My Grabber – 3D Objects Grabber - Documentation

Welcome to the documentation of ***My Grabber – 3D Objects Grabber***! Firstly, I would like to tell you I am very, very happy that you gave my asset a chance, and I really hope you enjoy it!
For any questions, bug reports or suggestions please contact me at my discord "Carlinhu#3159".
I'd be tremendously grateful if you could rate my package in your asset store downloads or leave a review on this asset page.

Getting Started

Setting up

The asset contains one main script called MyGrabber.cs, a secondary script called Interaction.cs, an Interface called IMyGrabberInputHandler.cs, two generic scripts to handle the player and camera controls, and one simple script to handle some objects properties visualization. The main script will allow you to grab any 3D objects on your scene, based on the user configurations that you can setup on the inspector, which are detailed below.

- MyGrabber.cs: This will be the main script to add to your player camera, which will handle all the grab system controls.
- Interaction.cs: This will be the script to handle the interaction behavior and inputs that the MyGrabber.cs will need.
- IMyGrabberInputHandler.cs: This will be the interface that handles the inputs.

MyGrabber.cs

Properties:

Property	Type	Description
grabType	Enum	The type of drag that you want.
Force Drag		
grabForce	Float	The force used to drag the objects.
throwForce	Float	The force used to throw the objects.
useObjectMassOnThrow	Bool	If should use the object's mass on throw calculation. (Force mode)

Property	Type	Description
Velocity Drag		
grabVelocity	Float	The velocity to drag the objects.
throwVelocity	Float	The velocity to throw objects.
parentObject	Bool	If the object should be parented to cameras transform (only for velocity).
useObjectMassOnThrow	Bool	If should use the object's mass on throw calculation. (Velocity mode)
objectRotationSpeed	Float	The object rotation speed.
grabPos	Transform	The transform that represents the position of the grabbed object.
maintainObjectOffset	Bool	If the object position on grab should maintain its original position.
returnToParent	Bool	If the object should be returned to its original parent when relased.
grabPosOffset	Vector3	The Vector3 used to create and position the grabPos object, if null.
breakDistance	Float	The maximum distance an object can get from grabPos before breaking the connection.

Interaction.cs

Properties:

Property	Type	Description
pointer	Image	The pointer default image.
pointerCircle	Image	The pointer interaction image.
interactionLayer	LayerMask	The layer of the interactable objects.
distanceToInteract	Float	The maximum distance to interact.
holdForGrab	Bool	If the grab input should be held or not.
grabInput	KeyCode	The key used to grab objects.
throwInput	KeyCode	The key used to throw objects.
rotationKey	KeyCode	The key used to enable object rotation.

Using the asset

- The asset comes with a generic player and camera controller, with the grab script attached to the camera. If you want to use your own camera controller, feel free to attach the MyGrabber.cs script to your camera and correctly assign the empty fields. If you are creating your own controller, remember to create the grabPos object under your camera controller and assign it to the grabPos field, or you can simply use the grabPosOffset to create the object programmatically on the chosen position.
- There are three different grab behaviors: Force, Velocity and Transform. The Force mode will use the mass of the object on the grab, so it is very useful to handle situations with heavy or light objects. The Velocity mode will just apply some velocity to the object, independent of its mass. The Transform mode will simply apply the position to your grabbed object, without the needs of physics. So, depending on your game mechanics, you can choose between these modes and tweak some values to fill your needs.

Integrations

Since the version update 1.3, the My Grabber system became easier to implement with other controllers. The Interaction class have full control of the grab system, and this will be the place to make any changes if you have a different input handler like eg neo's, new Unity Input System, Rewired, etc).
The NeoFPS toolkit is fully compatible and has a repository with everything that you need to make your assets work together smoothly.

On the demo scene you will find interactable objects to play with and tweak the grab values to fit your needs. If you have any question, feel free to contact me!

This asset and documentation were created by Carlos Menezes Concencio