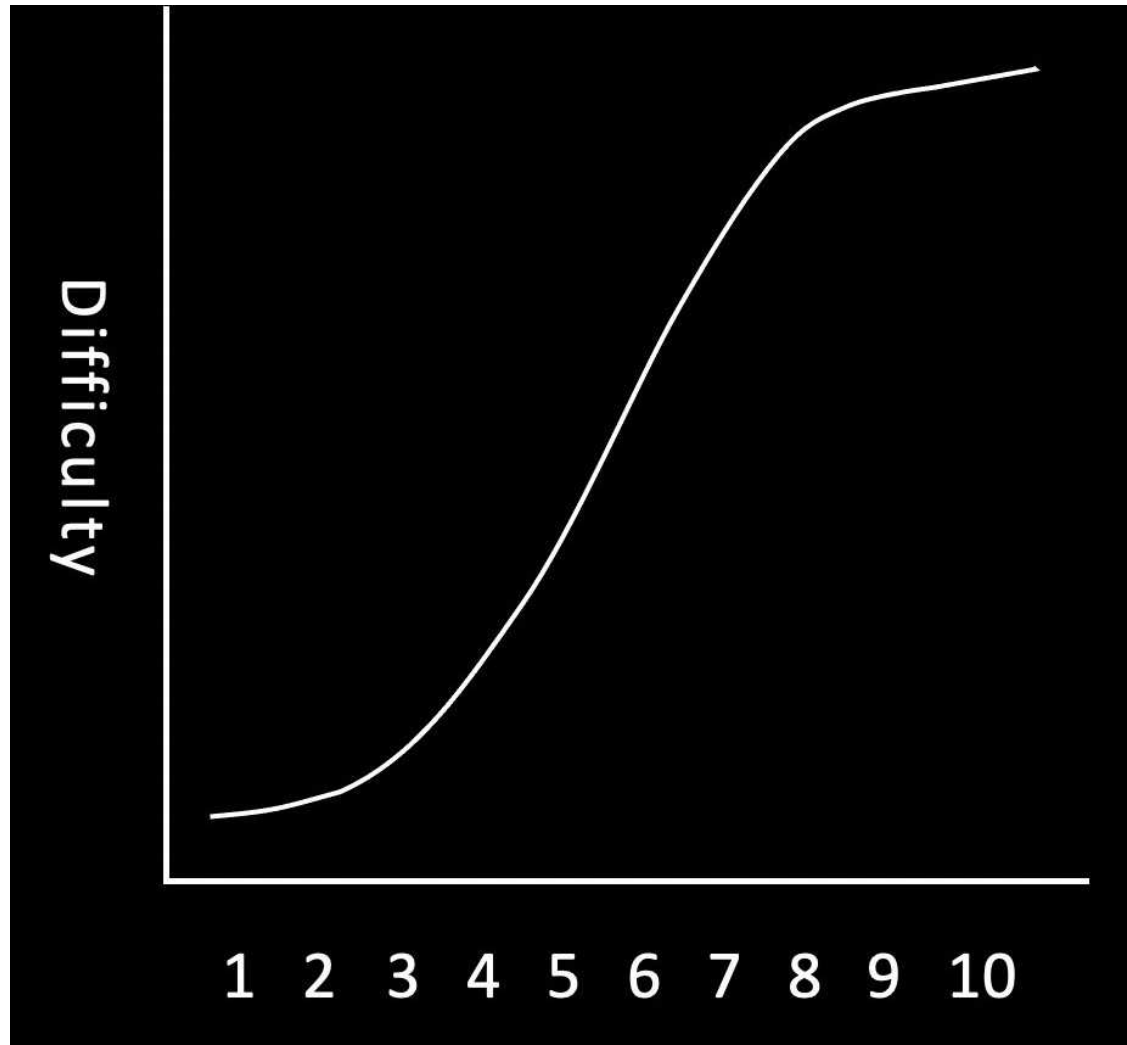


COMS30020 - Computer Graphics
Week 3 Briefing

Dr Simon Lock

Where are we ?



