

Кооперативный проект. Документация.

С чего начать?

Скачайте готовый проект с репозитория:

<https://github.com/Pacelin/CooperativeProject/tree/main>

В проекте есть 4 папки:

- *Cooperative*
- *Example*
- *ScriptTemplates*
- *Your Assets*

Папка *Cooperative* содержит файлы проекта (не изменяйте её, это может привести к непредвиденным последствиям).

Папка *Example* служит для примера, вы можете посмотреть в ней примеры работы в проекте.

Папка *ScriptTemplates* служит для замены стандартных шаблонов скриптов.

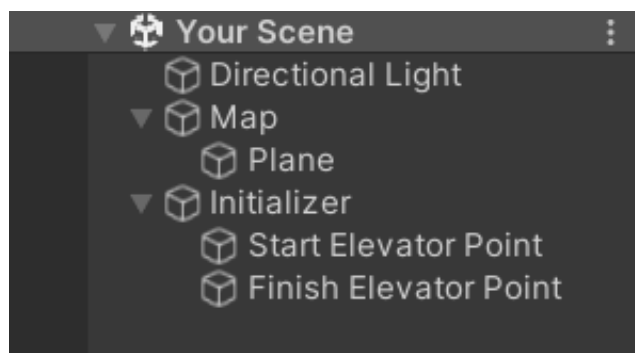
Ваша работа будет производиться в папке *Your Assets*. Внутри этой папки есть сцена, в которой вы будете работать.

Переименуйте папку и файл сцены. В названии можно указать ваш никнейм.

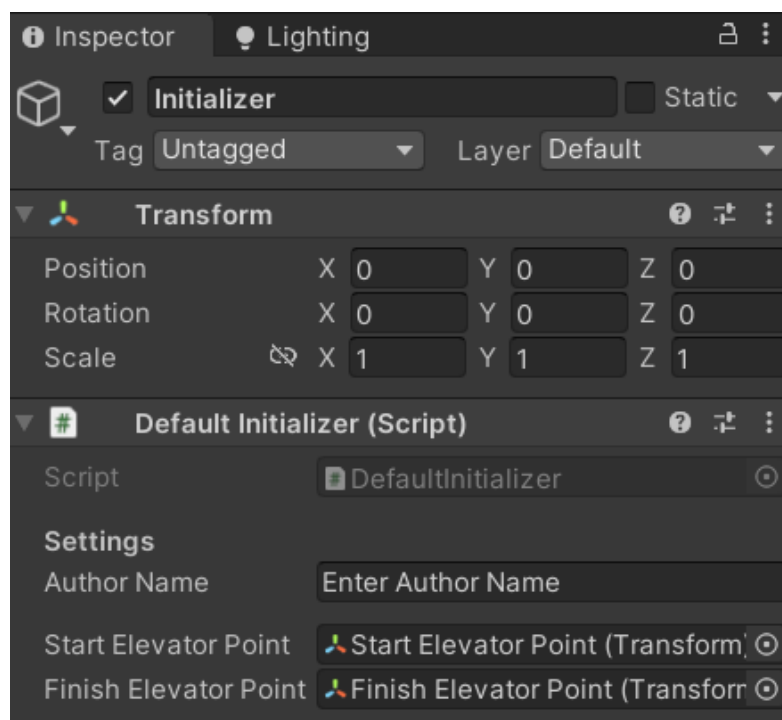
Начальная сцена

На начальной сцене размещены следующие объекты:

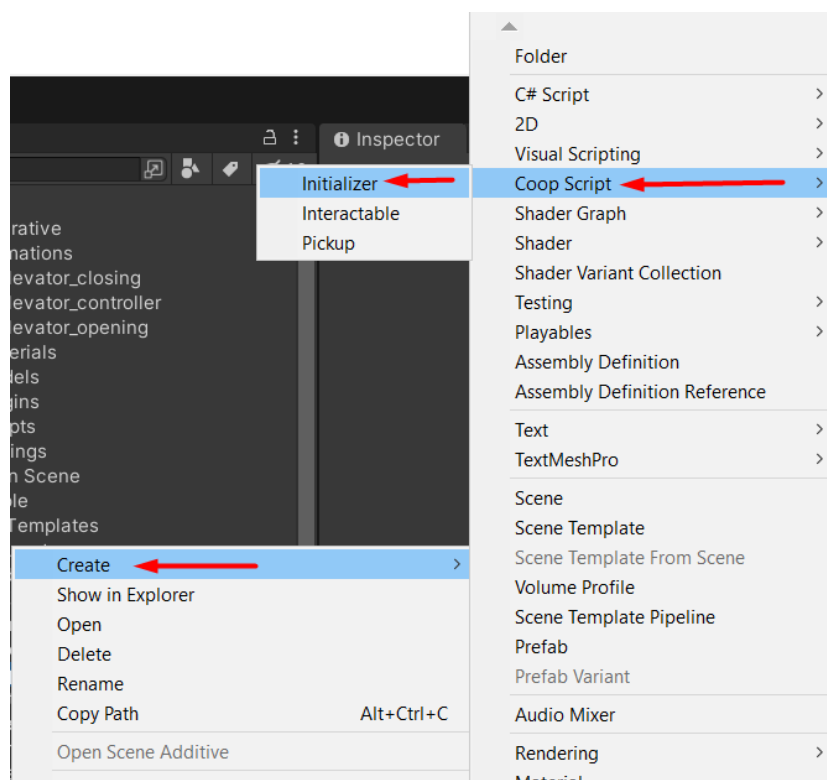
- Объект освещения (*Directional Light*). Служит в качестве примера освещения.
- Объект карты (*Map*) с плоскостью (*Plane*). Служит в качестве примера уровня.
- Объект инициализации (*Initializer*). Предназначен для запуска вашей сцены в кооперативном проекте.
- Вход и выход со сцены (*Start Elevator Point* и *Finish Elevator Point*). Это точки, где появляются лифты для входа и выхода со сцены.



К объекту инициализации прикреплен стандартный скрипт инициализации.



Вы можете написать свой собственный инициализатор, если откроете соответствующее меню.



Не забудьте указать автора в параметрах инициализатора в инспекторе.

Работа с игроком

Для какой-либо работы с игроком, вы можете обратиться к статическим полям класса `Player`.

Все изменения, которые вы выполняете с любыми из статических полей, необходимо сбрасывать до начальных значений при выходе с сцены! Для этого реализуйте свой инициализатор сцены.

Работа с объектами

В проекте уже реализованы классы, которые отвечают за взаимодействие с объектами и подбор предметов.

Для их реализации используйте классы `Pickup` и `Interactable`. Создать их можно в соответствующем меню.

Также реализован класс, который описывает объект, в который можно положить какой-либо предмет. Для его создания прикрепите к объекту скрипт `PickupPlace`.

Создавая свои скрипты, обязательно указывайте личный ***namespace***, в качестве названия можете использовать свой никнейм.

Техническая документация

| Player | |
|---|---|
| Статические поля | |
| <code>bool</code> <code>IsInitialized</code> | Поле служит для работы проекта. |
| <code>Camera</code> <code>PlayerCamera</code> | Содержит ссылку на основную камеру сцены. |
| <code>Camera</code> <code>OverlayCamera</code> | Содержит ссылку на дополнительную камеру. Служит для работы подбираемых объектов. Не изменяйте без крайней необходимости. |
| <code>FirstPersonController</code> <code>FPSController</code> | Содержит ссылку на контроллер игрока, |

| | |
|--------------------------------|---|
| | отвечающий за передвижение. |
| Interactor Interactor | Содержит ссылку на объект, отвечающий за взаимодействие с предметами. |
| HandInventory Inventory | Содержит ссылку на инвентарь игрока. |

Interactor

Поля

| | |
|--------------------------------------|--|
| Inventory Inventory | Содержит ссылку на инвентарь, в который будут помещаться подбираемые объекты. |
| InteractorRay RayPoint | Содержит ссылку на объект, который отслеживает изменение направления взгляда игрока. |
| InteractorCrosshair Crosshair | Содержит ссылку на объект, отвечающий за отображение прицела. |

InteractorRay

Поля

| | |
|---|---|
| event Action<GameObject> OnLookedObjectChanged | Событие. Вызывается, когда меняется объект, на который смотрит игрок. |
|---|---|

| | |
|------------------------------------|--|
| bool Enabled | Отслеживается взгляд игрока на данный момент или нет. |
| GameObject LookAtGameObject | Ссылка на объект, на который смотрит игрок. |
| float RayDistance | Дистанция, в пределах которой отслеживается взгляд игрока. |

