Water Game Design Documentation

Draft Version 1.1

# Technologies used in this project

HTML

Structuring content

CSS

Applying styles for the visualization

JavaScript

Customizing interactive functionality

The JavaScript libraries CreateJS (http://createjs.com/, createjs-2015.11.26.min.js), jQuery of (http://jquery.com/, version 3.1.1 of jquery.min.js), and jQueryUI (http://jqueryui.com/, version 1.12.1 of jquery-ui.min.js) are used in this project.

# Software Tools used in this project

Adobe Animate CC 2015.2 Release. (Please note that different version of Adobe Animate will publish different file of waterscenario1.js.)

Adobe Photoshop CC 2015.5.1 Release

Notepad++ v7.4.2 (text editor)

WinDiff Version 6.2 (text file comparison tool)

Python 2.7.11

Atlassian SourceTree Version 2.1.10.0 (source control software)

# File Structures and Summaries

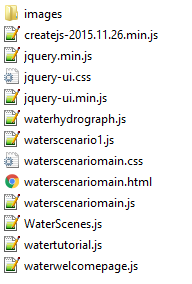
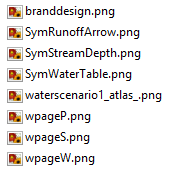


Image files in the folder “images”:

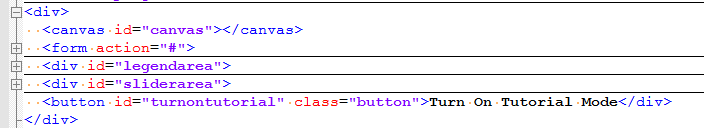


The files createjs-2015.11.26.min.js, jquery.min.js and jquery-ui.min.js (plus jquery-ui.css) are the local copies of the three third-party JavaScript libraries from CreateJS, jQuery and jQueryUI. The files waterscenariomain.html, waterscenariomain.css and waterscenariomain.js are corresponding to the three technologies illustrated in the previous section of “Technologies used in this project”. WaterScenes.js defines three computer simulations (Water Table, Precipitation, and Soil Texture) and tutorial steps. As their file names indicate, the files waterhydrograph.js, watertutorial.js and waterwelcomepage.js are related to these three components of this game. The file of waterscenario1.js is automatically generated by Adobe Animate via its Publish functionality. This file is responsible for the animation of the top portion of the game, marked in the red rectangle in the next section of “Layout”. The image file waterscenario1\_atlas\_.png is also automatically generated at the same time as waterscenario1.js.

# Layout (waterscenariomain.html)

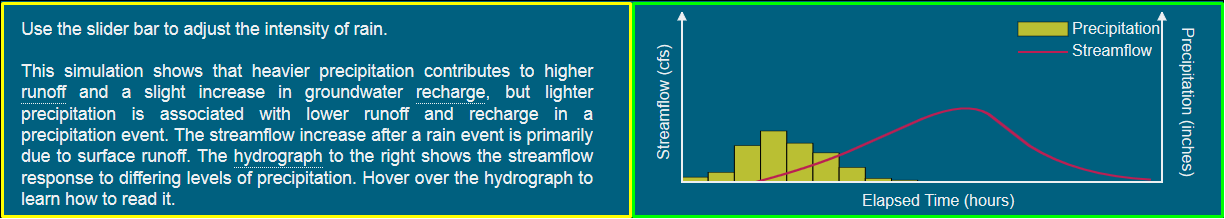
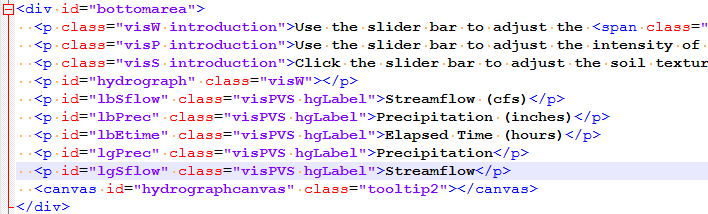


