GT New Horizons

Development

Introduction

Everyone can contribute to GTNH in one way or another. One possible way is to improve or expand the source code of the mods.

To get started, you first need a vision for what you want to do.
If you don't have it yet, you can look through issues (https://github.com/GTNewHorizons/GT-New-Horizons-Modpack/issues) and try to address some.
If you *do* have a vision but want to hear more opinions on it, you can discuss it in a new Github issue or in Discord.

Getting started

If you found a mod that you want to edit, then create a fork of it by pressing the Fork button on its GitHub page. If you are unsure what repo to fork for a particular issue, find the owner of Item/block, then search for this mod's name in this spreadsheet (https://docs.google.com/spreadsheets/d/1LHd8c4FLLHiJqsuTppF2XFTE6bh-mXE7afVwGbzOKms/edit?usp=sharing). If you cannot find a github link there, it's possible we uses the upstream mod without gtnh only modifications. In any case, you can always ask on discord #mod-dev channel.
After that, you can clone your fork to a local repository, import it to your IDE, and get started on development.
If you are not sure how to do something, try to look up how things similar to it were done before.

Starting from an existing repository

1. Clone the repository.
2. Update the buildscript automatically through `./gradlew updateBuildScript` or if it doesn't work, manually with this guide (https://github.com/GTNewHorizons/ExampleMod1.7.10/blob/master/docs/migration.md).

Starting from scratch

See the ExampleMod README.md (https://github.com/GTNewHorizons/ExampleMod1.7.10#getting-started) if the instructions below don't work.

1. Unzip project starter (https://github.com/GTNewHorizons/ExampleMod1.7.10/releases/download/master-packages/starter.zip) into project directory
2. Replace placeholders in LICENSE-template and rename it to LICENSE, or remove LICENSE-template and put any other license you like on your code. This is an permissive OSS project and we encourage you participate in OSS movement by having permissive license like one in

3. Ensure your project is under VCS. For example initialize git repository by running `git init; git commit --message "initialized repository"`
4. Replace placeholders (edit values in gradle.properties, change example package and class names, etc.)
5. Run `./gradlew build`
6. Make sure to check out the rest sections of this file
7. You are good to go!

Next steps

1. Run `./gradlew updateBuildScript` to update the Gradle build-script for the repository.
2. Run `./gradlew build --build-cache` to build the project. The flag `--build-cache` makes builds faster. For more info, see documentation (https://docs.gradle.org/current/userguide/build_cache.html).
3. If the build fails, check what was printed in console. Usually it fails because of outdated dependencies or Spotless. You can fix it with `./gradlew updateDependencies` or `./gradlew spotlessApply` respectively.
4. In Eclipse / IntelliJ IDEA, import your workspace as an existing Gradle project. If the import fails, you may need to adjust the Gradle version and/or the buildscript. If you are developing for Minecraft 1.7.10, read this guide (https://github.com/GTNewHorizons/ExampleMod1.7.10/blob/master/docs/migration.md).
5. After you made changes, test them:
    1. Build the project with `./gradlew build`. You will find several jars in `build/libs`. What you need is the one without affixes like `-dev` or `-api`.
    2. Download the last nightly build and import it in the launcher of your choice.
    3. Replace the target mod with your version or add if it's a new one.
    4. Launch the pack and check that things work as they should.

Common issues

- If your IDE tells you that stuff is missing even after you executed `gradlew build`, then try to reload Gradle.
- You need a 64-bit Java for development. Most mods work with any Java 8-19. However, some mods now use `enableModernJavaSyntax`in gradle.properties. In that case you need at least Java 11 to compile them. If you have multiple Java versions on your machine, then you may need to set the `JAVA_HOME` environment variable to the location of your desired Java installation and also configure it in your IDE.

Git Configuration

If you want to avoid signing commits with your real name, then please refer to the page Managing Git Identities.

Contribution Procedure

3. Test your changes thoroughly. If you change something, understand the changed class or method in order to understand what to test.
4. Push code into your fork.
5. Open a pull request against **master** branch and provide a description on what it changes/adds/removes. Include pictures/graphs if necessary. If there is a related issue, link it.
6. Address review problems. Resolve merge conflicts. Wait for the final merge. During this period, your PR might be merged into dev branch manually by project admins, on a case by case basis. If that happens, your changes will go into one of the nightlies for border testings. There is no strict rules over whether this merge-into-dev will happen at all, and it's usually not needed anyway. It's however expected that the original author should fix every issues arouse during nightly testings, if any.

