"QUADTREE VISUALIZER"

Major Project

(Fourth Year/ Sem VII)

Submitted in fulfilment of the requirement of
University of Mumbai For the Degree of

Bachelor Of Engineering
(Computer Engineering)

By

1.	**AMEY THAKUR**	**BE-COMPS B-50**	**TU3F1819127**
2.	**HASAN RIZVI**	**BE-COMPS B-51**	**TU3F1819130**
3.	**MEGA SATISH**	**BE-COMPS B-58**	**TU3F1819139**
4.	**AJAY DAVARE**	**BE-COMPS B-01**	**TU3F1718006**

Under the Guidance of
Prof. Randeep Kaur Kahlon

terna
Knowledge to Empower

Department of Computer Engineering
TERNA ENGINEERING COLLEGE
Plot no.12, Sector-22, Opp. Nerul Railway station,
Phase-11, Nerul (w), Navi Mumbai 400706
UNIVERSITY OF MUMBAI

2020-2021

Internal Approval Sheet

Terna Engineering College

NERUL, NAVI MUMBAI

CERTIFICATE

This is to certify that

1. **AMEY MAHENDRA THAKUR**
2. **HASAN MEHDI RIZVI**
3. **MEGA SATISH**
4. **AJAY RAMESH DAVARE**

Has satisfactorily completed the requirements of the

Major Project (Fourth Year/ Sem VII)

entitled

"QUADTREE VISUALIZER"

As prescribed by the University of Mumbai

Under the guidance of **Prof. Randeep Kaur Kahlon**

GUIDE **APC** **HOD**

Approval Sheet

Project Report Approval

This Major Project Report - I entitled

“QUADTREE VISUALIZER”

by the following students are approved for the degree of Bachelor in

"Computer Engineering (Sem-VII)".

Submitted by:

AMEY THAKUR	**TU3F1819127**
HASAN RIZVI	**TU3F1819130**
MEGA SATISH	**TU3F1819139**
AJAY DAVARE	**TU3F1718006**

Examiners Name & Signature:

1. ----------

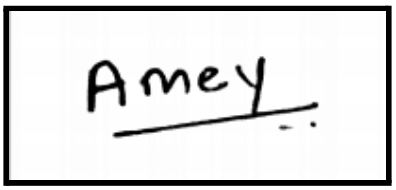
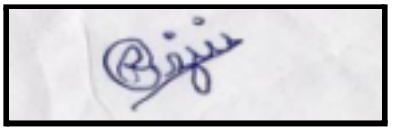
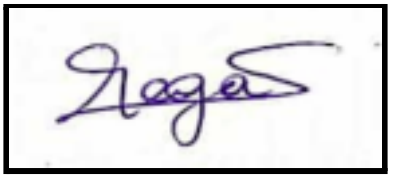
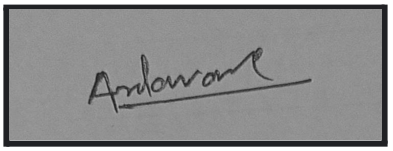
2. ----------

Date: 24-10-2021

Place: MUMBAI

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced The cartoon sources task-specific. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

AMEY THAKUR	**TU3F1819127**	
HASAN RIZVI	**TU3F1819130**	
MEGA SATISH	**TU3F1819139**	
AJAY DAVARE	**TU3F1718006**	

Date: 24-10-2021

Place: MUMBAI

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards our guide **Prof. Randeep Kaur Kahlon** for the help, guidance and encouragement she provided during the project development. This work would have not been possible without her valuable time, patience and motivation. We thank her for making our stint thoroughly pleasant and enriching. It was great learning and an honour being her student.

We are deeply thankful to **Dr Archana Mire** (H.O.D Computer Department) and the entire team in the Computer Department. They supported us with scientific guidance, advice and encouragement, they were always helpful and enthusiastic and this inspired us in our work.

We take the privilege to express our sincere thanks to **Dr L. K. Ragha** our Principal for providing encouragement and much support throughout our work.

