

Main Menu System - Quick Reference

🎯 Quick Setup (5 Steps)

1. Create new scene "MainMenu"
2. Build UI canvas with panels
3. Add MainMenuManager script
4. Create Credits.txt in Resources
5. Add input blocking to game scripts

📋 UI Hierarchy

```
Canvas
└── MainMenuPanel
    ├── StartContinueButton
    ├── OptionsButton
    ├── CreditsButton
    └── ExitButton
    └── OptionsPanel (inactive)
        ├── VolumeSlider
        └── BackButton
    └── CreditsPanel (inactive)
        ├── CreditsText
        └── BackButton
```

🔧 Essential References

yaml

Panels:

- mainMenuPanel
- optionsPanel
- creditsPanel

Buttons:

- startContinueButton
- optionsButton
- creditsButton
- exitButton

Options:

- volumeSlider
- volumeValueText

Credits:

- creditsText
- creditsFile (TextAsset)

Build Settings

File → Build Settings

Scenes in Build:

0. ✓ MainMenu (FIRST!)
1. ✓ GameScene

Button States

State	Text	Action
Initial	"Start Game"	Load scene
Loading	"Loading..."	Disabled
Loaded	"Continue Game"	Resume
Paused	"Continue Game"	Resume

Input Blocking

Add to **ALL** game scripts with Update():

csharp

```
void Update()
{
    if (MainMenuManager.IsGamePaused())
        return;

    // ... rest of update
}
```

Required Files:

- PlayerPartyController.cs ✓
- ActionButton.cs ✓
- CharacterButton.cs ✓
- ActionResultButton.cs ✓
- PrisonerTeachingUI.cs ✓
- GameUIManager.cs ✓
- CombatManager.cs ✓

🎵 Volume System

csharp

Range: 0.0 - 1.0

Display: 0% - 100%

Saved: PlayerPrefs "MasterVolume"

Applied: AudioListener.volume

📝 Credits File

Location: Assets/Resources/Credits.txt

Format: Plain text

Type: TextAsset

Loaded: Automatically on menu init

🚀 Scene Loading

csharp

Mode: LoadSceneMode.Additive
Result: Both scenes loaded
Active: Game scene
Menu: Persistent (DontDestroyOnLoad)

II Pause System

Escape Key:
First press → Pause (show menu)
Second press → Resume (hide menu)

Paused State:
Time.timeScale = 0
Input disabled
Menu visible

Running State:
Time.timeScale = 1
Input enabled
Menu hidden

🌐 WebGL Handling

```
csharp
if (Application.platform == RuntimePlatform.WebGLPlayer)
{
    exitButton.gameObject.SetActive(false);
}
```

🐛 Debug Commands

```
csharp
// Check state
Debug.Log($"Paused: {MainMenuManager.IsGamePaused()}");
Debug.Log($"Input: {MainMenuManager.IsInputEnabled()}");
Debug.Log($"Time: {Time.timeScale}");

// Force resume
Time.timeScale = 1f;
```

Testing Flow

1. Play → Main menu appears ✓
2. Click "Start" → Game loads ✓
3. Button shows "Loading..." ✓
4. Menu hides automatically ✓
5. Can move player ✓
6. Press Escape → Menu appears ✓
7. Can't move player ✓
8. Button shows "Continue" ✓
9. Click "Continue" → Menu hides ✓
10. Can move again ✓
11. Options → Slider works ✓
12. Credits → Text displays ✓
13. Exit → Game quits ✓

Quick Styling

csharp

// Button size

Width: 300px

Height: 60px

// Font size

Title: 72pt

Buttons: 24pt

Body: 18pt

// Colors

Background: Dark (#202020)

Buttons: Medium (#404040)

Text: Light (#E0E0E0)

Highlight: Bright (#FFFFFF)

Common Issues Fix

Problem	Solution
No pause	Check Escape key not blocked
Input works	Add IsGamePaused() check
No load	Scene in Build Settings
No sound	Check AudioListener.volume
Exit shows	Check WebGL detection

State Diagram

```
[Main Menu]  
    ↓ Start  
[Loading...]  
    ↓  
[Game Running] ↔ [Game Paused]  
    ↑ Escape   Escape ↓
```

Saved Data

PlayerPrefs:

- "MasterVolume" → float (0-1)
- "InputEnabled" → int (0/1)

Feature Checklist

- Start/Continue button
- Loading state
- Additive scene loading
- Menu hide/show
- Pause on Escape
- Input blocking
- Options panel
- Volume slider
- Volume save/load
- Credits panel
- Credits from file
- Exit button
- WebGL exit hide
- Time pause/resume
- DontDestroyOnLoad

Menu Ready 🎮 - Professional game entry and pause system!