UNIVERSITY OF MALTA
Institute of Digital Games

Creating Virtual Worlds

Lecture 2

Primitives and Polygons

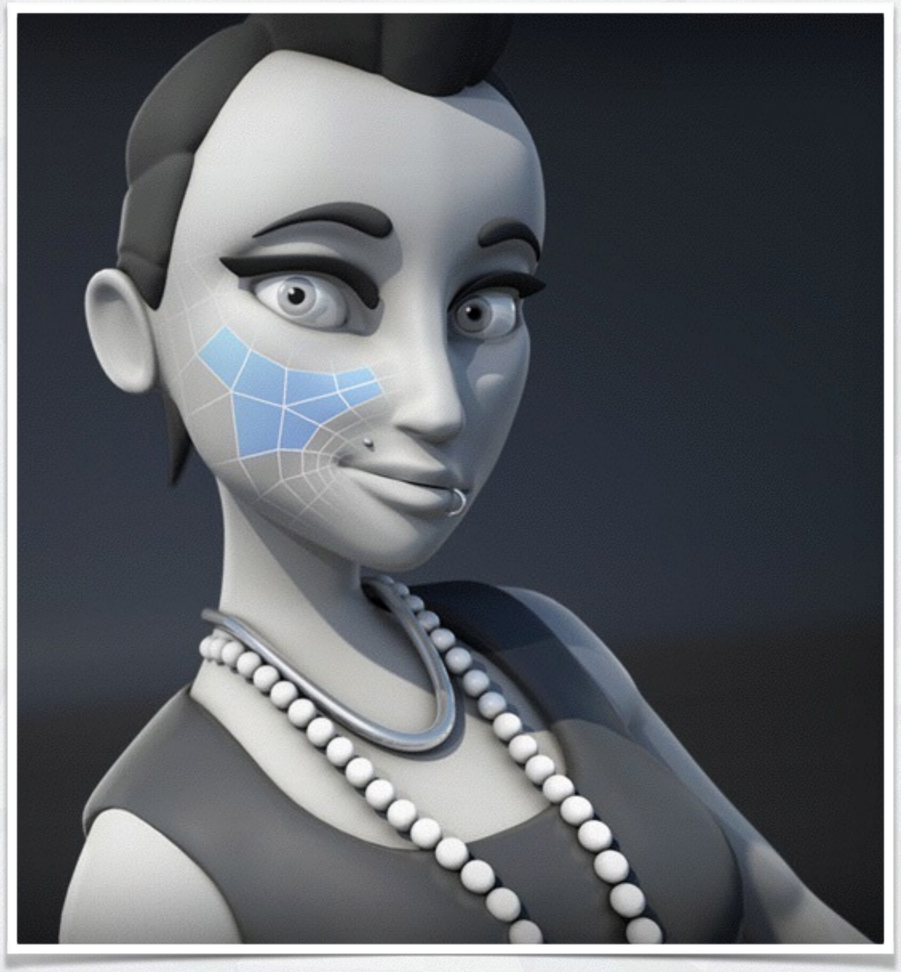
Our topics today …

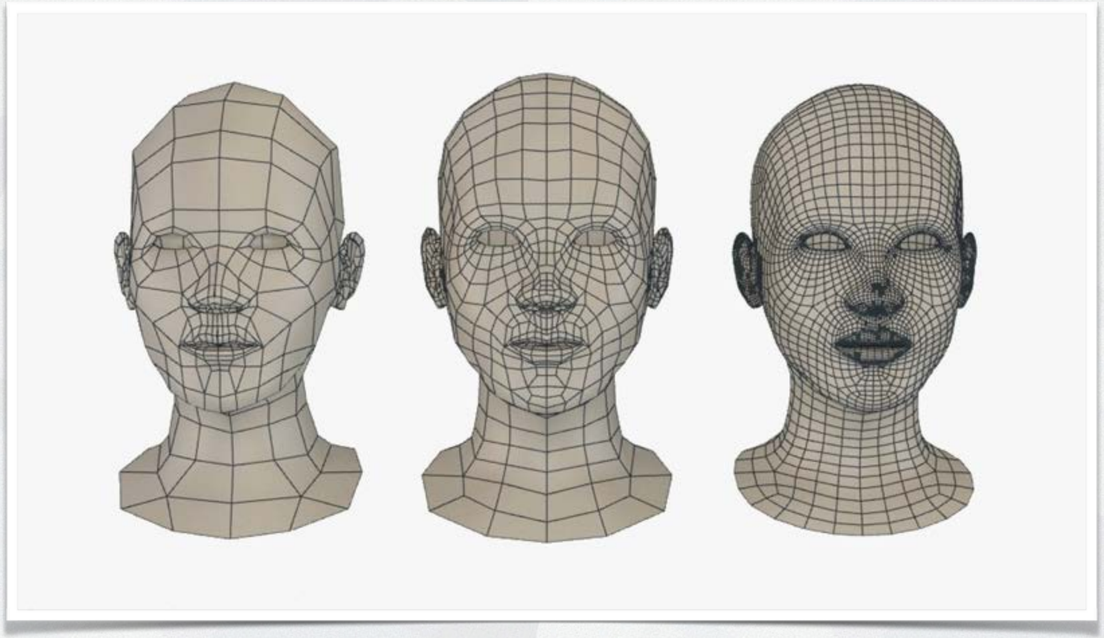
- **Points, Edges, Polygons, & Normals**
  Understanding the basic components of 3D models
- **Virtual Space**
  Working with three axes
- **Global and local coordinates**
  Positioning, transforming, and grouping objects
- **Unity Demonstration**
  Putting theory into practice

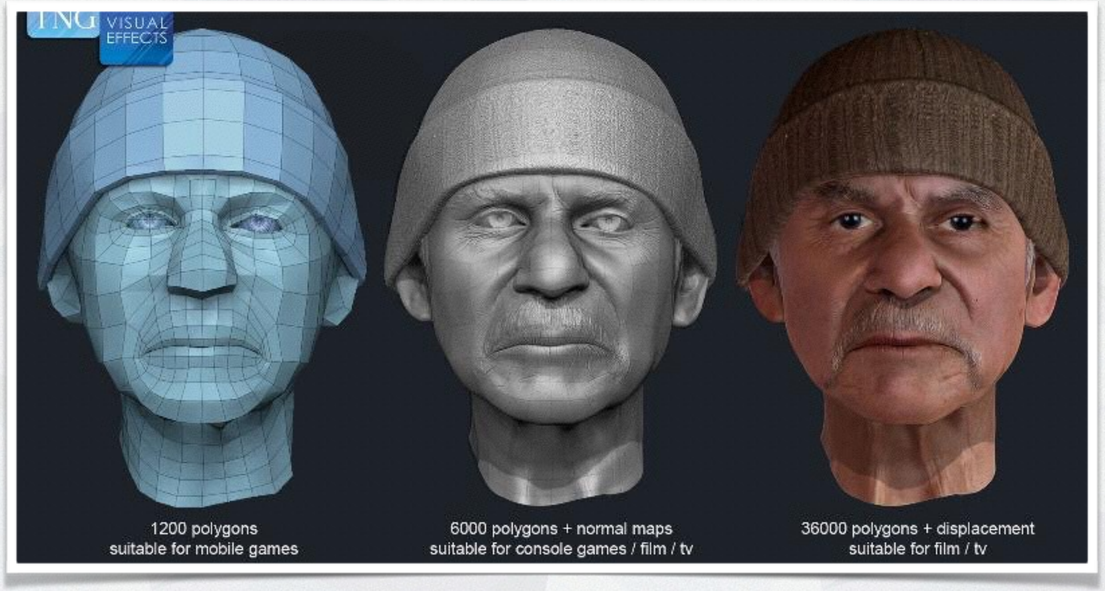
3D Essentials

Computers are capable, through specialised software, of showing **mathematical representations** of 3D objects

3D models represent a physical object by using a **collection of connected points** in 3D space.



