Introducing jMonkey Engine

Björn Martin, bjoern.martin@gmx.net

Agenda

An introduction to engine and SDK

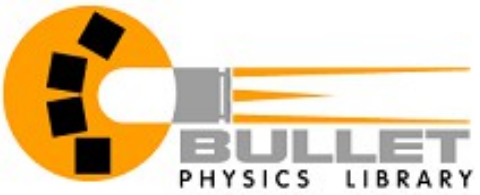
Implementing a sample project using jMonkey's core features

What else there is

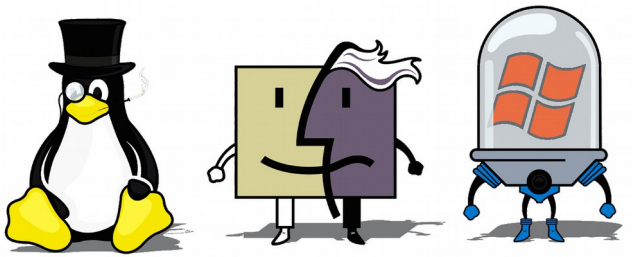
„jMonkeyEngine is a 3D game engine
for adventurous Java developers.
It's open source, cross platform and cutting edge.
And it is all beautifully documented."

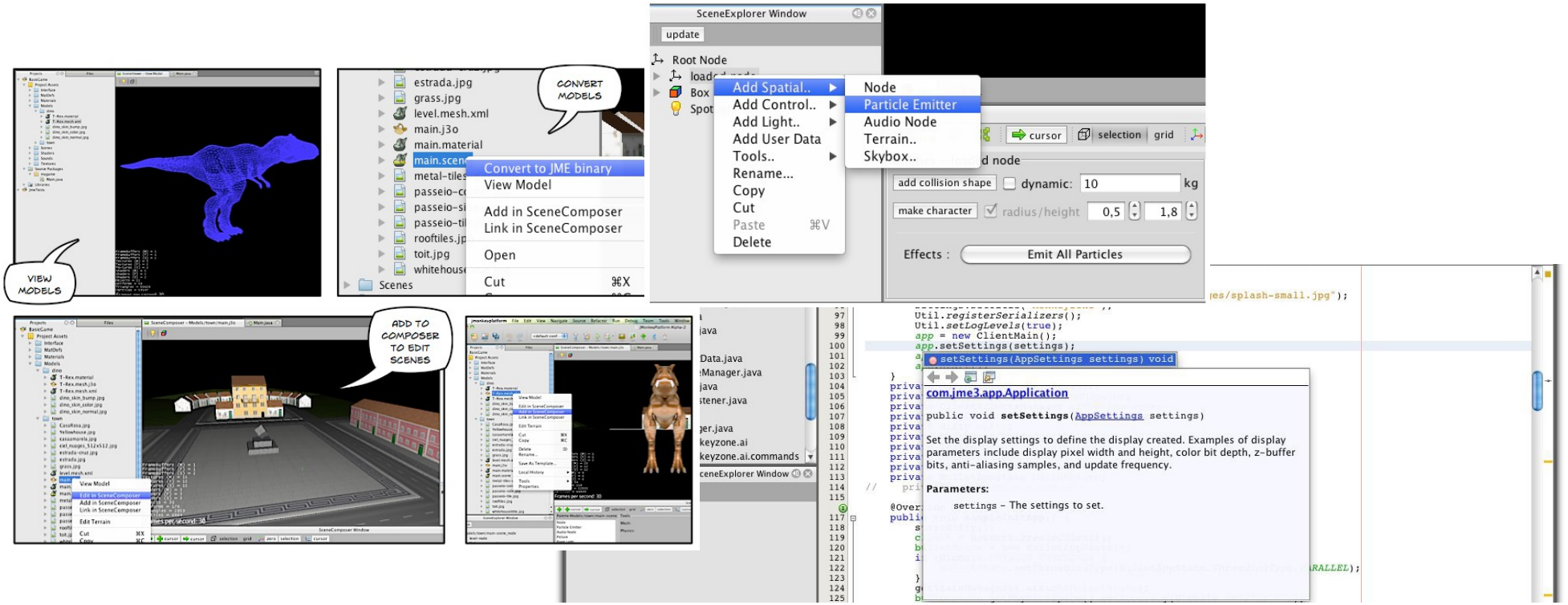
-- from the home page

jMonkey Engine

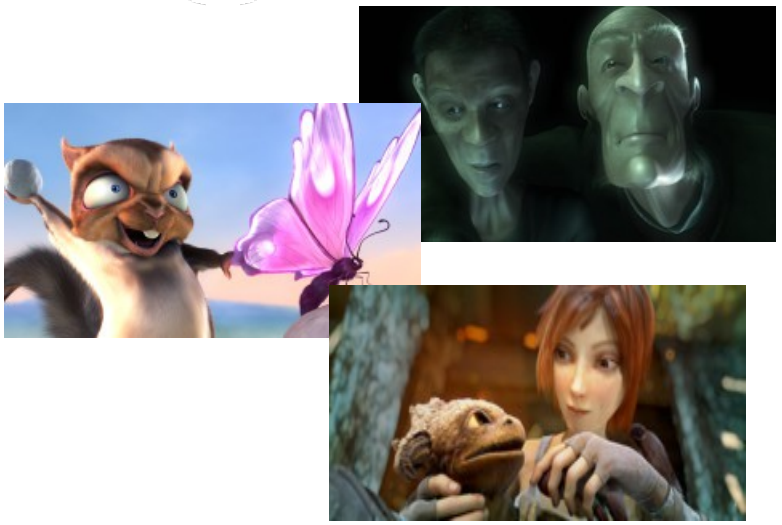


LWJGL

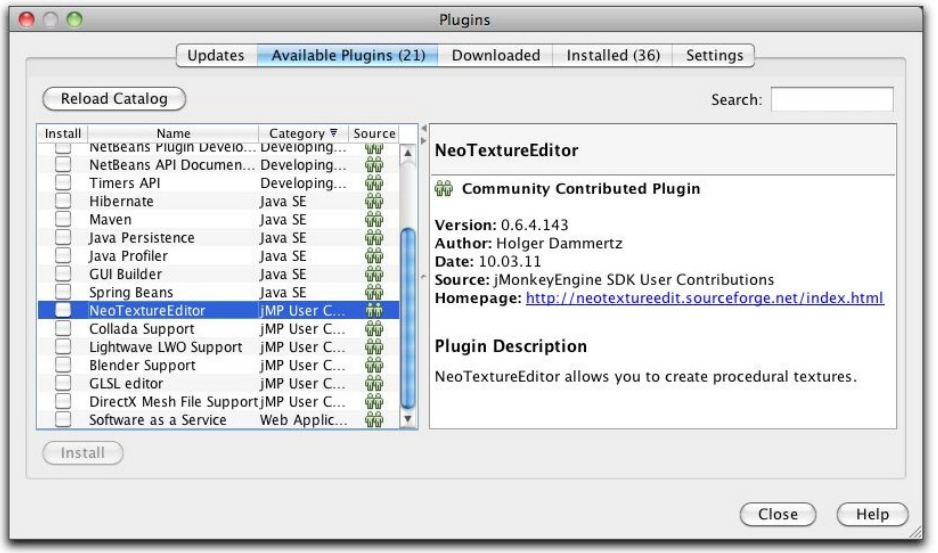




source: http://hub.jmonkeyengine.org/



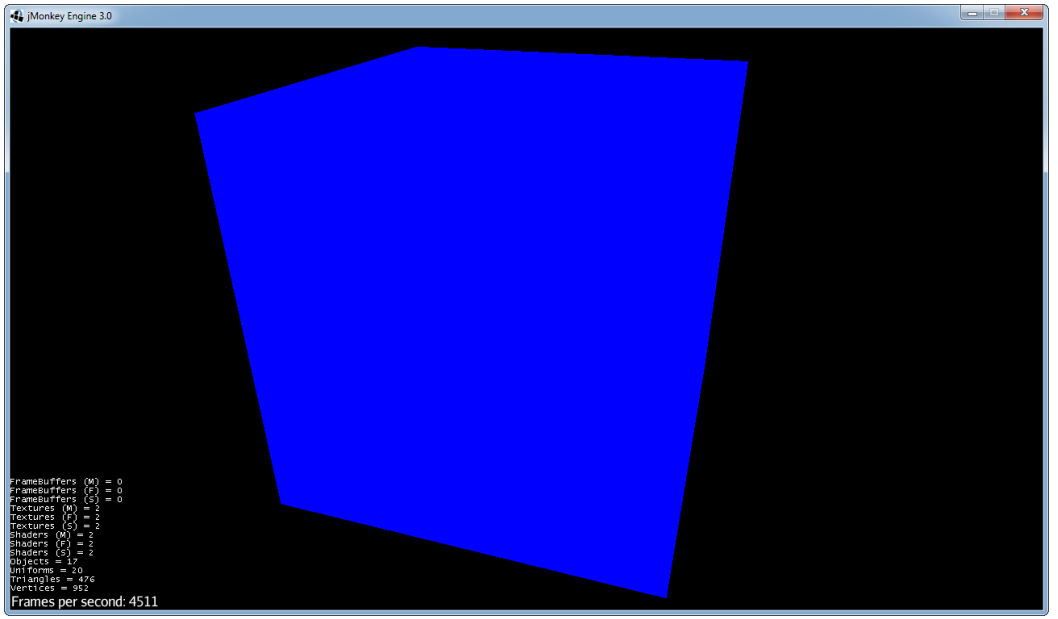
source: http://www.blender.org/features/projects/



source: http://hub.jmonkeyengine.org/

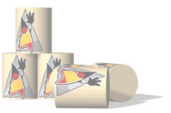
Bootstrapping

- Download SDK (Linux, Mac, Win) from http://hub.jmonkeyengine.org/downloads/
- Install and open it
- File > New Project > JME3 > BasicGame
- Open mygame.Main and hit Shift-F6
- Done!
- rootNode holds box with blue material

