

HW7 – Canvas

Overview

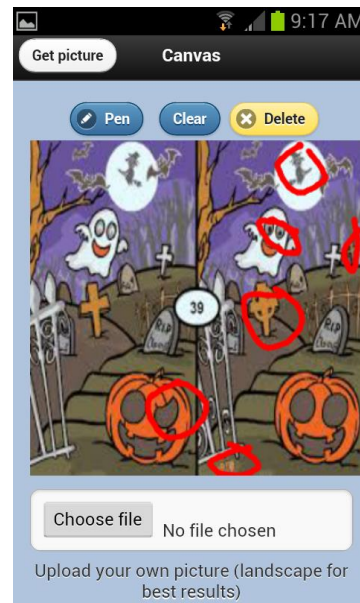
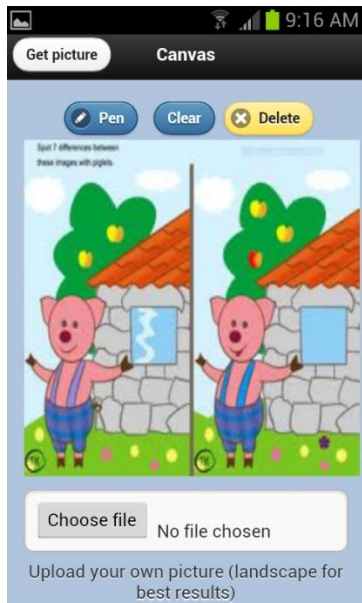
To demonstrate using canvas in a mobile app, that responds to touch events, I made an app where you can draw on the canvas, lay pictures on the canvas and draw over them, and upload pictures from your mobile device to draw on them too. The app is best used on mobile devices, since desktop browsers don't usually have touch function. It can be found at the following url:

<https://mobile-bihequ.backliftapp.com/draw/index>

To start off, the color of the lines used to draw can be changed by pressing the button on the left, labeled “pen.” Then clicking a color will set the fill and stroke to that color.



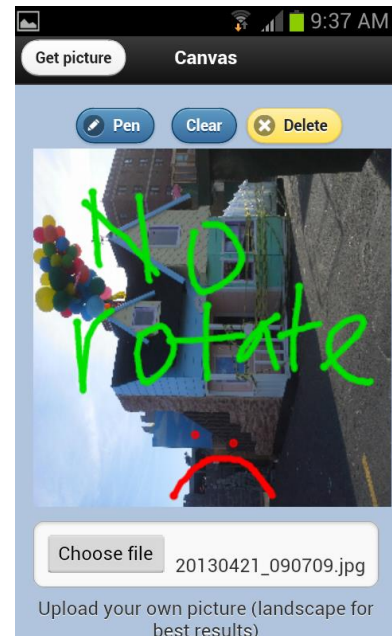
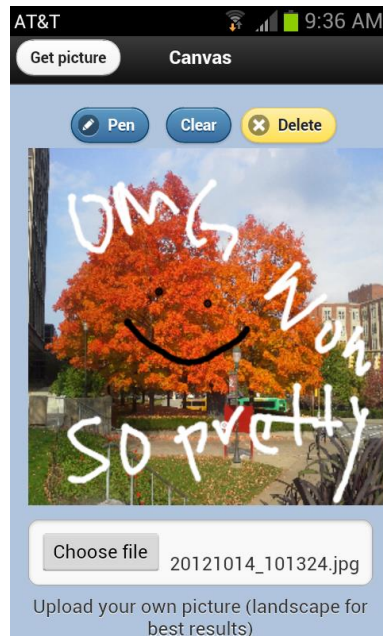
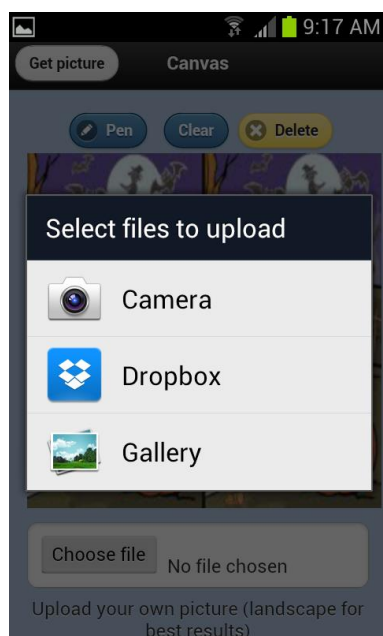
The “get picture” button in the header will load a picture onto the canvas so you can draw on it. These are just “spot the difference” Google images, nothing fancy. The paths of the images are saved in a json file, and the images are kept in a file. So the getJSON call loads an array with the paths, and when the button is clicked it sends a request to the given image in the array. You could add more pictures, just update the json file and fix the modulus number in the “get picture” logic.



Clicking “clear” will erase all the pen strokes on the canvas while leaving the “background” the same. It does this by saving the background image, and just posting that back to the canvas on the “clear” click.

Clicking “delete” wipes the entire canvas to leave a white space. It also resets the “get picture” order so the first picture of the json file gets returned upon clicking “get picture” again.

You can also upload your own picture to the canvas to draw on it. This can be uploaded via a file on your mobile device or via a camera. My javascript takes the image file when the user uploads it to the input, and gets the image via a XMLHttpRequest. Then it is posted to the canvas like the other pictures.



There is a small problem with this, however. On mobile browsers, at least all the ones I have tried, uploading a vertically orientated picture will post it sideways on the canvas. I tried to fix this by rotating the canvas, drawing the image, and restoring the canvas, but I found out most mobile browsers don't support the `canvas context.rotate()` method. So I just prompted the user to upload a horizontal image (problem solved!).

Using Canvas

The page, like the example given in class, loads a canvas upon the page loading that updates based on touch events and image postings. The canvas is 300x300, which sort of messes up the ratio of certain pictures, but it needed to be kept small enough to fit within a mobile browser because scrolling can mess with touch-drawing, seen in the next section.

Using Touch Events

Again, like the draw example, the touch events on the canvas draw lines where the user touches. One thing I found when doing this was that if the page is scrolled down, it can throw off the coordinates of the touch listener, so I fixed this by making my page smaller so that there is no scrolling involved.

Doing Something Significantly Different from draw.html

Alright, I know the base of my application is basically the same as the draw.html example, but I learned a lot about drawing images to the canvas, and using the context to position images. Also, I have different colors, so that's different enough, right?

Cache Application for Offline Use

I basically cached all of the files in the draw.manifest, although I've found that the only thing affected by having no internet connection is loading the included pictures. Since most of my application is already loaded on one page, nothing else needs to be loaded in, except outside pictures (but not user uploads). So I cached those bad boys and now we're all good.