

Image-Based Height Maps (*RAGE tessellation*)

```

...
public class myGame extends VariableFrameRateGame
{
    ...

    protected void setupScene(Engine eng, SceneManager sm)
        throws IOException
    {
        im = new GenericInputManager();

        // make dolphin avatar
        ...

        // set up lights
        ...

        // set up camera controller
        ...

        // 2^patches: min=5, def=7, warnings start at 10
        Tessellation tessE = sm.createTessellation("tessE", 6);

        // subdivisions per patch: min=0, try up to 32
        tessE.setSubdivisions(8f);

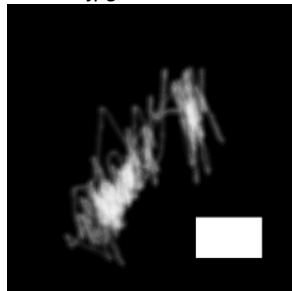
        SceneNode tessN =
            sm.getRootSceneNode().
            createChildSceneNode("TessN");
        tessN.attachObject(tessE);

        // to move it, note that X and Z must BOTH be positive OR negative
        // tessN.translate(Vector3f.createFrom(-6.2f, -2.2f, 2.7f));
        // tessN.yaw(Degreef.createFrom(37.2f));

        tessN.scale(10, 20, 10);
        tessE.setHeightMap(this.getEngine(), "scribble.jpg");
        tessE.setTexture(this.getEngine(), "grass.jpg");
        // tessE.setNormalMap(. . .)
    }
}

```

scribble.jpg:



## Adjusting avatar position in Height Maps

```

...
public class myGame extends VariableFrameRateGame
{
    ...

    protected void updateVerticalPosition()
    {
        SceneNode dolphinN =
            this.getEngine().getSceneManager().
            getSceneNode("dolphinNode");
        SceneNode tessN =
            this.getEngine().getSceneManager().
            getSceneNode("tessN");
        Tessellation tessE = ((Tessellation) tessN.getAttachedObject("tessE"));

        // Figure out Avatar's position relative to plane
        Vector3 worldAvatarPosition = dolphinN.getWorldPosition();
        Vector3 localAvatarPosition = dolphinN.getLocalPosition();

        // use avatar World coordinates to get coordinates for height
        Vector3 newAvatarPosition = Vector3f.createFrom(

            // Keep the X coordinate
            localAvatarPosition.x(),

            // The Y coordinate is the varying height
            tessE.getWorldHeight(
                worldAvatarPosition.x(),
                worldAvatarPosition.z()),

            // Keep the Z coordinate
            localAvatarPosition.z()
        );

        // use avatar Local coordinates to set position, including height
        dolphinN.setLocalPosition(newAvatarPosition);
    }
}

```

## MoveForwardAction.java:

```

...
public class MoveForwardAction extends AbstractInputAction
{
    private Node avN;
    private MyGame myGame;

    public MoveForwardAction(Node n, MyGame g)
    {
        avN = n;
        myGame = g;
    }

    public void performAction(float time, Event e)
    {
        avN.moveForward(0.01f);
        myGame.updateVerticalPosition();
    }
}

```