COMS30020 - Computer Graphics

Introductory Briefing

Dr Simon Lock

Welcome to Computer Graphics

I'm Dr Simon Lock

BSc in Computer Science

PhD in Software Engineering

Background in Digital Arts

simon.lock@bristol.ac.uk

Aim of unit: Introduce "Computer Graphics" !

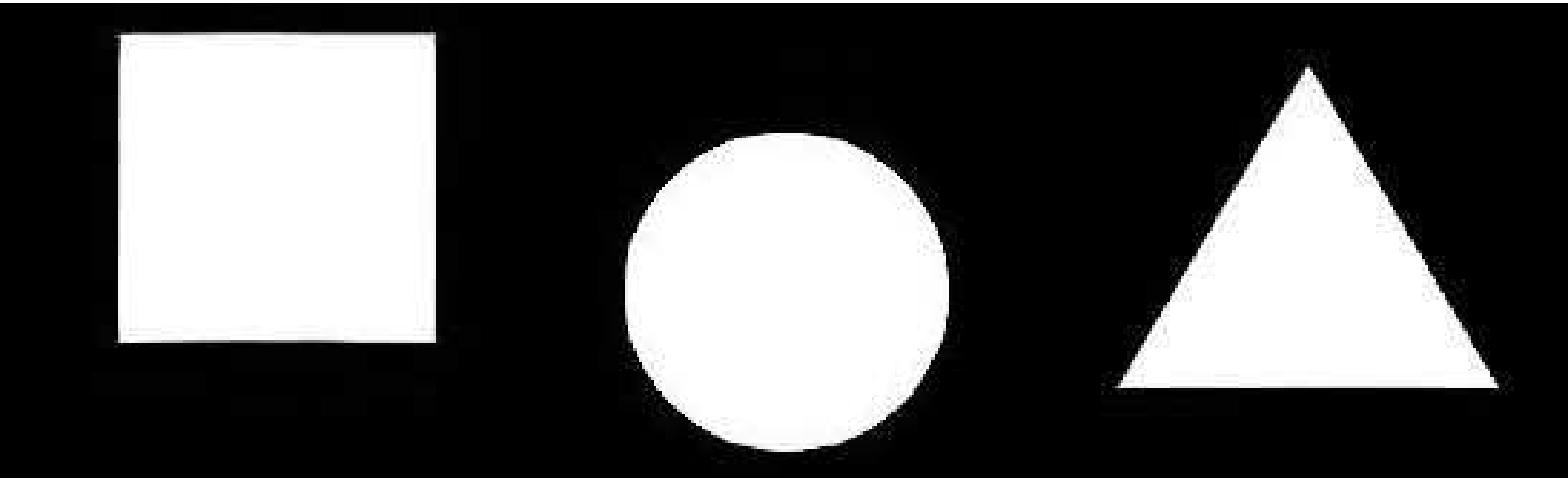
This term means different things to different people
Our perspective on the area will encompass:

- Low-level "individual pixel" manipulation
- Drawing primitives (lines and triangles)
- 2D and 3D geometry (numerical position data)
- Camera views, movement and navigation
- *Approximating* behaviour of light
- Realistic rendering of various textures/materials

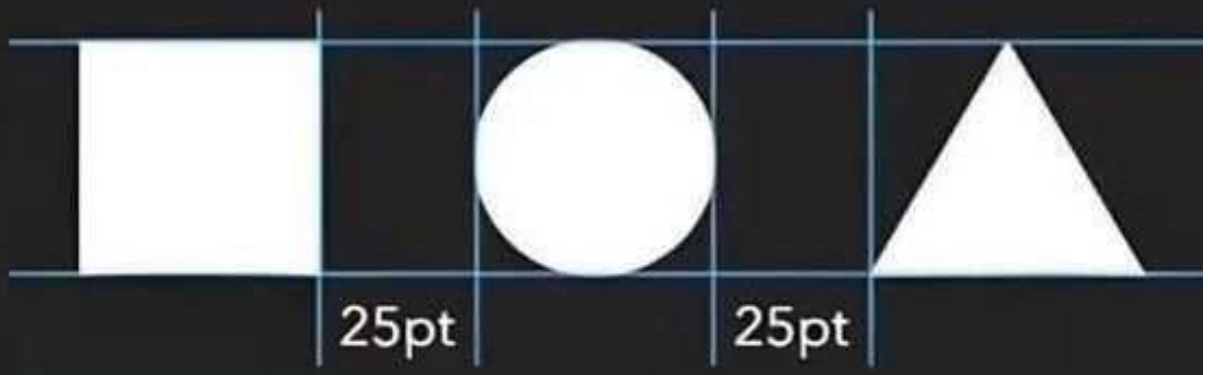
As you might have sensed, it's all pretty low-level...
This is definitely NOT a "Graphic Design" unit !

