Survival Template Pro

Welcome!

Official documentation/wiki for Survival Template Pro.

Overview

Welcome to the **official** STP Documentation/Wiki!

Survival Template Pro is a powerful toolkit designed to be as modular and scalable as possible, while also offering a lot of features found in modern survival games. It can be used as a starting place for your survival game or simply as a learning experience.

> Since this asset is still in an early stage, many systems could **change** drastically or get **removed** over the course of the next few months.

> **Good to know!** This documentation is constantly **updated**, if you're not finding what you're looking for please join our discord server and let me know. I'll add it as soon as possible.

> If you like **STP** and want to support its development, please consider reviewing it on the Unity Asset Store.

Links

> If this wiki doesn't cover what you were looking for or just want more insight into the asset, please don't hesitate to ask for support on either **Discord** or **E-Mail**.

- E-MAIL -> polymindgames@gmail.com
- DISCORD -> https://discord.com/invite/pkwPNEy
- ROADMAP -> https://trello.com/b/ecJxLwp2/survival-template-pro
- ASSETSTORE -> https://assetstore.unity.com/publishers/26298
- VIDEOS -> https://www.youtube.com/playlist?list=PLOSTcIvTSE-59t6IUpwv_RxyQYmneq4Vj
- DEMO -> https://drive.google.com/file/d/1QXi4B88UIB-RQ0L3oicgB1zsh3LDdBTl/view?usp=sharing

Getting Started

Installation Guide

If you're having problems with installing STP, this guide can help you.

Preparation

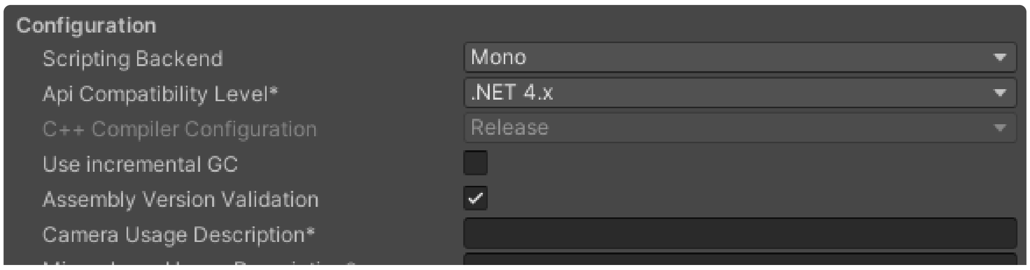
Considering STP is a template, make sure you're creating a new project for it instead of merging it with another project.

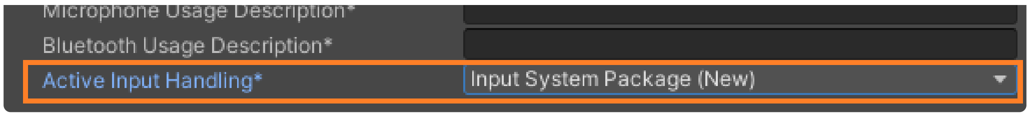
1. Make sure you're using Unity 2020.3 or newer.
2. Currently the only supported **Rendering Pipeline** is **Built-In**.
3. Make sure your storage at least 10gb left.

Installation

Recommended Steps:

1. Navigate to: "**Edit / Project Settings / Player / Configuration**" and set the Input Handling to **New** or **Both.**

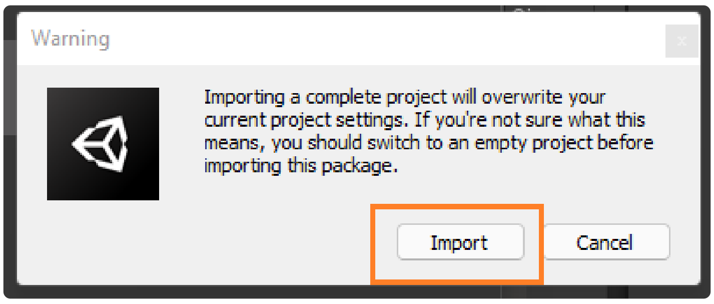

Configuration
Scripting Backend: Mono
Api Compatibility Level*: .NET 4.x
C++ Compiler Configuration: Release
Use incremental GC: ☐
Assembly Version Validation: ✓
Camera Usage Description*:


Microphone Usage Description*
Bluetooth Usage Description*
Active Input Handling* Input System Package (New)

ⓘ Changing the Input Handling System requires an editor restart.

2. Navigate to: "**Window / Package Manager / My Assets**" and find **STP: Survival Template PRO**.

3. After downloading the asset make sure to import the necessary project settings.


Warning

Importing a complete project will overwrite your current project settings. If you're not sure what this means, you should switch to an empty project before importing this package.

ⓘ STP only requires a few project settings to function such as: custom **Layers** and **Collision Matrix.**

That is all, STP should function perfectly fine now. In case you encountered any **problems** with the installation feel free to ask for **support**.

Demo Maps

Before you create a new scene or import the asset's functionality into your existing one, I recommend taking a look at the demo maps that come with the asset.

ⓘ Demo maps path: "**PolymindGames / SurvivalTemplatePro / _Demo / Scenes**".

Scene Setup

Whether you created a new scene or want to import the asset functionality into your existing one, you'll need a few objects to make sure the scene will function properly.

1. After you're in your scene that you want to setup. Navigate to: **"Tools / STP / Scene Setup"**

2. After pressing on "Setup Scene" your scene will become fully usable.
3. Don't forget to place some spawn points otherwise the Player will be spawned where the **Player*Setup*** *object is placed.*

> Path: **"PolymindGames / Survival Template Pro / Samples / Player / Player_SpawnPoint".**

> ⓘ Don't forget to bake the **Occlusion Culling** and **NavMesh.**

Frequently Asked

Here's the most asked questions that user's have regarding this asset.

Frequently Asked Questions

> "How can I create my own **Custom Items**?"

Create Item

> "How can I create and configure a **new Wieldable**?

Overview

> "How can I add **Custom Impact Effects** to STP?"

Create Surface

"Where can I change my Player's max health, stamina etc."

Health

Integrations

Integrated Third-Party assets

Universal AI 2.0

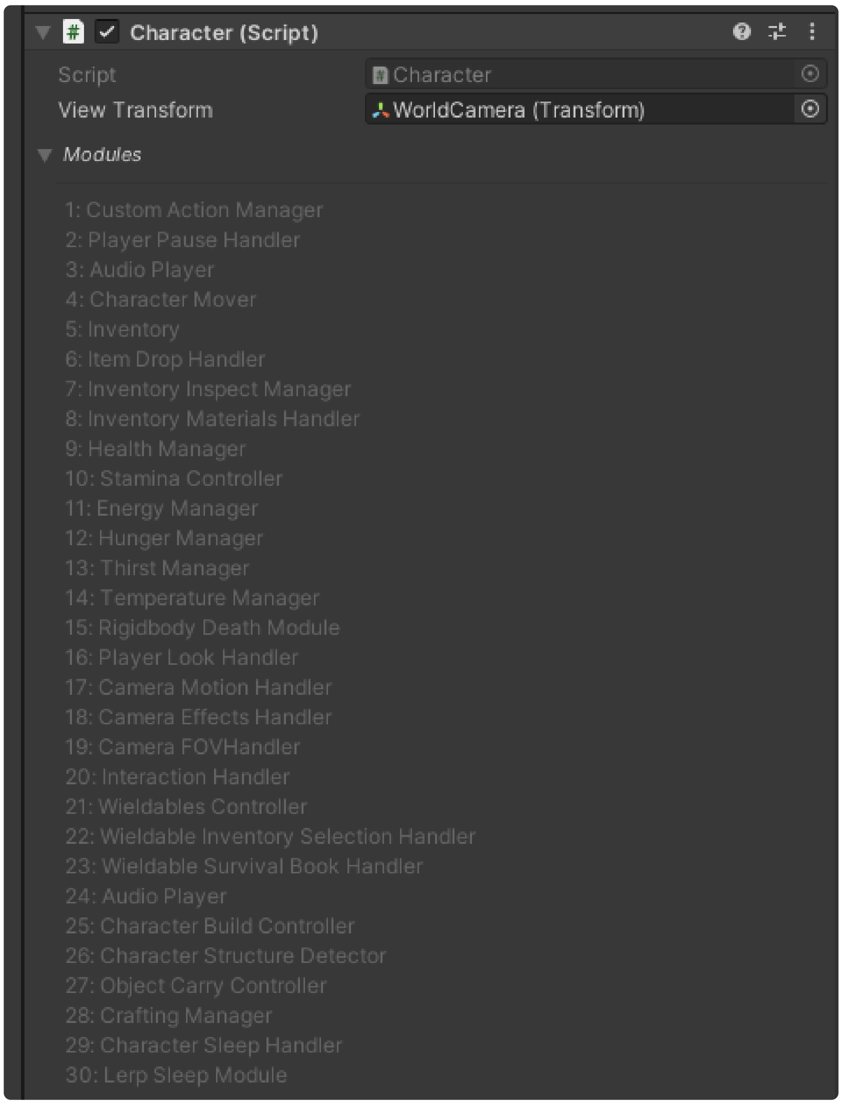
Info on how to use this integration:

Ready Integrations
Universal AI 2.0

Player

Character

Main character component used by every entity in the game. It acts as a hub for accessing modules.



Character (Script)

Script: Character
View Transform: WorldCamera (Transform)

Modules

1: Custom Action Manager
2: Player Pause Handler
3: Audio Player
4: Character Mover
5: Inventory
6: Item Drop Handler
7: Inventory Inspect Manager
8: Inventory Materials Handler
9: Health Manager
10: Stamina Controller
11: Energy Manager
12: Hunger Manager
13: Thirst Manager
14: Temperature Manager
15: Rigidbody Death Module
16: Player Look Handler
17: Camera Motion Handler
18: Camera Effects Handler
19: Camera FOVHandler
20: Interaction Handler
21: Wieldables Controller
22: Wieldable Inventory Selection Handler
23: Wieldable Survival Book Handler
24: Audio Player
25: Character Build Controller
26: Character Structure Detector
27: Object Carry Controller
28: Crafting Manager
29: Character Sleep Handler
30: Lerp Sleep Module

Modules & Behaviours

Detalis on the modules and behaviours present in STP.

> **Good to know!** A Character Module is a component that adds functionality to a character and can be accessed from any character component.

> **Good to know!** A Character Behaviour is a component that interacts with a module or multiple modules but not vice versa.

Inventory

Here you'll learn what each character inventory components does.

Inventory (Module)

> Stores items in containers. It also handles removal by name/id, count items etc.

Initial Containers: The initial item containers. You can also add more containers at runtime using the *AddContainer()* method.

