

Minecraft “Modding” Installation Procedures – Week 3





Finally...it is time to create our new block in Minecraft!....

Creating our new “kcd_Block”, we’re going to use 2 methods to create it....

- (1) using Eclipse, MCP and just Java
- (2) using Eclipse, MCP, ModLoader and Java

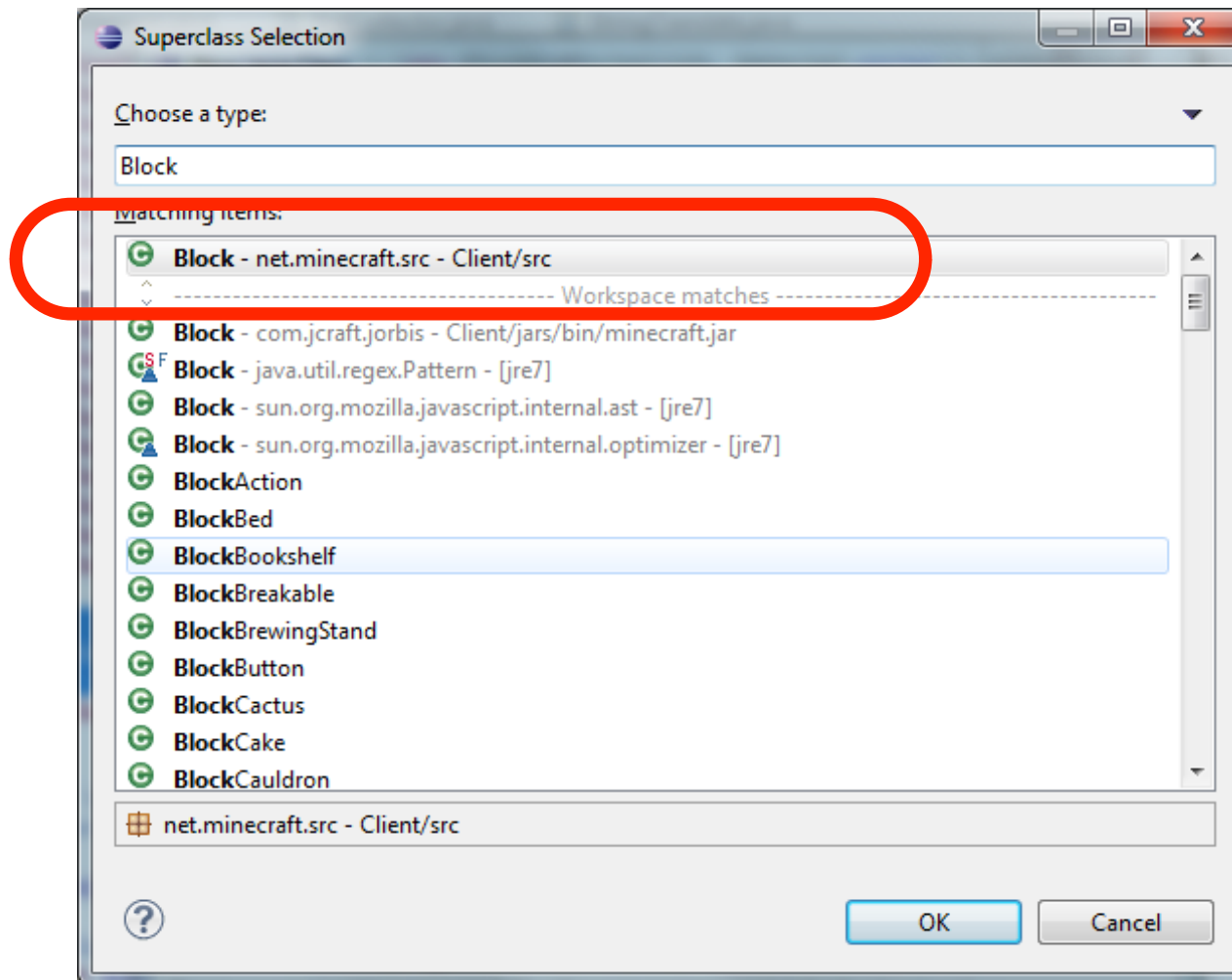


-
- The screenshot shows an IDE interface. On the left, the 'Package Explorer' displays a project structure. The 'net.minecraft' package is selected, and a context menu is open over it. The menu options are: 'New' (expanded), 'Go Into', 'Open in New Window', and 'Open Type Hierarchy'. The 'New' sub-menu is open, showing options: 'Java Project', 'Project...', 'Package', and 'Class'. On the right, the 'Block.java' file is open in the editor, showing several lines of code, all starting with 'public static final'.



