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
Given the following code, where event is a SoundLoadEvent, where should one put the sound file that is about to be registered?

event.manager.soundPoolSounds.addSound(”test:hey.ogg”);

Question 2
Given the sound from Question 1, how can one play it at a specific location in the world?

Question 3
How can one spawn a vanilla particle?

Question 4
How does one create a simple particle type? How can one give it an icon?

Question 5
The following code to add a own made particle works, however it has some flaws. What?

Minecraft.getMinecraft().effectRenderer.addEffect(new EntityTestingFX(world, particleX, particleY, particleZ, particleMotionX, particleMotionY, particleMotionZ));

Exercise 1
Create a particle effect that is being emitted by a block. The particle effect should have its own icon, and change color depending on how high up in the world it is. Make sure to update this value as it flies, don't just set the color upon the creation of the particle.

If you don't want to make your own textures you can find two very simple texture below
https://dl.dropboxusercontent.com/u/46486053/example_particle.zip

Exercise 2
Create a block that goes through and plays all 120 note block sounds. Observe that there are only 5 different sound files which are used to play the different sounds.

If you don't want to make the block icon yourself you can grab one texture from the link below https://dl.dropboxusercontent.com/u/46486053/note_sequencer.png

Further explorations 1
Make a world generation that can spawn an igloo in a snow biome.

Further explorations 2
When you're ready for it, there's an assignment waiting. Head over to the assignment page to get it(can be accessed from the course page).

Answers and solutions

Answer to Question 1
If the sound is a sound effect, it should be saved as:
/src/minecraft/assets/test/sound/hey.ogg

If the sound is a recording sound, it should be saved as:
/src/minecraft/assets/test/records/hey.ogg

If the sound is some background music, it should be saved as:
/src/minecraft/assets/test/music/hey.ogg

Answer to Question 2
Given the variables x, y and z the sound can be played like the following. Note that we shouldn't specify the extention when playing the sound.

Minecraft.getMinecraft().sndManager.playSound("test:hey", x, y, z, 1, 0);

The 1 is the volume, this could be lowered to make it quieter. The 0 is the pitch, leaving it at 0 makes the sound play normally, changing it will change the speed it plays at. A handy way of getting some dynamics to the sound.

Answer to Question 3
The following code will spawn a portal particle at the given position and motion. Note that this only works for vanilla particles.

world.spawnParticle("portal", particleX, particleY, particleZ, particleMotionX, particleMotionY,

```
particleMotionZ);
```

Answer to Question 4

To create a particle one must create a class which extends the EntityFX class. To add an icon simply call the unnamed method func_110125_a and use the icon as the first and only parameter. However, one also need to define which type of icon that is used. If the icon is registered by a block one should use layer 1, if it is registered by an item use layer 2. The layer is defined by overriding the getFXLayer method. A very simple particle could look like the following.

```
public class EntityPoisonFX extends EntityFX {

	public EntityPoisonFX(World world, double x, double y, double z, double motionX, double motionY, double motionZ) {
		super(world, x, y, z, motionX, motionY, motionZ);

		func_110125_a(Blocks.poison.particleIcon);
	}

	@Override
	public int getFXLayer() {
		return 1;
	}

}
```

Answer to Question 5

As has been stated in the question, the code works. However it will completely ignore the player's particle setting as well as the player's distance to the particle. A proper code would make sure that the particles are on (and if set on minimal decrease the amount) and make sure the player is close enough before actually spawning the particle.

Possible solution to Exercise 1

```
package example.client.particles;

import net.minecraft.client.particle.EntityFX;
import net.minecraft.world.World;
import example.blocks.Blocks;

public class EntityExampleFX extends EntityFX {

	public EntityExampleFX(World world, double x, double y, double z, double motionX, double motionY, double motionZ) {
		super(world, x, y, z, motionX, motionY, motionZ);

		func_110125_a(Blocks.example.particleIcon);
```

```
                this.motionY = motionY;
                setColor();
        }

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

                setColor();
        }

        private void setColor() {
                //first add red
                particleRed = (float)Math.min(1, posY / 86);

                //when it gets higher up, add some green
                particleGreen = (float)Math.max(0, Math.min(1, (posY - 86) / 86));

                //finally add some blue
                particleBlue = (float)Math.max(0, Math.min(1, (posY - 172) / 86));;
        }

        @Override
        public int getFXLayer() {
                return 1;
        }

}

package example.client.particles;

import net.minecraft.client.Minecraft;
import net.minecraft.client.particle.EntityFX;
import net.minecraft.world.World;

public enum Particles {
        POISON,
        EXAMPLE;

        public void spawnParticle(World world, double x, double y, double z, double motionX,
double motionY, double motionZ) {
                Minecraft mc = Minecraft.getMinecraft();
                if (mc != null && mc.renderViewEntity != null && mc.effectRenderer != null) {
                        int particleSetting = mc.gameSettings.particleSetting;

                        if (particleSetting == 2 || (particleSetting == 1 && world.rand.nextInt(3)
== 0)) {
```

```
                return;
            }

            double distanceX = mc.renderViewEntity.posX - x;
            double distanceY = mc.renderViewEntity.posY - y;
            double distanceZ = mc.renderViewEntity.posZ - z;

            double maxDistance = 16;
            if (distanceX * distanceX + distanceY * distanceY + distanceZ * distanceZ >
maxDistance * maxDistance) {
                return;
            }

            EntityFX particleEffect = null;
            switch(this) {
                case POISON:
                    particleEffect = new EntityPoisonFX(world, x, y, z, motionX,
motionY, motionZ);
                    break;
                case EXAMPLE:
                    particleEffect = new EntityExampleFX(world, x, y, z, motionX,
motionY, motionZ);
                    break;
            }

            if (particleEffect != null) {
                Minecraft.getMinecraft().effectRenderer.addEffect(particleEffect);
            }


        }
    }

}

package example.blocks;

import java.util.Random;

import net.minecraft.block.Block;
import net.minecraft.block.material.Material;
import net.minecraft.client.Minecraft;
import net.minecraft.client.renderer.texture.IconRegister;
import net.minecraft.creativetab.CreativeTabs;
import net.minecraft.entity.player.EntityPlayer;
import net.minecraft.potion.Potion;
import net.minecraft.potion.PotionEffect;
import net.minecraft.util.Icon;
import net.minecraft.world.World;
import cpw.mods.fml.relauncher.Side;
import cpw.mods.fml.relauncher.SideOnly;
import example.client.particles.EntityPoisonFX;
import example.client.particles.Particles;

public class BlockExample extends Block {

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

                setCreativeTab(CreativeTabs.tabBlock);
                setHardness(1F);
                setStepSound(Block.soundStoneFootstep);
                setUnlocalizedName(BlockInfo.EXAMPLE_UNLOCALIZED_NAME);
        }

        @SideOnly(Side.CLIENT)
        public Icon particleIcon;

        @Override
        @SideOnly(Side.CLIENT)
        public void registerIcons(IconRegister register) {
                blockIcon = register.registerIcon(BlockInfo.TEXTURE_LOCATION + ":" +
BlockInfo.EXAMPLE_TEXTURE);
                particleIcon = register.registerIcon(BlockInfo.TEXTURE_LOCATION + ":" +
BlockInfo.EXAMPLE_PARTICLE_TEXTURE);
        }

        @Override
        @SideOnly(Side.CLIENT)
        public void randomDisplayTick(World world, int x, int y, int z, Random rand) {

                float particleX = x + rand.nextFloat();
                float particleY = y + rand.nextFloat();
                float particleZ = z + rand.nextFloat();

                float particleMotionX = -0.5F + rand.nextFloat();
                float particleMotionY =  rand.nextFloat();
                float particleMotionZ = -0.5F + rand.nextFloat();

                Particles.EXAMPLE.spawnParticle(world, particleX, particleY, particleZ,
particleMotionX, particleMotionY, particleMotionZ);

        }
}

package example.blocks;

public class BlockInfo {

    public static final String TEXTURE_LOCATION = "example";

    public static int EXAMPLE_ID;
    public static final String EXAMPLE_KEY = "Example";
    public static final int EXAMPLE_DEFAULT = 2079;

    public static final String EXAMPLE_UNLOCALIZED_NAME = "exampleBlock";
    public static final String EXAMPLE_NAME = "Particle emitter";

    public static final String EXAMPLE_TEXTURE = "example";
    public static final String EXAMPLE_PARTICLE_TEXTURE = "example_particle";

}

package example.blocks;

import net.minecraft.block.Block;
import cpw.mods.fml.common.registry.GameRegistry;
import cpw.mods.fml.common.registry.LanguageRegistry;
public class Blocks {


    public static BlockExample example;

    //called from the mod's pre-init
    public static void init() {
        example = new BlockExample(BlockInfo.EXAMPLE_ID);
        GameRegistry.registerBlock(example, BlockInfo.EXAMPLE_NAME);
    }

    //called from the mod's init
    public static void addNames() {
        LanguageRegistry.addName(example, BlockInfo.EXAMPLE_NAME);
    }



}
```

Possible solution to Exercise 2

```
package example.tileentities;
```

```
import net.minecraft.client.Minecraft;
import net.minecraft.nbt.NBTTagCompound;
import net.minecraft.tileentity.TileEntity;

public class TileEntityNoteSequencer extends TileEntity {

    private int timer;
    private int noteNumber;

    private static String[] instruments = {
        "harp",
        "bd",
        "snare",
        "hat",
        "bassattack"
    };


    @Override
    public void updateEntity() {
        if (!worldObj.isRemote) {
            if (++timer == 5) {
                int instrumentId = noteNumber / 24;
                int notePitchNumber = noteNumber % 24;

                String instrument = "note." + instruments[instrumentId];
                float notePitch = (float)Math.pow(2.0D, (notePitchNumber - 12) /
12D);

                Minecraft.getMinecraft().sndManager.playSound(instrument, xCoord
+ 0.5F, yCoord + 0.5F, zCoord + 0.5F, 1, notePitch);

                noteNumber = (noteNumber + 1) % 120;
                timer = 0;
            }
        }
    }

    @Override
    public void readFromNBT(NBTTagCompound compound) {
        super.readFromNBT(compound);

        timer = compound.getByte("Timer");
        noteNumber = compound.getByte("Note");
    }

    @Override
    public void writeToNBT(NBTTagCompound compound) {
        super.writeToNBT(compound);

        compound.setByte("Timer", (byte)timer);
        compound.setByte("Note", (byte)noteNumber);
    }

}

package example.blocks;

import net.minecraft.block.Block;
import net.minecraft.block.BlockContainer;
import net.minecraft.block.material.Material;
import net.minecraft.client.renderer.texture.IconRegister;
import net.minecraft.creativetab.CreativeTabs;
import net.minecraft.tileentity.TileEntity;
import net.minecraft.world.World;
import cpw.mods.fml.relauncher.Side;
import cpw.mods.fml.relauncher.SideOnly;
import example.tileentities.TileEntityNoteSequencer;

public class BlockNoteSequencer extends BlockContainer {

    protected BlockNoteSequencer(int id) {
        super(id, Material.circuits);

        setStepSound(Block.soundMetalFootstep);
        setHardness(2.5F);
        setUnlocalizedName(BlockInfo.BOMB_UNLOCALIZED_NAME);
        setCreativeTab(CreativeTabs.tabDecorations);
    }

    @Override
    @SideOnly(Side.CLIENT)
    public void registerIcons(IconRegister register) {
        blockIcon = register.registerIcon(BlockInfo.TEXTURE_LOCATION + ":" +
BlockInfo.NOTE_TEXTURE);
    }

    @Override
    public TileEntity createNewTileEntity(World world) {
        return new TileEntityNoteSequencer();
    }

}


package example.blocks;
```

```
public class BlockInfo {

        public static final String TEXTURE_LOCATION = "example";

        public static int NOTE_ID;
        public static final String NOTE_KEY = "Note_Sequencer";
        public static final int NOTE_DEFAULT = 2078;

        public static final String NOTE_UNLOCALIZED_NAME = "noteBlock";
        public static final String NOTE_NAME = "Note sequencer";

        public static final String NOTE_TEXTURE = "note_sequencer";

        public static final String NOTE_TE_KEY = "noteTileEntity";

}

package example.blocks;

import net.minecraft.block.Block;
import cpw.mods.fml.common.registry.GameRegistry;
import cpw.mods.fml.common.registry.LanguageRegistry;
import example.tileentities.TileEntityNoteSequencer;
public class Blocks {


        public static Block note;

        //called from the mod's pre-init
        public static void init() {
                note = new BlockNoteSequencer(BlockInfo.NOTE_ID);
                GameRegistry.registerBlock(note, BlockInfo.NOTE_NAME);
        }

        //called from the mod's init
        public static void addNames() {
                LanguageRegistry.addName(note, BlockInfo.NOTE_NAME);
        }

        //called from the mod's init
        public static void registerTileEntities() {
                GameRegistry.registerTileEntity(TileEntityNoteSequencer.class,
BlockInfo.NOTE_TE_KEY);
        }

}
```