

## POST-CLASS LEARNING AND PRACTICE 2.5: UNDERSTAND DIFFERENT GAME PLATFORMS

### Lesson Objective 2.5:

Understand the different game platforms. *Topics:* console, PC, mobile; compare memory management.

### Additional learning resources:

#### MSDN®:

**Designing Applications for Windows Phone 7®, Chapter 2:** <http://msdn.microsoft.com/en-us/library/gg490770.aspx>

**Game State Management:** [http://create.msdn.com/en-US/education/catalog/sample/game\\_state\\_management](http://create.msdn.com/en-US/education/catalog/sample/game_state_management)

#### Other resources (books, e-reference):

**Create a Timeline Using Microsoft Excel® 2007:** <http://www.microsoft.com/education/en-us/teachers/how-to/Pages/timeline.aspx>

**History of Computer Games:**  
<http://www.computernostalgia.net/articles/HistoryofComputerGames.htm>

**History of Xbox®, Parts 1 and 2:** <http://www.youtube.com/watch?v=RYyIqNIBbQI>

**Windows Phone®:** [http://en.wikipedia.org/wiki/Windows\\_Phone#Games](http://en.wikipedia.org/wiki/Windows_Phone#Games)

**Xbox 360® History:** <http://en.wikipedia.org/wiki/Xbox>

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

1. A workstation with Windows 7®, Windows Vista®, or Windows XP®
2. Microsoft Visual Studio® or Microsoft Visual C# Express® (2008 or 2010)
3. XNA® Game Studio 2.0, 3.0, 3.1, or 4.0
4. Internet connectivity
5. CLR Profiler: <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=13382>

### Student Activity:

#### Directions to the student:

Read the following scenario. Follow the directions provided using an XNA game you have previously created. Upon completion of the instructions, answer the reflection questions.

#### Scenario:

Adventure Works is developing new video games for Windows Phone 7®. They are making amazing progress, with one exception. Most of the teams working on the new games are running into problems related to memory management. While everyone understands the demands that exciting video games place on the memory capacity of mobile devices, they are still having problems fitting their projects into this demanding environment.

**Content:**

There are several tools available for analyzing the .NET garbage collection heap. One tool available from Microsoft is the CLR Profiler, which is free and recommended for profiling memory. It is available for download at <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=13382>.

**To launch the profiler:**

1. Download the CLR Profiler, and then extract the files.
2. Navigate to CLRProfiler\Binaries\x86 (or x64), and then run CLRProfiler.exe.
3. Click Start Application, and then select your application or game.

The profiler generates a graph that displays information about all objects in the garbage collection heap, as well as a hierarchy of references related to these structures.

The CLR Profiler does not run on Windows Phone 7, so you must create a Windows version of your game. By using Game Studio tools, this task is easy.

**Questions:**

1. Summarize the steps you completed to finish the tutorial.
2. Did you have any problems? If so, how did you solve them?
3. What did you learn by completing the tutorial?
4. How did you learn about the memory demands of your game?
5. What can you do that will improve the memory demands for this game?

## **KEY 2.5: UNDERSTAND DIFFERENT GAME PLATFORMS**

### **Content:**

1. Answers will vary.
2. Answers will vary.
3. Answers will vary.
4. Answers will vary.
5. Answers will vary.