Setting linear velocity of a kinematic body is not supported. UnityEngine.Rigidbody:set\_linearVelocity (UnityEngine.Vector3) DarkTreeFPS.ZombieNPC:OnAnimatorMove () (at Assets/DarkTree FPS/Scripts/NPC/ZombieNPC.cs:439)

**ChatGPT сказал:**

Сообщение:

less

КопироватьРедактировать

Setting linear velocity of a kinematic body is not supported.

UnityEngine.Rigidbody:set\_linearVelocity (UnityEngine.Vector3)

DarkTreeFPS.ZombieNPC:OnAnimatorMove () (at Assets/DarkTree FPS/Scripts/NPC/ZombieNPC.cs:439)

говорит о том, что **нельзя установить velocity (или linearVelocity) для Rigidbody, у которого установлен флаг isKinematic = true**.

**📌 Почему это происходит**

Компонент Rigidbody, у которого включен isKinematic, **не управляется физикой Unity**, и любые попытки установить velocity для него игнорируются и вызывают ошибку. А в строке 439 файла ZombieNPC.cs происходит именно это.

**✅ Возможные решения**

**🔧 Вариант 1: Выключить isKinematic**

Если вы хотите, чтобы физика управляла перемещением NPC, и вы задаёте velocity, нужно отключить isKinematic:

csharp

КопироватьРедактировать

// Где-то в Start() или Awake()

rigidbody.isKinematic = false;

⚠️ **Осторожно:** Это включит управление объектом через физику — столкновения, падения, отскоки.

**🔧 Вариант 2: Не использовать velocity для кинематических объектов**

Если NPC должен быть управляем только анимацией (и быть кинематическим), **не задавайте velocity напрямую**. Вместо этого перемещайте объект через transform.position, transform.Translate() или Rigidbody.MovePosition().

Пример замены кода:

csharp

КопироватьРедактировать

// Неправильно:

rigidbody.velocity = desiredVelocity; // Ошибка, если isKinematic = true

// Правильно:

transform.position += desiredVelocity \* Time.deltaTime;

ИЛИ:

csharp

КопироватьРедактировать

rigidbody.MovePosition(transform.position + desiredVelocity \* Time.deltaTime);

**🔧 Вариант 3: Анимировать движение через Root Motion**

Если ты используешь Animator и OnAnimatorMove() (что и видно из стека вызовов), правильный способ — двигать объект с помощью:

csharp

КопироватьРедактировать

void OnAnimatorMove()

{

if (animator && rigidbody && rigidbody.isKinematic)

{

rigidbody.MovePosition(rigidbody.position + animator.deltaPosition);

rigidbody.MoveRotation(animator.rootRotation);

}

}

**💬 Что делать дальше?**

Если хочешь, могу:

* помочь найти и переписать строку 439 в ZombieNPC.cs
* адаптировать поведение под физическое или анимированное движение
* объяснить, как правильно сочетать AI, анимации и перемещения в Unity

Хочешь, посмотрим вместе содержимое OnAnimatorMove() в этом скрипте?

Scene persistence wasn't loaded due to condition mismatch UnityEngine.MonoBehaviour:print (object) DTInventory.LevelPoint:Start () (at Assets/DarkTree FPS/Scripts/DT Inventory/Level Transition/LevelPoint.cs:29)

**ChatGPT сказал:**

Сообщение:

vbnet

КопироватьРедактировать

Scene persistence wasn't loaded due to condition mismatch

UnityEngine.MonoBehaviour:print (object)

DTInventory.LevelPoint:Start () (at Assets/DarkTree FPS/Scripts/DT Inventory/Level Transition/LevelPoint.cs:29)

означает, что **переход между сценами или восстановление состояния сцены** не произошло, потому что **условие для загрузки не было выполнено**.

**📌 Что это значит**

Скрипт LevelPoint.cs — это часть системы переходов между уровнями. Судя по названию, он:

* определяет точку перехода между сценами (наподобие двери или зоны перехода);
* при старте проверяет, нужно ли загрузить сохранённое состояние сцены (позицию игрока, предметы и т. д.);
* если условие не выполнено — пишет print("Scene persistence wasn't loaded due to condition mismatch").

**📄 Строка 29 LevelPoint.cs**

Скорее всего, в этой строке примерно такой код:

csharp

КопироватьРедактировать

if (/\* какое-то условие \*/)

{

LoadSceneState();

}

else

{

print("Scene persistence wasn't loaded due to condition mismatch");

}

**🧠 Возможные причины**

1. **Нет сохранённых данных сцены.**  
   Например, ты перешёл на уровень напрямую из редактора без предварительного сохранения прогресса — и нечего загружать.
2. **Неверный идентификатор сцены.**  
   Система может ожидать, что игрок пришёл с другой сцены и должен быть восстановлен в определённой точке, но этого не произошло.
3. **Неверный переход или загрузка.**  
   Переход в сцену был не через систему LevelPoint, а вручную — и условие проверки сцены не совпало.
4. **PlayerPrefs / JSON / файл с данными пуст или повреждён.**  
   Если сохранение сделано, например, в PlayerPrefs или .json — и файл не найден или не соответствует формату — условие может не выполниться.

**✅ Что можно сделать**

**📍 1. Проверить условие в LevelPoint.cs**

Открой скрипт и посмотри, что за условие проверяется в Start() примерно на строке 29. Мы можем его адаптировать или отключить логику, если тебе не нужна загрузка.

Хочешь — пришли сюда этот фрагмент, и я подскажу.

**🧪 2. Запускать сцену из правильного места**

Если ты запускаешь уровень **напрямую из редактора**, он может не сработать как при настоящем переходе из игры. Попробуй:

* Запускать игру с начальной сцены (главного меню);
* Убедиться, что используется LevelManager или похожий скрипт, отвечающий за переход.

**🛠 3. Отключить предупреждение (временно)**

Если оно не критично, можешь просто убрать print(...), чтобы не мешал в консоли. Но это не решит причину.

**💬 Хочешь точную помощь?**

Просто пришли сюда фрагмент Start() из LevelPoint.cs, особенно строчки 20–35. Я подскажу, как адаптировать условие или исправить логику сохранения/загрузки.