

CSC3231: Graphics for Games

Lecture 1: Introduction

Gary Ushaw

Introduction



Rendering

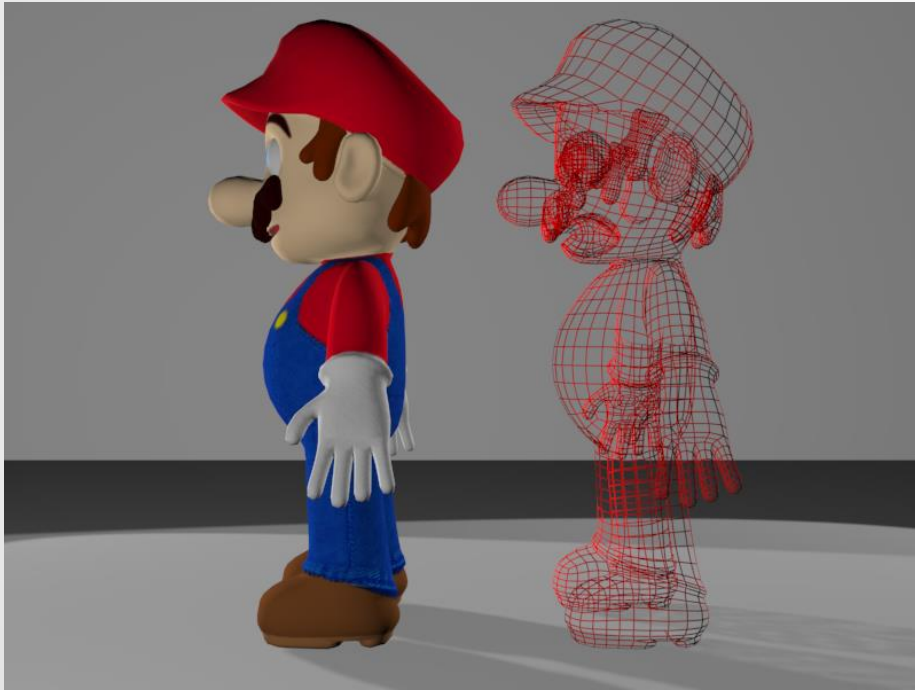


Pixar: 36
hours per
frame



Nintendo: 33
milliseconds per
frame

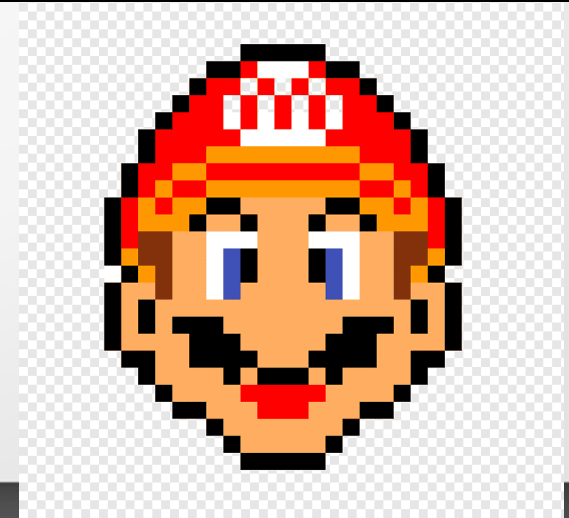
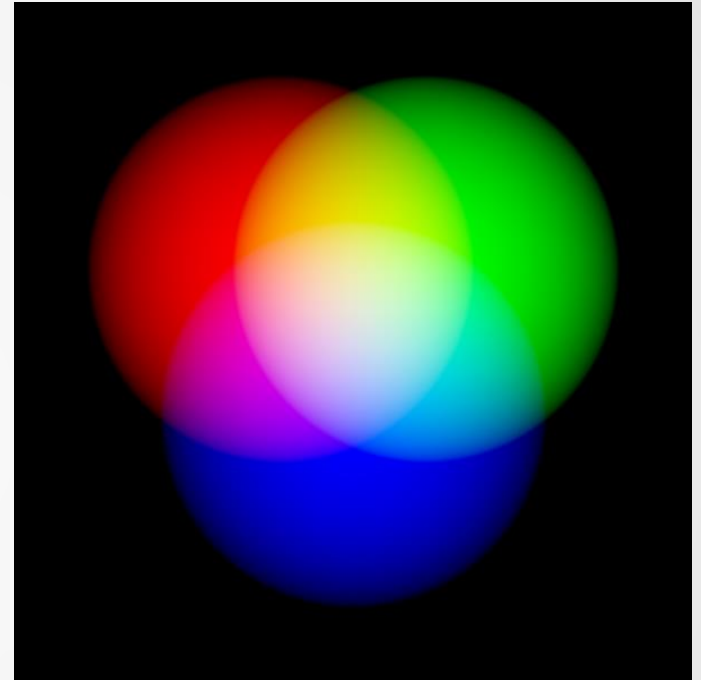
3D Models



