Project 3B - Team1 Documentation

## Overview

Team 1 (ie The Wookie Workgroup) created a command line tool that decodes and encodes morse code (Project3B). The group consists of Daniel Mitchel, Joshua Neustrom, and Chen Wang.

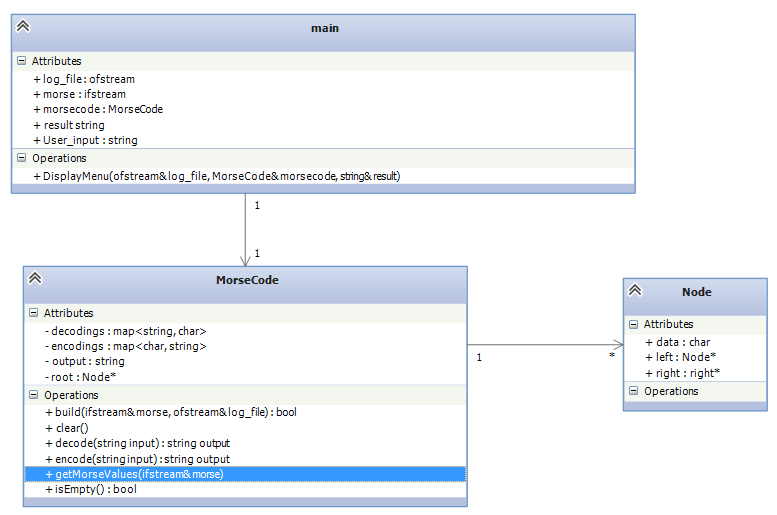
The report gives an overview of our solution including the following

1. Assumptions
2. UML Class Diagram
3. Github Project Link
4. Algorithm
5. Error Handling
6. Known Issues
7. Efficiency Analysis of Algorithms
8. References

## Assumptions

1. \* and – are valid morse code symbols
2. Use a Binary Search Tree for codes where \* means go left, – means go right and the root data is empty
3. Codes for letters in morse code separated by spaces (example: -- \*\*-)
4. Encodes letters only (upper and lower case have the same code)
5. Decode creates messages in lower case only
6. Spaces separate letters in encode and decode
7. Wookies rule

## UML Class Diagram



## Github Site

<https://github.com/WookieWorkgroup/Project3>

## Algorithm

1. Create table
   1. Open Morse Code file
   2. Create a map for encoding
   3. Create a tree for decoding
      1. Read in letter
      2. Read in code
         1. \* means go left
         2. – means go right
         3. Other symbol causes error
      3. Traverse tree to last part of the code
      4. Add node
         1. New node
         2. Set data
         3. Set final pointer
2. User Interface
3. Option 1 – Message to encode
   1. Read in letter
   2. Find letter in the map
   3. Output map based on the key (symbol set)
4. Option 2 – Message to decode
   1. Tokenize code by spaces
   2. Look left for every \*
   3. Look right for every –
   4. Get the letter
5. Option 3 – Display last result (last encoded or decoded message)
6. Option 4 – Clear input
7. Option 5 - Exit

## Error Handling

1. Bad Input – Error message retuned and user sent back to the main menu
2. Error Codes – Unique for every exception, stored in the log.txt for troubleshooting
3. Logging – Log.txt contains record of actions completed to help troubleshoot

## Known Issues

1. Does not encode numbers or symbols (throws exception and sends user back to the main menu)

## Efficiency of Algorithms

1. Build – O(2n) – Two consecutive loops. (not nested) One loop to place each letter in the map and one loop to place a letter in the tree
2. Encode/Decode – O(n) – one loop to encode/decode each letter

## References

1. Binary Tree from class Lectures used as a starting reference
2. The Force