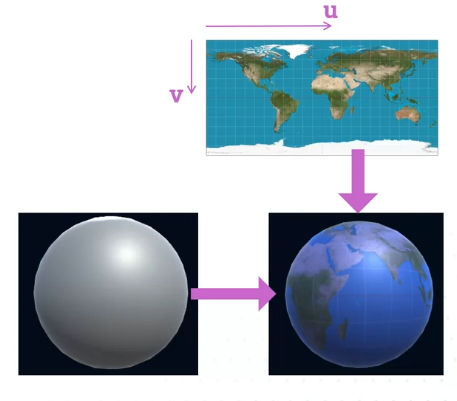
## Texture mapping



We actually call the number of pixels across U and the number of pixels down V. We're looking at UV coordinates, as we call them, also called texture coordinates, the positions of pixels in an image. You can represent them either as actual pixel values. You might have a 1080 pixels across or something.

Graphical user interface, text, application

Description automatically generated

XY coordinate is world coordinate, UV coordinates are fragment coordinate on object

Diagram

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

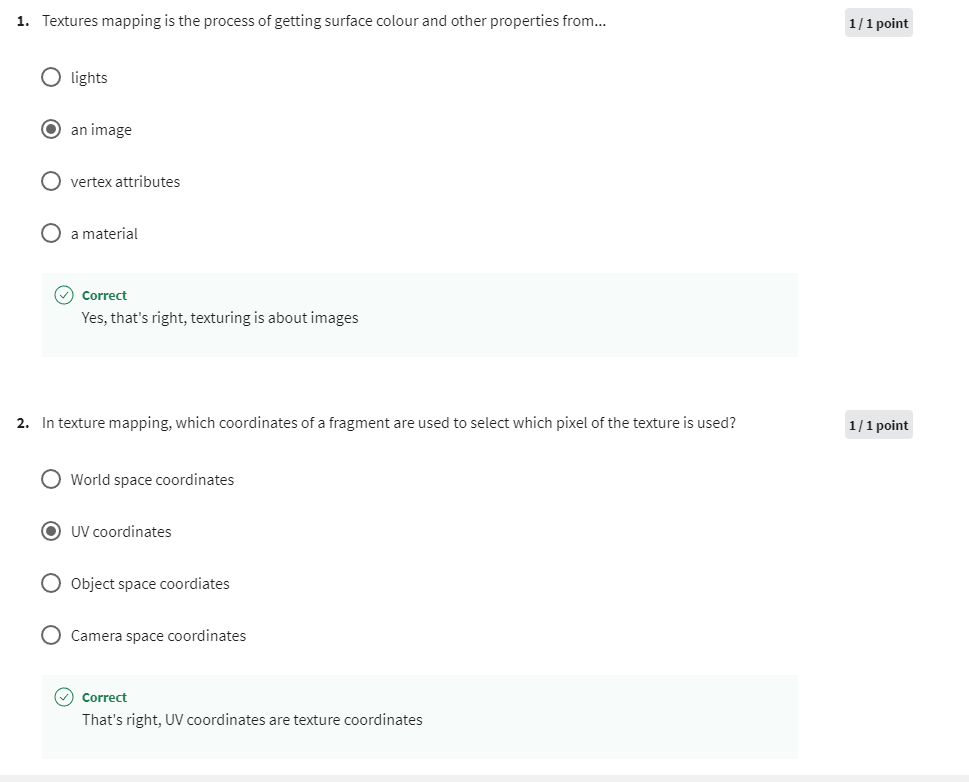
Description automatically generated with low confidence

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email, Teams

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email, Teams

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Advanced texturing

Graphical user interface, text, application, chat or text message

Description automatically generated

this was a pretty simple algorithm that involved taking the mod function to create stripes. Problem is that we're using the object space position of the fragment to generate the pixel values.

Graphical user interface, text, application

Description automatically generated with medium confidence

A close-up of a logo

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

Qr code

Description automatically generated

Noise isn't just randomness, the top image is actual randomness that looks like the static on an old TV sets. You don't get that much anymore now that we're all digital, but this is true randomness. Each pixel is completely random and different from the one next door. The bottom image is something called Perlin noise. Now, Perlin noise is what's called coherent noise that is generated so that it doesn't vary completely randomly from one pixel to another. But there is more gradual fades from one pixel. Each pixel will be relatively close to the previous one, but that level of variation is itself random.

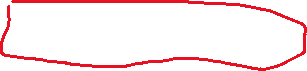
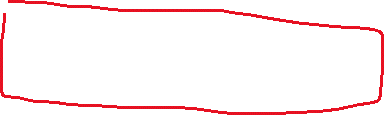
## Teleport effect

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



Need to do a few things to allow transparency. The transparency fundamentally changes the way everything works in the rendering queue. Normal which pixel is drawn is determined by the Z buffer. If a pixel is in front of another one, the fragment behind is not drawn. That doesn't work with transparency. You need to draw both the transparent pixels and whatever is behind it.

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Animate it

Text

Description automatically generated

Animate it with border

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated