

SIGGRAPH2015
Xroads of Discovery

The 42nd International Conference and Exhibition
on Computer Graphics and Interactive Techniques



Penn
UNIVERSITY OF PENNSYLVANIA

Preparing Students for Industry Using Open Source and GitHub

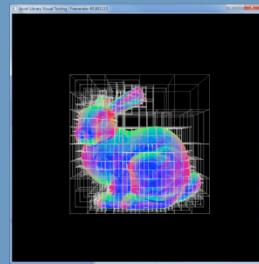
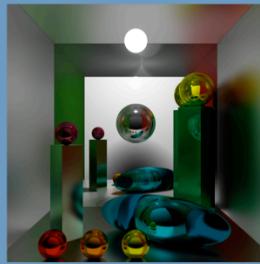
Harmony Li and Patrick Cozzi
University of Pennsylvania

Harmony: <http://harmonymli.com/>

Patrick: <http://www.seas.upenn.edu/~pcozzi/>

Background

- CIS 565: GPU Programming and Architecture
- Pragmatic, project-based, industry-focused
- CUDA, OpenGL, WebGL



Today we are going to talk about our experience with using GitHub and open-source projects in CIS 565 at the University of Pennsylvania. CIS 565 is a graduate-level graphics course on using the GPU for both massively parallel compute algorithms and graphics applications. Most students are second-year master's students who have already taken at least one intro graphics course. Usually, a few undergrads and sometimes a PhD student also take the course.

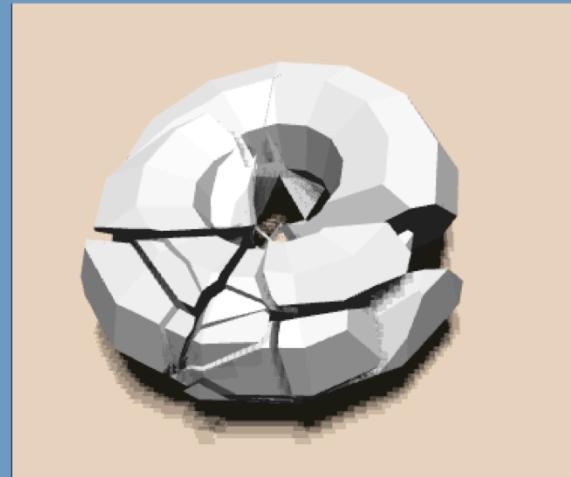
CIS 565 is project-focused. Patrick stopped giving exams and pure written assignments after his second semester teaching. The mindset for CIS 565 is a “a series of projects with supporting lectures.” It’s kind of like the class is a sports team, and the instructor is a coach preparing them for games.

Student work from left to right: GPU-Accelerated Dynamic Fracture in the Browser with WebCL by Kai Ninomiya and Jiatong He; Bidirectional Path Tracer in CUDA by Paula Huelin Merino and Robbie Cassidy; Uniform grid and kd-tree in CUDA by Jeremy Newlin and Danny Rerucha.

Student Projects

GPU Accelerated Dynamic Fracture in the Browser with WebCL

Kai Ninomiya & Jiatong He
(Fall 2014)



Over the years, a variety of students have taken the course. Most of our students are usually graphics or robotics majors. Spurred by their own interests, they have produced diverse projects, some of which have been well-received by the community.

GPU Accelerated Dynamic Fracture in the Browser with WebCL

Kai Ninomiya & Jiatong He

Fall 2014

Github: <https://github.com/kainino0x/cis565final>

Live Demo (requires Nokia WebCL Firefox plugin): <https://kainino0x.github.io/cis565final/src/>

Video <https://www.youtube.com/watch?v=13sbZia4Kjc>

Student Projects

Tile-Based WebGL Deferred Shader

Yuqin Shao & Sijie Tian
(Fall 2013)



Tile-Based WebGL Deferred Shader

By Yuqin Shao & Sijie Tian

Fall 2013

Github: https://github.com/YuqinShao/Tile_Based_WebGL_DeferredShader

Live Demo: <http://sijietian.com//WebGL/deferredshading/index.html>

Video: <https://vimeo.com/83616115>

Article: <https://hacks.mozilla.org/2014/01/webgl-deferred-shading/>

It's good to mention that these two students went on to collaborate with Patrick to write an article about their findings. Their Github repo has been starred and viewed by many users since.