The Discipline of Graphic Design

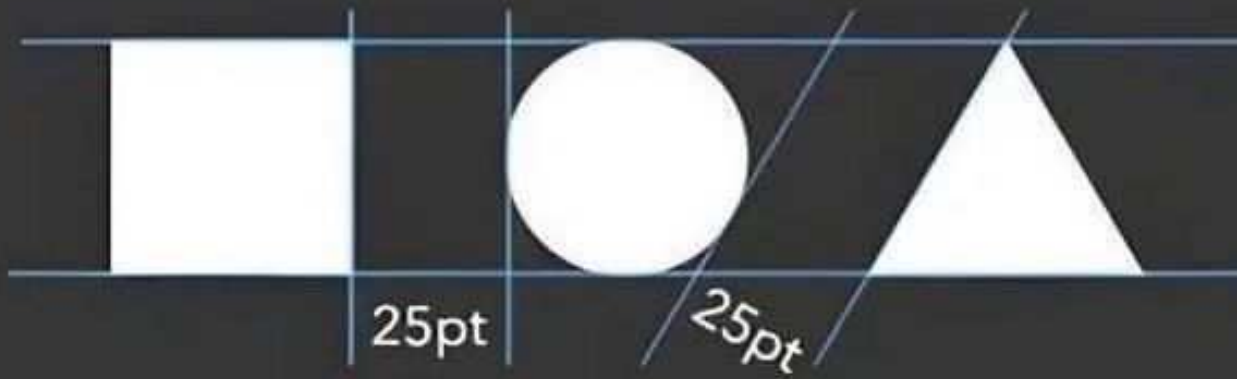
Simple task: layout these 3 shapes evenly in a line



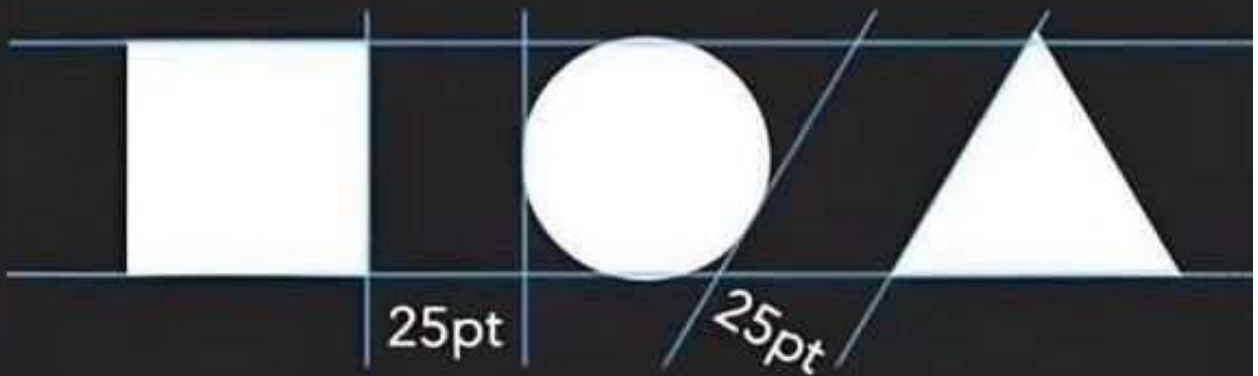
Bad Designer



Average Designer



Good Designer



Much of Graphic Design is based on Perception, Cognition and Psychology...

