Purely Functional 3D in Typed Racket

or

Spoon-Feeding For the Functional Purist

Neil Toronto

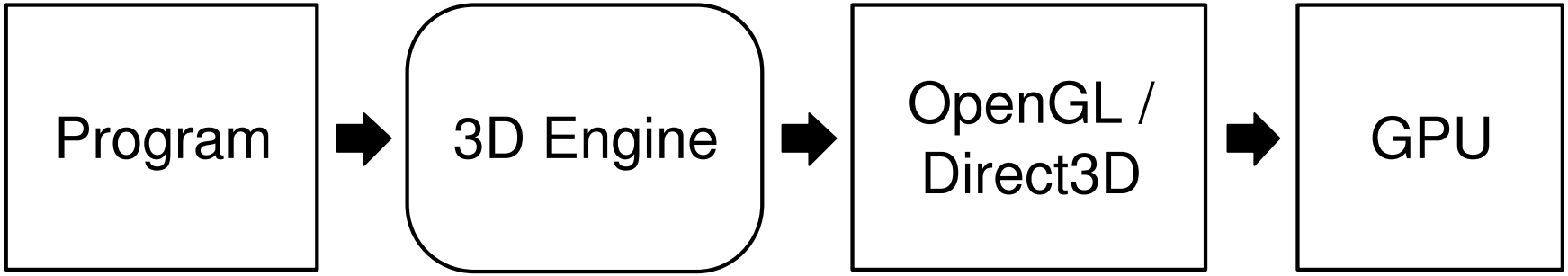
RacketCon 2014

3D Engines

- High-level abstraction layer (usually a library) between program and GPU

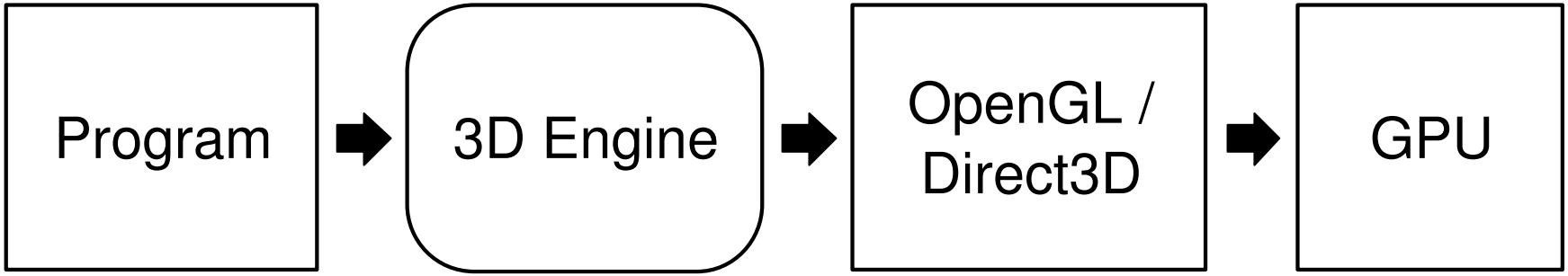
3D Engines

- High-level abstraction layer (usually a library) between program and GPU



3D Engines

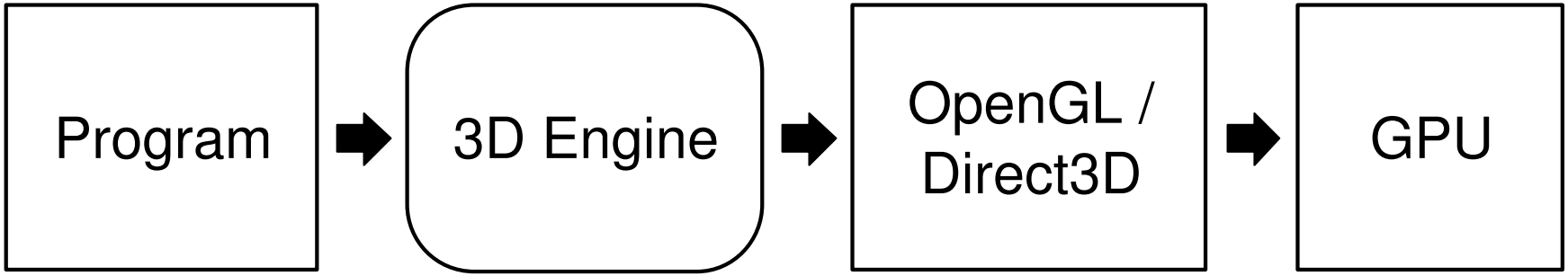
- High-level abstraction layer (usually a library) between program and GPU



