



## TeenCoder: Game Programming

### Printed Course Syllabus and Planner

*Updated October, 2015*

#### Printed Course Overview

**Course Title:** *TeenCoder: Game Programming*

**Textbook ISBN:** 978-0-9887033-2-2, published 2013 by Homeschool Programming, Inc.

**Length:** 1 Semester

**Student Pre-Requisites:** Basic computer literacy skills, 9<sup>th</sup> – 12<sup>th</sup> grade status. ***TeenCoder: Windows Programming* is a pre-requisite before starting this second-semester material.** Students will use the same C# language and Visual C# IDE skills learned in *TeenCoder: Windows Programming* to complete their game programming projects.

**Description:** *TeenCoder: Game Programming* is a one-semester course introducing computer video game design skills and concepts using the C# language and the XNA Game Studio.

#### Materials:

- Student Textbook
- Visual Studio IDE, XNA Game Studio
- Windows personal computer
- Course activities (hands-on programming assignments)
- Course supplemental documentation
- Course instructional videos (optional)

#### Labs and Grading

Every chapter contains one or more hands-on programming labs where students will design or implement programs to demonstrate understanding of the lesson topics. These labs, combined with end-of-chapter tests, are used to determine the student grade.



# Homeschool Programming, Inc.

## Course Planner

A typical school semester consists of approximately 18 calendar weeks or 90 days of school. This course plan covers 17 school weeks of core material, assuming students are working 3-5 hours per week to stay on pace. Some students may move faster or slower than the suggested pace.

Students may choose to implement a relaxed two-semester course by devoting less time per week or moving at a slower pace. The “Week” column in the planner below reflects a one-semester plan, and those on a two-semester plan can simply scale accordingly.

Week	Reading and Objectives	Labs
1	Chapter One: Intro to Game Programming <ul style="list-style-type: none"><li>• What You Already Know</li><li>• Types of Computer Games</li><li>• What You Will Learn in this Course</li><li>• Introduction to XNA</li></ul>	<b>Install XNA Game Studio</b> - Install Microsoft Visual C# IDE, XNA Game Studio.
2	Chapter Two: Game Design <ul style="list-style-type: none"><li>• The Game Proposal</li><li>• The Game Engine</li><li>• Creating an XNA Game Project</li><li>• The Game Loop</li></ul>	<b>Looping Colors</b> – The student use a game loop to periodically change colors on the screen.
3	Chapter Three: Graphics Concepts <ul style="list-style-type: none"><li>• Screen Coordinates</li><li>• Full Screen vs. Window Mode</li><li>• Colored Pixels</li></ul>	<b>Screen Toggle</b> – The student will create an application to blink the screen in a rotating series of colors.



# Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
4	<p>Chapter Four: Working with Images</p> <ul style="list-style-type: none"><li>• Surfing the Content Pipeline</li><li>• Drawing Images</li><li>• Image Transformations</li><li>• Drawing Text</li></ul>	<p><b>Starry Night</b> – The student will demonstrate scaling and rotating pixels and displaying transparent graphics to create a landscape.</p>
5	<p>Chapter Five: User Input</p> <ul style="list-style-type: none"><li>• Keyboard Input</li><li>• Mouse Input</li><li>• XBox 360 Controller</li></ul>	<p><b>Cat and Mouse</b> – The student will create a simple chase game to demonstrate keyboard and mouse processing.</p>
6	<p>Chapter Six: Sprites</p> <ul style="list-style-type: none"><li>• Introducing Sprites</li><li>• The Swarm Game</li><li>• Initializing Your Swarm</li><li>• Sprite Movement</li></ul>	<p><b>Raising the Swarm</b> – The student will write code to initialize the game screen and main objects.</p> <p><b>Buzzing Bees</b> – The student will write code to make the bees move on the screen.</p>
7-8	<p>Chapter Seven: Completing Swarm</p> <ul style="list-style-type: none"><li>• Adding Player Control</li><li>• Shooting Stingers and Smoke</li><li>• Collision Detection</li><li>• Ending and Restarting the Game</li></ul>	<p><b>Sliding Smoke Gun</b> – Make the smoke gun move in response to keyboard commands.</p> <p><b>Shooting the Swarm</b> – The student will make the smoke gun fire shots in response to keyboard commands.</p> <p><b>Feeling the Sting</b> – The student will add collision detection between sprites in Swarm.</p>



# Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
9	Chapter Eight: Animation <ul style="list-style-type: none"><li>• Animation Concepts</li><li>• Animation Textures</li><li>• Animation in the Sprite Class</li></ul>	<b>Animating the Swarm</b> – The student will add animated images to several sprites in the Swarm game.
10	Chapter Nine: Music and Sound Effects <ul style="list-style-type: none"><li>• Sound Files</li><li>• Playing Sound Effects</li><li>• Playing Music</li><li>• The XACT Tool</li></ul>	<b>Audible Swarm</b> – The student will add background music and sound effects to the Swarm game.
11	Chapter Ten: Game Physics <ul style="list-style-type: none"><li>• Velocity and Acceleration</li><li>• Gravity and Wind</li><li>• Reflection</li></ul>	<b>Snowball Fight</b> – The student will implement logic for a classic artillery-style game.
12	Chapter Eleven: Maze Generation <ul style="list-style-type: none"><li>• Maze Types</li><li>• Generating a Perfect Maze</li><li>• Solving a Perfect Maze</li></ul>	<b>A-Maze-Ing Backtracker</b> – The student's will write recursive backtracker algorithms to generate and solve a maze.
13	Chapter Twelve: Menus, Overlays, and Deployment <ul style="list-style-type: none"><li>• Title Screens and Option Menus</li><li>• Handling Different Screens</li><li>• Displaying Scores and Overlays</li><li>• Distributing Games</li></ul>	<b>Tic-Tac-Toe</b> – The student will implement a tic-tac-toe game that demonstrates title screens, menus, scoring, and overlays.



# Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
14	<p>Chapter Thirteen: Multiplayer Games</p> <ul style="list-style-type: none"><li>• Handling Multiple Inputs</li><li>• Scrolling Games</li><li>• Viewports and Cameras</li></ul>	<p><b>Star Racer</b> – The student will write the camera and viewport logic for a vertically scrolling racing game.</p>
15	<p>Chapter Fifteen: Artificial Intelligence</p> <ul style="list-style-type: none"><li>• Understanding AI</li><li>• Developing an AI Algorithm</li><li>• Simple Movement Algorithms</li><li>• AI for Star Racer</li></ul>	<p><b>Star Racer AI</b> – The student will improve the existing Star Racer AI routines from the previous lesson.</p>
16-17	<p>Chapter Sixteen: Final Project</p> <p>For a final project the student will create an arcade game called “Bumper Cars”. The project contains a series of guided steps as activities. The student can test their program at each step to ensure each feature works correctly.</p> <ul style="list-style-type: none"><li>• Bumper Cars Overview</li><li>• Menus and Controls</li><li>• Adding Cars</li><li>• Oil Slicks and Coins</li><li>• Bumper Car Sound Effects</li><li>• Add Artificial Intelligence</li></ul>	<p><b>Project Kick-Off</b> – The student will build and run the activity starter project.</p> <p><b>What’s on the Menu?</b> – The student will implement the game menu system.</p> <p><b>Start Your Engines</b> – The student will add the player car objects to the game.</p> <p><b>Hazards and Rewards</b> – The student will add the coins as targets and oil slicks as hazards.</p> <p><b>Make Some Noise</b> – The student will add background music and sound effects.</p> <p><b>Racing Buddy</b> – The student will write an AI routine to give the second car an auto-pilot option.</p>