lastly, you will read this

Copyright Section 31™

Later on, you notice this

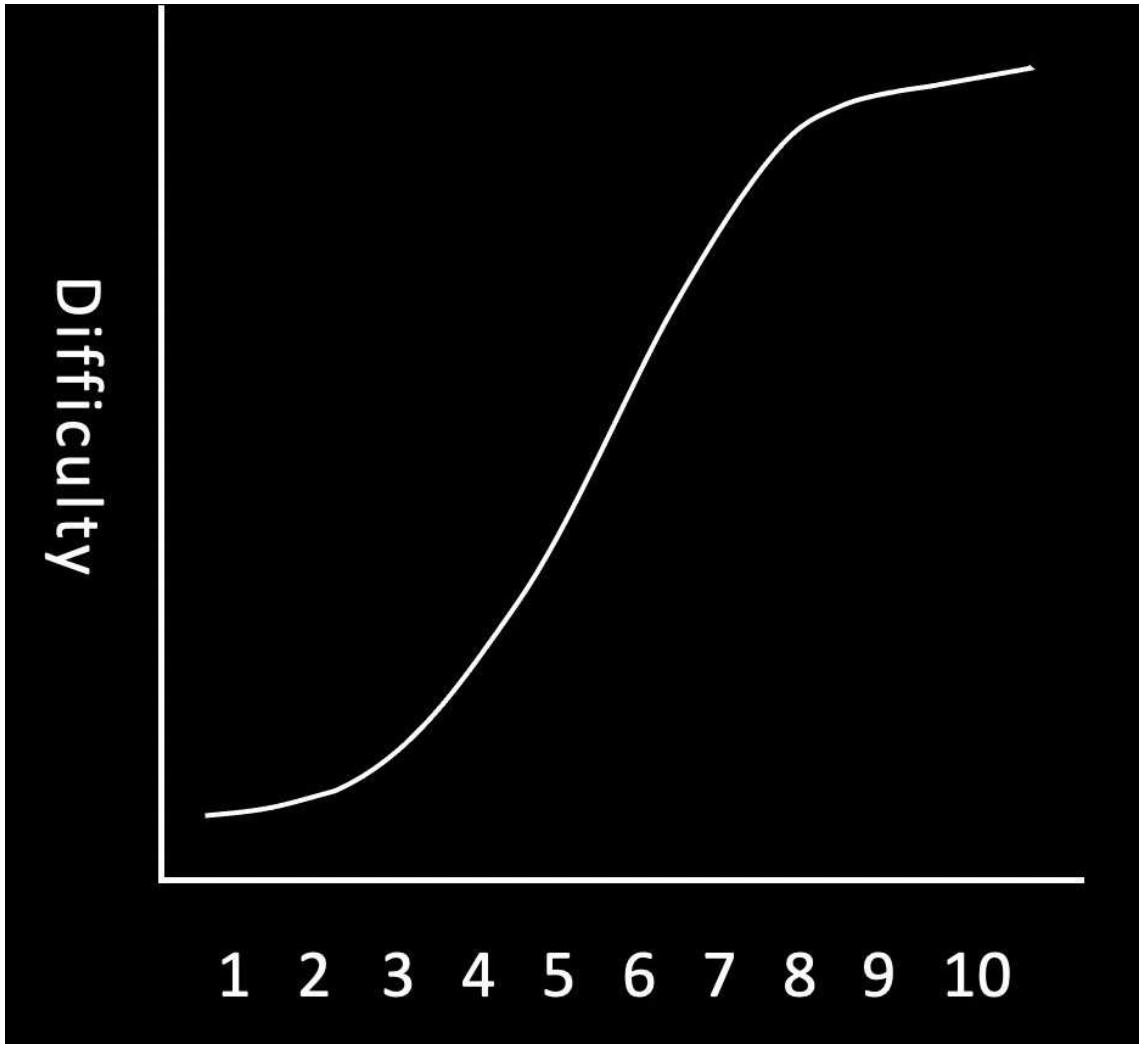
A Word of Warning !

This unit starts off very slowly and gently :o)
(to make sure that everyone is "on board")

