

Modern UI Development in Unity 6.5

Build a complete game UI from scratch using Unity's Canvas and UI Toolkit systems. For intermediate Unity developers.

Programming Instructor: Cameron Brooks • 5 sections • 52 lessons • 17.00h total

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

About this course

Most Unity tutorials teach UI basics in isolation, then leave you to figure out how it all connects in a real project. You end up with scattered knowledge but no clear picture of how to build a full game interface from start to finish. This course changes that by walking you through every core UI concept while simultaneously building out the entire interface for a 2D strategy card game.

You'll start with Canvas fundamentals, layout groups, and responsive design, then move into runtime UI generation, data binding with MVVM patterns, and shader-based effects. Each section builds on the last, and by the final module you'll have assembled a working main menu, in-game HUD, inventory grid, card hand system, and victory screens. Every lesson includes a real implementation you can follow along with and adapt to your own projects.

Course curriculum

UI Fundamentals and Setup

- **Setting Up Your Unity 6.5 Project for UI Development (12m)**
Configure a clean project with the right packages, input system, and screen settings so you're ready to build UI from the first lesson.
- **Canvas and Render Modes: What They Do and When to Pick Each (18m)**
Understand Screen Space Overlay, Screen Space Camera, and World Space canvases with practical examples of when each one makes sense.
- **RectTransform: How Positioning Actually Works (22m)**
A thorough breakdown of the RectTransform component, covering anchors, pivots, and the difference between stretch and fixed-size modes.
- **Basic Controls: Text, Button, and Image Components (15m)**
Wire up the three most-used UI components, configure their properties, and understand how TextMeshPro integrates into the workflow.
- **UI Toolkit vs. Legacy Canvas: Choosing the Right System (14m)**
Compare the two UI approaches in Unity 6.5 with honest pros and cons, so you can pick the best fit for your project's needs.
- **Building Your First Menu Screen from Scratch (25m)**
Put everything together to create a polished main menu with title, buttons, and background, covering layout and hierarchy best practices.
- **Handling Click Events and Basic UI Navigation (18m)**
Connect buttons to game logic using UnityEvents and C# callbacks. Includes keyboard navigation setup for accessibility.
- **Making the Menu Responsive Across Different Resolutions (20m)**
Test and fix your menu for different aspect ratios and screen sizes using the Canvas Scaler and anchor strategies you learned earlier.

Layout, Navigation, and Responsive Design

- **Horizontal and Vertical Layout Groups in Practice (18m)**
Use layout groups to automatically arrange buttons, stat bars, and list items without manually positioning each element.
- **Grid Layout Group for Card and Inventory Interfaces (22m)**
Build a grid-based inventory layout with proper cell sizing, spacing, and alignment for a card game aesthetic.
- **Content Size Fitters, Layout Elements, and Priority Rules (14m)**
Control how child elements expand, shrink, and prioritize space within layout groups for predictable results.
- **Deep Dive into Anchors and Pivots for Multi-Screen Design (20m)**
Master the anchor and pivot system to build UI that stays properly positioned on everything from a phone to an ultrawide monitor.

- **Canvas Scaler Settings for Phones, Tablets, and Desktops (15m)**
Walk through the three Canvas Scaler modes and set up reference resolution strategies that actually hold up across devices.
- **Building a Robust Keyboard and Controller Navigation System (22m)**
Implement the Navigation system for gamepad and keyboard users, with custom tab order and highlight visuals for selected elements.
- **Creating a Scroll View from the Ground Up (25m)**
Build a scrollable list for card collections, leaderboards, or settings menus, including scrollbar styling and elastic bouncing.
- **Managing UI State and Panel Switching (18m)**
Create a panel manager that handles transitions between screens, saves state, and prevents input conflicts during switches.
- **Responsive Design Workshop: Building a Settings Panel (20m)**
Apply everything from this section to build a full settings screen with tabs, sliders, dropdowns, and toggle groups that adapt to any resolution.
- **Nested Canvas and Canvas Group Optimization (14m)**
Learn when and why to nest canvases for performance, and use Canvas Groups to manage opacity and interactivity across child elements.

Dynamic UI Systems and Data Binding

- **Runtime UI Instantiation with Prefabs (18m)**
Create and destroy UI elements at runtime using prefabs, covering parenting, layout refresh timing, and common pitfalls.
- **Object Pooling for UI Elements to Reduce Garbage Collection (22m)**
Build a simple but effective object pool for frequently spawned UI items like notifications, damage numbers, and list entries.
- **ScriptableObject-Driven UI Content (15m)**
Use ScriptableObjects to define card data, character stats, and tooltip text so your UI reads from a single source of truth.
- **Health Bars, Mana Bars, and Status Indicators (20m)**
Create animated resource bars that respond to gameplay events, including smooth fill transitions and color changes at thresholds.
- **Skill Cooldown Displays and Circular Timers (18m)**
Build radial cooldown overlays using Image Fill Amount and shader approaches, with real-time timer synchronization.
- **Building an Inventory Grid with Drag-and-Drop (28m)**
Implement a fully functional inventory grid where items can be dragged between slots, with snap-back behavior and swap logic.
- **Introduction to MVVM for Unity UI Projects (22m)**
Understand the Model-View-ViewModel pattern and set up a lightweight framework that separates game logic from UI presentation.