## Components of the top portion of the game

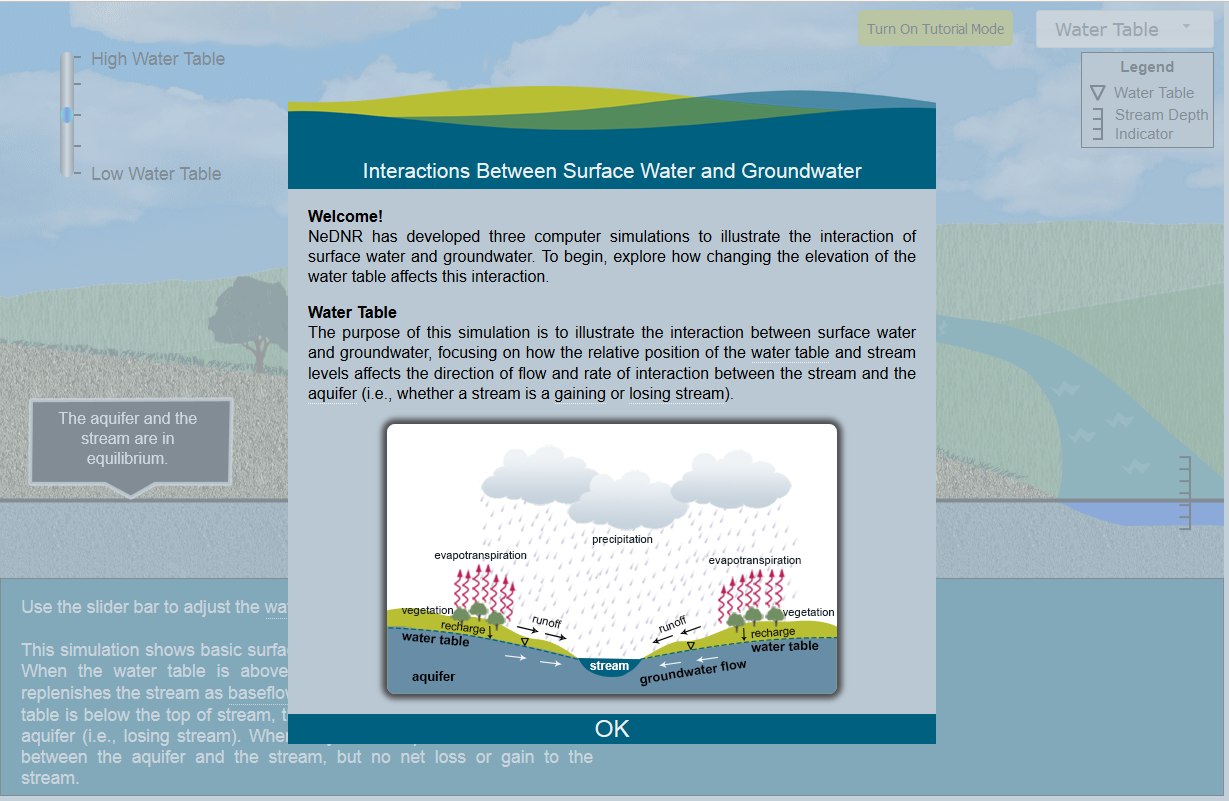


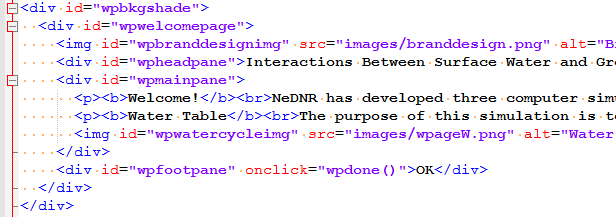
Where the “form” element corresponds to the pull-down menu of three simulations (Water Table, Precipitation, Soil Texture). The elements of “legendarea”, “sliderarea”, and “turnontutorial” (button) are very obvious. The “canvas” element is for the rest animation stuff in the red rectangle (see the generated file of waterscenario1.js).

## Components of the bottom portion of the game

 Where the introduction “p” element corresponds to the bottom left area of the game, marked in the yellow rectangle. We use the visual class attributes of “visW”, “visP”, and “visS” to control which simulation introduction paragraph to show. The bottom right area of the game, marked in the green rectangle is for hydrograph drawing. Another canvas element “hydrographcanvas” is used to draw the red streamflow curve, yellow precipitation bars and their legend (see waterhydrograph.js). All the labels are done by HTML5 “p” elements to obtain the clear visual effects.

