Keys, doors and chests

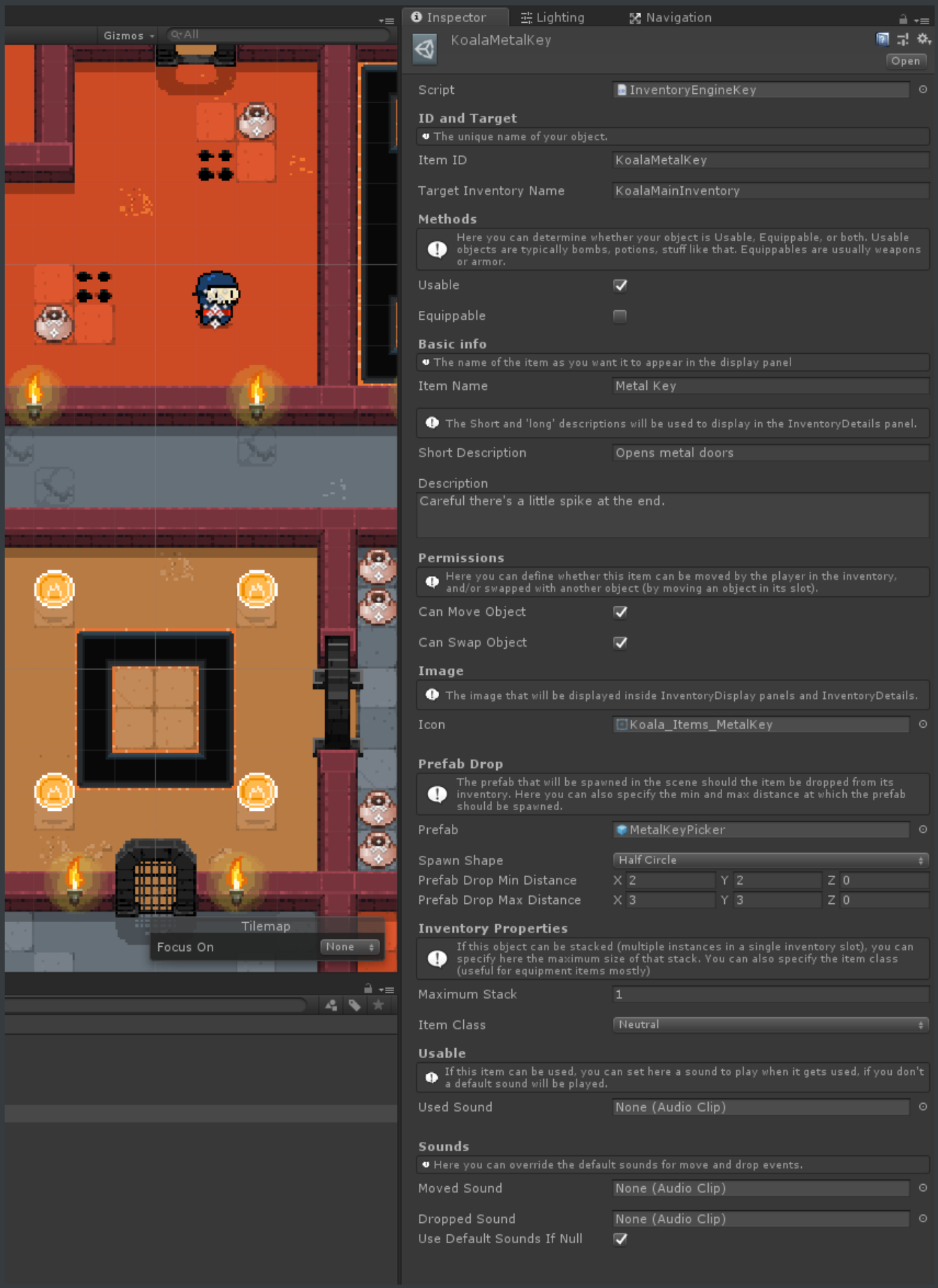
This page describes how to use keys in the TopDown Engine to unlock doors, open chests, get loot from them, and interact with stuff in general.

Introduction

Thanks to the Inventory Engine, the TopDown Engine natively supports the creation, collection and use of keys. Keys are inventory items that, combined with a few components on your scene elements, will allow you to interact with doors, open chests, loot their contents, and generally activate stuff in your scenes. If you want examples in action, you can check (among others) the MinimalSandbox2D demo scene, located in Demos/Minimal2D, or the KoalaDungeon scene, they'll provide working examples of key activated doors and chests.

