

# Main Menu System - Setup Guide

## 🎮 Overview

The main menu system provides a professional game entry point with:

- **Start/Continue** - Loads game additively, pauses/resumes
- **Options** - Volume control
- **Credits** - Display from text file
- **Exit** - Quit game (hidden on WebGL)
- **Escape key** - Pause/resume during gameplay

## 📁 File Structure

```
Assets/
└── Scenes/
    ├── MainMenu.unity (NEW)
    └── GameScene.unity (your existing scene)
└── Scripts/
    └── MainMenuManager.cs (NEW)
└── Resources/
    └── Credits.txt (NEW)
└── UI/
    └── MainMenuCanvas (prefab - optional)
```

## 🛠️ Setup Instructions

### Step 1: Create Main Menu Scene

1. File → New Scene
2. Name it "MainMenu"
3. Save to Assets/Scenes/MainMenu.unity

### Step 2: Create UI Canvas

Hierarchy:

Canvas

```
└── MainMenuPanel
    ├── Title (TextMeshPro)
    ├── StartContinueButton
    │   └── Text (TextMeshPro)
    └── OptionsButton
```

```
|   |   └── Text (TextMeshPro)
|   ├── CreditsButton
|   |   └── Text (TextMeshPro)
|   └── ExitButton
|       └── Text (TextMeshPro)
├── OptionsPanel (inactive by default)
|   ├── Title (TextMeshPro)
|   ├── VolumeLabel (TextMeshPro)
|   ├── VolumeSlider
|   ├── VolumeValueText (TextMeshPro)
|   └── BackButton
|       └── Text (TextMeshPro)
└── CreditsPanel (inactive by default)
    ├── CreditsText (TextMeshPro - scrollable)
    └── BackButton
        └── Text (TextMeshPro)
```

### Step 3: Configure Canvas

yaml

#### Canvas:

**Render Mode:** Screen Space - Overlay

**Canvas Scaler:**

**UI Scale Mode:** Scale With Screen Size

**Reference Resolution:** 1920×1080

**Match:** 0.5 (Width/Height)

### Step 4: Create MainMenuManager GameObject

1. Create empty GameObject: "MainMenuManager"
2. Add MainMenuManager script
3. Tag as "DontDestroyOnLoad" (optional)

### Step 5: Assign References

In MainMenuManager inspector:

yaml

#### Menu Panels:

Main Menu Panel: [Drag MainMenuPanel]

Options Panel: [Drag OptionsPanel]

Credits Panel: [Drag CreditsPanel]

#### Main Menu Buttons:

Start Continue Button: [Drag button]

Options Button: [Drag button]

Credits Button: [Drag button]

Exit Button: [Drag button]

#### Button Text:

Start Continue Button Text: [Drag TextMeshPro component]

#### Options UI:

Volume Slider: [Drag slider]

Volume Value Text: [Drag text]

Options Back Button: [Drag button]

#### Credits UI:

Credits Text: [Drag scrollable text]

Credits Back Button: [Drag button]

Credits File: [Drag Credits.txt from Resources]

#### Settings:

Game Scene Name: "GameScene" (or your scene name)

### Step 6: Create Credits File

1. Create Assets/Resources/ folder
2. Create new text file: Credits.txt
3. Use provided template or write your own
4. Assign to MainMenuManager

### Step 7: Configure Build Settings

File → Build Settings

#### Scenes in Build:

0. MainMenu
1. GameScene (your existing scene)

✓ Main Menu MUST be scene 0 (first)

### Step 8: Add Input Blocking

Add this line to the **beginning** of `Update()` in these scripts:

