

Game Development with XNA: Semester 1

A Teacher Roadmap

© 2011 Microsoft Corporation. All Rights Reserved. This curriculum material ("Material") is provided as part of *Game Development with XNA: Semester 1*. You may not resell this Material without first receiving permission from Microsoft. If you have questions about reuse of this Material, please write to us at innovativeteachers@microsoft.com with details about your request.

Microsoft, XNA, Visual C#, Visual Studio, Win32, Windows, Windows Phone, Kinect, and MSDN are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.

Microsoft[®]

Welcome to the exciting world of game development using Microsoft XNA Game Studio!

XNA game development appeals to students on many levels; it stimulates creativity, employs the latest tools and gadgets, delivers fun, and it provides the means for students to express their personal values and the power to make a difference. This curriculum was designed to inspire students to learn advanced programming skills, design exciting games, and create simulations that will help solve the world's toughest problems such as those presented in the Microsoft Imagine Cup. During the complete year-long course students will develop projects with C# and XNA for the Xbox 360, Windows Phone and Kinect.

This course is organized into two semesters of clustered, topic-centered lessons. A holistic approach builds skills and knowledge to enable students to analyze and design strategies to address real-world scenarios and projects. *Game Development with XNA: Semester 1* is the first half of a year-long curriculum created specifically as a capstone course for students who have previously studied computer science during one or two years of introductory and/or advanced courses. Students will build upon their foundational programming knowledge and skills by designing and implementing games and simulations that utilize input and output, involve complex logic, and apply object oriented programming (OOP), advanced algorithms, and data structures.

CONTENTS

Curriculum Materials

Teacher Role

Transitioning from XNA Jump Start

Software and Hardware

Technical Notes

Curriculum Supplements

About the Author

Timeline

Scope and Sequence

CURRICULUM MATERIALS

Lessons

Each lesson includes a multi-day Plan that focuses on a unique game development topic.

- Each “day” is defined as a 45-60 minute class period.
- Each Plan includes an overview, mastery objectives, a lesson sequence, and daily, step-by-step lesson guides.
- The daily guide typically includes some or all of these learning resources: an activator designed to elicit prior or related knowledge; lecture notes; a class activity; a hands-on lab programming project; a video tutorial; and a quiz/test.
- All of the curriculum documents are provided electronically to enable you to adapt them to your needs. It is suggested that you provide electronic copies of labs and activity sheets to students to avoid physical duplication costs.

Assign activities and lab pseudo-code assignments as out-of-class projects to increase the pace of the lesson; suggested lab extension activities can be used to slow the pace, provide more in-depth learning, and challenge advanced students. Curriculum resources are available at [Faculty Connection](#) or [Microsoft US Teacher](#).

Videos

The tutorial videos will guide students through many of the technical aspects of the lab projects. They enable students to progress at their own rate and are designed to provide technical support for teachers who may be relatively new to XNA development. Videos can be paused at selected points to provide time for students to implement the code in the tutorial. It is recommended that, if possible, students use headphones.

Textbook

The textbook, *Microsoft XNA Game Studio 3.0 (or 4.0): Learn Programming Now!* by Rob Miles, is the primary text resource for this curriculum and provides excellent explanations of complex game development concepts with humor and clarity. The book chapters correspond to the tutorial videos and provide additional explanation. You may choose to assign readings before or after the video tutorials. See more about the textbook below.

Microsoft XNA Game Studio 3.0 (or 4.0): Learn Programming Now! is currently available in two versions. References to page numbers in assignments and activities within the lesson plans are indicated according to the version of the book—v3 or v4. **Either version of the book or XNA can be used with the Semester 1 course materials.**

VERSION 3: *Microsoft XNA Game Studio 3.0: Learn Programming Now!* by Rob Miles.

- The e-book version, [Introduction to Programming Through Game Development Using Microsoft XNA Game Studio](#), is available free of charge.
- It can be used with the *Game Development with XNA: Semester 1* curriculum and XNA framework version 3.0 or 4.0.

