



Graphical User Interfaces: Crash Course Computer Science #26

Crash Course: Computer Science

<https://youtube.com/watch?v=XIGSJshYb90>

<https://nerdfighteria.info/v/XIGSJshYb90>

Hi I'm Carrie Anne and welcome to Crash Course Computer Science.

We ended last episode with a 1984 release of Apple's Macintosh personal computer. It was the first computer a regular person could buy with a graphical user interface and a mouse to interact with it.

This was a radical evolution from the command line interfaces found on all other personal computers of the era. Instead of having to remember or guess the right commands to type in, a graphical user interface shows you what functions are possible. You just have to look around the screen for what you want to do. It's a point and click interface.

All of the sudden, computers were much more intuitive. Anybody, not just hobbyists or computer scientists could figure things out all by themselves.

[Opening theme plays]

The Macintosh is credited with taking graphical user interfaces, or GUI's, mainstream but in reality they were the results of many decades of research. In previous episodes, we've discussed some early interactive graphical applications like sketchpad and spacewalk, both made in 1962. But these were one-off programs and not whole integrated computing experiences.

Arguably the true forefather of modern GUIs was Douglas Engelbert. Let's go to the thought bubble. During WWII while Engelbert was stationed in the Philippines as a radar operator, he read Vannevar Bush's article on the Memex.

These ideas inspired him and when his navy service ended, he returned to school completing a PhD in 1955 at UC Berkeley. Heavily involved in the emerging computing scene, he collected his thoughts in the seminal 1962 report titled Augmenting Human Intellect. Engelbert believed that the complexity of the problems facing mankind was growing faster than our ability to solve them.

Therefore, finding ways to augment our intellect would seem to be both a necessary and a desirable goal. He saw that computers could be used for beyond just automation and be essential interactive tools for future knowledge workers to tackle complex problems.

Further inspired by Ivan Sutherland's recently demonstrated sketchpad, Engelbert set out to make his vision a reality, recruiting a team to build the oN-Line System. He recognized that a keyboard alone was insufficient for the type of applications he was hoping to enable.

In his words - *"We envision problem-solvers using computer-aided working stations to augment their efforts. They required the ability to interact with information displays using some sort of device to move (a cursor) around the screen."*

And in 1964, working with colleague Bill English, he created the first computer mouse. The wire came from the bottom of the device and looked very much like a rodent and the nickname stuck.

Thanks Thought Bubble!

In 1968 Engelbert demonstrated his whole system at the Fall Joint Computer Conference, in what's often referred to as, "the mother of all demos". The demo was ninety minutes long and demonstrated many features of modern computing - bitmap graphics, video conferencing, word processing, and collaborative, real-time editing documents. There were also precursors to modern GUIs, like the mouse and multiple windows, although they couldn't overlap. It was

way ahead of its time and like many products with that label it ultimately failed, at least commercially, but its influence on computer researchers of the day was huge. Engelbert was recognized for this watershed moment in computing with a Turing Award in 1997.

Federal funding started to reduce in the early 1970s, which we discussed two episodes ago, at that point many of Engelbert's team, including Bill English, left and went to Xerox's newly formed, Palo Alto research center, more commonly known as Xerox Parc. It was here that the first true GUI computer system was developed, the Xerox Alto, finished in 1973.

So for the computer to be easy to use it needed to be more than just fancy graphics. It needed to be built around a concept that people were already familiar with so they'd immediately recognize how to use the interface with little or no training. Xerox's answer was to treat the 2D screen like the top of a desk, or a desktop. Just like how you can have many pages laid out on a desk, a user could have several computer programs on at once, each was contained in their own frame which offered a view onto the application called a window.

Also, like papers on a desk, these windows could overlap blocking the items behind them and there were desk accessories like a calculator and clock that the user could place on the screen and move around. It wasn't an exact copy of a desktop though, instead it was a metaphor of a desktop. For this reason, surprisingly, it's called the desktop metaphor.

There are many ways to design an interface like this but the Alto team did it with windows, icons, menus and a pointer, what's called a WIMP interface. It's what most desktop GUIs use today. It also offered a basic set of widgets, reusable graphical building blocks, things like buttons, check boxes, sliders, and tabs which were also drawn from real world objects to make them familiar.

GUI applications are constructed from these widgets. So let's try coding a simple example using this new programming paradigm.

First, we have to tell the operating system that we need a new window to be created for our app, we do this through a GUI API. We need to specify the name of the window and also its size, let's say 500 by 500 pixels. Now let's add some widgets: a text box and a button. These require a few parameters to create. First we need to specify what window they should appear in because apps can't have multiple windows, we also need to specify the default text, the x and y location in the window, and a width and height.

Okay, so now we've got something that looks like a GUI app but has no functionality. If you click the roll button nothing happens.

In previous examples we've discussed the code pretty much executes from top to bottom, GUIs on the other hand use what's called event-driven programming. Codes can fire at any time in different orders in response to events. In this case it's user driven events, like clicking on a button, selecting a menu item or scrolling a window, or if a cat runs across your keyboard it's a bunch of events all at once.

Let's say that when the user clicks the roll button we want to randomly generate a number between 1 and 20 and then show that value in our text box, we can write a function that does just that. We can even get a little fancy and say that if we get the number 20, set the background colour of the window to blood red.

The last thing we need to do is hook this code up so that it's triggered each time our button is clicked.



Graphical User Interfaces: Crash Course Computer Science #26

Crash Course: Computer Science

<https://youtube.com/watch?v=XIGSJshYb90>

<https://nerdfighteria.info/v/XIGSJshYb90>

To do this we need to specify that our function handles this event for our button by adding a line to our initialize function. The type of event in this case is a click event and our function is the event handler for that event.

