|  |  |  |  |
| --- | --- | --- | --- |
| Bonus: | Where: | Description: | Details: |
| entering a travel distance without a button | Window3 | Only allows input of 0-9 digits in distance text box until the “Enter” key is pressed. | 2 methods checking that the key is not:   * An action key * A character other than 0-9   If statement that checks if the “Enter” key was pressed… |
| Different markings used for the different statuses of the bus. | Main window | Each item in the list box will have a different color corresponding to it’s status.  **Green**=READY2GO  **GREY**=cannot be used because: refueling , being serviced, or middle of ride  **Red**= needs maintenance  **Purple**= needs to refuel  (km at exactly 1200) | * In the XAML code, under the   data template: “myListBoxDisplay”, The border and background were bound to the Status of the bus   * The bus class inherits from “INotifyPropertyChanged” and implements the event of property changes. |
| Enrichment of bus functionality: | Bus Class | * Method updateStatus() * Additional status of “NEEDREFUEL” and “NEEDSERVICE” | * Can update the status of a bus based on it’s data-instead of sending the new status of the bus to be checked and updated each time. * Notifies the user that the bus needs refueling or maintenance. |
| Enrichment of bus functionality: | Main window | * buses that are busy cannot be sent for refueling or travel in main window. * Buses that need maintenance cannot be sent for travel * buses that are busy cannot be sent for refueling maintenance in busDetails window. | * Data binding the ListBoxItem to the status of the bus * Data binding the buttons to the status of the bus. |
| displays time left until bus will be ready for travel. | Main window | * Buses that are busy display a progress bar marking their progress until they will be available. | * Data binding the progress bar value to the progress property in the bus class. * Data binding the progress bars visibility to it’s status in the XAML code. |