

## KEY IN-CLASS STUDENT ACTIVITY 1.7A: WORK WITH XNA

### Lesson Objective 1.7:

Work with XNA®. *Topics:* understanding the architecture of an XNA game; using built-in XNA tools.

### Resources, software, and additional files needed for this lesson:

1. None

### Guiding questions:

1. **How is an XNA game structured?** There are five methods provided in an XNA game skeleton: *Initialize*, *LoadContent*, *UnloadContent*, *Update*, and *Draw*. They are intended to be overridden. A solution in Microsoft Visual Studio® contains a project which contains properties, references, content references, a game icon, a game thumbnail image, a *Game1* class, a *Program* class, and a Content project folder.
2. **How are built-in XNA tools used?** XACT is used to create, organize, and play audio. The XACT Auditioning Utility is used to test XACT-created audio projects. The XNA Game Studio Device Center is used to communicate with certain devices, like an Xbox 360®. The XNA Framework Remote Performance Monitor provides diagnostics and instrumentation for the XNA framework. The XNA Game Studio command prompt allows users to execute commands using the command line.

### Student activity:

#### Directions to the student:

Read the following scenario and respond to the situation presented. Verify your answers with the instructor. Request assistance from the instructor as needed. Share your ideas with the class.

#### Scenario:

Martha is preparing to work with a new employee who has extensive experience in development, but not in game development. Her supervisor has asked her to help the new employee gain skills and knowledge with XNA Game Studio. She has decided that the first step should be a discussion and demonstration of the five methods that make up an XNA game template.

#### Content:

List the names of the five methods that make up the template for an XNA game. Describe what each method does and how many times per game each one is called.

#### Answer:

*Initialize:* The method that allows the game to perform any initialization it needs to before running. This is where any required services and load any non-graphic-related content is queried. It is called once per game.

*LoadContent:* The method through which all the game's content (images, fonts, videos, audio, and so on) are loaded. It is called once per game.

*UnloadContent:* The method through which all the game's content is unloaded. It is called once per game.

*Update:* The method that runs the game logic such as updating the world, checking for collisions, gathering input, and playing audio. It is called 60 times per second.

*Draw:* The method that is called when the game should draw itself. It is called 60 times per second.