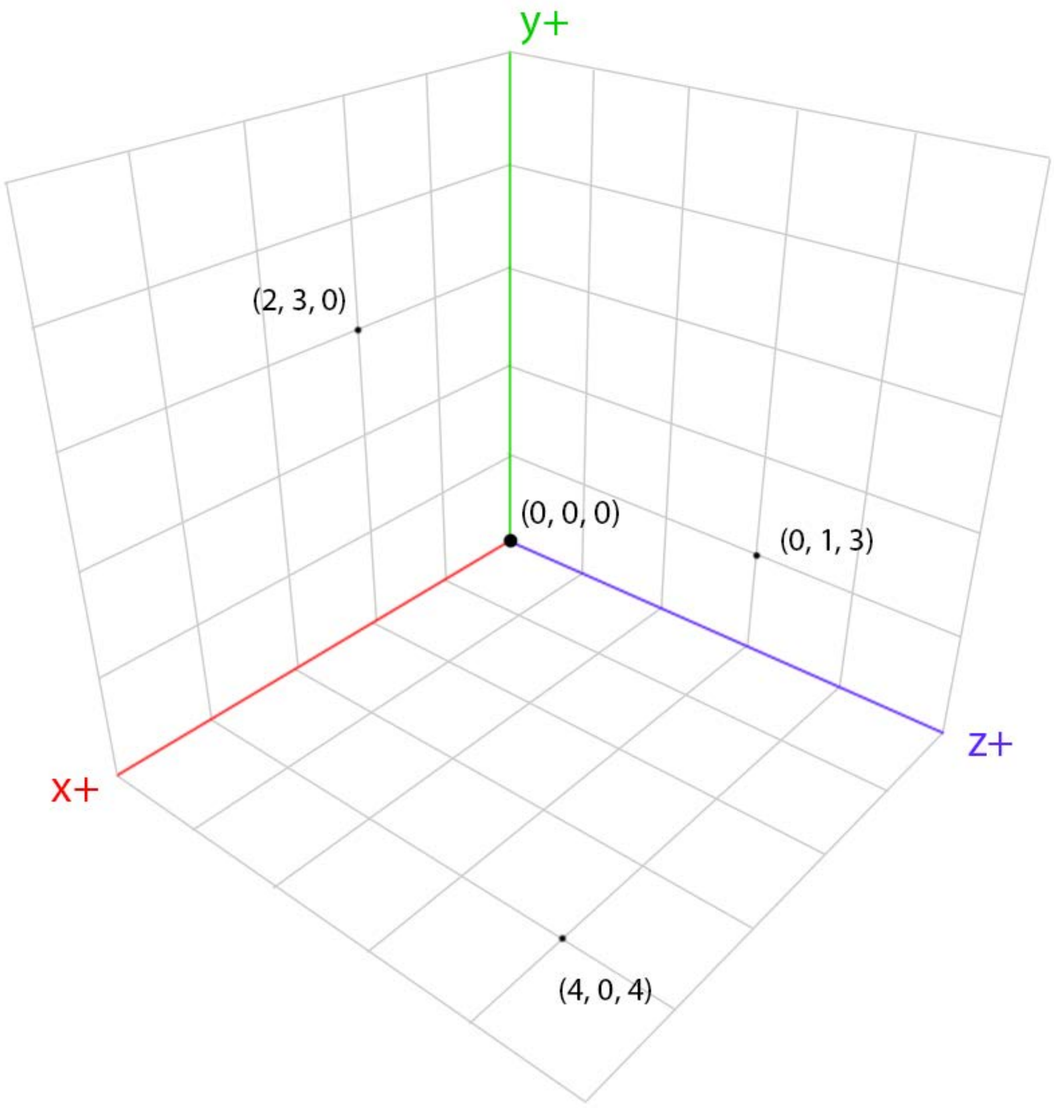




Every 3D object is made up of points connected by triangles, lines, curved surfaces, etc.

More points mean a higher resolution, and therefore more detail

And more demand on the computer!

