Web Multimedia Lesson 2

AccordianSlider Assignment – jQuery text effect and Accordian Menu animation

*Remember that you should not be changing the content of the .html or .css files to accomplish these effects. They should be achieved via jQuery code.

Part 1: The War Text Effect

Create a text effect (in a JavaSript file named **warTextEffect_yourUserName.js**) on the p.textDrop element's text content ("The War") using jQuery such that:

- The text begins scaled up to match what you see in the video specs (as closely as possible). It should be only 20% opacity and may need to be vertically positioned to be where you see it in the specs after scaling.

- Animate the text down into place by resetting the values of the text's opacity (80%), vertical position, and scale (transform – the ***transform2d*** plugin provided to you allows you to set the CSS transform property) over a duration of 1 second. It should use an easing value that causes the animation to begin slowly and accelerate toward its finish. Check out the provided easing plugin's code to see the possible easing values.

- After the above animation is complete do the following:

    ✓ Show the div#dustPuffs element and do the following with it:

        - Grab the 'span' *direct children* of div#dustPuffs and *chain* on a method to iterate through them one at a time doing the following to span.dustPuffs:

            - Using ***this*** to grab the currently iterated span.dustPuffs element, animate its opacity to 20% and scale it in the x-axis to 300% and in the y-axis to 200%. This animation should take 1.5 seconds and simulates a puff of dust created by the text falling into its place on the page.

            - After this flying dust animation completes, fade out the div#dustPuffs element over 800 milliseconds.

            - Once this fade out animation completes, immediately reset the current span.dustPuffs element's opacity to 100% and its scaleX and scaleY back to 100%. Also, animate "The War" text to the left over a duration of a half-second and using an easing value that gives the effect shown in the video specs.

- ✓ Animate "The War" text such that it bounces once slightly up and to the left before settling back into the place it fell down to (as shown in the video specs - look closely).

Part 2: Accordian Menu

Write a jQuery plugin (**jquery.accordianMenu_yourUserName.js**) to create a new jQuery method (named ***accordianMenu***) that creates an accordian menu animation effect as shown in the video specs.

Use the standard jQuery plugin design pattern we have explored together in class.

It should create a custom object named AccordianMenu with the following items:

- ✓ Create a constructor function for this AccordianMenu object that accepts any passed-in options from the method call (in a formal parameter named options) and the DOM element object that was selected in the jQuery object that called this plugin method (near the end of the .html file - div#subMenu).

  The constructor function should:

  1. store the passed-in element in a property named $menuDiv

  2. call a method of the new object named _init() passing it the passed-in options

- ✓ Assign an object literal containing the default options values to a property of our $.AccordianMenu object named ***defaults***. The object literal we assign should have the following property: value pairs:

  speed = 250
  easing = 'ease'
  defaultItem = 0
  menuWidth = '415px'
  sliceWidth = '90px'

- ✓ Use the new object's prototype object property to create the following methods for our new $.AccordianMenu object:

  - **_init**

    Parameters: options

Purpose: Create properties for our new object and call this object's _clickHandler() and _openItem() methods.

Create the following properties:

Property name	*Property Value*
options	should be an object formed by deep-merging our object's defaults with any passed-in options
$menu	jQuery object containing all direct 'ul' tag children of our $menuDiv property
$menuItems	jQuery object containing all direct 'li' tag children of our $menu property
$imgWrapper	jQuery object containing all direct 'a' tag children or our $menuItems property
$menuItemsPreview	jQuery object containing all direct children tags of our $imgWrapper property that have a class of 'menuPreview'
totalItems	should contain the number of elements contained the $menuItems property ('li' tags which are our menu's image slices)
currentIndex	will represent the index of the currently active 'li' image slice. Initialize to zero here
isAnimating	Boolean value which will indicate whether a menu animation is in progress (true) or not (false). Initialize to true

- **_validCurrent**

Parameters: none

Purpose: return true if the currentIndex property represents a valid menu item (slice) which means it is greater than or equal to zero OR less than the total number of menu items (slices).

This should be done in a single return statement using a ternary operator for the if-else logic.

➢ _openItem

Parameters: openedIndex - represents index of menu item that should be open on page load

Purpose: Open the menu items (slice) represented by the passed-in index value. Do this simply by using jQuery's eq() method to select the menu item to be opened from the $imgWrapper property of our object based on the passed-in openedIndex parameter value.

