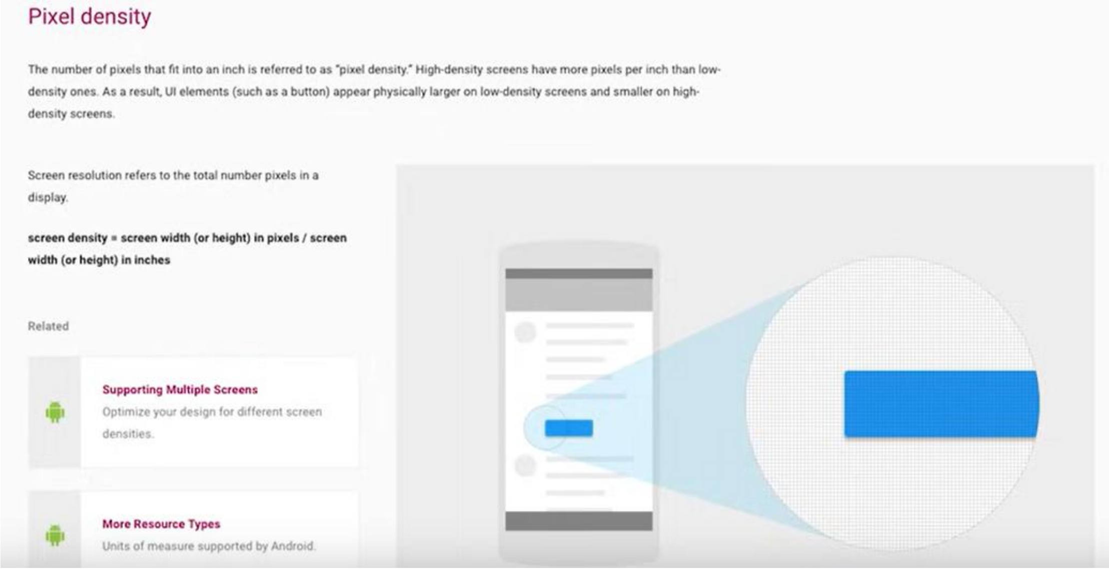
02 PROTOTYPING

Prototype design method:

Prototype design methods often begin with sketches and paper prototypes, which are simple and easy to share. Sketching is the best way to translate initial ideas into something tangible. The more details you add, such as button designs, text, interactions, and colors, the faster your progress will be.

In the past, my sketches were overly simple and lacked clarity in terms of details, interactions, and color schemes. This made it difficult for me to effectively communicate my ideas. As a result, I spent more time in the later stages of creating high-fidelity models, trying to fill in those gaps. This experience has taught me the importance of making the prototype as detailed and interactive as possible from the start.



Pixel density

The number of pixels that fit into an inch is referred to as "pixel density." High-density screens have more pixels per inch than low-density ones. As a result, UI elements (such as a button) appear physically larger on low-density screens and smaller on high-density screens.

Screen resolution refers to the total number pixels in a display.

screen density = screen width (or height) in pixels / screen width (or height) in inches

Related

Supporting Multiple Screens
Optimize your design for different screen densities.

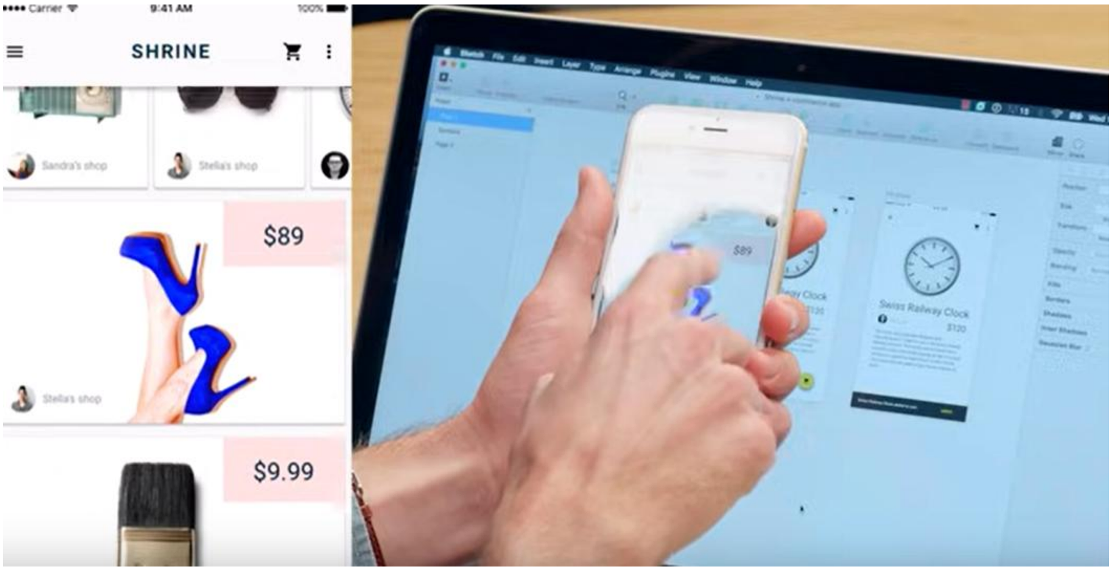
More Resource Types
Units of measure supported by Android.