ABSTRACT

We propose to develop a program that can show a QuadTree view and data model architecture using a single global library. Nowadays, many digital map applications have the need to present large quantities of precise point data on the map. Such data can be weather information, the population in towns. With the development of the Internet of Things, we expect such data will grow at a rapid pace. However, visualizing and searching in such a magnitude of data becomes a problem as it takes a huge amount of time. Quadtrees allow us to visualize the data easily and rapidly compared to other data structures. This project aims to build an efficient library for interactively visualizing such data, using a combination of grid-based clustering and hierarchical clustering, along with quadtree spatial indexing.

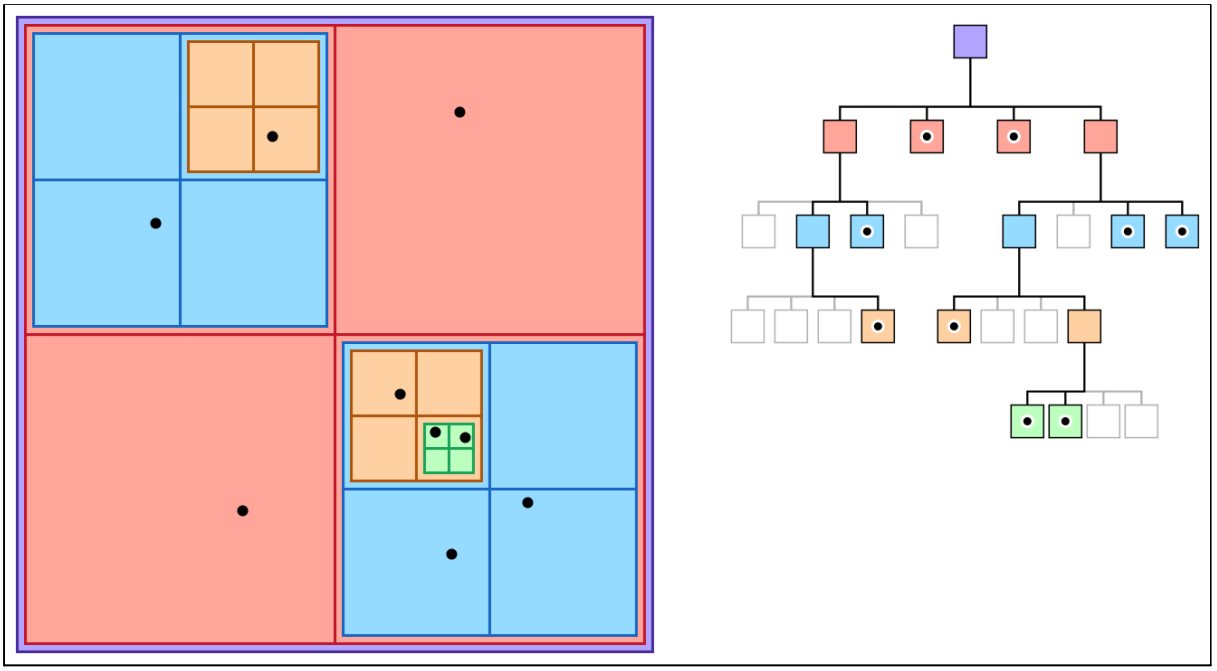
TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

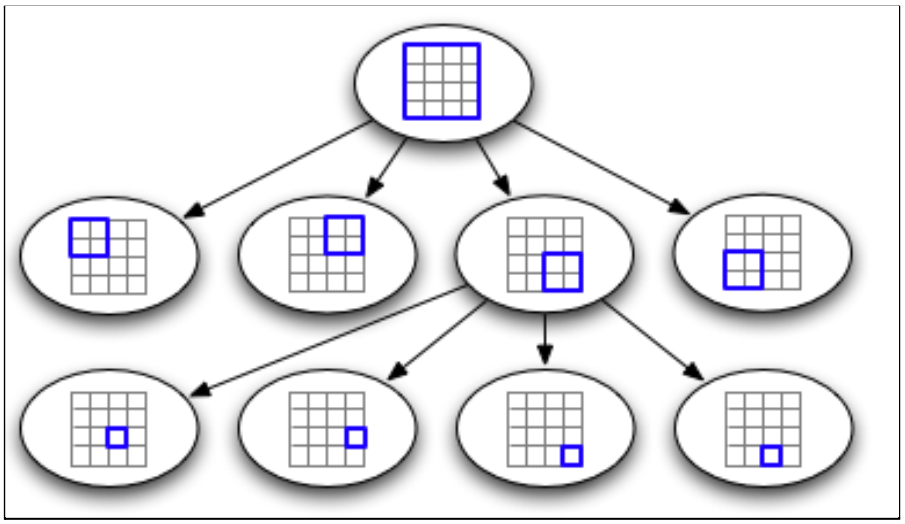
What is QuadTree?