- A vertex is a point in space
- A primitive connects vertices to form a polygon
- Multiple polygons make a model

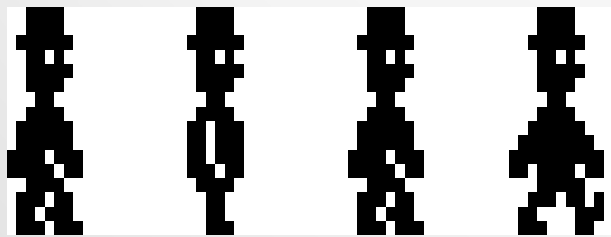
The Screen

- The screen is a matrix of pixels.
- More pixels = higher resolution.
- Each pixel's colour is determined by a RGB value.
- There is a dedicated area of RAM containing the matrix of pixel colour values to display on screen.
- 1080p is 1920 x 1080 pixels.
- That's over 2 million pixels to be generated every frame.

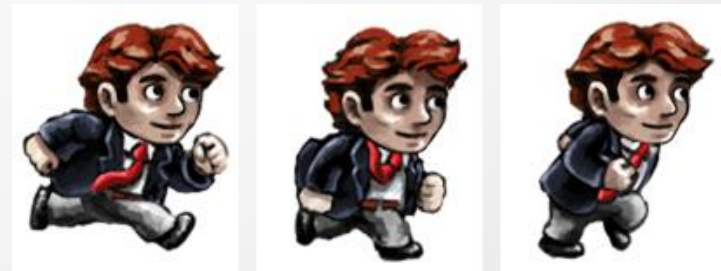


2D Game Graphics

- Early graphical games used *sprites*
 - Simple pictures (“bitmaps”) in memory copied into graphics memory (*blitting* – *bit level image transfer*)
 - Early graphical hardware could 'float' an extra image on top of a back image plane, like a ghost / “sprite”