Then, using chaining, simulate a click event to use our existing code to open that menu item.

This should all be done in a single statement.

➢ _clickHandler

Parameters: none

Purpose: Will handle any click's on our 'a' tags that are within the 'li' image slices. The logic flow should look like this:

Remember our new object (***this***) in a variable named ***self***.

Set up a click event handler using jQuery on each of the 'a' tags contained within our new object's $imgWrapper property. The click's event handler function should do the following tasks:

- Get and store the parent 'li' tag for the clicked 'a' tag in a jQuery object variable named $parentLI
- Find and store the index of this 'li' parent in relation to its sibling 'li' tags in the DOM. Store the index in a variable named clickedIndex.
- If the currentIndex is identical to clickedIndex, then the clicked 'li's link panel is currently open, so
    - slide it closed by calling our _slideItem() method AND reset currentIndex to -1 (meaning that no li's are open any longer).

      Note that you should pass the jQuery object containing the 'li' parent of the open panel's 'a' tag, set the state to false (closing - see _slideItem()'s documented parameters below), duration of 1.5 seconds, easing to

'easeOutQuint' and true for last parameter to indicate all panels are now closed.

- Else... currentIndex is NOT identical to clickedIndex, so

    - close any open 'li' by calling _validCurrent() to see if it is a valid index and, if it is, by calling _slideItem().

      Note: pass the the jQuery object containing the 'li' in $menuItems at the currentIndex, set the state to false (closing), duration to a quarter of a second, and easing to 'jswing'. Do not pass a value for the allClosed parameter.

    - Next, reset our currentIndex property to be the same as the clicked index and slide open the clicked 'li' by again calling _slideItem().

      Note: pass the jQuery object containing the 'li' parent of the open panel's 'a' tag, set the state to true (opening), duration to a quarter of a second, and easing to 'jswing'. Do not pass a value for the allClosed parameter.

- Prevent the browser from performing the default action of following the link that was clicked.

➢ _slideItem

Parameters:

$panelSlice	jQuery object containing the affected 'li'
state	tells us if we are opening (true)/ closing (false) the 'li'
speed	duration of the opening/closing animation
easing	easing formula to be used in the animation
allClosed	true only when all 'li's are in a closed state in which case we'll set opacity on all 'li's to 1

Purpose: Slide a panel slice ('li') open/closed based upon the parameters that are passed in.

Incorporate the following tasks:

- Store a reference to any 'span.menuImage' (span with color image background) that is a descendant of the passed-in $panelSlice's (clicked 'li') matched DOM element object in a variable named $colorImage.

- If we are to open the 'li' in question, then create an object literal which sets the width attribute to our menuWidth's option value storing the object in a variable

named bwOption. Also, create an object literal that sets the left attribute to '0px' storing the object in a variable named colorOption.

- Else (we are closing the 'li' in question), create an object literal which sets the width attribute to our sliceWidth's option value storing the object in a variable named bwOption. Also, create an object literal that sets the left attribute to our sliceWidth's option value storing the object in a variable named colorOption.

- If we are to open the 'li' in question, then stop any currently running animations on any 'span.menuPreview' elements (have sepia image as background -- hint - we already have these selected and stored in one of our object's properties) and then animate the opacity on these elements down to 10% over a duration of 1 second so they dim.

- Else if all 'li' slices are now closed (hint: think of using one of the parameters to check this), then stop any currently running animations on any 'span.menuPreview' elements and then animate the opacity on these elements down to 100% over a duration of 1.5 seconds so they brighten.

- Now, you will animate the slice open/closed based on what bwOption's width attribute was set to above. Stop any currently running animations on the passed-in jQuery object containing the affected 'li' and then animate its 'li' to expand or collapse based on the what its width attribute is set to in the bwOption object. Use your speed and easing option values to set the duration and easing for this animation.

- Next, slide the color image in or out based on the value of the left attribute set above in the colorOption object. This should be done by stopping any currently running animations on the $colorImage object and then animating its left attribute based on colorOption and using your speed and easing option values to set the duration and easing for this animation.

- Finally, if opening a slice panel, animate the color image's opacity to 100% over 2 seconds. Else, if closing a panel, immediately set the color image's opacity to 20% using jQuery.

Deliverable: Zip up your entire project into a folder named AccordianSlider_yourUserName and submit to the provided dropbox in E360.