Student Projects

A WebGL Point Cloud Renderer

Nathan Marshak & Uriah Baalke
(Fall 2013)



A WebGL Point Cloud Renderer

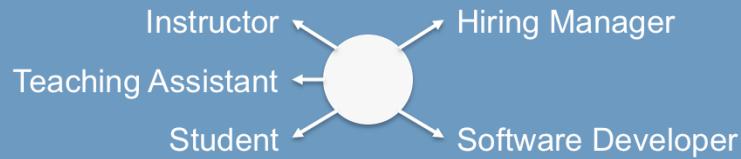
By Nathan Marshak & Uriah Baalke

Fall 2013

Github: <https://github.com/nmarshak1337/PointMan-Renderer>

Slides: https://docs.google.com/presentation/d/1toQ5c5Mw_8D_hWouQqrb1SKOvAJjrxdfCHs503GID20/edit

Our Perspectives



We try to look at using GitHub and open-source from many different angles.
Between the two of us, we cover many perspectives:

- Harmony was a student in the course before TAing it, and is now in industry.
- Patrick is the instructor, but works in industry full-time as a hiring manager and (open-source) software developer.

For the software developer perspective, there are two angles: (1) providing students with useful skills to become software developers, and (2) preparing the students for technical interviews with software developers.

Motivation

- A posting for an AMD internship in 2012:

“The profile for the candidates: C++, OpenGL, OpenCL. Applications with a source code and for the principle of it, a resume.”

<http://www.g-truc.net/post-0461.html>

This is just one example that the recruiting world is changing. Attending a prestigious university, earning a high GPA, and creating a website with descriptions of their projects is no longer enough to stand out. Employers want to see code.

AMD internship posting by Christophe Riccio: <http://www.g-truc.net/post-0461.html>

Motivation

- Advice from Rich Geldreich (Boss Fight Entertainment, ex-Valve)
 - 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**.

Motivation

- Advice from Rich Geldreich
 - **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.
 - **Become an authority** on some topic and blog about it in a **very visible/useful way**.

Common themes in Rich's advice:

- Find a real need
- Be visible

Mechanics

- GitHub student accounts
 - Free
 - 5 free private repositories

A large, bold, black GitHub logo centered on a blue background.

As per the title of our talk, we use Github as the main platform to distribute our base code and to view and grade our students' submissions.

Students typically get Github student accounts, which are free to use and have an unlimited number of public repositories. Github student accounts remain active for 4 years. For private projects, these accounts have up to 5 free private repos.

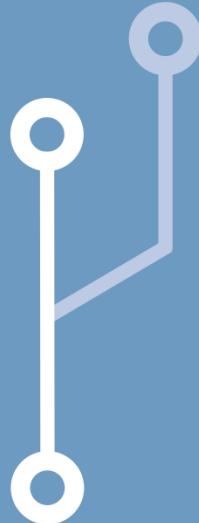
Miscellaneous:

- Github students accounts are provided via Github Education, which has also partnered with other companies to provide the Github Student Developer Pack
- Contains other useful software for shipping and developing for free (i.e. Atom, Bitnami, Visual Studio, Unreal Engine 4, etc.)
- Github Student Developer Pack : <https://education.github.com/>

Image Credit : Github logo, Github.com

Mechanics

- Starter code repo
 - Students fork repo
 - Pull request for fixes



- We maintain our own Github organization, which is a Github account that allows for multiple users to author and manage repositories.
- Our organization has start code repositories, which we push and make public when our homework goes “live”.
- Students then fork the starter code base repository so that they have a copy of the repository on their own Github accounts.
- Since we try to keep as up-to-date as possible, the HW changes from year to year. This makes it easy for us to version our projects, and patch fixes to our starter code base even when students are working on the project. (Students can rebase their fork).
- Similarly, students can also provide bug fixes by submitting pull requests for fixes, which we can manage and merge on will.

Image Credit : Github Fork A Repo logo from Github help

Mechanics

- Pull requests to submit work
- Diffs to show student work

- Instead of packaging up their work and submitting offline, we have the students submit their work via pull requests.
- The pull request can then be viewed as a diff between the starter code base and the student's code base, making it easy to see what students have submitted.

