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
How does one spawn an entity in the world?

Question 2
When moving an entity around in the world, why is it good to do so on the client side as well?

Question 3
What is the IEntityAdditionalSpawnData interface used for?

Question 4
What is the datawatcher of an entity?

Exercise 1
Make an item that can spawn an entity on the ground. When being used the amounts of items should be decreased (i.e. the item is placed as an entity). The entity should fly 30 blocks into the air and then come back. When it comes back to the start position it should explode and die.

The entity does not have to be rendered in a fancy way, the default look is fine.

If you don't want to create your own icon for the item you can get it from the link below.
https://dl.dropboxusercontent.com/u/46486053/simple_icon.png

Exercise 2
Make an item that you can throw like a snowball. If you hit a pig with it, convert the pig to a pig man. The entity that is used for throwing should be rendered to look like the item.

Take a look at the EntitySnowball class or the EntityEgg class to get some help.

If you don't want to create your own icon you can get one from the link below
https://dl.dropboxusercontent.com/u/46486053/pig_converter.png

Further explorations 1
Create a projectile that can be fired from a dispenser. When the projectile hits a block make the dispenser destroy the block and spit it out behind the dispenser. If it hit an entity, teleport the entity to the dispenser.

Further explorations 2
Expand the Spaceship code from the lecture. Allow the player to fly around with the spaceship using the keyboard.

Answers and solutions

Answer to Question 1
To spawn an entity in the world, create a new instance of it and use the method following code to spawn it

world.spawnEntityInWorld(entity);

where world is the world to spawn it in and entity is the variable of the new instance. One might also want to change values of fields and call methods of the instance before spawning it in the world, but that's optional.

Answer to Question 2
Moving the entity on the server defines where the entity currently is. By moving it on the server it will also sync it to the client side so it will be moved there as well. However, this is not synced every tick so the movement on the client side won't be smooth if the server is constantly moving the entity. Therefore if the client simulates the movement you'll get a smoother result. The server should still control most of the movement though.

Answer to Question 3
The IEntityAdditionalSpawnData interface can be used to easily give the entity on the client side some extra information when being created. The server side entity can write some data and the client side can then read it. Observe that this writing and reading happens once each time the entity is spawned, so one doesn't have to bother about game restarts or anything. If the server saves and loads the information on game restart the methods of the IEntityAdditionalSpawnData interface will synchronize the data at that time as well.

Answer to Question 4
The datawatcher of an entity can contain up to 32 different values. These values can be used on both the server and the client, since they are automatically synced. These values are a bit more restricted than the IEntityAdditionalSpawnData interface but can be used to sync values which changes. They will be synchranize as soon as they change.

Possible solution to Exercise 1

```
package example.items;

import cpw.mods.fml.relauncher.Side;
import cpw.mods.fml.relauncher.SideOnly;
import net.minecraft.client.renderer.texture.IconRegister;
import net.minecraft.creativetab.CreativeTabs;
import net.minecraft.entity.player.EntityPlayer;
import net.minecraft.item.Item;
import net.minecraft.item.ItemStack;
import net.minecraft.world.World;
import example.entities.EntityBouncer;

public class ItemBouncer extends Item {

	public ItemBouncer(int id) {
		super(id);

		setCreativeTab(CreativeTabs.tabCombat);
		setUnlocalizedName(ItemInfo.BOUNCER_UNLOCALIZED_NAME);
	}


	@Override
	@SideOnly(Side.CLIENT)
	public void registerIcons(IconRegister register) {
		itemIcon = register.registerIcon(ItemInfo.TEXTURE_LOCATION + ":" +
ItemInfo.BOUNCER_TEXTURE);
	}

	@Override
	public boolean onItemUseFirst(ItemStack stack, EntityPlayer player, World world, int x, int
y, int z, int side, float hitX, float hitY, float hitZ) {
		--stack.stackSize;

		if (!world.isRemote)  {
			EntityBouncer bouncer = new EntityBouncer(world);


			bouncer.setStartPosition(x + 0.5, y + 1.5, z + 0.5);
```

```
                world.spawnEntityInWorld(bouncer);

                return true;
            }else{
                return false;
            }
        }

}

package example.entities;

import net.minecraft.entity.Entity;
import net.minecraft.nbt.NBTTagCompound;
import net.minecraft.world.World;

public class EntityBouncer extends Entity {

        private int startPosY;
        private boolean up;

        public EntityBouncer(World world) {
            super(world);
        }

        public void setStartPosition(double x, double y, double z) {
            up = true;
            startPosY = (int)y;

            setPosition(x, y, z);
        }

        @Override
        protected void entityInit() {

        }

        @Override
        public void onUpdate() {
            super.onUpdate();

            if (!worldObj.isRemote) {
                if (up) {
                    motionY = 0.4F;
                    if (posY > startPosY + 30) {
                        up = false;
                    }
                }else{
```

```
                        motionY = -1.8F;
                        if (posY < startPosY) {
                                worldObj.createExplosion(this, posX, posY, posZ, 6, true);
                                setDead();
                        }
                }
        }

        setPosition(posX + motionX,  posY + motionY, posZ + motionZ);
    }

    @Override
    protected void readEntityFromNBT(NBTTagCompound compound) {
        up = compound.getBoolean("Up");
        startPosY = compound.getShort("Start");
    }

    @Override
    protected void writeEntityToNBT(NBTTagCompound compound) {
        compound.setBoolean("Up", up);
        compound.setShort("Start", (short)startPosY);
    }



}
```

```
package example.items;

public class ItemInfo {

    public static final String TEXTURE_LOCATION = "example";

    public static int BOUNCER_ID;
    public static final String BOUNCER_KEY = "Bouncer";
    public static final int BOUNCER_DEFAULT = 24203;

    public static final String BOUNCER_UNLOCALIZED_NAME = "bouncerItem";
    public static final String BOUNCER_NAME = "Dangerous Bouncer";

    public static final String BOUNCER_TEXTURE = "simple_icon";
}
```

```
package example.items;

import net.minecraft.item.Item;
import net.minecraft.item.ItemStack;
import cpw.mods.fml.common.registry.GameRegistry;
```

```
import cpw.mods.fml.common.registry.LanguageRegistry;

public class Items {


	public static Item bouncer;

	//called from the mod's pre-init
	public static void init() {
		bouncer = new ItemBouncer(ItemInfo.BOUNCER_ID);
	}

	//called from the mod's init
	public static void addNames() {
		LanguageRegistry.addName(bouncer, ItemInfo.BOUNCER_NAME);
	}

}
```

```
package example.entities;

import cpw.mods.fml.common.registry.EntityRegistry;
import example.StevesExample;

public class Entities {

	//called from the mod's init
	public static void init() {
		EntityRegistry.registerModEntity(EntityBouncer.class, "EntityBouncer", 2,
StevesExample.instance, 80, 3, true);
	}

}
```

