Throw Object 3D: Mobile, Desktop, FPS — Throw Control 3D

• April 4, 2019 🔥 542

Throw Control 3D (Throw Object) is highly customizable Advanced Throwing System for Unity. You can throw object forward or in any direction with custom force, center of mass and 100 other parameters.

AVAILABLE ON THE
ASSET STORE

Features

All modules are designed independently to keep this asset extendable & easy to understand:

- Cross-platform: Mobile, Desktop (Mouse);
- 2 Throwing Modes: “Flick” & “Click Or Tap”;
- 6 Demos (including FPS);
- 5 Events;
- 19 customized Throwing Objects;
- Easy implementation of your own Throwing Objects;
- Tons of Customizable Parameters;
- Dynamic Sound System (depends on Speed of Throwing Object);
- Layer Changing (actual for quick Throwing to neutralize mutual collisions);
- Fade Out;
- Object Pool.

5 Events

- OnThrow,
- OnNextThrowGetting,
- OnReset,
- OnFadeOut,
- OnInitialized.

Throw Object — Tons of Customizable Parameters

> *Input sensitivity, force, torque, delays, position, rotation, etc.*
>
> - ★ *For all Throwing Objects at once,*
> - ★ *For each Throwing Objects individually.*

19 customized Throwing Objects

- ★ Weapons:
  - ★ Axe & Sword & Broken Sword,
  - ★ Machete & Sickle,
  - ★ Rune Hammer & Rune Sword,
  - ★ 5 Combat Knives (paid separately);
- ★ Balls: Basketball, Brick Ball (Grey), Brick Ball (Red);
- ★ Miscellaneous: Coin, Shovel & Horseshoe, Tree.

6 Demos

- ★ 2 FPS Demos integrated with FPSController from Standard Assets by Unity Technologies (Desktop):
  - ★ Throw force takes into account the speed of the player's movement;
- ★ 4 Demos with a static position of the player in "Flick" & "Click Or Tap" modes (Mobile & Desktop).

Use Cases

- A game in the style of Pokemon GO (iOS, Android),
- A game in the style of Can Knockdown (iOS, Android),
- FPS with Throwing Objects,
- AR Basketball game,
- AR or VR game with Throwing Objects,
- Survival Game,
- Any of your Bold Fantasies.

Modes

The higher click (or tap) position on the screen, the greater the strength of the throw.

System completely works on desktop as well as on mobiles: you can throw object with mouse or with finger.

Click Or Tap

In this mode, game object is throwing in the direction of the click (desktop) or tap (mobile).

FPS

For FPS game there is an option for fixing input position in a specific place on the screen (by default in the center of the screen).

Flick

In this mode, you must take the Game Object and throw it in the style of Pokemon GO game (iOS, Android) (by flicking it from the bottom of the screen up toward the target).

Is Full Path For Flick?

If it’s false then it allows fast flicks only: positions in the last and previous frames are taken into account.

Scripts

Throwing system consists of three control scripts.

Throwing Object

The script contains basic functions and parameters to operate Throwing Object.
This script must be attached to each Throwing Prefab (check tutorial).

Throw Control

This is the main script that manages throw & all Throwing Objects on a scene.
The script uses disabling and enabling the following nodes: Colliders, Triggers, Renderers.

Coroutines

The only script that uses coroutines calls.

Because of coroutines behavior, it is impossible to place some methods inside Throwing Object script.
OOP does not work in this case, but we have a stable throwing with Object Pool.

Random Object Pooler

Open Documentation of Random Object Pooling System.

Colliders

Rotation (Torque) & Center of Mass

There are some nuances with Rigidbody.centerOfMass when rotation.

Symmetric Mesh

If you use Mesh Collider for Throwing Game Object and your mesh is ideally symmetric then it is the perfect case and game object will throw naturally.

Asymmetric Mesh

If you use Mesh Collider for Throwing Game Object and your mesh is asymmetric then you will have unnatural throw behavior when rotation.

Solution 1: Custom Center of Mass

You can indicate Custom Center of Mass manually in the Editor.
Package contains some examples with Custom Center of Mass.

There is no magic function which calculates right Center of Mass.

Solution 2: Simple forms of colliders

You need to use one or more simple forms of colliders (box, sphere, etc.) placing them on the same Throwing Game Object and along the straight line. So in this case, we have a right center of mass & right rotation.

Convex Mesh

Unity allows you to use only convex mesh colliders. In some cases, it is not what you need because the collider covers more space, than the object itself in some points.

In order to quickly & automatically get a convex mesh collider for complex objects, use Technie Collider Creator.

Tutorial

Getting Started with Throw Control (Throw Object 3D)

Unity 2018.3 and higher

All Free Unity Assets are already included in Unity Package.

