Minecraft Modding Workshop

Version 1.15

Some general notes before we start:

* You do not need Minecraft or a Minecraft account to mod at all
* We will be coding in Java, experience in Java is not necessary but if you have Java experience, you will understand the code and why it is set up how it is\
* \*\*Download the .zip file from the repository I have set up, we will download software to unzip it later\*\*
* <https://github.com/HackBI/HackBI4-MinecraftModWorkshop-HS>

**Set-up Steps:**

1. Download and Install the following:

Java JDK - <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

* This allows your computer to understand and run Java code, which Minecraft is built with.
* Some/most may already have this.
* May have to make an account
* Install it

Java JRE - <https://www.oracle.com/java/technologies/javase-jre8-downloads.html>

* Just like the last one, some of you may already have it
* Install it

Eclipse - <https://www.eclipse.org/downloads/>

* This is the application we will be coding in
* Install it
* Once Eclipse opens, select Eclipse IDE for Java Developers

WinRAR - <https://www.win-rar.com/start.html?&L=0>

* If you already have an application to open/edit compressed files, you do not need this
* This allows us to use .zip files
* Install it

Minecraft Forge - <https://files.minecraftforge.net/maven/net/minecraftforge/forge/index_1.15.2.html>

* Click the MDK of the recommended, NOT the latest
* \*IMPORTANT\* After you click the MDK, an ad will appear, DO NOT CLICK ANYTHING, wait 5 seconds then click the red “SKIP” box in the top right corner
* The .zip file will automatically download
* Nothing to install here
* I would move it to your desktop

1. Right click the Forge MDK, click “extract to Forge MDK”

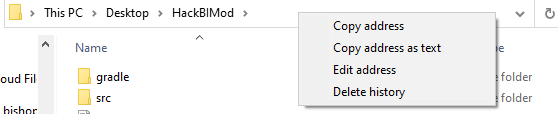
* This will make a new file on the desktop
* Delete the .zip file

1. Rename the folder to the name of your mod (ex. HackBIMod)
2. Set up environment variables

* Go to the desktop and look up “environment variables”
* Click “Edit System Environment Variables”
* Click “Environment Variables”
* In the top box click “New”
* Variable Name is: JAVA\_HOME (make sure it is all caps)
* Variable Value is: Click “Browse Directory” then select: This PC -> Windows (C: ) -> Program Files -> Java -> jdk1.8.0
* For the bottom area, do the exact same thing, same name, same value
* After both of those are made, scroll down in the bottom area and double click on “Path” variable
* Click “New”, click Browse, then select: This PC -> Windows (C: ) -> Program Files -> Java -> jdk1.8.0 -> bin
* Click “Okay” on these smaller windows to save and get rid of them

1. Open the command prompt

* Open the desktop search bar and type “cmd” then click enter, a black box will appear, this is the command prompt
* Type the command: cd \*the file path to your mod folder\*
  + Get the file path by opening the file in file explorer and copying the bar at the top



* Now that we are in the mod folder, type the command: gradlew genEclipseRuns
  + This can take a few minutes so be patient and just wait
  + May need to click “Allow Access” on a pop up
  + At the end it should say “BUILD SUCCESSFUL”
* Run the command: gradlew eclipse
* Once successful, close the command prompt



1. Open Eclipse

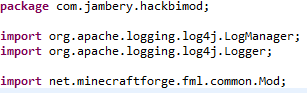
* Right click in the left-area “Package Explorer” and click import project
* Click “Existing Gradle Project”
* Click “Browse” and find your Mod folder
* Click Next and Next again, then Finish
* After that is done you should see your mod folder on the left!