Benefits



- Learning version control
- Understanding software engineering practices

Students benefit a fair amount from using github:

- Learn version control
 - Version control is a pervasive thing in software engineering companies
 - Many companies are looking into Git or other distributed versioning systems
 - Skills are transferrable to centralized versioning systems (svn, p4, etc.)
 - Understanding of the motivations behind these tools and gaining good habits early
- Understanding software engineering practices
 - Beyond version control systems, students are exposed to how to organize their code and convey their ideas
 - How to maintain code when working together
 - How to document code for others to see
 - How to maintain boundaries between third party and individual code
- Image Credit: Github Octocat, Github.com

Benefits

- Public portfolio of work
 - Personal Website (github.io)
 - Personal Blog (Jekyll)



Beyond technical skills, they gain an easy and strong platform to launch their resume:

- Personal Websites
 - Many employers ask for personal websites, either to see projects or to get to know the candidate better
 - Github.io is an easy way to host a static website through Github by simply using a Github repository
 - Also has the ability to host individual project pages
 - No longer any need for dedicated server, etc. for web-based demos
 - Can mostly be hosted through Github
 - Personal Blogs
 - Blogs form the crutch of technical community
 - Github makes it easy to host a personal static webpage blog via Jekyll
 - More documentation beyond READMEs to display more granular posts about progress in project
 - Very good way to showcase deeper understanding of many things that cannot be explained (and should not be explained) in a README
- Image Credit : Jekyll logo, github.com/jekyll/brand

Academic Integrity

- Similar to research
- Every project is different
- In-class presentations
- Graphics community
- READMEs

We treat academic integrity similar to research – you need to cite your sources (e.g., third-party code), and clearly identify your contribution.

Academic integrity can be an issue when all of your projects are open source

-We manage this by doing random in-class presentations.

- Students are notified that we will be doing presentations a particular day (typically a week after the project is done)
- Students are randomly called to give presentation on their work and their findings (no slides, just using the README on the repo)
- Field questions from the class and from the instructors about bugs, potential fixes, performance analysis, etc.
- Based on their presentation, you can get a general idea of whether or not they did their own work and if they had an understanding of the project

-Graphics community

- Since everyone can see these repositories, it is hard to hide if you have plagiarized
- Patrick typically tweets these projects. So industry members come to look at their code.
- The community helps in keeping the students in check.

-READMEs

- Github supports Markdown, so READMEs are easily accessible and readable via the Github repository landing

README

- Important in gauging student understanding
- Informal write-ups

These READMEs serve two purposes: gauging student understanding, and relaying project information to audiences as an informal write-up.

Informal write-up of their work:

- Introduction of what they have done
- Results
- Discussion
- Performance Analysis
 - Since some students have never done something as in depth, we guide them through a series of questions to start them off on performance analysis
- To enforce the quality of these write-ups, a portion (15%) of the grade is usually linked to the quality of the write-up
- Best way to understand how much a student has gleaned from this project (and whether or not the project is driving a point home) is by seeing the depth of the READMEs
- READMEs also act as a representative compilation of some of the work the student has done
 - Good, polished READMEs will go a long way
 - Offer more tips for READMEs later

README Tips

- Provide context and contact info
- Have a representative image



Accelerated Stochastic
Progressive Photon
Mapping on the GPU by
Ishaan Singh, Yingting
Xiao, and Xiaoyan Zhu
(Fall 2013)

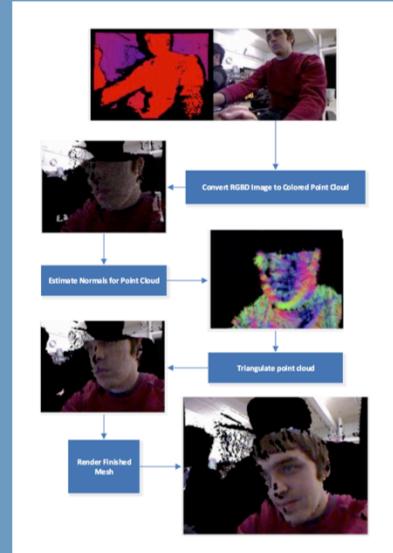
The README.md file in the root directory is displayed when we visit a GitHub repo. We encourage students to make this as compelling as possible to showcase their work.

