

# Welcome

The **Combat** module is now available for purchase on the Unity Asset store. Visit the asset's [product page](#) for details.

## Combat Demos







# Getting Started

C

100%

The **Combat** module builds on the **Game Creator Shooter** module to add enhanced, game-ready, combat features:

- Proximity-based weapon targeting.
- Target indicator.
- Homing projectile.
- Targeting by visibility.
- Support for destructible targets.
- Weapon Stashes (weapon carrying/swapping feature).
- Melee targeting integration (requires Melee module).

## Dependencies

Combat is an extension for **Game Creator** and the **Shooter** module. BOTH are required - **Combat** will not work without them.

They can be purchased from the Unity Asset Store:

- [Game Creator](#)
- [Shooter](#)

The **Melee** module is an optional dependency. Get it here:

- [Melee](#)

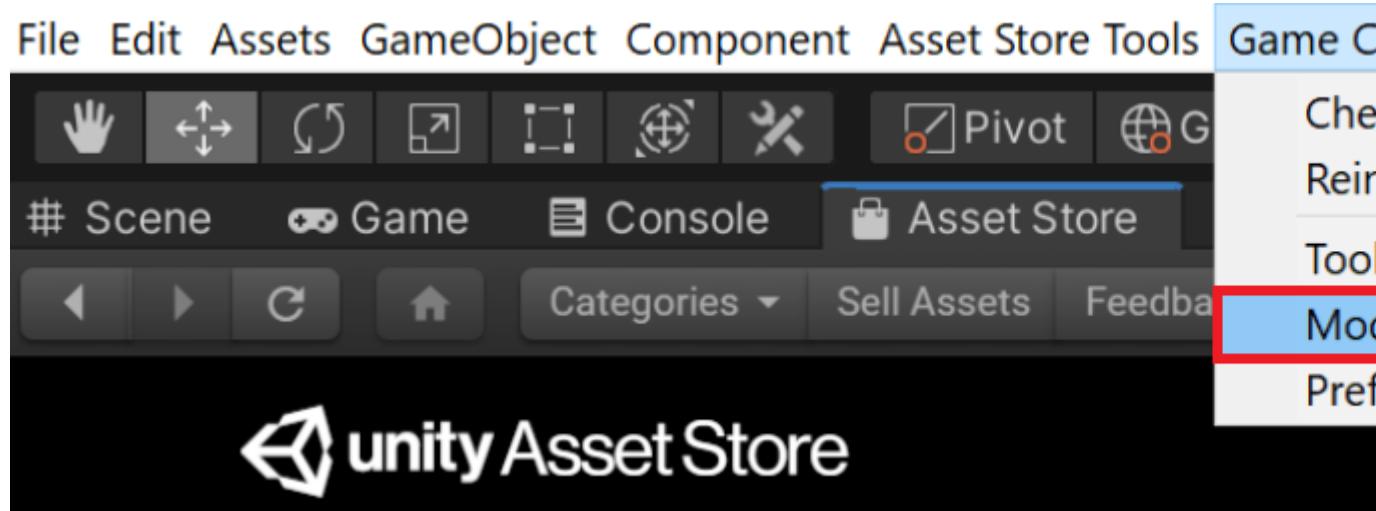
## What's Included

- Full source code.
- An examples module that contains scenes that demonstrate the features listed above.

## Module Installation

After purchasing and downloading the **Combat** module, it must be enabled with the Game Creator Module Manager.

Step 1: Open the Module Manager



Step 2: Enable the Combat Module

## Module Manager



Module Manager



Game Creator Examples



Shooter Examples



Shooter



Combat Examples



Combat



Combat

Backup

### Module ID

com.firechickengames.module.co

### Version

0.3.0

### Author

Ethan Hann

ethan@firechickengames.com

### Description

Enhanced and improved workflow

### Tags

module

game creator

fire chicken

### Dependencies (1)



com.gamecreator.module.sho

**Step 3 (optional): Install the Combat Examples Module**

## Module Manager



Module Manager



Game Creator Examples



Shooter Examples



Shooter



Combat Examples



Combat



Combat Examples

Backup

### Module ID

com.firechickengames.examples.

