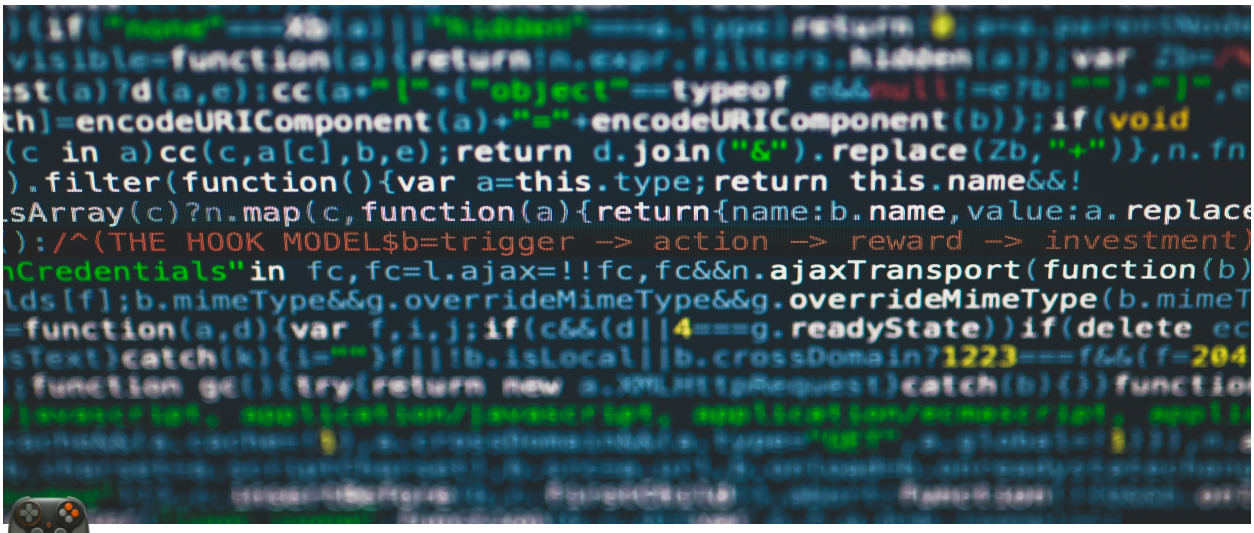

C# Scripting

Phase 1: Core Syntax & Algorithmic Thinking

Objective: Master basic syntax, control structures, and problem-solving with simple console applications.

Project 1: Even/Odd Checker & Basic I/O

- **Task:** Create a console app that accepts an integer and identifies if it's even or odd using the modulus operator (`%`).
- **Skills:**
  - Input validation with `TryParse` 35.
  - Conditional logic (`if` / `else`).
- **Extension:** Add support for negative numbers and zero.

Project 2: Fibonacci Series Generator

- **Task:** Generate the first *N* Fibonacci numbers using iterative loops or recursion.
- **Skills:**
  - Loop structures (`for` , `while`).
  - Recursion basics (if implemented).
- **Extension:** Optimize recursion with memoization.

  {···} <CODE>

Project 3: Palindrome Checker

- **Task:** Determine if a string reads the same backward (ignoring case and punctuation).
- **Skills:**

- String manipulation (`ToLower()`, `Replace()`).
- Array reversal and comparison 5.

```
<CODE>
```

Project 4: Console Calculator

- **Task:** Build a calculator supporting addition, subtraction, multiplication, and division.
- **Skills:**
  - Method encapsulation (e.g., `Calculator.DoOperation()` 3).
  - Error handling with `try`/`catch` blocks.
- **Extension:** Add support for exponents or modular arithmetic.

Phase 2: Object-Oriented Programming & Data Structures

Objective: Implement OOP principles and common data structures.

Project 5: Bank Account Manager

- **Task:** Create a `BankAccount` class with methods for deposits, withdrawals, and balance checks.
- **Skills:**
  - Class design (`private` fields, `public` methods).
  - Encapsulation and validation (e.g., prevent negative balances).
- **Extension:** Add transaction history using `List<Transaction>`.

