Presentation Notes

Amey

Abstract

We live in the world of machine learning and data science where everyday innumerable amount of data is being generated everyday.
Many digital map applications need to present large quantities of precise data points on the map. Visualising such data becomes a problem.
This project builds an efficient visualizer for interactively visualising such data and also we used collision detection to illustrate the functionalities of quadtree.

Introduction

A quadtree is a tree data structure in which each internal node has exactly four children.
Quadtrees are the two-dimensional analog of octrees and are most often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions.

Mega

Use Cases

The QuadTree can be implemented as a package in software development kit related to Maps Application as it is an efficient tool for spatial indexing. While finding a particular location on the maps, it searches recursively through the data points and is very fast compared to other algorithms.

Image Compression is another domain where Quadtrees can be implemented. Images can take up a lot of space, and require memory and processing power to keep them at their full quality. Quadtrees can compress the images without the loss of high quality like and can speed up the machine learning process of edge detection.
Quadtrees for image compression works by recursively dividing the image into four subspaces with each holding the average RGB color and the error determining that color for its subspaces.

Many games require the use of collision detection algorithms to determine when two objects have collided, but these algorithms are often expensive operations and can greatly slow down a game. Collision detection can be an expensive operation and can slow down the performance of your game. Quadtrees are one way we can help speed up collision detection and keep the game running at top speeds.

Quadtrees can be used in memory management for a big database. QuadTree can rapidly sparse data in the database and can speed up the process of retrieval of a particular data/information.