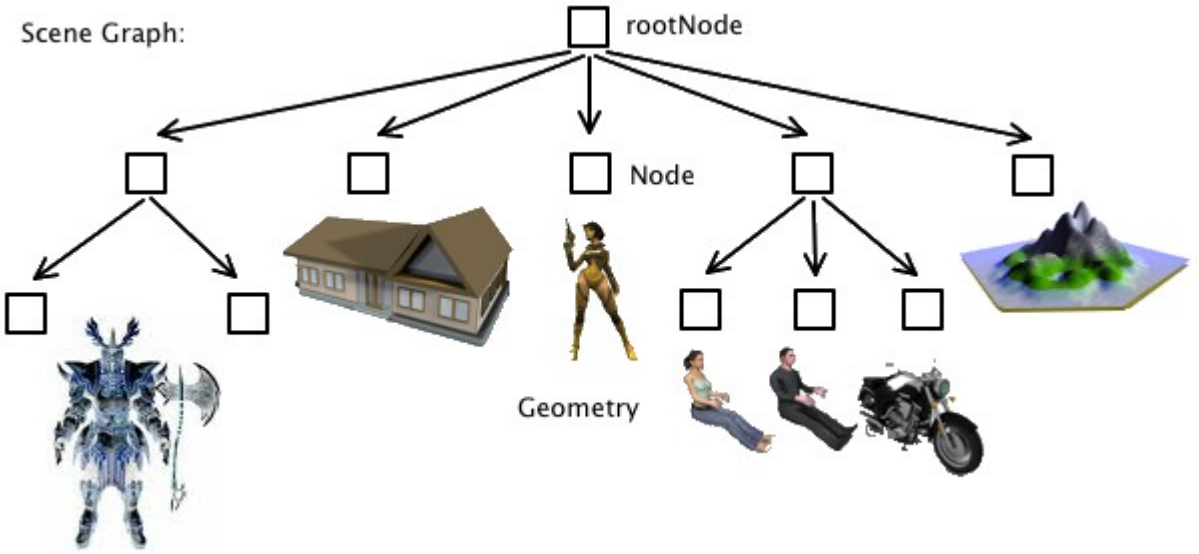


Bootstrapping

```
public class Main extends SimpleApplication {

    public static void main(String[] args) {
        Main app = new Main();
        app.start();
    }

    @Override public void simpleInitApp() {
        Box b = new Box(1, 1, 1);
        Geometry geom = new Geometry("Box", b);
        Material mat = new Material(assetManager, "Common/MatDefs/Misc/Unshaded.j3md");
        mat.setColor("Color", ColorRGBA.Blue);
        geom.setMaterial(mat);
        rootNode.attachChild(geom);
    }

    @Override public void simpleUpdate(float tpf) {
        //TODO: add update code
    }

    @Override public void simpleRender(RenderManager rm) {
        //TODO: add render code
    }
}
```

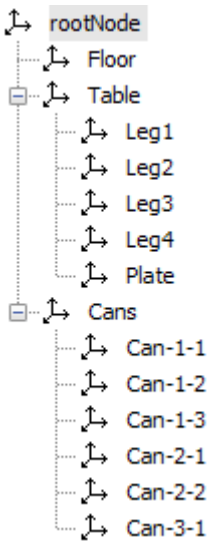


Spatials



source: http://hub.jmonkeyengine.org

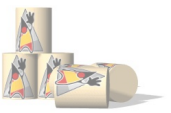
Spatials




```
Material mat = new Material(assetManager, "Common/MatDefs/Misc/Unshaded.j3md");
mat.setColor("Color", ColorRGBA.Gray);

Box bFloor = new Box(10.0f, 0.1f, 10.0f);
Geometry floor = new Geometry("Floor", bFloor);
floor.setMaterial(mat);
floor.setLocalTranslation(0.0f, -0.1f, 0.0f);
rootNode.attachChild(floor);

Node table = new Node("Table");

Box bLeg = new Box(0.1f, 1.0f, 0.1f);
Geometry leg1 = new Geometry("Leg1", bLeg);
leg1.setMaterial(mat);
leg1.setLocalTranslation(-3.9f, 0.0f, -1.9f);
table.attachChild(leg1);

Box bPlate = new Box(4.0f, 0.1f, 2.0f);
Geometry plate = new Geometry("Plate", bPlate);
plate.setMaterial(mat);
plate.setLocalTranslation(0.0f, 1.1f, 0.0f);
table.attachChild(plate);

table.setLocalTranslation(0.0f, 1.0f, 0.0f);
rootNode.attachChild(table);
```

Spatials

```
Node cans = new Node("Cans");
Cylinder cCan = new Cylinder(4, 16, canRadius, canHeight, true);

for (int row = 0; row < towerSize; row++) {
    for (int column = 0; column < towerSize - row; column++) {
        Material mat = new Material(assetManager, "Common/MatDefs/Misc/Unshaded.j3md");
        mat.setColor("Color", ColorRGBA.randomColor());

        Geometry can = new Geometry("Can-" + column + "-" + row, cCan);
        can.setMaterial(mat);
        can.rotate(-90.0f * FastMath.DEG_TO_RAD, 0.0f, 0.0f);
                        space between cans          shift to left from center          shift to right for upper rows
        float xShift = (canRadius * 2.5f) * (-(towerSize - 1) / 2.0f + column) + (row * canRadius * 1.25f);
        float yShift = row * canHeight;
        can.setLocalTranslation(0.0f + xShift, 0.25f + yShift, 0.0f);
        cans.attachChild(can);
    }
}

cans.setLocalTranslation(0.0f, 2.2f, 0.0f);
rootNode.attachChild(cans);
```

