API Changes

*Movement and Image API*

Added **public** **void** updateTurtleAbsoluteOrientation(**double** newAngle) and **public** **void** updateTurtleAbsoluteLocation(Point2D newLocation) to TurtleHandler. Also, added corresponding **public** **void** updateAbsoluteOrientation(**double** newAngle) and **public** **void** updateAbsoluteLocation(Point2D newLocation) to Turtle. These were added because we had forgotten to account for the case in which the user would specify a specific angle or location they would want the Turtle to move to without using math such as forward or backward commands or adding/subtracting angles to the Turtle’s orientation.

Also added **public** **static** **boolean** isValidPoint(Point2D newLocation) to ImageUpdater so that TurtleHandler (or potentially any other class) can make sure any Point2D is in the grid or not. This was specifically added for updateTurtleAbsoluteLocation in the event that the user specified location isn’t actually valid.

Changed method in ImageUpdater to **public** **void** drawLine(Point2D from, Point2D to) removing the Color parameter. We did this because ImageUpdater should be able to get the pen color itself – that is, the TurtleHandler that would call this method doesn’t need to have access to the Pen.

Made a GeneralTurtleHandler superclass (that TurtleHandler and TurtleGroup both extend). We did this to allow for the possibility of multiple Turtles. The methods are all abstract, keeping unity between TurtleHandler and TurtleGroup.

Added **getOrientation(), getTurtleLocation(),showTurtle(int** show**), updateImage(**Image newImage**), clearLines(), setPenPosition(int** penPosition**), getPenPosition(),getShowing(),getPenHandler(),setLineWidth(double** width**),** and **towards(**Point2D location**)** methods in the GeneralTurtleHandler to reflect necessary commands. While we had known about some of these commands beforehand, the need for these methods only became apparent as we further progressed in our code.

Removed the setBackgroundColor method from ImageUpdater as the ability to change background colors was moved elsewhere in the program.

Added a PenHandler in the **drawLine(**Point2D from**,** Point2D to**,** PenHandler mainPenHandler**)** method in ImageUpdater because we moved the PenHandler from being in the ImageUpdater to being inside the Turtle. Thus, the ImageUpdater wouldn’t instantly have access to the PenHandler.

Added **getTurtleCanvasSize()** and **clearLines()** because the TurtleHandler needed them, firstly to find the Turtle’s location within the canvas bounds (and since the TurtleHandler didn’t have access to the canvas itself, it asked the ImageUpdater for the dimensions) and secondly so that the TurtleHandler could tell the ImageUpdater to clear the lines drawn by the Turtle as per one of the commands.

Added **setLineWidth(double** width**)** and **getLineWidth()** to Pen so that the user could change the style of the Pen. Likewise, these methods were added to PenHandler as well.

**getPenColor()** and **getPenPosition()** were also added to PenHandler because TurtleHandler only wanted to tell ImageUpdater to draw if it knew the pen position was down but it didn’t have access to the Pen itself, and the ImageUpdater needed to know the pen color to draw different colored lines.

updateImage in Turtle was changed to **updateImage(**Image newImage**)**, eliminating the need to throw an error if the location was invalid and cutting out the step of finding the image itself.

setPenPosition, getPenPosition, and getPenHandler were all added to Turtle because we moved the PenHandler from being in the ImageUpdater to being inside of the Turtle.

**show(int** show**)** was added to Turtle as well to support the show/hide commands.

A TurtleGroup class was added with the same public methods as TurtleHandler, with additional addNewTurtle and setActiveTurtles methods to support the concept of adding multiple Turtles at the same time.

**int** **getID()** was added to TurtleHandler with the idea that every time a new TurtleHandler was created, that would mean another Turtle was created and every Turtle should have an ID number associated with it so that if there were multiple Turtles, the user could select which Turtles to set on/off via ID number.

*Feature API*

**public** **void** **makeButton()** was changed to **public** Button **void** **makeButton()**. It was changed so that the Button that corresponds to the feature could be accessed in the Workspace class.

*Help Page API*

**public** **void** **buildScene()** was removed, and added a **class** Browser **extends** Region to create a Browser for the HTML formatted Help page.