Behind the scenes of Garden Party

CSS Zen Garden is one of the best showcases on the Internet explaining the power of CSS design. The variety of designs that different designers have created are inspiring and craftsful, especially when you consider they all had the same instructions and limitations. Even more Zen Garden invites web designers and graphical artists to contribute. I decided to accept the challenge and was one of the happy ones when my design got picked out. Hereby my story about the process and thoughts behind Garden Party.

Listening to what the client wants

Dave Shea is the one that picks out the designs, so for this project he was my client. I started reading the instructions and CSS resources pages and filtered out the requirements and tips of other designers. I remembered that I had read somewhere that Dave aimed to help CSS get of that boxy image. Because I am one of those characters that believes that you can create any layout you can think of with XHTML and CSS, I decided I would use rounded shapes.

Looking at what the competition did

Next I looked at what other designers had created and focussed on the stronger and weaker points of their designs. Most of the official designs had very strong concepts and a few contained technological innovations. As a weaker point I noticed that in some designs I had to do effort to find the next design, for me one of the main functions in Zen Garden. This made me decide to put the main focus on the designs and their designers.

Going through the designs enabled me to see what already was explored; you want to be original after all. I noticed that a lot of designs only contained one column containing the main content, which made them look a bit stiff to me. I immediately felt challenged to create a two columns layout for the content. Next I browsed through the non-accepted designs, trying to find out why they weren't picked out.

Creating the concept

One hard lesson I have learned is that you should never ever start any design without a good and clear concept. After you have created a concept it is very easy to make design decisions: it either fits in or it doesn't. To get some inspiration I started to flip trough some magazines, books and pictures, with the words 'zen' and 'garden' echoing in my mind. Soon I had found some photos of the birthday party of my little niece last summer, of which most were taken in the garden. I figured that 'Garden Party' seemed like a nice working title.

Thinking about parties my mind wandered of to house parties and the posters that decorate the cities to invite people to go out. I pictured monochromatic colors, the names of DJs in big attracting fonts and I thought it would be a great idea to bring something like this to the web. And it made sense; in Zen Garden the designers are the DJs and the designs the music played.

Because I had already decided to use rounded shapes I thought a retro look-and-feel would fit nicely. A good concept was born in a very short timeframe and I felt very happy.

Creating the graphical design

In the next step I got pencil and paper and started drawing different designs. Even on paper I always draw a few guides so I can already picture how things will look like in a browser window. After I had created a few different designs I was happy with, I picked out the one I thought was best.

Next I opened Photoshop and set up my guides. I started drawing shapes and lines, manipulating pictures, playing with colour combinations, selecting and sizing fonts and laying everything out according to the sketched plan. I kept on weighing the different options and experimenting how elements fit in the best way until I was satisfied.

Designing the content flow

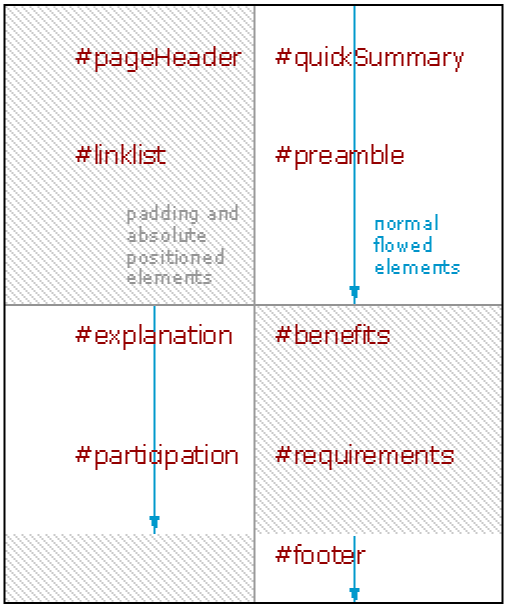
After the basic graphical design was done, I picked up pencil and paper and started designing the content flow. The design contained a flip-over that split my screen into two vertical pieces, so I drew two columns on the top half and two columns underneath it on the bottom half. I placed the most important elements like logo and designs in the left top column, the following elements in the second top column and so fort.

Because I had to use existing markup and content, I had to check if this page layout was feasible. So I clicked 'view source' to study the main page's structure. It looks like:

```
html
    body
        #container
            #intro
                #pageHeader
                #quickSummary
                #preamble
            #supportingText
                #explanation
                #participation
                #benefits
                #requirements
                #footer
            #linklist
                #linklist2
                    #lselect
                    #larchives
                    #lresources
            #extradiv1
            ..
            #extradiv6
```

Next I tried to CSS-position the different content blocks in my schedule. I figured out that #container should be positioned relative, so I could use it to position static, absolute and relative elements inside of it. I took #pageHeader and #linkList out of the normal flow by absolute-positioning them, together they formed the top left column. I decided to use extra left padding to make the normal flowed elements start in the top right column.

Under the flip-over I created the bottom left column by using the normal page flow and giving #explanation and #participation a smaller padding to the left. #benefits and #requirements would be taken out of the normal flow and be positioned absolute within the #supportingText block, which thus had to be relative. I ended the normal flow with the footer, by again adding extra padding to the left. Still with me? It looked something like:



Producing the design

Finally all the planning and preparations were done, so it was time to start working on the style sheet. I always hand code all of my style sheets in a regular text-editor using syntax color highlighting. I started bringing in the sliced images from Photoshop, building the layout layer for layer and positioning the content.

The first issue I had to solve was that my background required two different colors along the left and right sides of the centered content. Here I learned a new technique, which I actually had never seen before: the centered background image. It is very easy to center a background image with CSS:

```
body {
    background: #fff url(bg.gif) center top repeat-y;
}
```

I used a background image of 1600 by 100 pixels, the left half with one color and the right half with another. A very simple solution, especially for complex layered layouts.

I had to juggle a little bit to postion the #requirements block. I figured out that the #benefits block above it takes around 17.5% of the space, so I placed it at 17.5% from the top.

For font sizes I made the decision to use font size keywords and the box model hack to resolve keyword differences between Internet Explorer 5.x/Windows and other browsers. Keywords are one of the better ways to size your fonts, together with pixel sizes and ems (you can read more about font sizing at A List Apart).

Testing the design

For me testing starts and ends with validating code, in this case only the CSS. I developed Garden Party in Firebird 0.7, because it is one of the most standards compliant browsers available. So after validation it was time to open a range of other browsers; I tested in Internet Explorer 5.0, 5.5 and 6.0, Opera 7, Mozilla 1.4 and Netscape 7.1 on Windows XP and Safari 1.0 and Internet Explorer 5.2 on Mac OSX.

I squeezed out the CSS bugs and incompatibilities and found two weak points in my design. The different positioning schemes made my content look good in the default font size and scale up until a certain level, but would make content overlap at two places (the absolute positioned #linklist and #requirements blocks) while scaling down.

Because I didn't see another way to resolve this and my deadline was in sight I decided not to follow up on this scaling down issue. It was only after Garden Party went live that I realised that in some translations (e.g. Korean) the content overlaps at these two points.

Some closing words

My main advise to anybody that wants to create a design for CSS Zen Garden (or actually any design) is not just start right away, but think of a concept first and plan your steps ahead. Good luck!