csharp

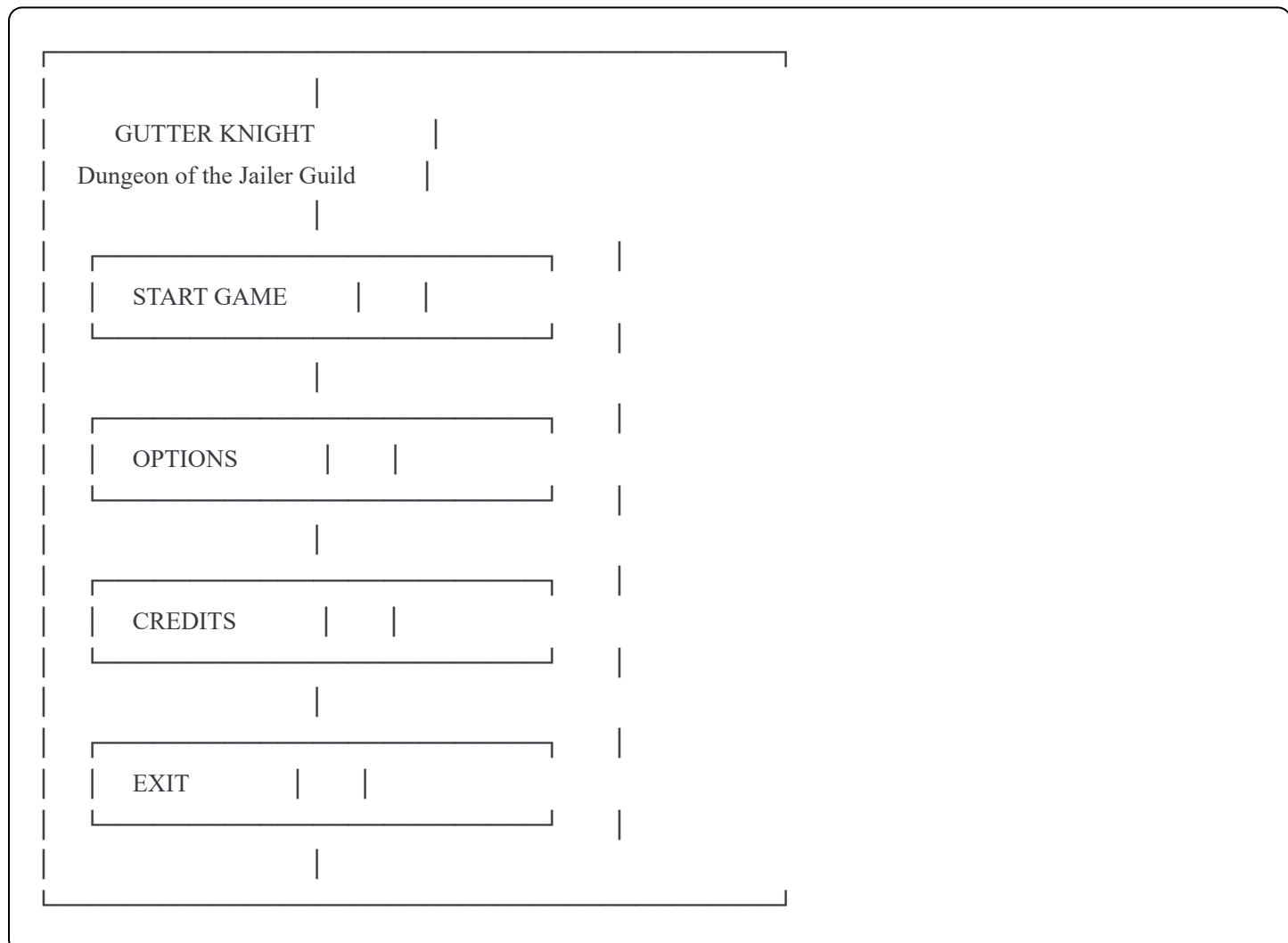
```
if (MainMenuManager.IsGamePaused())
    return;
```

