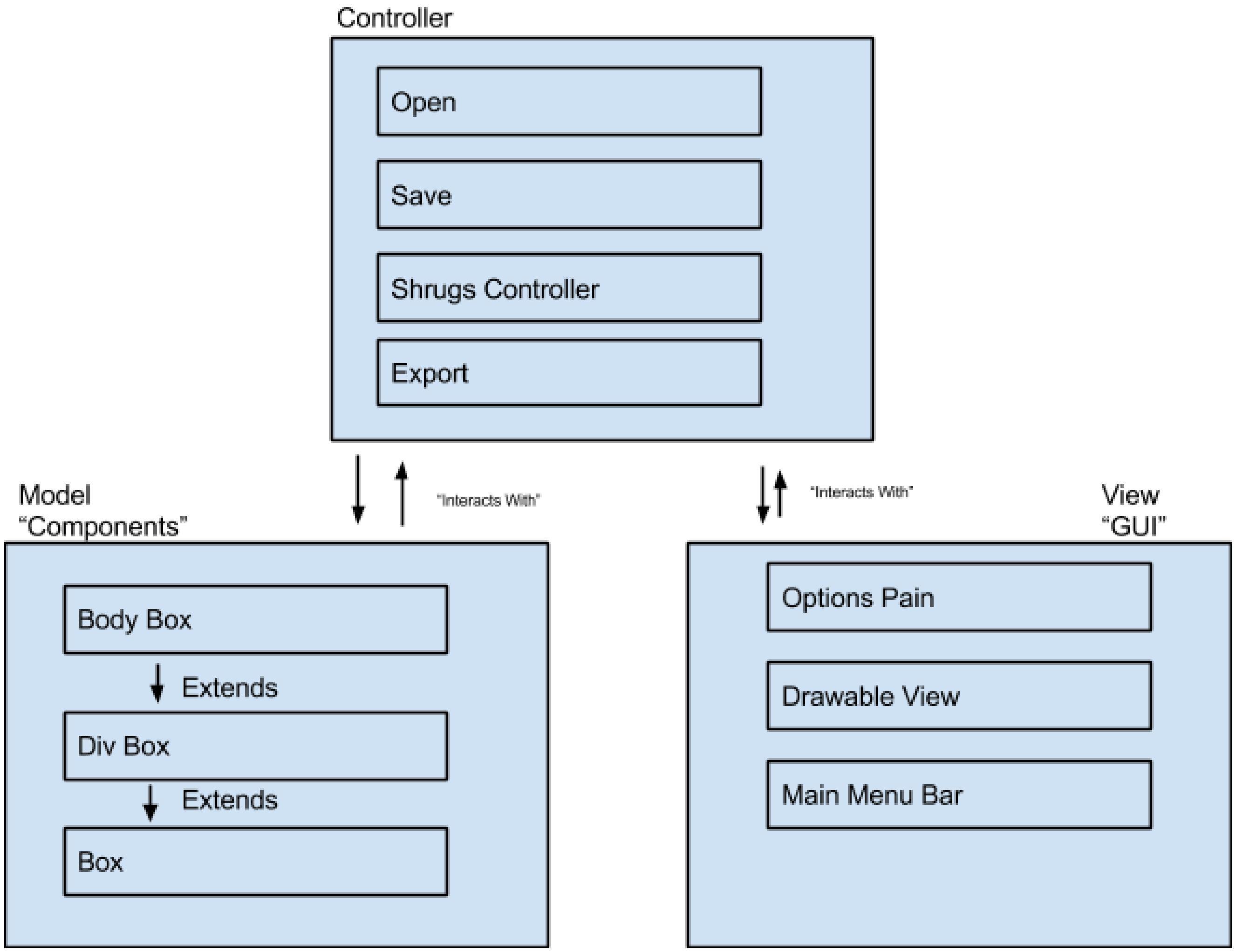
Shrugs

Shrugs Helps Regular Users Generate Sites

<u>Incrememtal/Regression Testing</u>

Classification of Components / Hierarchy



Outline of Components; Input & Output of components; and Dependent Components

Base Components:

- ***Model (Components)***
    - **Independent Components:**
        - **Box** (Superclass for all Box Subclasses) - Describes a box in terms of its two corners and its parent.
            - Input: Two corners, a Style object, a parent Box
            - Output: A valid Box object
        - **DivBox** (Extends the Box Object) - Includes a list of child boxes
            - Input: Two corners, a Style object, a parent Box, any number of child Boxes
            - Output: A valid DivBox object
        - **BodyBox** (Extends the DivBox Object) - The top-level box
            - Input: Two corners, a Style object
            - Output: A valid BodyBox object
- ***View (GUI)***
    - **Independent Components:**
        - ***OptionsPane*** - A graphical pane that will display various options that can be set for a given selected *Box* object or any of its subclasses.
            - Input: Takes in a *Box* component or any of its subclasses.
            - Output: A list of *Box* attributes which can be edited via user mouse/keyboard input from the *OptionsPane*.
            - Dependent Components: This component calls the *ShrugsController* component to handle the modification of the *Box* object model data. The *View (GUI)* and *ShrugsController* components both can call this component.
        - ***DrawableView*** - A graphical canvas pane which takes in user mouse input to interact with the ShrugsController to modify the *Box* object model data.
            - Input: Takes in user mouse clicks, mouse drags, and mouse releases and a flag to decide which *Box* component to create.
            - Output: A *Box* component of subclass type based on the flag passed in.
            - Dependent Components: This component calls the *ShrugsController* component to handle the creation of the *Box* object model data. The *View (GUI)* component calls this component.
        - ***MainMenuBar*** - A graphical menu bar which takes in user mouse click input. The MainMenuBar contains MenuItems which interact with each of the Controller's independent components. Namely the *Import, Export,* and *ShrugsController* components.
            - Input: Takes in user mouse clicks to decide what components to call.

        - Output: A call to a component respective to the menu item clicked by the user.
        - Dependent Components: This component can call either the *ShrugsController, Export,* or *Import* components of the *Controller* base component. *The View (GUI)* component is what calls this component.

- ***Controller***
    - **Independent Components:**
        - ***Import*** - This component handles the process of importing a Shrugs project file into the Shrugs editor. This component interfaces and interacts directly with the *Model* to create/update *Box, BodyBox, and DivBox* objects.
            - Input: Takes in a user-specified file location of a Shrugs project file.
            - Output: A list of *Model (Components)* to populate the *DrawableView* with.
            - Dependent Components: This component directly calls the *Model* component to handle the creation of *Box, DivBox, and BodyBox* components. The *MainMenuBar* component calls this component.
        - ***Export*** - This component handles the process of exporting a Shrugs project file to HTML and CSS. This component interfaces and interacts directly with the *Model* to convert *Box, BodyBox, and DivBox* objects into corresponding HTML and CSS based on the *Model's* data.
            - Input: Takes in a user-specified file location of a Shrugs project file and a list of all *Model (Components) to* export.
            - Output: Packaged HTML and CSS files.
            - Dependent Components: This component directly calls the *Model* component to handle the conversion of *Box, DivBox, and BodyBox* components to HTML and CSS. The *MainMenuBar* component calls this component.
        - ***ShrugsController*** - This component is the the main interaction with the *Model*. It handles the creation/updating/deletion actions to be performed on the *Model (Components)* base components as well as all of it's sub-independent components.
            - Input: Takes in *Model (Components), Box, DivBox, and BodyBox* components.
            - Output: Creation/updating/deletion of input *Model* data.
            - Dependent Components: This component directly calls the *Model* component to handle the creation/updating/deletion of *Box, DivBox, and BodyBox* components. The *View (GUI)* calls this component.