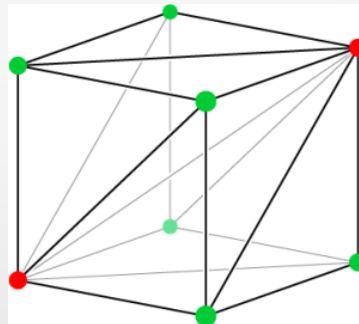
Jet Set Willy - 1984



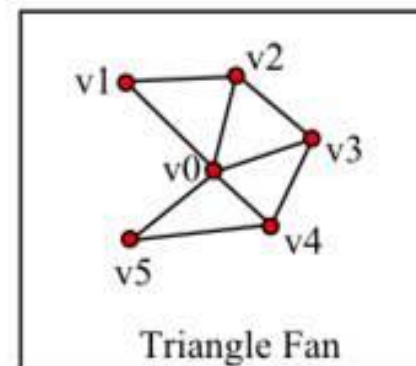
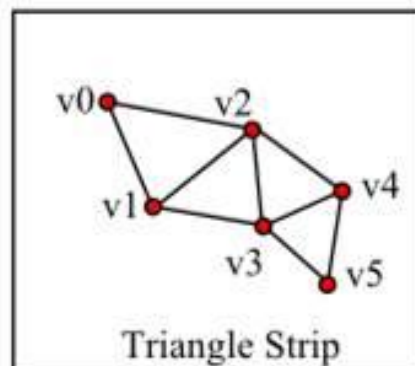
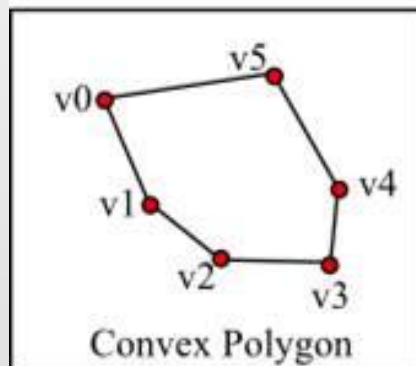
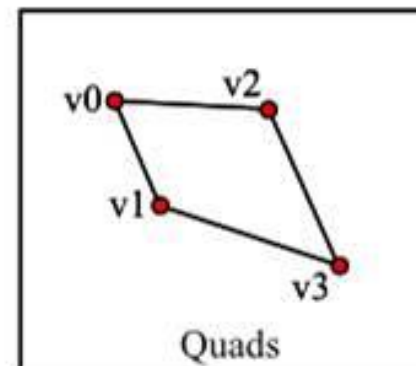
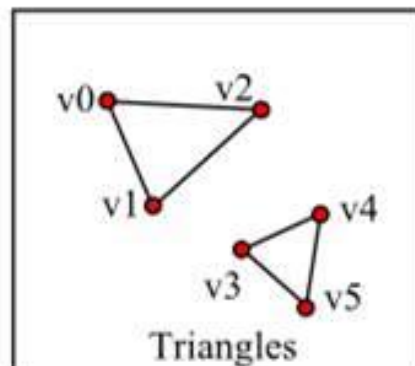
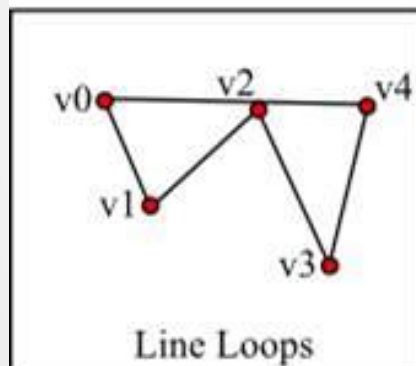
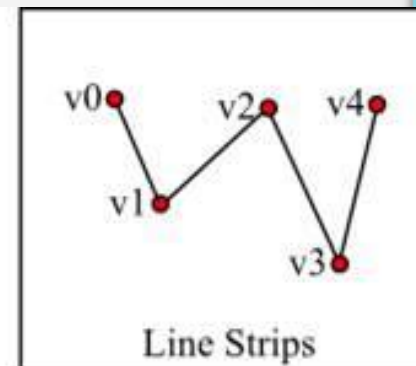
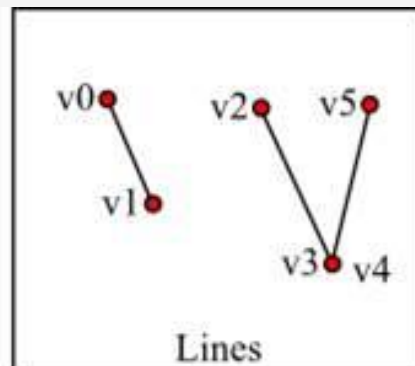
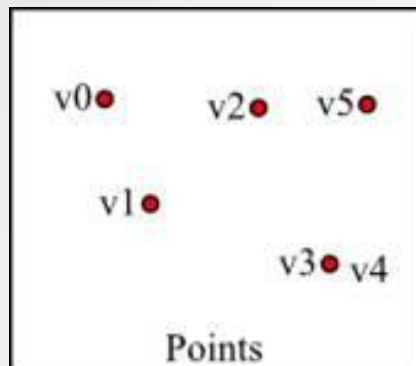
Braid - 2008