- Examples: Unity, Unreal, Rage (rendering parts)

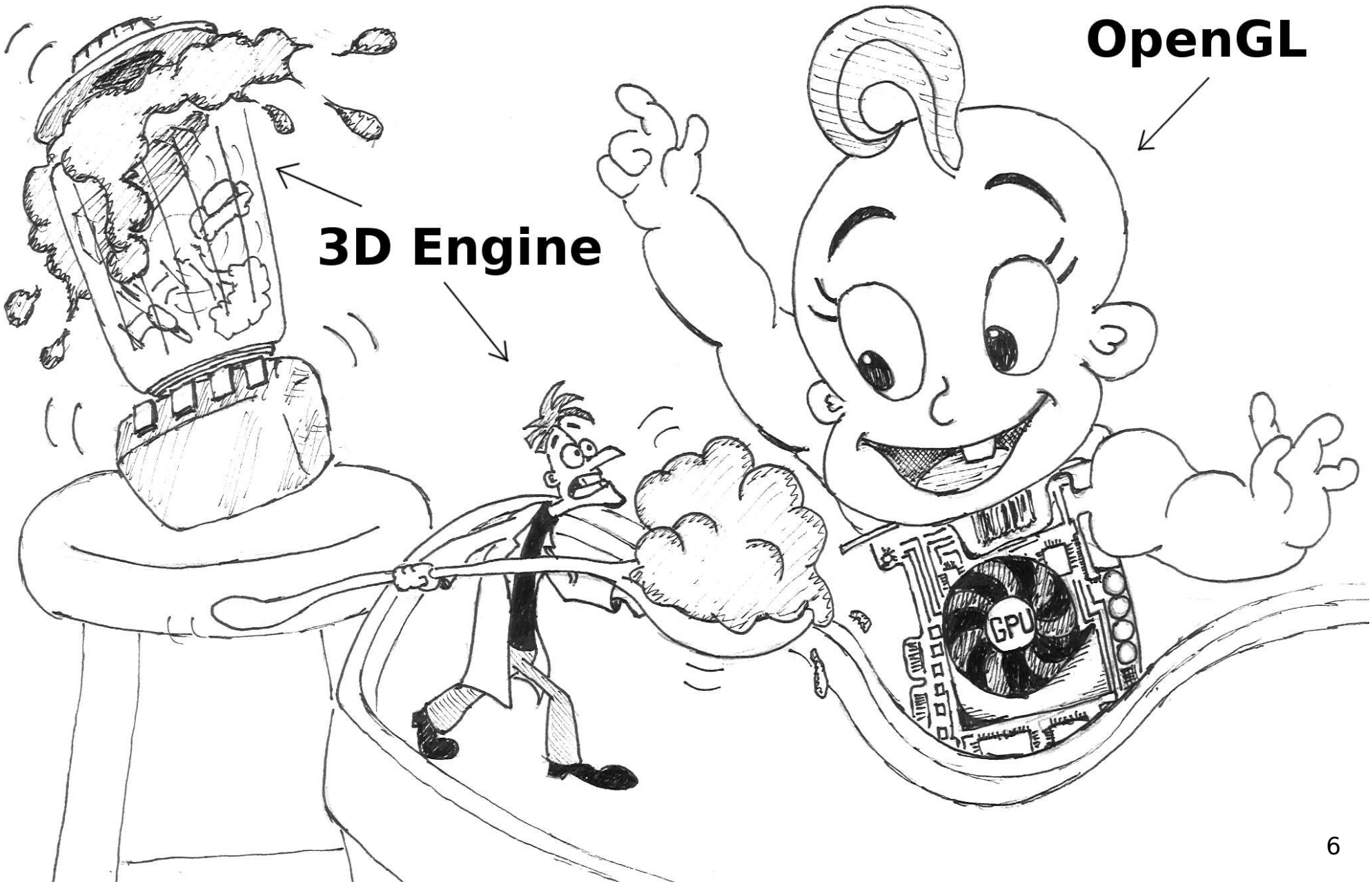
3D Engines

- High-level abstraction layer (usually a library) between program and GPU

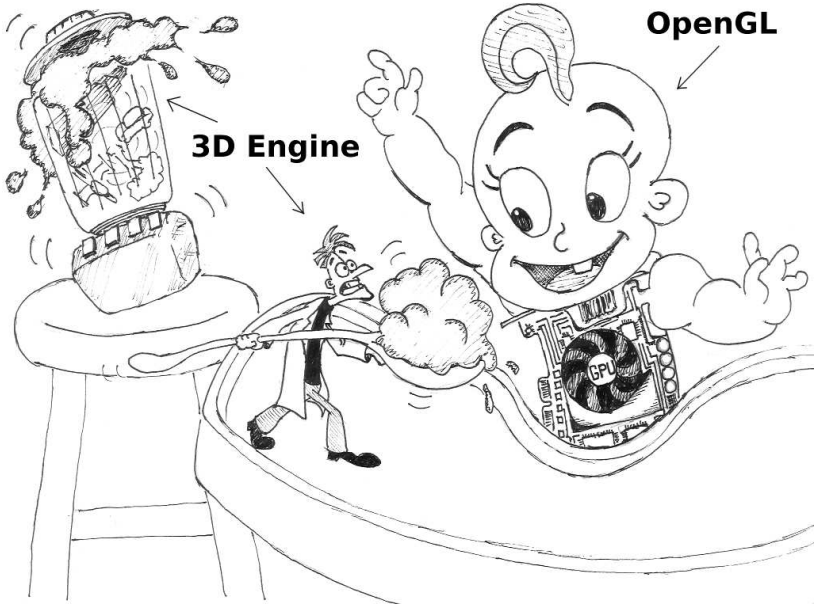