Create “new block” class, BlockKCD

- Select parent: Block



*Common mistake is to
Select the wrong Block
class*

*(be careful, there are a few
of them!)*



Create new class BlockKCD

- Set: Automatically generate constructors

New Java Class

Java Class

Type already exists.

Source folder: Client/src Browse...

Package: net.minecraft.src Browse...

☐ Enclosing type Browse...

Name: BlockKCD

Modifier: ☒ public ☐ default ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: net.minecraft.src.Block Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ Constructors from superclass (s)

☒ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments: (Configure templates and default value [here](#))

☐ Generate comments

Cancel Finish



Your new class, BlockKCD...

- Create a new Java class:

```
package net.minecraft.src;  
  
import java.util.Random;  
  
public class BlockKCD extends Block {  
  
    public BlockKCD (int par1, Material par2Material) {  
        super(par1, par2Material);  
    }  
  
    public BlockKCD (int par1, int par2, Material par3Material) {  
        super(par1, par2, par3Material);  
    }  
}
```



New block graphics....

- Placing the “terrain.png” file....

Windows: C:/<YourMCPDirectoryLocation>/mcp726a/bin/minecraft

Mac: MacHD/Users/jim_blair/Desktop/KCD726a/mcp726a/bin/minecraft/terrain.png



See it!!!!!

(it is the
“236”
defined
block)



Create instance of new class...

- Add the new block in Block.java, line 193

```
public static final Block stairsWoodJungle = (new BlockStairs(136, planks,  
3)).setBlockName("stairsWoodJungle").setRequiresSelfNotify();
```

```
/**  
 * Coder Dojo Block "kcdBlock"  
 */  
public static final Block kcdBlock = (new BlockKCD(146, 236,  
Material.glass)).setBlockName(" kcdBlock");
```

The new instance, kcdBlock is given the
Block ID = 146, and an icon number of 236

When using ModLoader, we won't need to
assign the icon ID manually



Rebuild Minecraft, with new Block

- Recompile and run...(Windows)

```
C:\MinecraftMod\mcp72> recompile.bat
== MCP 7.2 (data: 7.2, client: 1.3.2, server: 1.3.2) ==
# found ff, ff patches, srgs, name csvs, doc csvs, param csvs, renumber csv, astyle, astyle config
== Recompiling client ==
> Cleaning bin
> Recompiling
- Done in 6.54 seconds
```

```
C:\MinecraftMod\mcp72> startclient.bat
== MCP 7.2 (data: 7.2, client: 1.3.2, server: 1.3.2) ==
[23:26:35] 27 achievements
[23:26:35] 195 recipes
[23:26:35] Setting user: Player157, -
```

*Record Player number
(in this case, "Player157")*



Rebuild Minecraft, with new Block

- Recompile and run...(Mac) *(1) Compile in new block...*

```
jlm_dlat@jlm-dlat-s-MacBook-Pro ~/Desktop/KCD726a/mcp726a
-$ sh recompile.sh
MCP 7.26 (data: 7.26, client: 1.4.7, server: 1.4.7) ==
# found ff, ff patches, srgs, name csvs, doc csvs, param csvs, renumber csv, astyle, astyle config
== Recompiling client ==
> Cleaning bin
> Recompiling
- Done in 5.70 seconds
== Recompiling server ==
> Cleaning bin
> Recompiling
- Done in 4.62 seconds
jlm_dlat@jlm-dlat-s-MacBook-Pro ~/Desktop/KCD726a/mcp726a
-$ sh startclient.sh
MCP 7.26 (data: 7.26, client: 1.4.7, server: 1.4.7) ==
[20:32:34] 27 achievements
[20:32:34] 210 recipes
[20:32:35] Setting user: Player157, -
[20:32:35] Client asked for parameter: server
[20:32:35] LWJGL Version: 2.4.2
[20:32:36] ModLoader 1.4.7 Initializing...
[20:32:36] Done.
[20:32:36] Starting up SoundSystem...
[20:32:36] Initializing LWJGL OpenAL
[20:32:36] (The LWJGL binding of OpenAL. For more information, see http://www.lwjgl.org)
[20:32:36] OpenAL initialized.
[20:32:42] Jan 25, 2013 8:32:42 PM net.minecraft.src.IntegratedServer startServer
[20:32:42] INFO: Starting integrated minecraft server version 1.4.7
[20:32:42] Jan 25, 2013 8:32:42 PM net.minecraft.src.IntegratedServer startServer
[20:32:42] INFO: Generating keypair
[20:32:42] Jan 25, 2013 8:32:42 PM net.minecraft.server.MinecraftServer initialWorldChunkLoad
```