Option

If you need, HQ PBR Combat Knives pack must be purchased separately and then imported into the project. Special scene from this video is already customized for knives pack.

Unity 2018.2

For legal reasons, I can’t include some assets in the package, but you can get them for free separately.

If you have any issues with the first launch then just email me (info@makaka.org) with Invoice Number and I will send you complete project.

I used the next software for this tutorial:

- ★ Unity 2018.2.1;

Folders in the package by Default:

- ★ Makaka Games;
- ★ PostProcessing.

Steps

1 Download and import Throw Control into Unity;

2 Download and import 3rd-Party Assets:

- Standard Assets by Unity Technologies (to open FPS Demo Scene integrated with FPSController),
- Machete & Sickle, Rune Hammer & Rune Sword, Desert Environment by DevAssets.com & Brackeys,
- HQ PBR Combat Knives pack (paid separately) by KumaBeer.

3 Reopen Unity project;

4 Test in the Unity Editor;

5 Build for Mobile or Desktop.

How to create your own Throwing Object Prefab?

I used Unity 2017.3.1p4 and Demo Scene (included in the package) for this tutorial.

1. Create Prefab with a template:
    1. Drag one of the customized throwing prefabs into the scene,
    2. Name your Game Object,
    3. Drag named Game Object into Project View to create your own Prefab;

2. Select Game Object on the Scene & Customize it:
    1. Indicate Mesh in Mesh Filter component,
    2. Indicate Material in Mesh Renderer component,
    3. Customize Colliders:
        1. Indicate Mesh in Mesh Collider component (or in more suitable collider component);
    4. Option: Customize Center of Mass for more natural Throwing;

3. Customise the Transform component as you wish;
4. Option: Customize Fade Out (see below);
5. Apply Changes to Prefab;
6. Delete Game Object from Scene;

7 Option: Add Asset Label for Prefab to convenient searching in Project View;

8 Add Prefab to Random Object Pooler;

9 Throw your Game Object. ?

Option: Fade Out Customizing

1 Change Shader of Material to Shader with the Main Color property (e.g. Legacy Shaders > Bumped Diffuse);

2 Duplicate Material in the Project View and name it (i.e [MaterialName]_FadeOut);

3 Change Shader of Duplicated Material:

- [Standard]: Rendering Mode > Fade;
- [Legacy]: to shader from the Transparent Shader Family (e.g. Legacy Shaders > Transparent > Bumped Diffuse);

4 Indicate both materials in the Fade Out Section of Throwing Objects Script in Throwing Prefab.

How to change Spawn Position of Throwing Objects?

See “Position” section in the Editor of ThrowingObject.cs for each Throwing Prefab individually.

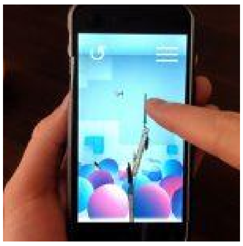
★ [X, Y]: “Position In Viewport On Reset”,

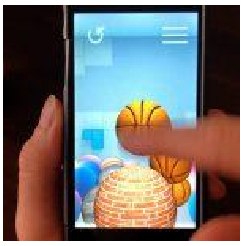
★ [Z]: “Camera Near Clip Plane Factor On Reset”.

Screenshots

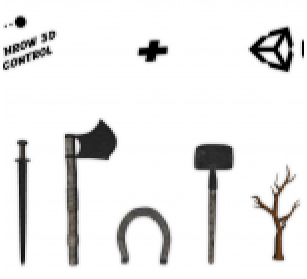
FPS Demo on Desktop

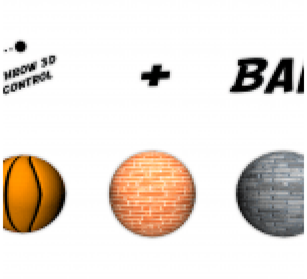
Mobile Demo

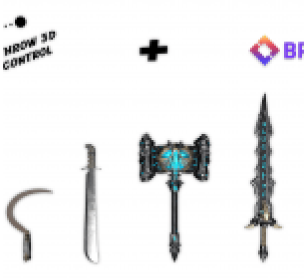




Weapons Showcase







Unity Editor

Licenses

This asset is governed by the Asset Store EULA; however, the following components are governed by the licenses indicated below:

1. Survival Weapons, Fantasy Rune Weapons, Desert Environment
Usage Guidelines by DevAssets.com
© 2018 DevAssets

2. Standard Assets, The Blacksmith: Environments, Adventure — Sample Game, Tanks! Tutorial
Asset Store EULA
© 2018 Unity Technologies

3. HQ PBR Combat Knives pack
Asset Store EULA
© 2018 KUMABEER

Support

First of all, read the latest docs online.
If it didn't help, get the support.