For more information on item containers see:

Item Container

Item Drop Handler (Module)

> Handles item pickup dropping.

Drop Obstacle Mask: The layer mask that will be used in checking for obstacles when items are dropped.

Drop Transform: Dropped items are spawned relative to this transform.

Drop Force Mode: The force mode that will be added to the rigidbody of the dropped item.

Item Drop Settings: Position, rotation offsets etc.

Sack Prefab: The prefab used when an item that's dropped doesn't have a pickup (e.g. clothes) or when dropping multiple items at once.

Inventory Startup Items (Behaviour)

> ⓘ Adds items to the inventory at startup using a predefined list.

Inventory Startup Info: A predefined list of items that will be added to the inventory.
To create one, right click in the project view and select:

"Create / Survival Template Pro / Inventory / Startup Items"

Inventory Inspect Manager (Module)

> ⓘ Handles any type of inventory inspection (e.g. Backpack, external containers etc.)

Toggle Threshold: How often can the inventory inspection be toggled (e.g. open/close backpack).

Inventory Materials Handler (Module)

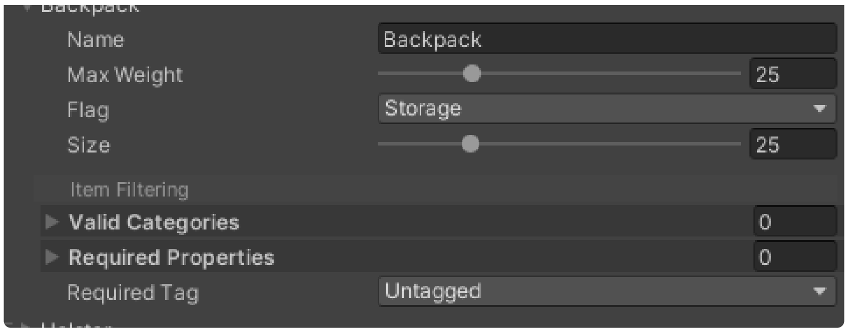
> ⓘ Acts as a wrapper for adding inventory based materials (e.g. sticks, rope etc.) to the structure (building preview) that's in view.

For information about items look into:

Create Item

Item Container

At their core item containers are just a list of items.



Name	Backpack
Max Weight	25
Flag	Storage
Size	25
Item Filtering	
Valid Categories	0
Required Properties	0
Required Tag	Untagged

Name: The name of the item container. If you need to get access to this container you can use the *Inventory.GetContainerWithName()* method and pass in the name of this container.

Max Weight: The max weight that this container can hold, no item can be added if it exceeds the limit.

Flag: Acts as a container type.

Size: Number of item slots that this container has (e.g. Holster 8, Backpack 25 etc.).

Valid Categories: Only items from the specified categories can be added.

Required Properties: Only items with the specified properties can be added.

Required Tag: Only items that are tagged with the specified tag can be added.

Movement

Here you'll learn what each character movement component does.

Player Movement Input (Behaviour)

Delegates movement input (using the new Unity Input System) to the ICharacterMover module.

Character Mover (Module)

Handles character movement. (Will be refactored into a more modular system)

Acceleration: How fast can this character achieve max state velocity.

Damping: How fast will the character stop when there's no input (a high value will make the movement more snappy).

Airborne Control: How well can this character move while airborne.

Step Length: How much distance does this character need to cover to be considered a step

Forward Speed: The forward speed of this character.

Back Speed: The backward speed of this character.

Side Speed: The sideway speed of this character.

Slope Speed Mod: Lowers/Increases the moving speed of the character when moving on sloped surfaces (e.g. lower speed when walking up a hill).

Anti Bump Factor: A small force applied to the character to make it stick to the ground when moving on a descending terrain/collider.

Enable Running: Is this character allowed to run?

Run Speed: The max running speed.

Run Step Length: Step length specific to running.

Enable Jumping: Is this character allowed to jump?

Jump Height: The max height of a jump.

Jump Timer: How often can this character jump (in seconds).

Enable Crouching: Is this character allowed to crouch?

Crouch Speed Mod: The velocity mod for crouching (multiplied with the base speed: forward, back speed etc.)

Crouch Step Length: Step length specific to crouch walking.

Crouch Height: The controllers height when crouching.

Crouch Duration: How long does it take to crouch.

Enable Sliding: Should this character slide when standing on slopes?

Slide Threshold: The angle at which the character will start to slide.

Slide Speed: The max sliding speed.

Gravity: The strength of the gravity.

Obstacle Check Mask: Layers that are considered obstacles.

Character Velocity Handler (Behaviour)

Low Vitals Velocity Mod: How much will the max velocity be affected by low vitals (e.g. hunger, thirst etc.)

Wieldable Weight Velocity Mod: How much will the max velocity be affected by the equipped wieldable weight.

Carriables Count Velocity Mod: How much will the max velocity be affected by the amount of carried "carriables".

Character Run Blocker (Behaviour)

Disable Run On Stamina Value: At which stamina value (0-1) will the ability to run be disabled.

Enable Run On Stamina Value: At which stamina value (0-1) will the ability to run be re-enabled (if disabled)

Character Jump Blocker (Behaviour)

Disable Jump On Stamina Value: At which stamina value (0-1) will the ability to jump be disabled.

Enable Jump On Stamina Value: At which stamina value (0-1) will the ability to jump be re-enabled (if disabled)

Camera

Here you'll learn what each character camera component does.

Player Look Input (Behaviour)

> ⓘ Delegates camera/look input (using the new Unity Input System) to the ILookHandler module.

Player Look Handler (Module)

> ⓘ Handles look updating the look direction of a character.

X Transform: Transform to rotate Up & Down.

Y Transform: Transform to rotate Left & Right.

Invert: If enabled, the up-down direction will be inverted.

Look Limits: Vertical look limits (in angles).

Sensitivity: Rotation Speed.

Raw: If enabled, the rotation will not be smoothed.

Smooth Steps: Smooth steps amount (a bigger sample will make means the rotations will be more smoothed).

Smooth Weight: Smoothness affect modifier, a value of 0 means that the rotation will be raw while 1 as smooth as possible.

FOV Camera: Used in lowering/increasing the current sensitivity based on the FOV.

Camera Motion Handler (Module)

> Will be refactored in the future.

Camera: The camera that will be rotated using a spring.

Camera Root: The camera root that will be rotated through bobbing.

Motion Lerp Speed: How smooth should the motion spring movement be.

Bob Transition Speed: How fast should the camera bob be.

Input: Input settings.

Sway: Sway forces (applied when moving the camera).

Fall Impact: Fall impact forces.

Idle State: Idle camera motion state.

Walk State: Walk camera motion state.

Run State: Run camera motion state.

Crouch State: Crouch camera motion state.

Shake Lerp Speed: How smooth should the shake spring movement be.

Explosion Shake: The explosion shake settings.

Shake Spring Settings: The default shake spring force settings (Stiffness and Damping).

Force Lerp Speed: How smooth should the force spring movement be.

Default Force Spring Settings: The default spring force settings (Stiffness and Damping), when external forces are being applied.

Camera Effects Handler (Module)

Background Blur: Background blur component.

Foreground Blur: Foreground blur component.

Background Color Tweaks: Background color tweaks component.

Foreground Color Tweaks: Foreground color tweaks component.

Camera Height Controller (Behaviour)

Y Lerp Speed: How fast should the camera adjust to the current Y position (up - down).

Crouch Offset: An offset that will be applied to the camera position after crouching.

Crouch Easing: Crouch movement easing type.

Health

Here you'll learn what each character health component does.

Health Manager (Module)

> Manages the parent character's health and death

Starting Health: The starting health of this character (can't be higher than the max).

Starting Max Health: The starting highest health of this character (can be modified at runtime).

Stamina Controller (Module)

Regeneration Rate: How fast will the stamina regenerate.

Regeneration Pause: How much time the stamina regeneration will be paused after it gets lowered.

Run Decrease Rate: By how much will the stamina decrease per second when running.

Jump Take: By how much will the stamina decrease after jumping.

Hunger Manager (Module)

> ⓘ Simple stat, can be increased/lowered and increases/decreases the health after the specified value.

Thirst Manager (Module)

> ⓘ Simple stat, can be increased/lowered and increases/decreases the health after the specified value.

Energy Manager (Module)

> ⓘ Simple stat, can be increased/lowered and increases/decreases the health after the specified value.

Temperature Manager (Module)

> ⓘ Not Implemented

Player Death Handler (Behaviour)

> ⓘ Handles a player's death and respawn behaviour.

Item Drop Type:

All: Drop all items upon death.

Equipped: Drop only the equipped wieldable item.

None: Don't drop any items.

Rigidbody Death Module (Behaviour)

> ⓘ Moves the character's view (in stp's case camera) to a rigidbody and vice-versa when disabled.

Fall Damage Handler (Behaviour)

> ⓘ Handles dealing fall damage to the character based on the impact velocity.

Enable Damage: Is this behaviour enabled?

Min Fall Speed: At which landing speed, the character will start taking damage.

Fatal Fall Speed: At which landing speed, the character will die, if it has no defense.

Building

Here you'll learn what each character building component does.

Player Building Input (Behaviour)

> ⓘ Delegates building input (using the new Unity Input System) to the IBuildController and IStructureDetector modules.

Character Build Controller (Module)

Follow Character Rotation: Should the "placeables" follow the rotation of the character?

View Angle Threshold: Max angle for detecting nearby sockets.

Rotation Speed: How fast can the "placeables" be rotated.

Build Range: "Max building range.

Buildable Mask: Tells the controller on what layers the "buildables" are.

Free Placement Mask: Tells the controller on what layers can "placeables" be placed.

Overlap Check Mask: Tells the controller what layers to to check when checking for collisions.

Place Sound: Sound to play when placing an object.

Invalid Place Sound: Sound to play when the controller tries to place an object but detects a collision.

Character Build Effects (Behaviour)

> ⓘ Plays effects based on the IBuildController events.

Character Structure Detector (Module)

> ⓘ Detects nearby building previews (structures)

Max Detection Distance: The max detection building preview distance.

Max Detection Angle: The max angle detection building preview.

Object Carry

Here you'll learn what each character object carry component does.

Player Object Carry Input (Behaviour)

> ⓘ Delegates object carry input (using the new Unity Input System) to the IObjectCarryController.

Object Carry Controller (Module)

Handles everything regarded object carrying except the visuals.

Drop Obstacle Mask: The layer mask that will be used in checking for obstacles when carriables are dropped.

Drop Transform: Dropped carriables are spawned relative to this transform.

Drop Force Mode: The force mode that will be added to the rigidbody of the dropped carriable.

Wieldable

Here you'll learn what each character wieldable component does.