Creating a key

If you're familiar with the Inventory Engine, this will be easy. If not, you should probably check out its documentation. To create a key, simply right click in one of your Resources folder, then go Create > More Mountains > TopDown Engine > InventoryEngineKey. Give this newly created scriptable object instance a name, and you're ready to fill its details in its inspector.


Gizmos
All
Tilemap
Focus On
None
Inspector
Lighting
Navigation
KoalaMetalKey
Open
Script
InventoryEngineKey
ID and Target
The unique name of your object.
Item ID
KoalaMetalKey
Target Inventory Name
KoalaMainInventory
Methods
Here you can determine whether your object is Usable, Equippable, or both. Usable objects are typically bombs, potions, stuff like that. Equippables are usually weapons or armor.
Usable
Equippable
Basic info
The name of the item as you want it to appear in the display panel
Item Name
Metal Key
The Short and 'long' descriptions will be used to display in the InventoryDetails panel.
Short Description
Opens metal doors
Description
Careful there's a little spike at the end.
Permissions
Here you can define whether this item can be moved by the player in the inventory, and/or swapped with another object (by moving an object in its slot).
Can Move Object
Can Swap Object
Image
The image that will be displayed inside InventoryDisplay panels and InventoryDetails.
Icon
Koala_Items_MetalKey
Prefab Drop
The prefab that will be spawned in the scene should the item be dropped from its inventory. Here you can also specify the min and max distance at which the prefab should be spawned.
Prefab
MetalKeyPicker
Spawn Shape
Half Circle
Prefab Drop Min Distance
X 2
Y 2
Z 0
Prefab Drop Max Distance
X 3
Y 3
Z 0
Inventory Properties
If this object can be stacked (multiple instances in a single inventory slot), you can specify here the maximum size of that stack. You can also specify the item class (useful for equipment items mostly)
Maximum Stack
1
Item Class
Neutral
Usable
If this item can be used, you can set here a sound to play when it gets used. if you don't a default sound will be played.
Used Sound
None (Audio Clip)
Sounds
Here you can override the default sounds for move and drop events.
Moved Sound
None (Audio Clip)
Dropped Sound
None (Audio Clip)
Use Default Sounds If Null

Creating a key

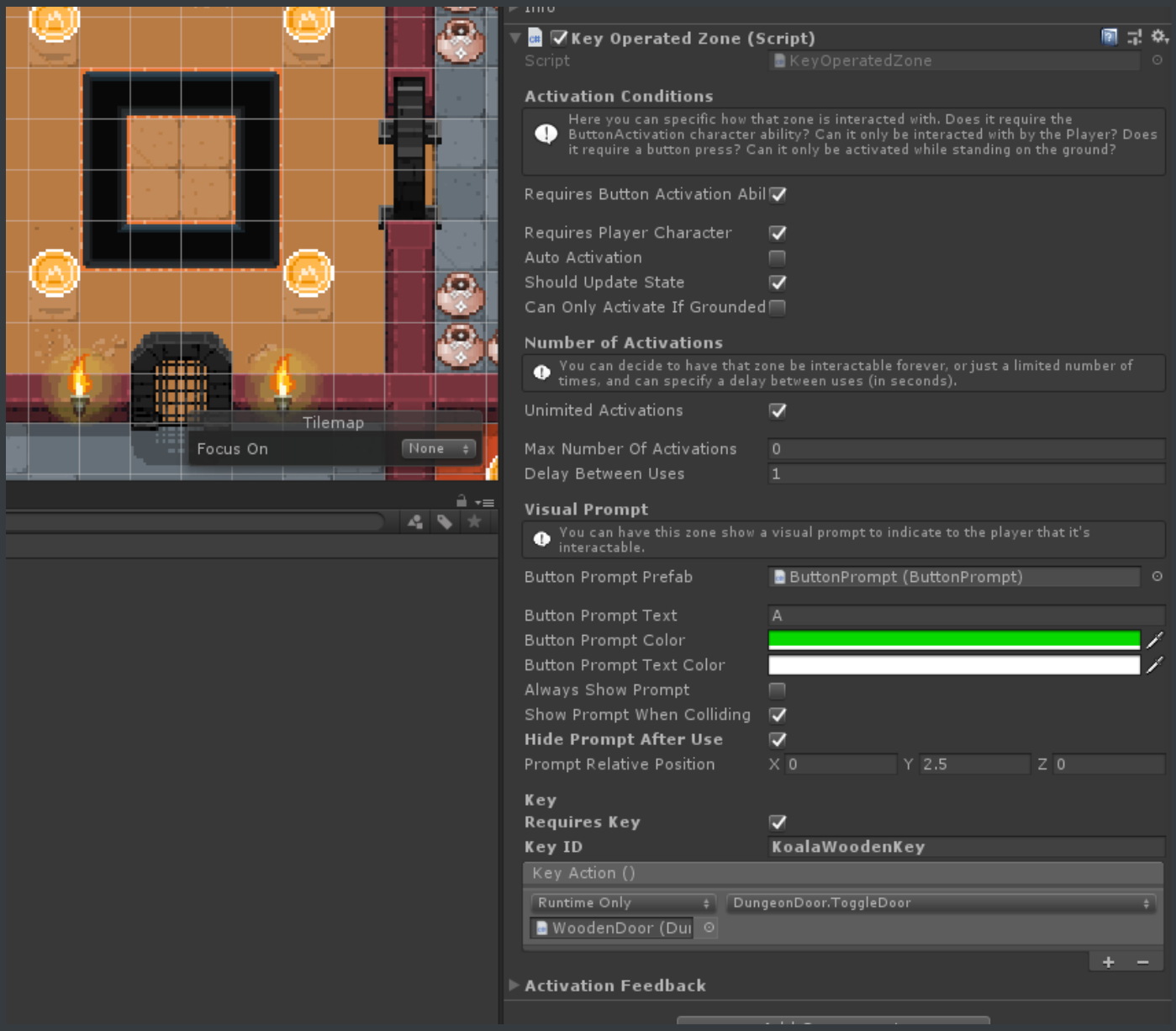
The key's inspector is the same as any Base Item you can create in the Inventory Engine, so we won't go over the details again here, but note that the Key's name (ItemID) is really important, as it's the string you'll use to say what key can open what door/chest/etc, so it needs to be unique.

