Real-time 3d modelling
Real-time 3D modelling

Real-time basics

*Real-time* refers to applications which perform tasks without delay. For example, in real-time games, the system responds to input immediately. They are used for such tasks as navigation, in which the computer must react to a steady flow of new information without interruption. *Real time* can also refer to events simulated by a computer at the same speed that they would occur in real life. In graphics animation, for example, a real-time program would display objects moving across the screen at the same speed that they would actually move.

Real-time 3D modelling

Modelling is the key issue and the most critical part of creating assets for real-time applications. This is because real-time graphics are dependent on speed to enable for a presentable game play. *Prerendered* models need to look as realistic as possible; *real-time* needs to be as fast as possible. Hence, real-time must use only the most necessary elements namely the geometry, the transform and surface properties of the mesh to achieve fast speed.

Most real-time engines use triangular polygons while some systems use quads polygon. In general, the best match of polygon type and software system will ensure a good delivery. The best results seem to come from triangular polygons. Triangular polygons renders faster and always display as intended. The flip side however is that it requires more storage space than other polygon types. 3ds Max uses triangular polygon, which is why it is so widely supported among real-time game engines.
Modelling considerations for real-time applications

1. Low-poly modelling

Poly count refers to the total number of polygons that make up a given object. A high poly count directly affects the speed with which a computer can process a model. Hence for real-time applications whereby quick response and instant speed is required, it is desirable to keep the poly count as low as possible. This brings us to low-poly modelling.

Low-poly modelling is used much for creation of real-time rendered media. The term low poly applies to any project where there is a set poly limit. Real-time 3d graphics, 3d games and animation are some examples of low polygon use.

3ds max includes a Polygon Counter feature which can be used to keep count of the number of faces in the scene as a whole as well as selected object or objects.
2. Put the details in the Map, not in the Mesh

Every polygon added to a real-time mesh takes a certain amount of time to render. To be real-time effective, any detail should be effectively simulated to the texture map and not modelled. A great example of when to map instead of model is the muscle tone in a character. Nice rounded muscles structure is very polygon for real-time. Alternatively, muscles can be added to a texture map and proper simulated highlights and shadows can achieve similar realistic effect at no cost in frame rate and render time.

Texture mapping is like the shine to a gemstone. With texture maps, game assets can remain low count while maintaining high level of believability. (You will learn more about texturing in the coming weeks) For now, think of texturing as putting on a pattern or material to an object’s surface.

There are many different types of textures for different purposes. For real-time applications alone, there are tileable textures, decals, U-V Map, displacement maps and animated textures. We will go into texturing details soon.
3. Don’t build what is not needed

Be particularly sensitive to the visibility of your models, such as where your objects and environments will and will not be seen. Also, think about how the animation/game is going to show off your model. If your project has nothing but faraway shots, don’t focus so much on the essentials. Above all, spend the detail where it counts.

4. High-res for low-res modelling

An important process in creating real-time models is to create a fully detailed, high-face-count model first. Then construct the low-poly version over the high-poly. Use the orthographic renderings of the high-poly version as texture maps which you can tweak in a paint programme. This process will give you a highly detailed texture on a low-face-count model.

This process is helpful to maximize the realistic representation of the model at the same time keeping it low in poly and file size.
5. Levels of Details (LOD)

Practitioners of computer graphics have always struggled with the tradeoff between complexity and performance. Levels of detail (LOD) techniques are increasingly used by professional real-time developers to strike the balance between breathtaking virtual worlds and smooth, flowing animation.

This method employed for real-time and not for prerendered 3d images is necessary to achieve speed to create an enjoyable game. They are ‘stand-in’ objects used to represent the real object which is subject to the user’s viewpoint. Example: When the object is close to the player, the highest-resolution model is drawn. Meanwhile, when the object takes up a small portion of the screen, a lower polygon count model is swapped in. This technique requires the real-time artist to model objects at various levels of details. The simplifications of the initial object are produced and contain fewer and fewer details as it goes.

A model in high, medium and low levels of details (LODs)

Basic Texturing Considerations for Real-time

Apart from modelling, texturing also requires special attention in real-time. Textures not only present a faux finish for a 3d object, it is also the ability to create realism which mimics the real thing and at the same time save improves processing performance.
1. Seamless Tiling Textures

Tiling is a method of repeating one image over a large area. This method of texturing is often used to cover general areas such as wall and grounds. The advantage of tiling is that these images use less memory because the image is stored in the computer memory only once. However, tiled images suffer from drawbacks too.

- they tend to look too uniform and becomes unnatural
- the tiles have a tendency to show seams or patterns because the images are all lined up in a neat row

A successful tiled map is one which looks seamless and does not appear to be tiled.

An example of an unnatural tiled surface texture

2. Texture map and colour depth specifications

Real-time texturing works within a texture budget. This means the texture maps must be in a specific format, .bmp and have a specific colour depth, such as 8-bit or 16-bit and in a specific size, 256x256 pixels. This specification helps the 3d engine calculate and apply maps more quickly.

Some questions to ponder:

1. A good game is measured by game play or the 3d graphics?
2. Why are game models constantly pushing towards realism?

References:

Inside 3d Studio Max Vol 2: Advanced modelling and Materials, New riders Publishing

3d Graphics & Animation, Mark Giambruno

The Dark Side of Game Texturing, David Franson