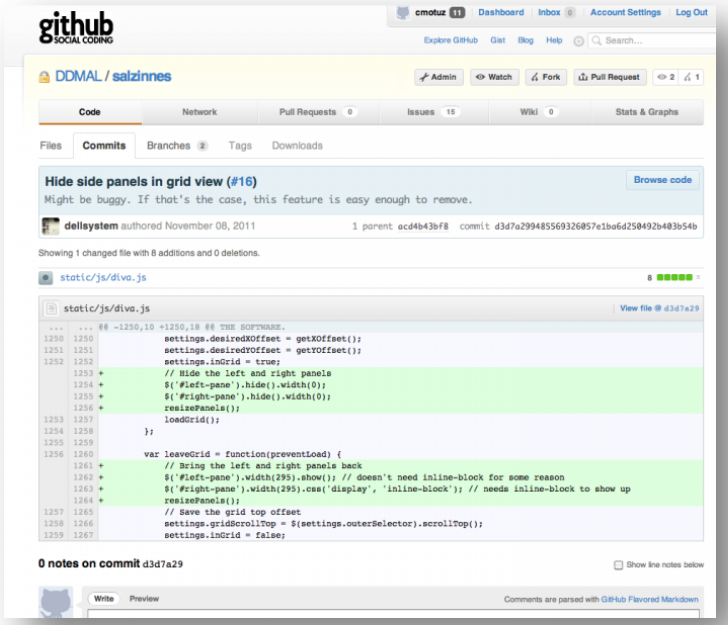
# What's The Fuss About GSoC?

Pararth Shah
CSE - IIT Bombay

April16, 2012
Talk organized by **WNCC-IITB**

# Yeh Open Source kya hota hai?

And why does your life depend on it?


github
SOCIAL CODING
cmotuz 11
Dashboard
Inbox 0
Account Settings
Log Out
Explore GitHub
Gist
Blog
Help
Search...
DDMAL / salzinnes
Admin
Watch
Fork
Pull Request
2
1
Code
Network
Pull Requests 0
Issues 15
Wiki 0
Stats & Graphs
Files
Commits
Branches 2
Tags
Downloads
Hide side panels in grid view (#16)
Might be buggy. If that's the case, this feature is easy enough to remove.
Browse code
dellsystem authored November 08, 2011
1 parent acd4b43bf8
commit d3d7a2994855693260857e1ba6d250492b403b54b
Showing 1 changed file with 8 additions and 0 deletions.
static/js/diva.js
8
static/js/diva.js
View file @ d3d7a29
@@ -1250,10 +1250,18 @@ THE SOFTWARE.
settings.desiredXOffset = getXOffset();
settings.desiredYOffset = getYOffset();
settings.inGrid = true;
// Hide the left and right panels
$('#left-pane').hide().width(0);
$('#right-pane').hide().width(0);
resizePanels();
loadGrid();
};
var leaveGrid = function(preventLoad) {
// Bring the left and right panels back
$('#left-pane').width(295).show(); // doesn't need inline-block for some reason
$('#right-pane').width(295).css('display', 'inline-block'); // needs inline-block to show up
resizePanels();
// Save the grid top offset
settings.gridScrollTop = $(settings.outerSelector).scrollTop();
settings.inGrid = false;
0 notes on commit d3d7a29
Show line notes below
Write
Preview
Comments are parsed with GitHub Flavored Markdown

# Open Source is everywhere

# Philosophy behind GSoC

Why does Google hold it every year?

# Long Term Benefits

- Increase open source participation among students
- Help open source software projects in finding interested and capable developers
- Promote social coding among student community

# Short Term Benefits

- Provide budding developers with exposure to industry-level code
- Help selected FOSS projects get closer to their next release
- Forge new relationships between companies and universities all around the world

# What's in it for you?

Benefits Of GSoC

$ 5000


Freebies

Google
SUMMER OF
CODE
2011
CODE.GOOGLE.COM/SOC

pcl

More Freebies

# The Obvious Benefits

- Money – USD 5000
- T Shirt – and other freebies
- Resume point
- Bragging rights

Money, Fame, Respect

<

Fun, Experience, Contacts

# The More Subtle Benefits

- Exposure to Open Source development
- Working with large software projects – around million lines of code!
- Working in a team environment – team distributed across the globe
- Working on code with real-world applications
- Making contacts in the developer community
- Learning from experienced developers, getting to work along them on important projects