Please do not make multiple changes/features in a single PR, make a separate PR for each. If they depend on each other, you can leave a note in the PR's description.

If you are not sure how to fork and open pull requests, read this gist (https://gist.github.com/Chaser324/ce0505fbed06b947d962) by Chaser324.

GTNH-specific Contribution Guides

- You can find the Code of Conduct of the GTNH dev community here (https://github.com/GTNewHorizons/GTNH-Dev-Doc/blob/master/developer's%20code%20of%20conduct.md).
- See which GTNH mods are supported outside of the modpack and see an overview of licenses in this spreadsheet (https://docs.google.com/spreadsheets/d/1LHd8c4FLLHiJqsuTppF2XFTE6bh-mXE7afVwGbzOKms/edit#gid=1469028684).
- For the GTNH Code Style guidelines please see the dedicated page Code Style.
- For quest development, read the Quests Development Guide (https://github.com/GTNewHorizons/GT-New-Horizons-Modpack/tree/master/config/betterquesting).
- The StructureLib documentation can be found here (https://www.gtnewhorizons.com/StructureLib/?version=master). Use this for your multiblock structures.
- To understand how chemical formulas are balanced in recipes, read Chemical Balance.

Tools

All applications that are grouped together can do about the same. This does not mean that you can pick any one of them, as they differ in functionality, style, performance and customizability. It is recommended you look at what they can do and try each one out, so you can decide which application you want to use for which use case.

Category	Links [Expand]

Helpful IDEA Plugins

Additionally, if you choose JetBrains IntelliJ IDEA as your IDE, there are some plugins that can be very helpful in some ways

- Minecraft Development: Autocompletion for mixin development, builtin NBT editor, minecraft project project type in new project window, and a bunch more small utility. Notice that for GTNH specifically you will want the fork maintained by eigenraven (https://github.com/eigenraven/MinecraftDev). We use RFG as the build tool and that one unfortunately is not compatible with the original version available on the plugin marketplace. Follow the install instructions in README of that fork.
- ASM Bytecode Viewer: Very helpful plugin for asm development. Can also be occasionally helpful for mixin development. Notice that there are multiple plugins with similiar name and near identical purpose, with one of them being a bundled one from JetBrains. The JetBrains one is slightly wonky and doesn't display ASMified code. The plugin suggested here works perfectly and can display ASMified code. If in doubt whether you got the correct plugin, here is a direct link: https://plugins.jetbrains.com/plugin/10302-asm-bytecode-viewer To use this plugin, focus on any opened Java editor tab (either it's decompiled library class without attached source, library class with attached source, or your own source files), then click ASM Bytecode Viewer in Code menu. The action is also available via Search Everything under the same name. For your own source files, it will trigger a project build to ensure .class files are up to date.

Debugging with Eclipse

Set up launch config:

1. Select the project you want to debug
2. Click the drop-down menu next to the debug symbol -> Debug configurations...
3. Select "Remote Java Application"
4. Click "New launch configuration" (top left)
5. Change the port to 5005
6. Click "Apply"

Actually debug:

1. Launch the game with `--debug-jvm` (I do this from the terminal)
2. Wait until it says "Listening for transport dt_socket at address: 5005"
3. Debug with the launch config you created (you may have to go into the debug configurations menu again, if it doesnt show up in the drop-down yet)
4. Change to the "Debug" view (button in the top-right corner). There may be a window popping up asking if you want to go there
5. Done!

Maven/Jenkins Selection

URL	Mods [Expand]

Resources

Not every wiki/tutorial/article/... listed here contains information about Minecraft version 1.7.10 but some topics haven't changed much and often the general methodology is the same. The newer the

Use Ctrl+F if you look for something specific.

You can look at the source-code of most mods included in GTNH here (https://github.com/GTNewHorizons).

