Can We Find Equal Sides of a Heptagon?

New Mexico Adventures in
Supercomputing Challenge
Final Report
April 6, 2005

038
Melrose High school

Team Members

Dominick Cicero
Rita Cicero
Annie Shaw
Brian Raulie

Teachers

Mrs.Raulie
Mr. Daugherty

Project Mentor

Mr. Daugherty

Table of Contents

Executive Summary

The project that we chose for this year's super computing challenge is to find a perfect regular heptagon using a compass and ruler. We were told by one of our sponsors that a regular heptagon is made up of an imaginary number so there is not a true regular heptagon. We are going to find out the chord length of a heptagon to the 12^{th} decimal 51.42758142756. This is an approximation when 360 is divided by 7, the number continues indefinitely.

A heptagon is a polygon with seven equal sides and seven equal angles. We chose the Java program because it is a way to construct a perfect heptagon. If we can find the perfect Heptagon polygon the results could be used to write code for a stronger computer firewall program for online protection.

We found that a heptagon could not be constructed using only a straightedge. We thought that if we could construct a perfect square, triangle and pentagon, and plot the points within a circle, we could construct a regular heptagon. Our program would plot the points, check distances and then rotate and repeat until the 'perfect' distance would be calculated. It is computational in that the computer

will have to plot and check indefinitely until the expected results are completed.

Introduction

In our project we are trying to find a perfect seven-sided polygon (a heptagon). A heptagon is a polygon with seven sides and seven angles. A regular heptagon has seven equal sides and seven equal angles. 'Heptagon' is usually considered more correct than 'septagon'. (thesaurus.maths.org) We chose this project because no one has been able to find a perfect heptagon; only an estimated chord length has been found. If we can find the perfect polygon the results could be used to make a stronger firewall for computers.

We will be using Java to find and plot the x and y coordinate points of where the three basic shapes (an equilateral triangle, a square, and a pentagon) intersect the circle, these will be our reference points. Then the x and y points will be entered into a data array. We will compare the mathematical distance between two points to the distance of each side of a heptagon to see if we can find

the points of the perfect heptagon. If none of these distances work, we will shift our reference points and try this procedure again.

We are still working on our Math for our program. We have researched the history of the heptagon and the polygon. We are using Java and NetBeans to design a computer program to analyze the coordinate points that we have plotted.

Expected Results: We are in the process of finding the chords to a heptagon at least 2 times of what is known now. We are finding ways to use the heptagon differently.

We have also determined that this might not find a perfect heptagon because it is an imaginary number. We will try to find the closest procedure of how to find a heptagon using the standard geometric construction technique. We will use the standard geometric construction technique (compass and straight edge) to construct the heptagon.

Program

In our project we are going to use a program to draw a triangle, square, and a pentagon in a circle. Then we are going to find a heptagon with the geometric construction technique using these shapes. We're going to be writing a simpler way to do this program. This is the way we put our thoughts together to write this program.

Circle

This is the main part of the program; this must put everything together. It has a boundary for the shape. The shapes must stay in side of it. We are going to have this program find the original point in the middle of a circle. It tells everything else where the middle is and relays information.

Drawing a Triangle

To draw a triangle we are going to find the origin reference Angle + 0 degrees, and go to 10 units. It will place a point. It will calculate the coordinates, and record this into an array. It will find the origin reference angle + 120 degrees and go 10 units. Make a point. It will calculate the coordinates then record this into an array. Then find the origin reference angle + 240 degrees and go 10 units. Make

a point. Then it will calculate the coordinates then record this into an array. It will draw a line from the three points to make a triangle.

Drawing a Square

To draw a square we must first find the origin reference Angle + 0 degrees, and go to 10 units. Place a point. It will calculate the coordinates then record this into an array. Find the origin reference angle + 90 degrees, go 10 units. Make a point. It will calculate the coordinates then record this into an array. Find the origin reference angle + 180 degrees, go 10 units. Make a point. It will calculate the coordinates then record this into an array. Find the origin reference angle + 270 degrees, go 10 units. Make a point. It will calculate the coordinates then record this into an array. It will draw a line from the four points to make a square.

