Introducing Processing 2.0

Andres Colubri
Fathom Information Design
Boston, MA, USA
andres@fathom.info

Ben Fry
Fathom Information Design
Boston, MA, USA
ben@fathom.info

1. Introduction

The Processing programming language and environment [RF07] is an open source project initiated by Casey Reas and Ben Fry at the MIT back in 2001, and it is a widely adopted tool in the current practice of creative coding and computer arts and design. Processing has the "Design By Numbers" project [MA01], developed at the MIT Media Lab by John Maeda, as one of its predecessors. DBN's main motivation was to introduce visual designers and artists to computational design. Processing continues with this initial goal of broadening "computer literacy" by making programming more accessible for creators who do not necessarily have a formal training in CS such as artists, designers, and hobbyists in general.

After a long alpha stage, Processing 1.0 was finally released in 2008. With an increased focus on 3D and video performance, the version 2.0 has been in development during the past two years, and recently reached the beta stage.

2. Exposition

Creative coders have been witnessing the increasing importance of mobile devices, the need for real-time data visualization, and the use of the web as an interaction platform. Processing 2.0 addresses these issues with the inclusion of new 3D and video libraries, as well as with a javascript mode which allows to run Processing applications directly into the browser.

Processing's new 3D library incorporates an entirely overhauled OpenGL-based renderer, called P3D, designed to handle large geometries at interactive frame-rates, and to ease the cross-development on desktop and mobile platforms. P3D works in conjunction with the new video library, based on the GStreamer multimedia toolkit, in order to allow playback and capture of HD media.

2.1 Elaboration

P3D has been implemented with OpenGL following the ES 2.0 specification, and its main features are the following:

- Two rendering modes: immediate and retained. The first is geared towards quick creation of dynamic geometry, while the second is optimized to render large static models.
- Efficient use of GPU resources in a transparent manner: automatic tessellation of complex shapes, geometry batching to optimize data transfers between CPU and GPU memories.
- Portability and feature parity across various platforms (PC, Mac, Android).
- Modification of the rendering pipeline with custom GLSL shaders.

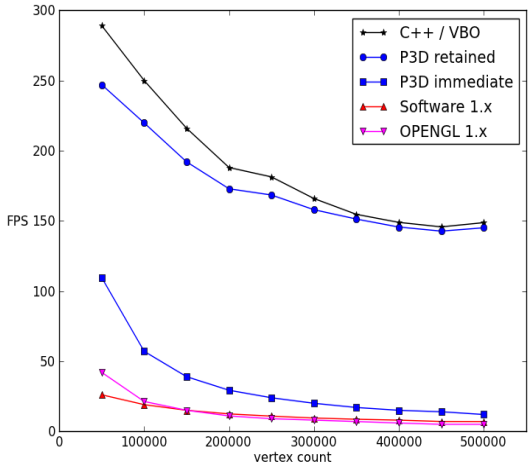

Figure 1. *Benchmark comparing OpenGL and software rendering in Processing 1.x, P3D in Processing 2.0, and C++ with VBOs.*

3. Results

API-wise, the new 3D renderer in Processing 2.0 is fully backwards compatible with 1.x, while offering substantial performance gains for most rendering operations. Compared with the previous OpenGL renderer in Processing 1.5.1, the speed-up can even reach 20X for very complex models. Performance is similar to that of equivalent C++ code in many situations. The chart in figure 1 summarizes some benchmarks comparing frame-rate as a function of the number of vertices in the scene.

4. Conclusions

Processing 2.0 is built on top of the current strengths of the language - simple API, library architecture, wide adoption in the design and arts communities – while it extends its range of applicability by offering higher-performance graphics and video and better web support. In particular, the new 3D renderer covers a wider range of usage scenarios: immediate mode for quick prototyping, retained mode for rendering complex models, and a customizable shader pipeline for more advanced applications.

References

[MA01] MAEDA J., ANTONELLI P.: *Design By Numbers*. The MIT Press, 2001.

[RF07] REAS C., FRY B.: *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press, 2007

SIGGRAPH 2012, Los Angeles, California, August 5 – 9, 2012.
ISBN 978-1-4503-1435-0/12/0008