(2) Run updated Minecraft game

*(3) Record Player number
(in this case, "Player157")*



Allocated new blocks to player...

- In the game... `/give Player157 146 20`

In detail:

`/give`

`Player157`

`146`

`20`

Player number...changes each
time you play the game

146 is block ID

Quantity given





Allocated new blocks to player...

- Give yourself a few of the new blocks...

/give player157 146 20

Player157: Your player name (“/t whatever” to find it)

146: Your block id

20: How many blocks you want to give



Finally...it is time to create our new block in Minecraft!....

Creating our new “kcd_Block”, we’re going to use 2 methods to create it....

(1) using Eclipse, MCP and just Java

(2) using Eclipse, MCP, ModLoader and Java



-
- The screenshot shows an IDE window with the Package Explorer on the left and a Java file on the right. The Package Explorer displays a project structure with a 'net.minecraft' package. A context menu is open over the 'net.minecraft' package, showing options like 'New', 'Go Into', 'Open in New Window', and 'Open Type Hierarchy'. The 'New' option is selected, and a sub-menu is visible with options like 'Java Project', 'Project...', 'Package', and 'Class'. The Java file on the right shows a list of 'public static final' declarations.

Create new class BlockCloverMod



- Set: Automatically generate constructors

The screenshot shows the 'New Java Class' dialog box. The 'Name' field is set to 'BlockCloverMod'. The 'Package' is 'net.minecraft.src'. The 'Superclass' is 'net.minecraft.src.Block'. The 'Modifiers' section has 'public' selected. Under 'Which method stubs would you like to create?', 'Constructors from superclass' and 'Inherited abstract methods' are checked. The 'Generate comments' checkbox is unchecked. The 'Cancel' and 'Finish' buttons are at the bottom.

BlockCloverMod is derived from
“Block”...just like BlockKCD

Create new class mod_CloverBlock



- Set: Automatically generate constructors

A screenshot of the 'New Java Class' dialog box in an IDE. The dialog has a title bar 'New Java Class' and a 'Java Class' section with a red error message 'Type already exists.' and a green 'C' icon. The 'Source folder' is 'Client/src', 'Package' is 'net.minecraft.src', and 'Enclosing type' is empty. The 'Name' field is 'mod_CloverBlock'. The 'Modifiers' section has 'public' selected. The 'Superclass' is 'net.minecraft.src.BaseMod'. The 'Interfaces' section is empty. The 'Which method stubs would you like to create?' section has 'Constructors from superclass' and 'Inherited abstract methods' checked. The 'Do you want to add' section has 'Generate comments' unchecked. The 'Cancel' and 'Finish' buttons are at the bottom right.

Java Class

Type already exists.

Source folder: Client/src Browse...

Package: net.minecraft.src Browse...

Enclosing type: Browse...

Name: mod_CloverBlock

Modifiers: ☒ public ☐ default ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: net.minecraft.src.BaseMod Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ Constructors from superclass
☒ Constructors from superclass
☒ Inherited abstract methods

Do you want to add... [here](#)

☐ Generate comments

Cancel Finish

mod_CloverBlock is derived from “BaseMod”...which means we’re creating a new Mod Class, not a new Block Class

Note: Mod’s MUST start with “mod_”, as in “mod_CloverBlock”, such that the mod is picked up when ModLoader scans for new mods.