### Files to modify:

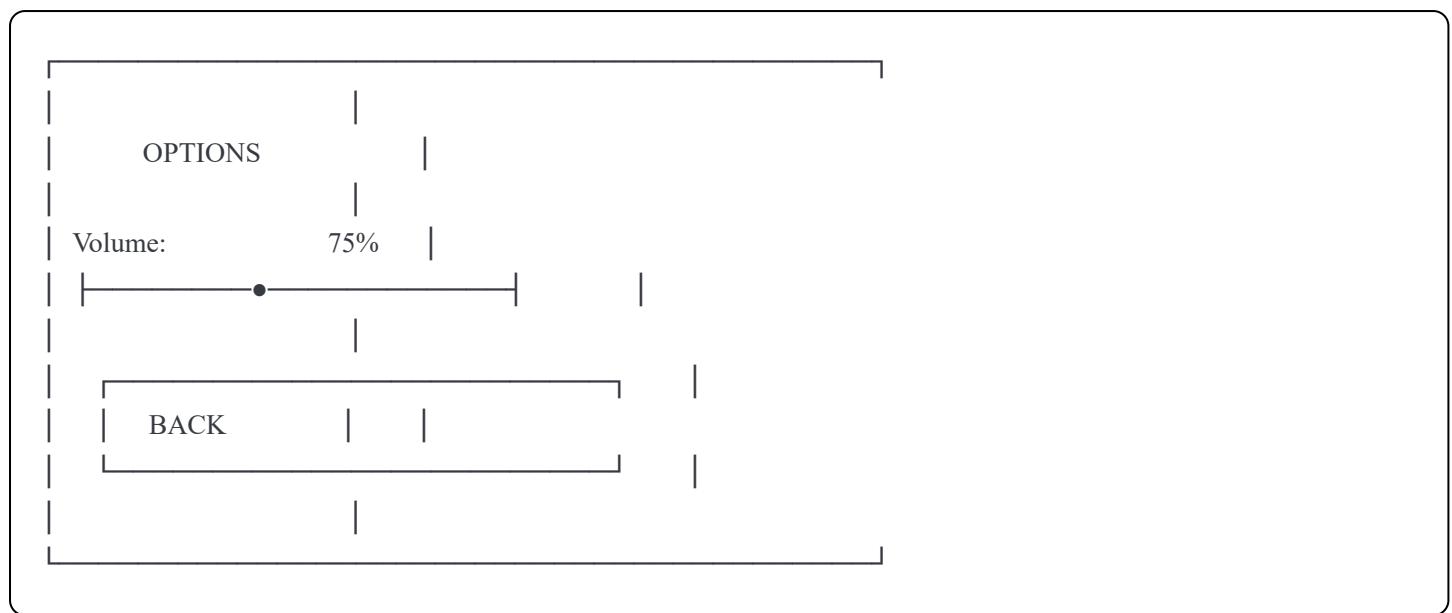
- PlayerPartyController.cs
- ActionButton.cs
- CharacterButton.cs
- ActionResultButton.cs
- PrisonerTeachingUI.cs
- GameUIManager.cs
- CombatManager.cs