➔ A data structure for organizing objects based on their locations in a **two-dimensional space**.



The quadtree partitioning strategy divides space into **four quadrants** at each level. When a quadrant contains more than one object, the tree subdivides that region into four smaller quadrants, adding a level to the tree.

A similar partitioning is also known as a **Q-tree**.

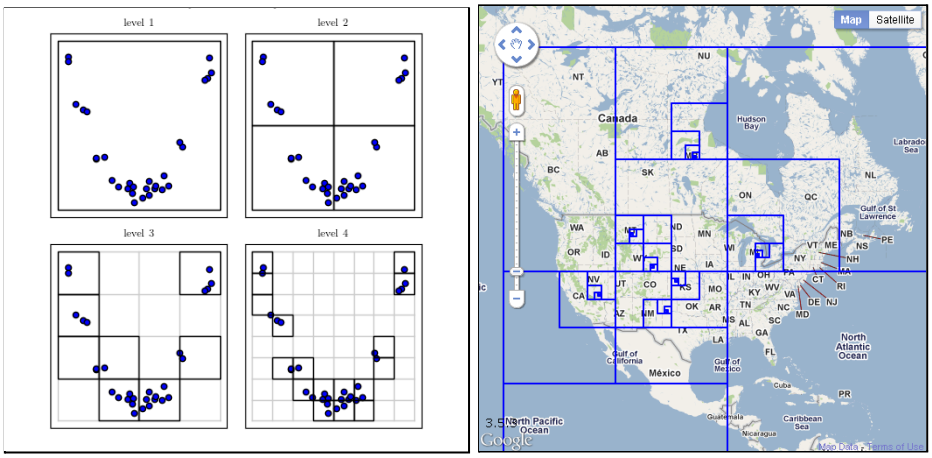


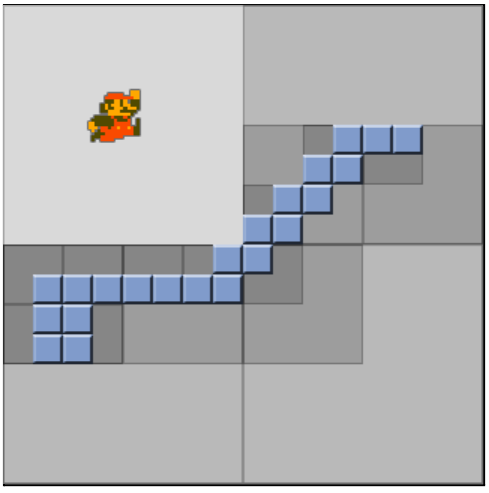
1.1 AIM AND OBJECTIVE OF THE PROJECT

This project aims to provide a single global library that can be shared, reused and readily accessed by any programmers in the C language. Quadtree Library aims to be versatile (can be used in dynamic and static contexts) yet simple. It should be lightweight and fast and must run in any system. The library must be easy to use for the programmers.

1.2 SCOPE OF THE PROJECT

Quadtrees may be implemented in computer graphics and games to visualize some aspects, or to create arcade games like Mario. It can also be beneficial in the domain of Computer Vision as it can be used for Image Compression to store data points. Additionally, this can help in the geospatial domain as it is very accurate in terms of pinpointing an exact location with the given x, y coordinates. Visualization of Graphing Complex Functions through Quadtrees is also possible as it makes the visualization smooth.





