

## Physics (RAGE) – bouncing ball example

```

...
import ray.physics.PhysicsEngine;
import ray.physics.PhysicsObject;
import ray.physics.PhysicsEngineFactory;

public class myGame extends VariableFrameRateGame
{
    private SceneNode      ball1Node, ball2Node, gndNode;
    private SceneNode      cameraPositionNode;

    private final static String GROUND_E = "Ground";
    private final static String GROUND_N = "GroundNode";

    private PhysicsEngine physicsEng;
    private PhysicsObject ball1PhysObj, ball2PhysObj, gndPlaneP;
    private boolean running = false;

    public MyGame()
    { super();
    }

    public static void main(String[] args) { ... }

    public void setupCameras(. . .) { ... }

    protected void setUpScene(Engine engine, SceneManager sm)
                                throws IOException
    { SceneNode rootNode = sm.getRootSceneNode();

      // Ball 1
      Entity ball1Entity = sm.createEntity("ball1", "earth.obj");
      ball1Node = rootNode.createChildSceneNode("Ball1Node");
      ball1Node.attachObject(ball1Entity);
      ball1Node.setLocalPosition(0, 2, -2);

      // Ball 2
      Entity ball2Entity = sm.createEntity("Ball2", meshFilename);
      ball2Node = rootNode.createChildSceneNode("Ball2Node");
      ball2Node.attachObject(ball2Entity);
      ball2Node.setLocalPosition(-1, 10, -2);

      // Ground plane
      Entity groundEntity = sm.createEntity(GROUND_E, "cube.obj");
      groundNode = rootNode.createChildSceneNode(GROUND_N);
      groundNode.attachObject(groundEntity);
      groundNode.setLocalPosition(0, -7, -2);

      initPhysicsSystem();
      createRagePhysicsWorld();
      ...
      System.out.println("Press SPACE to start the physics engine!");
    }

    protected void update(Engine engine)
    { float time = engine.getElapsedTimeMillis();
      if (running)
      { Matrix4 mat;
        physicsEngine.update(time);
        for (SceneNode s : engine.getSceneManager().getSceneNodes())
        { if (s.getPhysicsObject() != null)
          { mat = Matrix4f.createFrom(toFloatArray(
                                s.getPhysicsObject().getTransform()));
            s.setLocalPosition(mat.value(0,3),mat.value(1,3),
                                mat.value(2,3));
          }
        }
      }
    }
}

```

graphics world

```

private void initPhysicsSystem()
{ String engine = "ray.physics.JBullet.JBulletPhysicsEngine";
  float[] gravity = {0, -3f, 0};

  physicsEng = PhysicsEngineFactory.createPhysicsEngine(engine);
  physicsEng.initSystem();
  physicsEng.setGravity(gravity);
}

private void createRagePhysicsWorld()
{ float mass = 1.0f;
  float up[] = {0,1,0};
  double[] temptf;

  temptf = toDoubleArray(ball1Node.getLocalTransform().toFloatArray());
  ball1PhysObj = physicsEng.addSphereObject(physicsEngine.nextUID(),
                                              mass, temptf, 2.0f);

  ball1PhysObj.setBounciness(1.0f);
  ball1Node.setPhysicsObject(ball1PhysObj);

  temptf = toDoubleArray(ball2Node.getLocalTransform().toFloatArray());
  ball2PhysObj = physicsEng.addSphereObject(physicsEngine.nextUID(),
                                              mass, temptf, 2.0f);

  ball2PhysObj.setBounciness(1.0f);
  ball2Node.setPhysicsObject(ball2PhysObj);

  temptf = toDoubleArray(gndNode.getLocalTransform().toFloatArray());
  gndPlaneP = physicsEng.addStaticPlaneObject(physicsEngine.nextUID(),
                                              temptf, up, 0.0f);

  gndPlaneP.setBounciness(1.0f);
  gndNode.scale(3f, .05f, 3f);
  gndNode.setLocalPosition(0, -7, -2);
  gndNode.setPhysicsObject(gndPlaneP);

  // can also set damping, friction, etc.
}

public void keyPressed(KeyEvent e)
{ switch (e.getKeyCode())
  { case KeyEvent.VK_SPACE:
    System.out.println("Starting Physics!");
    running = true;
    break;
  }
  super.keyPressed(e);
}

private float[] toFloatArray(double[] arr)
{ if (arr == null) return null;
  int n = arr.length;
  float[] ret = new float[n];
  for (int i = 0; i < n; i++)
  { ret[i] = (float)arr[i];
  }
  return ret;
}

private double[] toDoubleArray(float[] arr)
{ if (arr == null) return null;
  int n = arr.length;
  double[] ret = new double[n];
  for (int i = 0; i < n; i++)
  { ret[i] = (double)arr[i];
  }
  return ret;
}
}

```

physics world