3D Game Graphics

- Rasterisation is the process of turning three dimensional shapes into pixels.
 - Vector-based shapes (no curves)
 - These shapes are made up of *vertices* (positions in a space, usually 3D), which are connected to form *primitives*
 - These primitives are then *transformed* (moved, rotated, scaled up or down) to form their final position on screen

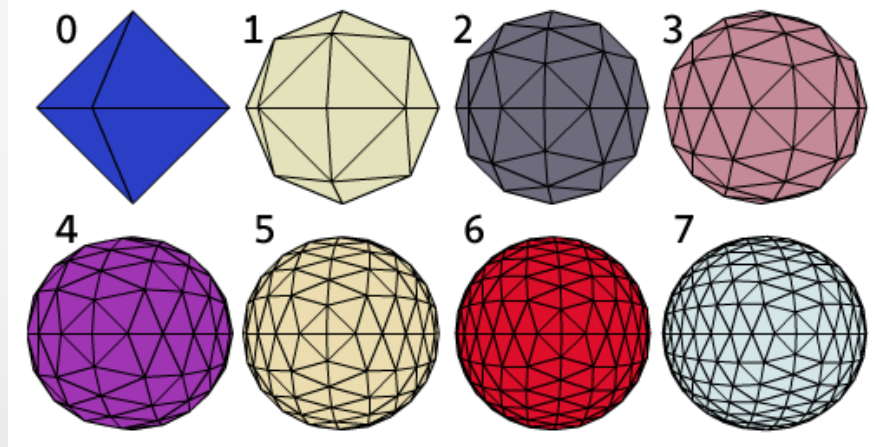
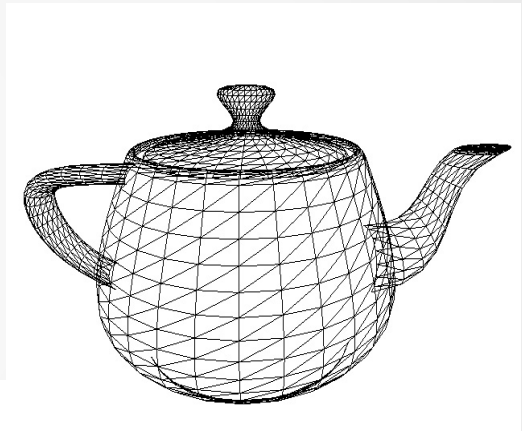
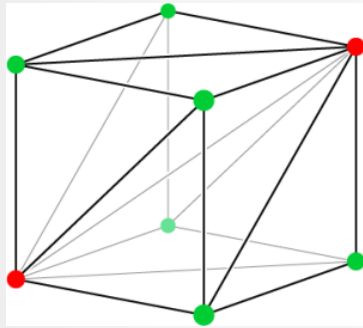


Rasterisation Primitives



Rasterisation Primitives

- These primitives are positioned together such that they form a solid shape, or *mesh*