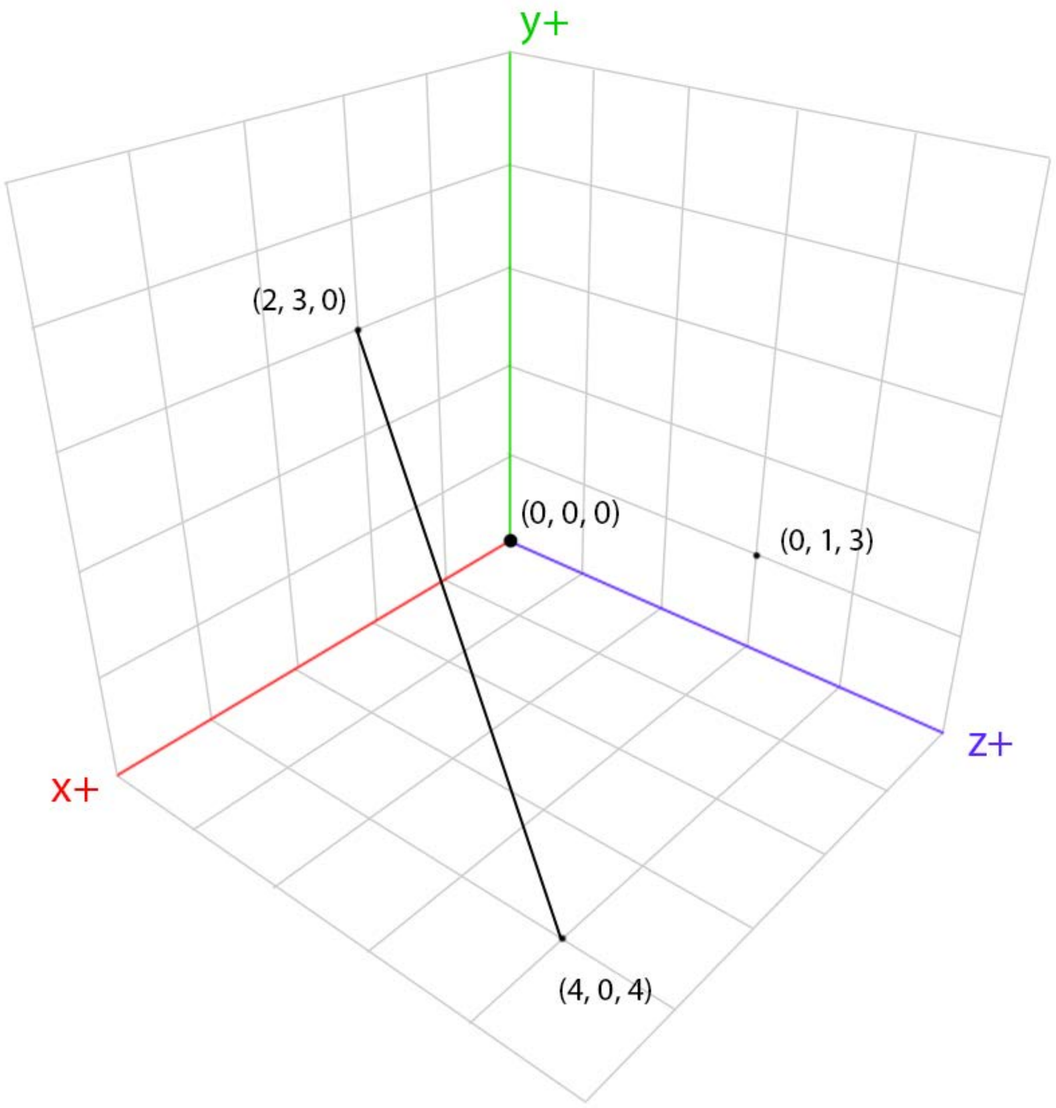


3D space is defined by three axes; X, Y, and Z

Every point in a scene has coordinates along these axes

The origin of a given 3D scene is point 0, 0, 0

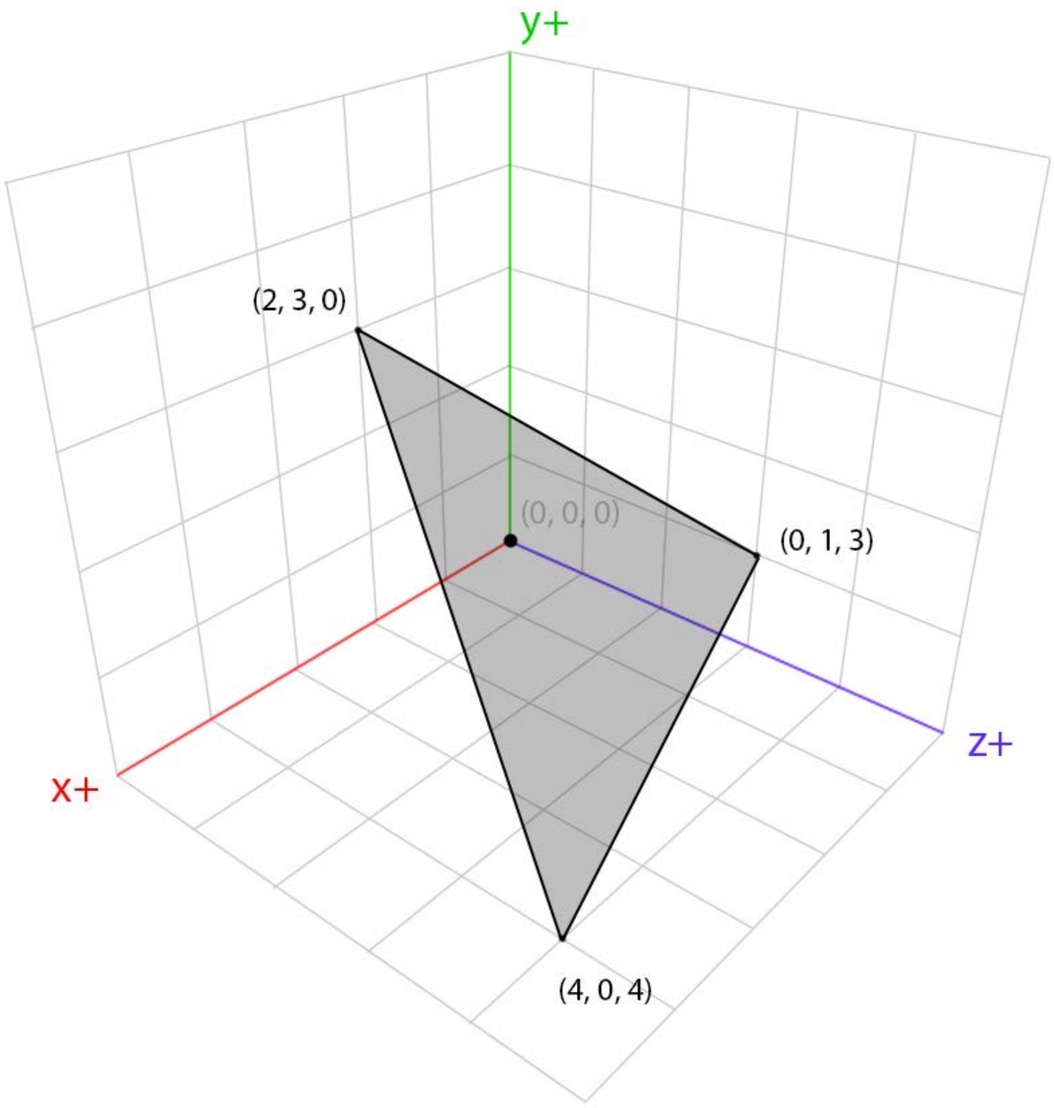
Other points have coordinates relative to the origin



Two points can be connected with a line

However, a line has no “mass”, or “depth”. We can visualise it, but it has no existence in 3D space