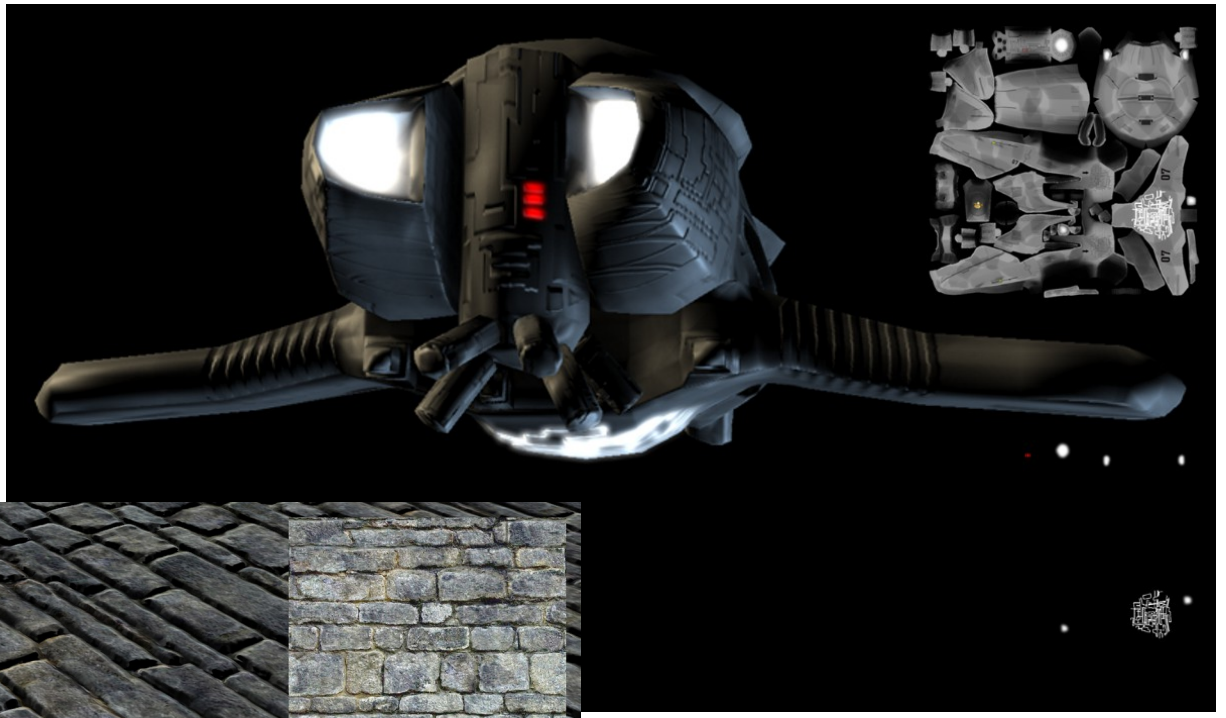


Spatials

Demo

Material

source: SDK tests

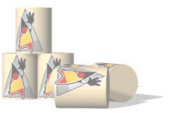


source: http://hub.jmonkeyengine.org/

source: SDK test data

Material

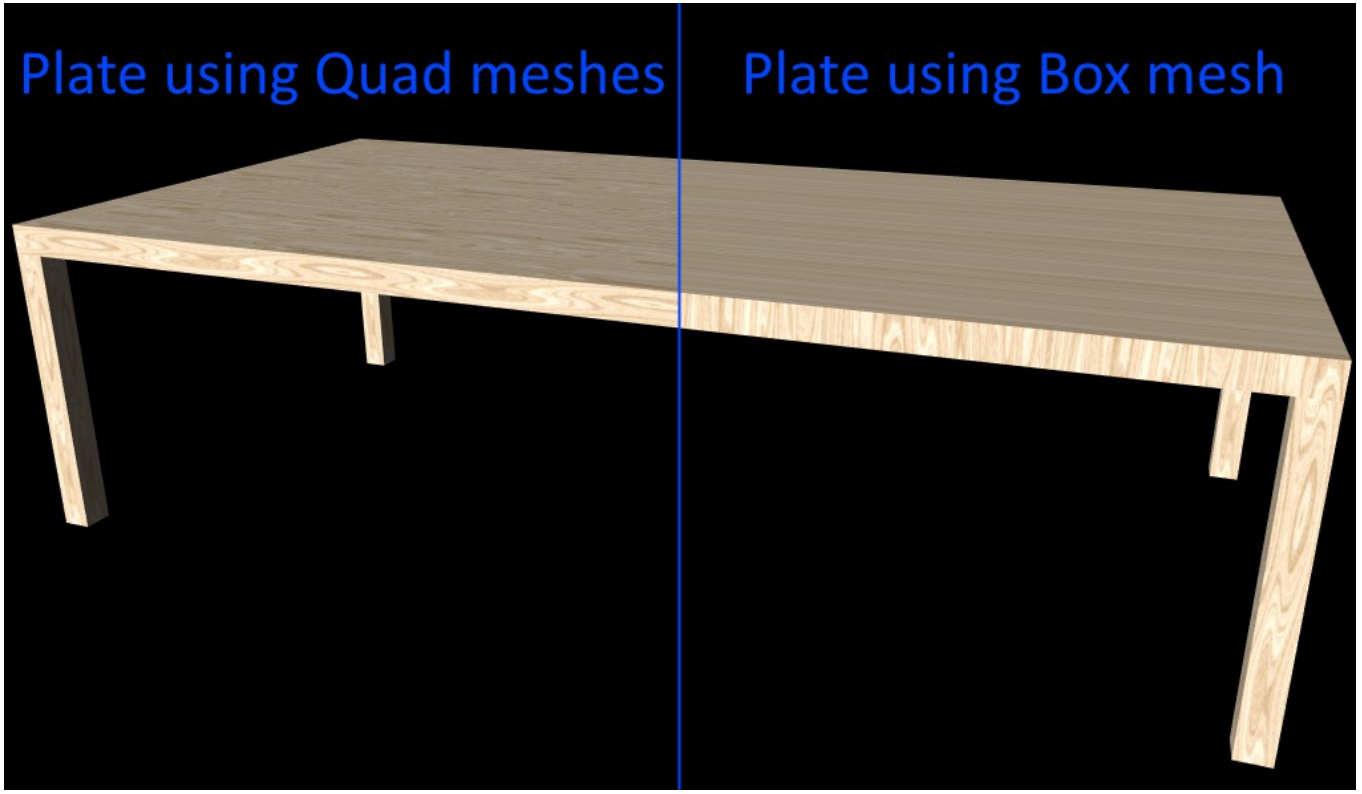
```
Material mat = new Material(assetManager, "Common/MatDefs/Light/Lighting.j3md");
mat.setTexture("DiffuseMap", assetManager.loadTexture("Textures/Terrain/splat/road.jpg"));
mat.getTextureParam("DiffuseMap").getTextureValue().setWrap(WrapMode.Repeat);
mat.setTexture("NormalMap", assetManager.loadTexture("Textures/Terrain/splat/road_normal.png"));
mat.getTextureParam("NormalMap").getTextureValue().setWrap(WrapMode.Repeat);
mat.setBoolean("UseMaterialColors", true);
mat.setColor("Diffuse", ColorRGBA.White);

Box bFloor = new Box(10.0f, 0.1f, 10.0f);
bFloor.scaleTextureCoordinates(new Vector2f(10.0f, 10.0f));
Geometry floor = new Geometry("Floor", bFloor);
floor.setMaterial(mat);
floor.setLocalTranslation(0.0f, -0.1f, 0.0f);
floor.setShadowMode(ShadowMode.Receive);
rootNode.attachChild(floor);



sun = new DirectionalLight();
sun.setDirection(new Vector3f(-0.5f, -0.5f, 1.0f).normalizeLocal());
sun.setColor(ColorRGBA.White);
rootNode.addLight(sun);

env = new AmbientLight();
env.setColor(ColorRGBA.White);
rootNode.addLight(env);
envy = true;
```



Material

```
Quad qPlate = new Quad(4.0f, 8.0f);
qPlate.scaleTextureCoordinates(new Vector2f(2.0f, 4.0f));
Quad qPlateLongEdge = new Quad(0.2f, 8.0f);
qPlateLongEdge.scaleTextureCoordinates(new Vector2f(0.1f, 4.0f));
Quad qPlateShortEdge = new Quad(0.2f, 4.0f);
qPlateShortEdge.scaleTextureCoordinates(new Vector2f(0.1f, 2.0f));
```

```
Box bPlate =
        new Box(4.0f, 0.1f, 2.0f);
bPlate.scaleTextureCoordinates(
        new Vector2f(4.0f, 0.1f));
```



texture source: http://opengameart.org/content/5-wood-textures

Light & Shadow

```
sun = new DirectionalLight();
sun.setDirection(new Vector3f(-0.5f, -0.5f, 1.0f).normalizeLocal());
sun.setColor(ColorRGBA.White);
rootNode.addLight(sun);

env = new AmbientLight();
env.setColor(ColorRGBA.White);
rootNode.addLight(env);
envy = true;
```

```
DirectionalLightShadowRenderer dlsr = new DirectionalLightShadowRenderer(assetManager, SHADOWMAP_SIZE, 3);
dlsr.setLight(sun);
dlsr.setLambda(0.55f);
dlsr.setShadowIntensity(0.4f);
dlsr.setEdgeFilteringMode(EdgeFilteringMode.PCFPOISSON);
viewPort.addProcessor(dlsr);
```

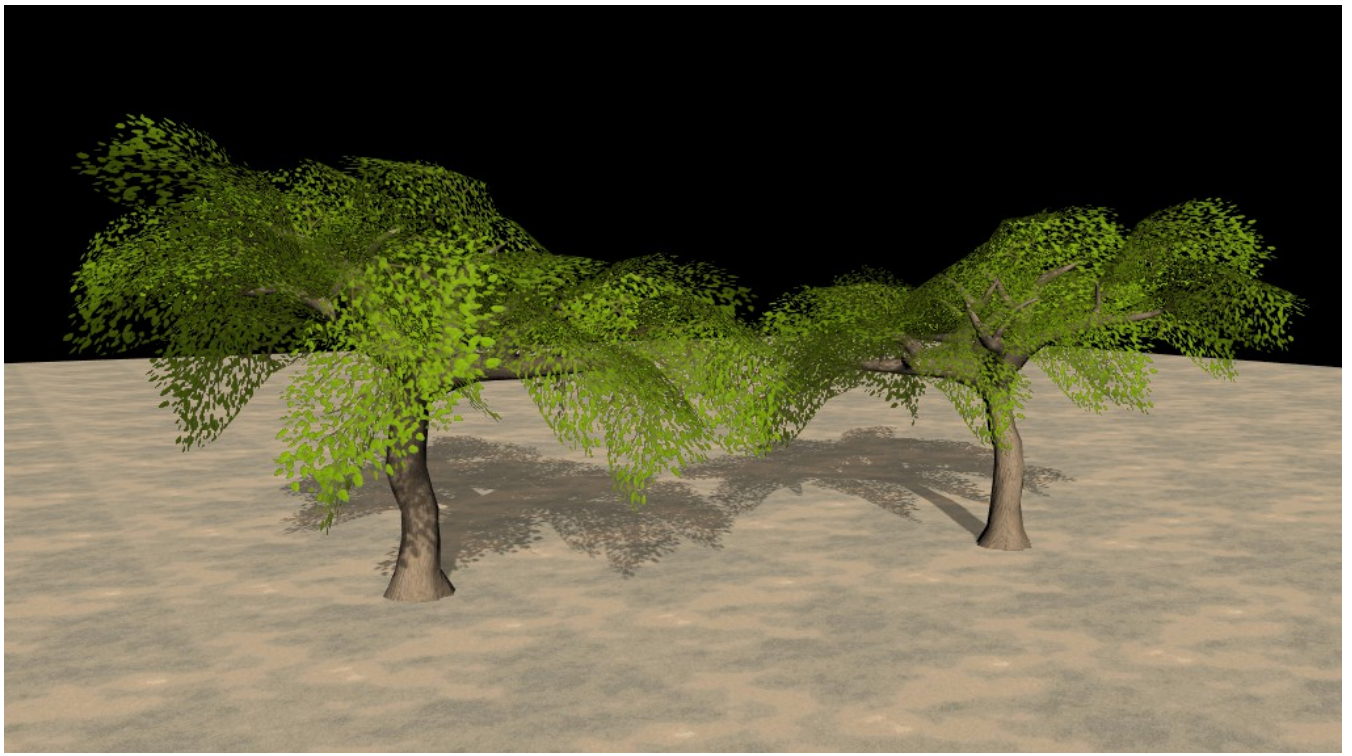
```
Material mat = new Material(assetManager, "Common/MatDefs/Light/Lighting.j3md");
mat.setTexture("DiffuseMap", assetManager.loadTexture("Textures/Can/Can_square.png"));
mat.setBoolean("UseMaterialColors", true);
mat.setColor("Diffuse", ColorRGBA.White);
mat.setColor("Ambient", ColorRGBA.DarkGray);
can.setMaterial(mat);
can.setShadowMode(ShadowMode.CastAndReceive);
```