### Version

0.3.0

### Author

Ethan Hann

ethan@firechickengames.com

### Description

Example scenes that show how to

### Tags

examples

game creator

fire chick

### Dependencies (2)



com.gamecreator.examples.s



com.firechickengames.modu

## Melee Module Integration

 New in 0.5.0

The **Combat** module provides a lightweight integration to allow seamless Shooter/Melee targeting and weapon switching.

To enable the integration, simply enable the **Melee Combat** module included with the **Combat** module.

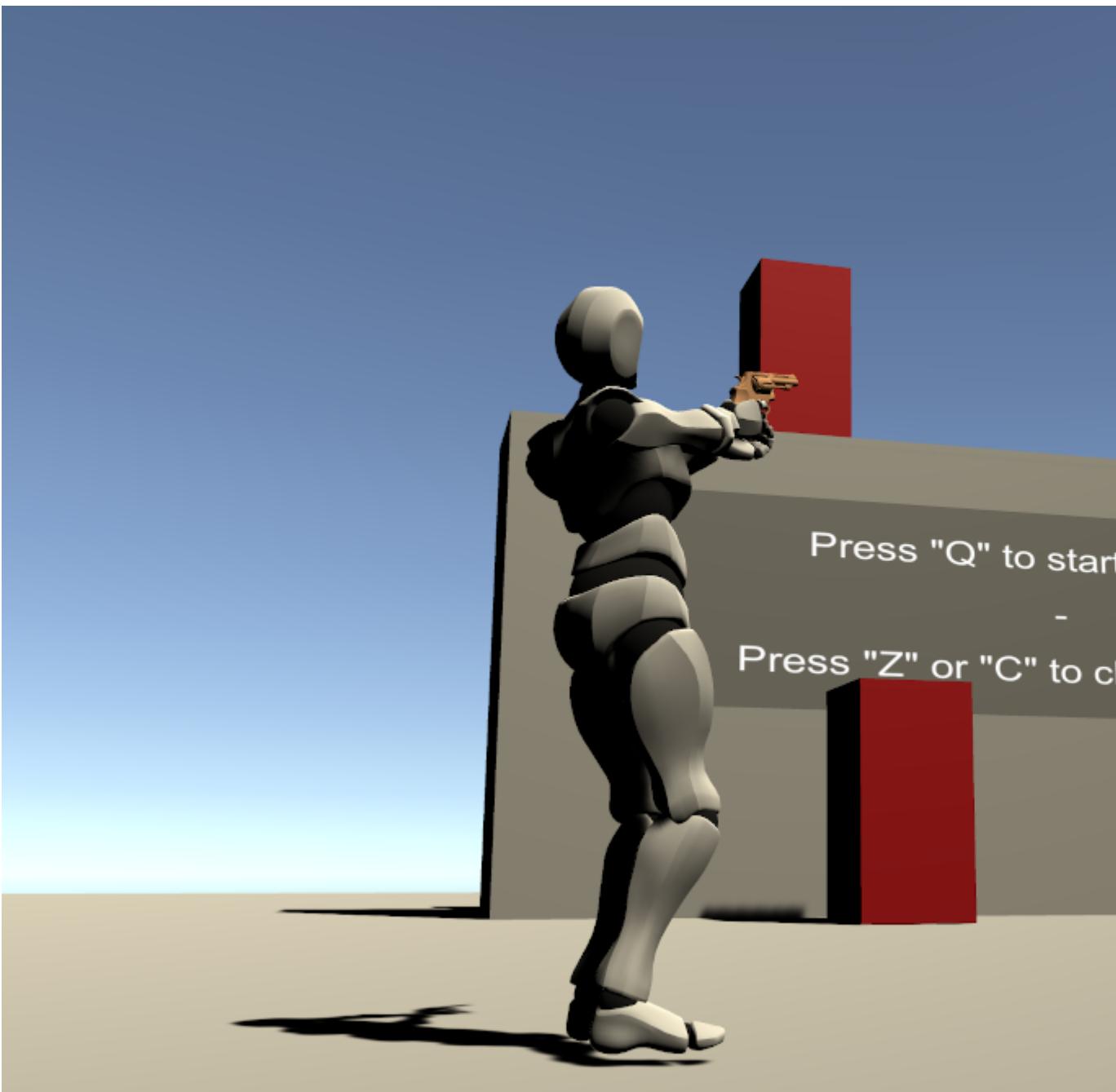
Note that there is also a **Melee Combat Examples** module that demonstrates Shooter/Melee targeting and weapon switching.



