### Team Black Magic (Group 4)

### Names \_Brad Bensch, Kevin Mulligan, Bill Spaw, Chad Koppes, Jenny Zhen

### Model-View-Control Exercise

The Model-View-Control is an important architectural pattern. You should have a solid understanding of the structure of this pattern, and the responsibilities for each component. It is also important to know how to map elements in a standard GUI-based application onto this architectural pattern.

|  |  |
| --- | --- |
| In general terms what are the responsibilities for the three components in the Model-View-Control architectural pattern?    In a simple application that is using the MVC architecture, *every application element* must reside in one of these three components. This then means that the Model, View, or Control component must be responsible for *every operation* that the application performs. Be detailed in listing the responsibilities for these three components trying to capture, in general terms, every operation that a generic GUI-based application performs. |  |
| **Component** | **Responsibilities** |
| **Model** | · Holds state or data independent from the other two components. |
| **View** | · Provides the interface for the user to interact with the program. |
| **Control** | · Provides the logic or functionality for changing states. |

|  |  |
| --- | --- |
| For these questions, provide answers which are specific to the application that you are designing and implementing through Units 2 and 3. |  |
| **Question** | **Answer** |
| What items constitute the "state" of the system encapsulated in the Model component? | · The state is spread across two different classes to a degree that we don't have an explicit Model.  · GUI and FileWindow both hold the majority of the state for the program. |
| Are there different "views" of the Model that the View component maintains? If yes, what are they? | · Yes. The main GUI provides the majority of the view for the Model, but other pop ups, for example link view, provide a separate view of different model functionality. |
| What are the interactions, if any, from Control to View? View to Control? Control to Model? Model to Control?  Model to View? View to Model? | · View to Control: The command pattern works to separate the function from the GUI.  · Control to View: Not really any.  · Control to Model: All the command objects update data in Model.  · Model to Control: Not really any.  · Model to View: Anything that's viewed! All the data returns to the view as the document gets edited.  · View to Model: Not really any - it all goes through some intermediary controller. |