

98-374

MTA: Gaming Development Fundamentals

About This Exam

The Microsoft Technology Associate (MTA) is a new Microsoft Certification program that validates the *foundational knowledge* needed to begin building a career using Microsoft technologies.

Successful candidates earn MTA certificates as well as access to benefits on the Microsoft Certification member site.

This program:

- is targeted primarily at students who attend high schools and two-year colleges.
- provides an appropriate entry point to a future career in technology.
- assumes some hands-on experience or training but does not assume on-the-job experience.

This exam is designed to provide candidates with an assessment of their knowledge of fundamental gaming development concepts. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams.

Audience Profile

Candidates for this exam are seeking to prove core gaming development skills. Before taking this exam, candidates should have solid foundational knowledge of game design, hardware, graphics, and animation. It is recommended that candidates be familiar with the concepts and have some hands-on experience with the technologies described here either by taking relevant training courses or working with tutorials and samples available on MSDN and in Microsoft Visual Studio.

Proprietary and Confidential - © 2011 Microsoft Corporation

The information contained in this document is proprietary and is not to be distributed or disclosed unless explicitly authorized by Microsoft Corporation.

Objective Domain

1. Understand Game Design

1.1. Differentiate among game types.

This objective may include but is not limited to: console, Xbox, MMORPG, mobile games, PC games

1.2. Differentiate among game genres.

This objective may include but is not limited to: fantasy, sports, role playing, card, board, First Person Shooter

1.3. Understand player motivation.

This objective may include but is not limited to: quests, tasks, activities, how to win, game goals

1.4. Design the user interface.

This objective may include but is not limited to: UI layout and concepts, asset management, game state, gamer services

1.5. Understand components.

This objective may include but is not limited to: differentiate between tool creation and game programming, understand artificial intelligence (AI)

1.6. Capture user data.

This objective may include but is not limited to: save and restore user data, save and restore game state, handle input states, store data, manage game state

1.7. Work with XNA.

This objective may include but is not limited to: understanding the architecture of an XNA game; using built-in XNA tools; work with XNA hierarchy (initialization, update loop, drawing)

2. Understand Hardware

2.1. Choose an input device.

This objective may include but is not limited to: mouse, keyboard, Kinect, console, mobile

2.2. Choose an output device.

Proprietary and Confidential - © 2011 Microsoft Corporation

The information contained in this document is proprietary and is not to be distributed or disclosed unless explicitly authorized by Microsoft Corporation.

This objective may include but is not limited to: screen, television, hand-held devices, sound (local speakers, surround sound systems)

2.3. Work with the network.

This objective may include but is not limited to: set up Web services, TCP, UDP, basic management; plan for areas without access to Internet

2.4. Manage game performance.

This objective may include but is not limited to: CPU vs. GPU, reach vs. HiDef, graphics networking performance

2.5. Understand the different game platforms.

This objective may include but is not limited to: console, PC, mobile; compare memory management

3. Understand Graphics

3.1. Understand rendering engines.

This objective may include but is not limited to: DirectX, video and audio compression, display initialization, resolution (full screen, Vsync, and windowed)

3.2. Plan for game state.

This objective may include but is not limited to: scene hierarchy engine, gametime to handle frame rate variations, understanding games' main loop (input/update/render), graphics pipeline; understanding the flow of a game, loading, menus, save-load, configuring options (video, audio, keyboard)

3.3. Draw objects.

This objective may include but is not limited to: 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

4. Understand Animation

4.1. Animate basic characters.

This objective may include but is not limited to: movement, lighting, projections, frames per second (FPS), shaders, apply filters to textures, sprite animation, generate objects from user indexed primitives, matrices, understanding keyframes, motion between keyframes

4.2. Transform objects.

Proprietary and Confidential - © 2011 Microsoft Corporation

The information contained in this document is proprietary and is not to be distributed or disclosed unless explicitly authorized by Microsoft Corporation.

This objective may include but is not limited to: forming, deforming, moving, point distances, planes, interpolation; frames per second (FPS); translation, scale, rotation

4.3. Work with collisions.

This objective may include but is not limited to: per pixel and rectangle collisions, collision detection, collision response, fundamentals of physics simulation