Distance Calculations and The Speed at Which They Go

Project Proposal

Utah Valley University

Caiden Kehrer, James Jenkins, Alex Jackman, Kaya Peter Kikudji, Dayson Hawkins

**Introduction**

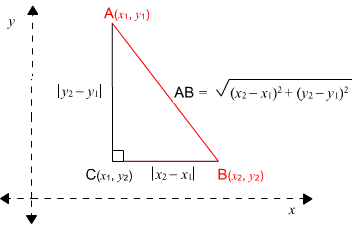
Distance calculations are used in many of areas. Whether you are planning a trip, doing calculations in gaming, or simply in a math class, we all use the distance formula in one way or another.

**Project Description**

The distance calculator will be written in assembly language. We will architect a high level diagram of the input, data logic, and output.

**Explanation and Background**

Finding the distance between two points involves an xy coordinate plane. We will make use of the distance formula. The program will take 4 input values. An x1, x2, y1, and y2 value. See the distance formula and visual reference below.



**Team Members and Project Responsibility**

Scrum Master: James Jenkins  
Architect: Caiden Kehrer

Developer: Alex Jackman

Developer: Kaya Peter Kikudji

Developer: Dayson Hawkins

**Project Discussion**

We will break the distance formula into steps. We need a function that calculates the square root of a number, a division function for that square root function, ways to output accurate results using floating point numbers, and ways to resolve bad input. We’ll work to make sure everyone stays coordinated and that everybody knows what to work on.

**Problem Statement**

The program will produce the distance between 2 points on an xy coordinate plane.

**Perceived Needs**

Doing distance calculations by hand is a long and tedious process, particularly if the points being calculated are any degree of precise. With this calculator, users can skip all the boring math and risk of mistakes and just get their answers right away. It could also be adapted as needed into, say, a video game, where it could be used to track ranges and set distance flags.

**Applications**

The applications for a distance calculator include math, 2D games, and maps. By developing this in assembly we expect it to be very efficient on hardware as well.

**Project Goals**

It’s not exactly clear what this section is for, but we intend to have this program completed by April 17th. Our goals are to complete a design document and an opcode list within two weeks, and to be in the final stages of the project before the end of March.

**Deployment Strategy**

The program will be uploaded to canvas where it will probably be used by exactly one person once, but it sure is going to calculate distances on a 2D plane like it’s meant to be used by people with too much free time. It’ll accept the X coordinate from the user’s keyboard, then the Y, and that’ll form the first point. The process repeats for the second point and then math is run and a result output. A simple executable that will hopefully be as idiot-proof as a thing can be while still fulfilling a practical mathematical function.

**Constraints**

The first constraint is time. Our delivery date is for the project and presentation is April 17th. Our other constraint is communication. We have set up communication tools such as Slack to help improve communication with our team. We have created user stories and have a standup meeting to insure we are each making progress. The second major constraint is the math. The distance formula makes use of squares and square roots, which can be difficult to adapt into assembly code thus presenting a limit on complexity and time available. Finally, the program must work in all situations without breaking, and with some complicated math and four user inputs there’ll need to be a lot of redundancies to make sure the program works in all conditions.

**Timeline**

We’ll start of by drawing out a design document. After that we’ll refine it into a general flowchart/decision tree to map out what possibilities need to be accounted for. From there it can be further refined into a list of the relevant commands and the registers that will need to be used. Then it’ll be a simple matter of turning the list into actual binary code and debugging the results. We intend to have the program completed and debugged by April 17th.