We need at least three points to define a surface

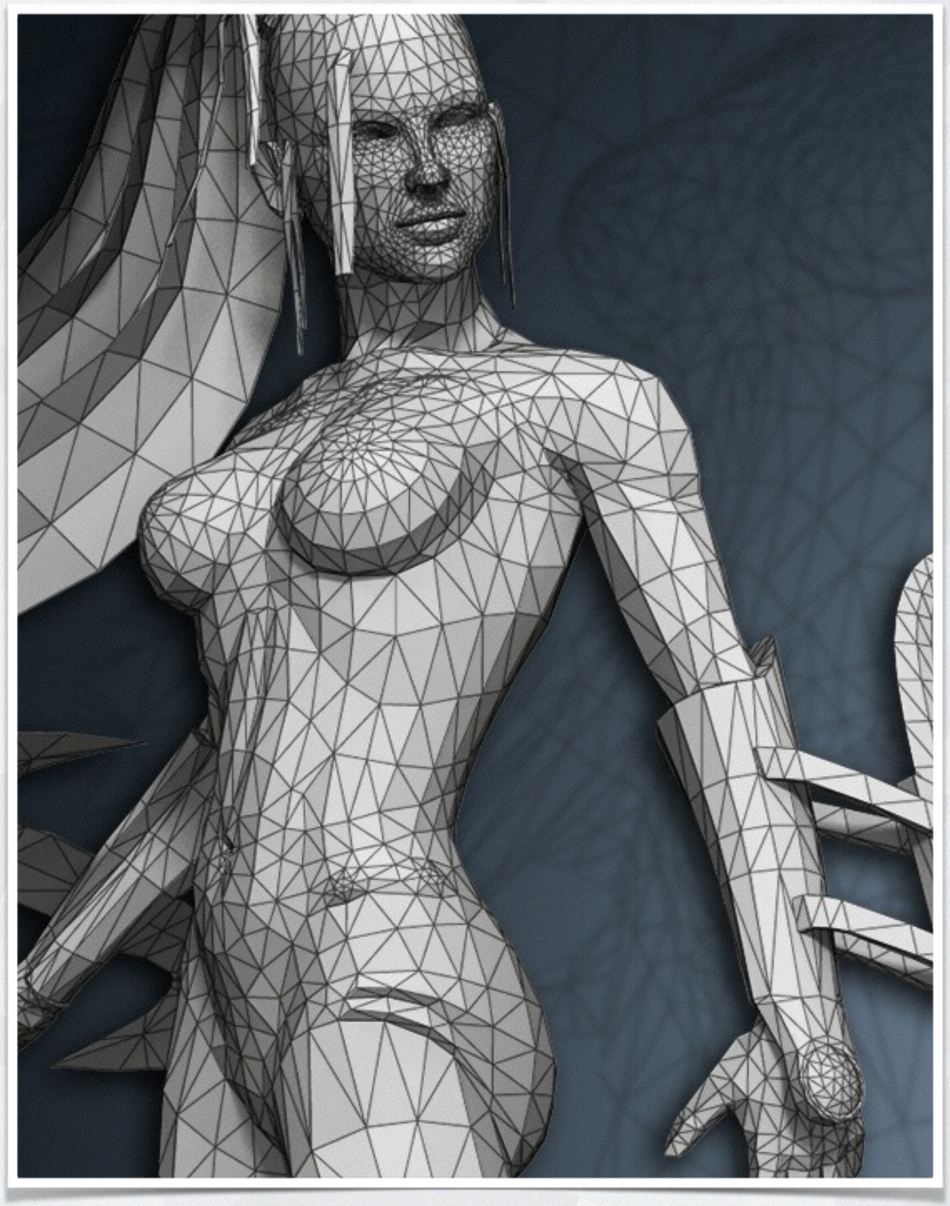


A **triangle** is the simplest 3D shape we can make

3D shapes can have more than three points, like rectangles, pentagons, etc.

In general, we refer to these shapes as **polygons**

But computers love triangles!



3D objects are made up of points, or **vertices**, in 3D space with **XYZ coordinates**

Connected, they form **polygons**

We also refer to the polygons as the **faces** of the 3D model, and the lines between them as **edges**

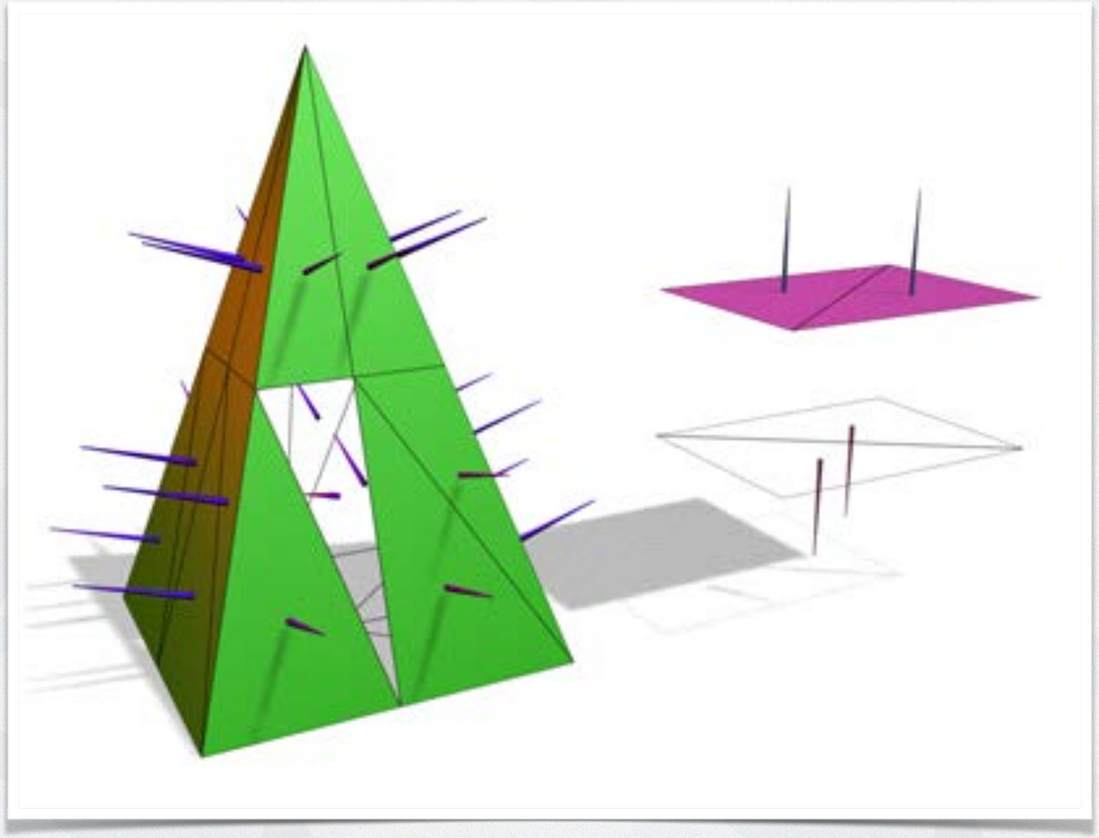
It is good practice to make a model out of **quads**

More points are added to areas that need more detail

Reasons for this are, for instance, **shape definition** and **animation requirements**

