QR Codes (Speaker Notes)

Slide 1

No speaker notes for this slide.
Hello everyone. Type 'yes' or 'no' in the chat, or put your hand up, if you recognise this image?

Participants could physically put their hands up or use the built in features of the video conferencing system you are using. Remember to ask people to put their hands down when you are ready to move onto the next question.

Our focus for today's session is Error Control in QR Codes. Firstly, what does QR stand for?

Wait for responses and respond appropriately.

That's right, QR stands for Quick Response. This is a Quick Response code - they get used in many situations as a quick and simple way to communicate information.

Now use your device and try to scan this QR code (most camera apps will do this if it's enabled on your device settings). Write in the chat what it brings up on your device.

Wait for a couple of responses.

Mine scans and brings up the words "Fun learning". If you see the same message, type yes in the chat, or give me a thumbs up. If it's not scanning for you check there's no sun reflecting on your screen or anything else interfering with the image.

Wait until the majority of participants have indicated that they have been able to scan the QR code from your screen. If anyone is having any problems scanning it, check they are happy to observe what's happening and suggest downloading a free QR code app at the same time or change their camera settings to recognise QR codes.

Each of these black and white squares represents one binary digit, one bit. If you took the time to count, this is a 21x21 grid, so there are 441 individual squares making up...
this QR code. This is more than enough to represent the characters in the words "fun learning". We'll refer to the squares as "bits" to emphasise that each one is a binary digit.

An important part of computational thinking is testing systems and finding the limits. This activity does just that.

**Slide 3**

QR codes have become ubiquitous, we see them everywhere. But how do they work? And how much can they be damaged before they stop working? Have you ever tried to scan a QR code and it didn't load? Or alternatively, did one work that you didn't expect to?
We've got a fun game to play with your class to test this out. It's called “Will it scan?”

**Slide 4**

Wait for video to finish.

As usual, this isn't a video to show your class, it's one for them to experience and create themselves. Have them test for sunlight, reflections, or shadows, which can stop a QR code from working. Creases, rips, stains, and other damage can also stop QR codes from working. Encourage your students to be as creative as possible.

**Slide 5**

We are going to replicate glare or other interference that can happen to a QR Code. If you have experienced the Error Control - Parity Card Magic Trick then this is the same concept as flipping a white card to black, and vice versa. Let's try out some error control.
I'm going to use this interactive to replicate a damaged QR code. I can click anywhere on this QR code to change a bit from white to black or from black to white. Before I change anything, have a go at scanning it and share in the chat what it says.

Make sure the cursor is moved out off the QR code each time it is scanned, otherwise it can act as a white bit. Give participants a chance to scan the QR code showing on your shared screen. This should only take a few seconds.

Ok. I can see in the chat that most of you have got it to scan, and that it reads "Hello world".

Now that you've all had it scan I'm going to change one of the bits to the opposite ie white to black or black to white. I'm going to change this one. Have another go at scanning it. Does it still work?

Click on one of the bits to reverse its colour.

If you were doing this in person, instead of using the app you might print out some QR codes and have students change them with a black pen, which is more realistic since QR codes are normally seen on paper.

Would anyone like to guess in the chat how many changes I'm going to be able to make before this QR code stops scanning?

Wait for responses.

I can see people guessing [read some numbers]. Let's start and see who guessed correctly. Let's go!

I'm going to keep changing the QR code one bit at a time until you tell me it isn't working for you. Put a 'Y' for 'Yes' in the chat each time it scans for you. When it stops scanning, type in a 'N' for 'No'.
At this stage, only change bits in the main part of the QR code, and do not change any of the bits that make up the orientation squares in the 3 corners, including the white bits immediately next to them.

Wait until you know the QR code is scanning for people, and you have had a number of responses in the chat, and respond appropriately.

Keep changing one bit at a time, and allow participants to scan it with each change, until you see in the chat it no longer scans.

Ok. I can see it has stopped scanning for several of you. I can see at the bottom of the screen here that it took [say the number of times it took] changes to cause it to stop working. Let's keep going until it has broken for everyone. When it stops working for you write the number of changes the interactive has recorded at that time in the chat.

Keep going until you have a lot of 'N's or numbers (of changes) in the chat. Have a quick check to see if it is still working for anyone, then move on to the next steps.

So how many changed “bits” (or squares) did it take until the QR code wouldn't scan any more? I saw a lot of '6's in the chat and a couple of different numbers too.

