

# Optimisation

Fog

Billboards

Level of Detail (LOD)

Batching

# Fog Motivation

- Fog increases realism and declutters a scene
- Fog also obscures distant shapes, hiding the far clipping plane
  - `Camera.setFrustumFar(float frustumFar)`
- jME provides a fog post processing filter

# FogFilter

- Extends `com.jme3.post.Filter`
- A 2D effect applied to the rendered scene
- FogFilter has:
  - A color
  - A fog density
  - A fog distance

# FogFilter

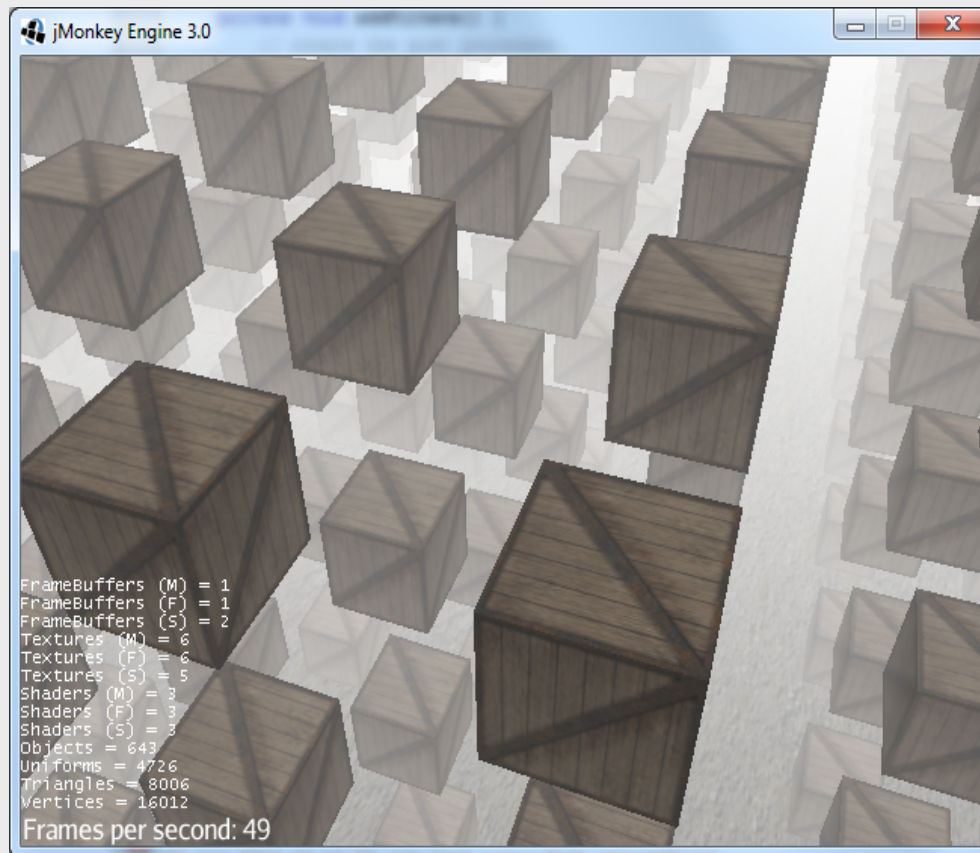
- An exponential function is used to calculate thickness
- Higher density creates thicker fog
- Distance controls how far the fog reaches

# FogFilter

- Create fog

```
FogFilter fog = new FogFilter();
fog.setFogColor(new ColorRGBA(1f, 1f, 1f, 1f));
fog.setFogDensity(2.0f);
fog.setFogDistance(100.0f);
FilterPostProcessor fpp = new
    FilterPostProcessor(assetManager);
fpp.addFilter(fog);
viewport.addProcessor(fpp);
```

# FogFilter Example

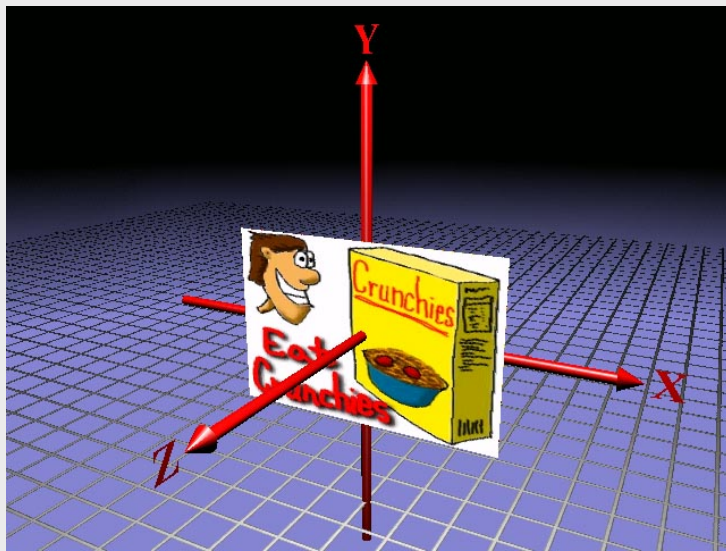


# BillboardControl

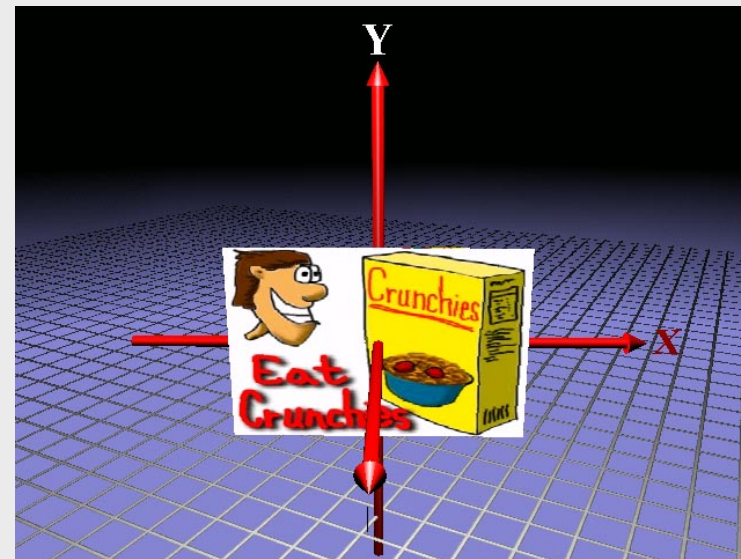
- BillboardControl is a control that:
  - Tracks the camera
  - Generates a rotation so that the object is facing the camera or screen
  - Writes that transform to its Spatial

# BillboardControl

Viewer steps to the right . . .



. . . and the control immediately rotates the spatial

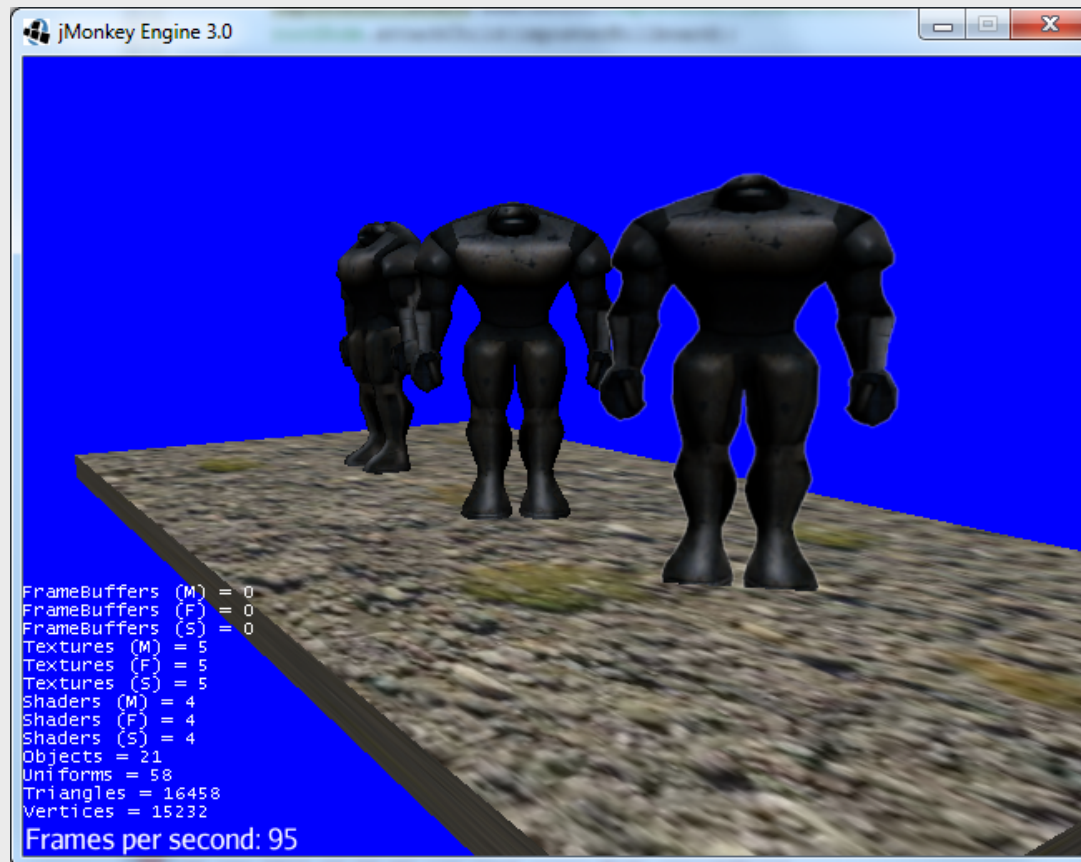




# BillboardControl.Alignment modes

- Determines how the billboard is aligned to the screen or camera
- *AxialY* aligns to the camera, but keeps the local Y axis fixed.
- *AxialZ* aligns to the camera, but keeps the local Z axis fixed.
- *Camera* aligns to the camera position
- *Screen* aligns to the screen