Material, Light & Shadow

Demo

Assets

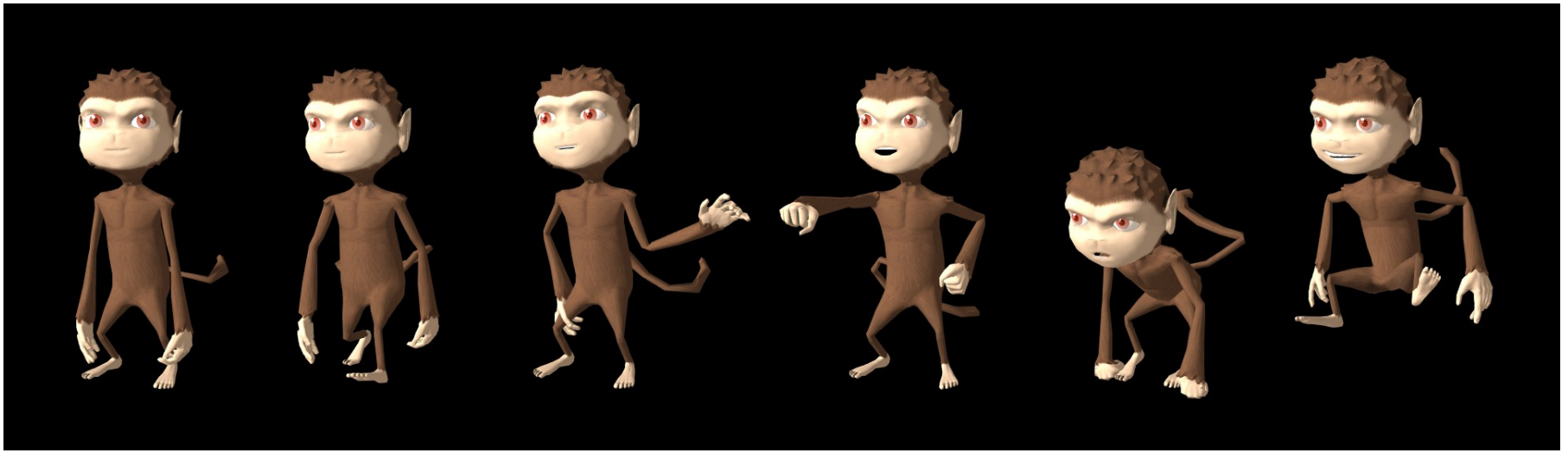
```
Spatial tree = assetManager.loadModel("Models/Tree/Tree.mesh.j3o");
tree.scale(2.0f);
tree.setLocalTranslation(7.0f, 0.0f, 7.0f);
tree.rotate(0.0f, DEG_90, 0.0f);
tree.setShadowMode(ShadowMode.CastAndReceive);
rootNode.attachChild(tree);
```



source: SDK test data

Animation

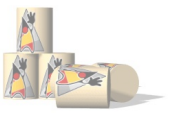
```
Spatial jaime = assetManager.loadModel("Models/Jaime/Jaime.j3o");
jaime.rotate(0.0f, DEG_180, 0.0f);
jaime.setLocalTranslation(3.0f, 2.2f, 0.0f);
jaime.setShadowMode(ShadowMode.Cast);
rootNode.attachChild(jaime);

AnimControl control = jaime.getControl(AnimControl.class);
control.addListener(this);
AnimChannel channel = control.createChannel();
channel.setAnim("Idle", 1.0f);
channel.setLoopMode(LoopMode.Loop);
```



source: SDK test data

Update Loop

```
public void simpleUpdate(float tpf) {
    flashLight.setPosition(cam.getLocation());
    flashLight.setDirection(cam.getDirection());
}
```



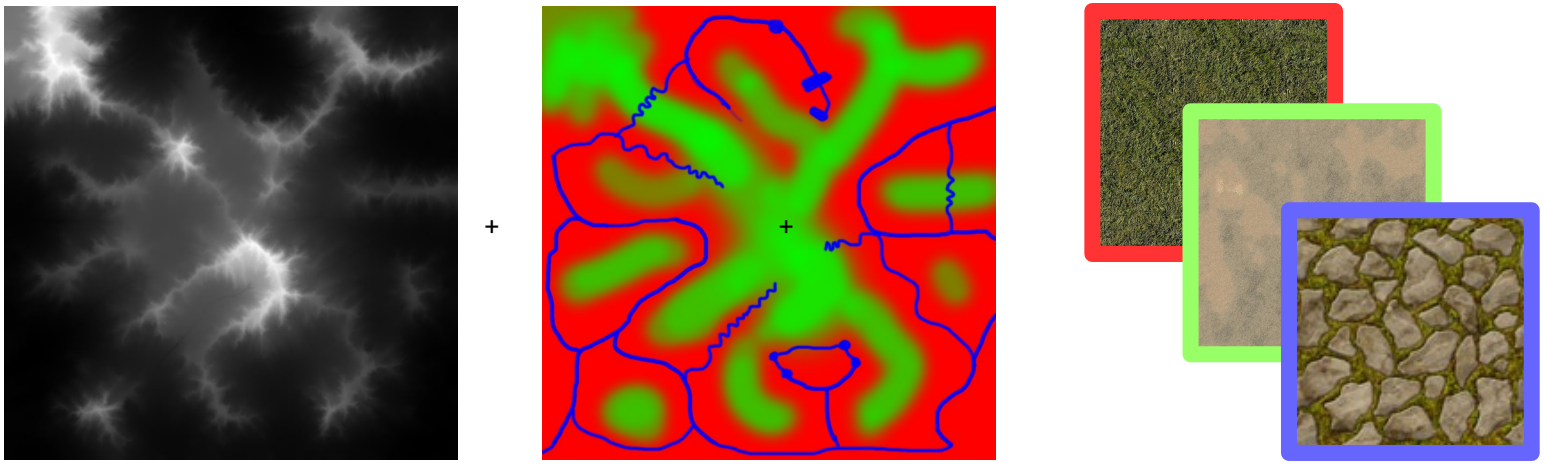
Assets, Animation & Update Loop

Demo

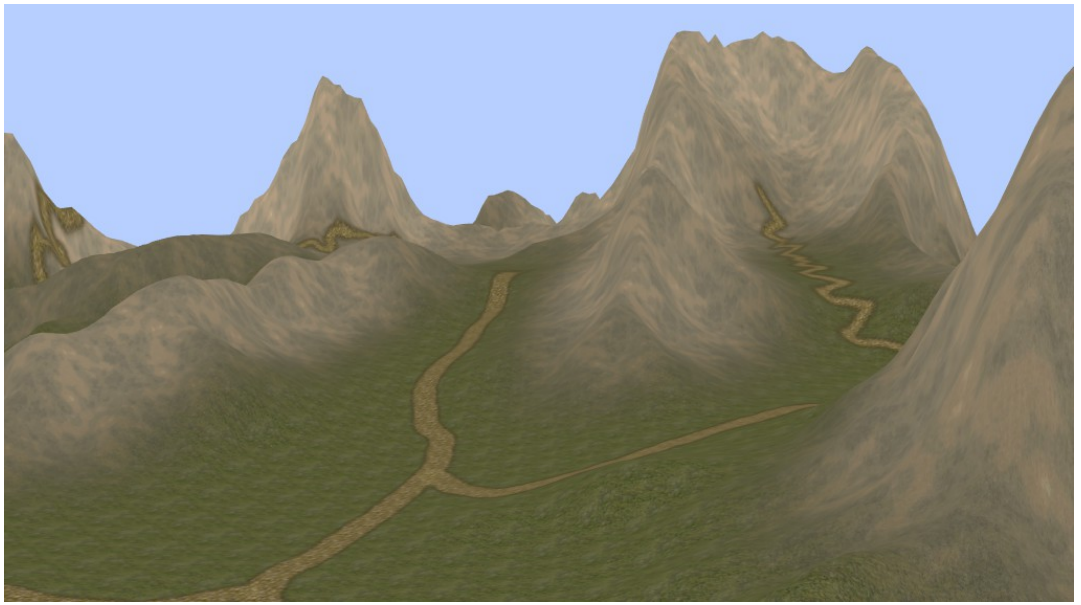
Terrain

```
mat_terrain = new Material(assetManager, "Common/MatDefs/Terrain/Terrain.j3md");
mat_terrain.setTexture("Alpha", assetManager.loadTexture("Textures/Terrain/splat/alphamap.png"));

Texture grass = assetManager.loadTexture("Textures/Terrain/splat/grass.jpg");
grass.setWrap(WrapMode.Repeat);
mat_terrain.setTexture("Tex1", grass);
mat_terrain.setFloat("Tex1Scale", 64f);
Texture dirt = assetManager.loadTexture("Textures/Terrain/splat/dirt.jpg");
dirt.setWrap(WrapMode.Repeat);
mat_terrain.setTexture("Tex2", dirt);
mat_terrain.setFloat("Tex2Scale", 32f);
Texture rock = assetManager.loadTexture("Textures/Terrain/splat/road.jpg");
rock.setWrap(WrapMode.Repeat);
mat_terrain.setTexture("Tex3", rock);
mat_terrain.setFloat("Tex3Scale", 128f);

AbstractHeightMap heightmap = new ImageBasedHeightMap(
        assetManager.loadTexture("Textures/Terrain/splat/mountains512.png").getImage());
heightmap.load();
heightmap.smooth(0.9f, 3);

int patchSize = 65;
terrain = new TerrainQuad("my terrain", patchSize, 513, heightmap.getHeightMap());
terrain.setMaterial(mat_terrain);
terrain.setLocalTranslation(0, -100, 0);
terrain.setLocalScale(2f, 1f, 2f);
rootNode.attachChild(terrain);
```


