Obsidian API Tutorial #3 – Entities – All versions

<< Previous (http://www.dabigjoe.com/obsidian-suite/tutorials/2-resources/) All tutorials (http://www.dabigjoe.com/obsidian-suite/tutorials/) Next >> (http://www.dabigjoe.com/obsidian-suite/tutorials/3a-entity-shadow-size-scale)

Let's sort out the entities

Our resources are imported, now lets convert our entities so they use The API, allowing them to use our animations. The video for this tutorial can be found here (https://youtu.be/ysEj6RvgMgE).

Prerequisites

You'll need to have imported the Obsidian Model file and the texture file as per the previous tutorial.

The Entity Class

So most entities have three classes – an entity class, a model class and a render class. We'll take each of these in turn, and I'll show you what you need to change to convert a normal modded entity to an Obsidian Suite entity.

We'll start with the Entity class, in my example (https://github.com/DaBigJoe/ObsidianAPITutorial) it's the EntitySaiga class. All entities will extend some type of entity super class, for example EntityCreature, but it could be any entity class depending on what you're doing. We need a little extra stuff to adapt the entity class for use with the API. All we need to do is make the entity class implement IEntityAnimated:

public class EntitySaiga extends EntityCreature implements IEntityAnimated
After you import IEntityAnimated, Eclipse will complain the entity class is missing a method, giving a red line under the class name and asking you to "Add unimplemented method". Just click "Add unimplemented method" to generate the required isMoving method.

The isMoving method returns a Boolean value representing whether the entity is moving (walking, running, swimming etc. – anything but standing still). It's used to determine which animation should be playing. For simple classes, it's enough to use an existing entity variable called limbSwingAmount; it will be approximately zero when the entity is still. As it never quite reaches zero, we'll check if it's greater than a value close to zero:

```
@Override
public boolean isMoving() {
    return limbSwingAmount > 0.02F;
}
```

The entity class is now integrated with the API, but you might have to come back here in the future to add extra stuff like entity logic and AI.

Moving on to the Model Class

Most model classes will extend ModelBase, but when using the API we want to extend something different. Substitute ModelBase for ModelAnimated:

public class ModelSaiga extends ModelAnimated
Eclipse will say that you need to add a constructor – it should auto complete that for you – giving something like this:

```
public ModelSaiga(String (http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+string) entityName, Wav
    super(entityName, wavefrontObj, textureLocation);
}
```

At this point, you might notice an error coming from the ClientProxy; don't worry we'll fix it in a second! That's everything required for the Model class; wasn't that easy?

And next, the Render Class

In a similar way to the model class, we need to change the parent class – RenderLiving (or equivalent) swaps out for RenderAnimated:

public class RenderSaiga extends RenderAnimated
We can clean everything out from this class, all we need is a constructor:

```
public RenderSaiga(ModelSaiga saiga) {
    super(saiga);
}
```

That's it! Much neater than before.

Finally, let's fix the ClientProxy

As I mentioned earlier, there's now an error in the ClientProxy, so let's go and fix it.

We changed the constructor for our model class, which was used by the ClientProxy. Rather than trying to create an instance of the model, let's get the API to do it for us. Currently, your ClientProxy will have a line like this, with `new ModelSaiga()` causing the error:

RenderSaiga saigaRenderer = new RenderSaiga(new ModelSaiga());
Delete `new ModelSaiga()`, and substitute it for the API FileLoader (obsidianAPI.file.importer.FileLoader):

```
new RenderSaiga(FileLoader.loadModelFromResources("saiga", saigaModel, saigaTexture, ModelSaiga.class));
```

This is where we use the ResourceLocations defined in the previous tutorial. FileLoader.loadModelFromResources expects four arguments: the name for the entity, the model ResourceLocation, the texture ResourceLocation, and the model class. So that looks a bit different, but it's nice and easy to use.

You can actually run the game now!

Now that the ClientProxy has been fixed, and you've changed all the entities, you should be able to run the game and see your models (if you've added spawn eggs). The models won't animate at all, but you'll be able to see them.

That's everything for this tutorial, next time we'll look at applying some animations to our new model!

<< Previous (http://www.dabigjoe.com/obsidian-suite/tutorials/2-resources/) All tutorials (http://www.dabigjoe.com/obsidian-suite/tutorials/) Next >> (http://www.dabigjoe.com/obsidian-suite/tutorials/3a-entity-shadow-size-scale)

Edited files

EntitySaiga

```
public class EntitySaiga extends EntityCreature implements IEntityAnimated {
    public EntitySaiga(World world) {
        super(world);
        this.tasks.taskEntries.clear();
        this.tasks.addTask(0, new EntityAIWander(this, 1.0D));
    }

    @Override
    protected boolean isAIEnabled() {
        return true;
    }

    @Override
    protected void applyEntityAttributes() {
        super.applyEntityAttributes();
        this.getEntityAttribute(SharedMonsterAttributes.maxHealth).setBaseValue(10.0D);
        this.getEntityAttribute(SharedMonsterAttributes.movementSpeed).setBaseValue(0.18D);
    }

    @Override
    public boolean isMoving() {
        return limbSwingAmount > 0.02F;
    }
}
```

ModelSaiga

```
public class ModelSaiga extends ModelAnimated {
    public ModelSaiga(String (http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+string) entityName, W
        super(entityName, wavefrontObj, textureLocation);
    }
}
```

RenderSaiga

```
public class RenderSaiga extends RenderAnimated {
    public RenderSaiga(ModelSaiga saiga) {
        super(saiga);
    }
}
```

ClientProxy

```
public class ClientProxy extends CommonProxy {
    private ResourceLocation saigaModel = new ResourceLocation("mod_api_tutorial:models/Saiga.obm");
    private ResourceLocation saigaTexture = new ResourceLocation("mod_api_tutorial:models/Saiga.png");

    public void registerRendering() {
        RenderSaiga saigaRenderer = new RenderSaiga(FileLoader.loadModelFromResources("saiga", saigaModel, saigaTexture, ModelSaiga.clas
        RenderingRegistry.registerEntityRenderingHandler(EntitySaiga.class, saigaRenderer);
    }
}
```


Edit (http://www.dabigjoe.com/wp-admin/post.php?post=509&action=edit)

LEAVE A REPLY

Logged in as dabigjoe (http://www.dabigjoe.com/wp-admin/profile.php). Log out? (http://www.dabigjoe.com/wp-login.php?action=logout&redirect_to=http%3A%2F%2Fwww.dabigjoe.com%2Fobsidian-suite%2Ftutorials%2F3-entities%2F&_wpnonce=6e8ffaa668)

Comment

POST COMMENT