Project 6: Inventory System

- **Task:** Model an inventory with items (name, quantity, price) and methods to add/remove stock.
- **Skills:**
  - Collections (`Dictionary<string, Item>`).
  - LINQ queries for filtering items 8.

Project 7: Text-Based RPG Combat Simulator

- **Task:** Simulate turn-based combat between a player and enemy with health, attack, and defense stats.
- **Skills:**
  - Inheritance (e.g., `Enemy : Character`).
  - Polymorphism (override `Attack()` method for different enemy types).

Phase 3: File I/O, Error Handling & Advanced Features

Objective: Work with external data and robust error management.

Project 8: Quiz Application

- **Task:** Load questions from a `.txt` or `.csv` file and track user scores.
- **Skills:**
  - File reading/writing (`StreamReader`, `StreamWriter`).
  - Serialization/deserialization with `System.Text.Json` 8.

Project 9: Log Analysis Tool

- **Task:** Parse a log file to count errors, warnings, and info messages.
- **Skills:**
  - Regular expressions for pattern matching.
  - Custom exceptions for invalid log formats.

Project 10: Multiplayer Tic-Tac-Toe via Sockets

- **Task:** Implement a two-player game using `TcpListener` and `TcpClient`.
- **Skills:**
  - Network programming basics 6.
  - Thread synchronization for real-time updates.

Phase 4: Desktop Applications & Advanced Patterns

Objective: Build GUI apps with Windows Forms/WPF and apply architectural patterns.

Project 11: Paint App (Windows Forms)

- **Task:** Allow users to draw shapes/colors with mouse input.
- **Skills:**
  - Event handling (`MouseDown`, `MouseMove`).
  - GDI+ graphics rendering 6.

Project 12: Budget Tracker (WPF/MVVM)

- **Task:** Develop an app to track income/expenses with charts.
- **Skills:**
  - MVVM pattern (`INotifyPropertyChanged`).
  - Data binding to `ObservableCollection<T>`.

Project 13: REST API Client

- **Task:** Fetch data from a public API (e.g., weather data) and display it in a GUI.
- **Skills:**
  - HTTP requests with `HttpClient`.
  - JSON parsing (`Newtonsoft.Json`).

Recommended Resources

Documentation & Tutorials

1. **Microsoft C# Guide** 8: Authoritative resource for syntax, features, and best practices.
2. **Visual Studio Tutorials** 3: Step-by-step guides for debugging and project setup.

Books

- **"C# in Depth" (4th Edition)** 7: Explores advanced language features like generics and async/await.
- **"Pro C# 7"** 7: Comprehensive coverage of .NET Core and OOP design.

Communities & Practice

- **Exercism.io**: Solve C# coding challenges with mentor feedback.
- **LeetCode**: Practice algorithms and data structures.

Project 14: Real-Time Terminal Space Shooter

Objective: Implement asynchronous input handling and game loop timing

Core Implementation

- **Real-Time Controls:**

  Use `Console.KeyAvailable` with `Task.Run()` for non-blocking input handling2

  ```
  csharpTask.Run(() => {
    while (true)
      if (Console.KeyAvailable)
        HandleInput(Console.ReadKey(true).Key);
  });
  ```

- **Projectile System:**

  Manage laser beams with `List<Vector2>` and delta-time movement calculations

- **Enemy AI Patterns:**

  Implement sinusoidal movement using `Math.Sin(DateTime.Now.Millisecond * 0.001f)`