Triangles

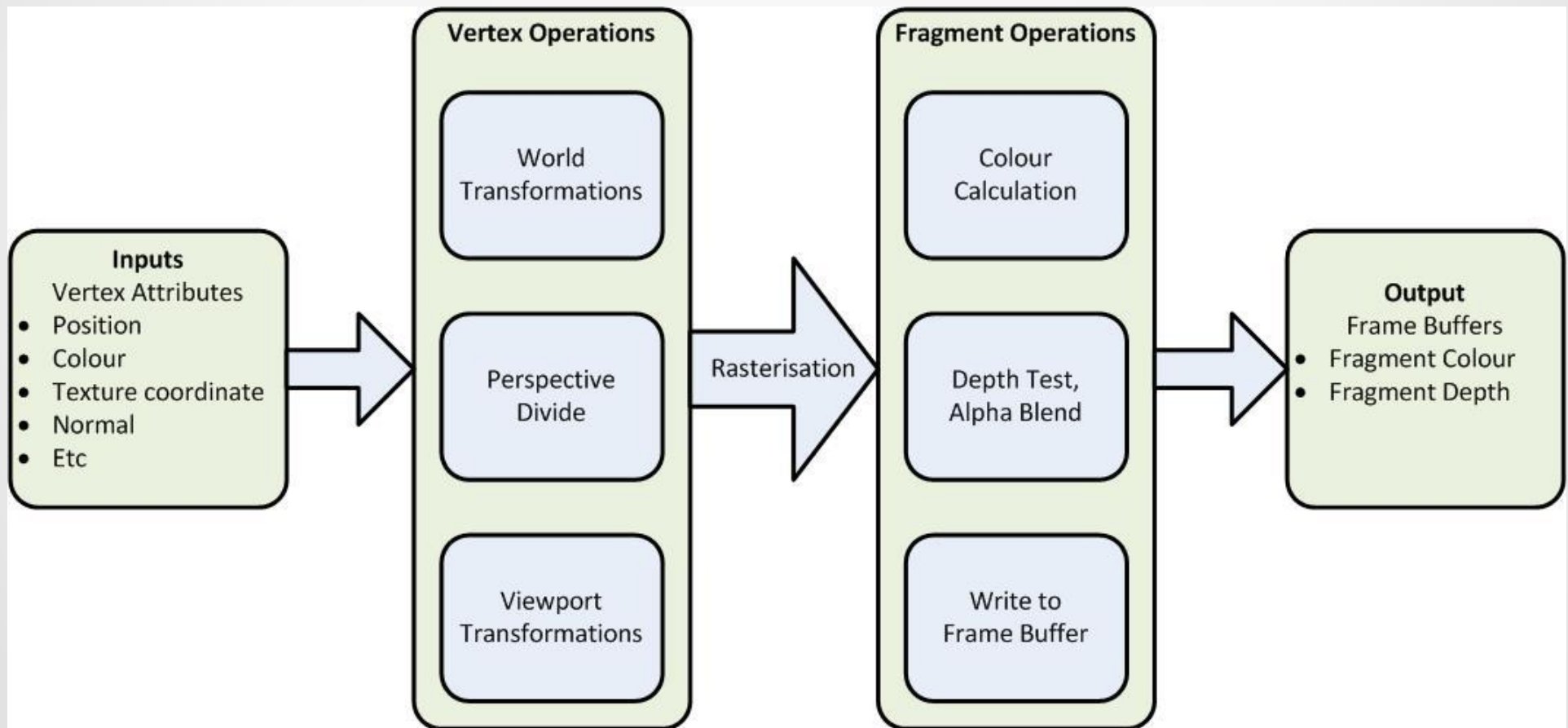


Textures

- Textures add detail to primitives.
- A bitmap of RGB colours is sampled



Rendering Pipeline



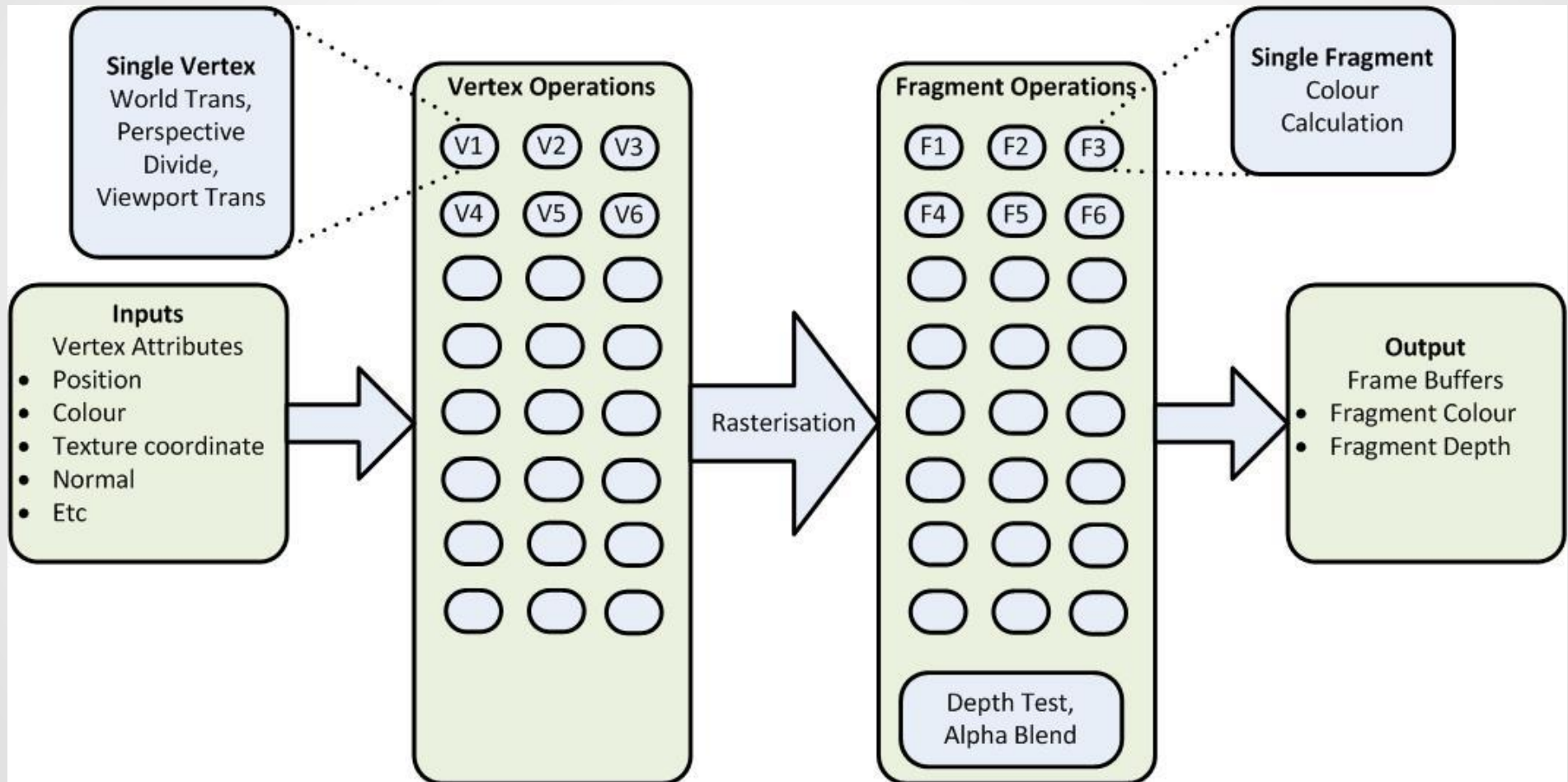
Vertex Operations

- All parts of the scene begin as vertices
- Primitives are processed through the pipeline and build up the final image
- Vertices are transformed via a number of operations, until they are in screen coordinates
- Primitives outside screen coordinates are culled
- Vertices outside screen coordinates are clipped
- The 3D scene is projected onto a 2D plane
 - This makes far away objects smaller