1.3 ORGANIZATION OF THE REPORT

- ➔ Chapter 1 gives a brief overview of the aim of developing this project. Also, it defines the scope of the project.
- ➔ Chapter 2 of the report includes the literature survey on the existing system, its limitations and a brief description of the types of said system.
- ➔ Chapter 3 defines the problem statement and proposes a new system as a solution.
- ➔ Chapter 4 gives an overview of the technologies used and the techniques used in the development of the project. Along with this, the Hardware and software requirements and software components are described.
- ➔ Chapter 5 elaborates the methodology used in our proposed system as well as the distinct architectures of the components. It also defines the mathematical concepts and algorithms used for the implementation of the project.
- ➔ Chapter 6 evaluates the performance of the project. We'll be setting up an experimental setup to do the same. Also, Quantitative Evaluation and Qualitative Comparisons will be given.
- ➔ Chapter 7 depicts the timeline of the project.
- ➔ Chapter 8 is the conclusion. This chapter gives a summary of the entire project.

CHAPTER 2

LITERATURE SURVEY

2.1 BRIEF HISTORY OF QUADTREE

A quadtree is a tree data structure in which each node has zero or four children. Its main peculiarity is its way of recursively dividing a flat 2-D space into four quadrants. The data associated with a leaf cell varies by application, but the leaf cell represents a "unit of interesting spatial information".

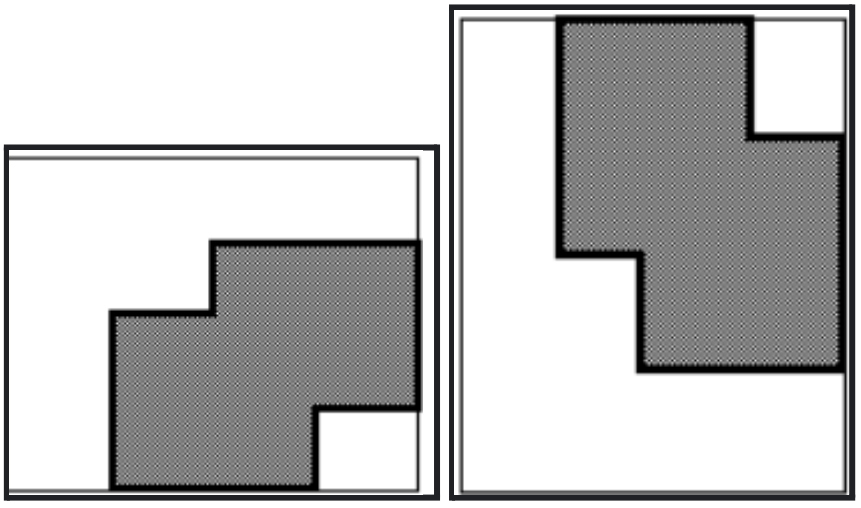
The subdivided regions may be square or rectangular or may have arbitrary shapes. This data structure was named a quadtree by Raphael Finkel and J.L. Bentley in 1974.

Some quadtree use-cases include Image processing, sparse data storage, spatial indexing, etc.

2.2 LIMITATIONS OF QUADTREE

The main disadvantage of quadtrees is that it is almost impossible to compare two images that differ only in rotation or translation. This is because the quadtree representation of such images will be so totally different.

The algorithms available for rotation of an image are restricted to rotations of 90 degrees (or multiples thereof). No other rotation is available, nor is there a facility for translation.



First Image Rotated Image

2.3 TYPES OF QUADTREE

Types:

- ➔ Point Quadtree
- ➔ Edge Quadtree
- ➔ Polygonal Map QuadTree

All forms of quadtrees share some common features:

- ➔ They decompose space into adaptable cells.
- ➔ Each cell (or bucket) has a maximum capacity. When maximum capacity is reached, the bucket splits.
- ➔ The tree directory follows the spatial decomposition of the quadtree.

So to speak in layman's terms, A quadtree is a tree whose nodes either are leaves or have 4 children. The children are ordered 1, 2, 3, 4.

CHAPTER 3

PROBLEM STATEMENT

As we live in a data-driven age, data has become increasingly important. Large amounts of exact point data must be shown on the map in many digital map applications. We anticipate that such data will rise at a rapid rate as the Internet of Things evolves. Visualizing and searching for a data point in such a large amount of data, on the other hand, becomes difficult. We propose that a **quadtree library** be implemented to make **data visualisation** easier for all users.

