Вот пример реализации enum для описания состояний игры на C#:

csharp

Copy

Download

public enum GameState

{

*// Вступительная сцена (интро)*

Intro,

*// Главное меню (постоянная сцена), игра еще не загружена*

MainMenu,

*// Игра загружена и работает (не на паузе)*

Gameplay,

*// Игра загружена и на паузе (например, через кнопку паузы в игре)*

GamePaused,

*// Игра загружена, но открыто меню через ESC (автоматическая пауза)*

*// При возврате из этого состояния игра продолжается*

InGameMenuWithAutoPause,

*// Игра загружена, была на паузе, и открыто меню*

*// При возврате из этого состояния игра остается на паузе*

InGameMenuFromPaused

}

Дополнительно можно добавить методы расширения для удобной работы с состояниями:

csharp

Copy

Download

public static class GameStateExtensions

{

public static bool IsGameLoaded(this GameState state)

{

return state != GameState.Intro && state != GameState.MainMenu;

}

public static bool IsPaused(this GameState state)

{

return state == GameState.GamePaused ||

state == GameState.InGameMenuWithAutoPause ||

state == GameState.InGameMenuFromPaused;

}

public static bool IsInMenu(this GameState state)

{

return state == GameState.MainMenu ||

state == GameState.InGameMenuWithAutoPause ||

state == GameState.InGameMenuFromPaused;

}

}

Такой подход позволяет четко разделить все возможные состояния игры и легко проверять текущий статус.

перепиши GameStateExtensions под GameManager.Instance ...

Если у тебя есть синглтон GameManager.Instance, который хранит текущее состояние игры (CurrentState), то можно переписать расширения для удобного доступа к проверкам состояния прямо из менеджера.

**1. Переписанный**GameStateExtensions**под**GameManager.Instance

csharp

Copy

Download

public static class GameManagerExtensions

{

*// Проверяет, загружена ли игра (не Intro и не MainMenu)*

public static bool IsGameLoaded(this GameManager gameManager)

{

return gameManager.CurrentState != GameState.Intro &&

gameManager.CurrentState != GameState.MainMenu;

}

*// Проверяет, находится ли игра на паузе*

public static bool IsPaused(this GameManager gameManager)

{

return gameManager.CurrentState == GameState.GamePaused ||

gameManager.CurrentState == GameState.InGameMenuWithAutoPause ||

gameManager.CurrentState == GameState.InGameMenuFromPaused;

}

*// Проверяет, находится ли игрок в меню (главном или внутриигровом)*

public static bool IsInMenu(this GameManager gameManager)

{

return gameManager.CurrentState == GameState.MainMenu ||

gameManager.CurrentState == GameState.InGameMenuWithAutoPause ||

gameManager.CurrentState == GameState.InGameMenuFromPaused;

}

*// Проверяет, активно ли геймплейное состояние (игра не на паузе и не в меню)*

public static bool IsActiveGameplay(this GameManager gameManager)

{

return gameManager.CurrentState == GameState.Gameplay;

}

}

**2. Пример использования в коде**

Теперь можно удобно проверять состояние игры через GameManager.Instance:

csharp

Copy

Download

if (GameManager.Instance.IsGameLoaded())

{

Debug.Log("Игра загружена!");

}

if (GameManager.Instance.IsPaused())

{

Debug.Log("Игра на паузе!");

}

if (GameManager.Instance.IsInMenu())

{

Debug.Log("Игрок в меню!");

}

if (GameManager.Instance.IsActiveGameplay())

{

Debug.Log("Игра активна, геймплей идет!");

}

**3. Дополнительно: управление паузой**

Можно добавить методы для удобного управления паузой прямо в GameManager:

csharp

Copy

Download

public class GameManager : MonoBehaviour

{

public static GameManager Instance { get; private set; }

public GameState CurrentState { get; private set; }

private void Awake()

{

if (Instance != null && Instance != this)

{

Destroy(gameObject);

return;

}

Instance = this;