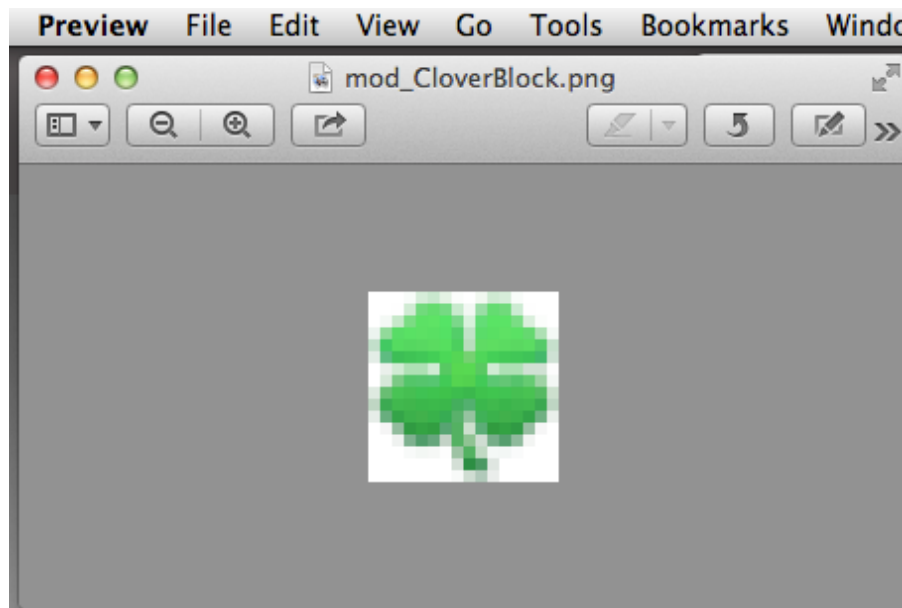


Mod block graphics....

- Placing the “mod_CloverBlock.png” file....

