

TRAINER PREPARATION GUIDE 3.3: DRAW OBJECTS

Lesson Objective 3.3:

Draw objects. *Topics:* using bitmaps, sprites, vector graphics, lighting, blending, text, textures, 3D geometry, parallax mapping, and different shaders; 2-D vs. 3-D; creating a sprite font.

Required materials to teach this lesson:

1. A workstation with Microsoft Office® installed
2. Microsoft Visual Studio 2010® and XNA® 4.0
3. Internet access
4. 98-374-ENU-3.3-LP
5. 98-374-ENU-3.3-IC
6. 98-374-ENU-3.3-IC_Key
7. 98-374-ENU-3.3-PC

Preparation Tasks

Technical preparation activities:

1. Visual Studio 2010 and XNA Game Studio 4.0 should be installed on all computers.
2. **Vocabulary:**

Draw(): a method in XNA where the content is drawn to the screen.

Geometry shader: used to add new vertices based on what is there or on the current design.

LoadContent(): a method in XNA where the content is all loaded into the game.

parallax mapping: enhanced mapping techniques used to make the images seem more real and have more depth.

Pixel/Fragment shader: a tool that calculates the color of the pixels. The main difference between the Vertex shader and the Pixel/Fragment shader is that the Vertex shader is based on the geometry of the image and the Pixel/Fragment shader is used to change the color of the pixels.

sprite: 2-D bitmaps that are drawn directly to a render target without using the pipeline for transformations, lighting, or effects.

SpriteBatch(): used to write text to the screen.

SpriteFont: used to take a vector-based font and turn it into a bitmap, which allows it to be drawn more quickly on the screen.

Texture2D: a 2-D graphic; this is a special data type that holds graphical data and can be drawn on the screen.

Vertex shader: a shader that manipulates the vertex based on mathematical functions.

Vector2: a 2-D vector; it has an *X* number and a *Y* number that can be set independently of one another. This is to place your two-dimensional graphic on the screen.

Vector3: a 3-D vector; it has an X, Y, and Z number that can be set independently of each other. This is to position your 3-D object on the screen.

3. **Additional readings and resources:**

MSDN®:

Drawing Text with Spritefont: <http://msdn.microsoft.com/en-us/library/bb447673.aspx>

2D Game Development Tutorial: http://create.msdn.com/en-US/education/tutorial/2dgame/getting_started

Displaying a 3D Model on the Screen: <http://msdn.microsoft.com/en-us/library/bb197293.aspx>

Other resources (books, e-reference):

Game Creation with XNA/3D Development/Shaders and Effects:
http://en.wikibooks.org/wiki/Game_Creation_with_XNA/3D_Development/Shaders_and_Effects

2D XNA Tutorials: <http://rbwhitaker.wikidot.com/2d-tutorials>

3D XNA Tutorials: Using 3D Models: <http://rbwhitaker.wikidot.com/using-3d-models>

Instructor computer setup:

1. Visual Studio 2010 and XNA Game Studio 4.0.

Instructional preparation activities:

1. Review all Lesson 3.3 documents.
2. Make copies of student documents available as needed.
3. Review and complete the tutorial to be sure you have the computers set up correctly and to be sure you can respond to any student questions as students go through the tutorial.

Lesson Sequence (50 minutes)

Activating prior knowledge/lesson staging (5 minutes):

Direct students to answer each question in the "Guiding questions" section of the In-class Activity document or their personal class notes.

Guiding questions:

1. **How do you add 2-D and 3-D graphics to the screen with XNA?** 2-D graphics are added to the screen by first adding the content resource to your solution. The graphics must be loaded in the *LoadContent()* method and finally, the graphics must be drawn to the screen in the *Draw()* method. 3-D graphics are added to the content of your solution by using the variable model.
2. **List the three types of shaders and their functions.**

A Vertex shader manipulates the vertex based on mathematical functions.

A Geometry shader is used to add new vertices based on what is there or on the current design.

A Pixel/Fragment shader manipulates the color of the pixels. The main difference between the Vertex shader and the Pixel/Fragment shader is that the Vertex shader is based on the geometry of the image, and the Pixel/Fragment shader is used to change the color of the pixels.
3. **What is parallax mapping?** Enhanced mapping techniques that are used to make images seem more real and have more depth.

Lesson activity (40 minutes):

1. Teacher instruction (15 minutes):

Use the included Microsoft PowerPoint® presentation to review the topics.

2. In-class activity (20 minutes):

Students are to complete the In-class Activity document 98-374-ENU-3.3-IC.

3. Post-class activity (5 minutes):

Provide instruction for the post-class activity as needed. Establish a completion date.

Lesson review (5 minutes):

1. Discuss the guiding questions.
2. Instruct students to write and submit any questions they have or any topics about which they would like more assistance.
3. After class, look through the student responses and follow up with any student requiring additional help.