# What I Learnt

#include<insights.h>

# Valuable Insights

- How a large code-base is managed
- How to work with a distributed team – assigning tasks, reporting bugs, communication over mailing list/IRC
- Good programming techniques practiced by developers in the industry
- Time management, proper anticipation of time required to complete a task, taking responsibility for bugs
- Improvement in email communication skills

# Tips For Future GSoCers

Statutory Warning: Blindly following instructions is injurious to health

# Generic Tips

- Choose a project wisely – Popular organizations like Ubuntu, Firefox, Python, etc get lots of applications, which means lower chances of selection
- Choose a project which matches your skills and interests
- Expect to spend at least 15-20 hours per week on your project
- Ask for help when stuck: most of the people in open source are friendly and enthusiastic to help
- Be polite and patient on mailing lists

# Basic Expectations From GSoC Applicants

- You can install and configure software packages on your own
- You have access to a functioning computer 40 hours a week
- You've got experience using the programming language and operating system of the project
- If applying for a project where you don’t meet the technical requirements, mention in the application how you would work on acquiring those skills before/during the work period

# Important Soft Skills

- You find out where to go for help with technical questions
- You take and respond well to feedback
- You can work independently
- You know when to ask questions
- You can communicate your thoughts effectively via email/mailing-lists/blog-posts

# IIT-Bombay Specific Tips

- There have been very few applications from IIT-Bombay: Lack of exposure and not talent!
- **If you have paid attention in your CS101 lectures, you have enough programming knowledge to be eligible for at least 50% of the projects**
- Do take time out for GSoC if you are generally free this summer
- A GSoC project is possibly equivalent to a semester long course in Software Engineering in terms of experience gained
- The experience counts for a lot in interviews for future internships/employment
- Most importantly: Don't take up a project and plan to ditch it halfway – You'll spoil yours as well as the Institute's reputation!!

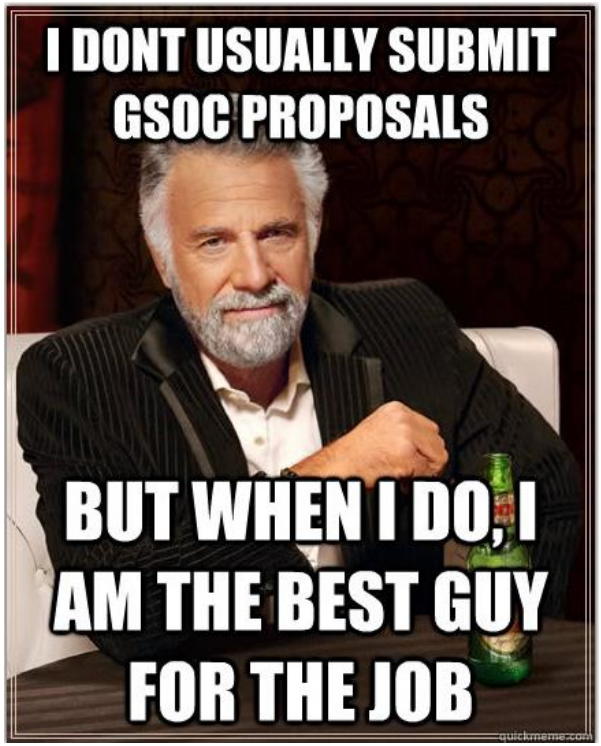

ONE DOES NOT SIMPLY
WALK AWAY FROM A FANTASTIC OPPORTUNITY
quickmeme.com

# Useful Links

**GSOC Website:**
http://www.google-melange.com
**GSOC 2012 Timeline:**
http://www.google-melange.com/gsoc/events/google/gsoc2013
**GSOC Student Guide:**
http://www.booki.cc/gsocstudentguide/
**GSOC General Mailing List:**
http://groups.google.com/group/google-summer-of-code-discuss
**List of Mentoring Organizations in 2013:**
http://www.googlemelange.com/gsoc/accepted_orgs/google/gsoc2013
**Google for "GSOC experience"** to get first-hand accounts of students @ GSOC with various projects
**Look for "GSOC" pages on websites of the open-source projects** you are interested in, to see what kind of projects they will accept

# Getting Selected

… the toughest part of the GSoC experience


✗ Bad Idea

✓ Good Idea

# Step 1:
# Select Mentoring Organization

- All types of FOSS projects participating in GSoC: Covering most programming languages, platforms, applications
- Need to choose according to your interests, skills, and mentor's expectations
- Very important to make the right choice, else possibility of not getting selected
- Study list of participating organizations carefully, look at their previous GSoC projects

