The University of Rhode Island

Convex Hull Data Structure:
Implementation of Graham's Scan algorithm.

Group Members:

Jackson Perry, Alex Lupo, Jake N, Michael

Presentation Date:

12/6/23

Table of Contents

Group Member Contributions:

Alex Lupo	Jackson Perry	Jake N	Michael
ConvexHull.cpp version 1	Report	ConvexHull.cpp version 2	Report
ConvexHull.cpp version 3	Designed test cases		

Table of Contents

The development of convex hull algorithms have dated back to ancient times in accor

Convex hulls serve as a fundamental concept in geometric and mathematical practices. A convex hull is the formation of the smallest possible shape that encompasses all given points. The convex hull includes the line segment connecting two separate points or vertices. Generally, the purpose of a convex hull is to provide a structural representation of a set of points with the inclusion of its outer boundary. Convex hulls' functionality extends to areas of encapsulation, analysis, outlier detection, and other applicable processes.

Spatial encapsulation refers to the processes behind the formation of a convex hull. The encapsulation of the points in each set is achieved by the nature of the bounding structure, ensuring all possible points are within the hull. Convex hulls' ability to capture any particular outliers is conducted through this encapsulation process. Additionally, convex hulls' computational efficiency promotes their ability to achieve desirable time complexities whilst handling large datasets. Parsing through datasets, the determinant of whether a point is a part of the hull can be incredibly useful in practical applications. For example, collision avoidance systems in a Tesla exemplifies a use case of convex hull through the detection of other roadway users, and objects. GPS systems highlight a fundamental feature of a convex hull, involved in deciphering the shortest line segment between points. This allows for path-decision making in computing the quickest path to the designated point or destination.

Intro To Project:

Method: