Front-end development

Module 7

Prototyping, User Testing, and Design

Bits and Bots
study group

Content

This module

This module is all about practical ideas for a final project. You may have thought of something that you wish to create by the end of the year or you may wish to keep going on lots of little projects. There are a few different areas for you to focus on in this module and at this stage it should be considered more guidance than official learning to be used as is most useful for your project or your work.

In this module we will look a bit more at UX but through the lens of user testing and prototyping. However, if you want more of an overview of this subject area this is a good starting point for links and other resources.

Prototyping and User Testing

Coding can take a really long time and there isn't much point doing a lot of work only to find out later that that isn't actually what you wished to make in the first place! You also don't want to get to the end of the project and realise that it doesn't solve the problem that you had set out to solve. User testing and prototyping allows you to catch these issues early and change the plan before you've done lots of work on it.

Prototypes
Prototypes are quick unfinished versions of what you wish to finally create. They can contain key elements of the website to test on users to see if a concept works or be a full overview of the final product. You don't have to start with a screen to prototype. You can start by sketching your design out on paper, using post it notes and moving things around till you are happy with what you want to make. Prototyping using paper can also be useful for user testing or play testing anything you create before you start coding. When you want to try out your design on screen then it could be time to explore the concept of rapid prototyping.

User Testing
Assuming you have chosen to make a game as a final project. Taking advice from this excellent blog post by Matteo Menapace you could be asking:

- Is the game functional?
- Is the game fun?
- Is the game learnable?

Most of the creation of a game, or website, or almost any project is user testing. Here Matteo has chosen three things that are required for the games created. Fun, functional and learnable. But you could pick more things, less things and have an entirely separate list. By posing questions about the key things your project needs to do you can then begin to answer these during user testing.

Write out the key goals of your project, and then a set of questions like the ones above that you wish to be answered when testing.

The key is to not wait until it is perfect for others to try it out, the earlier that it can be tested the more time you have to improve your project and make it better.

Take a look at Matteo's blog post (which is also great for tips on creating speedy game concepts), this one on play testing of board games and also this practical guide to user testing. Some of what is being suggested is more complex than others and also for quite large scale projects, the principles, however remain the same. Try writing down your own user testing methodology and questions that match the time you have available and the scale of your project. Then give it a go and have fun with it!

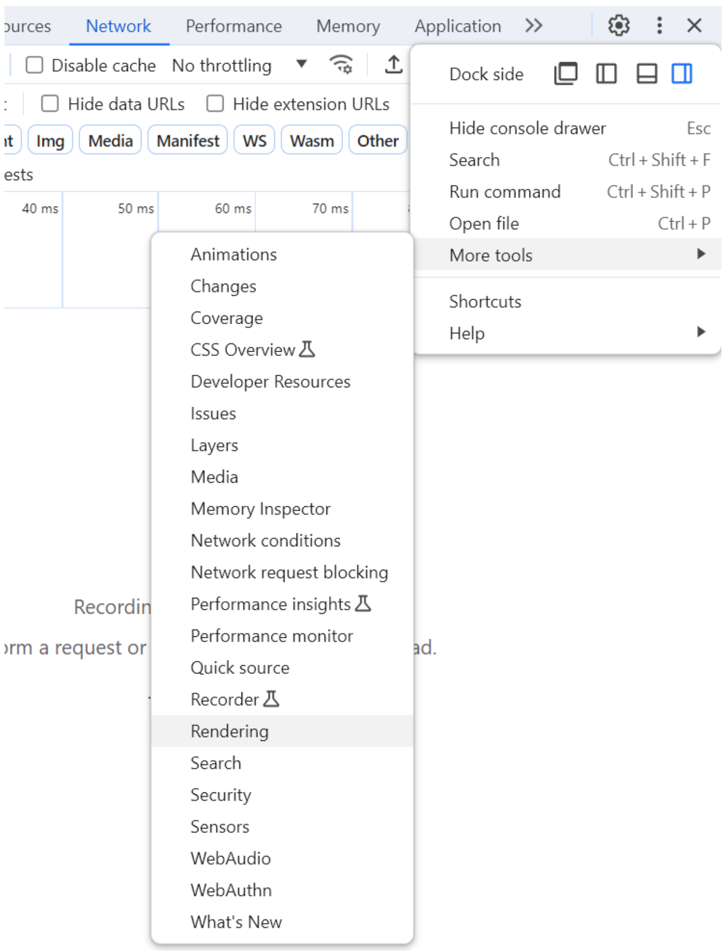
Accessibility

Accessibility is a huge area when it comes to website design and there is a lot of in depth information online. It is also possible that your organisation will have their own set of criteria to be measured against when they develop their websites, and often these will be assessed externally as well. For government websites in the UK, for example, these have to meet the AA of Web Content Accessibility Guidelines. Which may be worth a read if you wish to learn more in depth about this. However, for your own personal website this is a list of simple things to consider when designing and building your website, or to change in an existing one. I would suggest using this list as a checker for any of your web projects. Though this guide will talk about three of these a little more in depth.

Responsive Websites
One tip you may want to look at for a final project is to create a page that looks good both on the laptop you are developing it on, a mobile device or on a big screen. There are a number of small ways you can alter your code to do this, and chances are that if you are modifying other code then it already has been designed to do this. Common tips include setting the widths to be responsive to the size of the screen, for example not a set number of px but as a percentage. This page from W3 schools lists many of the ways that you can modify CSS to make your webpage responsive.

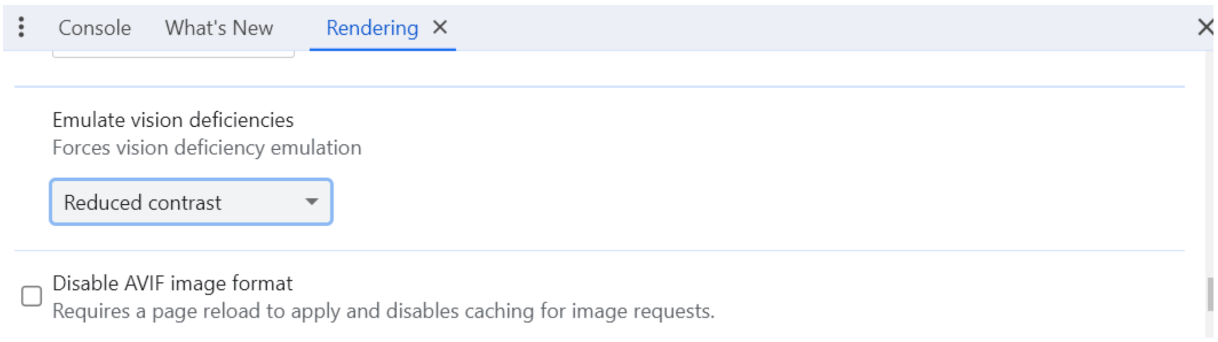
Alternative Text
Adding alternative text to images on your website means if your images cannot be displayed or your user is using a screen reader, that the intention behind the design can still be understood. ALT text can be added to your HTML easily as an attribute, as described here.

Using the Inspect Tool
If you haven't used the F12 button yet to investigate websites then you should! You can inspect a webpage either by right clicking and pressing inspect or pressing the F12 button. This allows you to view the source code of a website, make edits to see immediate results and quite a few other hacks and tricks.



One use for the inspect tool is to render your webpage to emulate vision deficiencies or highlight areas that may not be suitable for people prone to photosensitive epilepsy.

On your website enter the inspect area and click on the three dots. Then navigate to 'More Tools', 'Rendering' (as you can see in the screenshot on the right).
Once you have navigated here then at the bottom a selection of different rendering options become available.



Emulate vision deficiencies
Forces vision deficiency emulation
Reduced contrast

Disable AVIF image format
Requires a page reload to apply and disables caching for image requests.

This is a really helpful tool if you want to check that your colour choices, possibly chosen to stand out against each other, still do. Or that certain font choices can easily be seen.

There are lots of different features available with the inspect element. Look around and see what is most useful for you.

