VR Grabbing & Climbing

Unity package, by: VRmind

Table of contents:

1. Grabing

1.1 Introduction

Grabing is a module that enables player to pick up, hold, move and throw object around the scene. Grabbed object is still subject to physical simulation, it means that the object will collide with walls, hit enemies and bounce off colliders. This is achieved by separation of physics part and grabing part. Physics part is a nonkinematic rigid body of the GameObject and it's colliders responsible for physics interactions with environment, whilst grabing part is responsible for grabbing control of the object. Grabber part's collider in many cases has diffrent shape and/or size than physics' collider for example, you may only put grabing collider on the handle of hammer, not on a whole, allowing player to grab it only with the handle. Grabbing package is made out of two parts, **Grabber module** and **Grab Item module.**

1.2 Grabbing module

Grabbing module is attached to players hand (vive controllers by default). It consist of:

- Collider (sphere by default) – when grabber's collider overlaps with GrabItem collider, grab can occur,
- ColliderGrabber (script) – handles player input as well as colliders events,
- RigidBody –kinematic, always disable "use Gravity",
- whenever grab occurs there's also unity joint attached (varying depending on GrabItem script).

Input handling

By default players input is controller's trigger. Pressing it will enable grabbing, releasing will release held object. Also ColliderGrabber , is responsible for passing references of GrabItem in range to grabber. In case you want to extend these options, write new class that inherits after ColliderGrabber.

Object queuing

Grabber script hold references to every objects that's in range to be grabbed (colliders of both Grabber and GrabItem are overlapping). By default whenever Grabber's collider enters new GrabItem's collider, reference is added and the end of the list. References are removed whenever object exits the range. When player presses the trigger first object on the list is grabbed. That means that first object player put his hand into gets grabbed.

Physics interaction

Grabber's GameObject must be kept in Grabing Layer (which interacts only with itself). Moving object is done by unity's "joint" classes which simulate physics connections (different GrabItem scripts handle different joints). Hands can pass walls and obstacles making it impossible to hold object, which by default interacts with surrounding. That's why Grabber uses three functions to control releasing object: OnReachOut, Release and OnJointBreak. OnReachOut is called whenever held object exits range, Release – whenever trigger is released, and OnJointBreak whenever force applied to joint extends joint's strength, making it possible to distinguish these situations in GrabItem script.

1.3 GrabItem module

GrabItem is a module attached to object that you want to be grabable. To make object both physically simulated and grabable it needs to be made out of (at least) two gameObjects bind together with hierarchy (children/father relation). Physics part needs rigidbody and collider, while GrabItem module consists of:

- collider – which sets area by which object can be held.
- GrabItem script – can be any of GrabItem scripts (all inherits after IGrabItem interface)

CARE – GrabItem module must be in "Grabing" Layer.

Interface

IGrabItem interface defines all functions for every GrabItem script, so all you have to do to create your new script is inherit after this interface. For description of functions implemented by IGrabItem refer to chapter 1.6.

1.4 Setup

Manual

Before Grabber Setup make sure to setup layers (look for chapter 1.6).

1. Import SteamVR package from asset store into your project.
2. Create an empty GameObject as a child to Vive Controller gameObject.
3. To the child add: RigidBody (kinematic, use gravity = false), Collider, Grabber script, ColliderGrabber script.
4. Change child's layer to "Grabing" layer.

Automatic

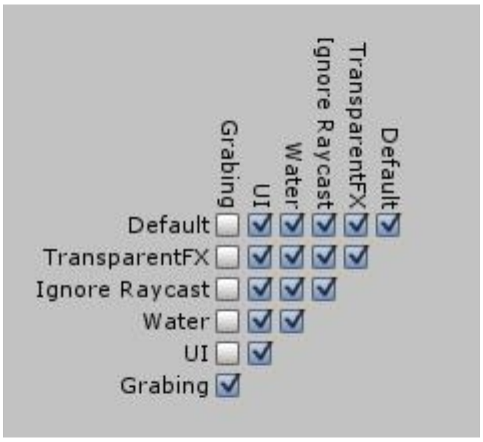
1. Go to Window>Grabber SetUp (Unity toolbar).
2. If not yet done, define layers by picking empty slots index and press "Set up layer" button just below.
3. Press "Set up grabber".

1.5 Layer setup

Manual

Grabber uses 1 layer that interacts only with itself (look example matrix img. 3.1):

1. Create layer in Edit>Project Settings>Tags and Layers. (Unity toolbar)
2. Go to Edit>Project Settings>Physics (Unity toolbar) and disable Grabing Layer's interactions (img. 3.1)



	Grabing	UI	Water	Ignore Raycast	TransparentFX	Default
Default	☐	☑	☑	☑	☑	☑
TransparentFX	☐	☑	☑	☑	☑	
Ignore Raycast	☐	☑	☑	☑		
Water	☐	☑	☑			
UI	☐	☑				
Grabing	☑					

Img. 3.1 Layer matrix.