Terrain



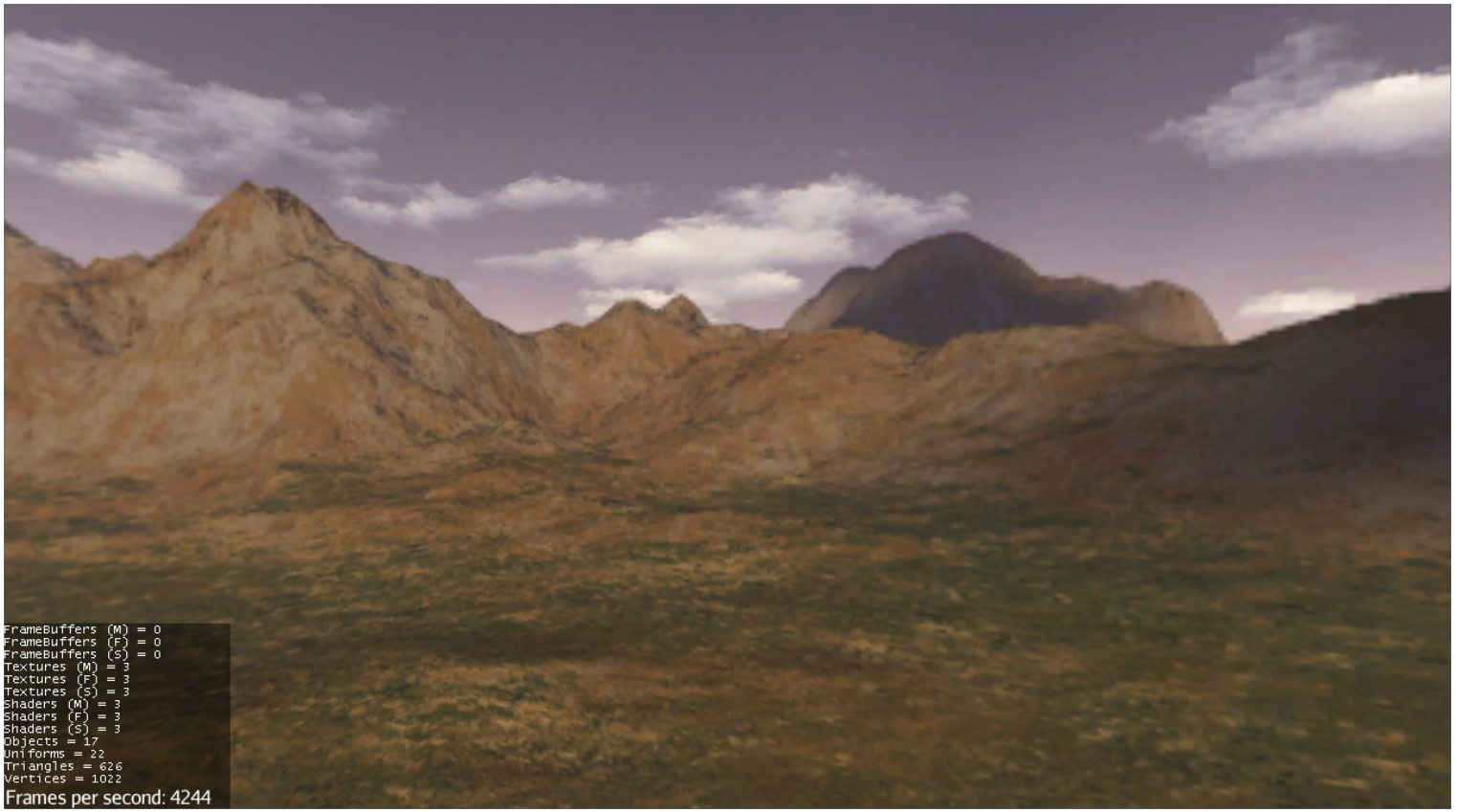
=



source: http://hub.jmonkeyengine.org/

Sky

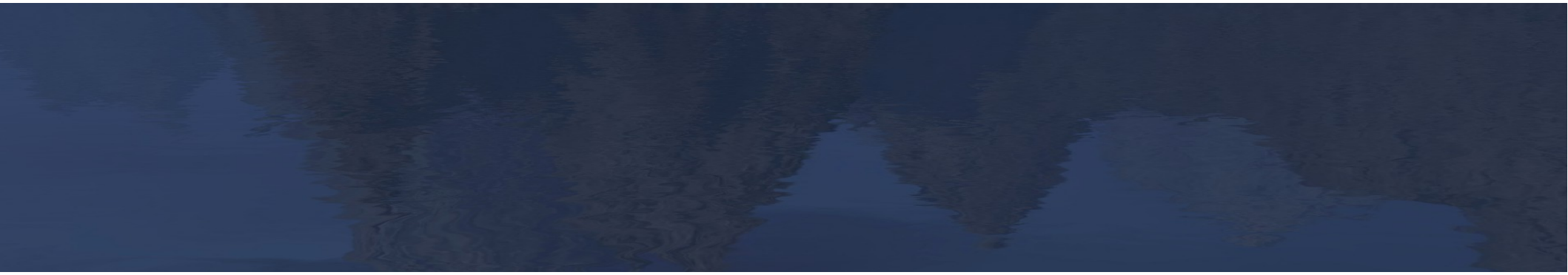
```
Spatial sky = SkyFactory.createSky(
    assetManager, "Textures/Sky/Bright/BrightSky.dds", false);
rootNode.attachChild(sky);
```



source: SDK test data

Water

```
SimpleWaterProcessor waterProcessor = new SimpleWaterProcessor(assetManager);
waterProcessor.setReflectionScene(mainScene);
waterProcessor.setWaterDepth(40);
waterProcessor.setDistortionScale(0.1f);
waterProcessor.setWaveSpeed(0.02f);
waterProcessor.setPlane(new Plane(Vector3f.UNIT_Y, -1.5f));
viewPort.addProcessor(waterProcessor);

Quad quad = new Quad(1024, 1024);
quad.scaleTextureCoordinates(new Vector2f(16f, 16f));
Geometry water = new Geometry("water", quad);
water.rotate(-DEG_90, 0, 0);
water.setLocalTranslation(-512.0f, -1.5f, 512.0f);
water.setShadowMode(ShadowMode.Receive);
water.setMaterial(waterProcessor.getMaterial());

rootNode.attachChild(water);
```



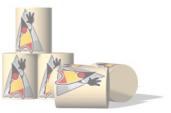
source: SDK test data

Terrain, Sky & Water

Demo

Input

```
inputManager.addMapping("Shoot",
        new KeyTrigger(KeyInput.KEY_SPACE),
        new MouseButtonTrigger(MouseInput.BUTTON_LEFT));
inputManager.addMapping("Reset", new KeyTrigger(KeyInput.KEY_R));
inputManager.addMapping("Cans+", new KeyTrigger(KeyInput.KEY_ADD));
inputManager.addMapping("Cans-", new KeyTrigger(KeyInput.KEY_SUBTRACT));
inputManager.addListener(this, "Shoot", "Reset", "Cans+", "Cans-");


public void onAction(String name, boolean isPressed, float tpf) {
    if (name.equals("Shoot") && !isPressed) {
        ...
    }
}
```



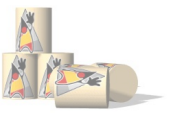
Picking

```
CollisionResults results = new CollisionResults();
Vector2f click2d = new Vector2f(
        settings.getWidth() / 2,
        settings.getHeight() / 2);
Vector3f click = cam.getWorldCoordinates(
        new Vector2f(click2d.x, click2d.y), 0f);
Vector3f direction = cam.getWorldCoordinates(
        new Vector2f(click2d.x, click2d.y), 1f)
        .subtractLocal(click).normalizeLocal();
Ray ray = new Ray(click, direction);
cans.collideWith(ray, results);
if (results.size() > 0) {
    cans.detachChild(results.getClosestCollision().getGeometry());
}
```