Next steps

At this stage in the course you have tried a lot of different elements of website design! The three different languages, lots of tricks in Twine, saw how you can embed different parts of other websites and tested out different platforms to host your websites. Maybe you really enjoyed learning JavaScript and you would like to spend more time building games using it, some inspiration could be found here. Maybe you would like to create more complex Twine games. Or maybe you loved using CSS and would like to work further on that. This course has brushed the surface of many different subject areas to give a wide understanding of a few different things. Hopefully you now have the knowledge and a few of the skills to move forward with web design!

Introduction to File Format Identification

This month our session will focus on **File Format Identification**. We will cover specific tools and the differences between them, such as **DROID** (Digital Record Object Identification) and **Siegfried**.
We will explore running these tools from the command line and integrating them with Python. We will also cover how to analyze file formats manually and how these tools function using information from file format registries (like PRONOM).
Finally, we will look at resources you can use when trying to understand the files you hold, whether on a personal computer or digital files in a collection.

Why is this useful?

Preservation: You cannot preserve a file if you don't know what it is. Identification is the first step in digital archiving.
Security: Malware often hides by masquerading as a harmless file type (e.g., an .exe renamed to .jpg). Identification tools look at the binary signature, not just the file extension.
Troubleshooting: When a file won't open, it is often because the extension doesn't match the actual format.

Pre-requisites

For this session it might be useful if you have access to a few tools.

1. A Hex Editor

You need a tool to view the raw binary data of a file, here are some tools that you can use, and others are available.
Windows: HxD
macOS: Hex Fiend
Linux: GHex or Bless

2. An Identification Tool

These exercises will focus on DROID and Siegfried. Siegfried has a more accessible command line interface whilst DROID can be more approachable with a GUI.

3. Access to sample files

We have placed some sample files in the Bits and Bots repository, GitHub is also an excellent resource for downloading sample files or you may already have some files sets that you wish to analyze on your computer. Some resources on finding sample files can be found in the PRONOM Starter Guide.

Some exercises to try out

These exercises are based extensively on the PRONOM Starter Guide and the monthly session (which will be made available online shortly after the session).

Pick one, a couple or all of the following exercises depending on your level of confidence!

1. Install one of the following on your computer and run it over a set of files: DROID or Siegfried. There are comprehensive user manuals for DROID online and Siegfried also has some information available online, but it requires a little more looking through GitHub. Take a careful look at the file set and answer the following questions:
    - What methods are the tools using to identify the files? Do they identify by extension, byte matcher/ signature or container
    - Are there any unidentified files to research further, what might the files be?
    - Are there any flags, extension mismatches or anything to be concerned about? For example did a file that is called image.png actually identify as a .jpg

If you are comfortable with the Command Line Interface (CLI), try installing Siegfried using your package manager.

Windows (using Scoop):
code Powershell
downloadcontent_copy
expand_less
```
  scoop install siegfried
```

macOS (using Homebrew):
code Bash
downloadcontent_copy
expand_less
```
  brew install siegfried
```

Once installed, verify it works by typing sf -v in your terminal.

Stretch exercise: What else can you do with the command line tool and Siegfried or DROID? Can you create any automated workflows for your files?

2. Research a file format

Are there any file formats in your collection that you can't identify, or do you have an area of interest you want to explore further. PRONOM also has a list of file formats that do not yet contain full descriptions or have signatures. If you want to take on a challenge and create a description for a file format in PRONOM, or (hard mode) attempt to create a signature for a file that doesn't have one then the PRONOM Research GitHub repository has all the tools and lists you would need in the readme. If you have your own files you would like to explore further then have a go following the steps in the Starter Guide.

3. Find a common pattern within the hex of a set of files.

For this exercise we will be looking at these files, Open the files and drag and drop these into your hex editor. Can you find any consistent patterns within the file format? Is this the “File Signature” or “Magic Number”? Can you work out how these files could be identified? For tips on using a hex editor and analysing files you can use the Starter Guide.

Optional Challenge: Python

For those comfortable with Python, try to write a script that mimics what Siegfried does. Create a python script that iterates through the **Sample Dataset**.
For example:

```
downloadcontent_copy
expand_less
    with open('filename', 'rb') as f:
    print(f.read(4).hex())
```

Create a simple dictionary (Key: Value) where the Hex Signature is the key, and the Format Name is the value.
Print out the format of the files based on your dictionary lookup.