For the first few changes, your scanners were able to perform error correction, which means they could work out the correct original message even though there were errors in the bits. By changing one bit at a time, we have demonstrated the exact point when error correction became error detection only. The QR code reader detected that there was an error in the information it was trying to scan, the QR code, and, rather than giving the wrong information, it gave no information at all. At that point it couldn't work out the correct original information.

QR codes have a lot more extra bits added to them than the parity system you may have seen before (which can be presented as a magic trick). The parity method can correct a one-bit error, but if there is more than one bit changed, it can only detect that there is an error, not where it is. In either case, this ensures the system is very
reliable at giving either the intended information, or no information at all (not scan). A damaged QR code is unlikely to accidentally give you the wrong information, as the likelihood of the damaged bits changing the original QR code to match another QR code exactly is extremely low. Of course, someone could purposefully change the whole QR code if they wanted.

The error control system used by QR codes is called Reed-Solomon error correction. Reed-Solomon error correction is also used for other situations where data needs to be stored reliably, such as disk storage. Often QR codes that look quite damaged will still work because of the error correction that is built into them.

Did you notice that some cell phone or tablet cameras read QR codes automatically? We have noticed that some of these cameras will read a QR code a certain number of times but will then stop reading it. This may not be because the QR code is corrupt, but because the camera has stopped looking specifically for a QR code.
Let’s have a closer look at how QR codes actually work. What do you think the significance of the squares in the three corners is? Write your ideas in the chat.

Wait for an appropriate number of responses and reply accordingly.

I saw some good answers there. You may have noticed that the changes we were making to the QR code avoided the three large squares and the surrounding white space in the corners. This is because they are the 'finder pattern', which allows the scanner to detect and orient the QR Code, no matter which way up it is. The software uses the corners to figure out the alignment. Even if it is being scanned at a funny angle most software can still scan it and use some mathematics to “flatten out” the image. The white spaces around the edge of these squares are relied on by most scanning software to identify where the finder patterns are.

When you run this activity yourselves, it's okay if people do change the QR code corners and it breaks immediately. In that situation have them try again and tell them to avoid the three squares. It's good to have several spare copies of the QR code if you are using printed versions!

If you’re using coloured QR codes and coloured pencils, and it still scans even after you've coloured in pretty much the whole QR code, that's a great exploration. It's probably still working because the pencil is not dark enough.
Error correction and detection is used widely - every digital download, disk drive, DVD, QR code, and barcode uses it.

And as computing continues to push boundaries, error control will become more and more important. For example, the future of technology includes quantum computing. Quantum computing uses quantum bits, or qubits, and harnesses their ability to exist in more than one state at the same time to create computers that are potentially exponentially faster, and more energy efficient. Quantum data storage is inherently unstable, and relies on error correction codes to make it usable.
Now that we've explored QR codes a bit, let's look closer at what error control, detection and correction mean.

Can computers find errors in data, and correct them, without any human intervention? What do you think? Share your ideas in the chat.

Wait for a few answers and respond as appropriate.

As we've seen with the QR codes, many digital files can be reconstructed even after they have been damaged in some way. This is due to the error control built into them. Programs add extra information when files are created so that if any of the original data is unintentionally changed the computer can detect this and convert it back to the original file, or at least not let damaged data be accessed. In the case of the QR code the error control is extra black and white squares inserted throughout the information itself.

Push the next slide button/key to show the second question.

What does Error Detection mean? Share your answers in the chat.

Wait for responses and respond appropriately.

Error detection is when software can identify if data has been corrupted (usually by accident) after it was transmitted or stored. In a QR code, usually if errors are detected but can't be corrected, nothing further happens and the data is just ignored altogether, but for data being downloaded from the internet or read from a disk, an error message is usually sent to the user telling them there was an error (such as "download failed" or "file corrupted"), or the computer automatically requests the file to be sent again. In the case of the QR code, error detection has happened when the QR code refuses to scan.
What does Error Correction mean? Share your answers in the chat.

Wait for responses and respond appropriately.

Error correction is when data has extra information coded into it so it can continue to be used accurately, even if some of the data is corrupted after it is transmitted or stored. In the case of QR Codes, this means being able to read the QR code even when some bits are changed, and we saw this happen when only a few bits were changed.

What does Error Control mean? Share your answers in the chat.

Wait for responses and respond appropriately.

Error control is when a computer (or app) is able to prevent an error in data from having an unwanted outcome, whether that is by detecting and correcting the error, or detecting the error and indicating that it's there.

Supporting Resources

Computer Science Field Guide - QR Code Generator