RemoteBrainSoft

RemoteBrain for Unity

Version 1.4

Whats new?

- RemoteBrain client now lists available servers on your local network
- Unity 5 is now supported
- Material synchronization performance improved
- Bug fixes

Asset Content

Your copy of RemoteBrain for Unity from the Unity Asset Store should contain the following packages:

- **RemoteBrain** - Contains RemoteBrain's client and server along with two demos.
- **RemoteBrainSource** - Contains RemoteBrain's source files.

Installation

Installing and integrating RemoteBrain with your project is done in a few simple steps.
It is recommended that you backup your project before installing RemoteBrain and regularly while working on your project.
First step is to install RemoteBrain from the package RemoteBrain:

1. Open the project you would like to install RemoteBrain on
2. On Unity's editor toolbar, click "Assets", "Import Package" and "Custom Package"
3. In the file selection window, choose "RemoteBrain.unitypackage"
4. In the Import package window, choose "All" and click "Import"

You have now completed the installation and should be ready to integrate RemoteBrain server with your project.

Integrating RemoteBrain server with your project

Now that RemoteBrain is installed, we can integrate RemoteBrain server with your project:

1. In your project, open the scene you would like to control with RemoteBrain
2. On the project window, under "Assets", "RemoteBrain", open the "RemoteBrainServer" folder

3. Add the RemoteBrainBehaviour component to a GameObject in your scene. It is advised to choose a game object that never gets removed from the scene and never gets disabled (for example - the main camera)

At this point, RemoteBrain Server is integrated with your project and should be ready to connect with the RemoteBrain Client.

Connecting with the client

1. Deploy your project onto the device
2. On the toolbar, click window and RemoteBrainClient to open RemoteBrain’s client window
3. In the RemoteBrain Client window, click on the device name / IP address under “RemoteBrain Servers”
4. If the device isn’t listed, type the devices IP address under "Target IP" and press connect. If you don't know the IP address of your device, follow the instructions in the next section "Finding your devices IP address"
5. Once connection is established, the "Connect" button should change to "Disconnect" and the "Status" should change to connected

At this point the Hierarchy window should start updating with the GameObjects from the scene running on the device. Congratulations - you have successfully completed the installation and integration and should be all set!
If for some reason the connection failed, please read the troubleshooting section.

Finding your devices IP address

Similar to the Unity profiler, RemoteBrain Client has to open a network connection with your project while it is running on the device.
There are two ways to find the devices IP:

Finding the IP from the Editor

Deploy your project to the device and open Unity’s profiler under "Windows" and "Profiler". Open the "Active Profiler" drop down and you should see an entry for your device along with its IP address.

Finding the IP from the device

On Android: Go to "Settings", "Wi-Fi", click on your connection and choose "View". Your devices IP address should show at the bottom of the window.
On iOS: Go to "Settings", "Wi-Fi" and click on your connection. Your devices IP address should show next to "IP address".

GameObjects

Once your project is connected to the RemoteBrain Client, Unity’s Hierarchy window is going to be populated will all the GameObjects added to the scene running on the device.
The GameObjects are synchronized in real-time, so any value changes on the device should show in the Editor and vice versa.

Currently, RemoteBrain supports the following GameObject values:

- GameObject.name
- GameObject.active
- GameObject.Transform
- GameObject.renderer
- GameObject.camera

Limitations

Unity makes it hard to find GameObjects with no parent that are inactive. For this reason, you won't see those GameObjects and their children in the Editor until they become active.

Materials

Similar to GameObjects, materials get synchronized in real-time. Any material changes on the client are going to show up on the server and vice versa. You can change any material parameter the same way as you normally would in the Editor.

Limitations

Material synchronization depends on the client having access to the same shaders used by the materials on the server. This limitation shouldn’t be a problem as long as the

client is running in the same project that was deployed with the server.
Textures get synchronized only if they are located directly under a resources folder. Unfortunately, Unity does not provide the full path a texture got loaded from during runtime. As a result, RemoteBrain server can't tell the client which path to load each texture from.

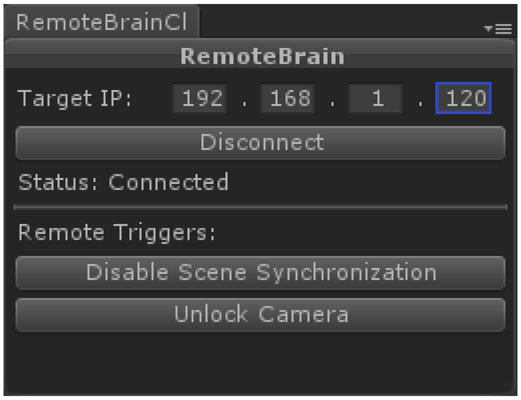
Remote Log

While connected to a device, all Unity log messages are transmitted by default to the RemoteBrain Client. Remote log messages are added to the RemoteBrain Client's console with the device IP address in front of the message.
Note that the stack trace provided with each remote log is the local stack trace and not the devices stack trace. This will be corrected in the future.
Log messages can affect performance and GameObject synchronization speed. Read the Default Triggers section to learn how you can disable remote log transmission.