Interface

```
inventory = new Node("Inventory");
inventory.setLocalTranslation(40, settings.getHeight() - 40, 0.0f);
guiNode.attachChild(inventory);

DirectionalLight light = new DirectionalLight();
light.setDirection(new Vector3f(0, 0, -1.0f).normalizeLocal());
light.setColor(ColorRGBA.White);
guiNode.addLight(light);
```

```
Geometry can = results.getClosestCollision().getGeometry();
cans.detachChild(can);
can.setLocalScale(100);
can.rotate(-0.2f, 0.1f, DEG_180 - 0.2f);
can.setLocalTranslation(50.0f * inventory.getChildren().size(), 0.0f, 0.0f);
inventory.attachChild(can);
```



Input, Picking & Interface

Demo

Physics


```
bulletAppState = new BulletAppState();
stateManager.attach(bulletAppState);
```

```
RigidBodyControl canPhysics = new RigidBodyControl(1.0f);
can.addControl(canPhysics);
bulletAppState.getPhysicsSpace().add(canPhysics);
```

```
RigidBodyControl tablePhysics = new RigidBodyControl(0.0f);
table.addControl(tablePhysics);
bulletAppState.getPhysicsSpace().add(tablePhysics);
```

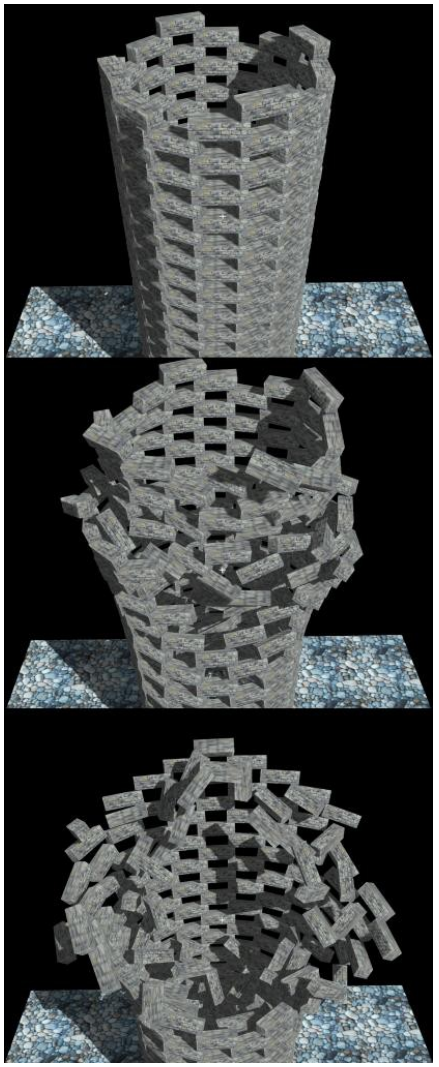

Physics

```
Material mat = new Material(assetManager, "Common/MatDefs/Misc/Unshaded.j3md");
TextureKey texKey = new TextureKey("Textures/Can/Can_square.png");
texKey.setGenerateMips(true);
Texture tex = assetManager.loadTexture(texKey);
mat.setTexture("ColorMap", tex);

Sphere sBall = new Sphere(16, 16, 0.2f, true, false);
sBall.setTextureMode(Sphere.TextureMode.Projected);

Geometry ball = new Geometry("Ball", sBall);
ball.setMaterial(mat);
ball.setLocalTranslation(cam.getLocation().add(cam.getDirection().normalize()));
balls.attachChild(ball);

RigidBodyControl ballPhysics = new RigidBodyControl(4.0f);
ball.addControl(ballPhysics);
bulletAppState.getPhysicsSpace().add(ballPhysics);

ballPhysics.setLinearVelocity(cam.getDirection().mult(25));
```

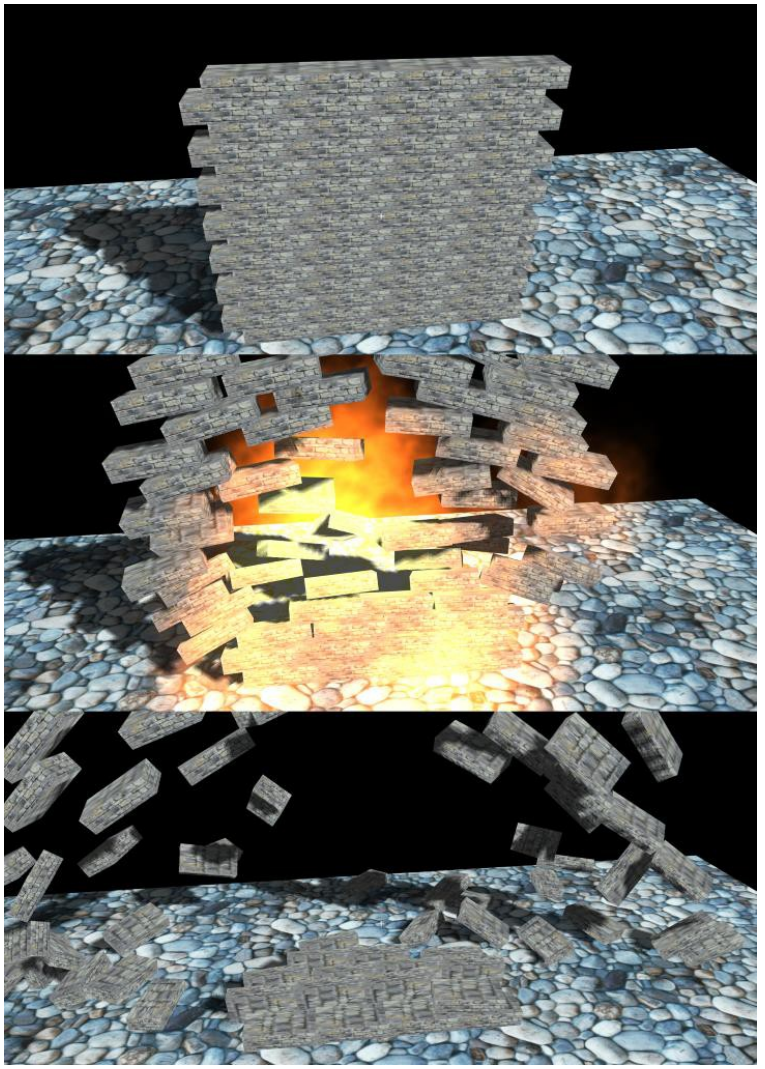

Physics

Demo

Physics



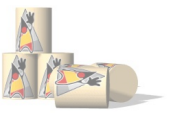
source: SDK tests



source: SDK tests

Audio

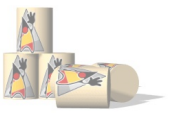
```
for (int i = 0; i < audioShots.length; i++) {
    audioShots[i] = new AudioNode(assetManager, "Sounds/Shots/Shot" + i + ".wav", false);
    audioShots[i].setPositional(false);
    audioShots[i].setLooping(false);
    audioShots[i].setVolume(2);
    rootNode.attachChild(audioShots[i]);
}

for (int i = 0; i < audioCans.length; i++) {
    audioCans[i] = new AudioNode(assetManager, "Sounds/Cans/Collide1.wav", false);
    audioCans[i].setPositional(true);
    audioCans[i].setLooping(false);
    audioCans[i].setReverbEnabled(false);
    audioCans[i].setVolume(1);
    rootNode.attachChild(audioCans[i]);
}

audioWater = new AudioNode(assetManager, "Sounds/Ambient/43760__digifishmusic__gentle-sea-on-flat-beach.wav", false);
audioWater.setLooping(true);
audioWater.setPositional(false);
audioWater.setVolume(0.3f);
rootNode.attachChild(audioWater);
audioWater.play();
```



Audio

```
audioShots[random.nextInt(audioShots.length)].playInstance();
```

```
public void collision(PhysicsCollisionEvent event) {
    if (event.getAppliedImpulse() >= physicsImpulseAudibleThreshold
            && (event.getNodeA().getName().startsWith("Can")
                || event.getNodeB().getName().startsWith("Can"))) {
        for (AudioNode audioCan : audioCans) {
            if (audioCan.getStatus() != AudioSource.Status.Playing) {
                audioCan.setLocalTranslation(event.getNodeA().getLocalTranslation());
                audioCan.playInstance();
                break;
            }
        }
    }
}
```



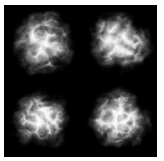
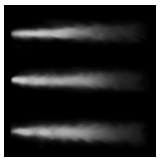
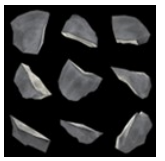
Audio