Windows: C:/<YourMCPDirectoryLocation>/mcp726a/bin/minecraft

Mac: MacHD/Users/jim_blair/Desktop/KCD726a/mcp726a/bin/minecraft/

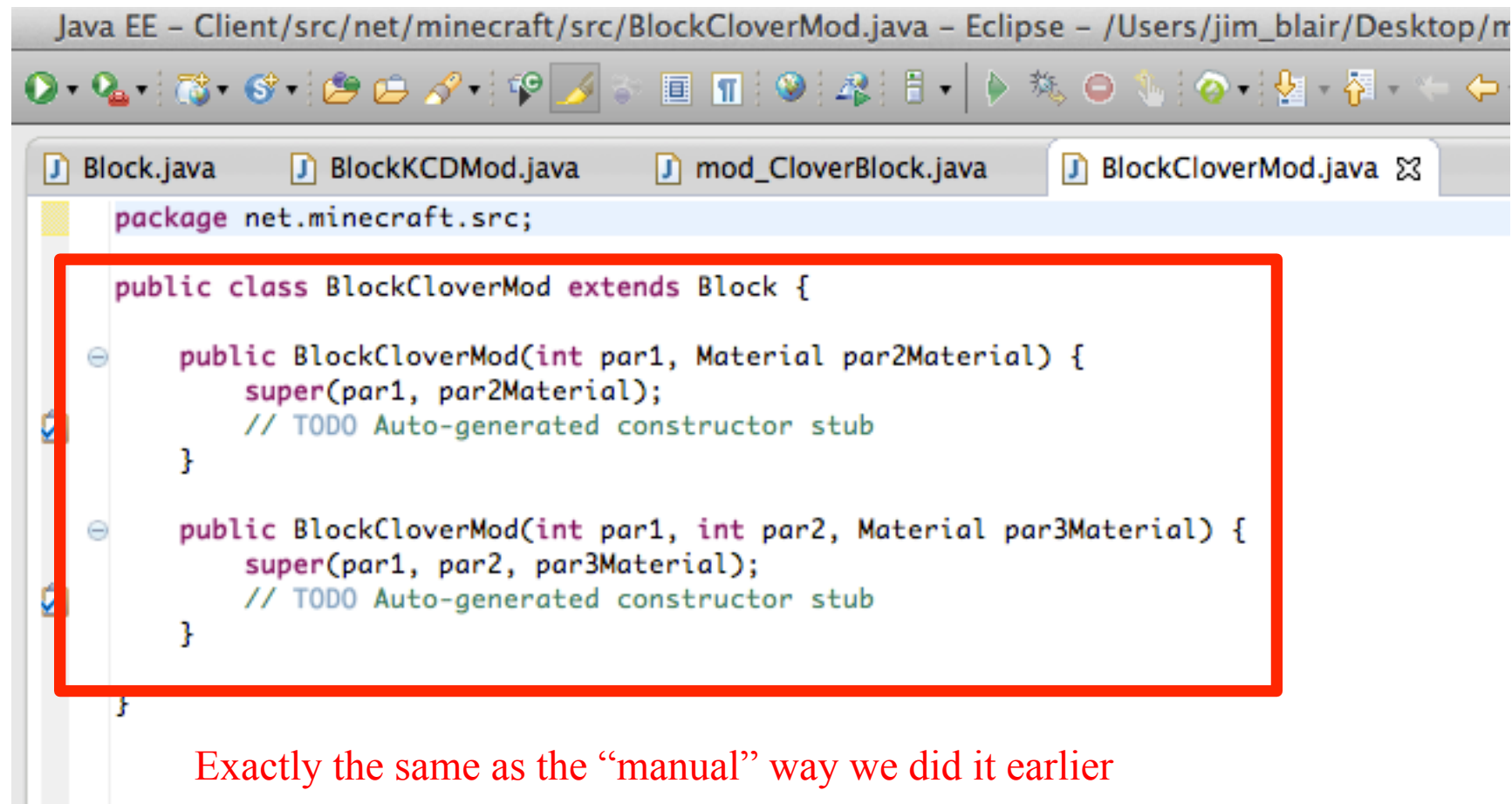


We don't need to know the number, the “mod” via ModLoader, takes care of the ID



Create new class definition...

- Adds new Block class...



```
Java EE - Client/src/net/minecraft/src/BlockCloverMod.java - Eclipse - /Users/jim_blair/Desktop/n

package net.minecraft.src;

public class BlockCloverMod extends Block {

    public BlockCloverMod(int par1, Material par2Material) {
        super(par1, par2Material);
        // TODO Auto-generated constructor stub
    }

    public BlockCloverMod(int par1, int par2, Material par3Material) {
        super(par1, par2, par3Material);
        // TODO Auto-generated constructor stub
    }

}
```

Exactly the same as the “manual” way we did it earlier

Create instance of new class...



New BaseMod class...

We still need to
identify
our new Block ID

We specify our own
icon, in a separate file

```
Sublime Text 2  File  Edit  Selection  Find  View  Goto  Tools  Project  Window  Help  Block.java
CraftingManager.java  mod_CloverBlock.java x
1  package net.minecraft.src;
2
3  public class mod_CloverBlock extends BaseMod {
4
5      public static final Block cloverBlock = (new BlockCloverMod(147, ModLoader.addOverride("/terrain.png", "/mod_CloverBlock.png"), Material.glass)).setBlockName("cloverBlock");
6
7      @Override
8      public String getVersion() {
9          // TODO Auto-generated method stub
10         return "v0.1";
11     }
12
13     public void addRecipes() {
14         ModLoader.addRecipe(new ItemStack(cloverBlock, 20), new Object[] { "##", "# ", Character.valueOf('#'), Block.dirt });
15     }
16
17     @Override
18     public void load() {
19         ModLoader.registerBlock(cloverBlock);
20         ModLoader.addName(cloverBlock, "cloverBlock");
21         this.addRecipes();
22     }
23
24 }
25
```