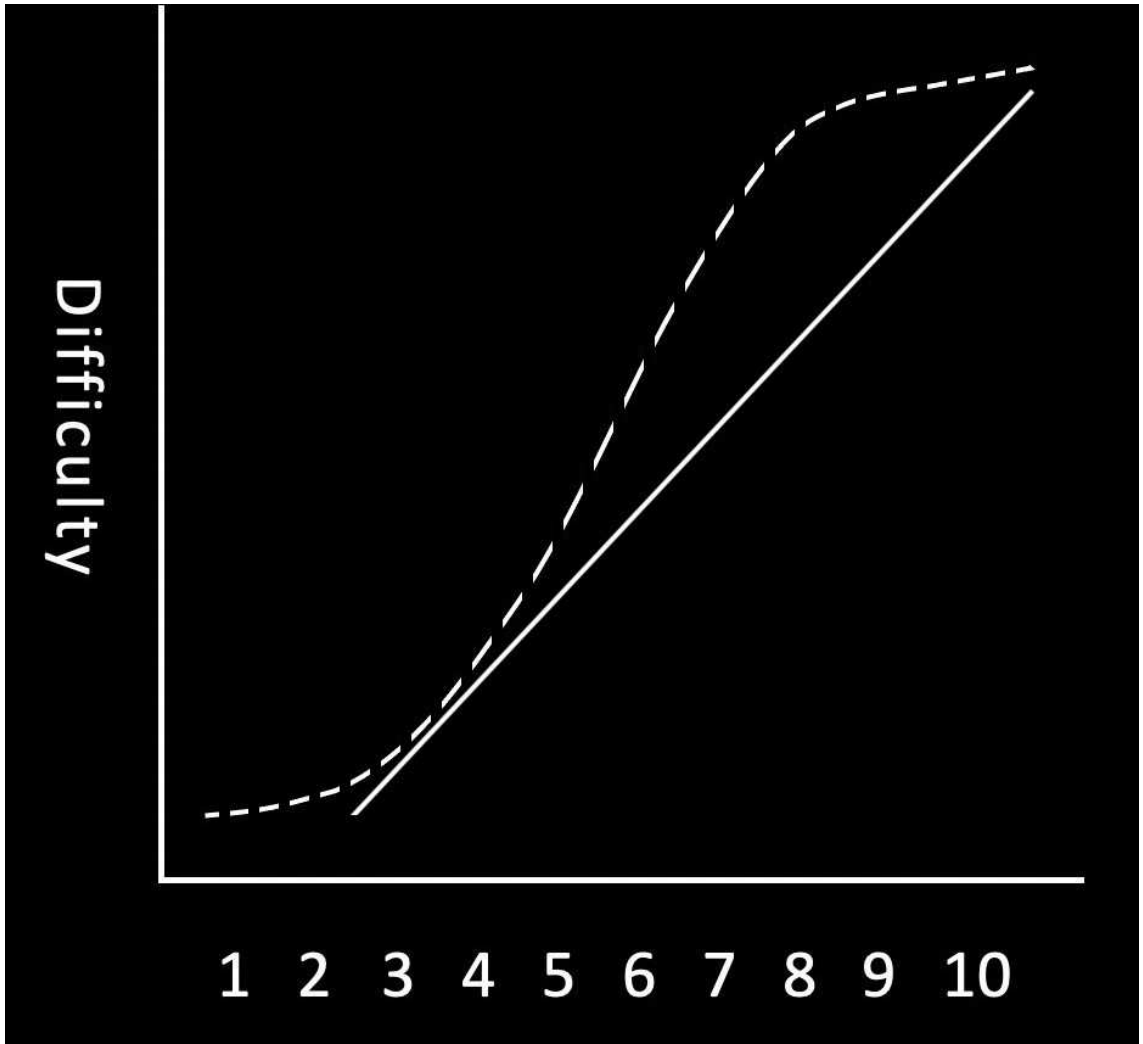
BUT it will speed up incrementally as we progress
DON'T get left behind - keep your eye on the ball !

It is NOT particularly hard to pass the unit, but...
Every year there are a bunch of fails and resits !!!