VERSION 4: *Microsoft XNA Game Studio 4.0: Learn Programming Now!* by Rob Miles.

- Available for purchase in paperback or e-book through major online retailers. Up to two examination/desk copies may be available to your school. Visit [O'Reilly Academic Solutions](#) for details.
- Chapters 1—16 contain the same content as the v3 book and can be used with XNA version 3.0 or 4.0.
- Chapters 17—19 require XNA 4.0 (includes Windows Phone development which is included in the Game Development with XNA: Semester 2 curriculum)

TEACHER ROLE

The teacher's role is to facilitate the learning process by reviewing key programming concepts, managing the pace of the lessons, staging the learning environment, and measuring achievement by using the assessments, keys, and project exemplars found in the curriculum documents.

Pair Programming is a cooperative learning paradigm that can be used during labs and projects and is a particularly useful learning strategy for the computer science classroom. However it is suggested that students should complete the tutorials independently. For more details on pair programming, refer to the *Pair Programming* and *Clock Buddies* documents in the curriculum Appendix folder.

TRANSITIONING FROM XNA JUMP START

The *XNA 0.5: Jump Start* course was designed as a five-week, mini-course introduction to XNA fundamentals to be used at the end of and within the normal time allocation of another programming course. *XNA 0.5: Jump Start* course is not a prerequisite for the *Game Development with XNA: Semester 1* course. *Game Development with XNA: Semester 1* and *Semester 2* provide a deeper learning experience of the fundamentals in **XNA 0.5: Jump Start**, and also include:

- More XNA: input through use of the triggers and joysticks, collision, text input, and Windows Phone-specific code
- Game design theory: elements of game design, progression, plot, play-testing
- Not just games, but also simulations related to other school subjects
- More projects for beginner and advanced students
- Inheritance and polymorphism
- Advanced algorithms (searching and sorting) and data structures (lists, stacks, and queues)
- Development for Windows Phone
- An on-ramp into the Microsoft Imagine Cup competition

SOFTWARE AND HARDWARE

For additional practice opportunities, students can download and install the free [Microsoft Visual Studio C# 2010 Express](#) and [XNA Game Studio 4.0](#) on their personal PC.

If you are using Microsoft Vista or Microsoft Windows 7, download [Windows Phone 7 Developer Tools](#) which includes Microsoft Visual Studio C# 2010 Express and XNA Game Studio 4.0,

as well as Expression Blend for Windows Phone and Silverlight (for the Windows Phone 7 system). The XNA framework will install on top of Visual Studio creating XNA Game Studio. For additional technical details refer to the text book, *Microsoft XNA Game Studio 3.0 (or 4.0): Learn Programming Now!*, or the resources found at [App Hub](#).

TECHNICAL NOTES

XNA 3.1 can be used for the *Semester 1* course; *Semester 2* requires the Windows Phone Developer Tools and XNA 4.0. While there are many code differences between the versions, most of the code used in this curriculum will be similar to that found in either version. One example of a difference is the use of the *SoundEffectInstance* class. All of the tutorial videos are recorded in Visual Studio 2010 Express and XNA Game Studio 4.0.

Instructions for both keyboard and gamepad input are included in this curriculum. Keyboard input can be substituted for gamepad input for videos and labs in most cases. One exception: the gamepad must be used when analog controls are required, such as the gamepad triggers. Either wired or wireless Xbox 360 controllers can be used; wireless controllers require a separate wireless gaming receiver adapter. Controllers are not required for completing the lessons, but they do provide additional opportunities for students to explore input and tactile feedback such as vibration. It is recommended that there be at least one controller for every two students.

See [What's New in XNA Game Studio 4.0](#) for more information on new features in XNA 4.0, software and hardware installation, and configuring a firewall for XNA network traffic.

CURRICULUM SUPPLEMENTS

Interrobang



Interrobang is an online social networking game that encourages students' natural inclination to help others. Players compete against other students around the world to make a real difference in their communities by completing missions that employ 21st Century skills like problem solving, creative thinking, collaboration, and impacting society.