Possible solution to Exercise 2

```
package example.entities;

import net.minecraft.entity.EntityLivingBase;
import net.minecraft.entity.monster.EntityPigZombie;
import net.minecraft.entity.passive.EntityPig;
import net.minecraft.entity.projectile.EntityThrowable;
import net.minecraft.util.MovingObjectPosition;
import net.minecraft.world.World;

public class EntityPigConverter extends EntityThrowable {
   public EntityPigConverter(World world) {
      super(world);
   }
```

```
    public EntityPigConverter(World world, EntityLivingBase thrower) {
        super(world, thrower);
    }

    public EntityPigConverter(World world, double x, double y, double z) {
        super(world, x, y, z);
    }


    @Override
    protected void onImpact(MovingObjectPosition position) {
        if (!worldObj.isRemote) {
            if (position.entityHit != null && position.entityHit instanceof EntityPig) {
                    EntityPig pig = (EntityPig)position.entityHit;

                EntityPigZombie zombie = new EntityPigZombie(this.worldObj);
                        zombie.setLocationAndAngles(pig.posX,  pig.posY,  pig.posZ,  pig.rotationYaw,
pig.rotationPitch);
                worldObj.spawnEntityInWorld(zombie);

                pig.setDead();
            }

            setDead();
        }
    }
}

package example.items;

import net.minecraft.client.renderer.texture.IconRegister;
import net.minecraft.creativetab.CreativeTabs;
import net.minecraft.entity.player.EntityPlayer;
import net.minecraft.item.Item;
import net.minecraft.item.ItemStack;
import net.minecraft.world.World;
import cpw.mods.fml.relauncher.Side;
import cpw.mods.fml.relauncher.SideOnly;
import example.entities.EntityPigConverter;

public class ItemPigConverter extends Item {
    public ItemPigConverter(int id) {
        super(id);
        setCreativeTab(CreativeTabs.tabCombat);
        setUnlocalizedName(ItemInfo.PIGGY_UNLOCALIZED_NAME);
    }
```

```
    @Override
    public ItemStack onItemRightClick(ItemStack stack, World world, EntityPlayer player) {
        if (!player.capabilities.isCreativeMode) {
            --stack.stackSize;
          }

        if (!world.isRemote) {
            world.spawnEntityInWorld(new EntityPigConverter(world, player));
        }

        return stack;
    }

    @Override
    @SideOnly(Side.CLIENT)
    public void registerIcons(IconRegister register) {
        itemIcon = register.registerIcon(ItemInfo.TEXTURE_LOCATION + ":" +
ItemInfo.PIGGY_TEXTURE);
    }
}

package example.items;

public class ItemInfo {

        public static int PIGGY_ID;
        public static final String PIGGY_KEY = "Pig_Converter";
        public static final int PIGGY_DEFAULT = 24204;

        public static final String PIGGY_UNLOCALIZED_NAME = "piggyConverter";
        public static final String PIGGY_NAME = "Pig Converter";

        public static final String PIGGY_TEXTURE = "pig_converter";
}

package example.items;

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

public class Items {


        public static Item piggy;

        //called from the mod's pre-init
```

```
	public static void init() {
		piggy = new ItemPigConverter(ItemInfo.PIGGY_ID);
	}

	//called from the mod's init
	public static void addNames() {
		LanguageRegistry.addName(piggy, ItemInfo.PIGGY_NAME);
	}

}

package example.entities;

import cpw.mods.fml.common.registry.EntityRegistry;
import example.StevesExample;

public class Entities {

	public static void init() {
		EntityRegistry.registerModEntity(EntityPigConverter.class, "EntityPigConverter", 3,
StevesExample.instance, 80, 3, true);
	}

}

package example.proxies;

import net.minecraft.client.renderer.entity.RenderSnowball;
import cpw.mods.fml.client.registry.RenderingRegistry;
import example.entities.EntityPigConverter;
import example.items.Items;

public class ClientProxy extends CommonProxy {


	@Override
	public void initRenderers() {
		RenderingRegistry.registerEntityRenderingHandler(EntityPigConverter.class, new
RenderSnowball(Items.piggy));
	}

}
```