Link Collections

Name	[Collapse] Version(s)
MMD Resource Index (https://github.com/MinecraftModDevelopment/Modding-Resources/blob/master/index.md)	Various
Minecraft Modding Overview (WIP) Resources for Mr. Turner (Archive) (https://archive.ph/YGQKG)	1.6, 1.7
Awesome Minecraft 1.7.10 (https://github.com/LegacyModdingMC/awesome-minecraft-1.7.10)	1.7.10

Tutorials / Blogs

Name	Version(s)	Topics (including, but often not limited to) [Expand]

Wikis / Docs

Name	Topics [Expand]

Raven's list of rendering resources

OpenGL resources:

- [1] (https://learnopengl.com/) - a good C++ tutorial for opengl
- [2] (https://lazyfoo.net/tutorials/OpenGL/index.php) - another step-by-step opengl c++ tutorial
- [3] (https://ahbejarano.gitbook.io/lwjglgamedev/) - lwjgl3 tutorial book about 3d gamedev with code samples
- [4] (https://open.gl/) - a short, but useful tutorial into modern opengl's rendering pipeline
- [5] (https://www.songho.ca/opengl/) - a lot of useful diagrams and linear algebra for 3d rendering with opengl
- [6] (https://github.com/LWJGL/lwjgl3-demos/) - a lot of example LWJGL3 programs showing off implementations of various rendering techniques, it's a small gold mine

Reference docs:

- [8] (https://www.khronos.org/files/opengl45-quick-reference-card.pdf) - a cheat sheet you can print out with the full opengl API

More advanced stuff:

- [9] (https://raytracing.github.io/) - Raytracing in a Weekend series, teaches you a lot about computer graphics (beware, raytracing != rasterization which is what opengl uses, but most of the math still applies)
- [10] (https://realtimecollisiondetection.net/blog/?p=86) - a blogpost on 1 efficient method of batching drawcalls for performance
- [11] (https://asawicki.info/news?x=titles) a great blog from an AMD gpu driver developer, lots of valuable info here

Search for GDC, AMD, NVIDIA and AMD talks on youtube about rendering topics too Super advanced stuff:

- [12] (https://fgiesen.wordpress.com/2011/07/09/a-trip-through-the-graphics-pipeline-2011-index/) - a series of blogposts going really deep into how GPUs work
- [13] (https://www.amd.com/content/dam/amd/en/documents/radeon-tech-docs/instruction-set-architectures/gcn3-instruction-set-architecture.pdf) - the full instruction set architecture doc for AMD GCN3 GPUs, useful to understand what kind of machine code GPUs actually execute

Random non-rendering stuff: https://gameprogrammingpatterns.com/contents.html - a book (available for free online) about common patterns in game code

Help

- Mathematics on StackExchange (https://math.stackexchange.com/): Forum for everything related to Mathematics
- StackOverflow (https://stackoverflow.com/): Forum for everything related to Programming
- #mod-dev (GTNH Discord (https://discord.gg/EXshrPV))
- #mc-dev-other (Minecraft Mod Development Discord (https://discord.mcmoddev.com/))
- #moddev (Minecraft Pack Development Discord (https://discord.gg/R7FThXBUMR))
- #mc-dev (CurseForge Discord (https://discord.com/invite/curseforge))
- #help-modded-java (Blockbench Discord (http://discord.blockbench.net/))

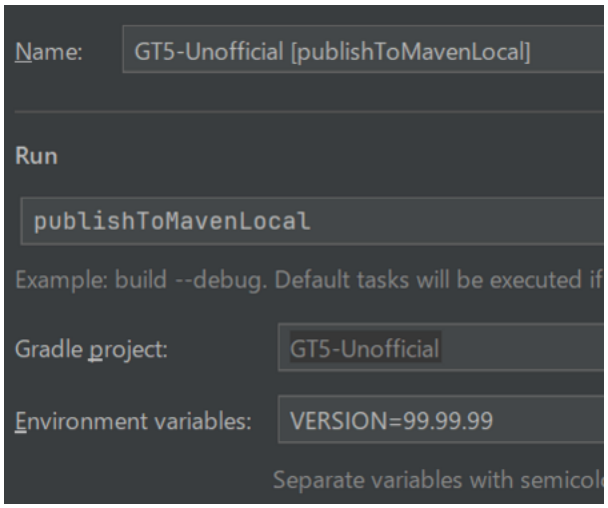
Maven Local Repository

Sometimes you want to make cross-mod changes. In order to make use of code changed locally from another mod, use maven local.