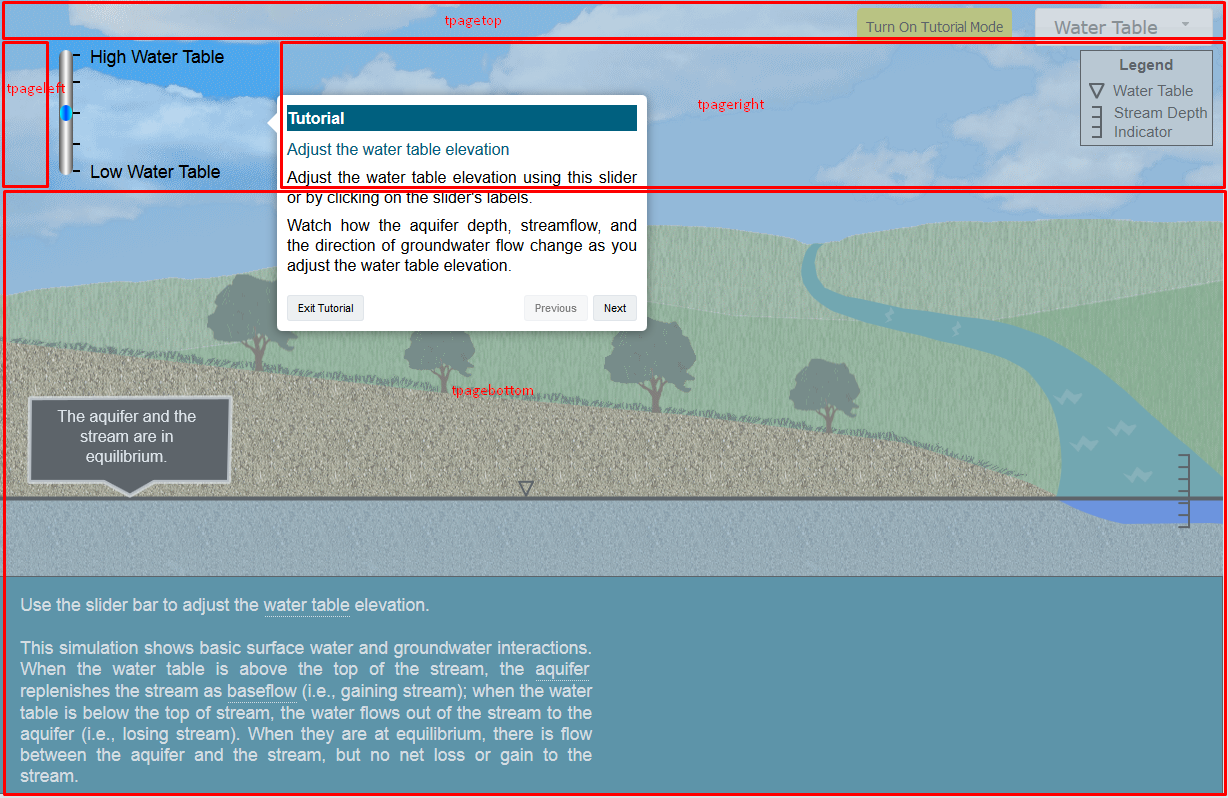
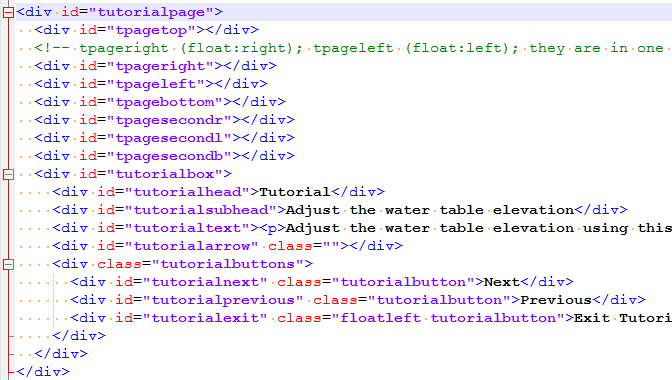
## Components of the Welcome page



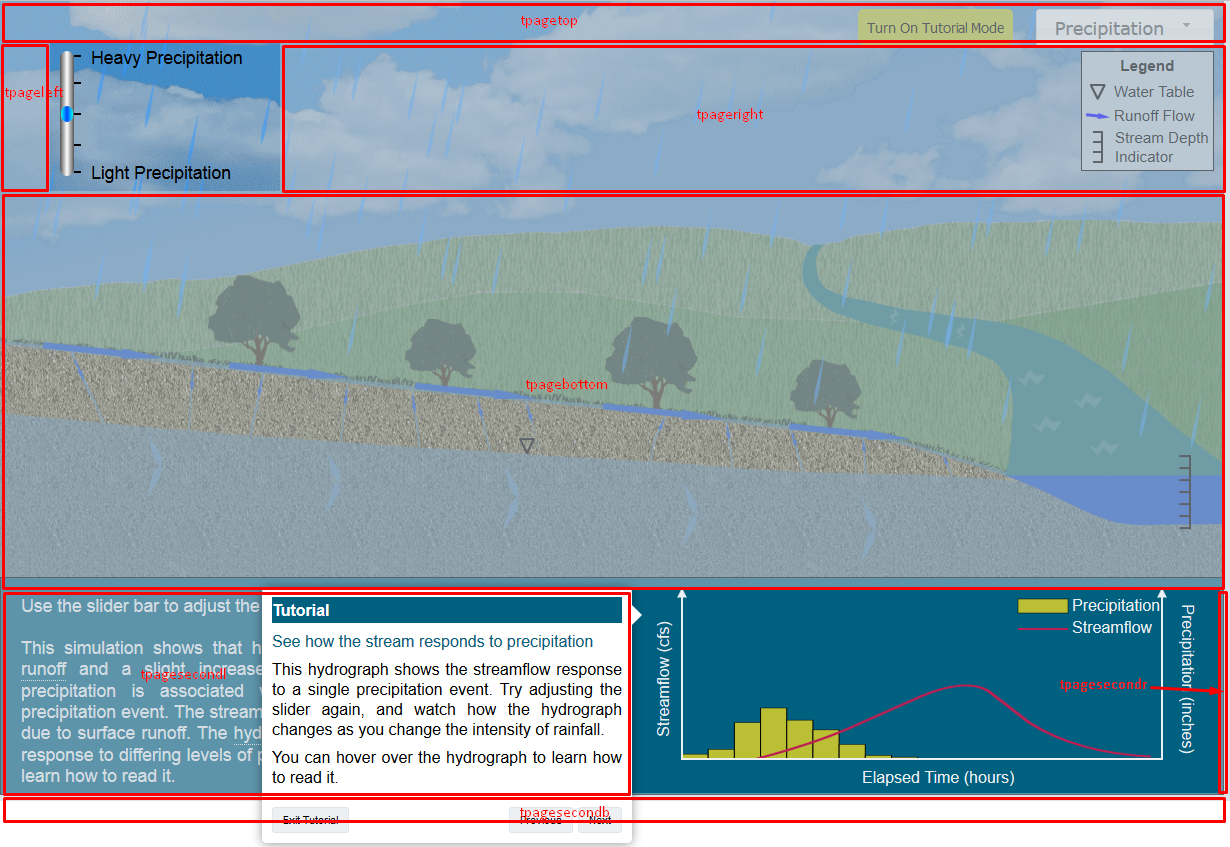


We put a background shade between the game and the welcome page to make users focus on the welcome page at this time. The implementation is pretty straightforward (see waterwelcomepage.js).

## Components of the Tutorial page

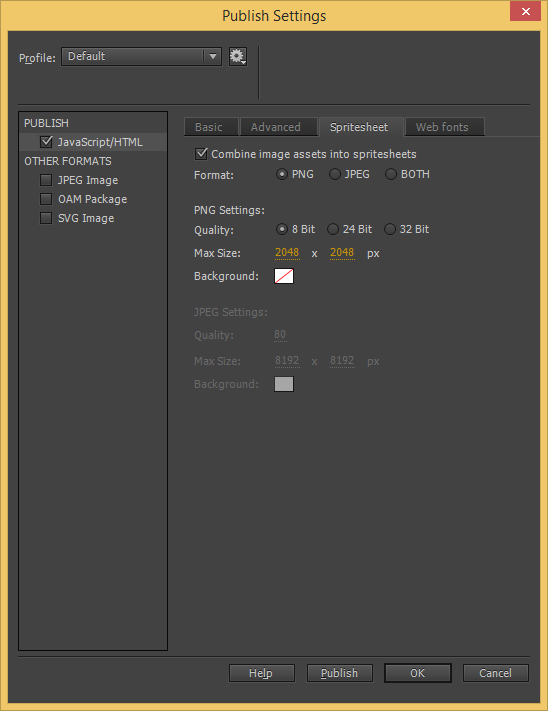
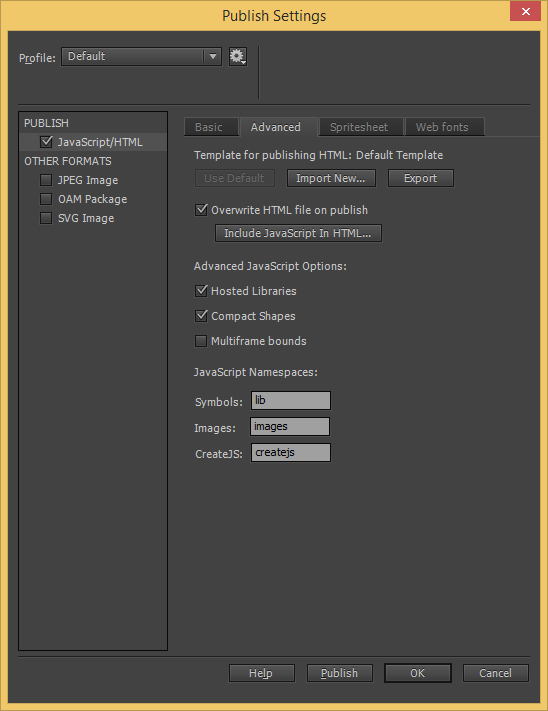
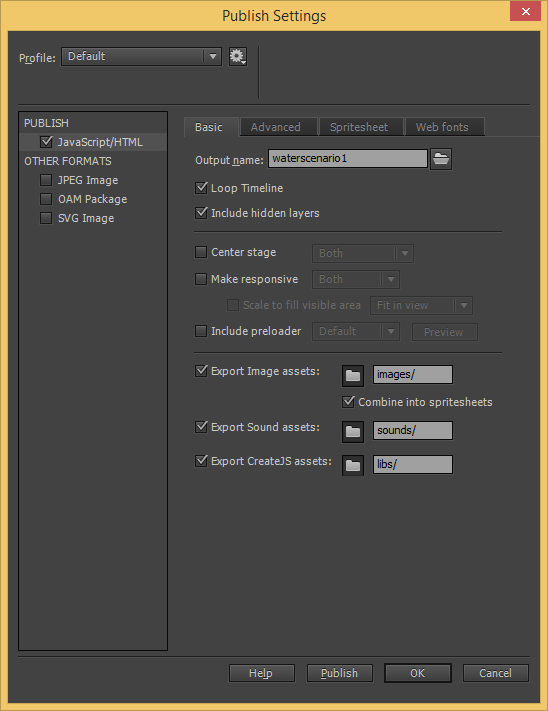
 We also put between the game and the tutorial box (id=“tutorialbox”) a kind of background shade with a rectangle hole in which uses can play this rectangle part of the game. As indicated in the above figure, the background shade consists of four rectangle blocks “tpagetop”, “tpageright”, “tpageleft”, and “tpagebottom”. For different locations of one rectangle hole, you just need to adjust the width and height of the four rectangle blocks (see the functions of “showtwholesliderbox()”, “showtbottomleftbox()”, “showthydrographbox()”, “showtmenubox()” and “showtendbox()” in watertutorial.js).

In the case of two rectangle holes, the rectangle blocks “tpagesecondr”, “tpagesecondl” and “tpagesecondb” come to help (see the function of “showthydrographbox()” in watertutorial.js). The layout of these rectangle blocks are illustrated as follows:

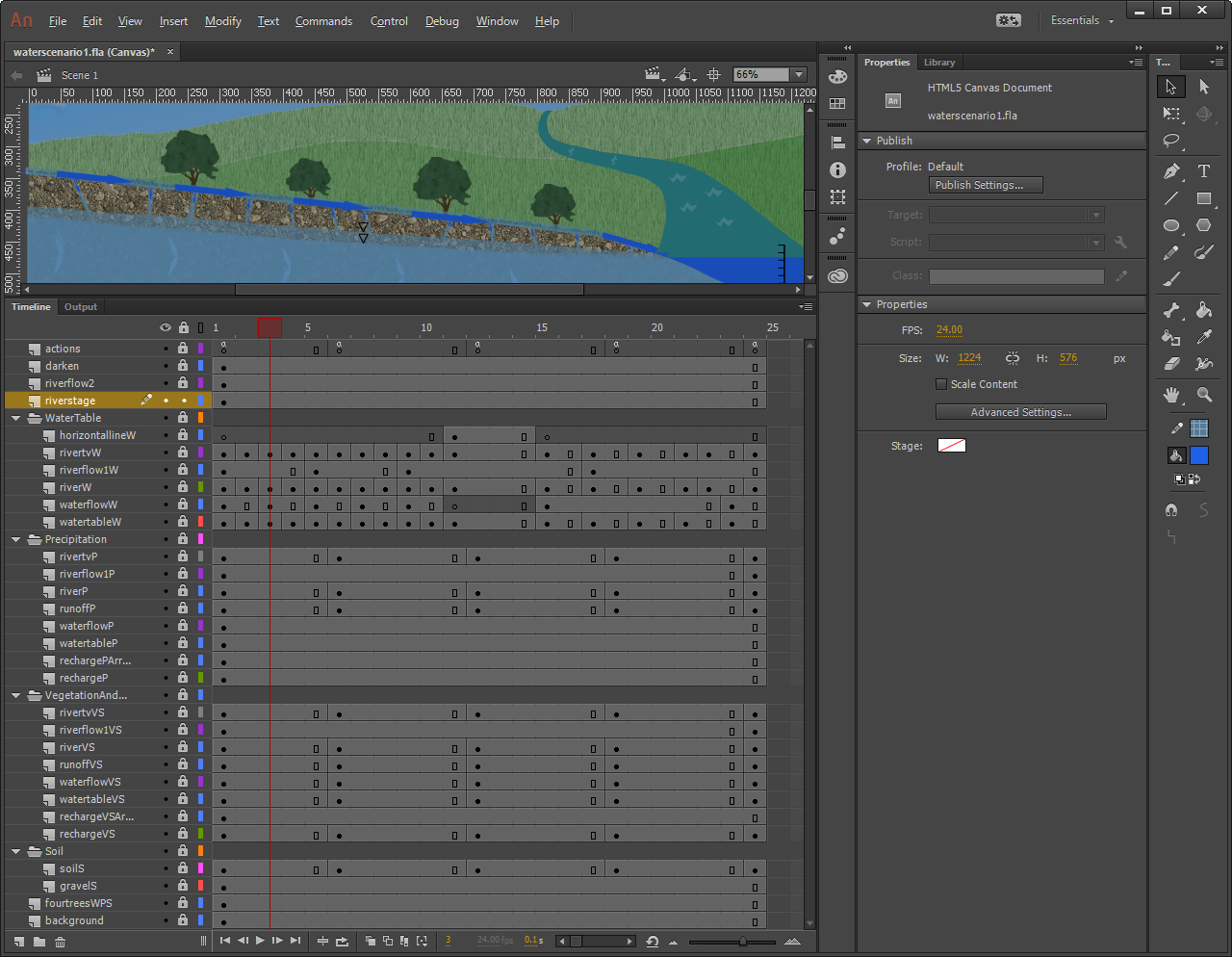


# File Generation (waterscenario1.js)

There are some other working files that help generate the final version of waterscenario1.js. The first working file is waterscenario1.fla, an animation project created by Adobe Animate. We will explain this project file in more details in the next section. Open the file waterscenario1.fla using Adobe Animate and select “Publish Settings…” from the “File” menu. Use the following publish settings and click on the “Publish” button. It will generate two files: waterscenario1.html and waterscenario1.js. The main content of the file waterscenario1.html has already been integrated into the file waterscenariomain.html and you can safely ignore the generated file waterscenario1.html. We are using a python script to read in the published file of waterscenario1.js and inserted some manually-coded JavaScript code into it. The python script is still writing out the same file, i.e., the previously published waterscenario1.js will be overwritten with some added content. So the second working file is this python script called modwaterscenario1js.py. The third one is called manualcopycode.js which contains JavaScript code regarding rain dropping.

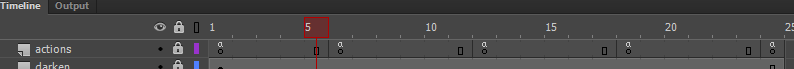


# Adobe Animate project file (waterscenario1.fla)

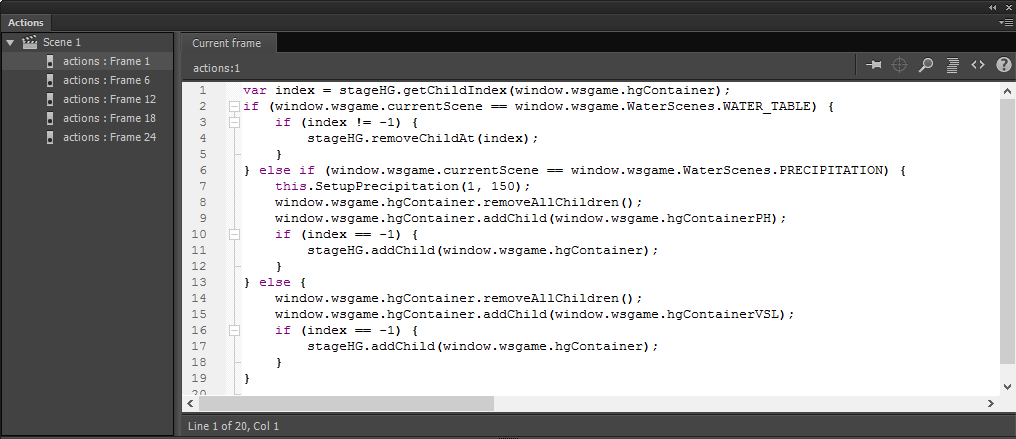


Please read the “Layers” section at https://helpx.adobe.com/animate/using/time.html. There are many layers and layer folders in this project file. The layer folder of “WaterTable” contains those layers that only apply to the simulation of “Water Table”. So does the layer folder of “Precipitation”. The layer folders of “VegetationAndSoil” and “Soil” contain those layers that apply to the simulation of “Soil Texture”. The other layers outside of these folders apply to all three simulations. When we publish the project, all these layers will be part of lib.waterscenario1. The names of layers will become simple comment statements in the file of waterscenario1.js, for instances, “// darken”, “// riverflow2”. For those layers that apply to the simulation of “Water Table”, we only need to execute their corresponding JavaScript statements in the lib.waterscenario1 part. So the Python script will add an “if” statement to execute them when the current scene (simulation) is “Water Table”. Similar for the other two simulations of “Precipitation” and “Soil Texture”.