Player Wieldables Input (Behaviour)

Delegates wieldable inputs (using the new Unity Input System) to the wieldable modules.

Wieldables Controller (Module)

Controller responsible for equipping and holstering wieldables.

Audio Player: The audio player that every wieldable will use.

Wieldable Inventory Select Handler (Module)

Takes care of selecting wieldables based on inventory items.

Holster Container: The corresponding inventory container (e.g. holster, backpack etc.) that this behaviour will use for selecting items.

Select On Start: Should this behaviour select the "*starting slot item*" on start.

Starting Slot: The fist slot that will be selected.

Inventory Based Wieldables: All of the Item Id - Wieldable pairs.

Wieldable Survival Book Handler (Module)

> Handles enabling/disabling the survival book wieldable.

Survival Book Prefab: Corresponding survival book wieldable prefab.

Left Pages Object Name: The left pages object name of the book (used to parent the UI).

Right Pages Object Name: The right pages object name of the book (used to parent the UI).

Toggle Cooldown: How often can the survival book be enabled/disabled (in seconds).

Wieldable Object Carry Handler (Behaviour)

> Handles object carry wieldable carriable display count.

Character Ray Generator (Module)

> Generates direction rays, used by the wieldables.

Anchor: Transform used in determining the base ray position and orientation.

Walk Spread Mod: How much will walking affect (randomly spread) the final ray.

Run Spread Mod: How much will running affect (randomly spread) the final ray.

Crouch Spread Mod: How much will crouching affect (randomly spread) the final ray.

Airborne Spread Mod: How much will being airborne affect (randomly spread) the final ray.

Player Look Follow (Behaviour)

> Makes this object follow the player's look without parenting.
>
> Removes the need for long transform hierarchies.

Interaction

Here you'll learn what each character interaction component does.

Player Interaction Input (Behaviour)

> ⓘ Delegates interaction inputs (using the new Unity Input System) to the IInteractionHandler module.

Interaction Handler (Module)

View: The transform used in raycasting.

Raycast Distance: The raycast max distance, anything further away will be ignored.

Raycast Radius: The raycast radius, if you're not looking directly at an object you can still interact with that said object if it's in the given radius.

Layer Mask: Interaction layer mask, everything this handler can "see".

Crafting

Here you'll learn what each character crafting component does.

Crafting Manager (Module)

> ⓘ Handles item crafting.

Craft Sound: Sound that will be played after crafting an item.

Sleep

Here you'll learn what each character sleep component does.

Character Sleep Handler (Module)

Go To Sleep Duration: How much time it takes to transition to sleeping (e.g. moving to bed).

Sleep Delay: How much time it takes to fall asleep.

Sleep Duration: Sleep duration in seconds.

Wake Up Delay: How much time to wait after the sleep is done, before getting up.

Wake Up Duration: How much time it takes to transition from sleeping to standing up.

Hours To Sleep: Max hours that can pass while sleeping.

Max Get Up Hour: Max hour this character can wake up at, we don't want to be lazy :)

Only Sleep At Night: If enabled, this character will not be allowed to sleep during the day.

Check For Enemies: Check for enemies before sleeping, if any of the are found, this character will be unable to sleep.

Check For Enemies Radius: The enemy check radius.

Enemies Layer Mask: The enemy layer, any object with this layer will be considered an enemy.

Cant Sleep Sound: Sound that will be played when attempting to sleep unsuccessfully.

Character Sleep Effects (Behaviour)

> Plays effects based on the ISleepHandler events.

Lerp Sleep Module (Module)

> Module responsible for moving the Player's view (camera) to an ISleepingPlace position and rotation.

Audio

Here you'll learn what each character audio component does.

Audio Player (Module)

> The main audio player of this character, can play sounds with delays, loop sounds etc.

Human Sounds Manager (Behaviour)

Responsible for listening to events raised by other modules (e.g. health, movement etc.) and playing "humanoid" sounds accordingly.

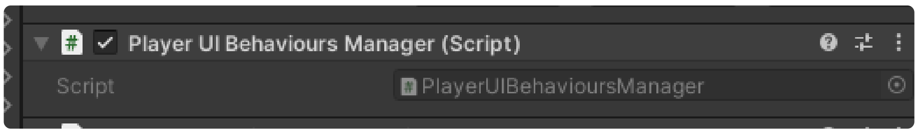
Footsteps Manager (Behaviour)

Responsible for playing footsteps sound based on the current IMover's module step length. See:

Movement

User Interface

UI Behaviours Manager



Player UI Behaviours Manager (Script)

Script PlayerUIBehavioursManager

Responsible for attaching and detaching UI behaviours to a player.

Behaviours

UI_Inventory

Here you'll learn what each UI Inventory behaviour does.

Player UI Inventory Input

> ⓘ Delegates input (using the new Unity Input System) to sub behaviours using Unity Events.

Inventory Panels Manager

> ⓘ Handles the inventory panels.

Item Auto Mover

Wieldable Tags: All of the item tag(s) that correspond to the wieldables.

Clothing Tags: All of the clothing tag(s) that correspond to clothing items.

Item Drag Handler

Dragged Item Scale: The visual scale of the dragged item.

Dragged Item Alpha: The transparency of the dragged item.

Slot Template: Slot template prefab that will be instantiate when an item gets dragged.

UI_Damage

Here you'll learn what each UI Damage behaviour does.

Damage UI

Alpha Damage Weight: How much will the alpha of the effects be affected by the amount of damage received.

Blood Screen Fader: Image fading settings for the blood screen.

Blood Screen Fader Damage Threshold: How much damage does the player have to take for the blood screen effect to show.

Directional Damage Indicator Fader: Image fading settings for the directional damage indicator.

Directional Indicator Distance: The damage indicator distance (in pixels) from the screen center.

UI_Vitals

Here you'll learn what each UI Vitals behaviour does.

> ⓘ Will be refactored into generic vitals in the future.

Player Vitals UI

Health Bar: The health bar image, the fill amount will be modified based on the current health value.

Energy Bar: The energy bar image, the fill amount will be modified based on the current energy value.

Thirst Bar: The thirst bar image, the fill amount will be modified based on the current thirst value.

Hunger Bar: The hunger bar image, the fill amount will be modified based on the current hunger value.

Player Stamina UI

Canvas Group: The canvas group used to fade the stamina bar in & out.

Fill Bar: The stamina bar image, the fill amount will be modified based on the current stamina value.

Hide Duration: Represents how much time it takes for the stamina bar to start fading after not decreasing.

Alpha Lerp Speed: How fast will the stamina bar alpha fade in & out.

UI_Wieldables

Here you'll learn what each UI Wieldable behaviour does.

Scope

Canvas Group: The canvas group used to fade the scopes in & out.

Scopes: All of the existing UI scopes.

Crosshairs

Default Crosshair: The default (starting) crosshair.

Canvas Group: The canvas group used to fade a crosshair in & out.

Crosshair Alpha: The max crosshairs alpha.

Alpha Lerp Speed: The speed at which the crosshairs will change their alpha.

Ammo

Animator: Reference to the animator used for the Ammo UI.

Magazine Text: A UI text component that's used for displaying the current ammo in the magazine.

Inventory Text: A UI text component that's used for displaying the current ammo in the storage.

Charge

Canvas Group: The canvas group used to fade the stamina bar in & out.

Charge Fill Images: UI images that will have their fill amount value set to the current charge value.

Fill Gradient: A gradient used in determining the color of the charge image relative to the current charge value.

UI_Interaction

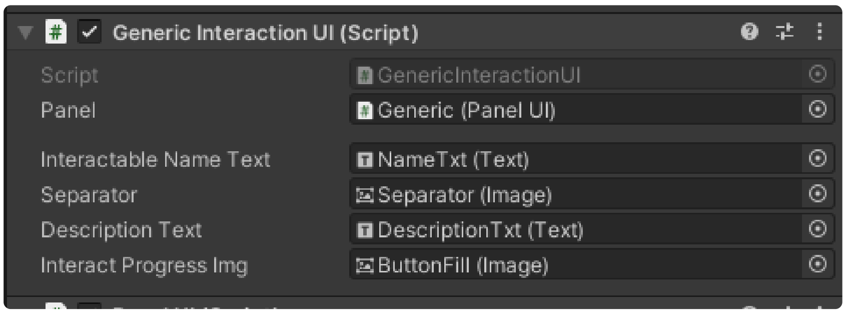
Here you'll learn what each UI Interaction behaviour does.

Interactable Info Display UI

> Check the type of the interactable that the Player has interacted with and finds the corresponding displayer for it.

Displayers

Generic



Generic Interaction UI (Script)

Script	GenericInteractionUI
Panel	Generic (Panel UI)
Interactable Name Text	NameTxt (Text)
Separator	Separator (Image)
Description Text	DescriptionTxt (Text)
Interact Progress Img	ButtonFill (Image)

Panel: The UI panel used in showing / hiding the underlying images.

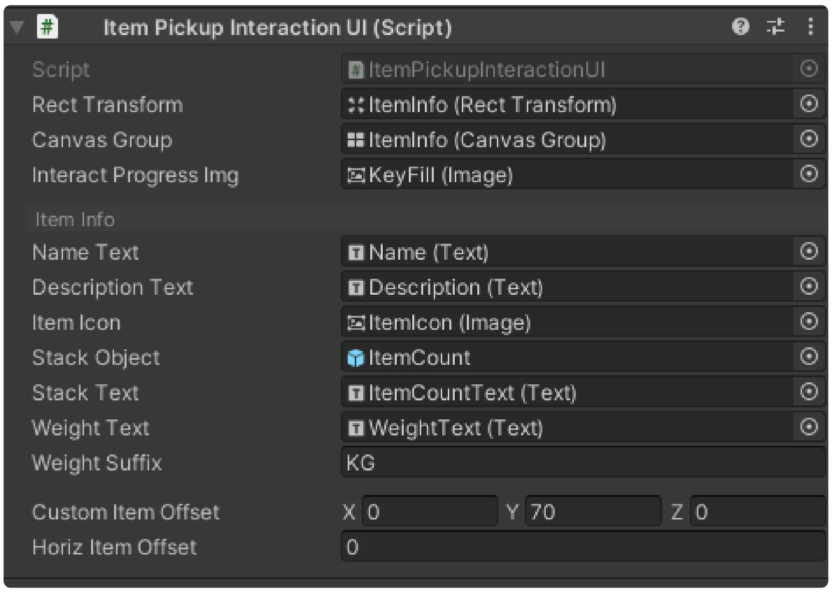
Interactable Name Text: A UI text component that's used for displaying the current interactable's name.

Separator: An image that separate the name text from the description text (optional).
It gets disabled when the current interactable doesn't have a description.

Description Text: A UI text component that's used for displaying the current interactable's description.