Методы

| | |
|-----------------------|--|
| void Enable() | Включает отслеживание взгляда игрока. |
| void Disable() | Выключает отслеживание взгляда игрока. |

InteractorCrosshair

Поля

| | |
|---------------------|-------------------------|
| bool Visible | Виден ли сейчас прицел. |
| Color Color | Цвет прицела. |

Методы

| | |
|--|---|
| void Show() | Показывает прицел. |
| void Hide() | Прячет прицел. |
| void SetDefault() | Устанавливает все значения прицела в стандартные. |
| void Set(Color color, float scale) | Устанавливает цвет прицела в color , а размер в scale . |
| void SetSprite(Sprite sprite) | Устанавливает спрайт прицела в sprite . |

| HandInventory | |
|---|--|
| Поля | |
| <code>bool</code> IsFull | Инвентарь полон. |
| <code>bool</code> IsEmpty | Инвентарь пуст. |
| <code>Vector3</code> HandPosition | Позиция руки игрока. |
| <code>Vector3</code> HandDirection | Направление руки игрока. |
| <code>Pickup</code> HodedPickup | Ссылка на предмет, который держит игрока. |
| Методы | |
| <code>void</code> AddPickup(<code>Pickup</code> pickup) | Добавить предмет <code>pickup</code> в инвентарь. |
| <code>void</code> RemovePickup(<code>Pickup</code> pickup) | Убрать предмет <code>pickup</code> из инвентаря. |
| <code>void</code> RemovePickup(<code>int</code> index) | Убрать предмет по индексу <code>index</code> из инвентаря. |
| <code>Pickup</code> GetPickup(<code>int</code> index) | Вовзращает ссылку на предмет по индексу <code>index</code> . |
| <code>bool</code> ContainsPickup(<code>Pickup</code> pickup) | Возвращает, содержится ли предмет <code>pickup</code> в инвентаре. |
| <code>void</code> Clear() | Очищает все ячейки инвентаря. |
| Interactable | |
| Поля | |
| <code>bool</code> CanInteract | Можно ли взаимодействовать с объектом. |

| | |
|--|--|
| <code>bool HideInteraction</code> | Показывать ли игроку, что с объектом можно взаимодействовать. |
| <code>protected Color _interactorColor</code> | Цвет прицела при наведении на объект. |
| <code>protected float _interactorScale</code> | Размер прицела при наведении на объект. |
| <code>protected Collider _selfCollider</code> | Ссылка на коллайдер объекта. |
| Методы | |
| <code>virtual void OnInteractorEnter(Interactor interactor)</code> | Вызывается, когда прицел входит в область объекта. |
| <code>virtual void OnInteractorExit(Interactor interactor)</code> | Вызывается, когда прицел выходит из области объекта. |
| <code>abstract void OnTryInteract(Interactor interactor)</code> | Вызывается, когда игрок пытается взаимодействовать с объектом при <code>CanInteract = false</code> . |
| <code>abstract void OnInteractDown(Interactor interactor)</code> | Вызывается, когда игрок нажимает кнопку взаимодействия, смотря на объект. |
| <code>abstract void OnInteractUp(Interactor interactor)</code> | <p>Вызывается, когда игрок отпускает кнопку взаимодействия, смотря на объект.</p> <p>Не вызывается, если после нажатия кнопки прицел вышел из области объекта.</p> |

Pickup : Interactable

Поля

`protected Rigidbody _selfRigidbody`

Ссылка на `Rigidbody` объекта.

Методы

`virtual void OnTake(Inventory inventory)`

Вызывается, когда игрок подбирает предмет.

`virtual void OnDrop(Inventory inventory)`

Вызывается, когда игрок выбрасывает предмет.

PickupPlace : Interactable

Поля

`event Action<Pickup> OnPlacePickup`

Событие. Вызывается, когда игрок кладет предмет на место.

`event Action<Pickup> OnTakePickup`

Событие. Вызывается, когда игрок берет предмет из места.

`bool CanPlace`

Можно ли положить предмет.

`bool CanTake`

Можно ли взять предмет.

`bool IsEmpty`

Предмет не лежит на месте.

`protected Pickup[] _placeablePickups`

Массив предметов, которые можно положить на место.

`protected Pickup _placedPickup`

Ссылка на предмет, который лежит на месте.