Google Summer of Code 2019 Proposal for Random Points Generation in boost.Geometry

Personal Details

- ➢ **Name**: Mudit Agarwal
- ➢ **College/University**: International Institute of Information Technology, Hyderabad, India
- ➢ **Course/Major**: Computer Science and Engineering
- ➢ **Degree Program (B.Sc., M, Sc., PhD, etc.)**: B.Tech. + MS by Research
- ➢ **Email**: agarwalmudit9876@gmail.com
  mudit.agarwal@research.iiit.ac.in

Availability

- ➔ **How much time do you plan to spend on your GSoC?**
- ➔ I plan to devote about 7-8 hours per day on weekdays and around 3-4 hours on the weekends for the project which makes it around 40 - 45 hours per week. I am ready to increase the working hours on weekends to 7-8 hours as per the project demands. My 5th semester at college begins in August and I am sure I would be able to devote around 5 hours a day at that period of time. My preferred time of availability is 12:00 P.M - 12:00 A.M IST (UTC +05:30).

- ➔ **What are your intended start and end dates?**
- ➔ I have summer break from 8th May - 27th July so ideally I would want the project to start in the beginning of May and stretch till the end of July. But I'm fine with the given SoC timeline. I plan on utilising the month of May to get a head start in the project.

- ➔ **What other factors affect your availability (exams, courses, moving, work, etc.)?**
- ➔ As mentioned above, I will be probably at home for summer break from May 8 to July 28. Hence, I can assure that there won't be any kind of availability issues in this period. In case of any such circumstances, I'll inform the mentor well in advance and make up for it by working more before and after the period so as, not to disturb the proposed timeline. My college begins on the 1st of August and there'll be less coursework (about 10-15 hours a

week) in the first month, so I'll be able to put in sufficient time for the project. I will be available for contact via email, video chat through Skype and Google Hangout and instant messaging on Facebook. In case of any unexpected circumstance, I can always be contacted via email.

Background Information

Please summarize your educational background (degrees earned, courses taken, etc.).

Degrees Pursuing:

I am currently in my 4th Semester (2nd Year) pursuing **B.Tech and MS by Research(in Computer Science)** at International Institute of Information Technology Hyderabad, India.

Courses Taken:

(The courses are grouped by relevance):

- ➔ **Programming**: Algorithms and Data Structures, Computer Programming.
- ➔ **Systems courses**: Operating Systems, Computer System Organisation
- ➔ **Information Technology Workshop**: ITWS-1 (Bash, HTML,CSS, git), ITWS-2(Python, Javascript, Bootstrap, web2py)
- ➔ **Mathematics**: Discrete Mathematics, Introduction to Linear Algebra and Vector Spaces, Group Theory, Complex theory, Probability, Statistics.
- ➔ **Engineering Electives**: Structured Systems Architecture and Design(SSAD), Engineering Systems*****.
- ➔ **Electrical and Electronics**: Digital Logics and Processors, Electrical Science, Basic Electronic circuits, Digital Signal Analysis and Application*****.
- ➔ **Others**: Artificial Intelligence*****, Graphics*****, Computer Networks*****

  (*) : currently pursuing

Please summarize your programming background (OSS projects, internships, jobs, etc.).

I started programming when I was in high school, where I learnt the basics of C++ syntax, pointers, classes etc. I started algorithmic programming and learnt about data structures in C in my first year of my college. I have implemented wide range of graph algorithms and data structures like heaps, BST, AVL tree, Dijkstra's shortest path algorithm , range trees (segment tree and BIT/ Fenwick tree), persistent range trees and union/find data structure. I was very much interested in competitive coding also so I took part in several programming contests on online platforms like Codeforces and Codechef which motivated me to use C++. The problems which interest me the most are the related to geometry.

I have used C++ to implement a wide range of advanced algorithms like Mo's algorithm, Heavy Light Decomposition(HLD), Centroid Decomposition of a tree etc.