Interact Progress Image: An image that used in showing the time the current interactable has been interacted with.

Item Pickup



Item Pickup Interaction UI (Script)

Script	ItemPickupInteractionUI
Rect Transform	ItemInfo (Rect Transform)
Canvas Group	ItemInfo (Canvas Group)
Interact Progress Img	KeyFill (Image)
Item Info	
Name Text	Name (Text)
Description Text	Description (Text)
Item Icon	ItemIcon (Image)
Stack Object	ItemCount
Stack Text	ItemCountText (Text)
Weight Text	WeightText (Text)
Weight Suffix	KG
Custom Item Offset	X 0 Y 70 Z 0
Horiz Item Offset	0

Rect Transform: The main rect transform that will move the interactable's center (should be parent of everything else).

Canvas Group: The canvas group used to fade the item pickup displayer in & out.

Interact Progress Image: An image that used in showing the time the current interactable has been interacted with.

Name Text: A UI text that's used for displaying the observed item pickup's name.

Description Text: A UI text that's used for displaying the observed item pickup's description.

Item Icon: An UI Image that's used for displaying the observed item pickup's icon.

Stack Object: An object that will be enabled based on if the observed item has a stack that's bigger than one and otherwise disabled.

Stack Text: A UI text that's used for displaying the observed item pickup's current stack size.

Weight Text: A UI text that's used for displaying the observed item pickup's weight.

Weight Suffix: A string that will be added at the of the weight amount (e.g. amount + "KG" or "LBS")

Custom Item Offset: An offset that will be applied to the position of the "*rect transform*".

UI_Sleep

Here you'll learn what each UI Sleeping behaviour does.

Sleep UI

Canvas Group: The canvas group used to fade the sleep UI in & out.

Max Canvas Alpha: The max alpha that will be reached when fading in.

Canvas Lerp Speed: The speed at which the sleep UI will be faded in & out.

Current Time Text: A UI text component that's used for displaying the current time while sleeping.

Background Rect: The background rect of this UI piece.

Animator: Reference to the animator used for the sleeping UI.

Show Trigger: The "*show*" animator trigger.

Hide Trigger: The "*hide*" animator trigger.

Number of Templates: How many time templates should be spawned each side.

Time Text Template: A prefab with a text component on it that will be instantiated.

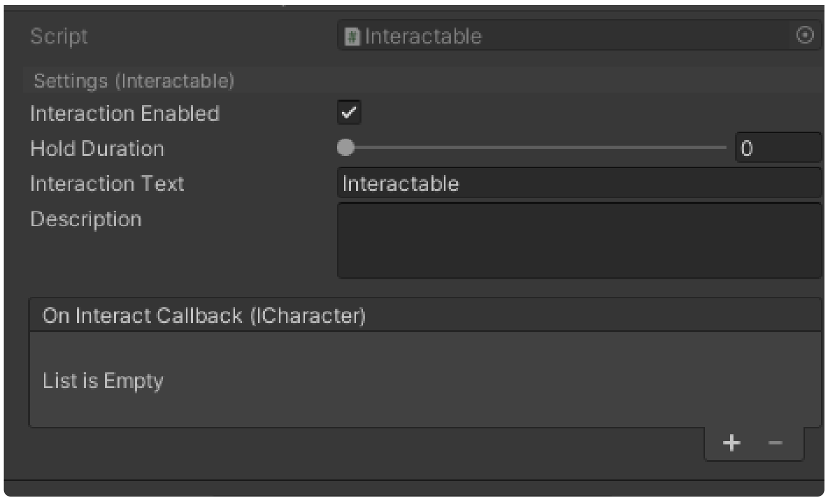
Text Size Over Distance: How will the text scale over the duration of its lifetime.

Text Y Position Over Distance: How will the text y position change over the duration of its lifetime.

Text Color Over Distance: How will the color of the text change over the duration of its lifetime.

Interaction

Interactable


Script: Interactable

Settings (Interactable)

Interaction Enabled: ✓

Hold Duration: 0

Interaction Text: Interactable

Description:

On Interact Callback (ICharacter)

List is Empty

Explanation

Interaction Enabled: Is this object interactable, if not, this object will be treated like a normal one.

Hold Duration: How time it takes to interact with this object. (e.g. for how many seconds should the Player hold the interact button).

Interaction Text: Interactable text (could be used as a name), shows up in the UI when looking at an object.

Description: Interactable description, shows up in the UI when looking at an object.

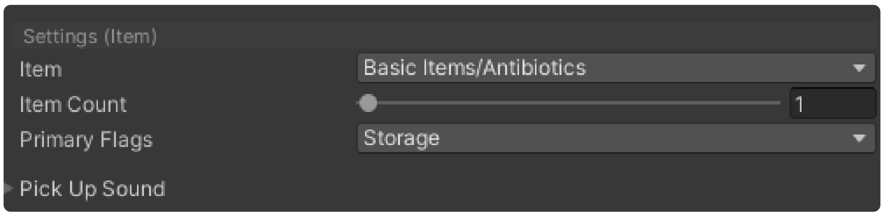
On Interact Callback: Unity event that will be called when a character interacts with this object.

Demo Interactables

Details on all of the included interactable types

Item Pickup

> Examples Path: *"PolymindGames / SurvivalTemplatePro / Samples / ItemPickups"*


Settings (Item)

Item: Basic Items/Antibiotics

Item Count: 1

Primary Flags: Storage

Pick Up Sound

Item: Corresponding item id.

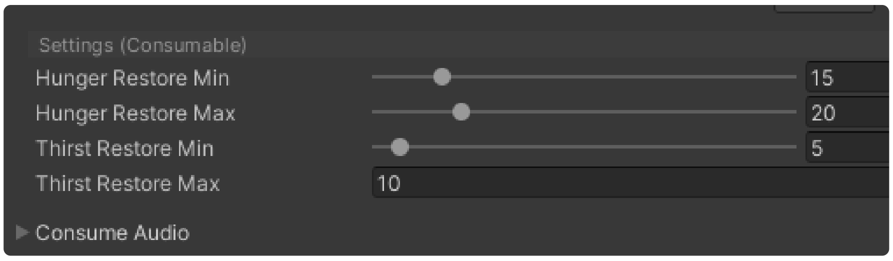
Item Count: How many items of the same type is this item going to contain.

Primary Flags: The first type of container that this item will be added to (e.g. a weapon would be added to the holster at first).

Pick Up Sound: Sound that will be played upon picking the item up.

Consumable

> Examples Path: *"PolymindGames / SurvivalTemplatePro / Samples / Consumables"*



Settings (Consumable)

Hunger Restore Min	15
Hunger Restore Max	20
Thirst Restore Min	5
Thirst Restore Max	10

Consume Audio

Hunger Restore Min: The **minimum** amount of hunger this consumable can restore.

Hunger Restore Max: The **maximum** amount of hunger this consumable can restore.

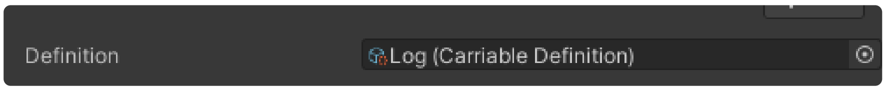
Thirst Restore Min: The **minimum** amount of thirst this consumable can restore.

Thirst Restore Max: The **maximum** amount of thirst this consumable can restore.

Consume Audio: Audio that will be played after a character consumes this.

Carriable

> Example Path: *"PolymindGames / SurvivalTemplatePro / Samples / Carriables*

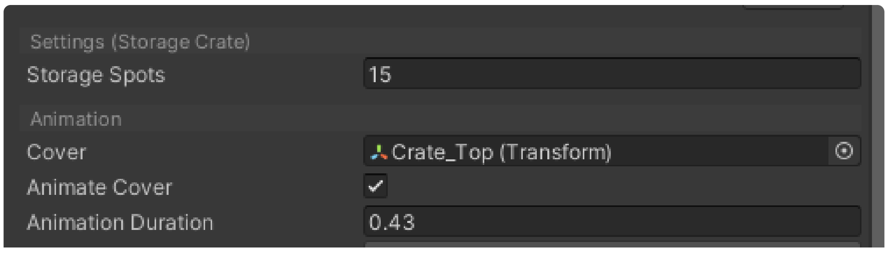


Definition: Log (Carriable Definition)

Definition: The corresponding carriable definition.

Storage Crate

> Example Path: *"PolymindGames / SurvivalTemplatePro / Samples / Buildables / Free"*

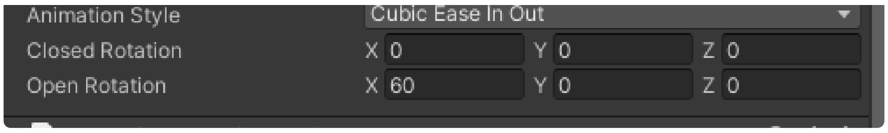


Settings (Storage Crate)

Storage Spots	15

Animation

Cover	Crate_Top (Transform)
Animate Cover	✓
Animation Duration	0.43


Animation Style: Cubic Ease In Out
Closed Rotation: X 0, Y 0, Z 0
Open Rotation: X 60, Y 0, Z 0

Storage Spots: How many slots should this storage crate have.

Cover: Crate cover transform (used for the open/close animations.

Animate Cover: Should the cover be animated?

Animation Duration: How long should the open/close animations last.

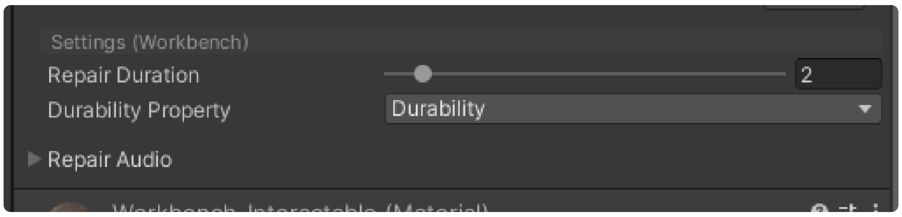
Animation Style: Animation easing type.

Closed Rotation: The crate cover closed rotation.

Open Rotation: The crate cover open rotation.

