Json: Read/ wite

Contents

[Objective: 2](#_Toc75447015)

[Functions: 3](#_Toc75447016)

[Design 3](#_Toc75447017)

[Toolbox: 4](#_Toc75447018)

[Input box: 4](#_Toc75447019)

[add: 4](#_Toc75447020)

[Remove: 4](#_Toc75447021)

[Find: 4](#_Toc75447022)

[Update: 5](#_Toc75447023)

[Json view: 6](#_Toc75447024)

[text view 6](#_Toc75447025)

[colours 6](#_Toc75447026)

[code design 7](#_Toc75447027)

[class: 7](#_Toc75447028)

# Objective:

To learn c# and wpf as well as get more familiar with json.

# 

# Functions:

* Load file- load in an existing file and update it
* create file – create and save a new file
* find in file- find words in file
* select and change- find and change
* error checking- check for errors when writing in json
* auto fix- eg bracket and other stuff – fix errors when witting in json

# Graphical user interface Description automatically generatedDesign

## Toolbox:

* change mode: find, add, remove and change
* change display mode
* could be slidable

## Input box:

* what data type being input
* data name
* data input box
* function buttons
* need to change with mode

### add:

* data check:
  + type
  + length
* array check:
  + ‘,’ for separator for names and data
    - Could use new lines/tables for advanced mode
  + Application

    Description automatically generated with low confidenceIf not array type just add to string

### Remove:

### Find:

### Update:

## Json view:

* what it will look like in json view
* edit data here

## text view

* what it will look like in text view
* edit data here
* format to be readable

## colours

* app colours should be changeable

# code design

## class:

json views

vars:

* json view
* text view

function:

* get text view()
* get json view()
* convert to string()
* convert to json()

json creation/load

vars:

* json file name: string
* json data

function:

* create json(data : string)
* load json( file:string)
* store JsonFile()
* set json file name
* get json data()
* find data(name:string, data: auto)

json string view

vars:

* Text file
* Text data

function:

* create string()
* get string data()
* find data(name:string, data: auto)

edit mode

vars:

* editMode

function:

* change mode()
* get edit mode()

enum: EditMode

* add
* remove
* find
* change

display settings:

vars:

* font style
* colours: list/array/map

function:

* get font style()
* get colours(coulorName:enum)
* get colour list()

enum: coulorPosName

* general background
* box colour
* tool box background
* input box background

font style:

vars:

* font colour
* font size
* font style

functions:

* change colour(colour)
* change size(size:int)
* change style(style)
* get colour()
* get size()
* get style()