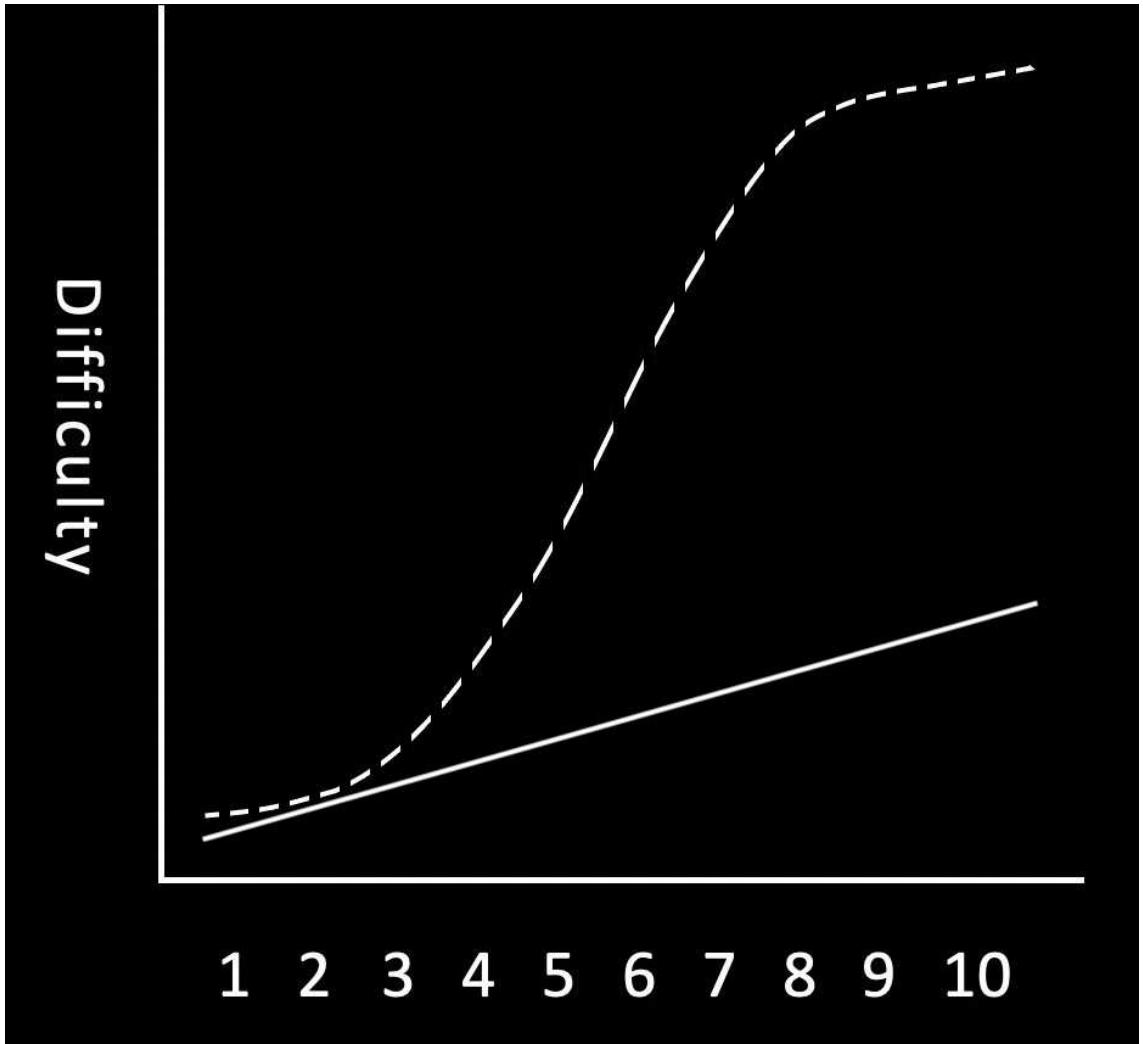
The unit "Learning Curve"



