3D Programming Concepts

Outline

- 3D Concepts
- Displaying 3D Models
- 3D Programming

3D Concepts

- 3D Model is a 3D simulation of an object.
  - Coordinate Systems
  - 3D Models
  - 3D Shapes

3D Concepts -- Coordinate Systems

- The coordinate systems for 3D and how they related to the 2D computer screen are important.
- The stages of the rendering pipeline, the steps involved in converting an abstract math model of an object into an effective on-screen picture.
- "Our world" is the XYZ-axis with height (Y-axis), width (X-axis) and depth (Z-axis).
- To simulate the third dimension (Z-axis) for "depth" in the screen, we use on-screen 3d simulations of the 3d object model using shading, shadows, textures and other visual characteristics.
In the right-handed coordinate system, Y and Z are oriented as in the left-handed system, but X is positive in the opposite direction, i.e. X is positive to the left, Y is positive up and Z is positive away from you.

If positive X-values are to the right, positive Y-values are up and positive Z-values are away from you, then you have the Left-handed Coordinated system.

The dimensional measure of a object is given by coordinates to mark each vertex (or corner) of the object.

We must decide:
- which variable will represent which dimension - height, width or depth
- in what order to list them
- where is the zero point?
- how do these specifications related to our object.

When thinking of the 3d object itself, each of the directions is represented by an axis.

When we consider a single object, in isolation, the 3d space it occupies is called object space.

The point in object where X, Y, Z are all 0 is the geometric center or an object.

The geometric center is usually inside the object.

If positive X-values are to the right, positive Y-values are up and positive Z-values are away from you, then you have the Left-handed Coordinated system.

Torque Game Engine (TGE) uses a right-handed system.

In the right-handed coordinated system, Y and Z are oriented as in the left-handed system, but X is positive in the opposite direction, i.e. X is positive to the left, Y is positive up and Z is positive away from you.

We can use the thumb, index finger and middle finger of our hands to figure out the “handedness” of the system.
3D Concepts -- Coordinate Systems

- For Torque, we also orient the system in a slightly different way:
  - the Z-axis is up-down,
  - the X-axis is *somewhat left-right*
  - the Y-axis is *somewhat near-far*
- The "somewhat" specifies that we are looking down on a map from above, with north at the top of the map. Right and left (positive and negative X) are East and West, respectively. And positive Y refers to North, negative Y refers to South. This defines our **World Space**.

3D Models

- Simple 3D models are called primitives: cylinder, cube, cone, sphere.
- Simplified model views often use an axis fog instead of a full axis rendition.
- We can define a model by giving its significant vertices (corners).
- We start with a simple 3d shape - the cube - in Fig 3.6.
- This cube is 2 units deep, 2 units high and 2 units wide (2 x 2 x 2). Fig 3.6 presents this in object space.
- Note the geometric center is offset to a position outside the cube.
- There is more information presented in Fig 3.6 than we really need and this can make it difficult to get a sense of the object itself.

3D Concepts -- Coordinate Systems

- **Object space**

3D Models

- We only need to state whether the object is in **object space** or **world space** and indicate the raw coordinates of each vertex. This is done in Fig 3.7, along with a small XYZ-axis notation in the corner. This is actually grey-scale-color coded:
  - dark yellow = light gray for X-axis
  - dark cyan = medium gray for Y-axis
  - dark magenta = dark gray for Z-axis
- It is common practice to place the XYZ-axis key at the geometric center of the model.
3D Models

- As an alternative, Fig 3.9 presents the cube with the geometric center within the object.
- Which of these diagrams more easily conveys the shape of the object?

3D Shapes

- We can specify more general models by connecting the vertices with lines, known as edges. If we connect three or more vertices with edges to create a closed figure, this is a polygon. The simplest polygon is a triangle.
- Usually triangles are used to define polygons.
- Triangles are effective for modeling complex 3D shapes, since any complex polygon can be decomposed into a collection of triangles, which we call a mesh.
- 3D accelerated graphics cards work best with triangle polygons.
- The polygons that make up a shape are called faces.
- Faces within a shape that are not viewable are called hidden faces.
- Faces on the backside of a shape are called back faces.