I have used the C++ graphics library(OpenGL3) to make 2D (Jetpack Joyride) and 3D (a basic version of a 3D fighter jet game) games. I've also implemented a bash-like terminal in C using syscalls and also able to successfully setup a working proxy server on my computer using socket programming in python which facilitate blacklisting IP Address, handle multiple client at a time, caching etc.

Please tell us a little about your programming interests. Please tell us why you are interested in contributing to the Boost C++ Libraries.

I like programming in C++, I use it for algorithmic programming and for writing multithreaded code as it has a very good Standard Template Library. But it does not offer a good support for geometric application, for simple geometric application like finding area of a polygon, convex hull, envelope etc., so every time when there is a need for any of these, I have to again write or copy paste or import file previous written code in the current file. I was introduced to Boost when I was searching for the library like STL in which all the functionalities are pre-implemented. I came to know about the Boost.Geometry and I was really fascinated by the features offered by it. I have also checked the github page of Boost.Geometry, where I found that it will also accept contributions from individuals. Then I thought that even I could help other people by contributing to Boost Libraries the way it helped me. So I approached a senior in college who had previously contributed to Boost. He also encouraged me to lookup the projects proposed under the GSoC program this year.

I found the project **'Implement algorithms for random point generation according to the given geometric distribution'** interesting and I really think I'm suitable for the project.

What is your interest in the project you are proposing?

I am extremely interested in Algorithms (especially related to geometry), Data structures and other optimisations one can achieve by using appropriate algorithm or data structures or carrying out minute changes in the code or by using other techniques. The process of designing new algorithm or improve current time complexity and memory usage of the algorithm fascinates me a lot. The project I'm proposing has a similar problem statement. It aims at designing an algorithm for generating random points inside a polygon according to the given geometric distribution and thus I am looking forward to enjoy it and learn a lot by working on it.

I also think this will be an appropriate platform to learn about contributing to big software and getting involved in the open source community. I would like to become a regular contributor in the Boost Community. I want to contribute to Boost.Geometry as it designs algorithm for geometrical purposes which aligns with my interests.

What are your plans beyond this Summer of Code time frame for your proposed work?
After completing the proposed work during the Summer of Code time, I would like to work on any extension of the project and also, I would really be interested in working further with the organization for extension of Boost.Geometry library in future.

Have you done any previous work in this area before or on similar projects?

1. I have implemented the competency test that has helped me get a better understanding about the C++ templates.
2. I have implemented various advanced algorithms like Heavy Light Decomposition and Mo's and data structures like Persistent Range Trees.
3. I have also worked on geometric algorithms like Convex Hull, calculating area of polygon etc. in some of my Course Assignments.

Please rate, from 0 to 5 (0 being no experience, 5 being expert), your knowledge of the following languages, technologies, or tools:

- **C++ 98/03 (traditional C++) :** 4
- **C++ 11/14 (modern C++) :** 3.5
- **C++ Standard Library :** 4
- **Boost C++ Libraries :** 2.5
- **Git :** 3.5

What software development environments are you most familiar with (Visual Studio, Eclipse, KDevelop, etc) ?
I am quite comfortable with VIM. Though I do prefer the niceties of Visual Studio sometimes and prefer to use it when too much of parallel work is involved.

What software documentation tool are you most familiar with (Doxygen, DocBook, Quickbook, etc) ?
I'm currently reading about **Doxygen documentation tool** (Boost Libraries are also documented in Doxygen) to document the Random Points Generation library which I've implemented as the

Competency test. Doxygen is a command line, open source documentation-generator suitable for use with C-style languages like C++, C, IDL, Java, and even C# or PHP. Aside from being open and non-proprietary, Doxygen distinguishes itself from vendor alternatives like JavaDoc by building in support for a wider range of document formats.

Proposal

Overview:

Aim:- To implement algorithms for random point generation according to some given geometric distribution for the Boost.Geometry library using Expression Templates and Meta-programming paradigm in C++. It requires studying the existing algorithm for generating random points inside polygon and modifying it to generate points according to the given geometric distribution and further optimising the algorithm.

Below are the descriptions of the algorithms that I have read and understood to solve the random point generation problem. I am planning to implementing this algorithms in this project but open to change or modify the algorithm as advices by the mentor.