Demo

Particles

```
Material mat = new Material(assetManager, "Common/MatDefs/Misc/Particle.j3md");
mat.setTexture("Texture", assetManager.loadTexture("Effects/Snow/Snowflake.png"));

ParticleEmitter snow = new ParticleEmitter("Emitter", ParticleMesh.Type.Triangle, 10000);
snow.setMaterial(mat);
snow.setImagesX(1);
snow.setImagesY(1);
snow.setRotateSpeed(0.1f);
snow.setStartColor(ColorRGBA.White);
snow.setEndColor(ColorRGBA.White);
snow.getParticleInfluencer().setInitialVelocity(new Vector3f(0, -0.2f, 0));
snow.setStartSize(0.1f);
snow.setEndSize(0.1f);
snow.setGravity(0, 0.2f, 0);
snow.setLowLife(30f);
snow.setHighLife(30f);
snow.setParticlesPerSec(200);
snow.getParticleInfluencer().setVelocityVariation(0.02f);

snow.setShape(new EmitterBoxShape(new Vector3f(100, 20, 100), new Vector3f(-100, 20, -100)));

rootNode.attachChild(snow);
```



source: SDK test data

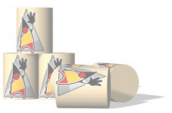
Particles

```
matTrail = new Material(assetManager, "Common/MatDefs/Misc/Particle.j3md");
matTrail.setTexture("Texture", assetManager.loadTexture("Effects/Smoke/Smoke.png"));

trail = new ParticleEmitter("Emitter", ParticleMesh.Type.Triangle, 100);
trail.setMaterial(matTrail);
trail.setImagesX(15);
trail.setImagesY(1);
trail.setStartColor(new ColorRGBA(1.0f, 0.855f, 0.608f, 0.5f));
trail.setEndColor(new ColorRGBA(0.5f, 0.428f, 0.304f, 0f));
trail.getParticleInfluencer().setInitialVelocity(Vector3f.ZERO);
trail.setStartSize(0.5f);
trail.setEndSize(0.1f);
trail.setGravity(0, 1, 0);
trail.setLowLife(0.3f);
trail.setHighLife(3.7f);
trail.setParticlesPerSec(ppsMax);
trail.getParticleInfluencer().setVelocityVariation(0.1f);
trail.setShape(new EmitterPointShape(Vector3f.ZERO));
```


source: SDK test data

Particles

```
private void update(float tpf) {
    timeActive += tpf;

    if (timeActive >= 18.0f) {
        trail.setParticlesPerSec(ppsMax);
        rootNode.detachChild(trail);
    } else if (timeActive >= 15.0f) {
        trail.setParticlesPerSec(
                Math.max(ppsMax - ppsMax * (timeActive - 15.0f), 0.0f));
        trail.setLocalTranslation(spatial.getWorldTranslation());
    } else {
        trail.setLocalTranslation(spatial.getWorldTranslation());
    }
}
```