Automatic

1. Go to Window>Grabber Setup (Unity toolbar)
2. In the input field, fill index of empty layer slot (number between 8-31).
3. Press Set up layer.

1.6 Class description

Grabber

Fields:

- **public Rigidbody rigid** - rigidbody of the hand.

methods:

- **void Start()** - searches for needed components if not added in inspector.
- **public void AddTarget(IGrabItem)** - called by colliderGrabber to add object in reach as target.
- **public void RemoveTarget(GrabItem)** - called by colliderGrabber to remove target, once it leaves hand's reach. this function calls Remove func. in IGrabItem of grabbed object.
- **public void Grab()** – when trigger is pressed and any IGrabItem is in range, this function will call Grab function in target GrabItem script.
- **public void Release()** - called when player releases button, calls OnTriggerUp func. of grabbed object (if grab occured).
- **void OnJointBreak(float)** - called by unity when joint breaks, calls JointBroken(Graber, float) from IGrabItem.

ColliderGrabber

Fields:

- **public Grabber grabber –** grabber attached to this gameObject.
- **SteamVR_TrackedObject device** – reference to controller.(needed for Vive input checks)

methods:

- **void Start()** - searches for needed components if not added in inspector.
- **void Update()** - manges Vive input. Pressing Trigger calls Grab() from grabber. Releasing trigger calls Release() from grabber.
- **void OnTriggerEnter(Collider collider)** - when collision with GrabItem’s part of object occurs, refference is added to Grabber.

- **void OnTriggerExit(Collider collider)** - when collision with GrabItem's part of object ends, target from Grabber is removed.

IGrabItem

- **float GripStrength {get; set;}** responsible for joint strength, makes you implement gripStrength field in every new script.
- **Rigidbody Rigid {get; set;}** rigidbody for this component.
- **void OnGrab(Grabber hand**)- called when player wants to grab object, **hand** is script which called function.
- **void OnReachOut(Grabber hand)** - Called when player moves his hand out of grabbing zone.
- **void OnTriggerUp(Grabber hand)** - Called when player releases button.
- **Joint CreateJoint(Rigidbody rigid)** - creates joint between object and hand.
- **void JointBroken(Grabber hand, float breakForce) -** called by **hand** when joint gets broken by unity physics, usefull for making on release particle, sounds etc.

1.7 FAQ

Q: How can I implement my own grab?
A: If you know how to code, create new MonoBehaviour and implement IGrabItem interface. Use any of existing GrabItem scripts as reference.

Q: I want to use different type of joint!
A: In function GrabItem script, change the type of joint in the "AddComponent" function used in CreateJoint function. And set up joint parameters with code.

Q: I want to change grabbing button on vive, how to do this?
A: in ColliderGrabber script, change the trigger part in> `steamvr_controller.buttonmask.trigger` to any you want (use ctrl+space to get the list).

Q: I don't know how to code and any of the existing GrabItem scripts doesn't suit my taste!
A: Write an email to support, we'll see what we can do. If we get lots of questions like these we may update our package with some new grab for all to use.

2. Climbing

2.1 Climber module

Climber uses Grabing module and GrabItemClimbing to get information about climbing grabs made by player. To pull yourself up, you have to move both hands at the same time in same direction, moving them in diffrent directions will result in hands getting of the grip and in the result player's falling. Climbing in fact moves whole CameraRig by the average movement of both hands.

Climbing algorithm works this way:

- if player holds himself by 2 hand he can climb (depending on mode it means different movement)
- if player hold himself only on one hand, he's not falling, but can't pull himself up, he gets time to change grip on the other hand

- if player doesn't hold himself on any hand, he falls. acceleration is calculated by mass of rigidbody attached to CameraRig and gravity settings in Unity physics.

2.2 ClimbingTypes

Climber module has three programmed climbing types of movement translation. Every enables different ways of climbing and

Full

However you move your hands, so will move CameraRig - this method allows player to both move vertically and horizontally.

Vertical

Player can only pull himself up, traversing requires moving in real life. As , this method seems to be the most immersing one, the con is that it only lets player climb in room's area.

Horizontal handicap

Player can pull himself in every direction but traversing is multiplied to make magnified or diminished impression.

2.3 Footing

Climbing would be nothing, if players couldn't fall from heights. That's why CharacterFootCollider script calculates player position by keeping collider under the camera (on 0 height in CamerRig local space). This collider imitates players foot and whenever it loses contact with footing, it falls until it does (with the help of Unity physics (gravity)). You can change collider's size to fit your needs, but the con remains, you can not look over the edge without falling.

