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 recomended 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
In a chest there are one stack of 10 iron ingots, one stack of 25 iron ingots and one stack of 64 coal. How many Items are represented in the chest?

Question 2
Given the following code to register an icon, where should the image be location and what should it be named?

itemIcon = register.registerIcon(”test:stuff”);

Question 3
Why should we use the item damage rather than just adding a field in the item class when storing values?

Exercise
Create an item with five subtypes, these should all have different names and different icons. If put in a crafting table one should get the next subitem (if you put the last on in the crafting table you should get the first one). Add a description to the subitems so it tells you which item is the one (the one you will get when crafting). Also make sure that all of your items appear in the creative inventory.

If you don't want to make your own textures you can download five textures for this purpose at the link below. These textures are designed to fit Further explorations 2 as well.

https://dl.dropboxusercontent.com/u/46486053/Deathstones.zip

Further explorations 1
Create a custom creative window tab for the items in the mod. Use the Wand's icons as the icon of the Creative tab. To do this you want to extend the CreativeTabs class with your own tab class.

Further explorations 2
Continue with the item from Exercise 1 and make it so when you hit a mob the item (not activate, hit it by left clicking) it should die if it is of the correct type. The correct type depends on which of the sub items the player is using. For instance, one might instantly kill endermen while another one might instantly kill creepers. When doing this you'll need the method hitEntity to be able to do things when you hit an entity with it(take a look in the Item class for the parameters). To check if the mob is of the correct type you can use instanceof, for instance ”if (target instanceof EntityCreeper)” would check if the target is a creeper.

Further explorations 3
Continue on the code from Further explorations 2 and make sure that the item breaks after being used. After this has been done, change it so you can use it three times before it breaks. Observe that you still should have five different subtypes. Make sure that the crafting recipes preserve the amounts you have left. So when you swap type of an item with two uses left it should still have two uses left afterwards.

Answers and solutions

Answer to Question 1
In the chest there are two Items represented: irong ingot and coal. In minecraft every item just exists once, a sort of base item. The different stacks we see however, refers to that base item as well as containing information about the amount as well as the item damage of the stack. These stacks are called ItemStacks in the code. In the chest there are three ItemStacks.

Answer to Question 2
The location of the image of course depends where your mcp folder is located(and what you have named your mcp folder) but inside the mcp folder it should be located like the following:

/src/minecraft/assets/test/textures/items/stuff.png

Answer to Question 3
Since there is only one instance of the Item, saving something there means that every ItemStack using this item would share that value. To make the value unique of different instances we need to store it in the ItemStack itself. To do this we modify the damage value of the ItemStack(we can' t modify the ItemStack class).

Possible solution to Exercise

```
package example.items;
```

```java
import java.util.List;

import net.minecraft.client.renderer.texture.IconRegister;
import net.minecraft.creativetab.CreativeTabs;
import net.minecraft.item.Item;
import net.minecraft.item.ItemStack;
import net.minecraft.util.Icon;
import cpw.mods.fml.relauncher.Side;
import cpw.mods.fml.relauncher.SideOnly;

//The item class
public class ItemDeathstone extends Item {

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

        public ItemDeathstone(int id) {
                super(id);
                setCreativeTab(CreativeTabs.tabCombat);
                setHasSubtypes(true);
        }

        @Override
        public String getUnlocalizedName(ItemStack itemstack) {
                return ItemInfo.STONE_UNLOCALIZED_NAME + itemstack.getItemDamage();
        }

        @Override
        @SideOnly(Side.CLIENT)
        public void registerIcons(IconRegister register) {
                icons = new Icon[ItemInfo.STONE_ICONS.length];

                for (int i = 0; i < icons.length; i++) {
                        icons[i] = register.registerIcon(ItemInfo.TEXTURE_LOCATION + ":" + ItemInfo.STONE_ICONS[i]);
                }
        }

        @Override
        @SideOnly(Side.CLIENT)
        public Icon getIconFromDamage(int dmg) {
                return icons[dmg];
        }

        @Override
        @SideOnly(Side.CLIENT)
        public void getSubItems(int id, CreativeTabs tabs, List list) {
                for (int i = 0; i < ItemInfo.STONE_NAMES.length; i++) {
```

```
				ItemStack itemstack = new ItemStack(id, 1, i);
				list.add(itemstack);
			}
		}

}


package example.items;

//contains the required information, don't forget to load the STONE_ID from a config file
public class ItemInfo {
		public static final String TEXTURE_LOCATION = "example";

		public static int STONE_ID;
		public static final String STONE_KEY = "Stone";
		public static final int STONE_DEFAULT = 24203;

		public static final String STONE_UNLOCALIZED_NAME = "deathStone";
		public static final String[] STONE_NAMES = {"Stone of Enderman death", "Stone of Pig
death", "Stone of Creeper death", "Stone of Skeleton death", "Stone of Player death"};

		public static final String[] STONE_ICONS = {"deathstone_enderman", "deathstone_pig",
"deathstone_creeper", "deathstone_skeleton", "deathstone_player"};

}


package example.items;

import net.minecraft.client.resources.Language;
import net.minecraft.item.Item;
import net.minecraft.item.ItemStack;
import cpw.mods.fml.common.registry.GameRegistry;
import cpw.mods.fml.common.registry.LanguageRegistry;

//the class creating the items
public class Items {

		public static Item stone;

		//should be called from the preinit method in the mod class
		public static void init() {
			stone = new ItemDeathstone(ItemInfo.STONE_ID);
		}

		//should be called from the init method in the mod class
		public static void addNames() {
```

```
		for (int i = 0; i < ItemInfo.STONE_NAMES.length; i++) {
			LanguageRegistry.addName(new ItemStack(stone, 1, i),
ItemInfo.STONE_NAMES[i]);
		}
	}

	//should be called from the init method in the mod class
	public static void registerRecipes() {
		for (int i = 0; i < ItemInfo.STONE_NAMES.length; i++) {
			ItemStack current = new ItemStack(stone, 1, i);
			ItemStack next = new ItemStack(stone, 1, (i + 1) %
ItemInfo.STONE_NAMES.length);

			GameRegistry.addRecipe(next, "S", 'S', current);
		}
	}
}
```