

## Questions and Exercises

*These questions and exercises is an opportunity to see what you've learnt from the lecture as well as practice the new things we've been talking about. In other words, these questions and exercises are completely optional but it's recommended to do them. In the end of the document you will find the answers to the questions as well as possible solutions to the exercises, note that one can solve an exercise in different ways. There will also be some suggestions about what one could code if one want to continue with some more advanced things. These suggestions will not come with a possible solution and might include things that haven't been covered in the lecture.*

### Question 1

What happens when a block is picked up and we want to place it again? What if we want to change this behaviour?

### Question 2

What is Metadata? Which values can it have?

### Exercise

Create a block with 8 different colors(subtypes). Give them different names. Each color should have two different states which should be swapped when right-clicked. Make sure that each color can be found in the creative window but just once(not once for each state).

*If you don't want to create the icons yourself you can find some example icons at the link below <https://dl.dropboxusercontent.com/u/46486053/MarkerTextures.zip>*

### Further explorations 1

Improve the BlockMachine from the lecture so it will work with redstone properly. It should only be "activated" when a new pulse is received(i.e. When the redstone is active but it wasn't active the last time). Use metadata to achieve this.

### Further explorations 2

Create a block with 6 different icons (one for each side). Allow the user to rotate the block by clicking on it.

## Answers and solutions

### Answer to Question 1

*By Vidar Swenning*

Every block needs an associated item, this item is used when holding the block in your hand (even though the item is rendered to look like a block). When right clicking with the block item it places the block version of itself. If one wants to change the behaviour of the placing or change something with the item of the block one can create an item for the block. If this isn't done a standard item is used for the block, this is good enough in most of the cases.

### Answer to Question 2

The metadata of a block in the world is a integer value to store extra information in. Since one only creates one instance of the Block class you can't store the information there. For this reason the metadata is used. A metadata value can only be in the range 0 to 15 (inclusive).

### Possible solution to Exercise

```
package example.blocks;
```

```
import java.util.List;
```

```
import net.minecraft.block.Block;
import net.minecraft.block.material.Material;
import net.minecraft.client.renderer.texture.IconRegister;
import net.minecraft.creativetab.CreativeTabs;
import net.minecraft.entity.player.EntityPlayer;
import net.minecraft.item.ItemStack;
import net.minecraft.util.Icon;
import net.minecraft.world.World;
import cpw.mods.fml.relauncher.Side;
import cpw.mods.fml.relauncher.SideOnly;
```

```
//the block class
```

```
public class BlockMarker extends Block {
```

```
    public BlockMarker(int id) {
        super(id, Material.rock);

        setStepSound(Block.soundStoneFootstep);
        setHardness(2F);
        setCreativeTab(CreativeTabs.tabDecorations);
    }
```

```
    @SideOnly(Side.CLIENT)
    private Icon[] icons;
```

```
    @Override
    @SideOnly(Side.CLIENT)
    public void registerIcons(IconRegister register) {
        icons = new Icon[BlockInfo.MARKER_COLORS.length * 2];
    }
```

```

        for(int i = 0; i < BlockInfo.MARKER_COLORS.length; i++) {
            icons[i * 2] = register.registerIcon(BlockInfo.TEXTURE_LOCATION + ":" +
BlockInfo.MARKER_TEXTURE + BlockInfo.MARKER_COLORS[i].toLowerCase());
            icons[i * 2 + 1] = register.registerIcon(BlockInfo.TEXTURE_LOCATION +
":" + BlockInfo.MARKER_TEXTURE + BlockInfo.MARKER_COLORS[i].toLowerCase() +
BlockInfo.MARKER_MARKED);
        }
    }

    @Override
    @SideOnly(Side.CLIENT)
    public Icon getIcon(int side, int meta) {
        return icons[meta];
    }

    @Override
    @SideOnly(Side.CLIENT)
    public void getSubBlocks(int id, CreativeTabs tab, List list) {
        for (int i = 0; i < BlockInfo.MARKER_COLORS.length; i++) {
            list.add(new ItemStack(id, 1, i * 2));
        }
    }

    @Override
    public int damageDropped(int meta) {
        return meta;
    }

    @Override
    public boolean onBlockActivated(World world, int x, int y, int z, EntityPlayer player, int
side, float hitX, float hitY, float hitZ) {
        if (!world.isRemote) {
            int meta = world.getBlockMetadata(x, y, z);

            int type = meta / 2;
            int state = meta % 2 == 0 ? 1 : 0;

            int newMeta = type * 2 + state;

            world.setBlockMetadataWithNotify(x, y, z, newMeta, 3);
        }

        return true;
    }
}

```

```
package example.items;

import java.util.List;

import cpw.mods.fml.relauncher.Side;
import cpw.mods.fml.relauncher.SideOnly;
import example.blocks.BlockInfo;
import net.minecraft.entity.player.EntityPlayer;
import net.minecraft.item.ItemBlock;
import net.minecraft.item.ItemStack;

//the item class
public class ItemMarker extends ItemBlock {

    public ItemMarker(int id){
        super(id);
        setHasSubtypes(true);
    }

    @Override
    public int getMetadata(int dmg) {
        return dmg;
    }

    @Override
    public String getUnlocalizedName(ItemStack stack) {
        return BlockInfo.MARKER_UNLOCALIZED_NAME + stack.getItemDamage() / 2;
    }

    @Override
    @SideOnly(Side.CLIENT)
    public void addInformation(ItemStack stack, EntityPlayer player, List list, boolean
extraInfo) {
        if (stack.getItemDamage() % 2 == 1) {
            list.add("Marked");
        }
    }
}

package example.blocks;

//Information class
public class BlockInfo {
```

```

    public static final String TEXTURE_LOCATION = "example";

    public static int MARKER_ID;
    public static final String MARKER_KEY = "Marker";
    public static final int MARKER_DEFAULT = 2077;

    public static final String MARKER_UNLOCALIZED_NAME = "markerBlock";
    public static final String MARKER_NAME = "Marker Block";

    public static final String MARKER_TEXTURE = "marker_";
    public static final String MARKER_MARKED = "_marked";

    public static final String[] MARKER_COLORS = {"Blue", "Brown", "Green", "Grey",
"Orange", "Purple", "Red", "Yellow"};

}

package example.blocks;

import net.minecraft.block.Block;
import net.minecraft.item.ItemStack;
import cpw.mods.fml.common.registry.GameRegistry;
import cpw.mods.fml.common.registry.LanguageRegistry;
import example.items.ItemMarker;
public class Blocks {

    public static Block marker;

    //called from the mod's pre-init
    public static void init() {
        marker = new BlockMarker(BlockInfo.MARKER_ID);
        GameRegistry.registerBlock(marker, ItemMarker.class, BlockInfo.MARKER_KEY);
    }

    //called from the mod's init
    public static void addNames() {
        for (int i = 0; i < BlockInfo.MARKER_COLORS.length; i++) {
            LanguageRegistry.addName(new ItemStack(marker, 1, i * 2),
BlockInfo.MARKER_COLORS[i] + " " + BlockInfo.MARKER_NAME);
        }
    }

}

```