Pex for Fun



Pex for Fun is an online instructional tool that enables students to solve code problems within a browser. Student code is executed and analyzed in the cloud for immediate feedback. Pex for Fun helps students to better understand and visualize their code solutions by using dynamic symbolic execution to thoroughly explore feasible execution paths. The real fun starts with Coding Duels in which students write code that implements problem specifications and track their progress against the efforts of other Pex for Fun users. Pex for Fun is used in the curriculum as a supplementary tool for practice, mastery, and analysis of algorithms.

Pex for Fun connects teachers, curriculum authors, and students in a unique social experience, tracking and streaming progress updates in real time.

The Pex for Fun assignments appear in the first-day guide of a lesson Plan document, and can be used as an introduction to the lesson, an in-class activity, or an out-of-class assignment.

ABOUT THE AUTHOR



Pat Yongpradit

Springbook High School, Silver Spring, Maryland

“A man can do nothing better than to eat and drink and find satisfaction in his work.”
— Ecclesiastes 2:24

Who actually gets to live out their dream job? Pat Yongpradit does... he teaches high school computer science. The logic and detail inherent in computer science appeal to his meticulous side, while its flexibility and design opportunities appeal to his creative side. In addition to teaching, Pat is a curriculum consultant and has worked with Microsoft, USA Today Education, and the National Academy Foundation. In 2010 he was recognized as a Microsoft Worldwide Innovative Educator for his work with game programming and women in technology. Pat's current projects include helping students create video games and mobile applications for social causes. His goal as a teacher is to develop students that imagine and create technology to improve our world.

“The man who finds a job he loves will never work a day in his life.” — Confucius


Who looks forward to Mondays? Pat does.

TIMELINE: A Quick Overview



Semester Topics	Days
PART 1	
Lesson 0: Kodu Prelude (optional): Object Oriented Programming Concepts with Kodu	2
Lesson 1: Introduction to Visual Studio C# and XNA	4
Lesson 2: The Game Loop—Variables, Types, Classes, and Objects in XNA	5
Lesson 3: Input	8
Lesson 4: Displaying Images	11
Lesson 5: Writing Text	4
PART 2	
Lesson 6: Creating a Multi-Player Game	7
Lesson 7: Playing Sounds	7
Lesson 8: Creating a Timer	10
Lesson 9: Game Design	14
Lesson 10: Reading Text Input (optional)	4
Lesson 11: OOP and Non-Violent Conflict Game	11
Lesson 12: Two-Dimensional Arrays	13

SCOPE AND SEQUENCE






denotes an assessment.  denotes a video.







0. Object Oriented Programming Concepts with **Kodu**

- a. What is Kodu?
- b. Object Oriented Design Concepts
- c. Video: Kodu Tutorial 
- d. Lab: Kodu 




1. Introduction to Visual Studio C# and XNA

- a. Course Introduction
- b. Clock Buddies
- c. C# and XNA
- d. Creating a first program
- e. Transitioning from Java to C#
- f. Worksheet: Transitioning to C#
- g. Video: Intro to Visual Studio, C#, and XNA 
- h. Chapter 1 Reading Questions 
- i. Quiz 

2. The Game Loop - Variables, Types, Classes, and Objects in XNA

- a. The Color structure
- b. Update, Draw, and the Game Loop
- c. Troubleshooting
- d. Video: MoodLight Tutorial 
- e. Classes and Objects within XNA
- f. Conditionals: Updating colors
- g. Chapter 2 Reading Questions 
- h. Worksheet: Game Loop 
- i. Quiz 
- j. Lab: Techno MoodLight 
- k. Test 

3. Input

- a. Random Moodlight
- b. GamePad and GamePadState
- c. Keyboard and Keyboard State
- d. Video: Manual MoodLight Tutorial 
- e. Chapter 3 Reading Questions 
- f. Worksheet: Control Layouts 