Screenshot: Accelerated Stochastic Progressive Photon Mapping on the GPU by Ishaan Singh, Yingting Xiao, and Xiaoyan Zhu, <https://github.com/ishaan13/PhotonMapper>

README Tips

- Use screenshots with captions to walk through features

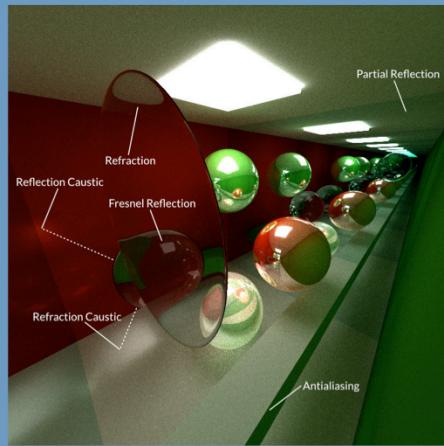
Surface Mesh
Reconstruction from RGBD
by Collin Boots and Dalton
Banks (Fall 2013)



Screenshot: Surface Mesh Reconstruction from RGBD by Collin Boots and Dalton Banks, <https://github.com/cboots/RGBD-to-Mesh>

README Tips

- Annotate

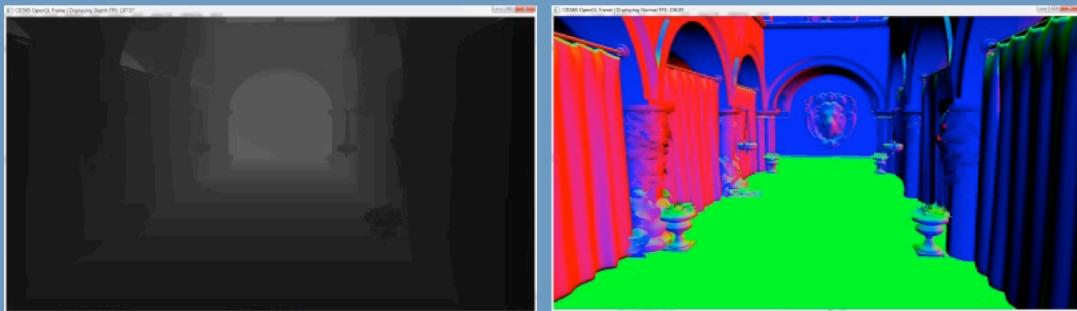


GPU Path Tracer by Kai Ninomiya (Fall 2014)

Screenshot: GPU path tracer by Kai Ninomiya, <https://github.com/kainino0x/Project3-Pathtracer>

README Tips

- Debug screenshots

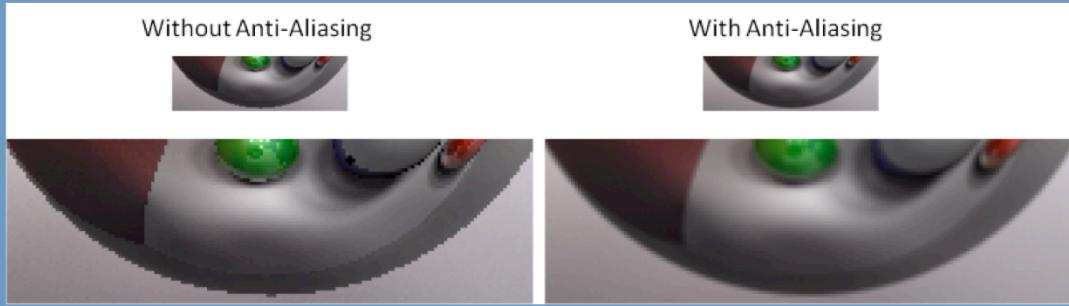


Deferred Shader by Cheng-Tso Lin (Fall 2013)

Screenshot: Deferred Shader by Cheng-Tso Lin, <https://github.com/otaku690/GLSLDeferredShading>

README Tips

- Side-by-side comparisons



GPU Path Tracer by Wei-Chien Tu (Fall 2014)

Screenshot: GPU Path Tracer by Wei-Chien Tu, <https://github.com/foxking0416/Project3-Pathtracer>

README Tips

- Bloopers
- Videos
- Demos
- Performance analysis
- Build instructions
- Focus on what is unique
- Note shortcomings
- Discuss future work and potential solutions to bugs

Twitter

The image displays three separate Twitter posts from the user @pjcozzi. The first post shows a low-poly rendering of a cracked sphere. The second post shows a screenshot of an ocean simulation in Unity with white-capped waves. The third post discusses a rasterization pipeline in CUDA. Each post includes a small profile picture of Patrick Cozzi, the tweet text, a preview image, and standard Twitter interaction metrics like retweets and favorites.