To improve, I now focus on designing and simulating key user interactions, visualizing workflows, and testing important elements early on. Creating page mockups or foldable effects and capturing them with photos to turn them into interactive videos helps communicate ideas better. Additionally, I print existing screens and repeatedly sketch common elements, exploring smooth transitions between screens. I pay more attention to color choices, shadows, and the overall visual design to avoid visual noise and make the app more intuitive.

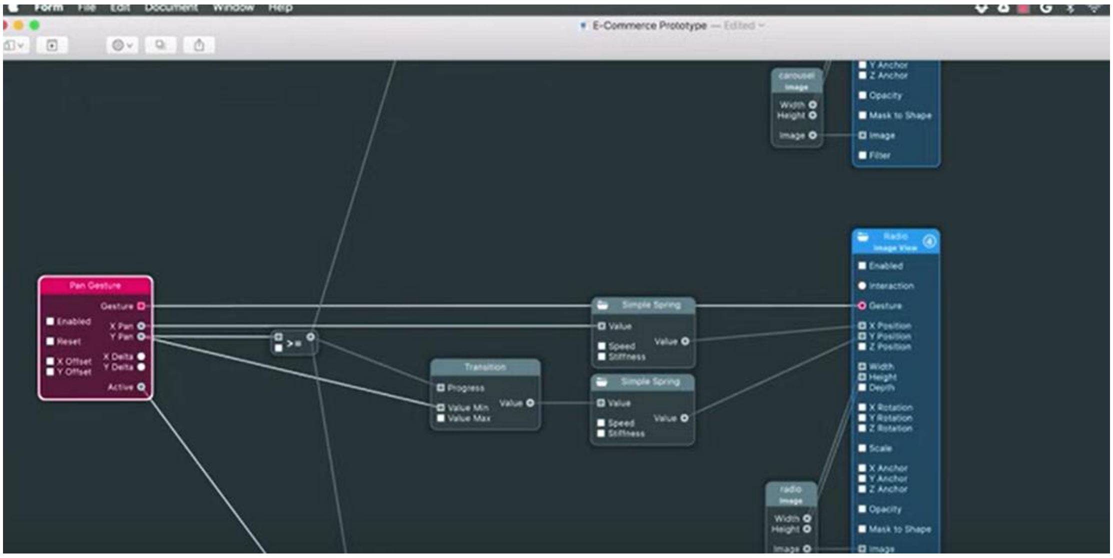
Finally, once my ideas are clearly illustrated, it’s easier to discuss them with my team and assess their feasibility, allowing for smarter design decisions. After refining the sketches, I can then transition to high-fidelity models more efficiently. This process ensures better results and saves time in the long run.

High-fidelity page design method:

High-fidelity page design involves creating interactive models that allow users to experience a design firsthand, making it easier to explain ideas. One of the main advantages of this approach is that you don’ t need to know programming to create effective, interactive visuals, making it an affordable and accessible method.



atomic, marvel, pixate, in vision are great options for this process.all good choices, but the designer in the video chooses a tool called **principle** and demonstrates how to use it, including how to import footage, set up a monitor bar for a better view of the real visuals, create interactive effects such as scrolling features, page appearing and disappearing effects.A key benefit of Principle is its real-time updates on mobile devices, allowing for more convenience.



The second tool form, which is a great tool for conveying motion gestures and and overall effects to experience interactions in a static mode. This software is a node-based prototyping tool for ios and Windows that uses patches to create interaction effects, which are also equally good for displaying interactions.