Drawing a pentagon

To draw a pentagon we must find the origin reference Angle + 0 degrees, and go 10 units. Place a point. It will calculate the coordinates then record this into an array. Find the origin reference angle + 72 degrees, go 10 units. Make a point. It will calculate the

coordinates then record this into an array. Find the origin reference angle + 144 degrees, go 10 units. Make a point. It will calculate the coordinates then record this into an array. Find the origin reference angle + 216 degrees, go 10 units. Make a point. It will calculate the coordinates then record this into an array. Find the origin reference angle + 288 degrees, go 10 units. Make a point. It will calculate the coordinates then record this into an array. It will draw a line from the five points to make a pentagon.

Heptagon check

The program will calculate the distance to all the points of the array. Compare the distance it has found to the heptagon length. Then if there are no distances that match to the heptagon the program will start over.

Changing Reference Angle

If the last step of heptagon doesn't work then the ref. angle will change. Find the next point in line, in the array. From this point then start over.

Program Point

The program Point must remember some things. It needs to know what shape it came from. It will put this information into a folder called array shapes. Point also must know were it is on an x and y plot of the circle. Point need to be able to tell things its x and y plot.

Array

The array will have to put the new and old points in an order by size. It will do this by using the original reference angle at 0 degrees and seeing how big each point is and where it fits. Then it will place the points in size order.

Results

We have not yet completed our math in java. We are still working on the math in java and we will attempt to update our paper when we have finished. We hope to have something constructed by the EXPO date. We have found that when working with math, although math is supposed to be an exact science, many times it is not. Approximations are more common than we realized. We expected to be able to calculate an ‘exact’ number, but the more we worked on it the more we realized that there is NO exact number for a seven sided figure. That is why it has never been done before and is one of ‘math’s mysteries’. We would like to pursue this further, because we think that we will be able to find a terminal number, although we do not know how far out the decimal will take us.

Conclusion

Our project is not finished yet. We still do not know if this project is something that can be completed. We have enjoyed working on this project and have explored many geometric concepts. We have functioned together as a team and are still friends. This project tested more than the math. Imaginary numbers is something that we may want to explore further.

The challenge has allowed us the opportunity to study and explore math and science in a new way. We have been able to introduce our research and writing skills in addition to using our imaginations.

We would still like to think that we could construct the perfect heptagon, with a straightedge after doing the calculations. This may be ambitious, but we think that it can be done.

Our project is weak on the math, as we had not counted on it being so difficult to access a compiler. We will make recommendations to our school administration to lift certain restrictions for future AiS participants and other computer classes and projects.

We have also learned to work and function as a team and to persevere especially when the project looked like it would not be finished. The AiS experience has been beneficial to us in a number of ways. We have had to work as a team, we have researched, tried to write or at least understand ‘code’, and learn and explore things that were beyond our comprehension.

Acknowledgments

We would like to thank all of our teachers at Melrose for allowing us to work on our project. We would also like to thank Mrs. Raulie and Mr. Daugherty for helping us with our project and putting up with us this whole time. They gave of their own free time and held workshops after school and on weekends for us.

We would also like to thank Mr. Nick Bennett for all the help he has given us with working Java and NetBeans. Thank you all very much.

Bibliography

- thesaurus.maths.org. 2001. thesaurus.maths.org. 05 Apr. 2005 <http://thesaurus.maths.org/mmkb/entry.html?action=entryByConcept&id=362>.
- http://mathworld.wolfram.com/Heptagon.html
- Courant, R. and Robbins, H. "The Regular Heptagon." §3.3.4 in What Is Mathematics?: An Elementary Approach to Ideas and Methods, 2nd ed. Oxford, England: Oxford University Press, pp. 138-139, 1996
- Bold, B. Famous Problems of Geometry and How to Solve Them. New York: Dover, pp. 59-60, 1982.
- Bankoff, L. and Garfunkel, J. "The Heptagonal Triangle." Math. Mag. 46, 7-19, 1973.
- Dixon, R. Mathographics. New York: Dover, pp. 35-40, 1991.