Important if you make your own models, but we won't do that here

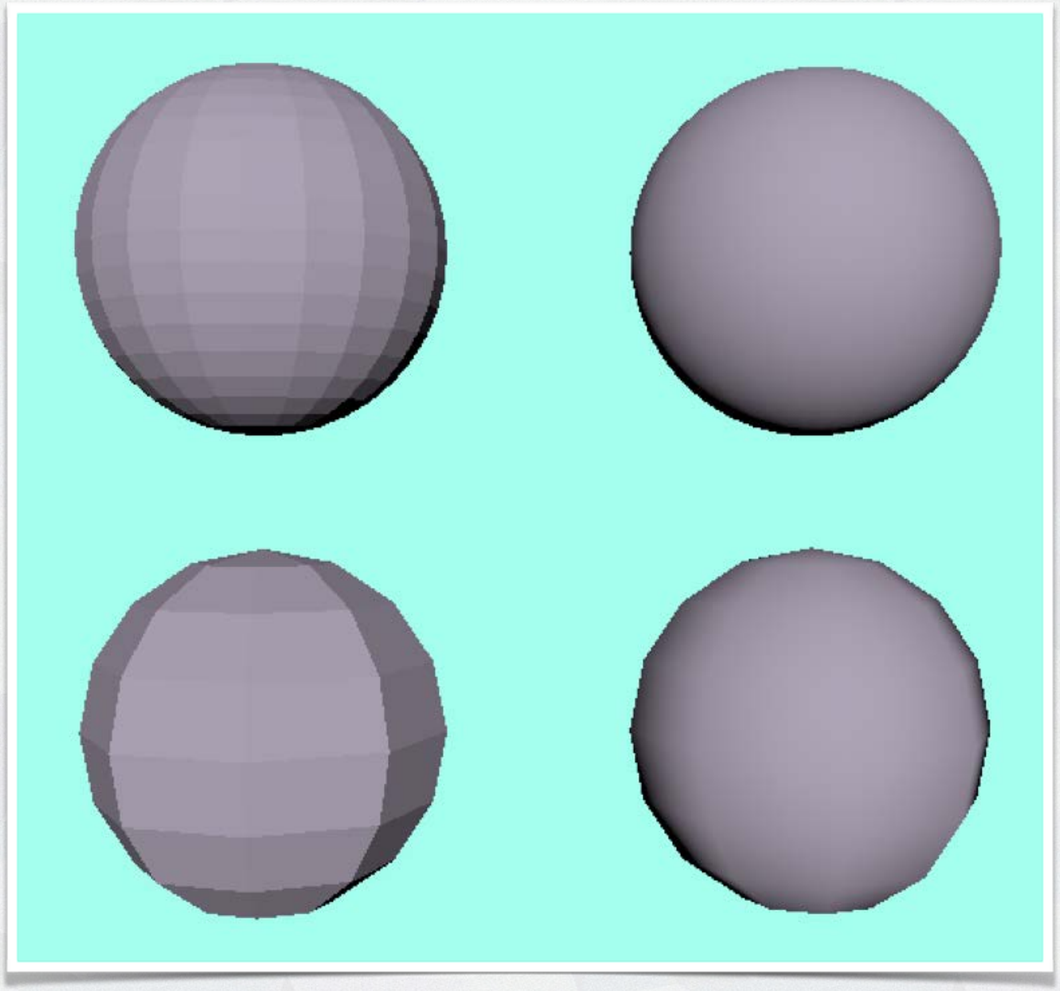
One more thing…



Every face has an orientation, or surface normal

It determines the angle at which light is reflected off the face

More on this in the class on Materials and Textures

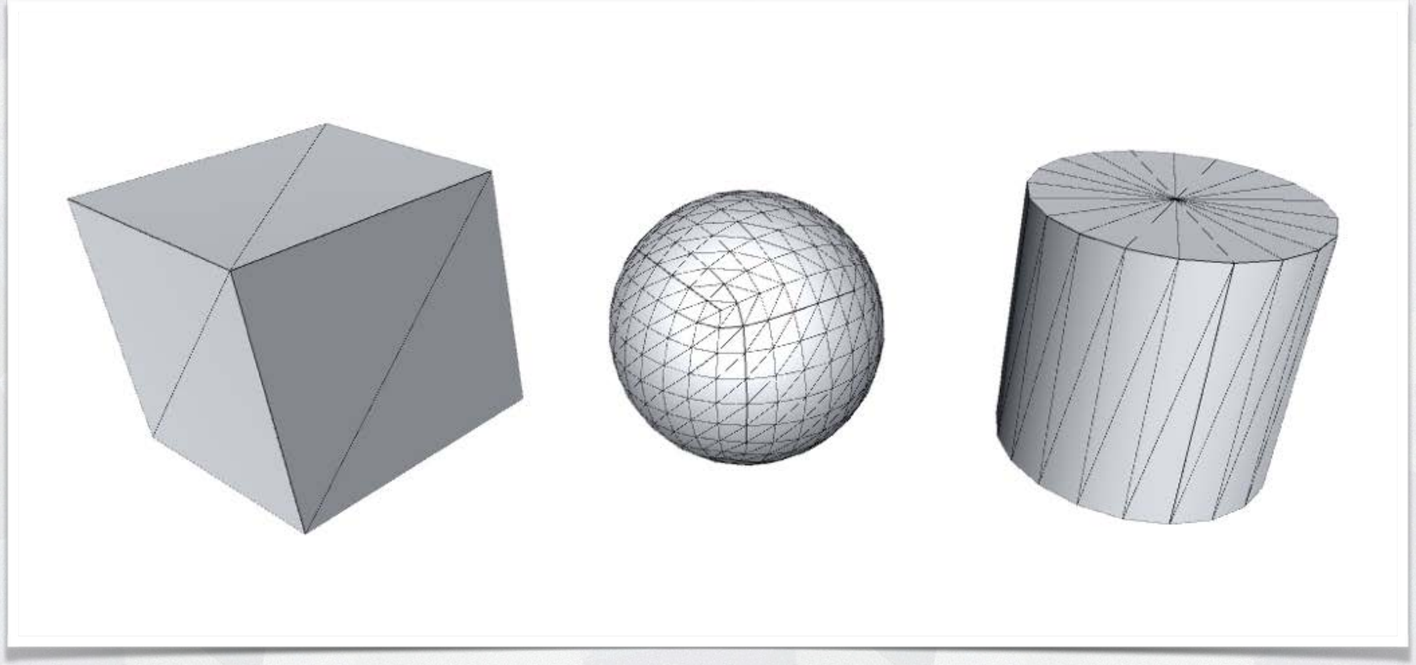


Edges can be made ‘hard’ or ‘soft’, depending on the angle of the normals

This can drastically change the appearance of the object

When you have issues with lighting, normals are often the first place to look

Primitives

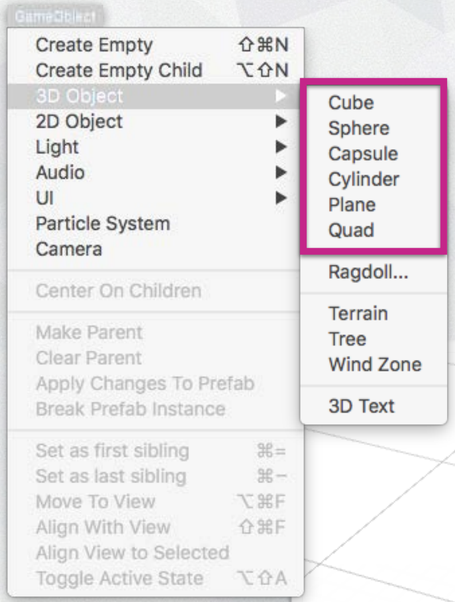


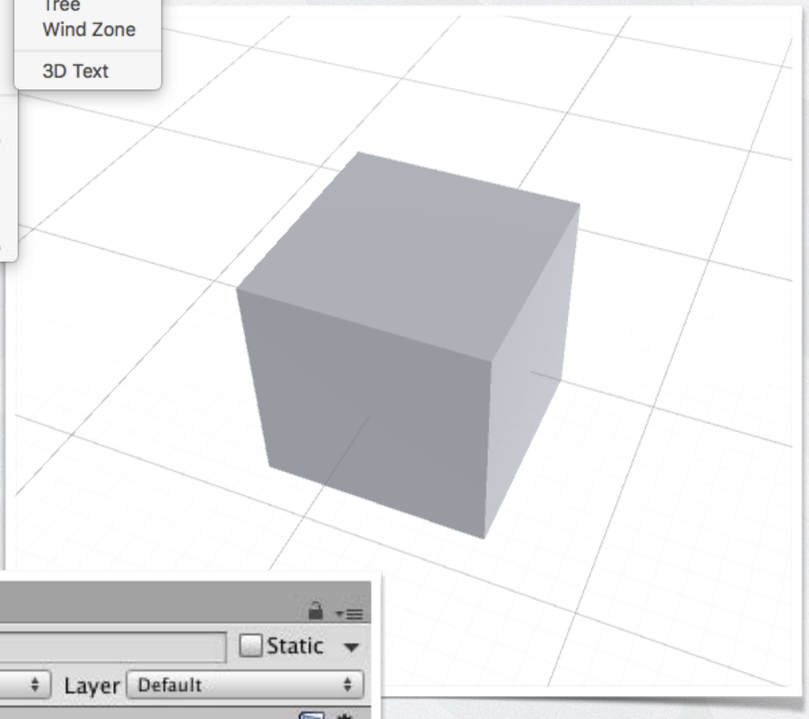
3D programs provide simple geometry, or **primitives**

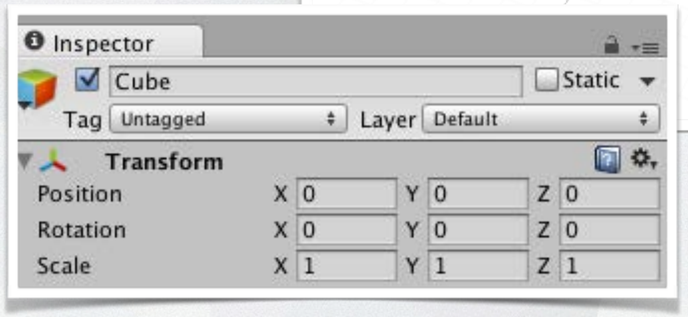
We can **transform** and **combine** these to make more complex objects