Workbench

> Example Path: *"PolymindGames / SurvivalTemplatePro / Samples / Buildables / Free"*


Settings (Workbench)
Repair Duration: 2
Durability Property: Durability
Repair Audio

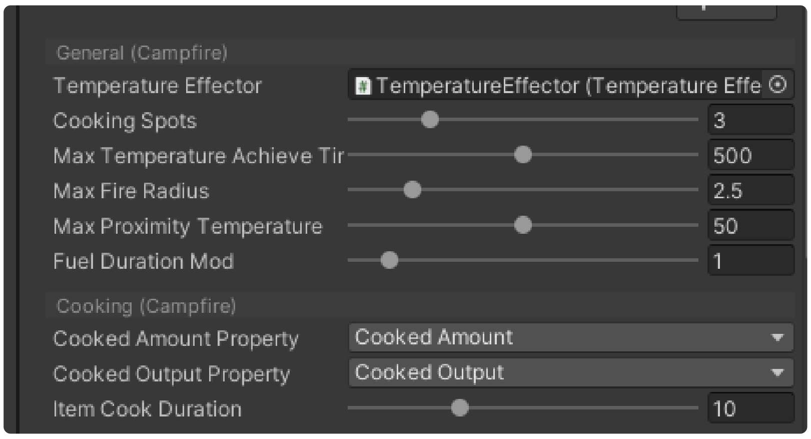
Repair Duration: The time it takes to repair an item at this workbench.

Durability Property: The id of the durability property. After repairing an item the workbench will increase the value of that property for the repaired item.

Repair Audio: Repair sound to be played after successfully repairing an item.

Campfire

> Example Path: *"PolymindGames / SurvivalTemplatePro / Samples / Buildables / Free"*


General (Campfire)
Temperature Effector: TemperatureEffector (Temperature Effe
Cooking Spots: 3
Max Temperature Achieve Tir: 500
Max Fire Radius: 2.5
Max Proximity Temperature: 50
Fuel Duration Mod: 1
Cooking (Campfire)
Cooked Amount Property: Cooked Amount
Cooked Output Property: Cooked Output
Item Cook Duration: 10

Temperature Effector: Not Implemented

Cooking Spots: How many cooking spots (item slots) this campfire has.

Max Temperature Achieve Time: Not Implemented

Max Fire Radius: Not Implemented

Max Proximity Temperature: Not Implemented

Fuel Duration Mod: Multiplies the effects of any fuel added (heat and added time).

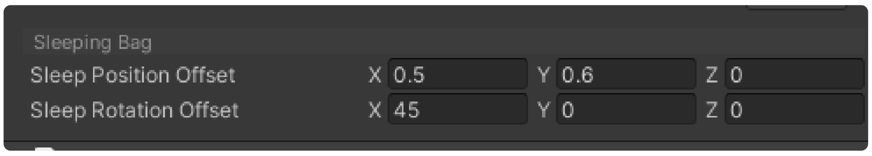
Cooked Amount Property: The property that tells the campfire how cooked an item is.

Cooked Output Property: The property that tells the campfire in what item should the cooked item transform.

Item Cook Duration: The amount of time it takes to cook an item.

Sleeping Bag

> Example Path: *"PolymindGames / SurvivalTemplatePro / Samples / Buildables / Free"*



Sleeping Bag

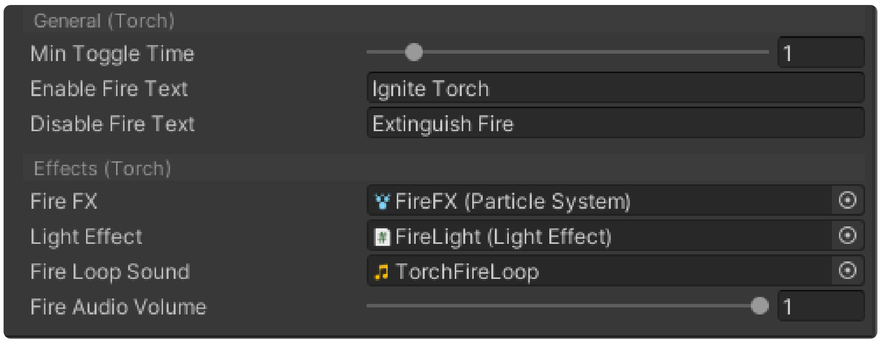
Sleep Position Offset	X 0.5	Y 0.6	Z 0
Sleep Rotation Offset	X 45	Y 0	Z 0

Sleep Position Offset: Position offset for the player sleeping handler. (Where the camera will be positioned).

Sleep Position Offset: Rotation offset for the player sleeping handler. (In which direction will the camera be pointed).

Standing Torch

> Example Path: *"PolymindGames / SurvivalTemplatePro / Samples / Buildables / Free"*



General (Torch)

Min Toggle Time	1
Enable Fire Text	Ignite Torch
Disable Fire Text	Extinguish Fire

Effects (Torch)

Fire FX	FireFX (Particle System)
Light Effect	FireLight (Light Effect)
Fire Loop Sound	TorchFireLoop
Fire Audio Volume	1

Min Toggle Time: Toggle fire on/off cooldown.
Enable Fire Text: Interaction text for when the fire is **not** ignited.

Disable Fire Text: Interaction text for when the fire is ignited.

Fire FX: Fire particle effects.

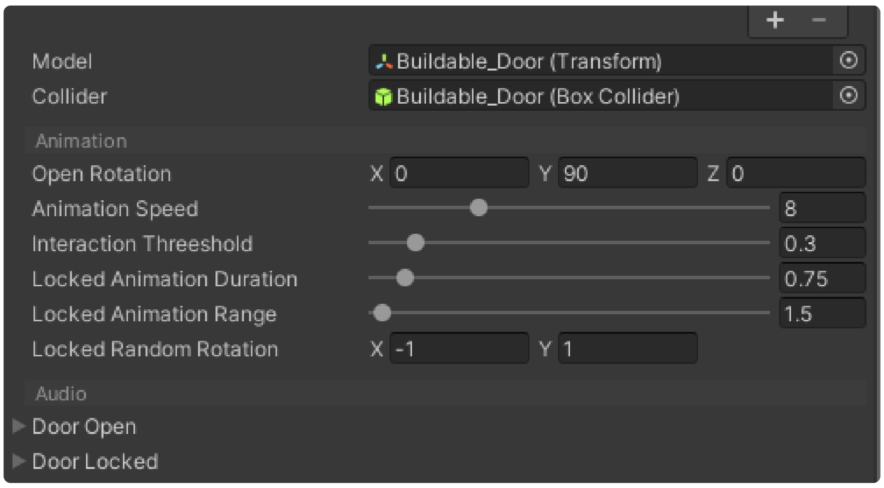
Light Effect: Fire light effect component.

Fire Loop Sound: Sound that will be looping while the fire is on.

Fire Audio Volume: Looping fire audio volume.

Door

> Example Path: *"PolymindGames / SurvivalTemplatePro / Samples / Buildables / Socket"*



Model	Buildable_Door (Transform)
Collider	Buildable_Door (Box Collider)
Animation	
Open Rotation	X 0 Y 90 Z 0
Animation Speed	8
Interaction Threeshold	0.3
Locked Animation Duration	0.75
Locked Animation Range	1.5
Locked Random Rotation	X -1 Y 1
Audio	
Door Open	
Door Locked	

Model: The door model.

Collider: The door collider.

Open Rotation: Open rotation offset (how much should this door rotate when it opens).

Animation Speed: Open animation rotation speed.

Interaction Threshold: Open/Close door time cooldown.

Locked Animation Duration: How much time should the locked animation last.

Locked Animation Range: How much should the locked animation move the door.

Locked Random Rotation: Locked animation randomness.

Door Open: Audio to play when the door opens.

Door Locked: Audio to play when the tries to open while locked.

Extension

Example (inherit from Interactable)

```
namespace SurvivalTemplatePro
{
    public class InteractableTest : Interactable
    {
        public override void OnInteract(ICharacter character)
        {
            base.OnInteract(character);

            // Do something when a character interacts with this.
            // e.g. press a button
        }

        public override void OnHoverStart(ICharacter character)
        {
            base.OnHoverStart(character);

            // Do something when a character starts looking at this object.
            // e.g. Enable outline effect
        }

        public override void OnHoverEnd(ICharacter character)
        {
            base.OnHoverEnd(character);

            // Do something when a character stops looking at this object.
            // e.g. Disable outline effect
        }
    }
}
```

Example (Implement the IInteractable interface)

```
namespace SurvivalTemplatePro
{
    public class InteractableTest : MonoBehaviour, IInteractable
    {
        public bool InteractionEnabled { get; set; }
        public string InteractionText => "Interaction Text";
        public string DescriptionText => "Description Text";

        public float HoldDuration => 0f;

        public event UnityAction onInteracted;
        public event UnityAction onDescriptionTextChanged;
        public event UnityAction<bool> onInteractionEnabledChanged;


        public void OnInteract(ICharacter character)
        {
            // Do something when a character interacts with this.
            // e.g. press a button
        }

        public void OnHoverStart(ICharacter character)
        {
            // Do something when a character starts looking at this object.
            // e.g. Enable outline effect
        }

        public void OnHoverEnd(ICharacter character)
        {
            // Do something when a character stops looking at this object.
            // e.g. Disable outline effect
        }
    }
}
```