What people usually do



What NOT to do



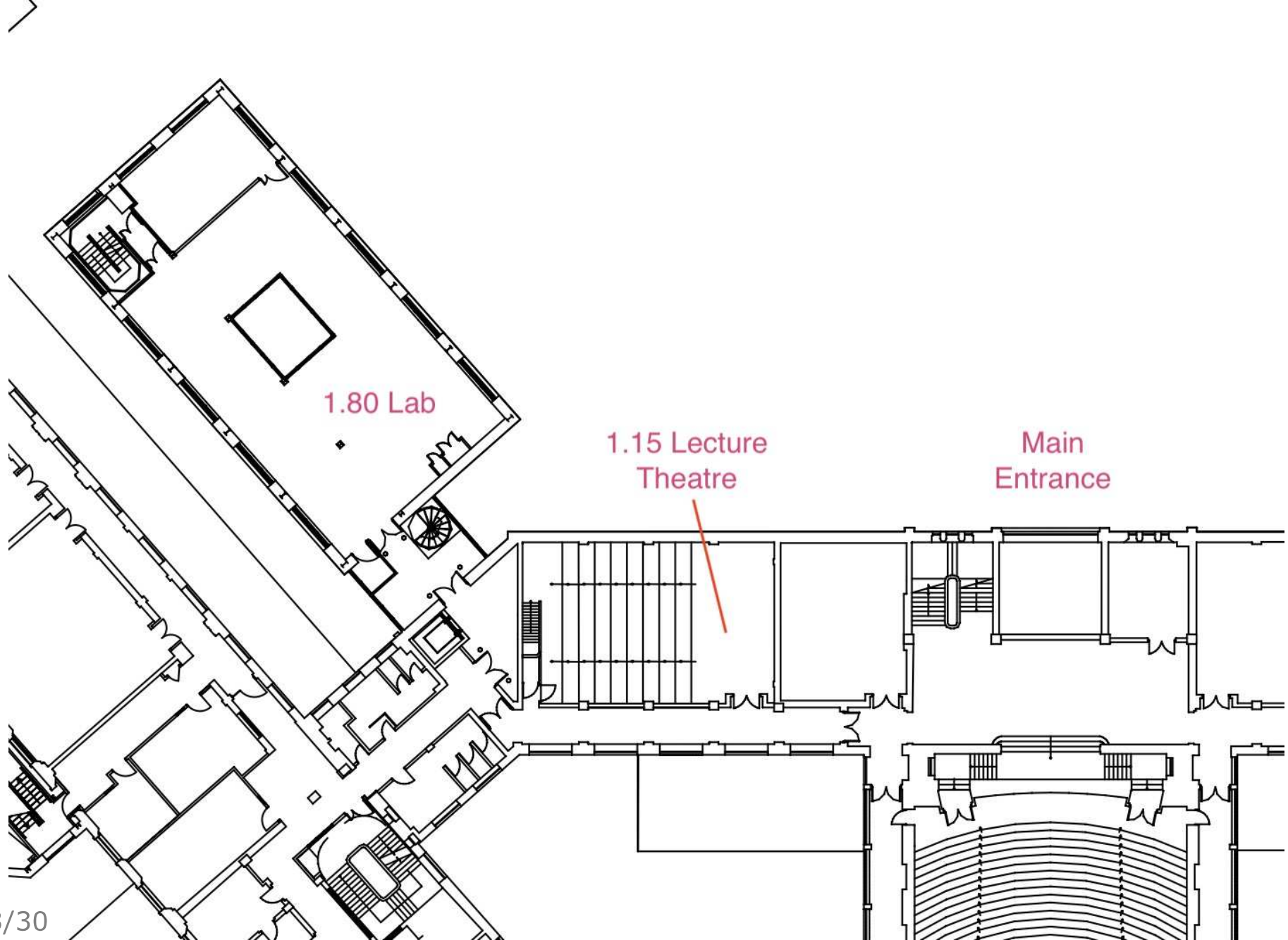
Weekly Activities

Each week we'll release a "workbook" of activities
(We'll discuss these in more detail in later slides)

Lecture briefing to introduce current week's topics
Tuesday at 11am in Queens Building 1.15 (SLT)

Practical session to support completing workbook
Friday 9am-11am Queens 1.80 (sorry about that !)

Asynchronous support via Teams discussion forum
(There is a link to this on the unit Blackboard page)


1.80 Lab
1.15 Lecture Theatre
Main Entrance

Weekly Workbooks

Each workbook contains a set of tasks to complete
Lead you step-by-step towards a practical end-goal

Key concepts introduced with rich-media materials:
Text, recorded audio, diagrams, 3D animations etc

Workbooks are made available via GitHub each week
(For ease of upload and download - just do a pull !)
A link can be found on the unit Blackboard page

Importance of Weekly Workbooks

It is ESSENTIAL that you keep up with practical tasks
You will need to work steadily throughout the term

This unit is definitely NOT one where you can just:
"cruise thru lectures; then do all the work at the end"

Workbooks are the backbone of the teaching materials
"An integrated bundle of theoretical concepts & activities"
Must engage with them, even if doing "exam only" unit

Assessment

MAJORS [20 Credit Point Unit]

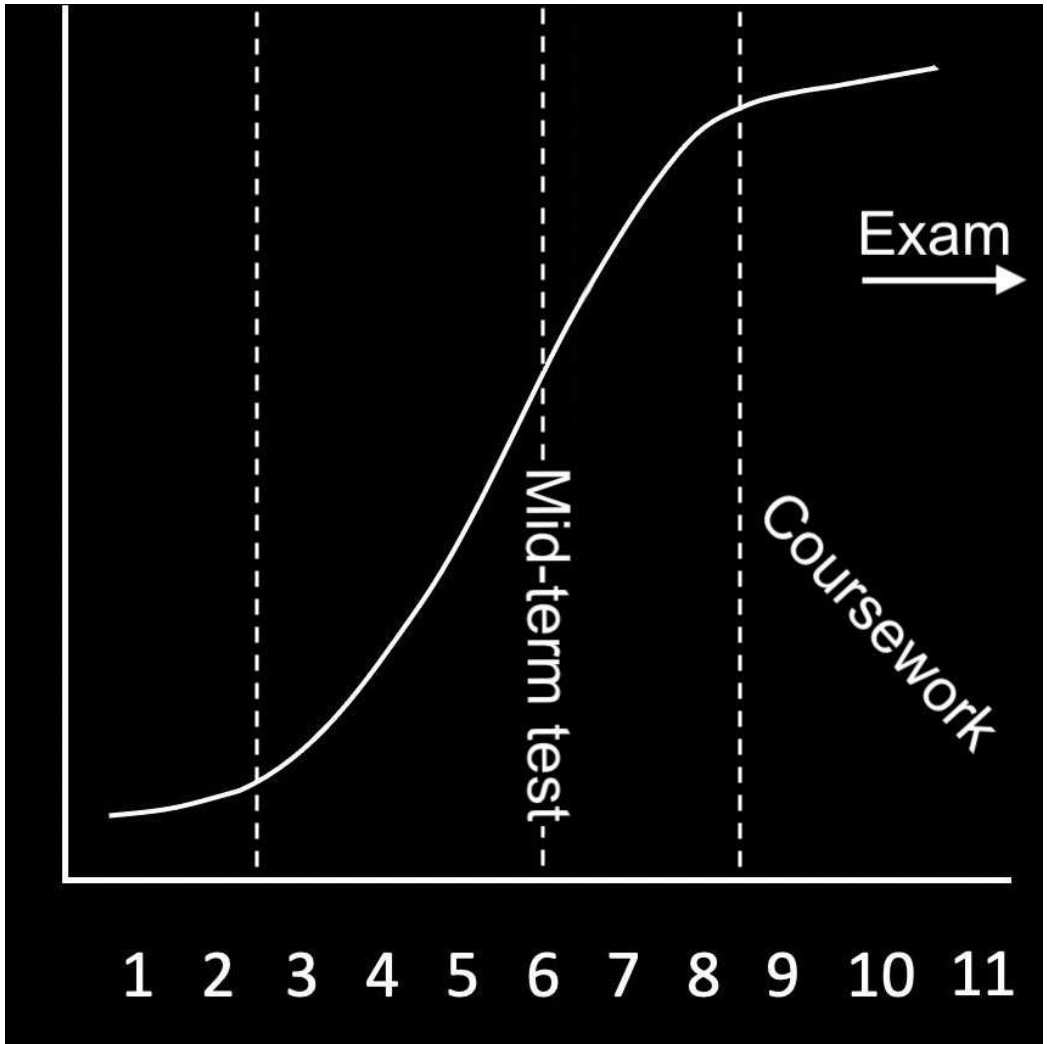
Closed-book on-paper in-class written test [30%]
(to be scheduled sometime during reading week)
Practical coursework assignment in weeks 9-11 [70%]