Remote Triggers

Remote Triggers are functions exposed from your project to the RemoteBrain client. Each exposed trigger will add a button to the RemoteBrain client window on the editor under "Remote Triggers:".

In the following example, two triggers were exposed to the client:



Target IP: 192 . 168 . 1 . 120

Status: Connected

Each time a trigger button is clicked on the RemoteBrain client, the exposed function will be called on the device.

Triggers can be used for toggling debug modes, saving game state when errors show up, spawning GameObjects in your project or to perform any other operation that you normally don't want to expose to the project end user.

Managing Remote Triggers

Remote triggers can be added, updated and removed at runtime through the RemoteBrain.RemoteBrainServer class using the following interfaces:
AddRemoteTrigger(string name, LocalFunc func)
UpdateRemoteTrigger(string oldName, string newName, LocalFunc func)
RemoveTrigger(string name)

Each trigger name has to be unique and will show on the corresponding button at the RemoteBrain client window. Once a trigger is removed or updated with a new name, the original name can be used again. See the "Dealing With Locked Camera" section for an example.

Default Triggers

RemoteBrain currently has two default triggers that always shows up when the client connects to the server.
The first default trigger is called "Scene Synchronization" which toggles the scene synchronization with the RemoteBrain client. When profiling your project, use this trigger to disable synchronization while measuring performance as scene synchronization can be expensive performance wise. Once synchronization is disabled, the trigger will update to enable synchronization.

The second default trigger called "Remote Log" toggles the transmission of Unity log messages from the device to the editor. When remote log is enabled, all Unity log messages from RemoteBrain server will show up in the editors console with the devices IP in front of the message.
Log messages can generate a lot of network traffic

Camera Control

Camera control is a method to attach the projects main camera to the Unity editors camera. This is useful in situations where a free control over the camera is required while the project is running on a device that has no mouse or keyboard inputs.

Controlling A Camera From The Unity Editor

1. In RemoteBrain Client, press the play button but stay in the Scene view.
2. Connect RemoteBrain Client to the device running your project.
3. Drag the script AttachToEditorCamera found under the “Assets/Scripts” folder on the camera you want to attach to the Unity editor Camera.
4. RemoteBrain Client will now update the corresponding camera on your project with the Unity Editor camera transformation.

Dealing With Locked Cameras

Some projects update the cameras transformation every frame. This constant update to the transformation overrides any values sent from the RemoteBrain Client making the camera seem locked.
To get around this issue, a trigger can be used to override the constant transformation update. As an example, check the script CameraLock in the demo project. That script adds a trigger that skips the camera update and lets RemoteBrain Client take control over its transformation.

FAQ

In this section we will cover common questions you may encounter while using RemoteBrain.

"Connection with device failed. Make sure the IP you entered is correct"

Aside from a wrong IP address, this message can show up in the following situations:

1. The device you are trying to connect to is not running a project with an instance of RemoteBrain Server.
2. The device is running multiple instances of your project at the same time or multiple projects using RemoteBrain Server running at the same time. You can

only have one instance of RemoteBrain Server running on each device at any given time.

3. RemoteBrain Server encountered a network problem and has to be reset. Try to force close your project and start it over.

Does RemoteBrain require Unity Pro?

Yes. RemoteBrain uses sockets to transfer data between the client and the server which are only supported in Unity Pro. Unfortunately, this is not likely to change in the future.

Which port does RemoteBrain use?

RemoteBrain uses port 61234. Make sure this port is not blocked by your firewall.

GameObjects missing in RemoteBrain Client

If your projects hierarchy contains disabled GameObjects, there is a chance that RemoteBrain won't be able to access those GameObjects or their children. As soon as those GameObjects become active RemoteBrain should add them and their children to the hierarchy.

"error CS0433: The imported type '...' is defined multiple times" when importing RemoteBrainSource

This error means you imported both source and DLL's into the same project. If you choose to use the source, please delete both RemoteBrain Client and Server DLL's.

Further Support Options

For the most updated documentation, please visit: http://support.remotebrainsoft.com
If you have any additional questions, please email our support team at:
support@remotebrainsoft.com

Disclaimer

RemoteBrain was thoroughly tested with various projects to insure it is bug free.
It is always recommended you keep backups of your project in case an unexpected problem occurs.
RemoteBrainSoft won't be liable to any issues occurring while using RemoteBrain for Unity.