Fragment Operations

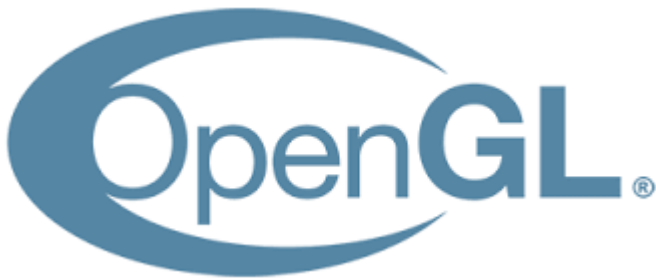
- Once a primitive's screen position is determined, it can be 'filled in'
- Rasterisation scans the primitive, and determines which fragments the primitive covers
 - A fragment is a unique sampled position on the surface, and the data required to shade it
 - A fragment is usually a single pixel on the screen
 - A pixel might be part of multiple fragments
- Each of these fragments is *shaded*

Parallelised Graphics Pipeline

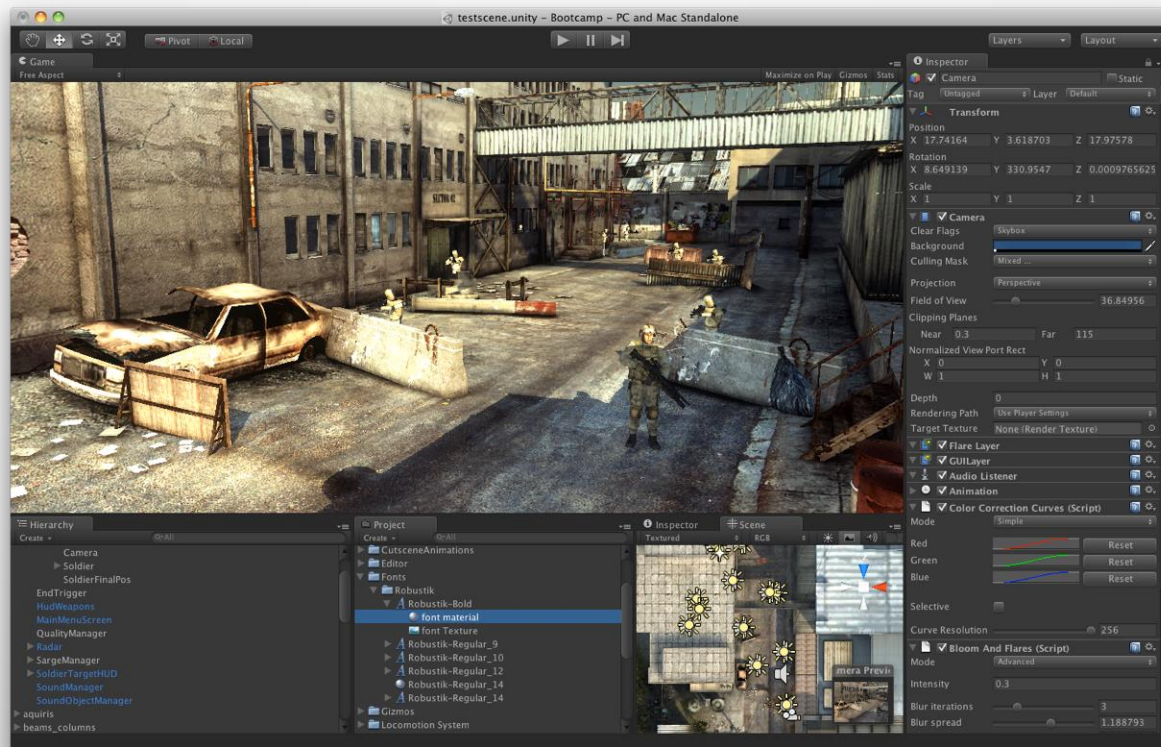


Graphics Hardware APIs

- How do we program for 2880 simultaneous core executions?
 - Simple answer: We don't
- Instead, access to the GPU is abstracted behind an Application Programming Interface (**API**)
 - Generally these APIs hide all the tricky stuff
 - No manual scheduling / copying / synchronisation