DontDestroyOnLoad(gameObject);

}

*// Включает паузу (из геймплея или из меню)*

public void SetPause(bool paused)

{

if (paused)

{

if (CurrentState == GameState.Gameplay)

CurrentState = GameState.GamePaused;

}

else

{

if (CurrentState == GameState.GamePaused ||

CurrentState == GameState.InGameMenuWithAutoPause ||

CurrentState == GameState.InGameMenuFromPaused)

{

CurrentState = GameState.Gameplay;

}

}

}

*// Открывает меню паузы (с авто-паузой или без)*

public void OpenPauseMenu(bool autoPause)

{

if (CurrentState == GameState.Gameplay && autoPause)

CurrentState = GameState.InGameMenuWithAutoPause;

else if (CurrentState == GameState.GamePaused)

CurrentState = GameState.InGameMenuFromPaused;

}

}

**Итог**

Теперь у тебя есть:  
✅ Четкое разделение состояний через enum  
✅ Удобные методы проверки (IsPaused, IsInMenu, IsGameLoaded)  
✅ Управление паузой через GameManager.Instance.SetPause()

Если нужно что-то доработать — пиши! 🚀

как объявить CurrentState в GameManager?

В GameManager нужно объявить CurrentState как публичное свойство (или поле) типа GameState, чтобы другие скрипты могли его читать (а изменять — только через методы менеджера для контроля логики).

**1. Простейший вариант (публичное поле)**

csharp

Copy

Download

public class GameManager : MonoBehaviour

{

public static GameManager Instance { get; private set; }

*// Текущее состояние игры (можно читать извне, но менять только внутри GameManager)*

public GameState CurrentState { get; private set; } = GameState.Intro; *// Начинаем с интро*

private void Awake()

{

if (Instance != null && Instance != this)

{

Destroy(gameObject);

return;

}

Instance = this;

DontDestroyOnLoad(gameObject); *// Если менеджер должен быть на всех сценах*

}

}

**2. Вариант с событием при изменении состояния**

Если нужно уведомлять другие скрипты о смене состояния (например, UI или геймплейные системы):

csharp

Copy

Download

public class GameManager : MonoBehaviour

{

public static GameManager Instance { get; private set; }

*// Событие при изменении состояния (можно подписаться из других скриптов)*

public event Action<GameState> OnGameStateChanged;

private GameState \_currentState = GameState.Intro;

public GameState CurrentState

{

get => \_currentState;

private set

{

if (\_currentState != value)

{

\_currentState = value;

OnGameStateChanged?.Invoke(\_currentState); *// Уведомляем подписчиков*

}

}

}

private void Awake()

{

*// ... (инициализация синглтона как выше)*

}

*// Пример метода для изменения состояния*

public void ChangeState(GameState newState)

{

CurrentState = newState;

}

}

**3. Пример использования**

Другие скрипты могут:

* **Читать состояние**:

csharp

Copy

Download

if (GameManager.Instance.CurrentState == GameState.Gameplay) { ... }

* **Подписаться на изменения**:

csharp

Copy

Download

void Start()

{

GameManager.Instance.OnGameStateChanged += HandleGameStateChange;

}

void HandleGameStateChange(GameState newState)

{

Debug.Log($"Состояние изменилось на: {newState}");

}

**4. Важные нюансы**

* **Изменение состояния только через методы** (например, PauseGame(), ReturnToMenu()), чтобы избежать неконтролируемых изменений.
* **Инициализация состояния** в Awake() или Start() (например, загрузка сохраненного состояния).
* **Сброс состояния** при перезапуске игры:

csharp

Copy

Download

public void ResetGame()

{

CurrentState = GameState.MainMenu;

}

Если нужна более сложная логика (например, проверки перед изменением состояния) — можно добавить методы вроде:

csharp

Copy

Download

public bool TryPauseGame()

{

if (CurrentState != GameState.Gameplay) return false;

CurrentState = GameState.GamePaused;

return true;

}