MINORS [50% of a 20 Credit Point Unit]

Closed-book multiple-choice exam (in December)

A very practical topic, naturally suited to coursework
Exam variant provided for students with CP constraints

Key Activities



Nature of MAJOR In-Class Test

In-class test will be fairly Mathematical in nature
Apply principles of Computer Graphics "on paper"

It is essential to complete workbook practical tasks
(to gain a deep understanding of all the techniques)

Test will only cover the first 5 weekly workbooks
(Coursework will encompass the full 7 workbooks)

Nature of MINOR Exam

There will be a new style of exam this year !
In previous years, exam involved much calculation

Compression of TB1 has reduced engagement time
(Not enough time for minor students to practice !)

New exam will be higher-level "general knowledge"
It will cover content of all 7 core weekly workbooks

A mock exam paper will be released a little later
(actual past papers aren't an accurate indicator)

Minimal use of Blackboard

A set of bookmarks that point to other platforms:
GitHub, Teams, Replay etc.

There are TWO Blackboard pages for this unit:

- The "Teaching Unit" page
- The MAJOR page (Test & CW)

There is a link on MAJOR page to the "Teaching Unit"
(Just so you don't get lost !)

You need to submit coursework via the MAJOR page
(Which is why we can't just get rid of it completely)

Implementation

We will be using C++ for implementation
(Pretty much the standard for low-level graphics)

For drawing to the screen, we will use "SDL2"
Platform-independent graphics library, used to:

- Create windows to show on the screen
- Manipulate individual screen pixels
- Allow user interaction via keyboard & mouse

Everything else YOU will build from scratch !
(Although we'll use a few maths functions from GLM)

"Suggested" Textbooks

For a basic introduction to topics covered in the unit:

Computer Graphics from Scratch

by Gabriel Gambetta

(ebook available via UoB library website)

More detailed coverage and advanced rendering:

https://www.scratchapixel.com/

You might need to do some additional reading
(especially for coursework "extended" topics)

Questions ?

Why not teach an established graphics API ?

One difficulty is choosing which one to teach...
DirectX, OpenGL, Vulkan or proprietary framework ?

Better to teach the fundamental concepts...
Which make it easy(er) to pick up any framework

Besides, "fundamentals" are much more BSc/MEng
We don't like to focus a unit around a single API

Can I use <insert_language>

Soz, no !

C++ is a well-established standard in the area

All templates/examples are written in C++

Teaching assistants are all skilled in C++

It's hard to mark an unfamiliar language

Need to maintain a level playing field !