Exciting for students to have their work noticed outside of class and the university.

Sometimes, someone famous will notice.

Open source isn't necessary for twitter, but it helps drive engagement.

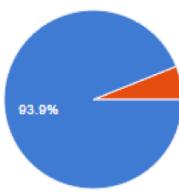
- Dynamic geometry fractures with WebCL and WebGL by Kai Ninomiya and Jiatong He, <https://twitter.com/pjcozzi/status/542364663166824450>
- Ocean simulation in Unity by Anda Li and Yehua Lyu, <https://twitter.com/pjcozzi/status/542007352925642753>
- Rasterization pipeline by Cheng-Tso Lin, <https://twitter.com/pjcozzi/status/396613045553623040>

Alumni Survey Results

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.

Survey Results - GitHub

Did using GitHub for class projects help prepare you for your full-time position?



Yes 31 93.9%
No 2 6.1%

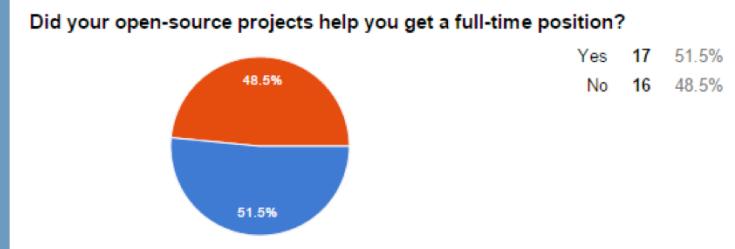
- Why didn't it help?
 - “I was familiar with git/github already. My employer uses centralized version control (Perforce).”
 - “I am not working in a full-time position yet.”

Survey Results - GitHub

- Why did it help?
 - “Github is **used extensively** in the company I work for now. They also listed git as **desired skill** when recruiting for this position.”

Most responses were similar to this. Companies are using git or GitHub, and are looking for students familiar with it.

Survey Results – Open Source



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.

Survey Results – Open Source

- “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**.”

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

Survey Results – Open Source

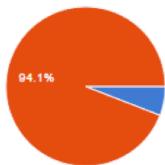
- “One of the companies I interviewed specifically **went through my code samples** on git and asked me questions based on that”
- “A lot of companies are **asking for my code sample** on git.”

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.

Survey Results – Academic Integrity

Are you aware of any academic integrity violations that took advantage of open-source projects from the current or previous semesters?



Yes	2	5.9%
No	32	94.1%

Survey Results – Academic Integrity

- Based on your experience, do you think open-source projects increase the chance of academic integrity violations?
 - “It is possible but in my experience for such specialized courses like GPU programming, **most students are in it to win it.**”
 - “I do not. If anything it makes them much **more blatant** and therefore high-risk.”
 - “If students want to share code, they can **regardless of policies**”
 - “I also think having rendered images and README files publicly viewable online is great motivation for current students who want to **one-up students from previous years.**”

Conclusion

- GitHub: adopt it as early as possible in coursework
- Open source projects: use them if it makes sense for your course

Aside: Hackathons



Tiju Thomas presents his hack (Fall 2012)

<http://blog.virtualglobebook.com/2012/12/hackathons-in-classroom.html>

Aside: Industry Speakers

- Bring excitement and experience to the classroom
 - AGI, AMD, ARM, ArrayFire, Autodesk, Floored, Google, IBM, Marmoset, Philadelphia Game Lab, Sandia National Laboratories, Terathon
- Always ask speaks to provide career advice

Topics have included terrain rendering, OpenGL drivers, mobile architecture, GPU profiling and debugging, teaching MOOC, deferred rendering, WebGL, GPU search, Image-Based Lighting, shadows, GPU ray tracing, and game engine design.

Resources

- Free GitHub educator's accounts:
https://education.github.com/guide/private_repos
- How to make an attractive GitHub repo:
<https://github.com/pjcozzi/Articles/tree/master/CIS565/GitHubRepo>
- CIS 565 course website:
<http://seas.upenn.edu/~cis565/>