- Examples: Unity, Unreal, Rage (rendering parts)
- Problem: All are built around mutation

What Does a 3D Engine Do?

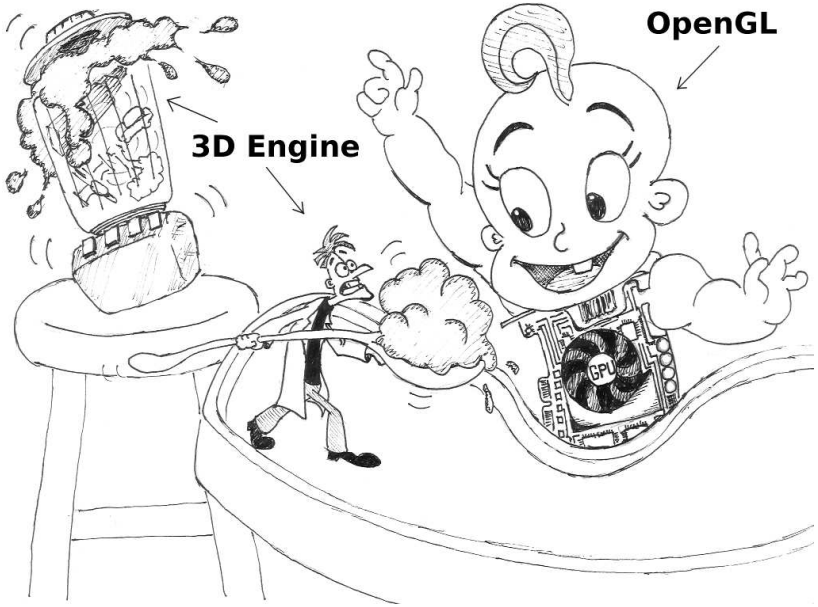


What Does a 3D Engine Do?