Once you've created that scriptable object instance, all that's left to do is to create a pickable item for it. If you don't remember how that's done, check out this documentation.

Creating a door

One of the most common things you can open with a key is a door. So let's start with that. The most simple way to create an openable door in the TopDown Engine is to reuse either the 2D or 3D door prefabs. It could also be a moving platform, or a wall, or anything you want. The only mandatory element will be a **Key Operated Zone component**.

Key Operated Zones



An example of a Key Operated Zone

Once you've got your moving door/platform, you need a zone to interact with the key and open it. For that, create an empty game object, add a Box Collider 2D to it, set it to IsTrigger, resize it to your liking and position it somewhere between your key picker item and your door. Then, add a KeyOperatedZone component to it.

KeyOperatedZones are an extension of the ButtonActivated class, which means they'll combine the advantages and options of ButtonActivated elements (dialogue zones for example) and add key support to them.

From your zone's inspector, you can set, among other things, whether the zone is auto activated or not (in this case it'll require you to press a button). You can also specify a number of activations, and a delay between each use. If your zone is not auto activated, you can decide to have a button prompt show up always or only when when colliding, and where it should appear. Finally, you can decide that this zone requires a key, and in that case what KeyID the zone should look for in your character's inventory.

The last thing to do is to bind an action to this zone. This action is a method on any object in your scene that will be triggered when the zone is activated (when you enter it and have met the conditions you just defined to activate it : button pressed, key in inventory, etc). In the case of the 2D door, you'd have to drag the DungeonDoor component into our action field, and select its ToggleDoor method.

With this simple component, you can control doors, but really any object with a public method bindable to this Key Action field. You can have doors that open when you approach, or when you press a button, or that won't open without a key, etc.

Creating a chest

Creating a chest is just as simple as creating a door. Create a Key Operated Zone just like before, and then drop a sprite or 3D model into your scene for your chest. On that chest object, add an InventoryEngineChest component. Then all you have to do is add ItemPicker components to it (one per item you want in your chest). Then bind your Chest to your KeyOperatedZone's inspector just like before, using the InventoryEngineChest.OpenChest method this time. You're done! When activated, the InventoryEngineChest component will automatically pick all the ItemPickers you've added to it and put their items into your character's inventory. Additionnally, you can add an animator to it to trigger an open animation. Your animator will then need a trigger Animation Parameter called "OpenChest".

You can find a working Chest example in the KoalaDungeon demo scene.