Using Barycentric Coordinate:

Barycentric coordinates for triangles are also known as area coordinates or areal coordinates, because the any coordinates P with respect to ΔABC are equivalent to the (signed) ratios of the areas of PBC, PCA and PAB to the area of the reference ΔABC. Barycentric coordinate system is extremely useful for a wide range of applications. Here I try to exploit one of the application of barycentric coordinates to solve the random points generation problem. I don't know if it can be extended further as it is the first ever attempt to solve this type of problem using barycentric coordinates.

Consider a triangle ΔABC defined by its three vertices,. Each point P located inside this triangle can be written as a unique convex combination of the three vertices. In other words, for each P there is a unique sequence of three numbers, $\alpha, \beta, \gamma \geq 0$, such that

$$\alpha+\beta+\gamma = 1 \text{ and}$$
$$P = \alpha*A+ \beta*B + \gamma*C$$

By exploiting the above fact in the reverse, taking any three consecutive coordinate of the polygon and random values of α, β and γ which satisfy the above mentioned constrain, I am able to generate

random points inside all types of polygon. I plan to modify the above algorithm to incorporate geometric distribution.

Implementation Details

The entire project can be broken down into three parts:

1. **Checkpoint-1:** Implementation of the random points generating algorithm for convex polygon and extending them to incorporate all types of polygon.
2. **Checkpoint-2:** Modifying the algorithm to generate random points according to the given geometric distribution.
3. **Checkpoint-3:** Optimising algorithm implemented till now.

I have decided to follow the coding standard adopted by the organization as mentioned on their oficial site https://www.boost.org/doc/libs/1_53_0/libs/spirit/doc/html/spirit/notes/style_guide.html and also the concepts of Object Oriented Programming (OOPS),so that my code can fit well into Boost.Geometry library. I will also incorporate expression templates wherever required to make the code efficient by eliminating creation of temporaries and virtual function calls. This will result into a maintainable code and can be easily integrated into the current Boost.Geometry library.

Implementing Algorithm (Checkpoint-1)

There are a lot of efficient algorithm implemented to generate random point inside a polygon but not many efficiently extended to incorporate all types of polygon. So the main idea behind the algorithm is to efficiently extend it to incorporate all types of polygon. After implementation, carry out time and space complexity analysis and choosing the best a among them.

Adding Geometric Distribution (Checkpoint-2)

For generating the points according to the given geometric distribution basically I would create a module for They can be extended from the proposed version to accommodate for different variety of parameters out of the scope of the project. I intend to implement it in such a way that further extensions and expansions of the library are easy.

Optimization (Checkpoint-3)

After completing both the above mentioned check points, I will try to optimize the algorithm further, carrying out time complexity analysis and plotting and studying the behavior/performance of the algorithm for various types of polygon and geometric distribution.

Integration of algorithm into Boost.Geometry

I plan to do the integration after the completion of all my checkpoints.

Proposed Milestones and Schedule

Time Period	***Proposed Work***
Pre-Project Phase ***(Community bonding period)***	
April 9	Submit the final draft of proposal
April 9- April 15	● Go through the documentation of Doxygen and document the Competency Test. ● Brush up my current knowledge of various geometric distribution and go over algebraic geometry concepts. ● Brush up C++ skills and get familiar with the features of C++ 17.
April 15 - May 5	I have end semester examinations from 18th April and I would be fully committed to them during this period
May 5 - May 27	Discuss the possible solution of extending and modifying currently available algorithm to generate random points inside the convex polygon to incorporate geometric distribution and all types of polygon with the community and draft solution and make a report of all the changes need to be done.
Project Phase	
Phase 1	
May 27 - June 3 **(Week 1)**	● Start working on the finalized Algorithms, from through discussion with mentors, for generating random points for convex polygon and later extend it further.
June 3 - June 10 **(Week 2)**	● Extend the implemented algorithm to incorporate all types of polygon
June 10 - June 17 **(Week 3)**	● Code cleanup ● Documentation for implemented algorithm ● Multiple review iterations

