# Approach

I created the XFL and AS3 to be flexible, easy to modify and easy to add more features and functionality. The UI is clean and the user experience is basic.

Run the master.swf located in FLASH > demonstration\_course > final.

## UI

Create a clean UI with white space. Shadows were used to show depth. Text prompts the user to do something.

## Code

### Make the code as flexible as possible.

Much of this code could be reused for other dragger types. It’s a good foundation and could easily be modified.

### Avoid hard coding values.

I tried to avoid hard coding values. Ideally, any values dynamic values could be added to an xml file, UI component that could be changed very quickly without changing the code.

### Use the movieclip symbols for dynamic positioning

Move the symbols on the timeline and the code will use the values when dynamically placing content on the screen. Publish to see the results.

### Library Organization

I created the folder structure to make it easy to expand and enhance the current library items. For example, the draggers were designed in include graphics, and custom hit areas.

### Code Organization

I tried to separate the code from the library items. This makes it easy to work on the code without having to jump back to Flash. Trying to be productive.

# Assumptions

Essentially, this is a drag and drop without the dragging. Users would need some prompt to indicate the interactivity. I also assumed that if built properly, the base AS3 and XFL could be expanded to include additional functionality without having to rebuild the page.

# What would I do or change if I was given more time on the project?

* Add more colors.
* Allow users to drag color swatch.
* Add question text prompting the user to perform a task using scenarios, etc.
* Use graphics instead of colors
* Place dynamic content in xml or json. Instead of hard coding in AS3.
* Separate the code more. Use a MVCX framework like RobotLegs.
* Record user click and interactions. This could then be saved to the LMS.
* Add visual depth to the graphics.
* One user interaction, have graphics visually indicate they are being clicked.
* Use transform3D to animate clicks.
* Add transitions when content appears on the screen.