Getting Started with Open-Source Software Development

Patrick Cozzi
@pjcozzi
pcozzi@agi.com
ComSpOC: http://apps.agi.com/SatelliteViewer/

3D Buildings: http://cesiumjs.org/NewYork/


NORAD Tracks Santa: http://cesiumjs.org/demos/noradtrackssanta.html
Today’s Topics

• Open-Source Communities
• Why contribute?
• How to Contribute
• Open Standards
• Licenses and CLAs
• Advice for Students

• Not technical, more social

• Based on my experience with Cesium
  – (admittedly geospatial biased)
What is Open Source?

• Access to the source code, and more:
  – http://opensource.org/docs/osd

• Free? Beer, Kitten, Freedom
Open-Source Communities
Some projects have just one founder (and even one contributor – and maybe one user); others have several founders or are lead by a group called a steering committee.

With Cesium, I picked several tools without really knowing anything about JavaScript: ant (build), Jasmine (test), jsdoc-toolkit (doc). Some decisions were good, somewhere less good.

A young project on the rise will offer you the best opportunity to make significant impact.
Just because you work for a company, doesn’t mean you are immediately a committer.

Committers are usually experts in one or more areas of the code, and are able to provide guidance and merge pull requests in that area.

In Cesium, we release monthly, and each month a different committer (albeit an AGI employee as of now) does the release.
With GitHub, it is so easy to contribute, that committer vs. contributor matters less than it use to.

A better measure of a project’s activity than total number of contributors is the number of contributors in the past 30 days. For example, on 11/05/2015, Cesium has 62 contributors, but only 25 in the past 30 days (still quite a high ratio).
In the very early days of Cesium, I focused too much on “grow the contributor community”, when I should have focused on “grow the user community”, because the contributor community comes largely from the user community who need bug fixes, believe in the project, etc.
Why Contribute?
The #1 reason developers outside of the core team contribute to Cesium is because they need a feature or bug fix.

Examples:
- New formats: TMS, URL template provider.
- Optimize: model cache, triangulation.
- Good open-source citizen: Cesium 1.0, doc fixes.
- GSoC, etc.: KML, GPX, CZML

The pain/cost/time of maintaining a fork is reason enough to contribute to a project.

Developers probably also want the experience and to learn, but I haven’t seen this much in my experience.
What is a Contribution?

- Code, of course. What else?
- Documentation and Tutorials
- Tests (well, this is also code)
- Example code (code too)
- Sample data
- Success stories and evangelism

Sample data, e.g., 3D models for testing glTF.

It is awesome – and more credible – when someone not involved in Cesium gives a talk at a conference on the benefits of Cesium.
Why Contribute?

(for students)
Fill a Real Need in a Visible Way

- Advice from Rich Geldreich (Unity, formerly Valve and Microsoft)
  - Find (or create) an open source project that game devs use, and contribute to it in a very visible way. For example, we [Valve] recently hired the author/maintainer of the SDL library. Google actively recruits devs who write libraries they use. Intel hires open source driver devs that contribute to MESA.
  - Identify a need and fill it with an open source project (that uses a license liberal enough that devs can actually use it). For example, right now a lot of teams are trying to target OpenGL ES, OpenGL, and D3D9/11 and are struggling with how to write a single set of shaders that can be translated/compiled to all these targets.
34 responses from CIS 565 alumni over six semesters (5 of which Patrick taught). The course is taught about once a year and has ~20 students. For more details, see: https://github.com/pjcozzi/Articles/raw/master/SIGGRAPH/2015/Preparing-Students-for-Industry-Using-Open-Source-and-GitHub.pdf

Keep in mind, it didn’t hurt the students who said No, and it helped 51.5%. I expect this to only increase as recruiting practices continue to change.

Several students also already had jobs before starting the course or received full-time positions based on their internships.
This quote is representative of many responses:

• The project portfolio is useful: demos, videos
• Employers generally don’t look at the code (or the student doesn’t realize it)
• The interview itself is what matters

I have my personal website, and I have the github link on it. Employers are interested in the projects, and we had a great discussion. It's actually not about the coding part. I think code portfolio is not that popular when interviewing a candidate. The key part is still about the interview itself.”
Being asked about your own code is a great way to stack the odds in your favor.

I expect more companies to ask for code samples; as a hiring manager, I do.
How to Contribute
Most external contributions to Cesium are a feature/bug a user needs

It needs to be small because
1) You will be more likely to get it right
2) The committers will be more likely to find time to review it

One forum at most – funnel the traffic to one place

Big project – ask on the forum
Small project – ask the maintainer
Follow the standards even if you don’t like them.

