GSoC 2020 Proposal

Extend Convex Hull and Centroid Algorithm to non-cartesian coordinate systems

Personal Details:-

Name: Aditya Mohan

College / University: BITS Pilani Goa, India

Course / Major: Computer Science

Degree Program: Bachelor of Engineering (B.E.)

Email: adimohan29@gmail.com / spwafer@gmail.com

Github homepage: https://github.com/spacewafer

Availability:-

I will be available for 40 hours a week during the main coding period. My intended start dates and end dates coincide exactly with the GSoC timeline, i.e. 18th May - 17th August 2020. I would be available throughout the day to work for the project. However, due to recent concerns of the Covid ‘19, I would be willing to work from any time prior to the proposed dates which can be decided upon after further consultation with the mentors.

Background Information:-

Educational Background:

I’m currently in my third year of undergraduate studies pursuing a degree in Computer Science in my university. With respect to my university I have taken the following courses which might be relevant to this project:-

- Discrete Structures in Computer Science
- Data Structures and Algorithms
- Design and Analysis of Algorithms
- Object-Oriented Programming
- Database Management Systems
- Principles of Programming Languages
- Theory Of Computation
- Compiler Construction

Apart from these, I have taken online courses in **Computer Graphics** and **Object-Oriented Programming in Java**. I had **interned** in the summer of 2018 at a Government Institute (Central Leather Research Institute) and worked on Image Classification using Machine Learning.

The reason the only organization I opted for for GSoC '20 was **Boost C++ libraries** and started contributing to it early on is because, C++ is the language I'm the most comfortable with by far and would love to know more about the intricacies involved in the language and its libraries. I'm passionate about Computer Graphics and Augmented Reality (and have attended workshops on it) and would like to work in these fields in the future. This suggests my interest in the field of Geometry and improving my knowledge in implementing and analysing these geometric algorithms would certainly be beneficial. Apart from that, my experience with competitive coding has contributed to my knowledge in C++, algorithms and data structures like the Convex Hull.

Beyond the Summer Of Code time frame, I plan to remain an active contributor in this organization, researching new geometric algorithms, improving documentation like I've done in the past and pitching in new ideas and implementation techniques.

Ratings:-

1. **C++ 98/03 (traditional C++): 4/5 -** Have worked with this in the past and would feel comfortable using it again, if at all it is required.
2. **C++ 11/14 (modern C++): 4/5 -** Have worked with this for various projects and coding challenges.
3. **C++ Standard Library: 3/5 -** Although I haven't used this extensively in the past (but I have used some of its functions), I'm positive with some research and practice, I'll get up to date with these libraries.

4. **Boost C++ Libraries: 3.5/5 -** I had not worked with Boost a lot before and simply heard of its usability and had come across some tutorials on the same. However, after working on the same for the past three months, I'm confident my skills in using these have certainly improved.
5. **Git: 3/5 -** Have worked with Git briefly in the past, but my command on the tech would certainly improve when I continue working on the project.

I currently use the Sublime Text Editor for working on these projects. However, I do have some experience working with Visual Studio and Eclipse IDE before.

So far, I'm the most comfortable with Doxygen as I have successfully generated documentations using the same for my PRs that have been merged.

Project Proposal:-

Introduction:-

This proposal is based on the Project 2 - Extending the convex hull algorithm and centroid algorithm for non-cartesian coordinates. The coordinate systems I'd like to extend this too are the Polar Coordinate System and the Geographic Coordinate System.

Apart from this, the mentors have revealed that this extension isn't currently available for the Centroid algorithm as well, so I'd like to extend the centroid algorithms to incorporate inputs to the Polar and the Geographic Coordinate system as well.

Finally, I'd like to resolve any issues pertaining to the 3D convex hull algorithm as well. Towards the end, I would round it off by finishing adding examples to the arithmetic operations in Boost.Geometry for the sake of a wholesome work product and it can prove to be beneficial to users of the same as well.

Subproject 1: Extending existing strategy (Graham-Andrew) for the Convex Hull Algorithm to incorporate inputs in the Polar Coordinate System:-

The existing Graham-Andrew Strategy can be extended to the polar/spherical coordinates systems through the following rough version of the algorithm:-

1. Initially sort all the points in increasing order of their polar angle and obtain the left-most point.

2. Iterating through each of these points, make each of these points as the reference point and calculate angle covered by the reference point, the previous point in the convex hull and the next sorted point.
3. If the angle made thus is clockwise positive (it doesn't turn left) then the next sorted point is included in the hull and the reference point is excluded. Else the reference point isn't excluded.