	Checkpoint 1 competed
June 17 - June 24 **(Week 4)**	● Start discussing the implementation details to find the best way to incorporate the geometric distribution functionality with the mentor. ● Start modifying algorithm to incorporate geometric distribution.
June 24 - July 1 **(Week 5)**	● Continue the implementing the algorithm
Phase 2	
July 1 - July 8 **(Week 6)**	● Code cleanup ● Extensive Unit testing ● Multiple review iterations **Checkpoint 2 competed**
July 8 - July 15 **(Week 7)**	● Start working on test cases for the implemented algorithms. ● Generate and test, set of test cases on the algorithms.
July 15 - July 22 **(Week 8)**	● Studying behavior of the algorithm for different types of polygon and different types of geometric distribution ● Analysis of time and space complexity of the algorithm ● Discussing results with mentor. ● Discussing possible solution with mentor on how to optimize run time, memory usage and various aspect of the algorithm
July 22 - July 29 **(Week 9)**	● Implementing suggestion given by the mentor ● Again carrying out analysis of time and space complexity and studying the behavior of the algorithm ● Sharing the my report with the mentor **Checkpoint 3 competed**
Final Evaluation phase	
July 29 - August 5 **(Week 10)**	● Start documentation of the implemented algorithms.
August 5 - August 12 **(Week 11)**	● Start working on **integration** of implemented algorithms with Boost.Geometry

August 12 - August 19 (Week 12)	• Complete the **integration Process** of the implemented algorithms with Boost.Geometry library • Complete the **Full Documentation** of Algorithms.
August 19 - August 26 (Final Week)	• Buffer Time • Submit Final Code and Evaluations
August 26 - September 26	• Mentors submit final student evaluations

This is the timeline I plan to stick but open to change the timeline according to mentor.

Programming Competency

- Here's the link to my programming competency test: https://github.com/Mudit-1999/Random_Points_Generation

    - ❖ I plan to extend and optimise this algorithm further to incorporate geometric distribution as of now it generate points for any types of given polygon.
    - ❖ Open to change or modify it further as suggested by mentor.

- **Other C/ C++ Projects:**

    - **JetPack JoyRide:** Implemented a slight variant of Mario using OpenGL 3 library, in which the player while dodging firelines,magnet,fireball, dragan has to collect coins. The Player can also shoot enemy using water balloon. https://github.com/Mudit-1999/JetPack-JoyRide

    - **Fighter Jet:** Designed and implemented a Fighter Jet Game in 3D using OpenGL 3 library, in which the player has to destroy all the enemy base while maximizing his score by shooting parachute, passing through ring. https://github.com/Mudit-1999/Fighter-Jet

    - **Algorithms Library:** I implemented some standard graph algorithms and data structures using C++(STL)

    - **TMBSH:** Implemented a basic shell in C which supports piping, redirection, forking and other features of bash shell using syscalls. https://github.com/Mudit-1999/TMBSH

- **Other Projects:**

    - **Extreme Tic-Tac-Toe bot:** AI bot for Xtreme Tic Tac Toe (a slight variant of ultimate tic tac toe with 2 big boards) using alpha beta pruning, winning heuristic and quiescence search. https://github.com/Mudit-1999/Xtreme-TicTacToe-Bot

    - **Quiz Portal:** Designed a fully functional Quiz Portal using Go and React JS Implemented Admin Interface which involves adding/deleting quizzes/questions. https://github.com/Mudit-1999/Quiz-Portal

    - **SubWay Surfer:** I tried to write shader and lighting module of a basic version of Subway Surfer using webgl. https://github.com/Mudit-1999/SubWay-Surfer

    - **Basic Ticket Booking:** Built a working prototype of Ticket Booking App (like Bookmyshow) with admin interface in Python. https://github.com/Mudit-1999/BlackBox

Thanking Note to organisation

I would like to thank all the members of Boost organization, to give me this opportunity to write proposal and providing possibility to work with Boost.
I will look forward for every feedback from organisation members reviewing this document and would be glad to discuss/change accordingly.