parallelism
- Data-flow graph easy to visualize
- Examples: Max, Pure Data, vvvv

History

- 1997: MESO Digital Media System Design founded
- 1998: Development of vvvv started by MESO
- 2000: Graphical user interface
- 2002: First public release
- 2006: vvvv group founded

General Information

- Visual programming language
- Data-flow language
- Written in Borland Delphi
- Only available for Windows (DirectX used)
- Hybrid visual, textual environment

Environment

- Own environment for developing programs and run them
- Environment has one mode: runtime
- Using frames (max. 120fps)
- All operations calculated in every frame once

Environment - Patches

- Applications called patches
- Consist of:
  - Nodes
  - Links between nodes
- Patches are stored as xml files on the disk

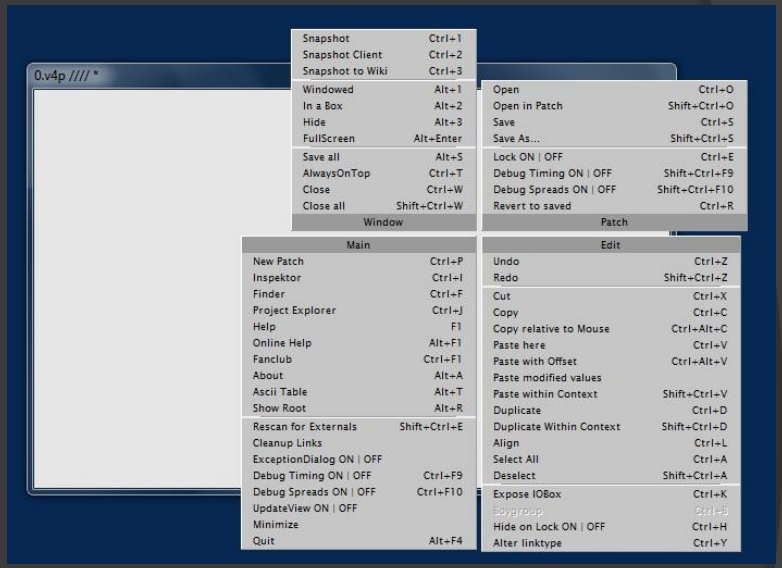

Figure 1: Empty patch with opened main menu

Environment - Nodes

- Present operations on the data
- Input and output pins
- NodeName(Category Version1 Version2 ... VersionN)

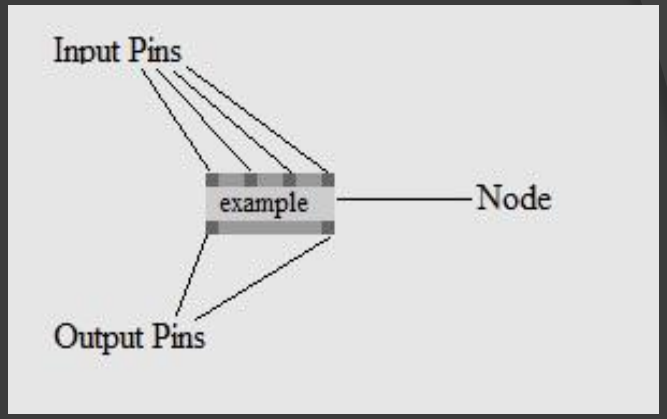

Figure 2: Example node after *[Fig.2]*

Environment - Node Category

- Polymorphism in vvvv
- Categories:
  - Value
  - String
  - Color
- Spread:
  - One-dimensional array of data
  - Slice is one element of a spread

Environment - Pins

- Every pin has its own type
- Types:
  - Numeric values
  - Subtypes
  - Strings
  - Color
- Only possible to connect:
  - pins with fitting types
  - input with output pins

Environment - Textual Languages

- vvvv-sdk:
  - Use to create own nodes
  - C#
  - Addonpack (user contributed nodes)
- HLSL (High Level Shading Language)

Capabilities & Features - 1

- Rendering engines for 2D and 3D animations
- Physics simulation engines
- Projection Mapping

Figure 3: Example for projection of textures to a 3D object *[Fig.3]*

Capabilities & Features - 2

- Boygrouping
- Multiple readers and writers for standard protocols
- OpenCV support
- Sound applications

Comparison to Max & Pure Data Similarities

- VPLs
- Data-flow programming
- Develop new functionalities using text-based languages
- Processing music and multimedia in real-time

Comparison to Max & Pure Data

- Advantages of vvvv:
  - Ability for parallel processing of objects much better (spreads)
  - More powerful in video and graphics applications
- Advantages of Max & Pure Data:
  - Features for sound processing are more powerful

99 Bottles of Beer v1

- Using two loops:
  - One for the counter
  - One to append the verses
- Loops only possible with FrameDelay node
- FrameDelay node:
  - Takes input from previous frame and puts it to the output
  - For 99 verses 99 frames are needed.
  - Needed time to generate song:
    1s/120fps * 99f = 0,825s

99 Bottles of Beer v2

- Using spreads
- I node generates numbers from 99 to 1
- Spread from I node used to build verses
- Appending all slices to get song text
- Can be done in one frame
  -> 1s/120fps = 0,00833s
  -> 1/99 of the time used for v1

Conclusion

- Easy to use prototyping and developing environment.
- Powerful tool:
  - fast in processing using spreads
  - easy to use, even for complicated tasks
- Usable for people without programming knowledge
- Might be confusing when used to common text-based programming

References

- M. Bohernitsan and M. S. Downes, Visual programming languages: A survey, 2006
- R. Navarro-Prieto and J. J. Canas, Are visual programming languages better? The role of imagery in program comprehension, 2001
- B. A. Myers, Taxonomies of visual programming and program visualization, 1990
- W. M. Johnston, J. Hanna and R. J. Millar, Advances in dataflow programming languages, 2004
- MESO, link: http://www.meso.net
- vvvv documentation, link: http://vvvv.org/documentation/documentation
- vvvv propaganda, link: http://vvvv.org/propaganda
- Wikipedia vvvv, link: http://de.wikipedia.org/wiki/Vvvv
- Wikipedia Max, link: http://en.wikipedia.org/wiki/Max_(software)
- Wikipedia Pure Data, link: http://en.wikipedia.org/wiki/Pure_Data
- [Fig.3] – link: http://vvvv.org/documentation/how-to-project-on-3d-geometry
- [Fig.2] – link: http://vvvv.org/documentation/tutorial-introduction