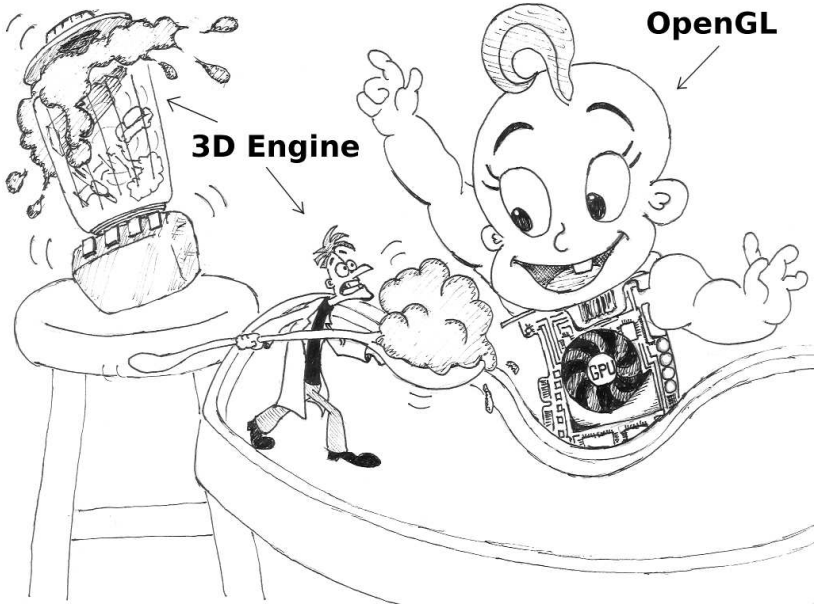
- Giant baby consumes homogenous, bland, first-order vertex data

What Does a 3D Engine Do?



- Giant baby consumes homogenous, bland, first-order vertex data
- Giant baby often must be tricked

What Does a 3D Engine Do?



- Giant baby consumes homogenous, bland, first-order vertex data
- Giant baby often must be tricked
- Giant baby is very hungry but can swallow only a few hundred spoonfuls per frame

Imperative Engine API

Commands

```
(add-shape! ...)
(move-shape! ...)
(add-light! ...)
(draw-scene!)
```

3D Engine

Scene Database

Algorithms

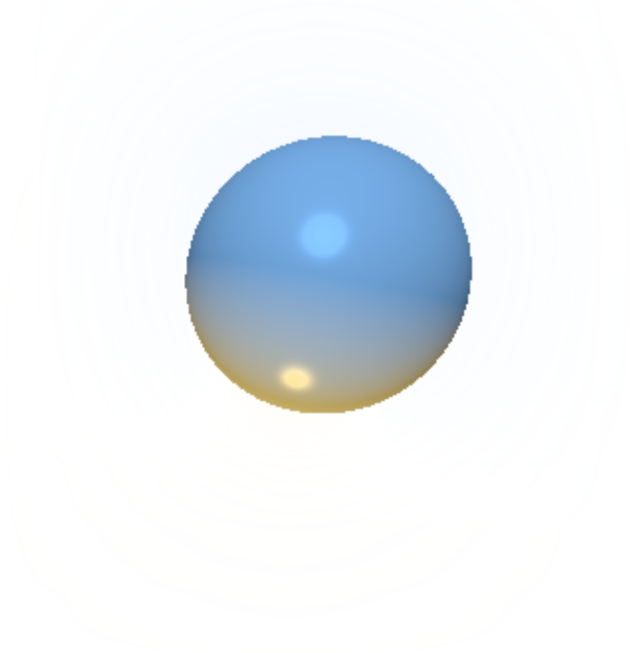
High-Powered Blender

Scene Databases

Scene database: a data structure that makes *spatial* queries (e.g. point-inside) efficient