Metaphor

There are three ways to start a motor race:

Standing Start

Rolling Start

"Le Mans" Start



Which One ?

The "Rolling Start" is arguably the safest...

Everyone is already moving

Reduced risk of running into the back of people

We can be sure all cars are working correctly

Everyone is familiar with the circuit

This is the approach we are using on this unit

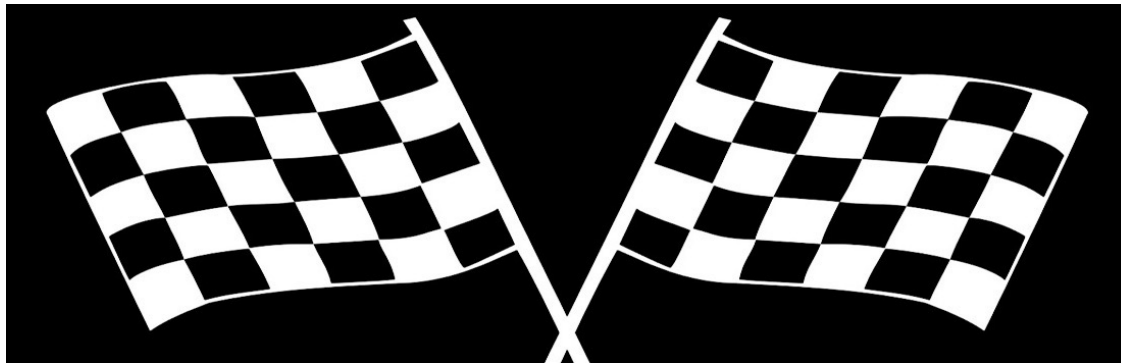
This coming week on "Computer Graphics"

We are actually going to draw something useful !

Our focus will be on a key drawing primitive:

△ △ △ "The Triangle" △ △ △

The primary building block for the rest of this unit !!!





Every triangle is a
love triangle when
you love triangles.

-Pythagoras

Pythagoras did NOT say that !

Powerful Triangles

Although simple, Triangles are VERY powerful
A convenient structure to cover ANY surface





Using MANY triangles, we can create complex shapes

The Stanford Bunny (70k triangles)



What kinds of Triangle ?

In coming week we'll draw various types of triangle:

- Unfilled (also known as "stroked") triangles
- Filled triangles (with a choice of colours !)
- Composite triangles (filled AND stroked !!!)

Drawing triangles might seem trivial at first glance

But the devil is in the detail...

There is hidden complexity that we must deal with !

Challenges

Surely a "stroked" triangle is easy ?

It's just three straight lines !



Sure, currently we don't have a `drawLine`` function

But we can just use the `setPixelColour`` function

Call it lots of times to draw a sequence of pixels

We'll just need to calculate X and Y

for each point along the line

(with a nice bit of interpolation)

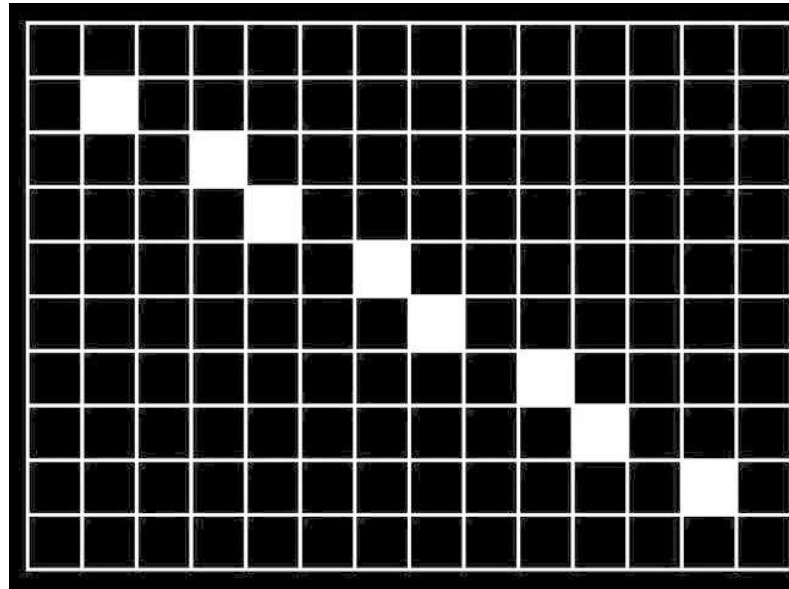
Not that easy !