The aim of the first practical session

Aim: To compile and run the "RedNoise" project
(the base template for all practical exercises)

The deeper purpose is more significant:
Find a "workable" way to compile and run SDL code
The order of preference is as follows:

`Native OS / WSL+X11 / Virt Machine / Remote Login`

You *could* do all your work on a lab machine...
But using your laptop is going to be more convenient

Use an IDE - if you like

A CMake file is provided to support the use of IDEs
This file describes various required dependencies
As well as defining a number of different build rules

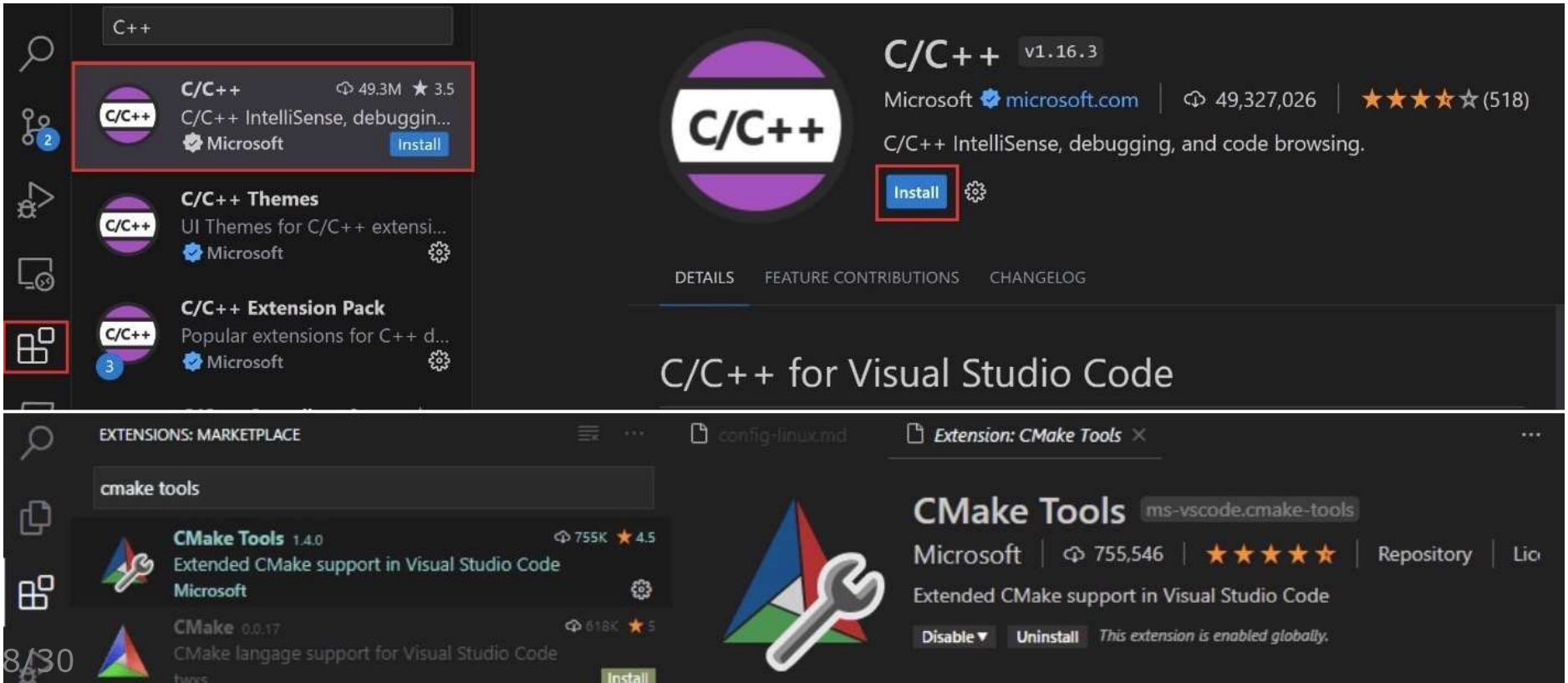
CLion works well - it can open RedNoise as a project
CLion _should_ already be installed on the lab machines
If you are prompted for it, the license server URL is...
http://ls-jetbrains.bris.ac.uk:8080

But hopefully you won't need this URL !

VSCode

Some of you may prefer to use VSCode instead

However you will need to install some extensions...



Disclaimer

You are free to use whichever IDE you like...
However support in the lab sessions may vary
(It all depends the lab TAs having prior experience)

Bottom line: use Visual Studio at your own risk !

Some useful help can be found here:

https://code.visualstudio.com/docs/cpp/cmake-linux

(contains useful info for ALL operating systems)

First Weekly Workbook

You *can* view workbooks online via GitHub...

https://github.com/COMS30020/CG2025

But it is better to clone the repo and view them locally:

README

Note: I have a markdown browser plugin installed

You could view index.html instead (but not as pretty):

index