# BillboardControl example



# Level of detail

- jME supports level of detail in its core
- Mesh can have LOD index buffers
  - `void setLodLevels(VertexBuffer[] lodLevels)`
  - `int getNumLodLevels()`
  - `VertexBuffer getLodLevel(int lod)`
- Is used instead of the mesh index buffer if set

# Level of detail

- Spatial has method to set which LOD to use
  - void setLodLevel(int lod)
    - Node propagates to children
    - Geometry stores the lod level

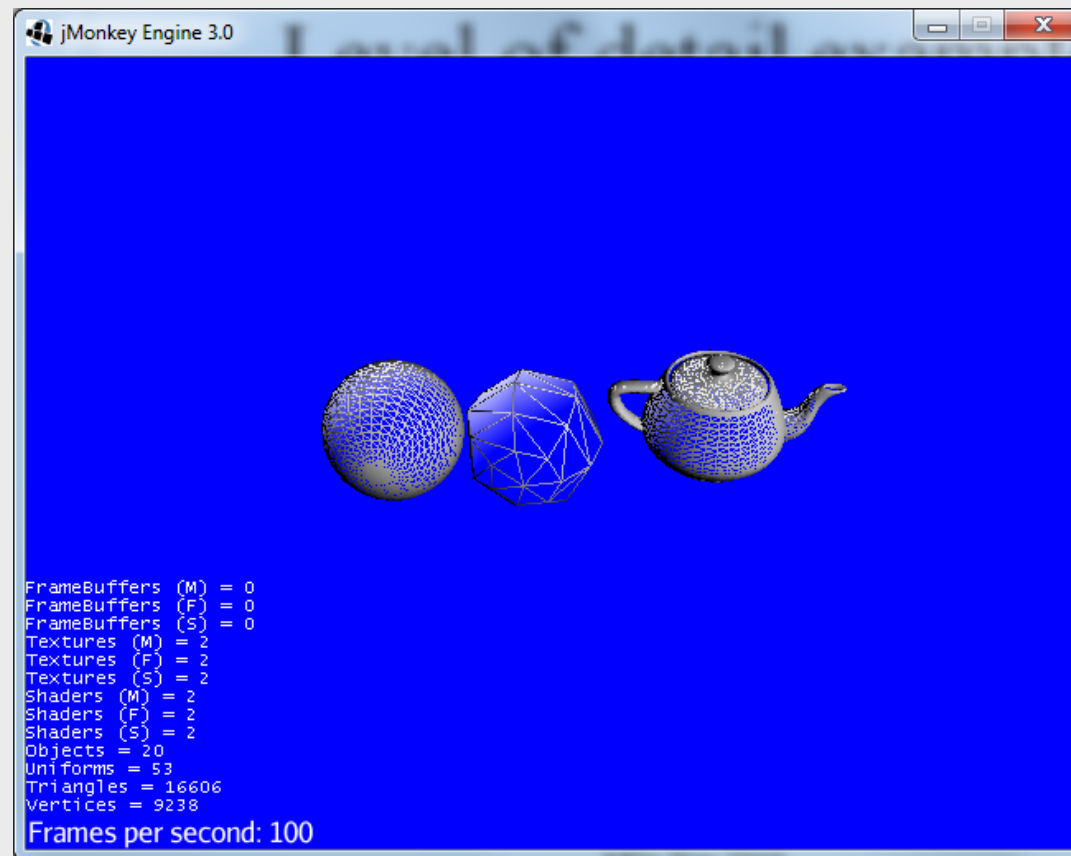
# LodControl

- Determines what level of detail a Geometry should be
- Based on how many pixels on the screen the Geometry is taking up
  - `get/setTrisPerPixel`
- `BoundingBox` used to calculate screen coverage
- Geometry must move atleast the distance tolerance relative to the camera before switching to a new LOD
  - `get/setDistTolerance`
- Can only be added to geometries

# MeshLodControl

- Not part of jME but part of the course material
- Allows switching between meshes instead of index buffers
- Still have to use the same material

# Level of detail example



# BatchNode

- Geometries in sub scenegraph is combined
- Holds one geometry per material
- Must call batch() after changing subgraph
- Less draw calls
  - Graphic cards good at processing larg chunks of data
- Culled as one
  - Can degrade performance



# BatchNode example