registerBlock and addName ensure
Block is advertised



CREATING “RECIPES” FOR OUR BLOCKS

1. Recipe for “manual” Block, `kcdBlock`
2. Recipe for our “Mod” Block, `cloverBlock`

Create “Recipe” for our new first block



```
Sublime Text 2  File  Edit  Selection  Find  View  Goto  Tools  Project  Window  Help  CraftingManager.java

CraftingManager.java  mod_CloverBlock.java x
118  this.addRecipe(new ItemStack(Block.woodenButton, 1), new Object[] {"#", "#", Block.planks});
119  this.addRecipe(new ItemStack(Block.pressurePlateStone, 1), new Object[] {"###", "#", Block.stone});
120  this.addRecipe(new ItemStack(Block.pressurePlatePlanks, 1), new Object[] {"###", "#", Block.planks});
121  this.addRecipe(new ItemStack(Block.dispenser, 1), new Object[] {"###", "#X#", "#R#", "#", Block.cobblestone, 'X', I
122  this.addRecipe(new ItemStack(Block.pistonBase, 1), new Object[] {"TTT", "#X#", "#R#", "#", Block.cobblestone, 'X',
123  this.addRecipe(new ItemStack(Block.pistonStickyBase, 1), new Object[] {"S", "P", 'S', Item.slimeBall, 'P', Block.pi
124  this.addRecipe(new ItemStack(Item.bed, 1), new Object[] {"###", "XXX", '#', Block.cloth, 'X', Block.planks});
125  this.addRecipe(new ItemStack(Block.enchantmentTable, 1), new Object[] {" B ", "D#D", "###", '#', Block.obsidian, 'B
126  this.addRecipe(new ItemStack(Block.anvil, 1), new Object[] {"III", " i ", "iii", 'I', Block.blockSteel, 'i', Item.i
127  this.addShapelessRecipe(new ItemStack(Item.enderEye, 1), new Object[] {Item.enderPearl, Item.blazePowder});
128  this.addShapelessRecipe(new ItemStack(Item.fireballCharge, 3), new Object[] {Item.gunpowder, Item.blazePowder, Item
129  this.addShapelessRecipe(new ItemStack(Item.fireballCharge, 3), new Object[] {Item.gunpowder, Item.blazePowder, new
130  this.addRecipe(new ItemStack(Block.kcdBlock, 20), new Object[] {"###", "##", Character.valueOf('#'), Block.dirt});
131  Collections.sort(this.recipes, new RecipeSorter(this));
```

Add new block recipe for our first block,
“kcdBlock”

Create “Recipe” for our new “mod” block



Add new block recipe
directions

```
1 package net.minecraft.src;
2
3 public class mod_CloverBlock extends BaseMod {
4
5     public static final Block cloverBlock = (new BlockCloverMod(147, ModLoader.addOverride("/terrain.png", "/mod_CloverBlock.png"), Material.glass)).setBlockName("cloverBlock");
6
7     @Override
8     public String getVersion() {
9         // TODO Auto-generated method stub
10        return "v0.1";
11    }
12
13    public void addRecipes() {
14        ModLoader.addRecipe(new ItemStack(cloverBlock, 20), new Object[] { "##", "# ", Character.valueOf('#'), Block.dirt });
15    }
16
17    @Override
18    public void load() {
19        ModLoader.registerBlock(cloverBlock);
20        ModLoader.addBlock(cloverBlock, "cloverBlock");
21        this.addRecipes();
22    }
23
24 }
25
```

Finally, we rebuild Minecraft, with new mod/
Block, and two new Recipes...



Recompile and run...(Windows)

```
C:\MinecraftMod\mcp72> recompile.bat
== MCP 7.2 (data: 7.2, client: 1.3.2, server: 1.3.2) ==
...

C:\MinecraftMod\mcp72> startclient.bat
== MCP 7.2 (data: 7.2, client: 1.3.2, server: 1.3.2) ==
...
[23:26:35] Setting user: Player157, -
```

*Record Player number
(in this case, "Player157")*

OR

Recompile and run...(Mac)

```
$ sh recompile.sh
== MCP 7.2 (data: 7.2, client: 1.4.7, server: 1.4.7) ==
# found ff, ff patches, srgs, name csvs
== Recompiling client ==
> Cleaning bin
> Recompiling
- Done in 5.70 seconds
== Recompiling server ==
> Cleaning bin
> Recompiling
- Done in 4.62 seconds
$ sh startclient.sh
== MCP 7.2 (data: 7.2, client: 1.4.7, server: 1.4.7) ==
[20:32:34] 27 achievements
[20:32:34] 210 recipes
[20:32:35] Setting user: Player157
```

Run Minecraft....