## Layer of “actions”

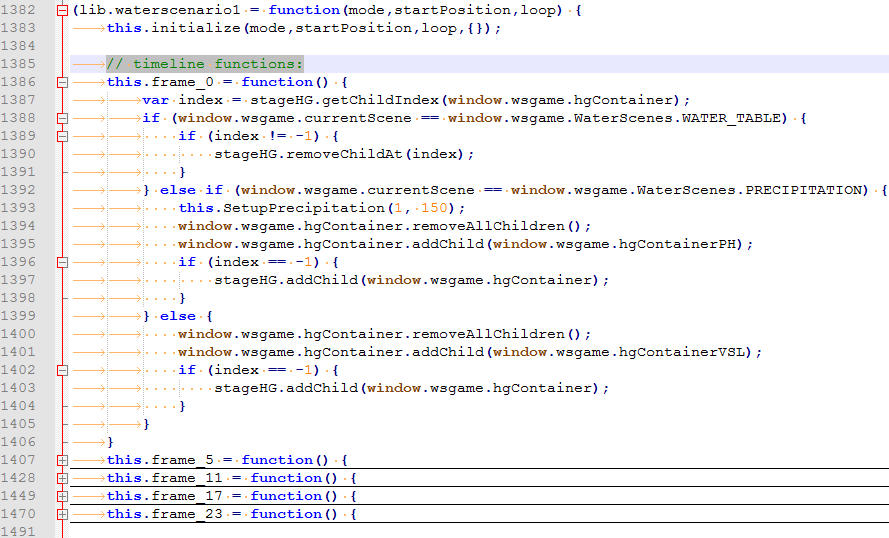


The layer of “actions” above contains five key frames which correspond to the five stop positions of the slider in the UI, i.e., the five different levels of water table elevation.



From the “Actions” panel above, we can see some JavaScript codes are added to these five key frames.

In the generated file of waterscenario1.js, we can find these codes under the comment statement of “// timeline functions:” in the definition of lib.waterscenario1 (see the picture below). Please note that frame numbers in waterscenario1.js start at 0 instead of 1.



# Selected Functions explanation

The functions init(), handleFileLoad(evt), and handleComplete(evt) in the file of waterscenariomain.js are automatically generated by Adobe Animate via its Publish feature. They are originally located in the <script> section of waterscenario1.html and are moved to the file of waterscenariomain.js since the file waterscenario1.html gets discarded. We have added to these functions the initialization of some variables and status, for instance, window.wsgame, wsgame.currentScene, wsgame.currentStep, wsgame.rainingTicker, etc.. The function init() will get called when the page of waterscenariomain.html has completely loaded all content including images, script files and css files.

The function initHydrograph() in the file of waterhydrograph.js is called from within handleComplete(evt). Since it is related to the handling of canvas “hydrographcanvas”, most of the codes are similar to the function “handleComplete(evt)” which handles canvas “canvas”.

The function drawHydroGraphs() in the file of waterhydrograph.js is called from within initHydrograph(). All hydro graphs are drawn (created) at the time of initialization and saved in CreateJS containers. There is a current working container called wsgame.hgContainer. According to the different combination of scenes (simulations) and slider positions, a corresponding hydro graph container will be added to the current working container wsgame.hgContainer. The wsgame.hgContainer will be in the display list of Stage (stageHG) of canvas “hydrographcanvas” in the simulations “Precipitation” and “Soil Texture”. Only the contents of containers that are in the display list of canvas Stage will get displayed in that canvas. So hydro graphs will be displayed in the simulations “Precipitation” and “Soil Texture”, not in the simulation “Water Table”. See those “// timeline functions:” in the file of waterscenario1.js.

At the bottom of the file of waterhydrograph.js, a jQuery plugin function of tooltip2text() is defined. It allows us to add the tool tip to the canvas of “hydrographcanvas” when the mouse enters the canvas. Usually we use the following format to add tool tip to a specific word:



The functions in the file of waterwelcomepage.js are quite straightforward.

The functions in the file of watertutorial.js are no big deal after the layout of those rectangle blocks is clarified in the section of “Components of the Tutorial page” above.

# Source Control repository

https://jhogan10@bitbucket.org/jhogan10/waterscenario.git

# References

1. http://createjs.com/
2. http://jquery.com/
3. http://jqueryui.com/
4. https://www.w3schools.com/html/default.asp
5. https://www.w3schools.com/css/default.asp
6. https://www.w3schools.com/js/default.asp
7. https://blogs.adobe.com/creativecloud/html5-canvas-javascript-basics/ for rain drops.
8. http://www.adobe.com/products/animate.html
9. NeDNR Brand Book for RGB color selection
10. Beginning HTML5 Games with CreateJS, book by Brad Manderscheid
11. JavaScript: The Good Parts, book by Douglas Crockford
12. https://www.python.org/