Particles

Demo

More Features - Interface

source: Illarion, http://illarion.org/

Nifty GUI

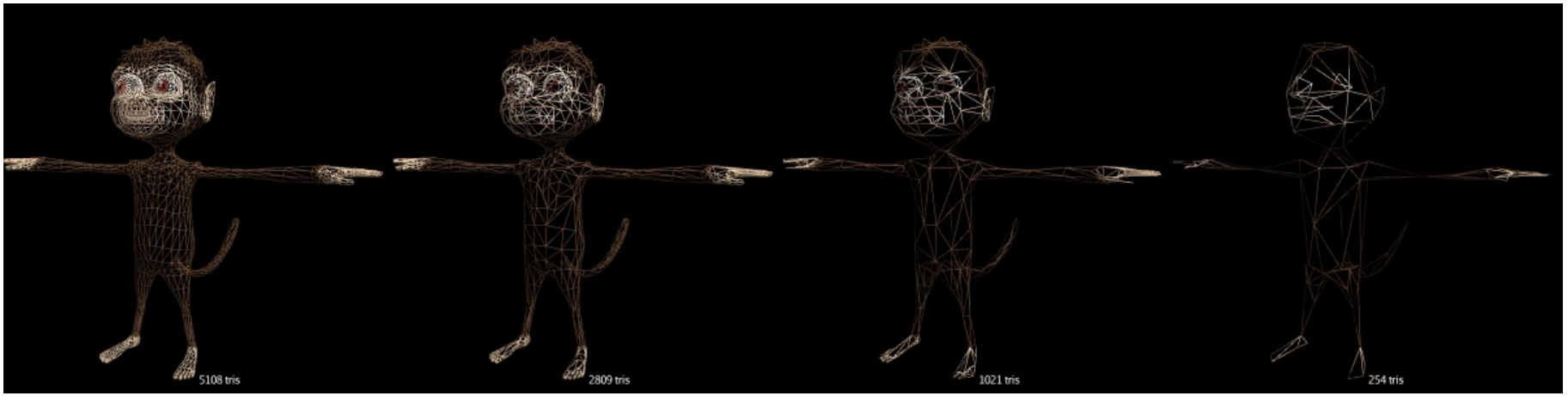
Your Open Source Java OpenGL GUI

source: Pirate Hell, http://www.indiedb.com/games/piratehell

jMonkeyEngine

More Features – AppState & LOD

source: Pirate Hell, http://www.indiedb.com/games/piratehell

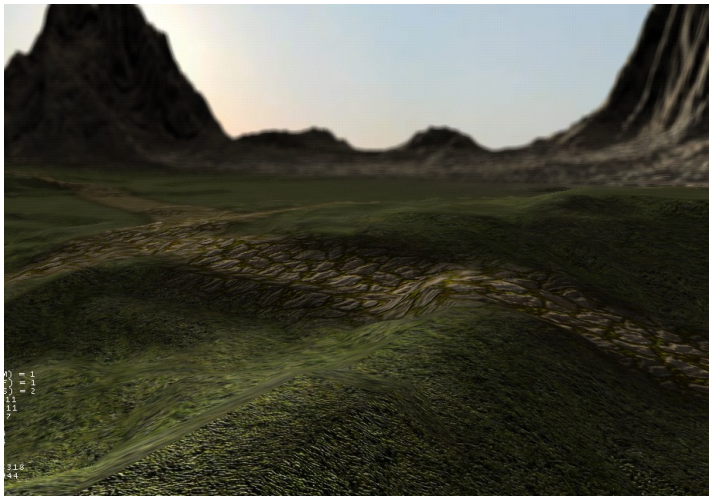


source: http://hub.jmonkeyengine.org/

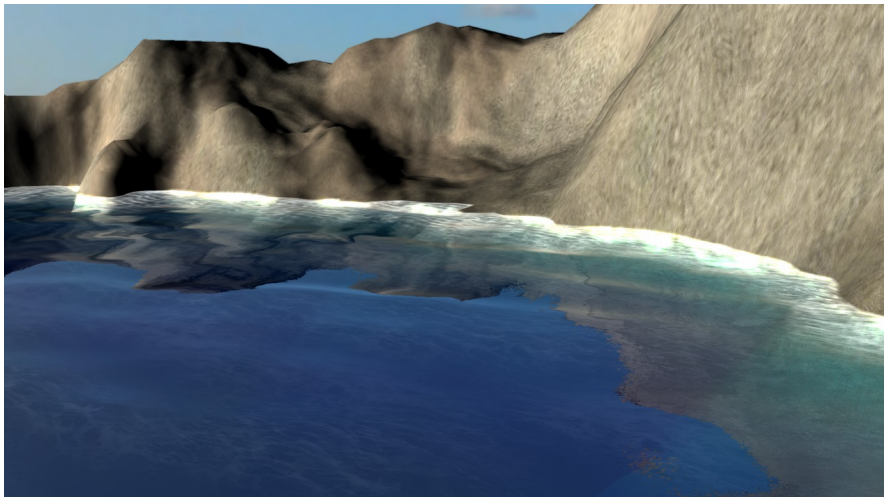
More Features - Shaders

source: http://hub.jmonkeyengine.org/

More Features - Shaders



source: http://hub.jmonkeyengine.org/

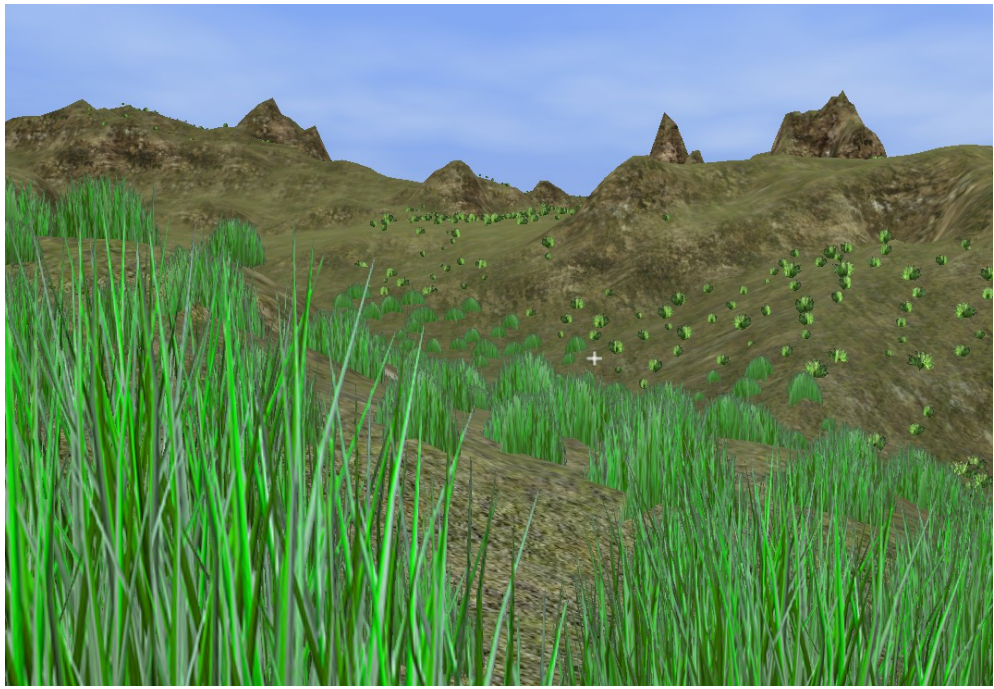


source: SDK tests

source: http://hub.jmonkeyengine.org/

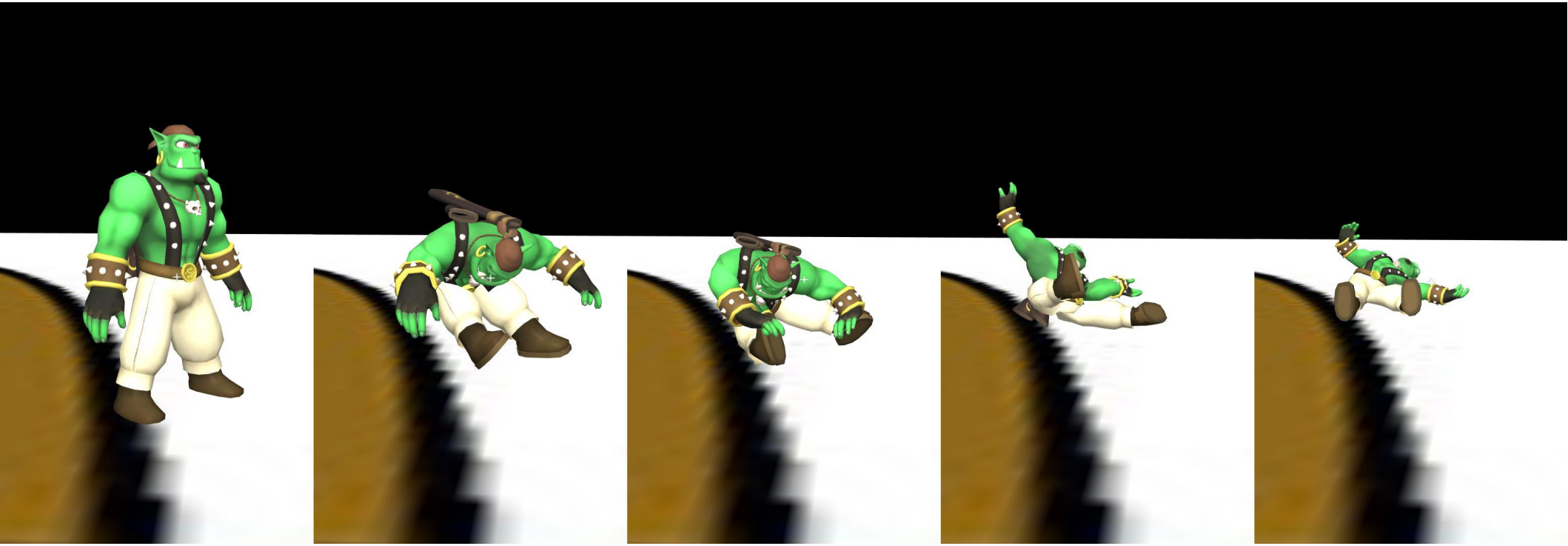
source: http://hub.jmonkeyengine.org/

More Features – The Forester

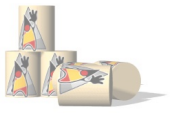


source: http://hub.jmonkeyengine.org/forum/

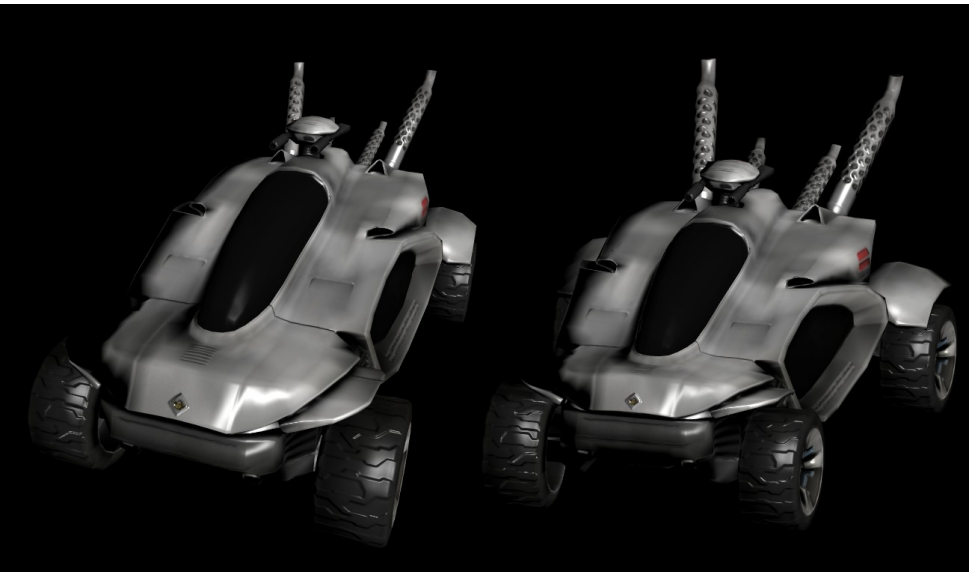
More Features - Physics



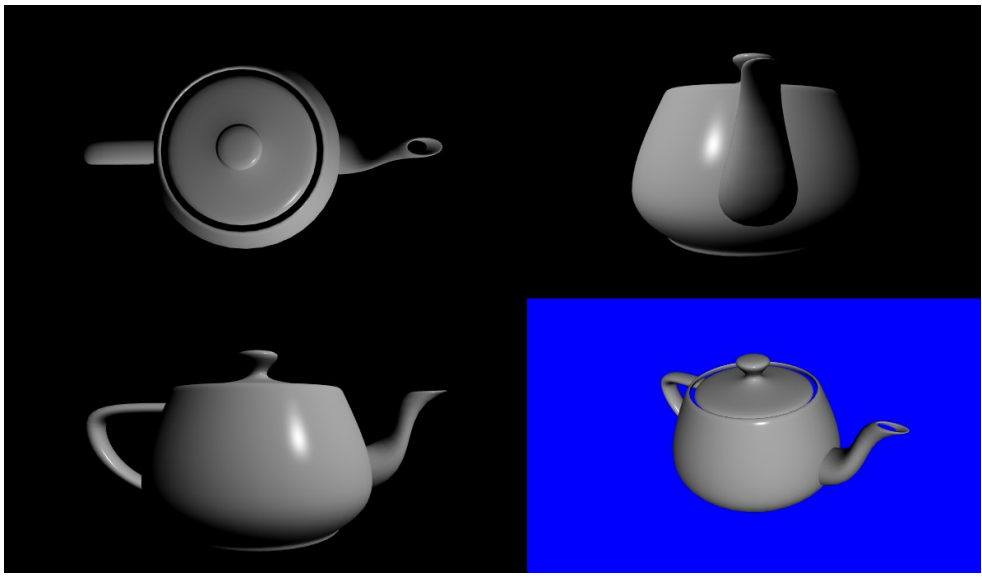
source: SDK tests



More Features – Projection & MultiView



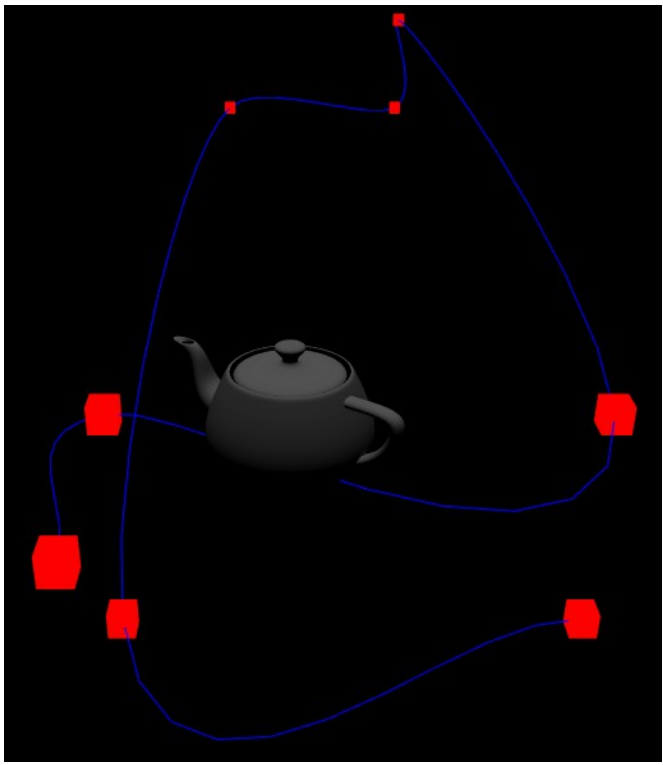
source: SDK test data



source: SDK tests

More Features - Cinematics

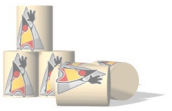
source: http://jmonkeyengine.org/



source: SDK tests

Even More Features

- Java Swing canvas
- Video + audio recording (built-in)
- Multiplayer networking (SpiderMonkey)
- Android supported, iOS on the way
- Debugging features (wireframe rendering)



Resources

- http://jmonkeyengine.org/
- http://hub.jmonkeyengine.org/
- http://opengameart.org/ (textures)
- http://www.freesound.org/ (sounds)

Have fun!

source: http://jmonkeyengine.org/