Reflecting on this process, I realize that these tools offer great flexibility and save time in the design phase. In my own work, I’ve sometimes spent too much time on static designs before realizing the importance of user interaction. Using these tools allows for quick iterations and real-time testing, which is a valuable lesson for improving both efficiency and design quality.

Native Prototyping:

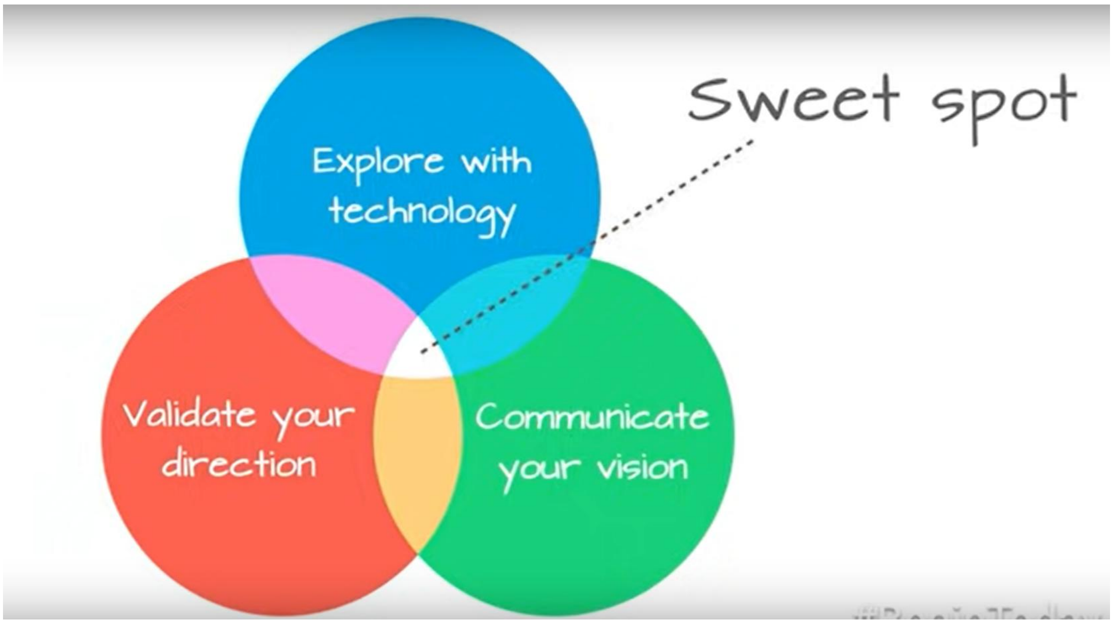
When we talk about prototyping, we usually refer to using programming to create a page, like writing code in languages such as Java, Python, or JavaScript. In our classes, we’ve learned how to create applications using HTML, CSS, and JavaScript for browsers. Unlike the previous videos, this method involves real devices, real data, and real users, which makes it more authentic and practical.

Technologies like gyroscopes, accelerometers, and speakers all contribute to the user’s experience. When you’re coding, it’s important to approach the process with a “hacker’s mindset”—taking risks and experimenting to see what works.

User experience is key. When creating your design, it’s essential to focus on how users will interact with it, getting feedback on aspects like color, smoothness, or even the app's potential value. This feedback is invaluable for iterating and improving the design, making it more appealing and intuitive.

Once you' ve refined your prototype, the next step is selling your idea. Letting people experience your vision firsthand is one of the best ways to communicate its potential. If you' ve done the groundwork well, you can back up your project with solid data that shows it has promise.

The three most important elements of programming design are exploring the technology, validating your direction, and effectively communicating your vision. Reflecting on this process, I realize how critical it is to stay open to experimentation and make sure your design resonates with real users.



Useful Resources for Prototyping:

A great resource for learning about prototyping is the book "Lean UX: Applying Lean Principles to Improve User Experience" by Jeff Gothelf. It focuses on the iterative, user-centered design process and how to rapidly prototype and test ideas. This book is valuable because it encourages a collaborative, feedback-driven approach to design and helps you understand how to build prototypes that are not only functional but also effective in solving real user problems. This book is good for beginners because it is basically zero-based and produces effective prototypes without extensive coding or resources.

source:https://books.google.co.uk/books/about/Lean_UX.html?id=K-RmSR_jo5MC&redir_esc=y