# Overview

The **Combat** module provides a proximity-based targeting system. It allows a player to target characters (or other game objects) with the **Targeter** and **Targetable**.

When a player with a **Targeter** component approaches objects with the **Targetable** component attached, the player can use the keys **Q**, **Z** or **C** to switch targets. These keys can be used to switch targets. Please note that these keys can be remapped.





## Targeter Component

Included in the **Combat Examples** module is a prefab called **PlayerTargeter** that demonstrates how to use the **Targeter** component. A **Targeter** component (depicted in the inspector screenshot below) is required. Note that the range of the **Targeter** is dictated by the **Radius** property of the Game Creator Player object.

**Inspector**   **Navigation**

**PlayerTargeter**

Tag Untagged Layer

**Transform**

Position	X 0
Rotation	X 0
Scale	X 1

**Targeter**

Script	
Player	None (Game Object)
Auto Aim At Target	<input checked="" type="checkbox"/>

**Target Visibility**

Only Target Visible To Camera	<input checked="" type="checkbox"/>
Only Target Non Occluded	<input type="checkbox"/>

**Layers To Ignore For Visibility Occlusion**

Size	2
Element 0	Player
Element 1	Targetable

**User Input**

Target Lock Enabled On Key Up	Q
Select Next Target On Key Up	C
Select Previous Target On Key Up	Z

**Sphere Collider**

Edit Collider	
Is Trigger	<input checked="" type="checkbox"/>
Material	None (Physics Mater...)

## Auto Aim At Target

Automatically aiming at a target can be disabled with the **Auto Aim At Target** option - the character will still be locked for some games.

## Target Visibility

 **New in 0.4.0**

The component's target visibility options allow for target selection to be limited by what the camera can see.

### Only Target Visible To Camera

If enabled, only targets possibly visible to the camera (i.e. in its view frustum) are targetable.

### Only Target Non-Occluded

If enabled, targets hidden behind objects are not targetable.

Note that this option is turned off by default because the player and targetables need to be on dedicated layers which

### Layers To Ignore For Visibility Occlusion

The layers to ignore when determining target visibility when [Only Target Non-Occluded](#) is enabled.

Typically, there should be two layers:

- A "Targetable" layer that contains all targetable objects.
- A "Player" layer that contains the player.

## User Input

The **User Input** section of the **Targeter** component allows the keys that control target locking and switching to be cu



# Targetable Component

Making game objects targetable using the **Combat** module's **Targetable** component is trivial for Game Creator Charac

## Basic Setup

### Characters

To make a character targetable, simply add the **Targetable** component to it.

Inspector

Navigation



Jerry

Tag Untagged



Layer

Targetable

Prefab

Open

Select

Overrides



Transform

Position

X -1.52

Y -0.6

Rotation

X 0

Y 181.639

Scale

X 1

Y 1



Character



Character Animator



Character Controller



Targetable

Script

Targetable

### Targetability

Can Be Targeted

Local Variable

Game Object ▾

None (Game O

Visibility Detection Renderer

None (Renderer)