We need to be a little bit careful

If we just loop through each ROW of the screen

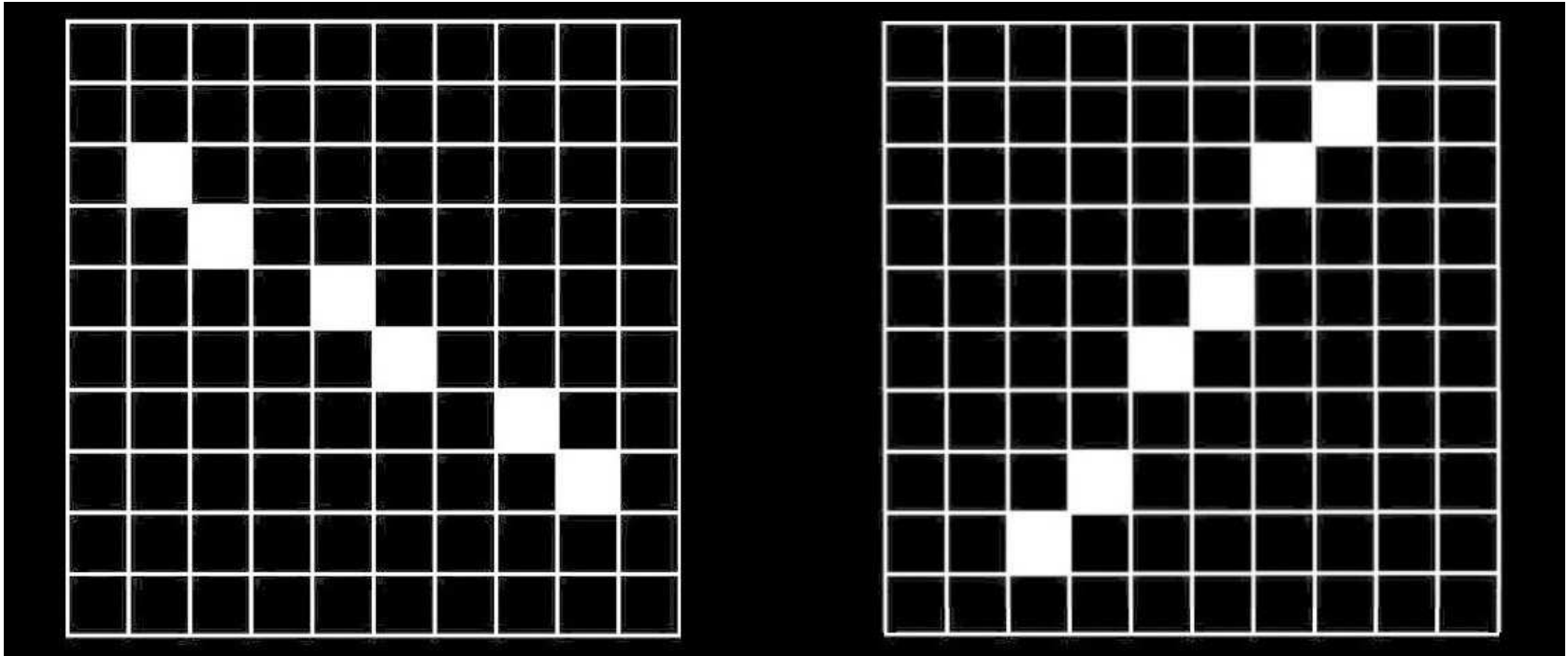
```
for(int x=1; x<=8 ;x++) y = interpolate(x, ...)
```

We could end up drawing something like this:



Needs Careful Handling

Neither can we ALWAYS just loop through columns
We must consider gradient of line when drawing pixels
Workbook provides details on how to deal with this



Filled Triangles - Any Easier ?

Filled triangles must surely be easier then ?

All we need to do is draw some horizontal lines

Starting and ending at the correct x positions

We just draw from the "left hand side"...

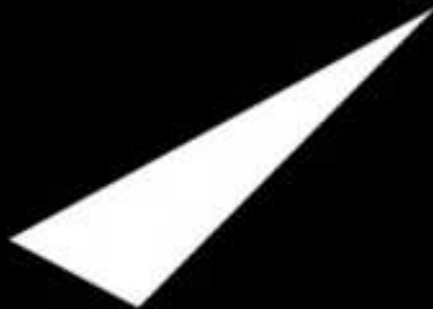
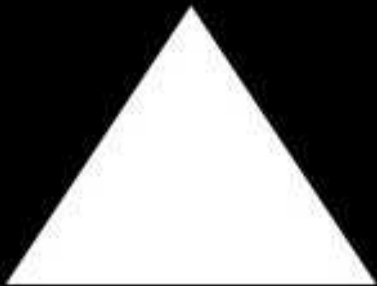
All the way to the "right hand side"

One row at a time

Top to bottom

Different types of triangle

However, non flat top/bottom triangles are harder
There isn't a "left hand side" and "right hand side"
This is because each triangle has THREE sides...



Workbook explains a technique for dealing with this !
(Hint: It's a lot easier if we split the triangle into 2)

And if all that wasn't exciting enough...