CHAPTER 4

REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS

- ➔ C Compiler
- ➔ GNU Make
- ➔ X11 window system for graphical output and development libraries

4.2 HARDWARE REQUIREMENTS

- ➔ 2 GB RAM
- ➔ Any Operating System

4.3 TOOLS USED

- ➔ CMake
- ➔ Visual Studio Code
- ➔ Tiny XML

4.4 TECHNOLOGIES USED

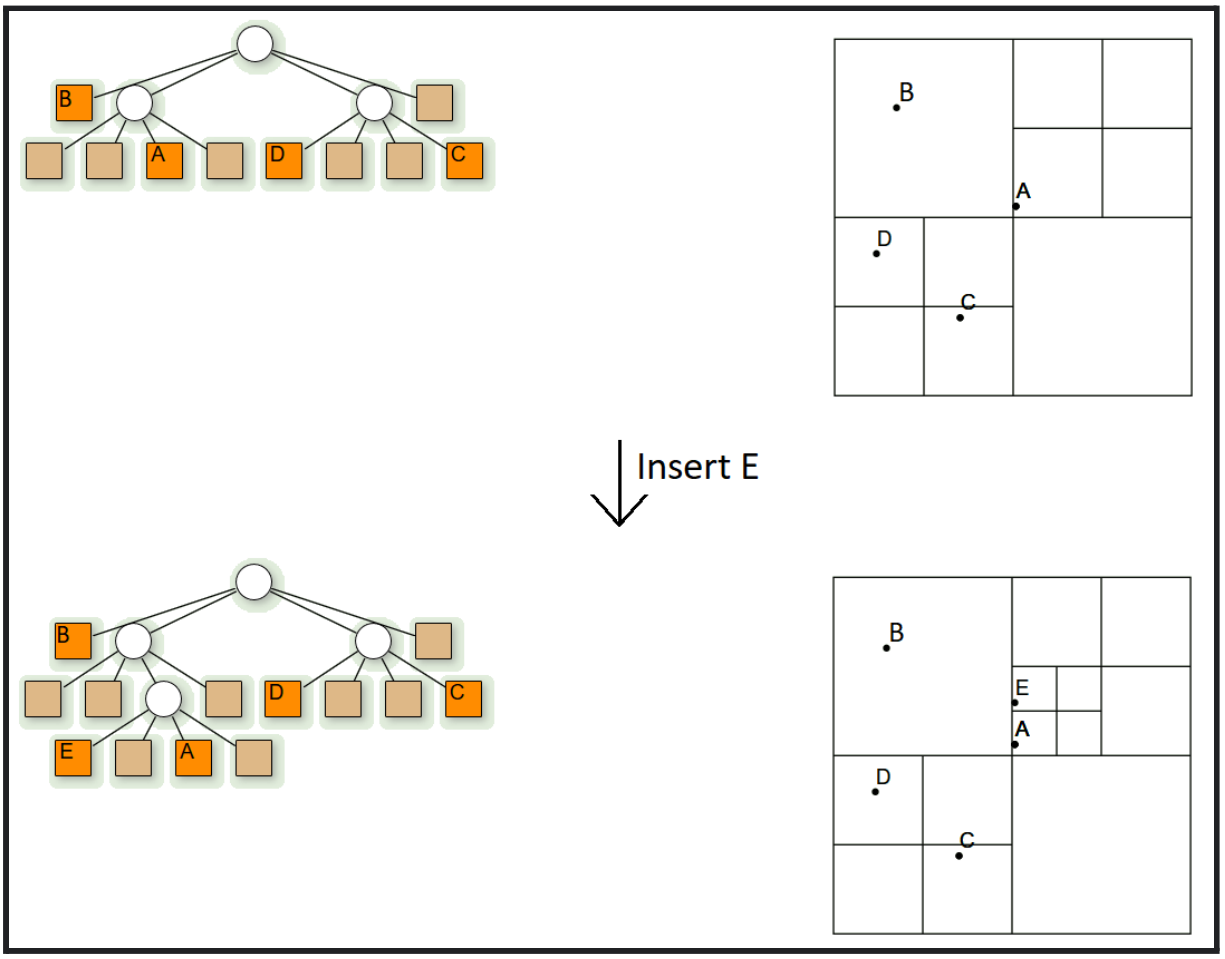
- ➔ C/C++ Language

CHAPTER 5

METHODOLOGY

5.1 WORKING OF QUADTREE

The image below shows how a quadtree changes with insertion:



5.2 ALGORITHM

Three types of nodes are used in quadtree:

- ➔ **Point node:** Used to represent a point. Is always a leaf node.
- ➔ **Empty node:** Used as a leaf node to represent that no point exists in the region it represents.
- ➔ **Region node:** This is always an internal node. It is used to represent a region. A region node always has 4 children nodes that can either be a point node or an empty node.

5.3 INSERTION IN QUADTREE

➔ Insertion: This is a recursive function used to store a point in the quadtree.

1. Start with the root node as the current node.
2. If the given point is not in the boundary represented by the current node, stop insertion with error.
3. Determine the appropriate child node to store the point.
4. If the child node is empty, replace it with a point node representing the point. Stop insertion.
5. If the child node is a point node, replace it with a region code. Call insert for the point that just got replaced. Set the current node as the newly formed region node.
6. If the selected child node is a region node, set the child node as the current node. Go to step 2.

5.4 SEARCH IN QUADTREE

➔ Search: This is a boolean function used to determine whether a point exists in 2D space or not.

1. Start with the root node as the current node.
2. If the given point is not in the boundary represented by the current node, stop searching with error.
3. Determine the appropriate child node to store the point.
4. If the child node is empty, return FALSE.
5. If the child node is a point node and it matches the given point return TRUE, otherwise return FALSE.
6. If the child node is a region node, set the current node as the child region node. Go to step 2.

5.5 COMPLEXITY

➔ Time complexity:

1. Find: O(log2N)
2. Insert: O(log2N)
3. Search: O(log2N)

➔ Space complexity: O(k log2N)

Where k is the count of points in the space and space is of dimension N x M, N >= M.