We chose to test our application using a bottom-up incremental testing approach. In this approach, we started testing the very lowest individual components of the software using drivers, and then worked upwards, integrating base components incrementally going up the software hierarchy. In order to do this, we had to write drivers and use user GUI input to provide the test input for the modules being testing (Until we reached the top of the hierarchy).

MODULE: Export

INCREMENTAL TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	If unable to open/create specified file, application shouldn’t crash	2	Throw an IOException if the file can’t be opened

REGRESSION TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	After adding IOException throw, no error is displayed if file isn’t opened	2	GUI should catch IOExceptions from export and create an error message

MODULE: Open File

INCREMENTAL TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	If unable to open/create specified file, application shouldn’t crash	2	Throw an IOException if the file can’t be opened
2	Method does not account for invalid layout files	2	Method should return false if layout tree isn’t valid

REGRESSION TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	After adding IOException throw, no error is displayed if file isn’t opened	2	GUI should catch IOExceptions from export and create an error message
2	GUI doesn’t handle method returning false	2	If method returns false, GUI should display an error and not process layout

MODULE: Save File

INCREMENTAL TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	If unable to open/create specified file, application shouldn’t crash	2	Throw an IOException if the file can’t be opened

REGRESSION TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	After adding IOException throw, no error is displayed if file isn’t opened	2	GUI should catch IOExceptions from export and create an error message

MODULE: Components

INCREMENTAL TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	If the box is drawn overlapping a child object, nothing is done to prevent object overlapping	1	Prevent the object from being created
2	If the object is drawn outside of the possible range of objects, the box will still be drawn	2	Prevent the object from being created

REGRESSION TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	The object if built outside the range of the canvas, will overlap parts of the menu. preventing these parts from being usable.	1	Prevent the object from being created, or a less desirable method would to have the object act behind the menu class. Preferable both.
2	If the screen would be resized, the objects would not take this into account and will not dynamically resize itself.	3	Have box objects dynamically resize on screen resize.
3	If the application window is moved, there is a rare case where the box objects will not move along with the pane.	2	Have the box objects relocate once the application screen is moved.
4	When the menu pane is resized, the box objects are overlapped and repositioned.	3	Prevent the object pain frobecomingnmisshapened, making the draw screen unchanging so the box objects will be consistent.

MODULE: DrawableView

INCREMENTAL TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	If the user draws a box with ending coordinates that are less than the start coordinates, the application shouldn’t fail to draw the box.	2	Use a math function to correctly determine how to draw through the given coordinates.
2	Boxes should not be able to be drawn intersecting each other but there is a corner case where a larger box erroneously can intersect a smaller.	2	If the two corners of a box are enclosed in a box being drawn, it'll check whether it intersects any of the first-level children of that box before creating it.

REGRESSION TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	Fixed the box drawing to support the user drawing a box with endpoint coordinates that are less than start coordinates.	2	Implemented methods from the Math class to determine how to draw through the given coordinates for this special case.
2	Corrected corner case where a larger	2	Don’t create a box object being drawn if

	box was able to intersect with a smaller box.		it is found to intersect any of the first-level children of that box.

MODULE: MainMenuBar

INCREMENTAL TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	MenuBarItems should respond to user mouse input with corresponding function calls to its labels.	1	Use the actionPerformed handler and ‘instanceof’ to determine which MenuBarItem was pressed.

REGRESSION TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	Fixed MenuBarItems not responding to mouse presses.	1	Added missing action listeners to MenuBarItems.

MODULE: OptionsPane

INCREMENTAL TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	Options pane should scale proportionally with main window but it doesn’t if the window gets too large.	3	Use a layout manager to ensure options pane stays in it’s respective area and scales accordingly with the main window.

REGRESSION TESTING

DEFECT #	DESCRIPTION	SEVERITY	HOW TO CORRECT
1	Fix the options pane scale issue.	3	Use dimensions of the main menu and a layout manager to ensure the options pane continues to scale on main window resizing.