These three steps are iterated till we reach the rightmost point. This generates the top half of the hull. In a similar fashion, we proceed to create the bottom half of the convex hull.

Subproject 2: Extending existing strategy (Graham-Andrew) for the Convex Hull Algorithm to incorporate inputs in the Geographic Coordinate System:-

There are several problems with extending this to the spherical case:

- All of the points must be in the same hemisphere for a definition of "polygon" to make sense. No problem, we can just require that of our input, run a test, and stop execution if this requirement is violated.
- The center of longitude/latitude points on a map is *not* mean(lng),mean(lat). A better definition of "center" is had by taking the average of every point's (x,y,z) coordinates and projecting that point back to the surface.
- The cross product calculation of two vectors on the sphere's surface is different.
- Data has to be normalized when adding/subtracting near the poles and the international dateline.

Hence, the best way to go about it is to project it to 2D:-

Given Geographic Coordinates, we can find the convex hull of the given points with their latitude and longitude values provided they lie on the same hemisphere. After running the hemisphere test, to ascertain that they lie on the same side, these points are then projected to 2D and the existing algorithm is implemented to find the convex hull.

Subproject 3: Extending existing strategy for the Centroid Algorithm to incorporate inputs in the Polar Coordinate System:-

Finding the centroid of a geometry given in Polar Coordinates has no hard and fast algorithm. So, I tend to follow an approach similar to the one followed in the cartesian system. The following are the steps:-

1. For each point (ri, θi) on the vertex of the polygon, calculate ri*cos(θi) and ri*sin(θi).
2. Obtain the summation of ri*cos(θi) and ri*sin(θi) and divide each of this by n (the total number of points given)
3. Calculate the new r by taking the pythagorean sum of the values obtained above sqrt((r * (sin(θ'))² + (r * (cos(θ')²))
4. Calculate the new polar angle θ by obtaining the value of arctan(sin(θ') / cos(θ'))

   The new coordinate described by (r, θ) is the required centroid.

Subproject 4: Extending existing strategy for the Centroid Algorithm to incorporate inputs in the Geographic Coordinate System:-

For this again, there is no dedicated algorithm pertaining to the specific coordinate system, hence the algorithm to be followed is pretty straightforward. Given the inputs in (lat, lon) in radians:-

1. Calculate cos(lat) * cos(lon), cos(lat) * sin(lon) and sin(lat) for each of the given points in (lat, lon)
2. Obtain the average by dividing the summation of these three by n, (the total number of points given) and let these points be x,y and z.
3. The new lat, lon are described by the following equations:-
   - Lon = atan(y, x)
   - Hyp = sqrt(x * x + y * y)
   - Lat = atan2(z, hyp)

   The new coordinate described by (Lat, Lon) is the required centroid.

These algorithms are open to be tweaked given any suggestions by mentors and can do with slight adjustments during the course of more extensive and dedicated research

over the GSoC period. These algorithms will likely be adopted as different strategies and the strategy to be executed will be resolved at run time.

Proposed Milestones and Schedule:-

May 4th - June 1st: Community Bonding Period

- ➢ Will get up to speed with mentors about exact implementation strategies, solidify understanding the codebase better and going through other algorithms in the Boost Geometry library to get a strong hold on how to go about implementing or adding a new strategy to existing algorithms.
- ➢ I plan to stay active on Gitter and Boost.Geometry mailing list throughout this period and beyond.
- ➢ Will look for additional algorithms to improve choice of selection of strategies and will understand how to implement them.
- ➢ Last, but certainly not the least - establishing a healthy relation with the community.

PHASE I:-

Week 1 (June 1st - June 7th)

- ➢ Work on implementing Subproject 1: Implementing **Polar Coordinate System** strategy in the **Convex Hull** algorithm.
- ➢ Will continue research on the existing algorithm and will browse for existing and correct implementations for the same and adapt accordingly.
- ➢ Will successfully implement the new strategy in the existing algorithm after getting the approval from the mentors.

Week 2 (June 7st - June 14th)

- ➢ Write test cases for the new implementation, track bugs if any and resolve them.
- ➢ Create examples in documentation for better readability for users.
- ➢ Create documentation successfully and verify locally for the same.

Week 3 (June 14th - June 21st)

- ➢ Work on implementing Subproject 2: Implementing **Geographic Coordinate System** strategy in the **Convex Hull** algorithm
- ➢ Will continue research on the existing algorithm and will browse for existing and correct implementations for the same and adapt accordingly.