There are limitations, but they can bring you far in making a 3D environment

Unity has its own set of primitives to choose from through the GameObject menu



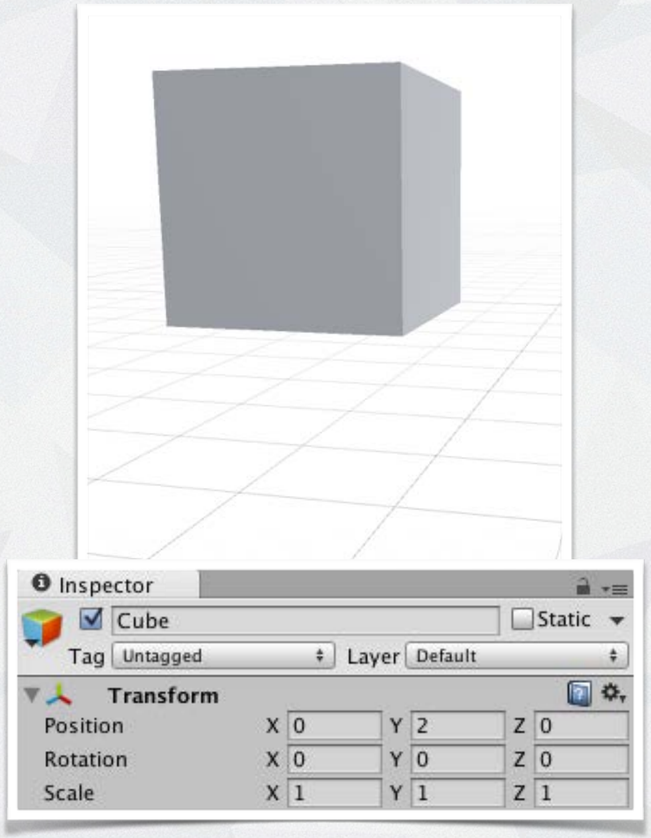




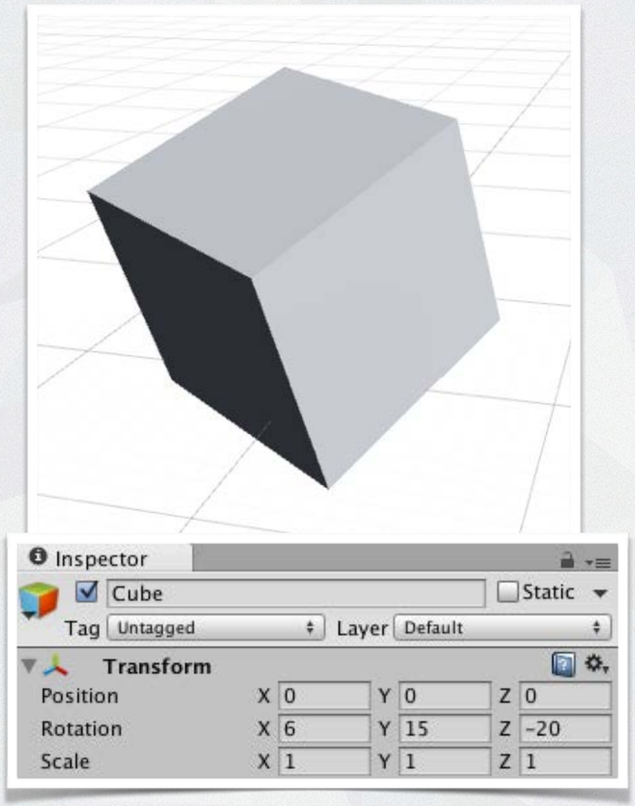
Created objects will appear in the scene, where they can be transformed

This is done through position, rotation, and scale

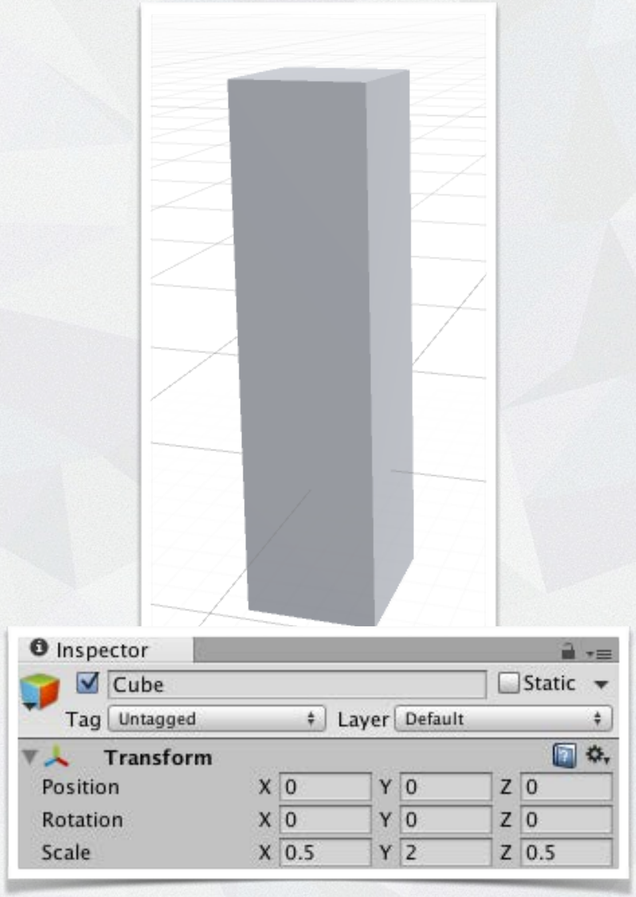
Transformation



Position



Rotation



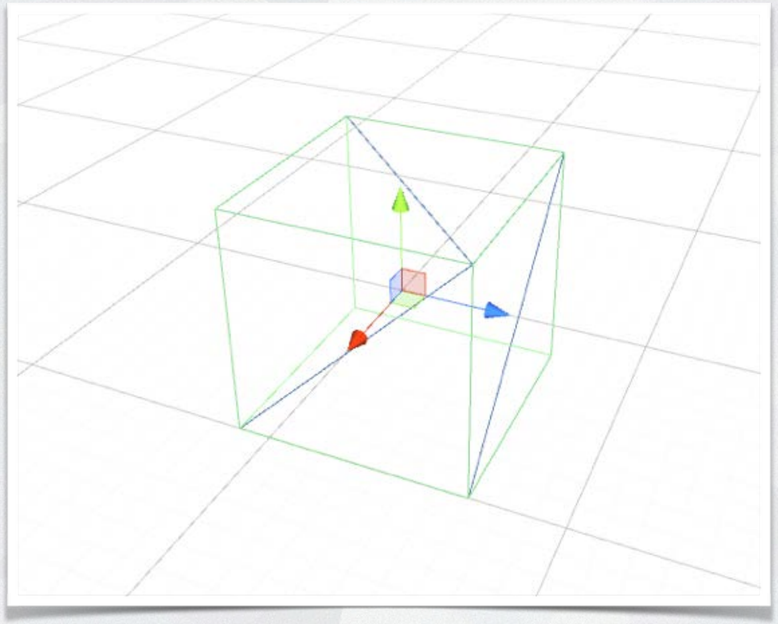
Scale

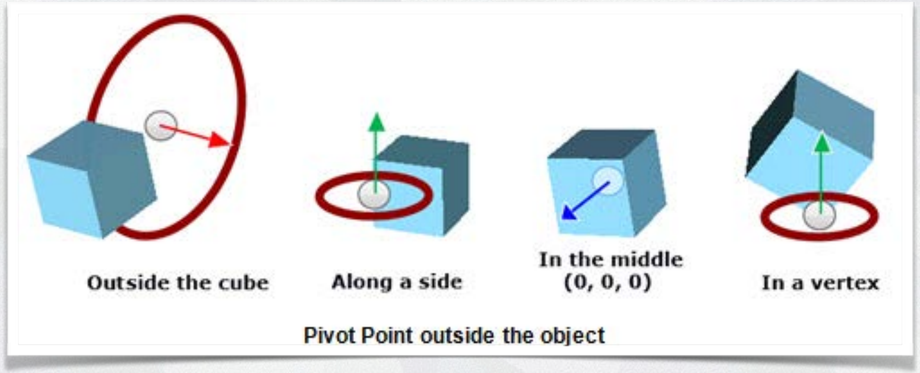
This cube does not occupy a single point in space

