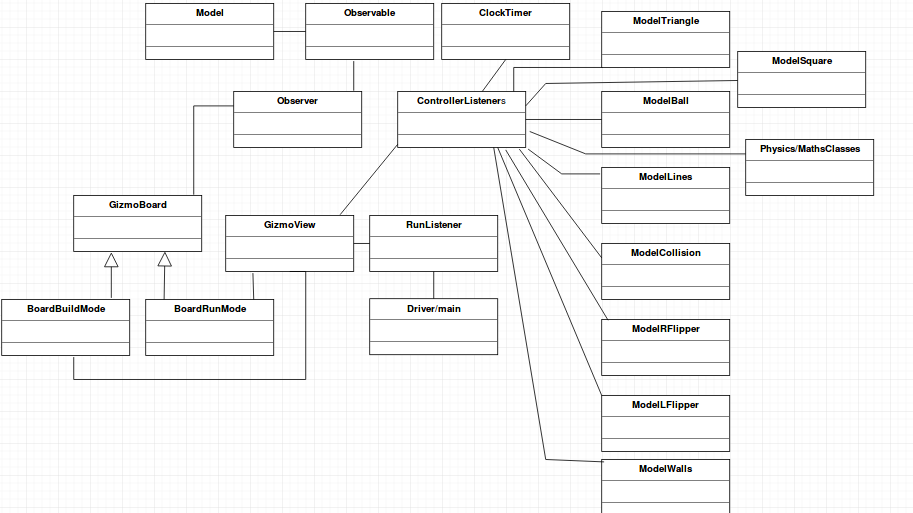
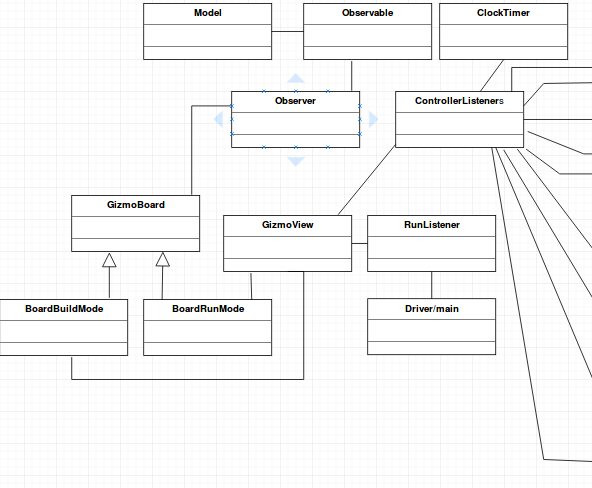
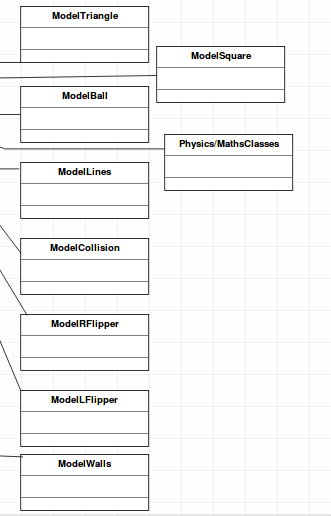
**Class Diagram & Description of Classes MW Team 4**

Class Diagram:



Zoomed in Screen shot in 2 parts:





Description of Classes:

Model Classes

Square: lest the user add the square gizmos to the build mode of the game, creating squares of dynamic sizes that the user chooses. Handles operation such as key connection and specifying the colour of the square.

Lines: Can be used to create lines implementing shapes on the GUI, these can be adjusted to the user’s size as they please.

Triangle: lest the user add the triangle gizmos to the build mode of the game, creating triangle of dynamic sizes that the user chooses. Handles operation such as key connection and specifying the colour of the triangle.

Ball: handles the size of the ball and the coordinates within the interface in both build mode and run mode. User can place the ball at a specified location within the GUI for game play.

Collision: Class to calculate the time until the collision and if the ball is to collide, the data will be updated in the view and the interface will be repainted.

Right and Left Flippers: lets the user create the flippers in the interface. Also will react to mouse click events which will trigger the flippers to rotate when the user is in game-play mode.

Wall Class: Lets the user create an object which can act as an obstruction a part of the interface, to provide an object which the ball can collide with. Will be user-defined i.e. the size will be dependent on how and where on the interface the wall will be placed.

Model Main Class: Will contain most of the main functionality of the of the model classes. Will use this class to reference the individual model classes used to create gizmos, the ball and flipper etc.

Physics classes: a variety of Java API classes will be used such as geometry, which will become more apparent as the project implementation progresses.

Controller Listeners: the controller package will contain a variety of classes which has not been included in the diagram as it will not fit the size of the page. Controller classes will include: Add Ball Listener, Add Absorber Listener, Add Flipper listener, Delete Listener, LoadListener, BuildGame Listener, ConnectGizmoListener and RunListener which will trigger a key event to start/run the game.

Driver / Main class: will contain an instance of the model and also an instance of the view with the model instance being passed into the parameters of the view object instance, similar to the driver of the folio tracker.

BoardRunMode: is the graphical user interface which the user will view when in game-play mode. This will be the main part of the game displaying the flippers in action, ball movement and collisions as well as gizmos placed on different locations of the interface.

BoardRunMode: will be the main interface the user views when in build mode. In this mode the user will be able to add a gizmo as the default when the system is loaded. The user can also add the ball, flippers, and walls, specify the colour of the different gizmos and also set the gravity and friction setting of the game.