Game Engines



Graphics for Games



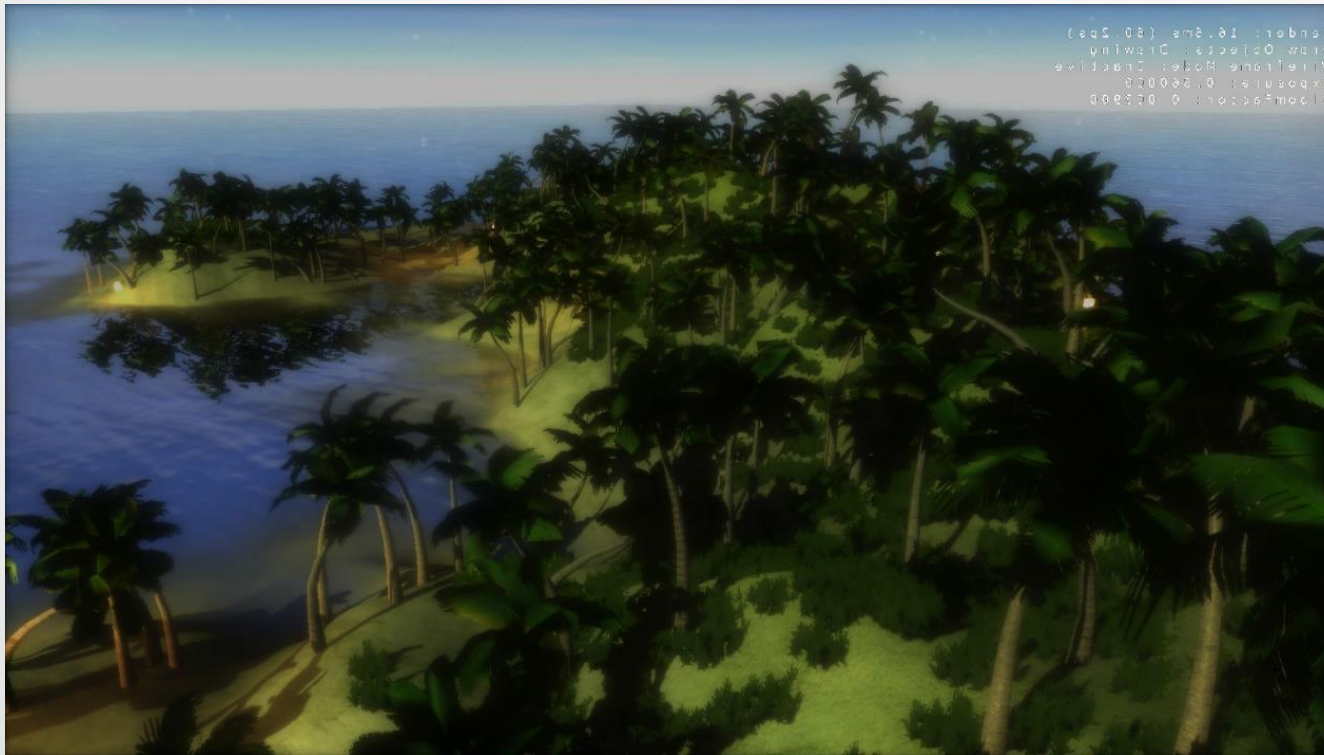
Zelda image from Nintendo

Course

- 10 Lectures to introduce concepts central to graphical rendering
- 12 Tutorials to implement these concepts in practice
- 1 piece of coursework to demonstrate understanding of material (50%)
- 1 written exam (50%)

Coursework

- Coursework with open-ended specification



- 3D models
- Perspective
- Textures
- Lighting
- Shaders
- Animation
- Complex scene

Graphics for Games



Skull and Bones image from Ubisoft