## 🎨 UI Layout Example

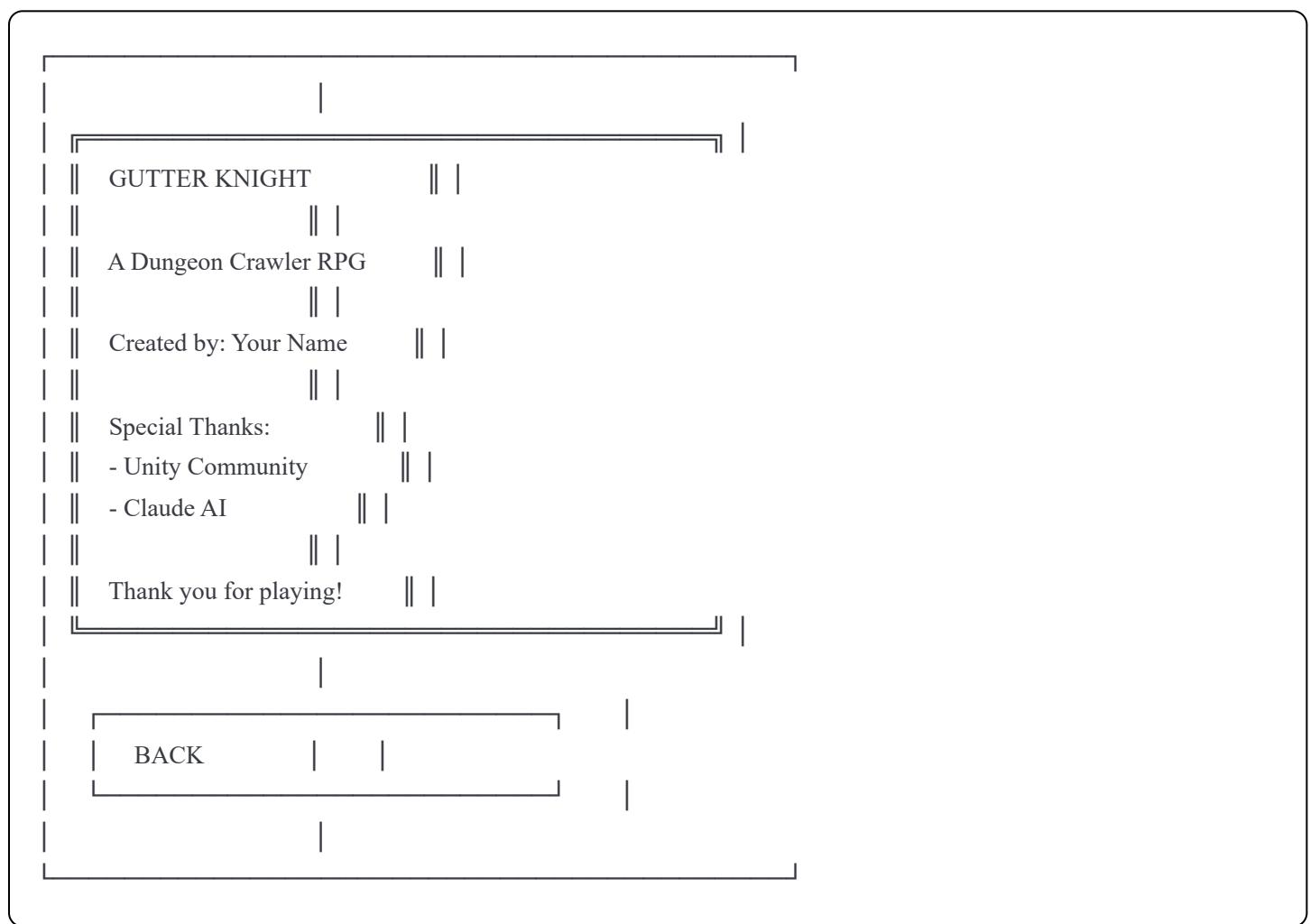
### Main Menu Panel



## Options Panel



## Credits Panel



## 🎯 Button Behavior

### Start/Continue Button

Initial State:

Text: "START GAME"

Action: Load game scene additively

While Loading:

Text: "LOADING..."

Disabled: Yes

After Loading:

Text: "CONTINUE GAME"

Action: Hide menu, resume game

When Paused:

Text: "CONTINUE GAME"

Action: Hide menu, resume game

## Options Button

Action:

1. Hide main menu panel
2. Show options panel
3. Load saved volume (default: 100%)

## Credits Button

Action:

1. Hide main menu panel
2. Show credits panel
3. Display credits from file

## Exit Button

Standalone/Editor:

Visible: Yes

Action: Quit application

WebGL:

Visible: No (auto-hidden)

Reason: Can't quit browser

## Input Handling

### Escape Key Behavior

In Main Menu:

→ Does nothing

In Game (first time):

- Pause game
- Show menu
- Disable player input
- Time.timeScale = 0

In Game (menu open):

- Resume game
- Hide menu
- Enable player input
- Time.timeScale = 1

## Blocked Input When Paused

All these inputs are blocked:

- ✗ WASD (movement)
- ✗ QE (turning)
- ✗ 1-9 (selections)
- ✗ Space (confirmations)
- ✗ Mouse clicks on game UI

Menu navigation still works:

- ✓ Mouse clicks on menu buttons
- ✓ Escape key (pause/resume)

## 🎵 Audio System

### Volume Control

csharp

```
// Slider range: 0.0 to 1.0
AudioListener.volume = sliderValue;

// Saved to PlayerPrefs
PlayerPrefs.SetFloat("MasterVolume", value);

// Loaded on startup
float savedVolume = PlayerPrefs.GetFloat("MasterVolume", 1f);
```

### Display Format

Slider: 0.0 - 1.0

Display: 0% - 100%

Conversion: Mathf.RoundToInt(value \* 100) + "%"

## 🔧 Advanced Features

### Persistent Menu

```
csharp

void Awake()
{
    DontDestroyOnLoad(gameObject);
}

// Menu survives scene transitions
// Only one instance exists
```

### Additive Scene Loading

```
csharp

// Load game scene WITHOUT unloading menu
SceneManager.LoadSceneAsync(gameSceneName, LoadSceneMode.Additive);
```