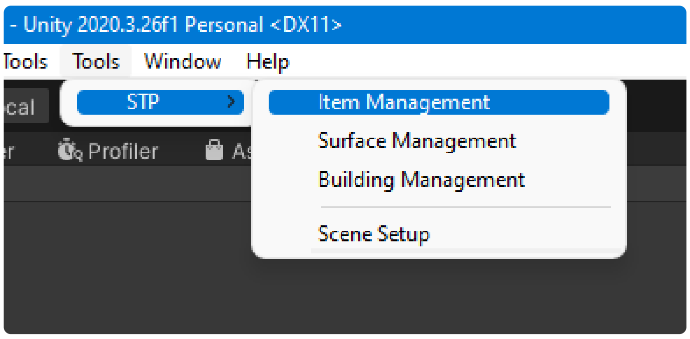
Inventory

Create Item

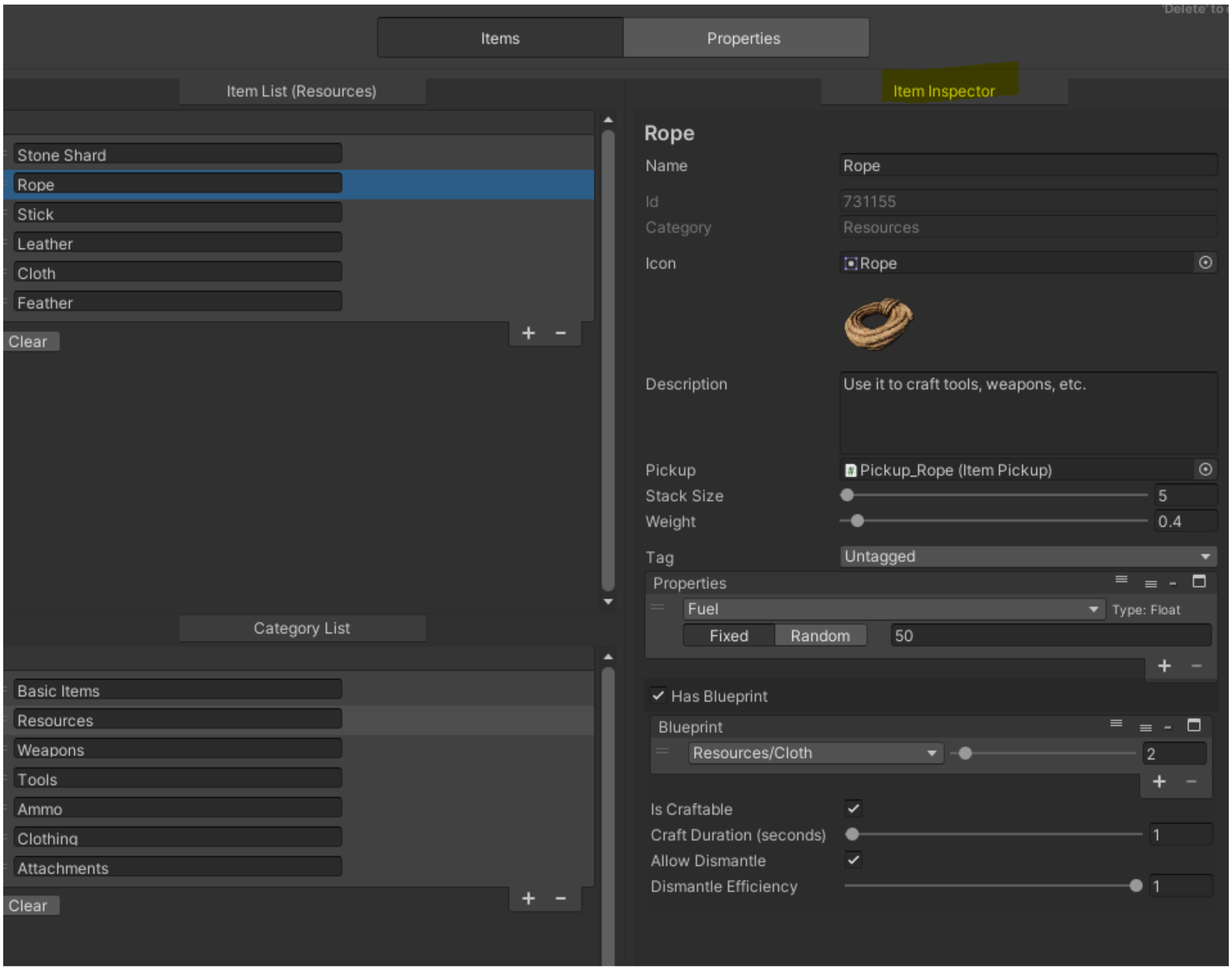
Here you'll learn how to create an item

Item Info

- To create any item navigate to: "Tools / STP / Item Management".



- While there, you'll see the items organized by category. Choose a category, or create your own, and then create your item in that category, just click the "+" button for adding new elements.
- When you've added a new item in your desired category, click on it, you'll see all of it's info in the Item Inspector.



Items | Properties

Item List (Resources)

- Stone Shard
- Rope
- Stick
- Leather
- Cloth
- Feather

Clear

Category List

- Basic Items
- Resources
- Weapons
- Tools
- Ammo
- Clothing
- Attachments

Clear

Item Inspector

Rope

Name	Rope
Id	731155
Category	Resources
Icon	Rope
Description	Use it to craft tools, weapons, etc.
Pickup	Pickup_Rope (Item Pickup)
Stack Size	5
Weight	0.4
Tag	Untagged

Properties

Fuel	Type: Float	
Fixed	Random	50

✓ Has Blueprint

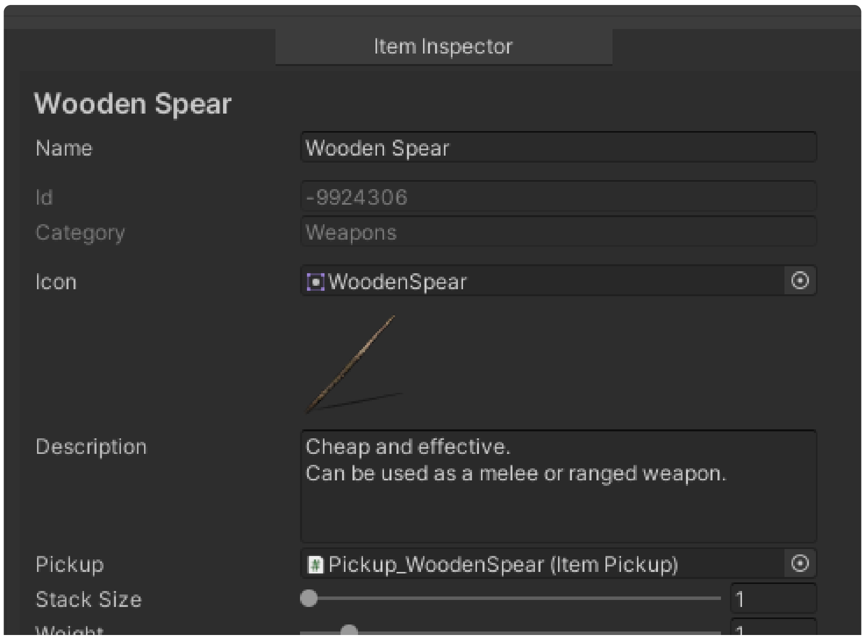
Blueprint

Resources/Cloth	2

Is Craftable	✓
Craft Duration (seconds)	1
Allow Dismantle	✓
Dismantle Efficiency	1

Item Settings

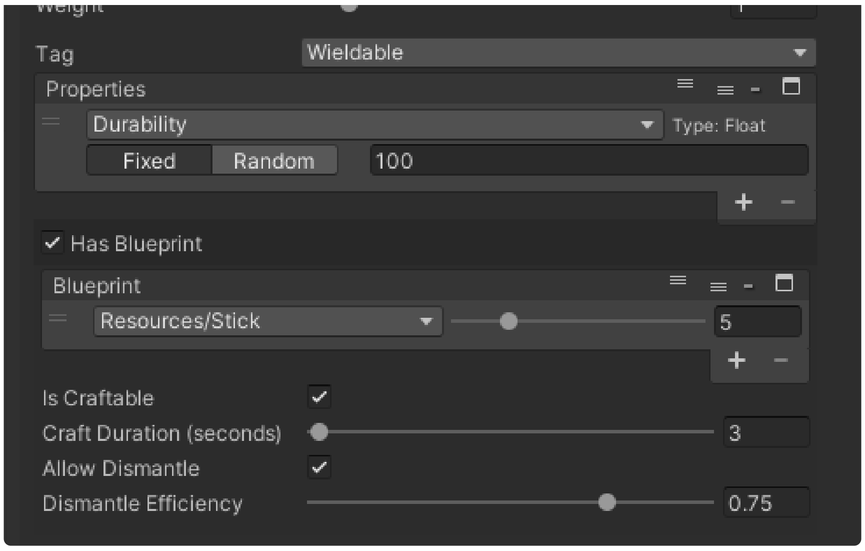
Here you can learn what every item setting means



Item Inspector

Wooden Spear

Name	Wooden Spear
Id	-9924306
Category	Weapons
Icon	WoodenSpear
Description	Cheap and effective. Can be used as a melee or ranged weapon.
Pickup	Pickup_WoodenSpear (Item Pickup)
Stack Size	1
Weight	1


Tag: Wieldable

Properties

Durability — Type: Float

Fixed | Random — 100

Has Blueprint ✓

Blueprint

Resources/Stick — 5

Is Craftable ✓

Craft Duration (seconds) 3

Allow Dismantle ✓

Dismantle Efficiency 0.75

Name: Item name.

Id: Unique id (auto generated).

Category: Category of this item.

Icon: Item Icon.

Description: Item description to display in the UI.

Pickup: Corresponding pickup for this item, so you can actually drop it, or pick it up from the ground.

Stack Size: How many items of this type can be stacked in a single slot.

Tag: String tag (similar to GameObject.Tag), it can have many use cases but the main one is to limit the item from being placed in certain item containers (e.g. stop this item from being placed in the holster container if it's not tagged as a wieldable). You can also define custom tags (see next tab).

Properties: all the item's properties, like Durability, Health Restore, Fuel and so on.
You can also define custom properties (see next tab).

Has Blueprint: Does this item have a blueprint? (Required for crafting and dismantling)

Blueprint: A list with all the "ingredients" necessary to craft this item, it's also used in dismantling.

IsCraftable: If enabled this item will show up in the survival book as a craftable item.

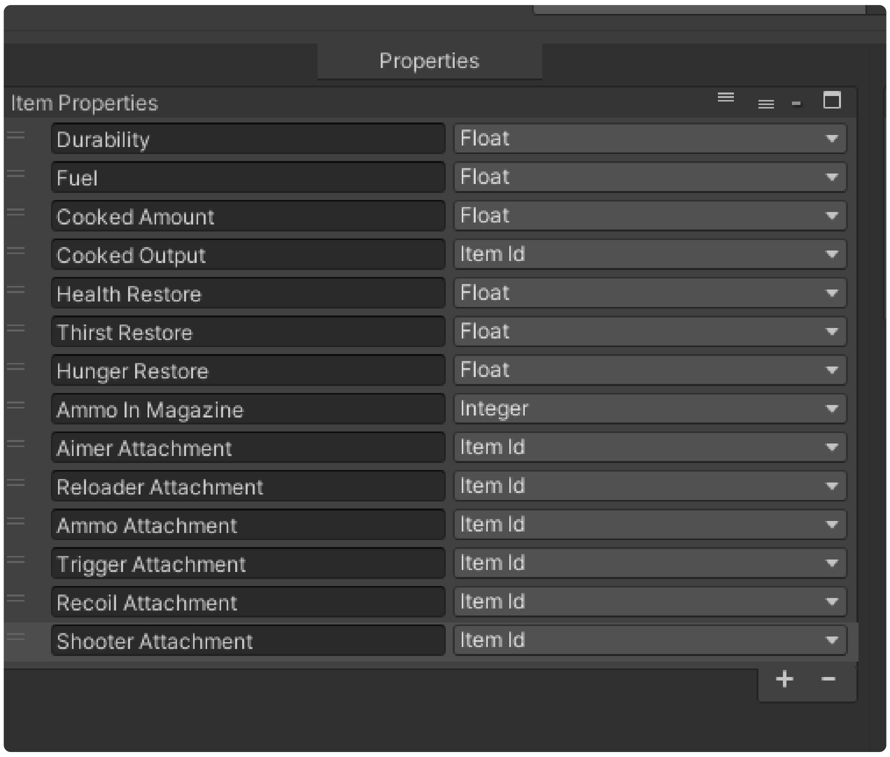
Craft Duration: How much time does it take to craft this item, in seconds.