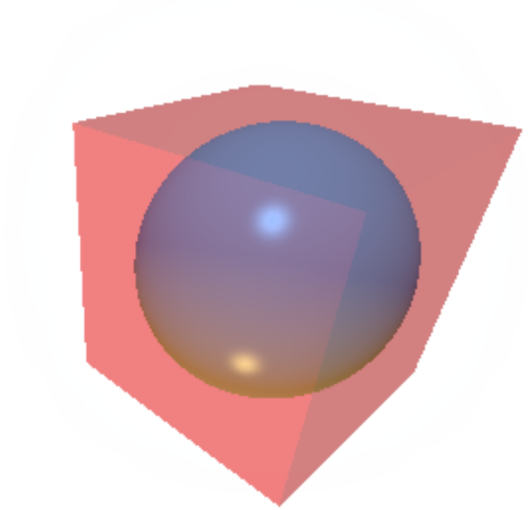
Scene Databases

Scene database: a data structure that makes *spatial* queries (e.g. point-inside) efficient



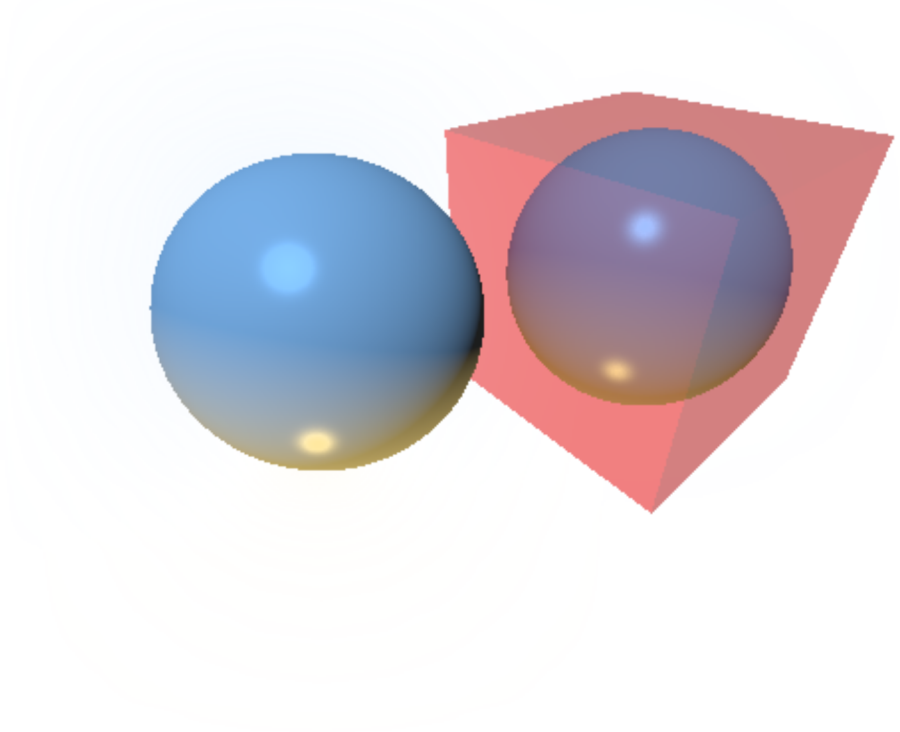
Scene Databases

Scene database: a data structure that makes *spatial* queries (e.g. point-inside) efficient