# What I Did:

- Looked at organizations close to my **interests** (Machine Learning, Computer Graphics, Parallel Computation) and **skills** (C++, Java, Desktop GUI Apps, Algorithms)
- Shortlisted three projects:
  - Apache Mahout: Library for ML algorithms
  - VideoLAN: VLC media player
  - Point Cloud Library: Library for processing 3D point-cloud data (eg. from Kinect camera)
- Studied what they are expecting from GSoC

# Step 2: Formulate Project Proposal

- Find out what the mentoring organization expects from a GSoC project:
  - Impact on project community
  - Student skill level
  - Student participation
- Look at previous projects
- Look at project ideas posted on their websites
- Look at mailing list archives (very useful!!)
- Think hard about an interesting and feasible project idea

# What I Did:

- All three organizations that I shortlisted had a definitive list of project ideas that they would like to be done in the GSoC
- Studied their source code, understood what they are expecting to be done, what would be possible for me to do
- Found a couple of ideas suiting my interests and skills in both Apache Mahout and PCL
- Sketched out a rough proposal and project timeline for those ideas

# Step 3:
## Communicate With Project Community

- VERY IMPORTANT!!!
- Before proposal submission deadline, get feedback from the mentoring organization community on feasibility and usefulness of your proposal
- Contact them through email/mailing list
- They will give constructive feedback
- Make appropriate changes to your proposal
- Also helps your chances if you introduce yourself to the community in this way

# What I Did:

- Introduced myself on the mailing lists, gave a rough idea of my proposal
- Got responses from a couple of developers
- Spent time installing Apache Mahout and PCL on my laptop, tried out basic features
- Sent in a bug-fix for a (very) minor issue in PCL: Helped in introducing myself to the community, proved that I have a basic understanding of their software

# Step 4:
# Finalize Proposal And Submit

- Make sure you are sufficiently aware of the mentoring organization's source code, expectations, and work ethic
- Incorporate all suggestions given by their community
- Make submission on Google website, as well as on mentoring organization mailing list
- IMPORTANT: Plan to do it at least a day before deadline, Google is very strict with submission deadlines, no extensions for any reason!!

# What I Did:

- Finally went ahead only with Point Cloud Library project proposal
- Made necessary improvements, added rough timeline, and finalized proposal
- Submitted it on GSoC website
- After few days, got email from PCL mentor asking if I would like to work on a slightly different project
- I replied positively… got acceptance email from Google a few days later!!

# Working On The Project

Get me to the code!!

# Point Cloud Library

- My mentoring organization
- The Point Cloud Library (**PCL**) is a standalone, large scale, open project for 3D point cloud processing
- Set of algorithms written in C++, for performing complex tasks on 3D image data, like that obtained from a Kinect camera


http://pointclouds.org

# My Project:
# Point Cloud Registration in PCL

- Goal: To build an automated benchmarking framework for evaluation of Point Cloud feature extraction and registration algorithms present in PCL
- Details:
  **http://pointclouds.org/blog/gsoc/pararthshah/index.php**
- Had to work with C++ templated classes, to build a generic module that can run a variety of algorithms on different datasets, and output statistics

# Work Schedule

- Worked with my mentor on dividing the project into phases
- Spent time understanding the existing code, since I had to build upon it
- Got a developer account and SSH access to the PCL SVN codebase
- Got a blog set up where I had to post updates on my work
- Spent a minimum of 15 hours each week working on the project
- Spent time documenting my code and writing a tutorial for future users of my code

# Overall Experience

- Coding was in C++, which I was well-versed in, so it was convenient
- Learnt a lot of stuff about templates, overloading, using libraries like Boost, Makefiles, building, executing and debugging code, and much more
- Mentor was very friendly, helped in improving my code design to make it more efficient
- Spent time on IRC/mailing list talking with the PCL developer community to solve issues in my code
- Also, seeing how other GSoC students were tackling their projects was a learning experience

# Project-Specific Insights

- OOP class design tricks
- C++ Templates
- Quickly understanding how a large software is organized: Figuring out integration points for my own code
- Working with svn/ssh – Uses and misuses of versioning systems
- Properly documenting code
- Rigorous testing

# Questions?

"Ask not what Open Source can do for you,
Ask what you can do for Open Source!"

Gracias!