### Active Target Indicator

Show Indicator



Indicator Prefab

ActiveTargetIndicator

Indicator Offset

X 0

Y 2.1

Indicator Text Content

Local Variable

Invoker



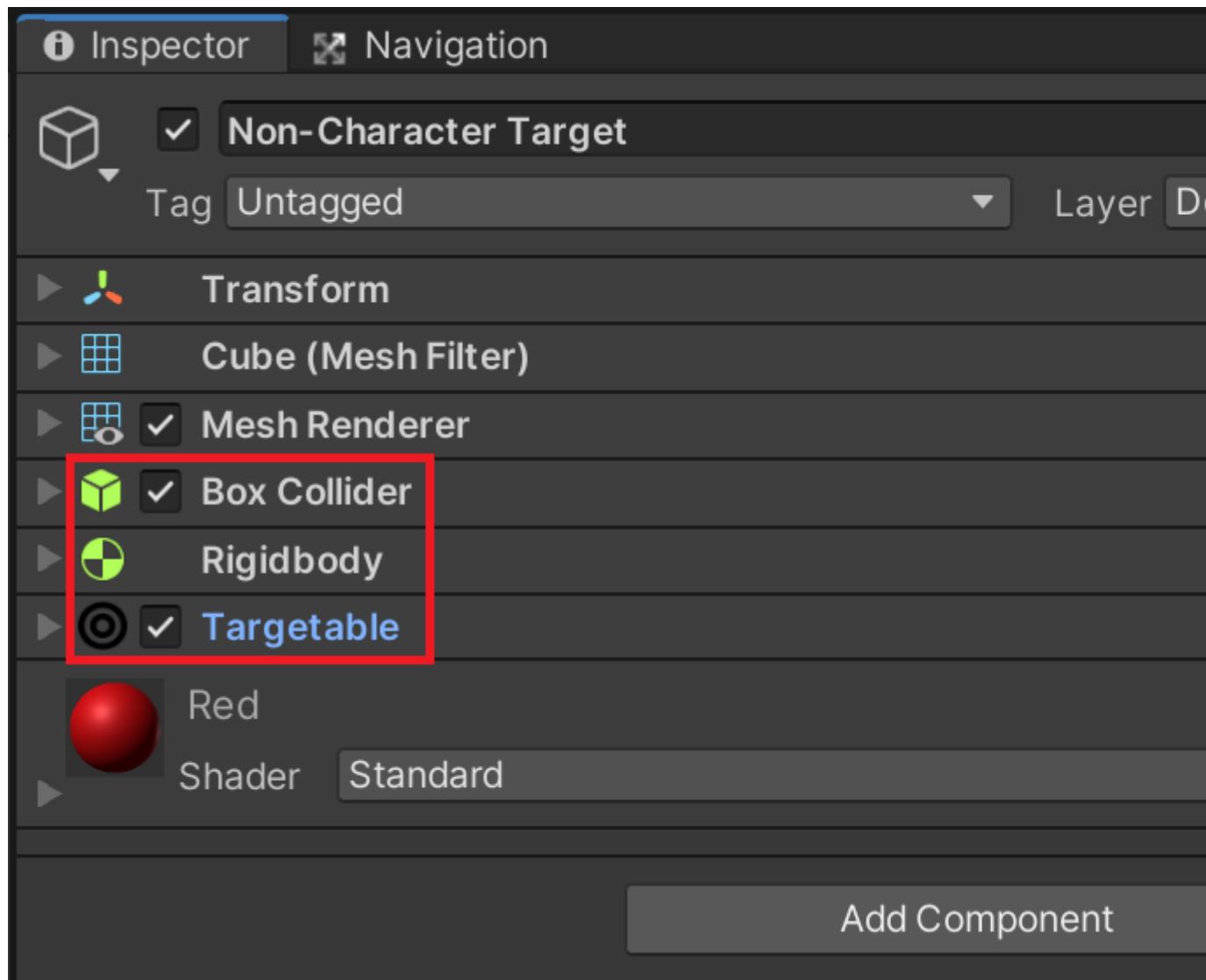
None (Game O

MyName

Indicator Text Color

## Non-Character Game Objects

Any game object can be targetable if it has the **Targetable**, **Rigidbody**, and **Collider** components.



## Making a Target Untargetable

The **Targetable** component contains a boolean Game Creator Variable property, called **Can Be Targeted**, that can ma

For example, once the target has been defeated, it is likely desirable for the player to automatically stop targeting it an

This is accomplished by:

1. Adding a boolean **Local Variable** to a character (e.g. "IsAlive").
2. Assigning the boolean variable to the **Can Be Targeted** property.
3. In the character's **On Receive Shot Actions**, set the boolean value to false - this will deselect the target and make

The **Combat Examples** module's **Example4-KillableCharacters** demo scene contains a pre-configured **KillableChar**

### Visibility Detection Renderer

This property is used to determine if a **Targetable**'s mesh is [visible](#) to a **Targeter** component. It does not normally need to be set if the **Targetable** component's parent object does not have a mesh renderer in its hierarchy, or a mesh renderer other than a **Targetable**.

If visibility features are not used, this property can be ignored completely.

### "On Target Become Untargetable" Trigger

#### New in 0.4.0

There is also a trigger called "On Target Become Untargetable" that allows for actions to be executed when the target becomes untargetable. It showcases how to use this trigger to implement destructible targets.

## Advanced Options

### Active Target Indicator

The **Targetable** component provides an "indicator" feature that highlights the currently targeted game object. The component has a **Targeter** field that defines which targeter will be highlighted.