3D Shapes

- The area of the model is known as the surface.
- The polygonal surfaces are called facets, though this term is changing to be faces.
- Sometimes a surface can only be viewed from one side, so when you are looking at it from its "invisible side", it is called a hidden surface (or hidden face).
- A double-sided face can be viewed from either side.
- The edges of hidden surfaces are called hidden lines.
- With most models, there are faces on the back side of the model, facing away from the viewer, called backfaces.
Displaying 3D Models

- Transformation
  - Scaling
  - Rotation
  - Translation
  - Full transformation
- Rendering
  - Flat shading
  - Lambert shading
  - Gouraud shading
  - Phong shading
  - Fake Phong shading
  - Texture mapping
  - Shaders
  - Bump mapping
  - Environmental Mapping

Displaying 3D Models -- Transformation

- Scaling
- Rotating
- Translating

Displaying 3D Models -- Scaling

- Multiply measurements in each axis
- Apply to all of a shape's vertices
- Scale factors larger than 1 will make object bigger
- Scale factors smaller than 1 will make object smaller
- Scale factors of 0 or less are not applicable

Three-step process:

1. Convert to world space (transformation)
2. Convert to view space (3D rendering)
3. Convert to screen space (2D rendering)
Displaying 3D Models -- Scaling

- Each axis is rotated in turn
- Order of rotation matters
- Roll: rotate around longitudinal (Z) axis
- Pitch: rotate around lateral (X) axis
- Yaw: rotate around vertical (Y) axis

Common convention is roll-pitch-yaw ordering

Displaying 3D Models -- Translation

- Moves an object in space
- Vectors are applied to a shape’s vertices to translate it
- Shape is not changed
- Shape is not re-oriented
Displaying 3D Models -- Translation

![Translation Diagram](image1)

Figure 3.15 Translation.

Displaying 3D Models – Full Transformation

![Transformation Diagram](image2)

Figure 3.16 Fully transforming the cube.

Displaying 3D Models -- Rendering

- Rendering is a process of converting a 3D model of a shape to 2D picture
- A face is a set of contiguous adjacent triangles
- Shading
- Mapping
- Graphing
Displaying 3D Models -- Rendering

Flat shading:
- The simplest kind of shading is flat shading, which simply fills the polygons with one color. Flat shading is not acceptable for 3D gaming.
- Lambert Shading
  - Shading according to the light source
- Gouraud Shading
  - Gouraud shading is a complex process using algorithms to create a color gradient.
- Phong Shading
  - Phong shading is one of the more sophisticated techniques for doing this. It works like Gouraud shading but requires more computer horsepower and yields better results.
- Fake/fast Phong Shading
  - Allow fast frame rates

Displaying 3D Models -- Rendering

Gouraud shading:
- More advanced technique.
- A single shading color is determined per vertex.
- The color is then interpolated over the edges, and ultimately interpolated over the normals, effectively averaging the vertex colors fluently over the entire polygon.
- If used correctly, this technique makes objects look round.
- The technique can not be used in a convincing way if there are multiple lightsources in the scene.

Displaying 3D Models -- Rendering

Phong shading:
- One of the most realistic techniques for dynamic lighting.
- A texture is attached to every light source.
- This texture is then projected on every polygon, by using the normals of each polygon vertex as an index in the lightmap.
- This way, highlights can occur in the middle of a polygon.
- Also, the lightmap is fully configurable: It can be dithered, smooth, or very sharp in the centre. It is very hard however to have directed spotlights with this technique, or to have multiple lights on a single polygon.
Displaying 3D Models -- Rendering

- **Texture Mapping**
  - Texture mapping is the process of identifying points on objects you define with points in a texture map to achieve images that can include strong visual interest while using simpler geometry.
  - Think of the texture as an image applying to a graphical object to achieve a more realistic image.
  - When you render your objects they will be colored with the color values in the texture map.

- **Shaders**
  - There are two kinds of shaders:
    - **vertex shaders**: allow the manipulation of vertex data and
    - **pixel shaders**: allow the manipulation of pixel data.
  - The shader code is loaded into the graphics card memory and plugged directly into the graphics pipeline.
  - Shader code is in assembly language.
  - Microsoft have HLSL (High-Level Shading Language) for use with DirectX.
  - OpenGL has the GLSL (OpenGL Shading Language).
  - Hardware vendors also provide some high level languages.

- Bump Mapping is a technique to give an object more detail without adding more polygons.
- It is a way of simulating small bumps on the surface by changing the way the light effects are calculated.
- A bump will usually have one side that is bright from a light source while the other side is dark because it is on the shadow side.
- Bump Mapping modifies the light calculations to make this happen.
- Bump Mapping does not change the surface of the object it only changes the way light is reflected by the surface.
An environment map, sometimes referred to as a reflection map, is a method for applying environment reflections to a surface.