Allow Dismantle: Can this item be dismantled?

Dismantle Efficiency: An efficiency of 1 will result in getting all of the item back after dismantling, while 0 means that no item from the blueprint will be made available.

Properties & Tags

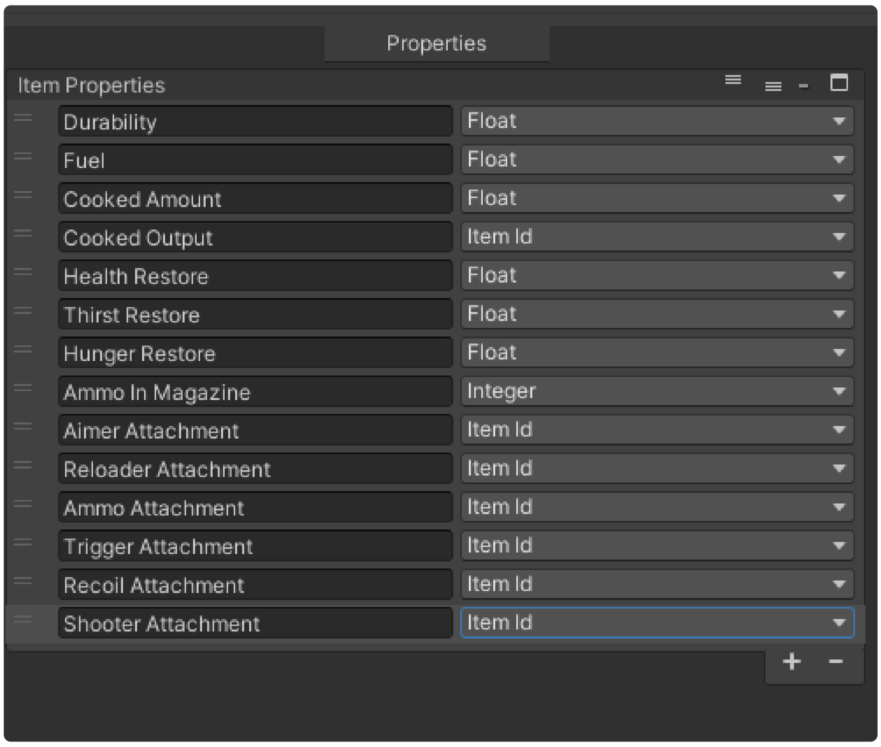
Properties


Properties

Item Properties

Durability	Float
Fuel	Float
Cooked Amount	Float
Cooked Output	Item Id
Health Restore	Float
Thirst Restore	Float
Hunger Restore	Float
Ammo In Magazine	Integer
Aimer Attachment	Item Id
Reloader Attachment	Item Id
Ammo Attachment	Item Id
Trigger Attachment	Item Id
Recoil Attachment	Item Id
Shooter Attachment	Item Id

To create a property press on the **"+"** button and set it's **name** and **type**.


Properties

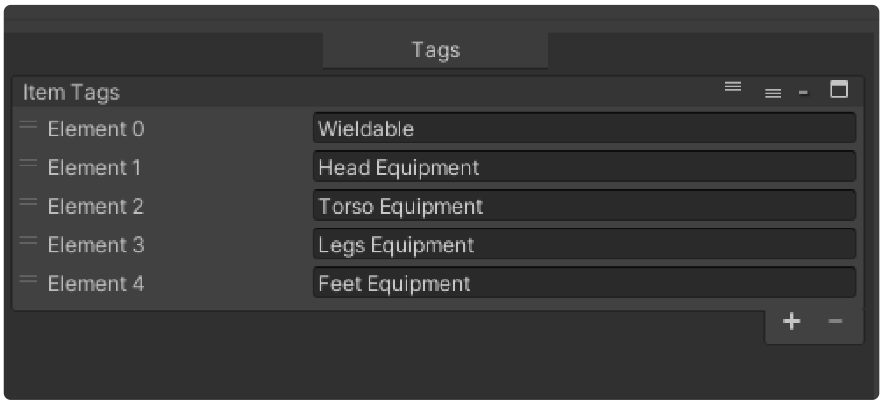
Item Properties

Durability	Float
Fuel	Float
Cooked Amount	Float
Cooked Output	Item Id
Health Restore	Float
Thirst Restore	Float
Hunger Restore	Float
Ammo In Magazine	Integer
Aimer Attachment	Item Id
Reloader Attachment	Item Id
Ammo Attachment	Item Id
Trigger Attachment	Item Id
Recoil Attachment	Item Id
Shooter Attachment	Item Id

Property Types:

- Boolean: Can be **true** or **false.**
- Integer: Basic integer, just a number.
- Float: Decimal number.
- Item Id: Holds an **item id**..

Tags


Tags

Item Tags

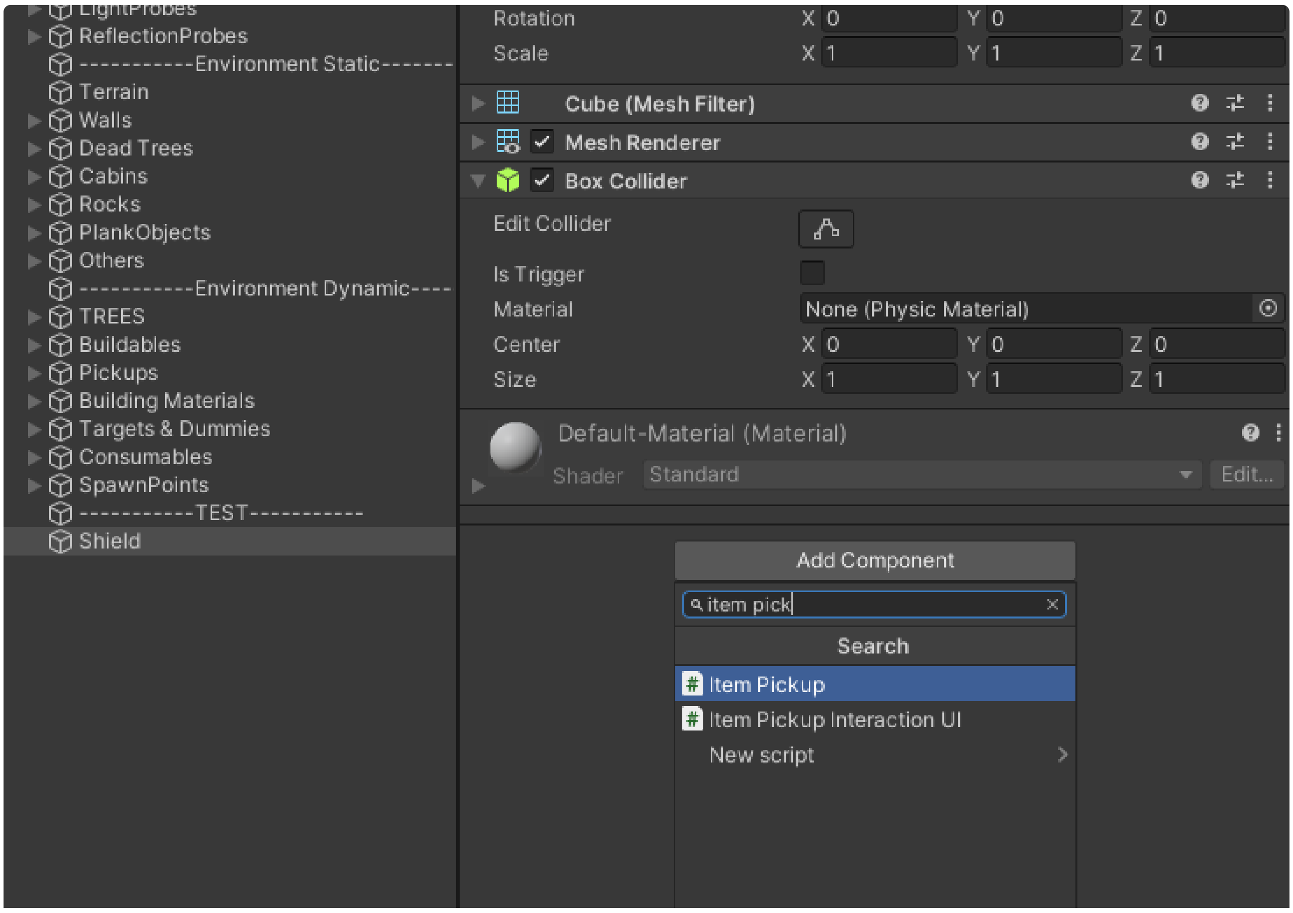
Element 0	Wieldable
Element 1	Head Equipment
Element 2	Torso Equipment
Element 3	Legs Equipment
Element 4	Feet Equipment

To create a property press on the **"+"** button and set it's **name**.

Item Pickup

How to Create

- To start creating an item pickup, add an "**Item Pickup**" component to your object:


LightProbes
ReflectionProbes
-----------Environment Static-------
Terrain
Walls
Dead Trees
Cabins
Rocks
PlankObjects
Others
-----------Environment Dynamic----
TREES
Buildables
Pickups
Building Materials
Targets & Dummies
Consumables
SpawnPoints
-----------TEST-----------
Shield

Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

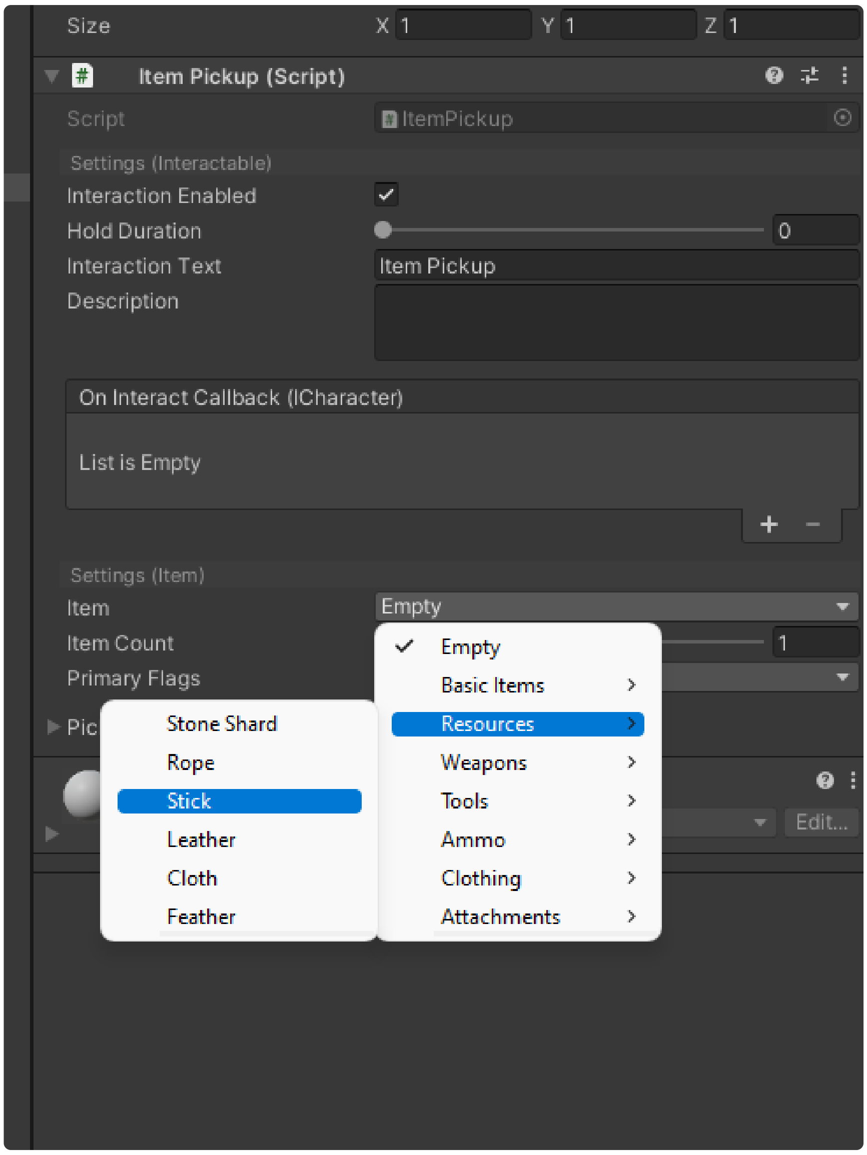
Cube (Mesh Filter)
Mesh Renderer
Box Collider

Edit Collider			
Is Trigger			
Material	None (Physic Material)		
Center	X 0	Y 0	Z 0
Size	X 1	Y 1	Z 1

Default-Material (Material)
Shader Standard Edit...

Add Component
item pick
Search
Item Pickup
Item Pickup Interaction UI
New script

- After that you'll need to set the corresponding item to the one that you want:



Size X 1 Y 1 Z 1

Item Pickup (Script)

Script: ItemPickup

Settings (Interactable)

Interaction Enabled: ✓

Hold Duration: 0

Interaction Text: Item Pickup

Description

On Interact Callback (ICharacter)

List is Empty

Settings (Item)

Item: Empty

Item Count: 1

Primary Flags

Explanation

Item: Corresponding item id.

Item Count: How many items of the same type is this item going to contain.

Primary Flags: The first type of container that this item will be added to (e.g. a weapon would be added to the holster at first).

Pick Up Sound: Sound that will be played upon picking the item up.

> **Good To Know!** Since this is an item pickup, there's no need to fill up the interaction and description texts, the pickup will do that automatically using the corresponding item.

Wieldables

Overview

A **Wizard** that helps creating new Wieldables is in the works, for now the best way to do that is copy and pasting any of the existing wieldable prefabs. Here's where you can find them:

"PolymindGames / SurvivalTemplatePro / Samples / Wieldables"

> The wieldable system is currently getting overhauled.

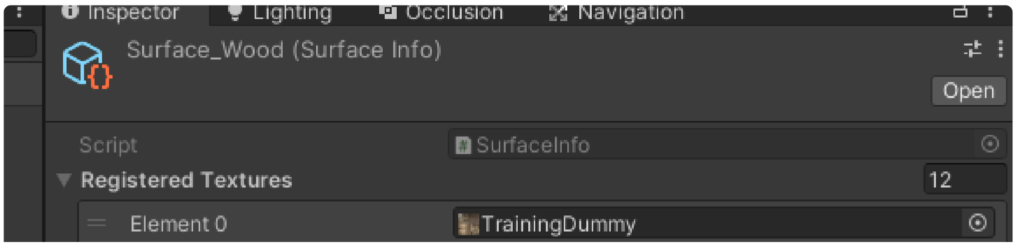
Surfaces

Create Surface

Create

1. Duplicate one of the existing surfaces or create a new one at: **"PolymindGames / SurvivalTemplatePro / Resources / Surfaces"**
2. You can configure your new surface either through the inspector or the management window: **"Tools / STP / Surface Management"**

Configure

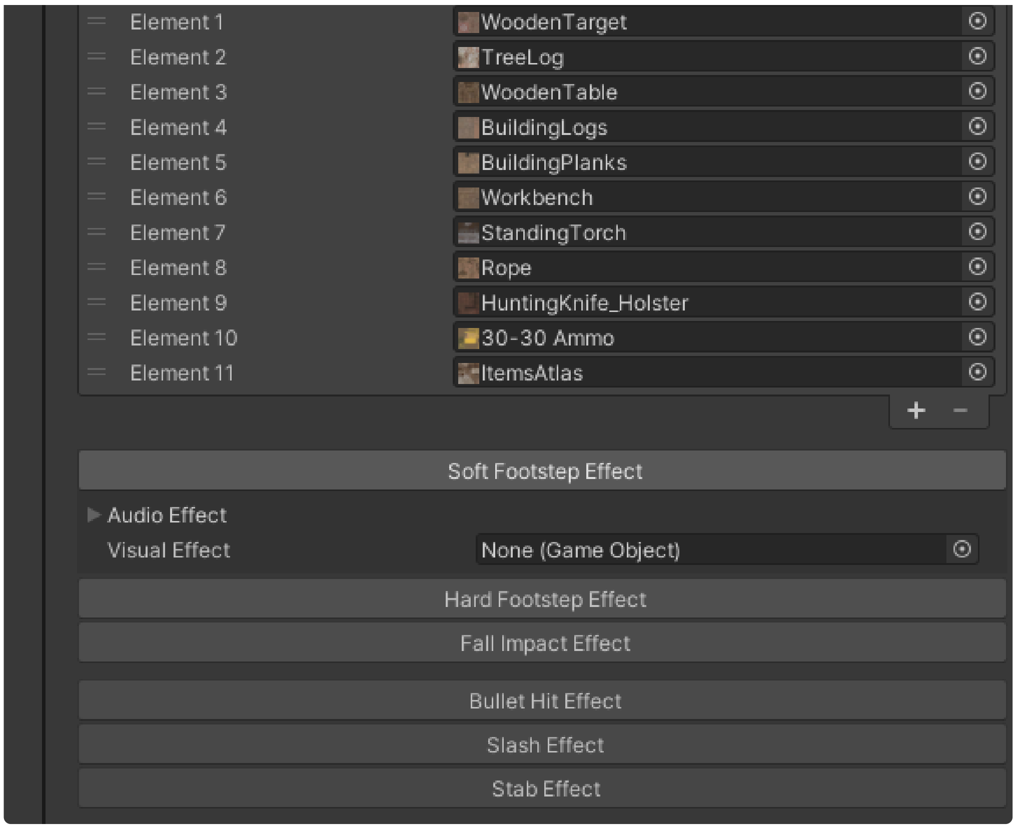


Surface_Wood (Surface Info)

Script SurfaceInfo

Registered Textures 12

Element 0 TrainingDummy



Element 1	WoodenTarget
Element 2	TreeLog
Element 3	WoodenTable
Element 4	BuildingLogs
Element 5	BuildingPlanks
Element 6	Workbench
Element 7	StandingTorch
Element 8	Rope
Element 9	HuntingKnife_Holster
Element 10	30-30 Ammo
Element 11	ItemsAtlas

Soft Footstep Effect

Audio Effect

Visual Effect: None (Game Object)

Hard Footstep Effect

Fall Impact Effect

Bullet Hit Effect

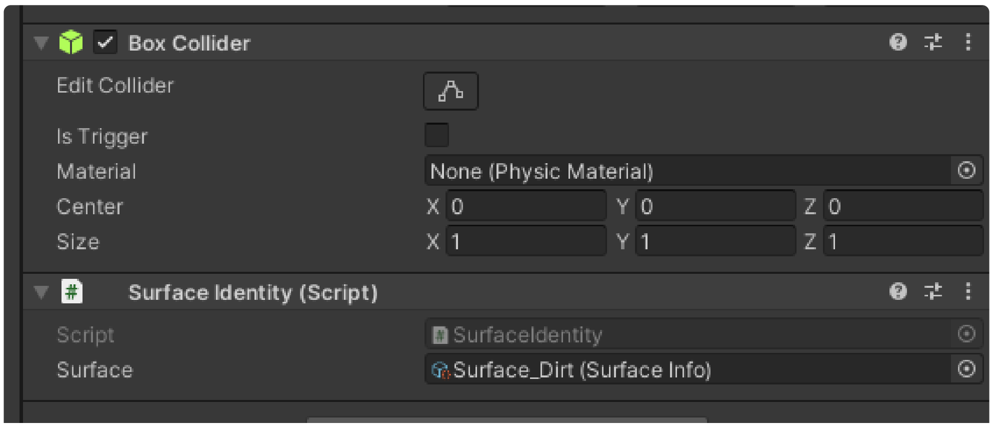
Slash Effect

Stab Effect

1. **Registered Textures:**
   Specify which textures correspond to this surface.
2. **Foot Steps**
   Soft and Hard footsteps (usually walk and run respectively)
3. **Fall Impact**
   Fall impact effects and audio that correspond to this surface.
4. **Bullet Hit Effect**
   Bullet hit effects and audio that correspond to this surface.
5. **Slash & Stab**
   Slash & Stab effects that correspond to this surface.

Note:

If your collider is not a mesh collider (e.g. box, sphere etc.) you'll need to add a surface identity component to it instead of registering it's corresponding texture.
After adding the component select a corresponding surface info.



Box Collider

Edit Collider	
Is Trigger	
Material	None (Physic Material)
Center	X 0 Y 0 Z 0
Size	X 1 Y 1 Z 1

Surface Identity (Script)

Script	SurfaceIdentity
Surface	Surface_Dirt (Surface Info)

Code Examples

This function leverages the **SurfaceManager** to spawn an effect that corresponds to the hit surface type (e.g. grass, dirt etc.)

```
if (Physics.Raycast(ray, out RaycastHit hitInfo, m_MaxDistance, m_RayMask, QueryTriggerInt
{
    // effect type
    SurfaceEffects effectType = SurfaceEffects.BulletHit;

    // audio volume multiplier
    float audioVolume = 1f;

    // useful for parenting decals to moving targets
    bool shouldParentEffect = false;

    SurfaceManager.SpawnEffect(hitInfo, effectType, audioVolume, shouldParentEffect);
}
```