1. Set up classes

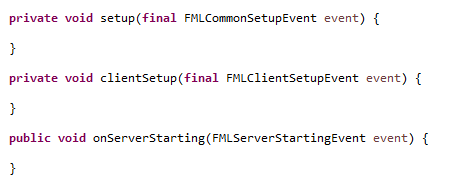
* Delete the com.example.examplemod package
* Right click src/main/java and click new package
* Name it: com.\*your name/handle\*.\*your mod name\*
  + Mine will look like: com.jambery.hackbimod
* Right click New class in that package called “Main”, click finish
* @Mod(Main.MOD\_ID) tag above public class
* Set up these variables:



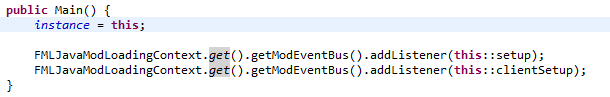
* These should be the imports, make sure to get the apache ones:



* Set up these methods:



* Set up the constructor:



\*\*NOTE\*\* Import everything as you go, there will be a lot of import statements

1. Go to src/main/resources package

* Inside of META.INF click on the mods.toml file
* Delete everything in this file
* PASTE IN the txt file in the repository called tomlFile.txt
* Change modID to the exact same name you put in the name class, in quotes
* Change version to “1.0”
* Change display name to the name you want to appear in-game, like “Jack’s Awesome Mod”
* You can change credits to whatever you like “I would like to thank the academy and my parents”
* Change author to your name/handle
* Change description to whatever you like, make sure this, like everything else is in quotes””
* Carefully change the dependencies
  + Where it says dependencies.examplemod (twice), change it to your Mod ID which is above a few lines

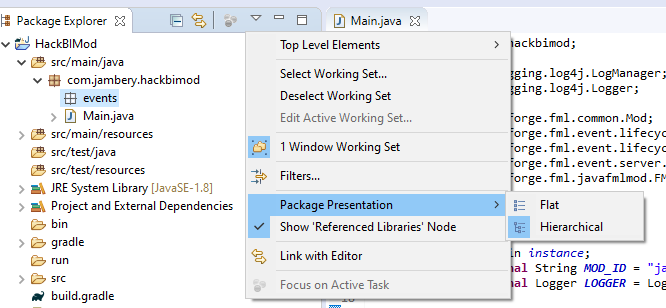
1. Now, we should be able to run the game to test it out

* Right click runClient.launch and click Run As -> runClient
* If you go into Mods you will be able to see everything you just entered in the toml file

------- Setup is finally done! Let’s add some items! ------

**Adding Items**

1. Switch to Hierarchical view in the three dots at the top of the package explorer:



1. New package in com.jambery.hackbimod called com.jambery.hackbimod.events

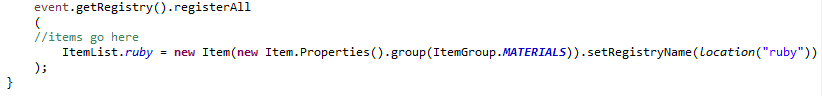
* New class called RegistryEvents
* Go to the repository files and paste into this class, under the imports that are in there, the registerEvents.txt file in to replace the class that is in there now

1. New package in com.jambery.hackbimod called com.jambery.hackbimod.lists

* New class called ItemList
* In this class we will list all the items of our mod just inside the class like this:

 I am going to add rubies to the game as an upgrade from diamond.

* You can make any item you like, if it has more than one word, write\_it\_like\_this
* Now save and close this tab and go back to the RegistryEvents Class
  + In the registerItems method, add this line under the registerAll area where the comment is:



* We can set more properties here but this is all we will do for now
* All items you ever make will be listed here and listed in the ItemList class
* If you want to add another Item here, put a comma at the end of that line and do the exact same thing on the next line
* The item now exists in the game but has no textures

1. Adding Resource Packages

* New package in src/main/resources called assets.your MOD ID.models.item
* New package INSIDE of that one called assests.yourMODID.lang
* It will split them up into like branches
* New package inside of the assets.yourMOD ID called assests.your MOD ID.textures.items

1. .json Files

* New Untiltled Txt File in models.item save as your item name exactly as you had it, PLUS .json
* Mine would be ruby.json
* Save it all the way down the resources rabbit holes until you get to models.item
* Go to the repository folder and find the models.item folder and use my rubyJSON.txt file as a guide for yours
* All you have to do is paste it in, change the mod ID, and item name