- g. Lab: Sumo Gamepad Fighter 📄
- h. Vibration
- i. Lab: Color Nerve 📄
- j. Worksheet: Random 📄
- k. Lab: Color Match 📄
- l. Test 📄

4. Displaying Images

- a. Character Development
- b. Video: Jake Display Tutorial 📺
- c. Chapter 4 Reading Questions 📄
- d. Lab: Moving with Speed 📄
- e. Adding resources and loading images
- f. Texture2D and Rectangle
- g. Drawing with SpriteBatch
- h. Moving an image with user input
- i. Video: Collision Tutorial 📺
- j. Video: Mouse Input Tutorial 📺
- k. Lab: Health Bar 📄
- l. Lab: Title Screen 📄
- m. Test 📄
- n. Reading Questions 📄

5. Writing Text

- a. XML files and the SpriteFont class
- b. Video: 3D Big Clock Tutorial 📺
- c. Chapter 5 Reading Questions 📄
- d. Lab: Pop Art MoodLight 📄
- e. Test 📄

6. Creating a Multi-Player Game

- a. Level and edge detection
- b. Video: Button Bash Tutorial 📺
- c. Chapter 6 Reading Questions 📄 📺
- d. Quiz 📄
- e. Lab: BeadforLife 📄

7. Playing Sounds

- a. SoundEffect and .wav files
- b. Song and .mp3 and .wma files
- c. Video: Drum Pad Tutorial 📺



Figure 1. Health Bar Lab



Figure 2. Pop Art MoodLight Lab



Figure 3. BeadforLife Game

- d. Chapter 7 Reading Questions 📄
- e. Exceptions
- f. Sound Modulation
- g. Lab: Drum Pad Modulation 📄
- h. Lab: Sound Design 📄
- i. Video: Drum Pad Tutorial 🎥
- j. Quiz 📄

8. Creating a Timer

- a. Timing events
- b. Creating, using, and scanning an array
- c. Video: Reaction Timer Tutorials Part 1 and 2 🎥
- d. Chapter 8 Reading Questions 📄
- e. Lab: Loading Screen 📄
- f. Lab: Loading Screen Prank 📄
- g. Lab: UN MDG Quiz System 📄
- h. Test 📄

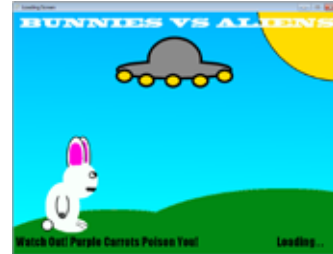


Figure 4: Loading Screen Lab

9. Game Design

- a. What is a game?
- b. Seven Elements of Game Design
- c. Worksheet: My Favorite Games 📄
- d. What is fun?
- e. Worksheet: Manipulating Game Elements 📄
- f. Video: State Machine Tutorial 🎥
- g. Game Design Document
- h. Lab: Game for the Visually Impaired 📄
- i. Lab: Super Audio Kart 📄
- j. Elevator Pitch 📄
- k. Testing and balancing a game
- l. Worksheet: Play Testing 📄

10. Reading Text Input (optional)

- a. References and values
- b. Enumerated types
- c. Reading text input
- d. Chapter 9 Reading Questions 📄
- e. Lab: Customization 📄



Figure 5: Customization Lab

11. OOP and Non-Violent Conflict Game





- a. Methods
- b. Debugging
- c. Object-Oriented Game Design
- d. Chapter 10 Reading Questions 
- e. Worksheet: OOP! There It Is. 
- f. Lab: Freeze Tag 
- g. Lab: Nashville 1960 – A Non-Violent Conflict Game 



Figure 6: Nashville 1960 Lab

12. Two-Dimensional Arrays





- a. Two-Dimensional Arrays
- b. Lab: The Legend of Zigmund 
- c. Test 
- d. Recursion
- e. Lab: Contagion 
- f. Lab: Board Game 



Figure 7. The Legend of Zigmund