- **Data Binding Patterns: Connecting Game State to UI Updates (18m)**
Implement observable properties and event-driven updates so your UI refreshes automatically when data changes without polling.
- **Tooltip and Context Menu Systems (20m)**
Build hover tooltips that follow the cursor and right-click context menus that adapt their options based on the target element.

Advanced Styling, Animation, and Effects

- **USS Fundamentals: Styling UI Toolkit Elements (18m)**
Learn Unity Style Sheets syntax, pseudo-classes, and selectors to style UI Toolkit elements with custom themes and hover states.
- **UXML Structure: Building Reusable Templates (15m)**
Write UXML documents that define reusable component templates, reducing duplication and keeping your UI hierarchy clean.
- **Theming and Runtime Style Switching for Dark Mode and Beyond (20m)**
Build a theming system that lets users toggle between light and dark modes, or apply seasonal visual themes at runtime.
- **Unity Animation System for UI Transitions (22m)**
Use the Animator and Animation Clips to create panel slide-ins, pop-up bounces, and multi-state transitions with easing curves.
- **Tweening UI Elements for Smooth Micro-Interactions (18m)**
Implement lightweight tweening for button hover scales, notification slides, and subtle feedback animations without the full Animator.
- **Shader-Based UI: Dissolve and Masking Effects (25m)**
Write custom shaders that create dissolve transitions for card reveals and masking effects for health bar damage indicators.
- **Shader-Based UI: Blur, Distortion, and Refraction (22m)**
Apply post-process-style blur behind pop-up panels and distortion effects for magical ability UI elements using shader graph.
- **Particle Effects and Trail Renderers on Canvas (18m)**
Add sparkle, smoke, and trail effects to UI elements by layering particle systems on a World Space canvas overlay.
- **Reducing UI Draw Calls and Fixing Overdraw Issues (20m)**
Diagnose and fix overdraw problems, batch UI materials, and use sprite atlases to cut draw calls in half on complex screens.
- **Profiling UI Performance with Unity's Built-in Tools (15m)**
Use the Frame Debugger, Profiler, and UI Toolkit Debugger to identify bottlenecks and measure the impact of your optimizations.
- **Building a Custom Reusable UI Component (25m)**
Create a polished, reusable radial menu component from scratch, covering design, animation, event callbacks, and editor integration.

- **Adding Accessibility Features to Your UI (18m)**
Implement screen reader hints, colorblind-friendly palettes, text scaling options, and input remapping for a wider audience.

Capstone Project: Building a Complete Game UI

- **Planning the Full Game UI Architecture (15m)**
Map out every screen, panel, and overlay for the strategy card game. Set up the folder structure and establish naming conventions.
- **Main Menu and Title Screen Implementation (22m)**
Build a complete title screen with animated logo, menu buttons, background art, and transitions into sub-screens.
- **Settings Panel: Audio Sliders, Graphics Options, and Key Bindings (25m)**
Implement a full settings screen with volume controls, resolution dropdowns, quality presets, and rebindable input fields.
- **In-Game HUD: Resource Counters and Unit Info Display (20m)**
Build the persistent in-game interface showing gold, mana, turn count, and selected unit stats with live data binding.
- **Card Hand System: Fan Layout and Drag-to-Play (32m)**
Create the player's hand of cards with a curved fan layout, hover-to-zoom preview, and drag-to-play on the game board.
- **Minimap and Tactical Board Overlay (18m)**
Render a minimap of the game board and add hex-grid overlays that highlight valid move ranges and attack zones.
- **Turn Indicator and Action Queue Display (20m)**
Build a turn order bar showing upcoming units with portrait thumbnails, health previews, and active turn highlighting.
- **Notification System and Event Popups (18m)**
Create a toast notification queue for in-game events and modal popups for confirmations, with proper input blocking.
- **Victory, Defeat, and Round Summary Screens (15m)**
Design end-of-game screens with score breakdowns, rewards display, and buttons for replay or return to menu.
- **Connecting UI to Game Logic: Selection, Commands, and Feedback (25m)**
Wire all UI elements to actual gameplay systems, handling card play validation, unit selection callbacks, and resource updates.
- **Saving and Loading UI Preferences and Progress (15m)**
Persist settings, unlocked cards, and UI layout preferences using JSON serialization and Unity's persistent data path.
- **Platform-Specific Adjustments for Desktop and Mobile (20m)**
Adapt the UI for touch input, different safe areas, and platform-specific input prompts using preprocessor directives.

- **Final Polish: Consistent Styling, Bug Fixes, and Build** (22m)

Do a final pass on visual consistency, fix edge-case layout issues, run the profiler one last time, and build the finished game.

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](#)