- Run `publishToMavenLocal` for your dependency mod. Setting environment variable `VERSION` allows you to publish it with specified tag, like 99.99.99
- Add `mavenLocal()` to your `repositories.gradle` in your dependent mod, and change `dependencies.gradle` to your desired version.

Note that this doesn't work on GitHub, so in order to pass CI, you need to publish pre tag online. If you're not in the GTNH team, ask others to publish it.

Hotswap allows you to reflect code changes without relaunching the game. Use `1b. Run Client (Java 17, Hotswap)` or `2b. Run Server (Java 17, Hotswap)` to launch the game, and use `Run > Debugging Actions > Reload Changed Classes` to hotswap.



Environment variable can be set on IDEA GUI

Dependency types

The following table shows if a source dependency - later called the Source - you included in Your Mod is visible to Your Mod and the mods that depend on Your Mod. For instance, `implementation` makes the Source visible to Your Mod at compile and runtime, but only at runtime to the Dependent mods, meaning that the Dependent mods are not able to use a class from Your Mod if it implements an interface from the Source. There are also *test* versions of some, you probably just want *testImplementation*.

	Your mod		Dependent mod	
	compile	**runtime**	**compile**	**runtime**
api	Yes	Yes	Yes	Yes
implementation	Yes	Yes	No	Yes
compileOnly	Yes	No	No	No
compileOnlyApi	Yes	No	Yes	No
runtimeOnlyNonPublishable	No	Yes	No	No
devOnlyNonPublishable	Yes	Yes	No	No
runtimeOnly	No	Yes	No	Yes
compile	Deprecated. Use *api* or *implementation*			

You can also read document on ExampleMod repository (https://github.com/GTNewHorizons/ExampleMod1.7.10/blob/master/dependencies.gradle) for details.

JVM flags

These are some useful JVM flags for development purposes. A full, detailed list can be found in the table below.

- `-Dmixin.debug=true` Enable Mixin Debug
- `-Dmixin.debug.verbose=true` Verbose Debug Messages
- `-Dmixin.debug.export=true` Export classes after mixins have been applied. If you include fernflower in your classpath, or copy org.jetbrains.fernflower into your mixin jar the exported classes will be decompiled

- `-Dlegacy.debugClassLoading=true` Class Load Debugging
- `-Dlegacy.debugClassLoading=true -Dlegacy.debugClassLoadingFiner=true -Dlegacy.debugClassLoadingSave=true` Dump classes to `./RFB_CLASS_DUMP`
- `-Dfml.debugAccessTransformer=true` Enable debug logging of access transformers

Flag	Default	Result	Code Reference [Expand]

Remote debugging the whole pack

- `-agentlib:jdwp=transport=dt_socket,server=y,suspend=y,address=5005` to wait for a debugger to be attached to the game before running code, put this in launcher args to debug the full pack
- In IntelliJ, use Run->Attach to process to attach the debugger to the running instance.

v · t · e (https://wiki.gtnewhorizons.com/wiki/Template:Navbar_GTNH?action=edit) [Collapse]

GT New Horizons

Meta	Acronyms • Discord Commands • Modlist • Official Servers • Server List • Quick Links
Tiers	Beginner Tips • Stone • Steam • LV • MV • HV • EV • IV • LuV • ZPM • UV • UHV • UEV • UIV • UMV • UXV • MAX
Mechanics	Bees • Biome • Electricity • Electricity for Beginners • Guides • Hazard • IC2 Crops • JourneyMap • Loot • Multiblock • NEI • Oil • Ore Generation • Pollution • Power Generation • Quest Book • Tinkers Tools • Waila
Setup	Additional Mods • Backups and Recovery • Commands and Configurations • Debugging • Release Stability • Installing and Migrating • Lag • Low End PCs • Server Setup • Resource Packs • Shaders • World Generation • World Seeds
Development	Pollution Revamp • Development • Ore Spreadsheet Export • Salt Nerf Options • Technical Oregen Information • Upcoming Features