Jerry



## Text

An indicator can have custom text, defined via a Game Creator **Global/Local Variable**. Practically speaking, it almost Local Variable would then be configured on the instance of the prefab when used in a scene.

**Inspector** **Navigation**

**TargetableCharacter**

Tag Untagged Layer Default

**Transform**

**Character**

**Character Animator**

**Character Controller**

**Targetable**

Script  Targetable

Can Be Targeted Local Variable

Game Object ▾ None (Game)

**Active Target Indicator**

Show Indicator

Indicator Prefab None (Game Object)

Indicator Offset X 0 Y 2

Indicator Text Content Local Variable

Invoker ▾ None (Game)

TargetName

Indicator Text Color

**Actions**

On Become Active Target Actions None (Actions)

On Not Active Target Actions None (Actions)

**L Local Variables**

= TargetName

## Prefab

If not set, the Game Creator **Floating Message** prefab is automatically set as the target indicator prefab at runtime. T

**Inspector**   **Navigation**

**TargetableCharacter**

Tag Untagged   Layer Default

Transform

Character

Character Animator

Character Controller

**Targetable**

Script  Targetable

Can Be Targeted  Local Variable

Game Object  None (Game)

**Active Target Indicator**

Show Indicator

Indicator Prefab

Indicator Offset X 0 Y 2

Indicator Text Content Local Variable

Invoker  None (Game)

TargetName

Indicator Text Color

**Actions**

On Become Active Target Actions None (Actions)

On Not Active Target Actions None (Actions)

**Local Variables**

Add Component

## Positioning

The target indicator is positioned relative to the parent game object. By default, the **Indicator Offset** vector will position

**Inspector**   **Navigation**

**TargetableCharacter**

Tag Untagged   Layer Default

Transform

Character

Character Animator

Character Controller

**Targetable**

Script  Targetable

Can Be Targeted  Local Variable

Game Object

**Active Target Indicator**

Show Indicator

Indicator Prefab

Indicator Offset X 0 Y 2

Indicator Text Content Local Variable

Invoker

TargetName

Indicator Text Color

**Actions**

On Become Active Target Actions

On Not Active Target Actions

**Local Variables**

Add Component

## Targeting Actions

A **Targetable** game object can optionally execute actions when it becomes the active target, and another set of actions

Inspector

Navigation



ReactToTargetingCharacter

Tag Untagged



Layer D



Transform

Position

X 0

Rotation

X 0

Scale

X 1



Character



Character Animator



Character Controller



Targetable

Script

Targetable

Can Be Targeted

Local Variable

Invoker

Non

### Active Target Indicator

Show Indicator



Indicator Prefab

None (Game Object)

Indicator Offset

X 0

Indicator Text Content

Local Variable

Invoker

Non

MyName

Indicator Text Color



### Actions

On Become Active Target Actions

OnBecomeActiveTa

On Not Active Target Actions

OnNotActiveTarget

### **"On Become Active Target"**

When the target becomes the active target, these actions can (for example) make the target crouch. A more practical flee).

**Inspector**   **Navigation**

**OnBecomeActiveTargetActions**

Tag Untagged   Layer Default

**Transform**

Position	X 0.9536481	Y 1.01	Z
Rotation	X 0	Y 178.361	Z
Scale	X 1	Y 1	Z

**Actions**

=  Change ReactToTargetingCharacter state in Layer1

Character	Character
	 ReactToTargetingCharacter (Character)
Action	Change
State	State Asset
State Asset	 Crouch (CharacterState)
Mask (optional)	None (Avatar Mask)
Weight	
Layer	Layer 1
Transition Time	0.25
Speed	Value
	1

Add Action 

Run in Background

Destroy After Finishing

Add Component

### **"On Not Active Target"**

Related to the previous section, when the target is changed or targeting is disabled, this action will reset the target ch

**Inspector**   **Navigation**

**OnNotActiveTargetActions**

Tag **Untagged** Layer **Default**

**Transform**

Position	X 0.9536481	Y 1.01	Z 0
Rotation	X 0	Y 178.361	Z 0
Scale	X 1	Y 1	Z 1

**Actions**

=  Reset ReactToTargetingCharacter state in Layer1

Character	Character
	ReactToTargetingCharacter (Character)
Action	Reset
Layer	Layer 1
Transition Time	0.25
Speed	Value
	1