New “kcdBlock”

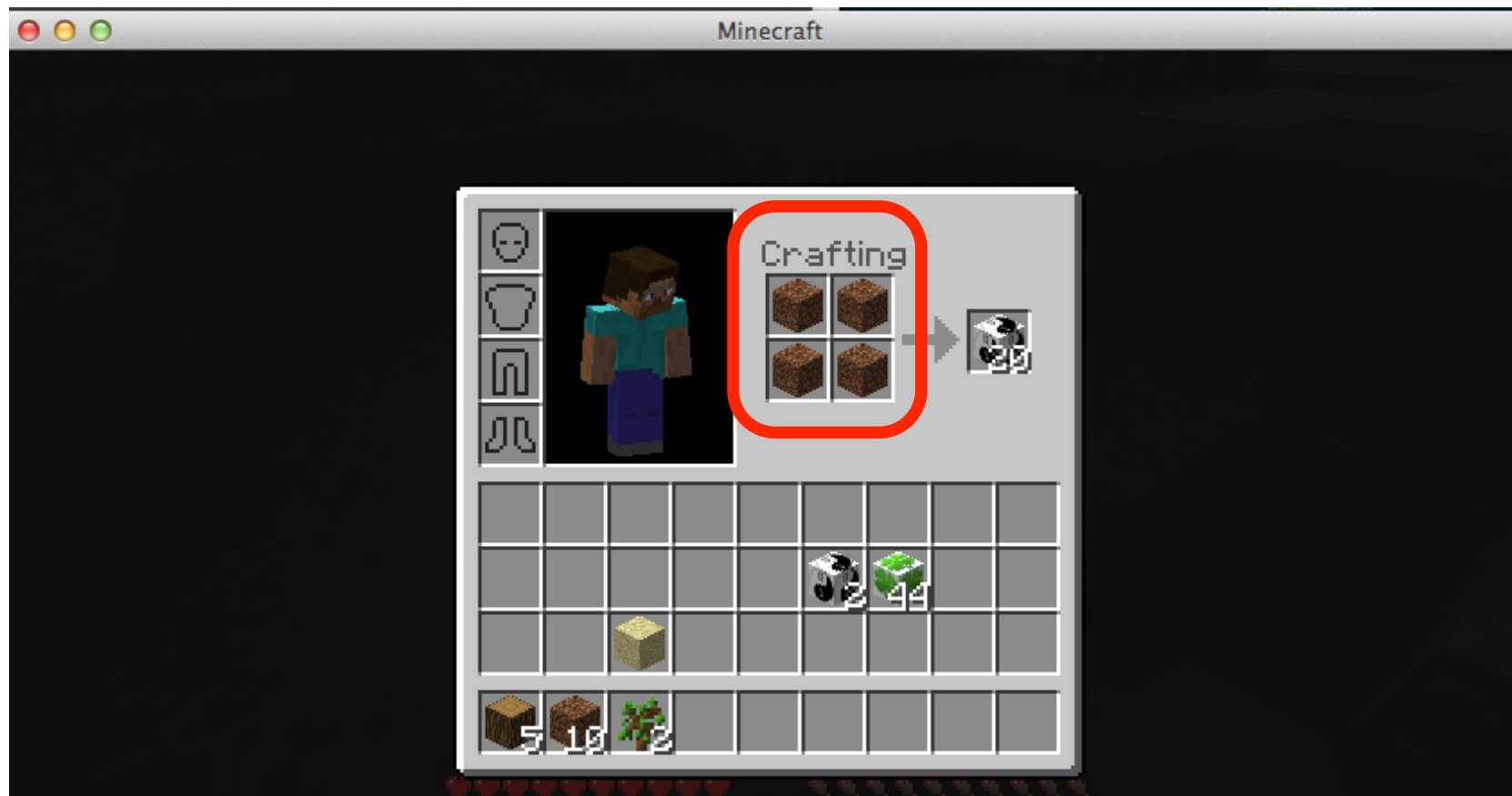
New “cloverBlock”



Run Minecraft....