Benefits:

- Menu ready to show instantly
- Faster pause/resume
- No scene reload needed

### Time Scaling

```
csharp

Paused: Time.timeScale = 0f // Everything stops
Running: Time.timeScale = 1f // Normal speed
```

Affected:

- Physics
- Animations
- Particle systems
- Time.deltaTime

Not Affected:

- UI animations (use Time.unscaledDeltaTime)
- Audio
- Input

# Debugging

## Check Menu State

csharp

```
if (Input.GetKeyDown(KeyCode.F12))
{
    Debug.Log($"Game Loaded: {isGameLoaded}");
    Debug.Log($"Game Paused: {isGamePaused}");
    Debug.Log($"Time Scale: {Time.timeScale}");
    Debug.Log($"Input Enabled: {MainMenuManager.IsEnabled()}");
}
```

## Force Resume

csharp

```
[ContextMenu("Force Resume")]
void ForceResume()
{
    Time.timeScale = 1f;
    isGamePaused = false;
    HideMenu();
}
```

## Test Loading

csharp

```
[ContextMenu("Test Load Game")]
void TestLoadGame()
{
    StartCoroutine(LoadGameScene());
}
```

## Common Issues

### Menu doesn't appear on Escape

Check:

- MainMenuManager exists in scene
- mainMenuPanel reference assigned
- Update() is running
- Escape key not blocked elsewhere

## **Input still works when paused**

Check:

- MainMenuManager.IsGamePaused() called in Update()
- Added to ALL input scripts
- Return statement executes

## **Scene doesn't load**

Check:

- Scene name matches exactly
- Scene in Build Settings
- LoadSceneMode.Additive used

## **Volume doesn't save**

Check:

- PlayerPrefs.Save() called
- Slider onValueChanged connected
- AudioListener.volume set

## **Exit button visible on WebGL**

Check:

- Platform detection in Start()
- exitButton.gameObject.SetActive(false)
- Build platform set to WebGL

## **Platform-Specific**

### **Standalone (Windows/Mac/Linux)**

All features enabled:

- ✓ Start/Continue
- ✓ Options
- ✓ Credits
- ✓ Exit
- ✓ Escape key

### **WebGL**

Exit button hidden:

- ✓ Start/Continue
- ✓ Options

- ✓ Credits
- ✗ Exit (auto-hidden)
- ✓ Escape key

## Mobile (Future)

Considerations:

- Touch controls instead of keys
- Back button = Escape
- Larger buttons
- Different layout

## ✓ Testing Checklist

- Main menu scene loads first
- Start button loads game
- Button text changes to "Loading..."
- Game scene loads additively
- Menu hides when game starts
- Escape shows menu
- Continue button resumes game
- Player can't move when paused
- Number keys blocked when paused
- Options panel opens
- Volume slider works
- Volume saves/loads
- Volume display updates
- Credits panel opens
- Credits load from file
- Credits display correctly
- Back buttons work
- Exit button works (standalone)
- Exit button hidden (WebGL)
- Time pauses correctly
- Time resumes correctly
- No errors in console

## 🎨 Customization

### Change Colors

csharp

```
// Button normal color
button.colors = new ColorBlock
{
    normalColor = new Color(0.2f, 0.2f, 0.2f),
    highlightedColor = new Color(0.3f, 0.3f, 0.3f),
    pressedColor = new Color(0.1f, 0.1f, 0.1f),
    selectedColor = new Color(0.25f, 0.25f, 0.25f),
    disabledColor = new Color(0.5f, 0.5f, 0.5f),
    colorMultiplier = 1f,
    fadeDuration = 0.1f
};
```

## Add Background Music

```
csharp

[Header("Audio")]
public AudioSource menuMusic;

void Start()
{
    if (menuMusic != null)
        menuMusic.Play();
}

void OnGameStarted()
{
    if (menuMusic != null)
        menuMusic.Stop();
}
```

## Add Animations

```
csharp
```

```
[Header("Animation")]
public Animator menuAnimator;

void ShowMenu()
{
    menuAnimator.SetTrigger("FadeIn");
}

void HideMenu()
{
    menuAnimator.SetTrigger("FadeOut");
}
```

---

**Professional Entry Point** 🎮 - A complete main menu system with pause functionality!