We transform it by its **local origin**

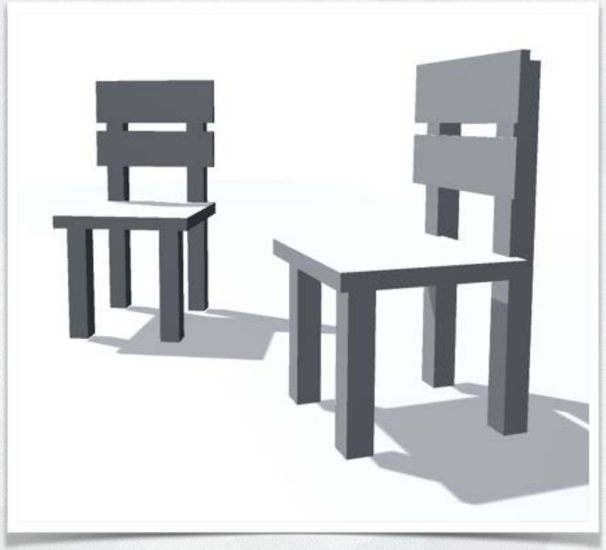
By default this is the **centre** of the object. With custom models, the local origin can be adjusted

Transformation happens relative to the local origin or **pivot point**





Combining Primitives



Many objects can be broken down into simple geometric shapes

Parenting / Grouping

Objects can be grouped together, also called **parenting**

One object can have multiple **children**

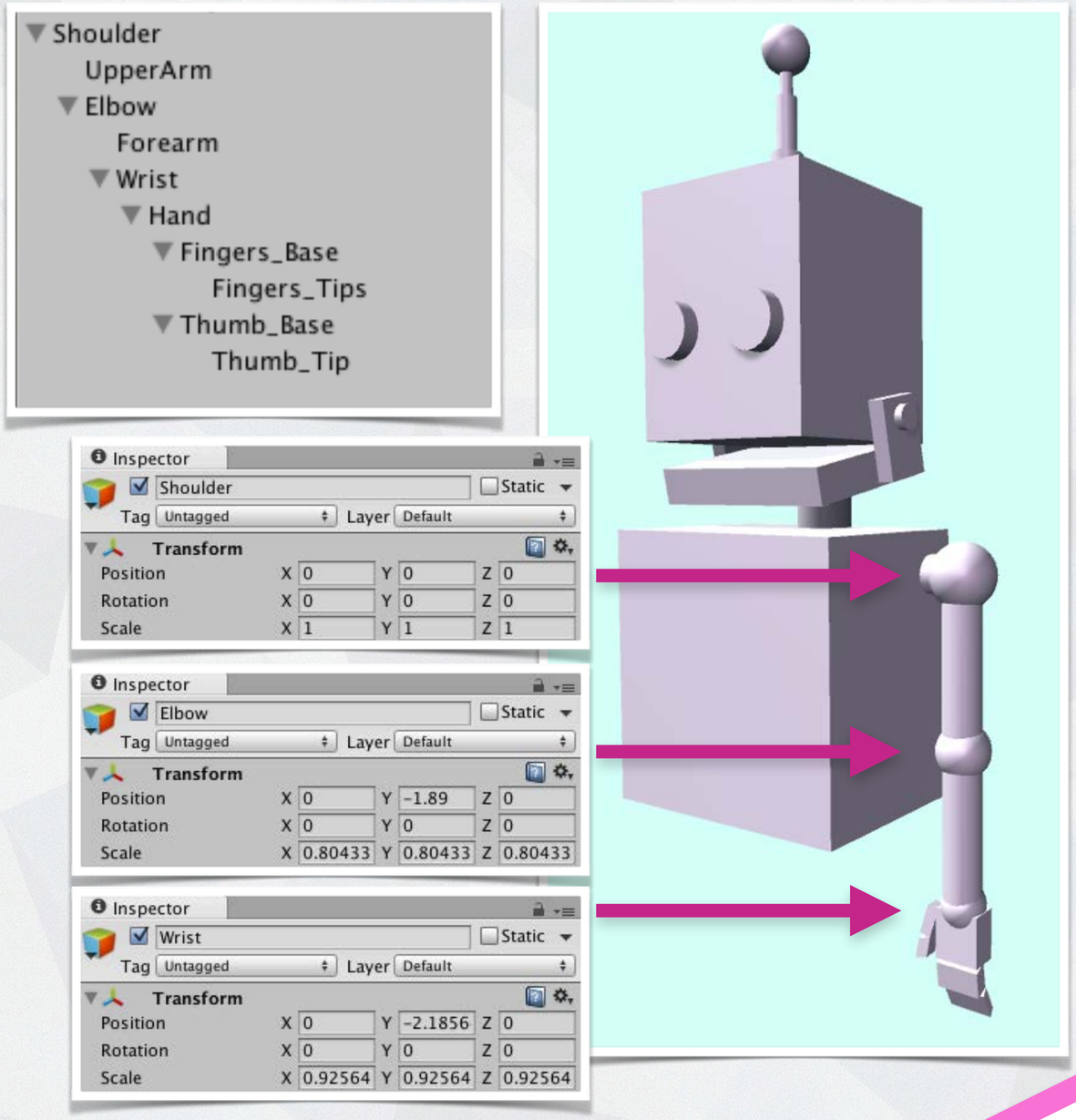
Children **transform together with the parent**, but **retain their own transformations** as well

