GitHub Access

Many of my products now allow for direct GitHub access to the repositories. This has many advantages over using the asset store's distribution system.

Asset Store:

- The package manager is severely broken in many versions of Unity, giving you old, cached versions when you upgrade, but telling you it has the latest.
- The UAS has an approval process for new assets and upgrades that can take anywhere from a day to three months

Git Repositories:

- All updates are available the moment I check them in, no approval process needed
- Old versions available
- See a history of changes and diff them
- A bit more setup

How to get access

Currently the way to get access is through the support channels discord bot. Eventually this will be possible via my website, jdbtechservices.com as well. You can access the discord server at:

https://discord.gg/Tq9qDZFV

To get access to the server, and to get github access, message "GitHubBot" with:

!verify <invoice>

Where <invoice> is the Unity invoice number, order number, or invoice number from my store on jdbtechservices.com. Note, do not include the <> and make sure their is a space between !verify and the invoice number.

The bot will verify any products you have in those invoices, and make the various channels of the discord server available for you.

To get github access, send this to the bot:

!redeem

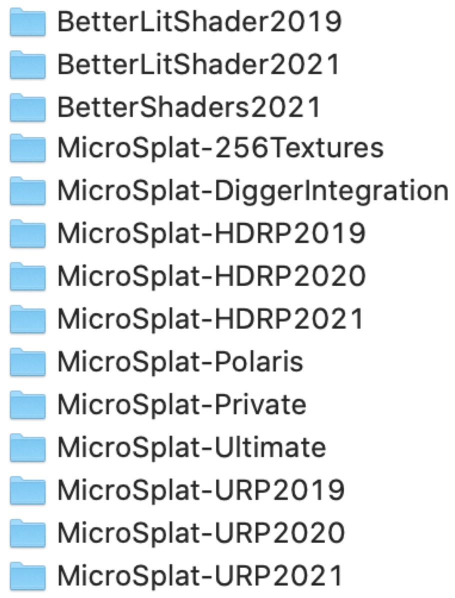
If any of your products have available github depots available, it will ask for your github account name and grant that user access. If you use an invoice from jdbtechservices.com, it will also ask you for the email used on that account. An email will be sent to the email used at github or jdbtechservices.com to invite you to the repository, you will have 7 days to respond before the invite becomes stale.

Note that only one user can be tied to an invoice.

Sharing across projects

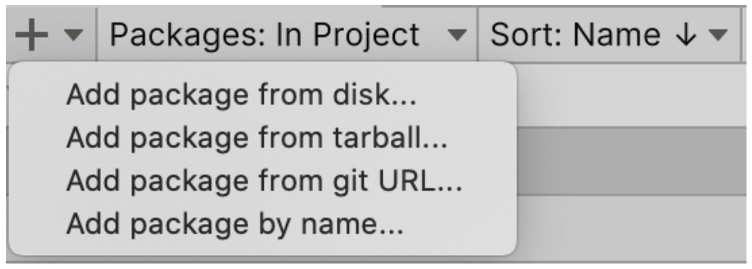
You can sync any of my products directly into your asset folder, however, all of my assets are delivered as packages, which allows you to reference them instead of including them in your projects directly. This is a much more convenient workflow, and allows you to update them in multiple projects at once. For developing my products it's especially useful, since I can develop in multiple unity versions while still editing shared code.

For instance, I have all of my products setup in a single folder like so:



BetterLitShader2019
BetterLitShader2021
BetterShaders2021
MicroSplat-256Textures
MicroSplat-DiggerIntegration
MicroSplat-HDRP2019
MicroSplat-HDRP2020
MicroSplat-HDRP2021
MicroSplat-Polaris
MicroSplat-Private
MicroSplat-Ultimate
MicroSplat-URP2019
MicroSplat-URP2020
MicroSplat-URP2021

Then any projects I have simply reference the package in each of these folders to include them into the project. To do this, open the Unity package manager and select the "add package from disk" option:



Then navigate to the folder and select the package file. Note that MicroSplat-Ultimate can be brought in as a single package, or each folder can be included as a package if you don’t want to import all of it.

Also of note is that Unity will store this as an absolute path, however, you can open the Packages/manifest.json file in your project and change this to a relative path if you are working with a team.