# ShatteredTextual

Textual game engine

## Ongoing

### Axioms

### Learn

### Tasks

#### Quick Tasks

* Make Readme on github (see udemy example)
  + Speak about the reason of singleton usage ( small project )
* Transform project to be generic and look like an engine

#### To do

###### WPF App

* UI/View
* Logic
* Icon system
* Transfer to .Net Core 3

###### Engine

* Json inEngine
* Make inventory optional
  + Use Boolean

## Engine

### To do

* Try to use Regex for parsing
* Replace Singleton pattern by DI or service locator
* See factory function for class with init function
* Fixe Error
* Implement Is final in state ?

### Polish

* See if all files are closed
* Precise when noexcept
* Use initialization {} instead of affection or () if possible

## Data Creation WPFApp

### Learning

#### MVVM

* Read articles
* See downloaded video

### ToDo

* Review udemy course app in detail
* Review wpf sample
* Look for mvvm example application

### Tips

* Use app.xaml for global

### Ongoing

* Treeview

### UI

#### Polish

* Could use border
* Style
* Icon
  + Settings menuitem
  + Application
  + Level in treeview
  + Choice in treeview
* Add Question mark icon for each input
  + When icon hovered, a tooltip(controltip) appears explaining the usage of this input in the engine
* Highlight selected level & selected choice

#### Layout

###### Level & choice

* How to bind choiceindex
* Add option to sort Treeview
* Add filtering option

###### Link

* Listview or datagrid

### Logic

#### Must Have

##### Model

###### Validation

* IDataErrorInfo
* Used when entering condition of link
* Use on game title

###### PropertyChanged

* Implement it in each model class

##### View Model

* Create VM for level
  + Deal with Unlocked items & progressionevent
* Create VM for Choice
  + Deal with links ect
* Contained by MainVM

##### Saving / Loading

* Use json serialization
  + See weather app & other tutos
  + Use Newtonsoft
* Data folder destination is entered in settings
* Default destination => ../ShatteredTextualEngine/data
* Re-load data when path changed

##### Setting

* Indicate game path
* Create VM
* Path property
  + Return “no directory selected … “ if path == null in get

###### PB

* Where and how to store user settings ?
* How to handle new window creation ?
  + Use User control ?

##### ProgressionEvent List

* Add delete, add & edit button
* Grey them if no event selected

##### Treeview

* How to sort ?
* How to implement in MvvM ?

###### SelectedItem

* Try to make second command for decond level of tree
* Other solution => behavior + attached property
  + How to deal with different type in treeview ?

###### Choice Display

See index

#### Should Have

##### Keyboard Shortcut

###### PB

* How to implement in MVVM ?

#### Could Have

##### User control

##### View Model Locator

##### Custom file format

* 1 file per game
* Still need to assign data folder
* See openfile dialog

#### Polish

* Implement ViewModel First
* Add || modify design to add unit test
* Comment code
* Define access keys
  + See wpf tuto

## Json InEngine

## Icon

* Make Txtfile for each Letter & number
* When saving
  + Wpf app launch bash script giving game title
  + Bash script do some magic by combining separate txt file in one dest file “Icon.txt”, that has been reset before

## Streaming

* Make engine settings class with option for json streaming or custom streaming
* Search for custom file format , how to make them, use them ect

## Other

#### Installer

* **Look at file association**
* **Make installer to install both engine and editor**

#### Custom file format

* Must be placed at the root of the project
* 1 file per game

#### Combine engine & Editor

## Legend

Unsure To Develop Focus