

JavaScript Game Development: Learn by Building 15 Real Games

Build 15 JavaScript games. Learn variables, functions, DOM, and events by coding Snake, Pong, Tic Tac Toe, and more.

Programming Instructor: David Clarke • 5 sections • 43 lessons • 12.25h total

[View this course online](#) [Browse all courses](#)

About this course

Most JavaScript tutorials teach theory without real application, leaving you unsure how to actually build things. This course solves that by putting you into game development, where you will learn variables, functions, loops, objects, and arrays while creating playable games. You will code 15 complete games including Snake, Pong, Tic Tac Toe, and original concepts, gaining hands-on experience with DOM manipulation, event handling, and problem-solving. The course includes video walkthroughs, downloadable source code, and case study analysis, so you finish with both skills and a portfolio of games.

Course curriculum

Foundations and Setup: JavaScript Basics for Games

- **How JavaScript Powers Interactive Games (12m)**
Understand the relationship between JavaScript and game logic, and see how variables and functions create dynamic gameplay experiences.
- **Setting Up Your Game Development Environment (10m)**
Configure your code editor, browser tools, and file structure for efficient game development workflow.
- **Variables and Data Types in Game Context (15m)**
Learn how to declare and use variables to track game state, player scores, and character positions with practical examples.
- **Functions: Reusable Game Logic Blocks (18m)**
Create modular functions for game actions, collision detection, and scoring systems that you'll reuse across multiple games.
- **Conditional Logic for Game Decisions (14m)**
Master if/else statements and switch cases to handle game rules, win conditions, and character behavior.
- **Loops for Animation and Game Mechanics (16m)**
Use for and while loops to create smooth animations, generate game levels, and process player input.
- **Arrays: Managing Game Elements and High Scores (13m)**
Store multiple game objects, track collections of items, and maintain leaderboards using array methods.
- **Objects: Organizing Game State and Characters (17m)**
Structure game data with objects, creating player characters, enemies, and game settings in an organized way.

DOM Manipulation and Event Handling for Games

- **Selecting and Modifying Game Elements (14m)**
Use querySelector and DOM methods to update game boards, display scores, and show/hide game screens.
- **Keyboard Events for Game Controls (16m)**
Capture arrow keys, spacebar, and custom keys to control characters and trigger game actions.
- **Mouse Events for Click-Based Games (12m)**
Handle clicks, drag operations, and hover states for puzzle games and interactive interfaces.
- **Creating Smooth Animations with requestAnimationFrame (20m)**
Build efficient game loops that create smooth animations without overwhelming the browser.
- **Displaying Real-Time Game Information (11m)**
Update score displays, timers, and status messages in real-time as the game progresses.
- **Game State Management with DOM Updates (18m)**
Track and display game phases, level progression, and win/lose states through strategic DOM manipulation.

- **Creating Modal Windows for Game Menus (13m)**
Build start screens, pause menus, and game over dialogs that enhance the player experience.
- **Responsive Game Design Considerations (15m)**
Adapt game layouts and controls for different screen sizes while maintaining gameplay quality.
- **Debugging Game Logic in the Browser (14m)**
Use browser developer tools to identify and fix common game development issues.

Building Classic Games: Tic Tac Toe and Memory Match

- **Tic Tac Toe: Game Board Setup and Logic (22m)**
Create the classic 3x3 grid, implement turn-based gameplay, and detect winning patterns.
- **Tic Tac Toe: AI Opponent with Basic Strategy (19m)**
Build a simple computer opponent that makes strategic moves and provides a challenging experience.
- **Memory Match: Card Layout and Shuffling (18m)**
Generate randomized card layouts and implement the memory matching gameplay mechanics.
- **Memory Match: Match Detection and Scoring (16m)**
Track matched pairs, update scores, and create win conditions for the memory game.
- **Refactoring Game Code for Reusability (17m)**
Restructure your code using functions and objects to make it cleaner and more maintainable.

Advanced Game Development: Snake and Pong

- **Snake Game: Grid-Based Movement System (24m)**
Implement the classic snake movement mechanics using arrays to track body segments and position.
- **Snake Game: Food Generation and Growth Mechanics (20m)**
Randomly place food items, detect collisions, and dynamically increase the snake's length.
- **Snake Game: Collision Detection and Game Over Logic (18m)**
Handle wall collisions, self-collision detection, and create end-game scenarios with score display.
- **Pong: Ball Physics and Bounce Mechanics (22m)**
Create realistic ball movement with angle calculations and collision responses on paddles.
- **Pong: Paddle Controls and AI Opponent (19m)**
Implement responsive paddle movement for both player and computer-controlled opponents.
- **Pong: Score Tracking and Game Flow (15m)**
Track points, detect scoring events, and manage round transitions with serve mechanics.
- **Creating Reusable Game Utility Functions (17m)**
Extract common game patterns like collision detection and random number generation into reusable utilities.
- **Optimizing Game Performance (16m)**
Identify and fix performance bottlenecks in your games for smooth gameplay experience.

Portfolio Development and Advanced Techniques

- **Building a Game Selection Menu Interface (20m)**
Create an attractive menu system that showcases all your games with screenshots and descriptions.

- **Adding Game Instructions and Help Systems (14m)**
Design clear instruction screens and in-game help features that improve user experience.
- **Implementing Local Storage for High Scores (18m)**
Save and retrieve game data locally to maintain high score boards and player preferences.
- **Creating Original Game Concepts (25m)**
Design and implement your own game ideas using the patterns and techniques learned throughout the course.
- **Code Organization with Modules and Patterns (22m)**
Structure complex games using modular design patterns and separation of concerns principles.
- **Adding Polish: Sound Effects and Visual Feedback (19m)**
Enhance games with audio cues, particle effects, and visual feedback that improve engagement.
- **Building a Responsive Game Portfolio Website (24m)**
Create a professional portfolio site that showcases your games with descriptions and live demos.
- **Writing Clean Code Documentation (16m)**
Document your game code with comments and README files that explain functionality and setup.
- **Testing and Debugging Strategies (17m)**
Implement systematic testing approaches to identify and fix bugs across different browsers.
- **Version Control for Game Projects (15m)**
Use Git to track your game development progress and maintain organized project history.
- **Performance Optimization Techniques (21m)**
Advanced methods for improving game performance including efficient rendering and memory management.
- **Preparing Your Portfolio for Job Applications (18m)**
Tailor your game portfolio for specific job opportunities and highlight relevant technical skills.
- **Next Steps: Advanced JavaScript Game Development (14m)**
Explore advanced topics like WebGL, game frameworks, and multiplayer game development.

Discover our blog

Learn more with free articles written by our professors and guest experts.

Latest article: **[Flight Attendant Preparation: What to Study Before You Apply](#)**

[Visit the blog](#) [All courses](#) [This course](#)