Ore Generation

Posted on July 17, 2018

Ore Generation

Let's add some world generation into our mod that makes it so our tutorial block generates in the world. The first step to this is to create the class TutorialWorldGen in the package com.cubicoder.tutorialmod.world.gen. This class will implement IWorldGenerator. After you add the unimplemented method, create a constructor and another, private method named genStandard().

```
package com.cubicoder.tutorial.world.gen;

import java.util.Random;

import net.minecraft.world.World;
import net.minecraft.world.chunk.IChunkProvider;
import net.minecraft.world.gen.IChunkGenerator;
import net.minecraft.world.gen.feature.WorldGenerator;
import net.minecraftforge.fml.common.IWorldGenerator;

public class TutorialWorldGen implements IWorldGenerator {

    public TutorialWorldGen() {

    }

    @Override
    public void generate(Random random, int chunkX, int chunkZ, World wor

    }

    private void genStandard(WorldGenerator generator, World world, Rando

    }

}
```

Next, we need to add a field to hold the WorldGenMinable instance that will be used to generate the ore. We will initialize this in the constructor.

```
// ...

private final WorldGenMinable tutorialOverworldGenerator;

public TutorialWorldGen() {
    tutorialOverworldGenerator = new WorldGenMinable(TutorialBlocks.BASIC
}

// ...
```

The first parameter in the WorldGenMinable constructor specifies the block (specifically, the block***state***) that will be generated. The second parameter is the average vein size, and the third parameter is the block it will be replacing. Now we need to write our standard generation code in the genStandard() method.

```
// ...

private void genStandard(WorldGenerator generator, World world, Random ra
    if(minHeight < 0) minHeight = 0;
    if(maxHeight > 255) maxHeight = 255;

    if(maxHeight < minHeight) {
        int i = minHeight;
        minHeight = maxHeight;
        maxHeight = i;
    } else if(maxHeight == minHeight) {
        if(maxHeight < 255) {
            maxHeight++;
        } else minHeight--;
    }

    BlockPos chunkPosAsBlockPos = new BlockPos(chunkX << 4, 0, chunkZ <<
    int heightDiff = maxHeight - minHeight + 1;

    for (int i = 0; i < spawnTries; i++) {
        generator.generate(world, random,
                chunkPosAsBlockPos.add(
                        random.nextInt(16),
                        minHeight + random.nextInt(heightDiff),
                        random.nextInt(16)
                )
        );
    }
}
```

The first part of this code simply does checks to make sure that the minHeight and maxHeight values are valid. Then, it finds the chunk position and randomly generates the ore based on the values we enter. It does this once of each spawn try.

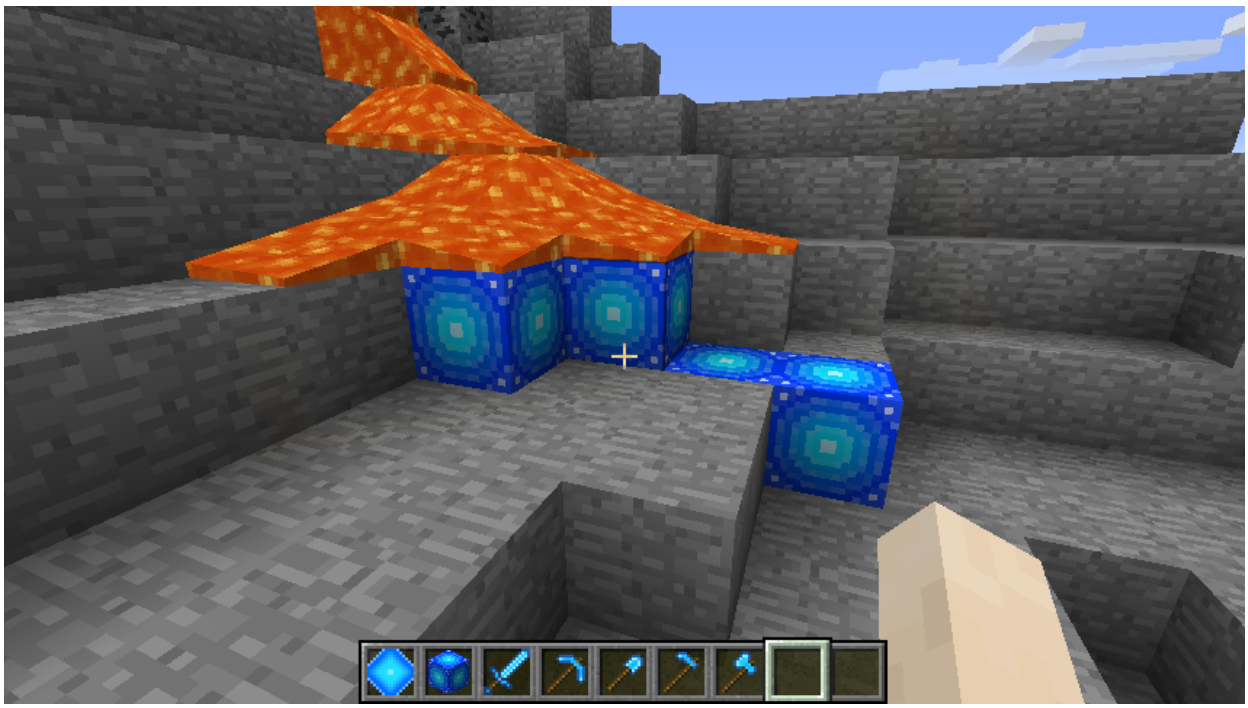
Finally, we need to call the genStandard() method in the generate() method for it to generate. In order to do this, we must check the dimension we are in, so that we can make sure it only spawns in the Overworld.

```
@Override
public void generate(Random random, int chunkX, int chunkZ, World world,
    switch (world.provider.getDimensionType()) {

    case NETHER:
        break;

    case OVERWORLD:
        genStandard(genTutorial_overworld, world, random, chunkX, chunkZ,
        break;

    case THE_END:
        break;

    }
}
```

Finally, we need to register our ore generator. In your main class's init() method write this line of code:

```
GameRegistry.registerWorldGenerator(new TutorialWorldGen(), 0);
```

Then look in the world to find your ore! Note that you'll have to travel far away or create a new world in order to find it, as this does not generate in already-generated chunks.



(htt... (ht... (ht...

(/feed.xml) (https://github.com/cubicoder)