**Contents**

1. Introduction

2. How it works.

3. Things to look after.

4. References

# Introduction:

This project allows you to read SRT files and provides a search functionality to get dialogues between a given intervals

# How it works:

The class library accepts a SRT file which it reads and organizes dialogues and time intervals in the form of Binary Tree. The time interval forms the Left/Right node of a B Tree. To build a B Tree following input parameters required:

1. Physical path of the SRT file.
2. Duration of the video/Audio.
3. Type of Video /Audio in terms of time i.e. H [Hour], M [Minute], S [Second].

## B Tree Creation and Criteria:

While preparing a B Tree, dialogues are read from SRT file one by one. Dialogue format is as follows:

00:00:00,970 --> 00:00:03,000

Jellyfish at the Monterey Aquarium

00:00:04,080 --> 00:00:06,080

Dude - get out of the way!

00:00:9,350 --> 00:00:13,350

Shaky Hands...

Each dialogue is preceded by time interval [00:00:9,350 --> 00:00:13,350].A check is made to get time range .This time range is split into 2 parts i.e. To/Max and From/Min time.

E.g. 00:00:00,970 --> 00:00:03,000

This time range will be split into two parts and will be treated as

From => 00:00:00,970 and To => 00:00:03,000.

Each time range is associated with corresponding dialogue. A unique Key is made using above properties.

Key comprises of:

1. From/Min value
2. To/Max value
3. Dialogue

This Key, Value pair [Node] which is inserted into B Tree.

## Insertion Criteria:

While inserting a node into B Tree a check is made on Duration. It is calculated as [Total Time period / 2] where Total Time period is Input to the API.

If From/Min and To/Max time of a particular key is less than Duration then key is placed on the left side of the B Tree else on the right side.

## Search dialogue based on Provided Time Span:

To get a dialogue from B Tree of a particular duration, API requires a Time Span object which states Hour, Minute, Seconds, and Milliseconds.

A search operation is made on B Tree as follows:

1. If provided time is <= Duration then look for left side nodes else right side nodes.
2. If provided time comes between min and max time [i.e. From and To time of Key] of Left/Right node then corresponding dialogue is returned.

# Things to look after :

Implement a more profound insertion criteria and search criteria on B Tree.

# References:

Binary Search Trees (BSTs) in C#

<http://snipd.net/binary-search-trees-bsts-in-c>