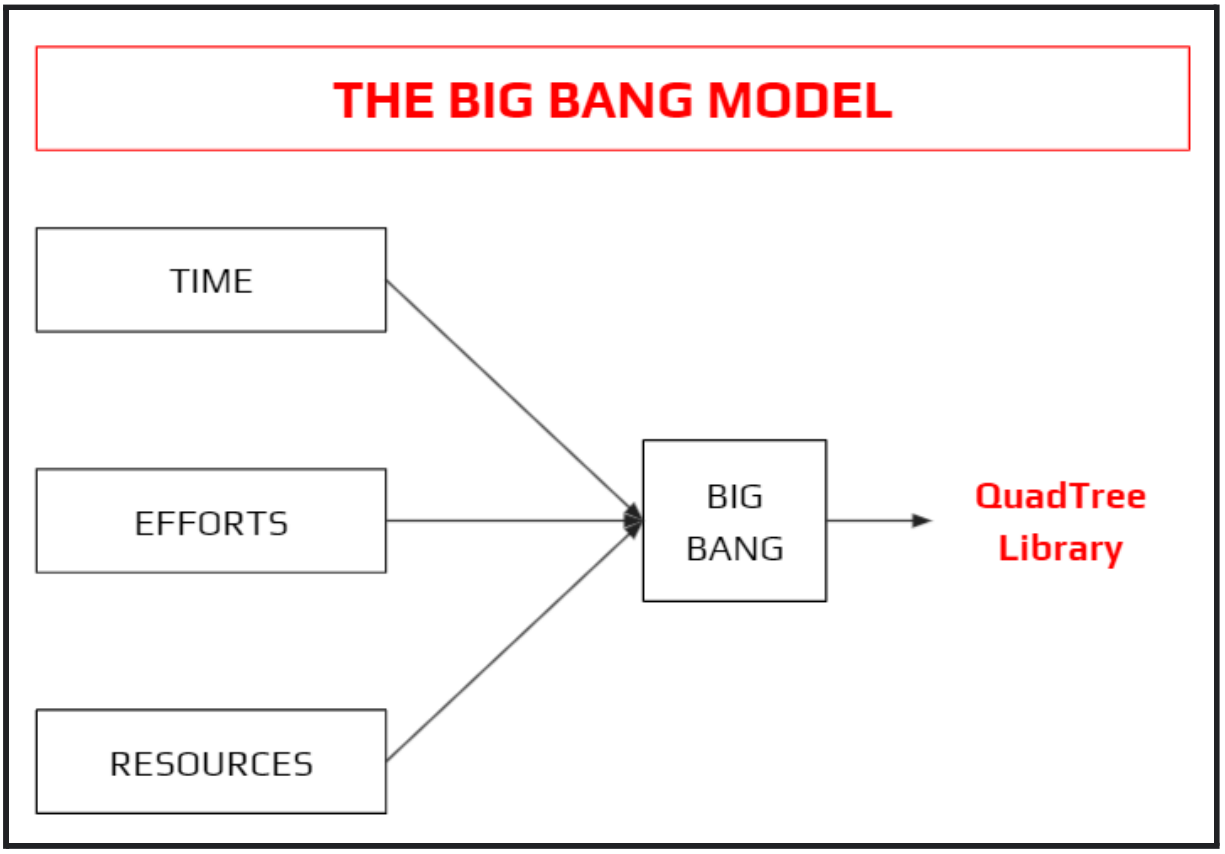
5.6 APPLICATIONS

➔ Used extensively in computer graphics.

➔ Used for image compression.

➔ Used to represent spatial relations.

5.7 SOFTWARE DEVELOPMENT LIFE CYCLE MODEL



The Big Bang model is an SDLC approach in which no precise procedure is followed. The development process begins with the necessary funds and efforts as inputs, and the result is software-generated, which may or may not meet the needs of the client.

This Big Bang Model has no set method or procedure and requires a very little forethought. Even if the consumer is unsure of what he wants, the needs are implemented without much thought on the spot.

CHAPTER 6

PERFORMANCE EVALUATION

6.1 FEASIBILITY ANALYSIS AND RISK ANALYSIS

The Quadtree Visualizer project is unique as there are no previous quadtree libraries available in the market. Therefore, it increases the chance of commercialising this product as it is something that anyone can use to visualise the data. Because the requirements have previously been clearly specified, and the project will be regularly assessed, the project also ensures technical feasibility. To guarantee that the library is error-free, proper testing will be carried out.

CHAPTER 7

TIMELINE

7.1 PROJECT PLAN -

#	Task	Start Date	End Date
1	Understand Object-Oriented Programming in C	17-09-2021	01-10-2021
2	Understand Design Patterns in C	01-10-2021	15-10-2021
3	Learn how to use TinyXml	15-10-2021	29-10-2021
4	Define ADT for QuadTree	29-10-2021	12-11-2021
5	Define the file format for QuadTree	12-11-2021	26-11-2021
6	Get your hands on the pcf_ui library	26-11-2021	10-12-2021
7	Understand the Drawing View Control of pcf_ui	10-12-2021	24-12-2021
8	Sequence diagram for your final application	24-12-2021	08-01-2022
9	Implement the Visualizer	08-01-2022	05-02-2022
10	Design the architecture of the application	05-02-2022	12-02-2022

CHAPTER 8

CONCLUSION

By the time of completion of this project, we'll be able to develop a full-featured, scalable, multi-purpose library for Quadtree implementations in C language alongside understanding the principles of object-oriented philosophy and design thinking in writing production-grade programs.

REFERENCES

1. Q. Cai and Y. Zhou, "A quadtree-based hierarchical clustering method for visualizing large point dataset," 2016 Sixth International Conference on Information Science and Technology (ICIST), 2016, pp. 372-375, DOI: 10.1109/ICIST.2016.7483441.
2. “An effective way to represent quadtrees” Communications of the ACM, Volume 25, Issue 12, Dec 1982 pp 905–910, doi:10.1145/358728.358741.
3. “Optimal quadtree construction algorithms” Computer Vision, Graphics, and Image Processing, Volume 37, Issue 3, March 1987, pp 402–419, doi:10.1016/0734-189X(87)90045-4.