2.4 Setup

Manual

1. Attach rigidbody to **[CameraRig]** and make sure it has freezed rotation.
2. Attach Collider, and make it desirable size.
3. Add **CharacterFoorCollider** script
4. Add **Climber** script
5. Change **[CameraRig]** layer to „Character”

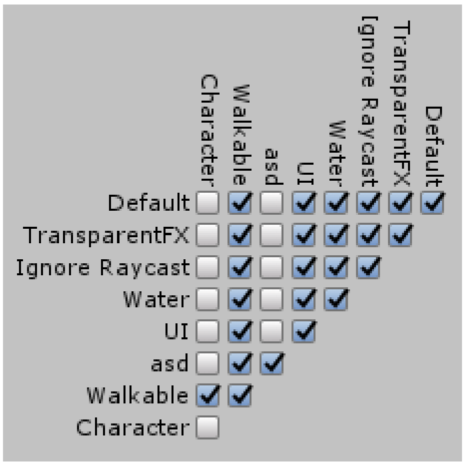
Automatic

1. Make sure you have Grabber already set up.
2. go to window>>GrabberSetup and set up layers, then press Climber set up.

Layers

For Climbing to work properly, two layers were used. „Character” layer which interacts only with „Walkable”, and „Walkable” layer for everything you want to walk on. Using two layers makes character only interact with the ground while enabling every other GameObject to interact with the ground as well without interacting with player.

Layer set up can be done in simìlliar mannet to Grabber Layer setup, and it's matrix is presented on img 2.1.



	Character	Walkable	asd	UI	Water	Ignore Raycast	TransparentFX	Default
Default	□	☑	□	☑	☑	☑	☑	☑
TransparentFX	□	☑	□	☑	☑	☑	☑	
Ignore Raycast	□	☑	□	☑	☑	☑		
Water	□	☑	□	☑	☑			
UI	□	☑	□	☑				
asd	□	☑	☑					
Walkable	☑	☑						
Character	□							

Img. 2.1 Layer Collision Matrix for Climbing.

2.5 Class description

Climber

Fields:

- **enum Controltype controls**{full, vertical, horizontalHandyCap} - set it in inspector to change translation of movement.
- **int Grip** - amount hands taking part in climbing.
- **public transform lHand, rHand** - respectively hands from which movement room position is calculated.
- **public Transform room** - CmaeraRig refference, object that's moved by hands.
- **Rigidbody body** - rigidobody of CmeraRig

methods:

- **void Start()** - searches for needed components.
- **void LateUpdate() -** changes location of **room** if needed.
- **void Climb()** - calculates new position of **room**
- **public void AddGrip()** together with:
- **public void RemoveGrip() -** responsible for character climbing options (**Grip =** 0 - falling, = 1 holding, =2 ability to pull up)

CharacterFootCollider

Fields:

public transform head - reference to player's head.

methods:

void Update() - makes collider attached to [**CameraRig]** follow head by changing it's center.

void Start() – sets up all references.

3. GrabItems

Package contains grab options that are implementations of IGrabItem. To extend grabing just write your own new class inheriting after IGrabItem. Provided grabbing options:

GrabItem – simple grab, using Fixed joint. Grip strength is maximum load that can be applied to joint before breaking it.

GrabItemSnap – on grab, snaps it to the hand keeping it with defined rotation and offset. Quite usefull when making any kind of tool/weapons. (however player grabs, always sticks the same way to the hand).

GrabItemMultipleHands - needs specified amount of grabs to pick object up, usefull for creating heavy-object like feeling, may be used in multiplayer i.e. needs 3-4 hands to be picked up, enforcing cooperation. When object is released component VelocityMaintain is added to it, so it can follow the direction it was thrown.

GrabItemClimbing – solely for climbing purpouse. Grabing them lets player climb.

GrabItemConfigurable – uses ConfigurableJoint. Grabbed object only follows hand's position, leaving it's rotation subject to physical simulation. This type of GrabItem is best for making interaction object that are jointed to other objects (for example, lever sticking out of the ground)

GrabItemFixed – object is not released when trigger is released but when trigger is pressed again.

Coming soon…

As the package is in constant usage in our studio, new grab options will be added whenever we came across some idea. So if you have any idea, or these options are lacking, just contact us. Coming this month will surely be snapping to the target mesh.

4. Editor Windows

4.1 Grabber Setup

Grabber Setup window is used to correctly set up layers (and matrixes) in unity project settings as well as adding components needed for grabber to work properly (i. e. colliders).

4.2 Grab Item

This window helps in configuring lots of objects as grab able in a short time. First select objects you want to have grabable then pick your typ in the window and press button. This will configure and add GrabItem children, copying objects collider (if needed it adds rigidbody and collider to the object).