Advanced Features

- **Screen Buffer Optimization:**

  Use `StringBuilder` for efficient frame rendering2

- **Particle Effects:**

  Create explosion animations with randomized ASCII characters

- **Score System with File Persistence:**

  Store top scores using `System.IO.File` operations

Project 15: Procedural Dungeon Explorer with RPG Elements

Objective: Master procedural generation and complex state management

Key Components

- **Dungeon Generation Algorithm:**

  Implement drunkard's walk algorithm with `Random` seed control

- **Inventory System:**

  Use `Dictionary<ItemType, int>` with LINQ queries for equipment management6

- **Turn-Based Combat:**

  Create state machine handling attack phases and status effects

Technical Challenges

- **Save/Load System:**

  Serialize game state to JSON using `System.Text.Json`

- **Pathfinding:**

  Implement A* algorithm with priority queue6

- **Procedural Loot Tables:**

  Use weighted probability system with `Enumerable.Range`

Project 16: Multiplayer Console MMO Prototype

Objective: Synthesize all concepts into networked application

Architecture

- **Client-Server Model:**

  Implement TCP server using `TcpListener` with thread pool8

- **Protocol Design:**

  Create custom binary protocol with header/body structure
  `text[4-byte length][1-byte opcode][n-byte payload]`

- **World Simulation:**

  Use `ConcurrentDictionary` for thread-safe entity management

Advanced Features

- **Chat System:**

  Implement IRC-style commands with regex parsing5

- **Guild Management:**

  Create role-based permission system with inheritance

- **Realtime Stats Tracking:**

  Use `Stopwatch` for combat timing and XP calculations

Unity Advanced Scripting Projects

Project 17: Custom Networked Co-op Puzzle Game

Objective: Bypass Unity's HLAPI with raw socket programming

Implementation Strategy

- **Network Layer:**

  Create `GameTransport` base class with UDP/TCP implementations8

- **Entity Serialization:**

  Develop custom binary serializer using `MemoryStream`

- **Prediction/Reconciliation:**

  Implement client-side movement prediction with server reconciliation

Key Scripts

```
csharppublic class NetworkedPlayer : MonoBehaviour {
  private Vector3 _serverPosition;
  private Queue<PlayerInput> _inputBuffer = new Queue<PlayerInput>();

  void Update() {
    if (IsLocalPlayer) {
      SendInputToServer();
      PredictMovement();
    }
  }
}
```

Project 18: Procedural Terrain Tool with Editor Extensions

Objective: Master Unity Editor scripting and advanced algorithms

Feature Set

- **Custom Editor Window:**

  Create `TerrainGeneratorWindow` with IMGUI controls7

- **Noise Algorithms:**

  Implement fractional Brownian motion (fBm) using `Mathf.PerlinNoise`

- **LOD System:**

  Develop quad tree-based level of detail with coroutines

Advanced Techniques

- **Shader Graph Integration:**

  Create biome blending using vertex colors and texture arrays

- **Undo/Redo Support:**

  Implement `Undo.RecordObject` for terrain modification history

- **ScriptableObject Pipeline:**

  Store terrain presets as asset files using `CreateAssetMenu`

Curated Resources for Advanced Projects

1. **RogueSharp Tutorials** (Search Result 6):

   Study pathfinding implementations in `RogueSharp.Algorithms`

2. **Orleans Adventure Sample** (Search Result 9):

   Analyze distributed actor model for MMO architecture

3. **Unity Networking API** (Search Result 8):

   Review `TcpGameServer` implementation for low-level patterns

4. **Console Rendering Techniques** (Search Result 2):

   Learn buffer swapping from Sokoban-like terminal game

This progression takes you from advanced console programming through to professional-grade Unity development, ensuring deep understanding of both vanilla C# and game engine integration. Each project introduces industry-relevant patterns while reinforcing core language concepts.