Add Action

Run in Background   
Destroy After Finishing

Add Component



# Weapon Stashes

 **New in 0.5.0**

**Weapon Stashes** is a lightweight inventory system to assign weapons to a Player Character and allow the player to s

## Weapon Stash Component

A **Weapon Stash** must be added to a game object (usually the player):

Inspector    Navigation    Inspector

Player

Tag Player    Layer Player

▶  Transform

▶   Player Character

▶   Character Animator

▶   Character Controller

▶  HookPlayer

▶   Player Shooter

▶   Character Melee

▼   Weapon Stash

Target	Player
UI	None (Weapon Stash Ui)
Stashed Weapon	None (Scriptable Object)
Stashed Ammo	None (Ammo)

▶ Weapons

Add Component

### Adding a Weapon

There are two actions for adding weapons: Give Shooter Weapon and Give Melee Weapon. When configuring these

## Changing Weapons

The current weapon can be switched to the next (or previous) weapon in the stash with the **Cycle to Next Weapon** ac

## Weapon Stash UI Component

The Weapon Stash UI component allows a stash's current weapon and ammo (if any) to be displayed on screen. Thi

The **Combat Examples** module includes a **WeaponStashUI** prefab that demonstrates how to use the component:





# Homing Projectiles

As the name suggests, a **Homing Projectile** seeks its target even if the weapon firing the projectile is not pointed directly at it.

Setup is trivial. Simply attach the Combat module's **Homing Projectile** component to any projectile. The component object contains a Rigidbody component.

The **Propulsion** settings control the movement behavior of the projectile:

- **Seconds To Wait Before Propelling**: Delays the propulsion of the projectile by a number of seconds. A value of 0 means it immediately begins propelling.
- **Maximum Turn Angle**: The maximum angle, in degrees, that the projectile will turn while homing in on its target.
- **Velocity**: How fast the projectile moves toward its target - this should likely match or exceed the max velocity of the projectile's Rigidbody component. Setting this to 0 will effectively override the projectile ammo's min/max velocity.



## Rigidbody Gravity

Turning off gravity on the Rigidbody is optional, but might be desired depending on the specific projectile.

**Inspector**

**Navigation**

**Inspector**



**HomingArrow**

Tag Untagged



Layer



**Transform**



**Pool Object**



**Homing Projectile (Script)**

Script

HomingProjectile

Ammo Rigidbody

None (Rigidbody)

### **Propulsion**

Seconds To Wait Before Propelling

0.5

Maximum Turn Angle

10

Velocity

25



**Rigidbody**

Mass

1

Drag

0

Angular Drag

0.05

Use Gravity



Is Kinematic



Interpolate

Interpolate

Collision Detection

Continuous Specula

► Constraints

► Info



**Box Collider**



# Roadmap

## v0.7.0

### **Targeting Features**

- New public API for Targeter component: -- HasTarget -- GetCurrentTarget -- IsCurrentTarget -- SetCurrentTarget -- SetTargetingEnabled -- ToggleTargetingEnabled -- CycleToNextTarget
- New actions to enable, disable, and toggle targeting.
- New action to switch to next target.
- New "On Target Changed" trigger.
- Accessibility Module (by Pivec Labs) mobile touchstick compatibility.
- Mouse targeting w/ optional hover indicator (i.e. Targetable can be selected with mouse).
- Targeter's built-in user input controls can now be disabled.
- Targeter can now be configured to not auto-acquire first/next target.

### **Spawning Features**

- Spawn by Weight (i.e. random spawn chance).

### **Bug Fixes**

- Targetable's internal event now correctly cleaned up.
- When created from GameObject menu, a Spawner's collider now is set as a trigger by default.

## v0.6.0

- Spawner component.

## v0.5.0

- Melee targeting integration.
- Weapon Stashes (weapon carrying/switching feature).

## v0.4.1

- Removed .blend files that caused an issue when Blender was not installed.

## v0.4.0

- Targeting by visibility.
- Support for destructible targets.

## v0.3.0

- Proximity-based weapon targeting.
- Target indicator.
- Homing projectile.

## Possible Future Features

- Aim-assist.
- Headshots/body part targeting.
- Simplified death action for targetable characters.
- In editor weapon positioning.
- Proximity mines.
- Dual-wielding.
- When a target is defeated, switch to its nearest neighboring target (instead of closest to the player).
- Left/right target cycling (instead of nearest/farthest from player).
- AI targeting.