1. Item Textures

* So now we need to find picture for our items, you could go through the trouble of making one online or in photoshop but I steal mine from google which you can do to
* I have put my ruby image in the images folder in the repository if you want it
* Any image that is 16x16 pixels will work
* Now when you have your image saved somewhere, drag it into textures.items in eclipse
* You can also copy and paste it
* \*\*MAKE SURE the .json file and the file you just added have the same name like ruby.json and ruby.png

1. Lang file

* New untitled text file in the lang package
* Save it as “en\_us.json”
* Find the en\_us.txt file in the repository and paste it in

------Your item is now in the game! Load up a new world to check it out!------

**Adding Blocks**

1. Open RegistryEvents class

* Copy and paste registerItems method and paste it below the actual registerItems method
* Change Items to Blocks (registerBlocks) and Item to Block (Register<Block>)
* SCREENSHOT METHOD HEADER
* Keep the ItemList line of code in there for now

1. Create new class inside of lists named BlockList

* Inside create your block with whatever name you like:
  + public static Block ruby\_block;
* Save and close this class

1. Go back into the RegistryEvents class inside of your registerBlocks method:

* Replace the ItemList line with:
  + SCREENSHOT OF LINE
  + BlockList.ruby\_block = new Block(Block.Properties.create(Material.ROCK).harvestLevel(1-3).harvestTool(ToolType.PICKAXE).sound(SoundType.STONE)).setRegistry(location(“ruby block”))
* You can set any properties you want by just adding a period and looking at the options
* Sound is the noise it will make when you break it
* harvestTool is the tool best suited for breaking it
* Make sure you look up what the parameters like the harvestLevel for example which indicates what tool is required to break the block:
  + 0 – Wood
  + 1 – Stone
  + 2 – Iron
  + 3 – Diamond
* Again if you want to add another block, put a comma on the outside of the last parenthesis and do the same thing on the next line, just like items.

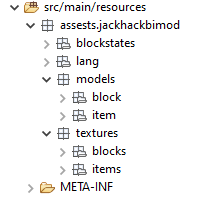
1. Go to ItemList Class

* In Minecraft, blocks are considered a type of item so we need to add all of our blocks to the ItemList class as well
* Add the line:
  + public static Item ruby\_block;
  + SCREENSHOT

1. Go to RegistryEvents Class in the registerItems method:

* Add a comma after the last parenthesis of our first item and add on the next line:
  + ItemList.tutorial\_block = new BlockItem(BlockList.ruby\_block, new Item.Properties().group(ItemGroup.MISC)).setRegistryName(BlockList.ruby\_block.getRegistryName())
* The block is now fully added and it is time for the textures, very similar process to the item textures

1. New package inside of resources -> assets called blockstates
2. New package inside of resources -> assets -> models called block
3. New package inside of resources -> assets -> textures called blocks

These should look like this:

1. Go into our lang -> en\_us.json file

* Add a comma to the end of the first line
* On the second line add a very similar line:
  + “block.yourTutorialName.ruby\_block”: “Ruby Block”
* Save and close this file

1. Find the ruby\_blockBLOCKSTATE.txt file in the repository

* Add it to your blockstates package BUT rename it to ruby\_block.json (the JSON tag is the important change here, also change the name to whatever block you are doing)
* \*\*DO THE JSON IN VIDEO\*\*

1. Find the ruby\_blockMODEL.txt file in the repository

* Add it to your block package IN THE model package BUT rename it to ruby\_block.json (the JSON tag is the important change here, also change the name to whatever block you are doing)
* \*\*DO THE JSON IN VIDEO\*\*

1. Find the ruby\_blockITEM.txt file in the repository

* Add it to your item package IN THE models package BUT rename it to ruby\_block.json (the JSON tag is the important change here, also change the name to whatever block you are doing)
* \*\*DO THE JSON IN VIDEO\*\*

1. Find a texture for your block

* A block texture is usually 16x16 but can go up to 32x, 64x, and even 512x with better hardware
* You can find your own or use mine in the repo
* I will also mention that you have access to all of the regular Minecraft resources in:
  + Project and External Dependencies -> client-extra.jar -> assets -> Minecraft -> textures -> block
* Put your texture into the block package inside of the textures package