- ➢ Will successfully implement the new strategy in the existing algorithm after getting the approval from the mentors.

Week 4 (June 21st - July 3rd)

- ➢ Write test cases for the new implementation, track bugs if any and resolve them.
- ➢ Create examples in documentation for better readability for users.
- ➢ Create documentation successfully and verify locally for the same.

Phase I completed: Successfully extended the Convex Hull algorithm to the new coordinate systems - Polar and Geographic Coordinates.

PHASE II:-

Week 5 (July 3rd - July 10th)

- ➢ Work on implementing Subproject 3: Implementing **Polar Coordinate System** strategy in the **Centroid** algorithm.
- ➢ Will continue research on the existing algorithm and will browse for existing and correct implementations for the same and adapt accordingly.
- ➢ Will successfully implement the new strategy in the existing algorithm after getting the approval from the mentors.

Week 6 (July 10th - July 17th)

- ➢ Write test cases for the new implementation, track bugs if any and resolve them.
- ➢ Create examples in documentation for better readability for users.
- ➢ Create documentation successfully and verify locally for the same.

Week 7 (July 17th - July 24th)

- ➢ Work on implementing Subproject 4: Implementing **Geographic Coordinate System** strategy in the **Centroid** algorithm
- ➢ Will continue research on the existing algorithm and will browse for existing and correct implementations for the same and adapt accordingly.
- ➢ Will successfully implement the new strategy in the existing algorithm after getting the approval from the mentors.

Week 8 (July 24th - July 31st)

- ➢ Write test cases for the new implementation, track bugs if any and resolve them.
- ➢ Create examples in documentation for better readability for users.
- ➢ Create documentation successfully and verify locally for the same.

Phase II completed: Successfully extended the Centroid algorithm to the new coordinate systems - Polar and Geographic Coordinates.

PHASE III:-

Week 9 (July 31st - August 6th)

- ➢ Research on better ways to extend the Convex Hull algorithm to **3-D Coordinates** and implement the **QuickHull Algorithm** to achieve the same.
- ➢ Write test cases for the new implementation, track bugs if any and resolve them.

Week 10 (August 6th - August 13th)

- ➢ Analyze the **current** implementation of incorporating 3-D inputs in the Convex Hull strategy.
- ➢ Track and resolve bugs if any and resolve them.
- ➢ Compare and Contrast the current algorithm with the Quick- Hull algorithm and decide on which one to go forward with or implement both simultaneously as different strategies if no major difference between the two is discovered.

Week 11 (August 13th - August 21st)

- ➢ Make final changes to documentation and incorporate suitable examples for the chosen strategy/strategies.
- ➢ Track and resolve bugs, if any.
- ➢ Verify if the documentation can be produced locally.

Final Week (August 21st - End of GSoC period)

- ➢ Continue improving documentation for the users by adding examples to the other arithmetic operations - Add, Subtract, Multiply, Divide and produce it locally.
- ➢ All PRs to Boost.Geometry repository are submitted.
- ➢ Fix Documentation and respond to PR feedback

Phase III completed: Successfully resolved issues of Convex Hull in 3D coordinates and improved documentation for the arithmetic operations.

End of GSoC Period.

Programming Competency Test:-

Link to the repository containing my competency test solution:-

https://github.com/spacewafer/Boost-Geometry_Competency_Test

Contributions to Boost::Geometry:-

- https://github.com/boostorg/geometry/pull/665 [Merged]
- https://github.com/boostorg/geometry/pull/660 [Merged]

Why Me:-

I've been an active member in Boost since the turn of the year and it was because I had come to learn about Boost library through my colleagues and with the help of some browsing it got me thoroughly convinced about its benefits and usability. Knowing that whatever contribution I make would benefit the world of C++ as a whole motivates me. I was patient in making my pull requests and worked hard to accomplish them. The process aided me in gaining a better understanding of the codebase and certainly motivated me to do more good for the community.

Acknowledgements:-

I sincerely thank the whole of the Boost Geometry community for helping me make contributions and for guiding me whenever I needed some help. I've been able to hone my skills in working in tandem with the community and it wouldn't have been possible without the guidance of Boost members especially - **Vissarion Fysikopoulos** , **Adam Wulkiewicz**, **Mateusz Loskot**. I understand that **Vissarion** and **Adam** would be my potential mentors and I look forward to working with them throughout the whole period and accomplishing the goals stated in the proposal.

Thank You!