Citations:

1. https://www.youtube.com/playlist?list=PLv3FpT6JrAs4ZBl0pH9lTdUVEMubidCJk
2. https://www.youtube.com/watch?v=5owKZgbIB9Q
3. https://github.com/OriCohen05/Text-Based-Adventure-Game-in-CS
4. https://www.reddit.com/r/csharp/comments/zoa847/can_someone_give_me_a_good_idea_for_c_console_app/
5. https://www.c-sharpcorner.com/UploadFile/97ec13/how-to-make-a-chat-application-in-C-Sharp/
6. https://www.roguebasin.com/index.php/C_Sharp
7. https://docs.unity3d.com/Manual/editor-CustomEditors.html
8. https://github.com/ycarowr/UnityNetworkingAPI
9. https://learn.microsoft.com/en-us/dotnet/orleans/tutorials-and-samples/adventure
10. https://dev.to/zacharypatten/more-c-console-games-27bg
11. https://www.interviewbit.com/blog/c-sharp-projects/
12. https://www.codeproject.com/Articles/1222027/Creating-a-Very-Simple-Console-Chat-App-using-Csha
13. https://www.youtube.com/watch?v=Og4ZwQV43lQ
14. https://docs-multiplayer.unity3d.com/netcode/current/tutorials/helloworld/
15. https://www.youtube.com/watch?v=TjuIAff0JVs
16. https://www.youtube.com/watch?v=dZSLm4tOI8o
17. https://learn.microsoft.com/en-us/visualstudio/get-started/csharp/tutorial-console?view=vs-2022
18. https://www.youtube.com/watch?v=jiKf-9MLm7Y
19. https://www.youtube.com/watch?v=5p2GePo8jmY
20. https://github.com/topics/terminal-game
21. https://www.toolify.ai/gpts/create-an-epic-textbased-adventure-game-in-c-168886
22. https://www.reddit.com/r/csharp/comments/l8wtkx/c_cli_app_ideas/

23. https://learn.microsoft.com/en-us/dotnet/core/tools/global-tools-how-to-create
24. https://www.youtube.com/watch?v=8NEcTdkw_dw
25. https://www.youtube.com/watch?v=8F2F_P8tXfg
26. https://unity.com/how-to/programming-unity
27. https://statanalytica.com/blog/c-project-ideas-for-beginner/
28. https://www.youtube.com/watch?v=5BrxzEoYZfM
29. https://docs-multiplayer.unity3d.com/netcode/current/tutorials/get-started-ngo/
30. https://www.reddit.com/r/csharp/comments/199w1l5/for_anyone_interested_in_writing_text_adventures/
31. https://learn.microsoft.com/en-us/dotnet/core/tutorials/top-level-templates
32. https://github.com/kathleenwest/AdvancedConsoleApplicationIII
33. https://anthonysimmon.com/the-best-csharp-repl-is-in-your-terminal/
34. https://www.guvi.in/blog/best-c-sharp-project-ideas/
35. https://www.youtube.com/watch?v=I-Xmp-mulz4
36. https://www.youtube.com/watch?v=_gJ0NbNKKZg
37. https://learn.microsoft.com/en-us/dotnet/core/tutorials/with-visual-studio-code
38. https://www.c-sharpcorner.com/UploadFile/dommym/building-the-really-really-really-simple-roguelike-v0-1-with/
39. https://blog.udemy.com/c-sharp-projects/
40. https://www.reddit.com/r/csharp/comments/1cln718/advanced_net_project_ideas/
41. https://www.upgrad.com/blog/project-ideas-topics-in-c-for-beginners/
42. https://unity.com/how-to/advanced-programming-and-code-architecture
43. https://www.youtube.com/watch?v=VEnZzwW_OJM
44. https://www.masterycoding.com/blog/beginner-unity-project-ideas
45. https://codersera.com/blog/advanced-c-programming-project-ideas
46. https://docs.unity3d.com/560/Documentation/Manual/editor-CustomEditors.html
47. https://discussions.unity.com/t/custom-editor-and-visual-studio-solutions/235025
48. https://www.youtube.com/watch?v=kFLaS1Nyrdw
49. https://github.com/Pycorax/Unity-Custom-Editor-Example
50. https://www.reddit.com/r/csharp/comments/lo45cz/i_am_new_to_programming_could_you_guys_give_me/
51. https://github.com/dotnet/dotnet-console-games
52. https://codersera.com/blog/c-console-app-ideas-a-comprehensive-guide

---

Answer from Perplexity: pplx.ai/share