Now we're done. You can click that button all day long and each time our roll D20 function gets dispatched and executed. This is exactly what's happening behind the scenes when you press the little bold button in the text editor or select shutdown from the drop down menu. A function linked to that event is firing. Hope I don't roll a twenty, aah!

Okay, back to the Xerox Alto. Roughly two thousand Altos were made and used at Xerox and given to university labs. They were never sold commercially, instead the Parc team kept refining the hardware and software culminating in the Xerox Star system released in 1981.

The Xerox Star extended the desktop metaphor. Now files looked like pieces of paper and they could be stored in little folders all of which could sit on your desktop or be put away into digital filing cabinets. It's a metaphor that sits on top of the underlying file system. From the user's perspective this is a new level of abstraction.

[A New Level of Abstraction jingle plays]

Xerox, being in the printing machine business also advanced text and graphics creations. For example, they introduced the terms cut, copy, and paste. This metaphor was drawn from how people dealt with making edited documents written on typewriters. You would literally cut text out with scissors and then paste it with glue into the spot you wanted in another document, then you photocopied the page to flatten it back down into a single layer, making the change invisible. Thank goodness for computers!

This manual process was moot with the advent of word processing software which existed on platforms like the Apple 2 and Commodore PET. But Xerox went way beyond the competition with the idea that whatever you made on the computer should look exactly like the real world version if you printed it out. They dubbed this "what-you-see-is-what-you-get" or WYSIWYG. Unfortunately, like Englebert's oN-line system, the Xerox style was ahead of its time. Sales were sluggish because it had a price tag equivalent to nearly \$200,000 today for an office set up. It also didn't help that the IBM PC launched that same year followed by a tsunami of cheap IBM compatible PC clones.

But the great ideas that Parc researchers had been cultivating and building for almost a decade didn't go to waste. In December of 1979, a year and a half before the Xerox Star shipped, a guy you may have heard of visited, Steve Jobs.

There's a lot of lore surrounding this visit, with many suggesting that Steve Jobs and Apple stole Xerox's ideas, but that simply isn't true. In fact, Xerox approached Apple, hoping to partner with them. Ultimately Xerox was able to buy a million dollar stake in Apple before its highly anticipated IPO, but it came with an extra provision: disclose everything cool going on at Xerox Parc.

Steve knew they had some of the greatest minds in computing but he wasn't prepared for what he saw. There was a demonstration of Xerox's Graphic User Interface running on a crisp Bitmap display, all driven with intuitive mouse input. Steve later said, *"it was like a veil being lifted from my eyes. I could see the future of what computing was destined to be."*

Steve returned to Apple with his engineering entourage and they got to work inventing new features, like the menu bar and a

trashcan to store all the files to be deleted, it would even bulge when full. Again with the metaphors.

Apple's first product with Graphical User Interface and a mouse was the Apple Lisa, released in 1983. It was a super advanced machine with a super advanced price, almost \$25,000 today. That was significantly cheaper than the Xerox Star but it turned out to be an equal flop in the market.

Luckily, Apple had another project up its sleeve, the Macintosh, released a year later in 1984. It had a price of around \$6,000 today, a quarter of the Lisa's cost, and it hit the mark, selling 70,000 units in the first 100 days. But after the initial craze sales started to falter and Apple was selling more of its Apple 2 computers than Macs. A big problem was that no one was making the software for this new machine with its new radical interface and it got worse because the competition caught up fast.

Soon other personal computers had primitive but usable GUIs on a computers a fraction of the cost. Consumers ate it up and so did PC software developers. With Apple's finances looking increasingly dire, and tensions growing with Apple's new CEO, John Scully, Steve Jobs was ousted. A few months later, Microsoft released Windows 1.0.

It may not have been as pretty as Mac OS but it was the first salvo in what would become a bitter rivalry and near dominance of the industry by Microsoft. Within ten years, Microsoft Windows was running on almost 95% of personal computers.

Initially fans of Mac OS could rightly claim superior graphics and ease of use. Those early versions of Windows were all built on top of DOS, which was never designed to run GUIs but, after Windows 3.1, Microsoft began to develop a new consumer orientated OS with an upgraded GUI called Windows95. This was a significant rewrite that offered much more than just polished graphics. It also had advanced features Mac OS didn't have like program multitasking and protective memory. Windows95 introduced many GUI elements still seen in Windows versions today like the start menu, task bar, and Windows Explorer file manager.

Microsoft wasn't infallible though. Looking to make the desktop metaphor even easier and friendlier, it worked on a product called Microsoft BOB and it took the idea of using metaphors to the extreme. Now you had a whole virtual room on your screen with applications embodied as objects that you could put onto tables and shelves. It even came with a crackling fireplace and a virtual dog to offer assistance. And you see those doors on the sides? Yep, those went to different rooms in your computer where different applications were available. As you might have guessed, it was not a success.

This is a great example of how the user interfaces we enjoy today are the product of what's essentially natural selection. Whether you're running Windows, Mac, Linux, or some other desktop GUI, it's almost certainly an evolved version of the WIMP paradigm first introduced on the Xerox Alto. Along the way a lot of bad ideas were tried and failed everything had to be invented, tested, refined, adopted, or dropped.

Today GUIs are everywhere and while they're good, they're not always great. No doubt, you've experienced design-related frustrations after downloading an application, used someone else's phone, or visited a website and for this reason computer scientists and interface designers continue to work hard to craft computing experiences that are easier and more powerful. Ultimately working towards Englebert's vision of augmenting human intellect.

I'll see you next week.