Ask too few questions, and you might waste time. Ask too many questions, and you might stop getting answers.
How to Contribute to a Project

• Open a pull request
  – Sign the CLA (more on this soon)
  – Link to the issue and provide a **concise** description
Some projects will just merge without much of a code review.

Be prompt – keep the momentum so you don’t lose the committer’s interest (or vice versa if it is your project).

Given that the project now has to maintain your code, they may not want some code changes, e.g., a 10% performance gain at the cost of significant complexity.
Open Standards
I don’t know this would happen.

When I started Cesium, it was all about code, and about implementing existing standards for widely used formats.
Now Cesium is giving birth to many open standards and formats.
See How I got involved in glTF and Khronos:
It would be a stretch to say that I now spend more time working on standards than code, but I spend way more time writing specs than I would have predicted when Cesium first started.
Interop with Open Standards

- Interop between different software
  - Open or closed source
- No vendor lock-in. Why?
  - Compete on a better product, service, price, ...
  - Stay ahead and innovate
Licenses and CLAs
MIT – include copyright/terms, no warranty
BSD 3-clause - include copyright/terms, no warranty, no endorsement
Apache 2.0 – include copyright/terms, no warranty, patent grant, explicit no trademark grant
GPL – copyleft

Also Creative Commons (popular for text and images): http://creativecommons.org/
Licenses Tips

• Avoid creating a new license. Why?
• Include a LICENSE.md file in your project
• Know the licenses of third-party libraries. Why?
Contributor License Agreements (CLA)

• Terms under which IP is contributed to a project.
• Used by Google (Chrome, Android, ...), Facebook, Open-Source Foundations, and all major projects I know of
• Protects you, the project, and the users
• Example terms:
  – “The project can use this code; I’m allowed to contribute it; I keep the copyright; I provide a patent grant”
• Apache Foundation CLA
  – Corporate: https://www.apache.org/licenses/cla-corporate.txt
  – Individual: https://www.apache.org/licenses/cla.txt

Google - https://cla.developers.google.com/clas
Facebook - https://code.facebook.com/cla

Every project should have a CLA
Today alone (11/06/2015), we received two CLAs.

**CLAs**

- Are CLAs a barrier to entry?
  - In Cesium, CLAs were hard to get signed at first
    - Big companies took months
  - Now that Cesium is established, CLAs come in all the time
Developer Certificate Of Origin (DCO)

- Signed-off commits
- Used by the Linux kernel
- We tried it in Cesium. No one used it

The code could have came from elsewhere.

It might be the contributor’s employers code.
CLA Tips

• Convince (um, ask) your employer before you sign
• Use them in your project
  – CONTRIBUTING.md
  – Kindly ask contributors to sign

https://github.com/AnalyticalGraphicsInc/cesium/blob/master/CONTRIBUTING.md
Advice for Students
Advice for Students

- Contribute to a young rising project that interests you
  - Start with a small contribution
  - Consider a project run by a company you want to join
  - Engage the contributors
- Write useful open-source tools
- Have a strong GitHub presence
  - Profile page
  - README.md for each project
    - [https://github.com/picozzi/Articles/blob/master/CISSS65/GitHubRepo/README.md](https://github.com/picozzi/Articles/blob/master/CISSS65/GitHubRepo/README.md)
  - Cross-reference your resume, LinkedIn, GitHub, etc.
  - GitHub alone is not the new resume

Tools – for example, I made wetzel open-source (not that it is popular):

The contributors at the company will take note. They are more likely to try to hire you as a student, then if you were a full-time employee they were trying to convince to change jobs.

A resume is still useful to a hiring manager because it very quickly gives them a picture of the highlights of your work.
Advice for Students

• Producing Open Source Software
  – http://www.producingoss.com

• Pro Git
Trivia – With Giveaways!

• Two reasons why to contribute to open-source?
• Two more reasons?
• A benefit of open standards?
• A difference between the MIT and BSD 3-clause licenses?
• A difference between MIT and Apache 2.0?
• Two reasons a CLA is important?
Thank You!

Projects I am involved in:

- Cesium - http://cesiumjs.org
- glTF - https://www.khronos.org/gltf
- CIS 565
- CIS 399

AGI is hiring for the Cesium team
Full time and internships
http://cesiumjs.org/jobs

Contact me
Patrick Cozzi
@pjcozzi
pcozzi@agi.com