Sergio Odeith



Sergio Odeith



Tom Bragado Blanco



Tom Bragado Blanco

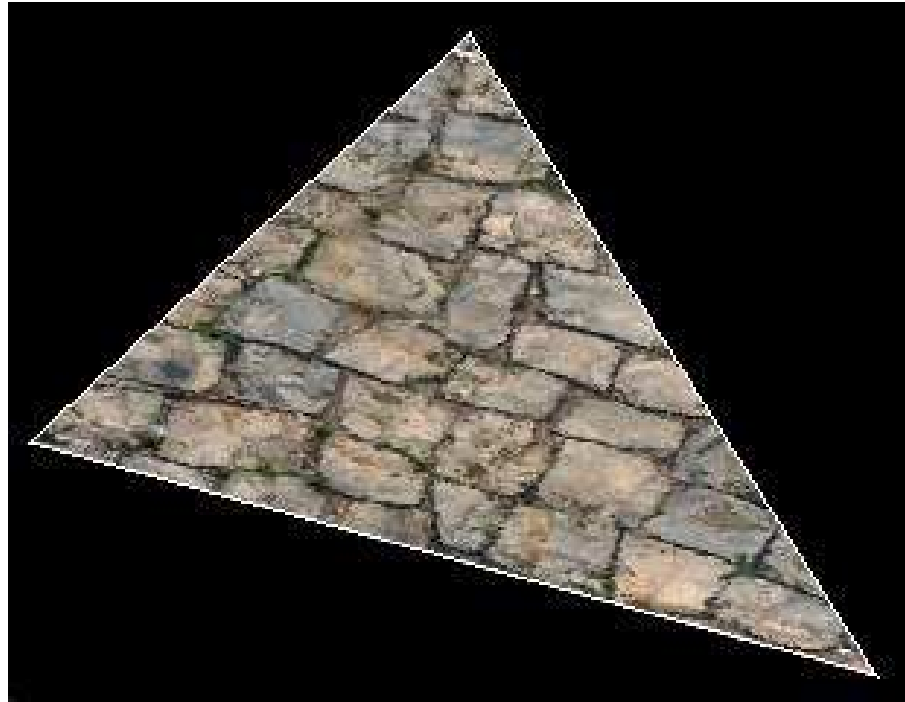


Your Objective

A texture is provided for you in the workbook

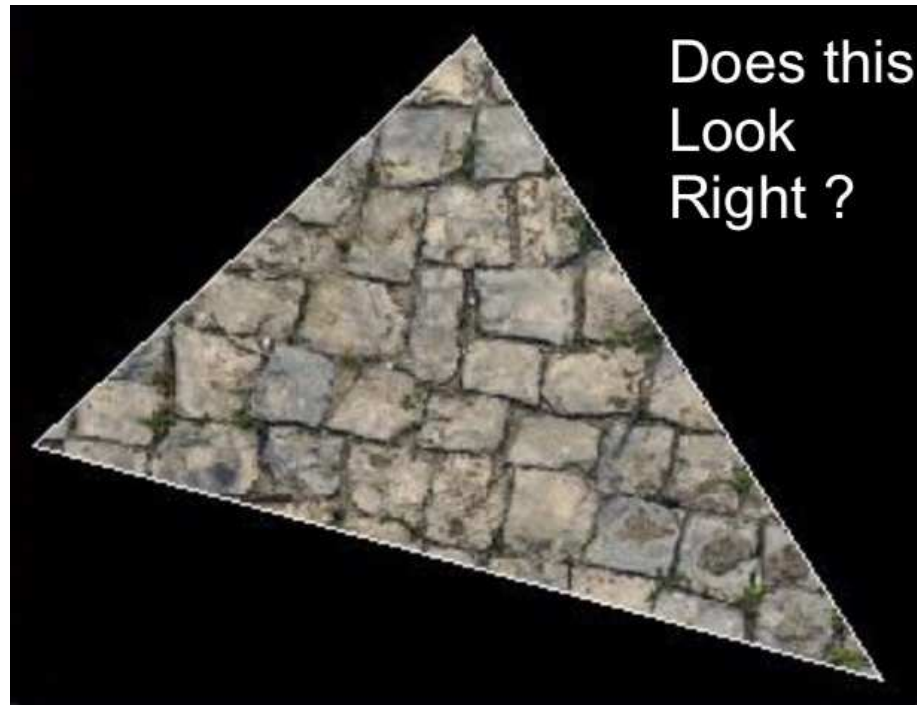
A reference image illustrates your final objective

This allows you to "visually verify" your success



Implicit Feedback

Reference image provides passive/implicit feedback
You can gauge how well you are currently doing
and what aspects of your work need improving



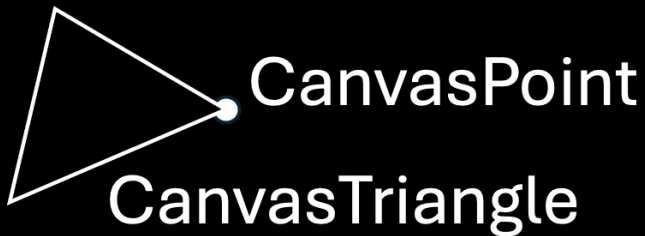
Terminology and Provided Classes

All class source files can be found in `libs/sdw`

-Screen-

DrawingWindow

-Canvas-



-File System-

-PPM Image File-

TextureMap

TexturePoint



Final Thoughts

Many of you still just use println/cout for debugging
You might like to consider using a **proper** debugger

gdb on the command line, integrated tools in IDEs

No pressure - whatever you are comfortable with...
...it's just that debugging 3D renders is a bit tricky
You are going to need all the help you can get !

debug