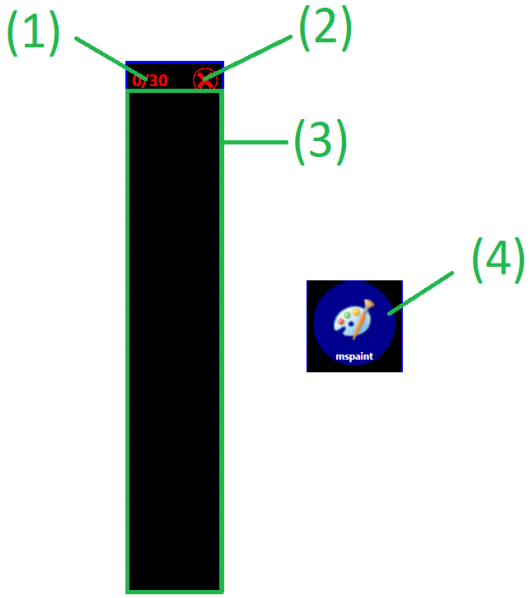
VIRTUAL TOOLBELT

-INTRODUCTION-

Virtual Toolbelt helps the user in managing frequently used files and folders by providing quick and safe access to them through a minimalist GUI. When exited the program will generate a text file in its folder containing all the paths to the inserted files. The text can be modified by adding or removing some paths and the Virtual Toolbelt will use it in its next initialization.

-USAGE INSTRUCTIONS-



(1) Top

Contains a counter which tells how many files are holded at that moment, click on it to be able to drag the Virtual Toolbelt around and double click to minimize the window;

(2) Exit button

Holded files are automatically saved when exit is pressed;

(3) Body

Contains the icons. If there are more icons than the ones visualized, it can be scrolled using the mouse wheel;

(4) Icon

Double left-click it to open what holds, and right-click it to remove it;

-TECHNICAL DETAILS-

The project follows the MVVM pattern, so anything that needs to access underlying data is moved to the ViewModel. A couple methods are left inside the MainWindowView.xaml.cs because they don't interact with the underlying data in any way. I've tried to keep the logic as simple as possible and flexible for future modifications.

The only Model is the Holder, which contains the basic informations necessary for the creation and usage of an Icon. It's an Immutable object that is stored inside the HolderViewModel, which mutates only when highlighted, but immutable if seen from the classes not part of the framework.

HolderViewModel act as DataContext for the HolderView, which is used to represent every HolderViewModel through a Template. It binds its events(like left-clicking) to commands in the HolderViewModel. MainWindowViewModel is in charge of dealing with events that concern the MainwindowView exclusively, and holds the ObservableCollection HolderVMs, which is binded to the ListView and allows the displaying of the Icons.

Commands are binded to events using InvokeCommandAction from Microsoft.Xaml.Behaviors.Wpf. When commands were in need to access the event arguments, a override of InvokeCommandAction's parent class is used.

-VERSION HISTORY-

1.0.0

1.0.1 Added the possibility of dragging the window by grabbind the top

1.0.2 Modified the icon's highlight color to reduce its contrast with the text

1.0.3 Fixed a bug causing scrolling to not work in certain situations

-CONTACTS-

Email: Truscellojobs@gmail.com