- Crafting “kcdBlock” with 4 Dirt blocks
 - Remember....

```
this.addRecipe(new ItemStack(Block.kcdBlock, 20), new Object[] {"##", "##", Character.valueOf('#'), Block.dirt});
```

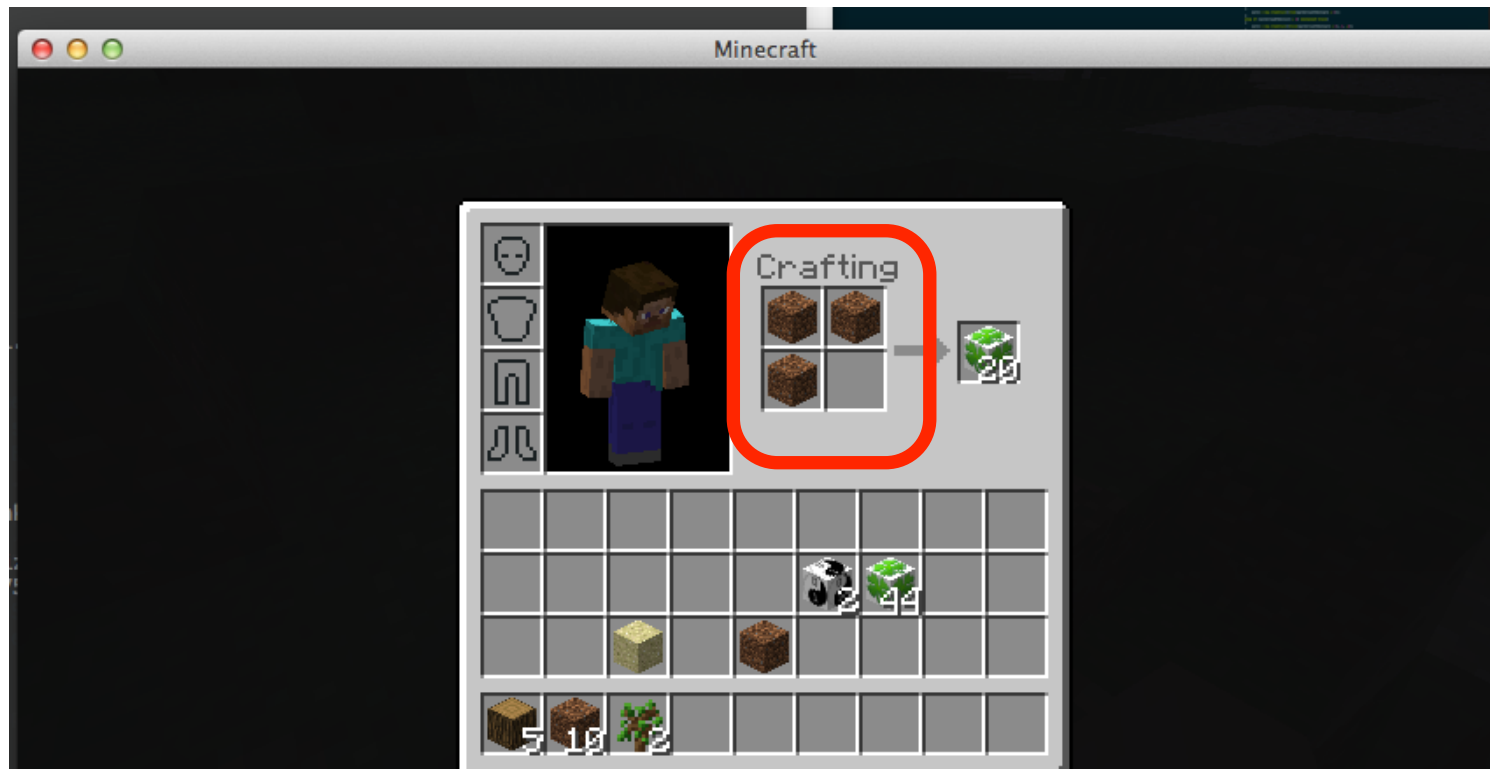




Run Minecraft....

- Crafting “mod_CloverBlock” with 3 Dirt blocks
 - Remember...

```
public void addRecipes() {  
    ModLoader.addRecipe(new ItemStack(cloverBlock, 20), new Object[] { "##", "# ", Character.valueOf('#'), Block.dirt });  
}
```





TA DA!!!!.....WE'RE ALL DONE!

- Modding can be used to create all sorts of wonderful additions to Minecraft....only limited by your imagination! It is hoped that these Block and Recipe sessions will give you the spark to go create fantastic new things in Minecraft!