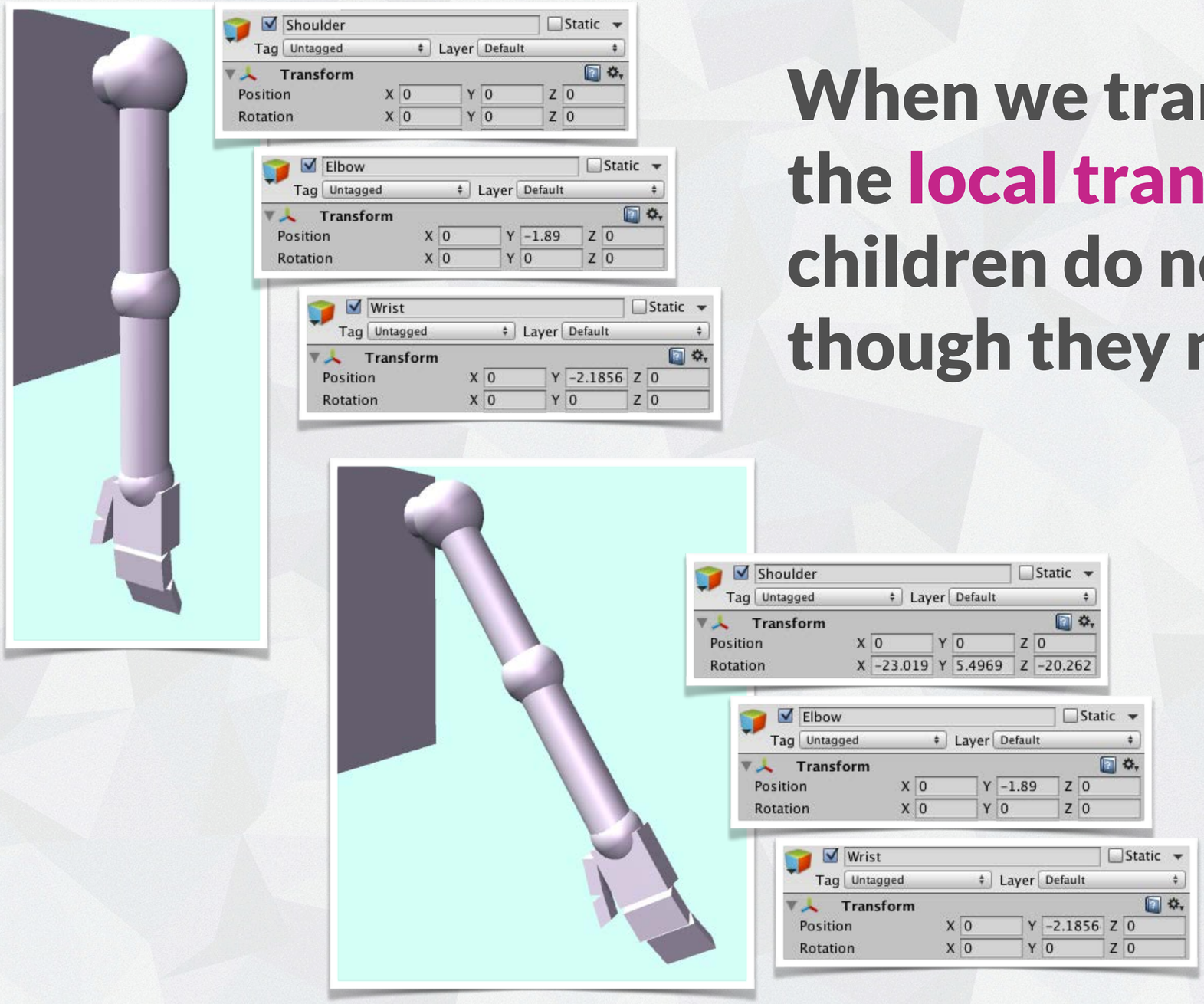
Local & Global

Transforms of the (root) parent are **global**, or relative to the **scene origin**

Transforms of children are **local**, or relative to their **parent's origin**

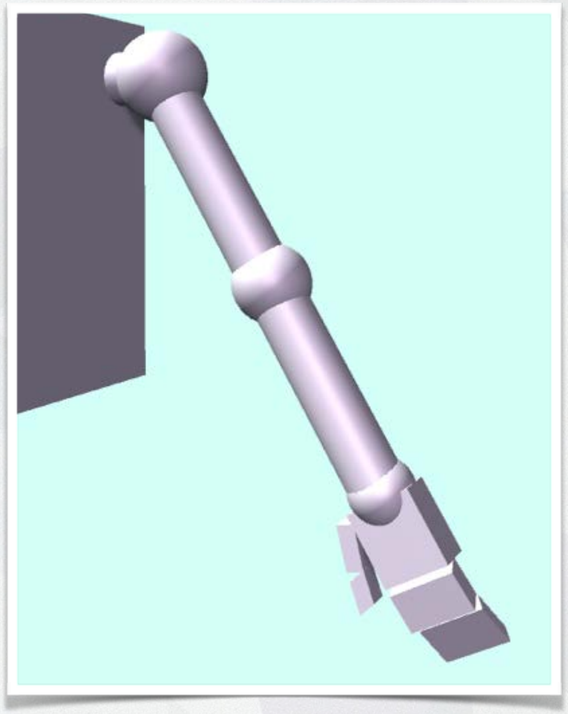
Robo is made out of primitives too!

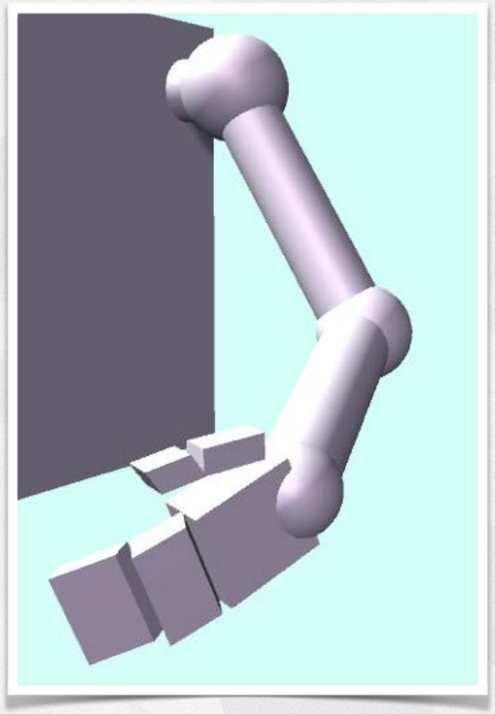




When we transform the parent, the local transforms of the children do not change, even though they move in global space

Note that the children have position transforms because they have been moved relative to their parents





Shoulder — Static
Tag: Untagged — Layer: Default
Transform
Position: X 0 Y 0 Z 0
Rotation: X -23.019 Y 5.4969 Z -20.262

Elbow — Static
Tag: Untagged — Layer: Default
Transform
Position: X 0 Y -1.89 Z 0
Rotation: X 0 Y 0 Z 0

Wrist — Static
Tag: Untagged — Layer: Default
Transform
Position: X 0 Y -2.1856 Z 0
Rotation: X 0 Y 0 Z 0

Shoulder — Static
Tag: Untagged — Layer: Default
Transform
Position: X 0 Y 0 Z 0
Rotation: X -31.014 Y -102.20 Z -4.2215

Elbow — Static
Tag: Untagged — Layer: Default
Transform
Position: X 0 Y -1.89 Z 0
Rotation: X 39.9701 Y -63.157 Z 19.3231

Wrist — Static
Tag: Untagged — Layer: Default
Transform
Position: X 0 Y -2.1856 Z 0
Rotation: X 19.1344 Y 26.466 Z 31.8737

Transforms “stack”, or rather, children **inherit** transforms from their parents

Each child is transformed by its parent. **Additionally**, it has its own transformation

This is important in animation; if a parent is animated, the child will inherit the animation as well

Putting it into practice!

Break Time!

But before, please give us some feedback:

http://tiny.cc/degplus1

After the break:

Unity Practice — Building a city with primitives

Cheat Sheet

View Controls

Alt + LMB = Rotate
Alt + MMB = Pan
Alt + RMB = Zoom

Transform Tools

Q = Select
W = Move
E = Rotate
R = Scale

Or the toolbar in the top

Create Primitives

Menu: GameObject > 3D Object > Cube, Sphere, Capsule, Cylinder, Plane

Create Empty Object

Menu: GameObject > Create Empty

Duplicate Object

Ctrl + D