VPython Architecture

Bruce Sherwood
Visiting Scholar, Department of Physics
University of North Texas

Ruth Chabay
Professor, Departments of Physics and Chemistry
University of North Texas
Empowering Nonexpert Programmers

• VPython: Python programming language plus 3D graphics
• Novice programmers can create navigable real-time 3D animations
• Main users: students, educators, researchers
Examples of VPython Programs

- A complex program can run in multiple environments
- A simple program
  - Well-designed defaults
  - Vector computations
  - 3D animations as side effects of computations
- Can embed in web pages
GlowScript VPython

Browser

RapydScript: Python to JavaScript

GlowScript graphics library

WebGL

web page

Server (glowscript.org)

Datastore

This architecture is also used by trinket.io
VPython in Python Installations

Local server
- Python
- vpython module
- HTTP server
- websocket server
- other Python modules
- user program

Browser
- JavaScript process
- WebGL
- GlowScript graphics library
- web page
VPython with Jupyter Notebook

Local server
- Python
- vpython module
- Jupyter browser communication
- other Python modules
- user program

Browser
- JavaScript process
- WebGL
- GlowScript graphics library
- notebook
- Jupyter browser communication
GlowScript VPython: Compilation Details

User code (Python) → Pre-processing → RapydScript-NG

JavaScript → Operator overloading → Post-processing → Executable
Operator Overloading: How Vector Addition Works

- Using the PaperScript library with the Acorn parser of JavaScript, convert
  \[ a + b \rightarrow a['+'](b) \]

- JavaScript permits changing the behavior even of built-in classes such as Number and String
  
  ```javascript
  String.prototype['+'] = function(r) { return this + r }
  ```

  ```javascript
  Number.prototype['+'] = function(r) {
    return (r instanceof vec) ? add_error() : this + r
  }
  ```

  ```javascript
  vec.prototype['+'] = function(r) {
    return (r instanceof vec) ?
      new vec(this.x + v.x, this.y + v.y, this.z + v.z) add_error() : this + r
  }
  ```
GlowScript Rendering of 3D images

About 60 times/sec:

Send object data to WebGL

GPU “vertex shaders” -> GPU rasterizer -> GPU “fragment shaders”

Web page
Animation Loop

- rate(200): no more than 200 loop iterations/s
- About 60 renders/s
- Sleep for remaining time
- Assigning to an object attribute (pos, size, etc.) sets a "changed" flag for that object, and at render time its current attributes are repackaged to send to GPU
- Five 4-vectors: pos, axis, size, color, up, with texture, opacity, shininess, emissive packed into 4th slots; total of 80 bytes per object instance
Object Models in GPU Memory

- A “model” box object is stored in GPU memory
- Represented by 12 triangles, each described by 3 vertex objects specifying position, normal, color, and texture coordinates
- Data for a particular box (an instance of the box class) plus model information is sufficient for the GPU to display that box appropriately in 3D
- GPU memory has models of a box, sphere, cylinder, cone, and pyramid; compounds treated like primitives
- Arbitrary objects built from triangles; extrusions, 3D text
Speed Issues

• Python is an interpreted language and so execution is significantly slower than compiled languages.
• Computationally intensive GlowScript VPython programs run about an order of magnitude faster than VPython 7 programs, because they are compiled to (fast) JavaScript (but there is no access to Python modules).
Additional Technical Details

• Portions of objects hidden behind other objects are not seen thanks to “z-depth” blocking by GPU hardware
• Transparency handled by “depth peeling” algorithm
• Mouse “picking” uses false colors
Major Contributors to
GlowScript VPython and VPython 7

• David Scherer: originator of VPython; major contributions to the start of the GlowScript project
• John Coady: originator of Jupyter VPython
• Matt Craig: installers for VPython 7
Brief History

• 2000: Classic VPython created by David Scherer, an undergraduate student at Carnegie Mellon University, in collaboration with Chabay and Sherwood
• 2011 GlowScript begun by Scherer and Sherwood
• 2014 GlowScript VPython by Sherwood
• 2015 Jupyter VPython begun by John Coady
• 2016 VPython 7: Jupyter VPython made consistent with GlowScript VPython by Chabay and Sherwood, in collaboration with Coady; Classic VPython no longer supported

See brucesherwood.net for a detailed history
For More Information

- **vpython.org** – obtaining and using VPython
- **glowscript.org** – full VPython documentation, many examples
- **trinket.io** – embed both editing and execution of VPython in your own web page
- **matterandinteractions.org** – calculus-based contemporary intro physics curriculum in which VPython plays an important role
- **matterandinteractions.org/student** – includes a large number of physics demo programs written in VPython