The technique of Mipmapping involves preprocessing a texture to create multiple copies, where each successive copy is one-half the size of the prior copy.

When a 3D card has to map a polygon with a texture of a 1:1 relationship of 1 texel (texture element) in the original texture map corresponds to 1 pixel on a polygon.

If the polygon you are displaying is scaled down to half size, then effectively the texture is displaying every other texel.

Normal-Mapping is a technique used to light a 3D model with a low polygon count as if it were a more detailed model.

Normal Mapping is a powerful technique used to create the appearance of detail in a 3D model while maintaining the model itself in a highly streamlined form.

This offers the advantage of significantly reducing the data size of the original model, accelerating the rendering process and generating higher-quality rendered result.

The technique originated from the need to increase the visual fidelity of 3D games while maintaining real-time performance and interaction.
Parallax Mapping (also, Offset Mapping or Virtual Displacement Mapping) is an enhancement of the bump mapping or normal mapping techniques applied to textures in 3D rendering applications.

Parallax mapping is implemented by displacing the texture coordinates at a point on the rendered polygon by a function of the view angle in tangent space (the angle relative to the surface normal) and

the value of the height map at that point.

At steeper view angles the texture coordinates are displaced more, and so give the illusion of depth due to parallax effects as the view changes.

A scene graph is a hierarchical approach to describing objects and their relationship to each other.
Displaying 3D Models -- Rendering

- Draw the star
- Save the current matrix
- Apply a rotation
- Draw Planet One
- Save the current matrix
- Apply a second rotation
- Draw Moon A
- Draw Moon B
- Reset the matrix
- Draw Planet two
- Save the current matrix
- Apply a rotation
- Draw Moon C
- Draw Moon D
- Reset the matrix
- Reset the matrix

Displaying 3D Models -- Rendering

- Draw the Star
- Save the current matrix
- Apply a rotation
- Save the current matrix
- Draw Planet 1
- Apply a rotation
- Draw Planet 2
- Save the current matrix
- Apply a rotation
- Draw Moon A
- Draw Moon B
- Reset the matrix
- Reset the matrix
- Save the current matrix
- Apply a rotation
- Save the current matrix
- Apply a rotation
- Draw Planet 2
- Save the current matrix
- Apply a rotation
- Draw Moon C
- Draw Moon D
- Reset the current matrix
- Reset the current matrix

Displaying 3D Models -- 3D Audio

- Add the sense of realism to a game
- Positioning sound in 3D using stereo
- Balancing left & right & volume control can approximate 3D position of sound
Displaying 3D Models -- 3D Programming

- Torque has built-in script functions to use
- Scene graph insertion is automatic
- Placing (spawning)
- Transforming
- Allows us to manually manipulate objects as well

Displaying 3D Models

- Lab 1: “Simple Direct Movement”
- Move an object from one location to another
- Input script statements directly via the console

Displaying 3D Models

- Lab 2: “Programmed Movement”
- Move an object from one place to another over time
- Use script files to move objects
- Real-time scene manipulation
Displaying 3D Models

- Lab 3: "Programmed Rotation"
- Make an object rotate about one or more axes
- Use script files to move objects
- Real-time scene manipulation

Displaying 3D Models

- Lab 4: "Programmed Scaling"
- Change an object's size smoothly over time
- Use script files to re-size objects
- Real-time scene manipulation

Displaying 3D Models

- Lab 5: "Programmed Animation"
- Perform complex transformations smoothly over time
- Utilize a loop in game script
- Practice ad hoc game animation using script
- Real-time scene manipulation

3D Programming

- Programmed Translation
- Programmed Rotation
- Programmed Scaling
- Programmed Animation
- 3D Audio

3D Audio
Summary

- 3D coordinates can be expressed as left-handed or right-handed
- Objects are designed in object space, placed in game world in world space
- Object space may use a different coordinate system than world space, and each uses a different center (zero reference)
- Changing an object's size, orientation, or location is called transforming the object
- Order of transformation matters
- The axis order of rotation matters

Converting the mathematical description of an object to a visual representation on a screen is called rendering

- Scene graphs contain and organize objects in a world in preparation for rendering and manipulation
- Torque provides many script commands for manipulating and moving 3D objects in real time
- 3D Audio effects can be handled in exactly the same way as regular 3D objects