Changelog

Check the current version on Asset Store.
The latest versions will be added as soon as possible.

3.12:

- ★ Unity 2018.3 version;
- ★ Now All Free Assets are already included in Unity Package. Enjoy.

3.11:

- ★ New Customized FPS Scene with Desert Environment (by Brackeys & DevAssets.com) & HQ PBR Combat Knives pack (paid separately) by KumaBeer.

3.10:

- ★ HQ PBR 5 Combat Knives pack (paid separately) by KumaBeer.

3.9:

- ★ Machete & Sickle by DevAssets.com & Brackeys.

3.8:

- ★ Rune Hammer & Rune Sword by DevAssets.com & Brackeys.

3.7:

- ★ Move cameraNearClipPlaneFactor to ThrowingObject.cs to customize the whole position (X, Y, Z) of Throwing Object prefab in one place;
- ★ Fix enabling and disabling colliders for compound Throwing Prefabs.

3.6:

- ★ Unity 2018.2 version;
- ★ Fix errors: When Flick Mode you just click on the object without finger position change.

3.5 (New 10 customized Throwing Objects):

- ★ Weapons: Axe, Sword, Broken Sword,
- ★ Throw Balls: Basketball, Brick Ball (Grey), Brick Ball (Red),
- ★ Miscellaneous: Coin, Shovel, Horseshoe, Tree.

3.4 (Layer Changing in Throw Control):

Actual for quick Throwing to neutralize mutual collisions.

- Customizations:
    - ON / OFF,
    - Layer Mask on Throw,
    - Layer Mask on Reset,
    - Delay.

3.3 (Throw Customizations in Throw Control):

- Input Sensitivity,
- Force Factor Extra,
- Torque Factor Extra,
- Torque Angle Extra,
- Parent on Throw.

3.2 (“Flick” Mode Customizations in Throw Control):

- Option: Is Full Path:
    - If it’s false then it allows fast flicks only (positions in the last and previous frames are taken into account);

- Lerp Time Factor On Touch.

3.1:

- FPS Demo integrated with FPSController from Standard Assets by Unity Technologies (Desktop);
- Throw force takes into account the speed of the player’s movement;
- Fixing input position in a specific place on the screen (by default in the center of the screen).

3.0 (New Architecture: Throw Control):

Main Control Script (Throw Control) operates Object Pool & All Throwing Objects now (Throwing Object Script is attached to each Throwing Object prefab).

Work Scheme:

1. Initialize Pool,
2. Initialize Throwing Objects,
3. Get the First Throw,
4. Get Next Throw.

- Customizations in Throw Control:
    - Throw Mode;
    - Events with Delays:
        - OnInitialized,
        - OnThrow,
        - OnNextThrowGetting,
        - OnReset,
        - OnFadeOut.

2.6 (Rotation Customizations):

- Option: Rotate Object in Throw Direction;
- Rotation on Reset:
    - Default,
    - Last,
    - Random,
    - Custom.

2.5:

- Position in Viewport On Reset.

2.4 (Center of Mass Customizations):

- Custom Center of Mass,
- Logging of Center of Mass by Default.

2.3 (Torque Customizations):

- Torque Axis,
- Torque Angle,
- Torque Factor,
- Max Angular Velocity at Awake.

2.2 (Force Customization):

- Force Factor,
- Force Direction Extra.

2.1:

- “Click or Tap” Throwing Mode for Desktop & Mobiles;
- Customization:
    - Throwing Mode.

2.0 (Object Pool — Throwing of Multiple Objects):

- Prefabs Using:
    - Single (actual for Testing target prefab; None => Multiple),
    - Multiple;

- Generation order for Multiple Prefabs:
    - In random order,
    - In the right order;

- Customizations:
    - Pool Parent,
    - Init pooled amount,
    - Event (OnInitialized).

1.3 (Dynamic Sound System):

Sound System depending on Speed of Throwing Object.

- ⭐ Customizations:
  - ⭐ Array for randomly selecting Sounds,
  - ⭐ Speed Clamping,
  - ⭐ Pitch (Minimum & Factor),
  - ⭐ Volume Factor.

1.2 (Fade Out option for Thrown Objects):

- ⭐ Customizations:
  - ⭐ ON / OFF,
  - ⭐ Delay,
  - ⭐ Speed,
  - ⭐ Materials,
  - ⭐ Event (OnInitialized).

1.1:

- ⭐ Desktop version for Mouse. Use it for Standalone apps or for testing of mobile apps in Unity Editor.

1.0 (“Flick” Throwing Mode for Mobiles):

- ⭐ Throwing of Single Object in “Flick” Mode for Mobiles (AR Basketball GO — Unity Asset).