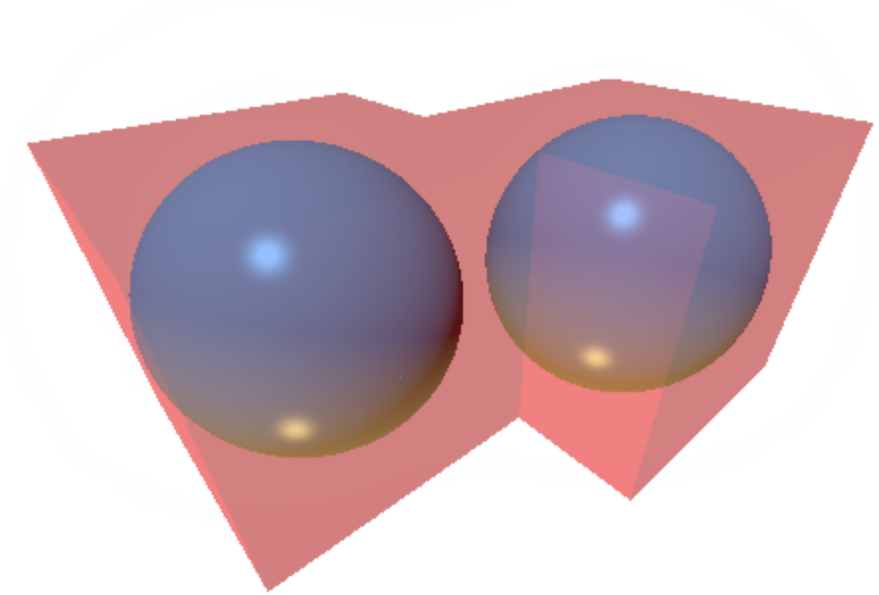
Scene Databases

Scene database: a data structure that makes *spatial* queries (e.g. point-inside) efficient



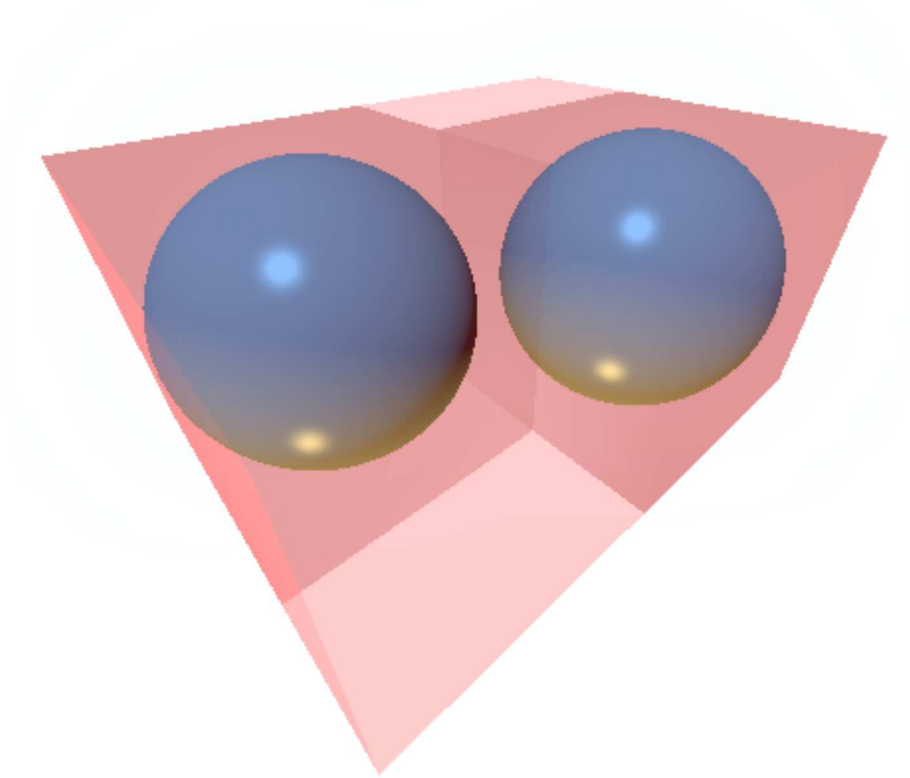
Scene Databases

Scene database: a data structure that makes *spatial* queries (e.g. point-inside) efficient



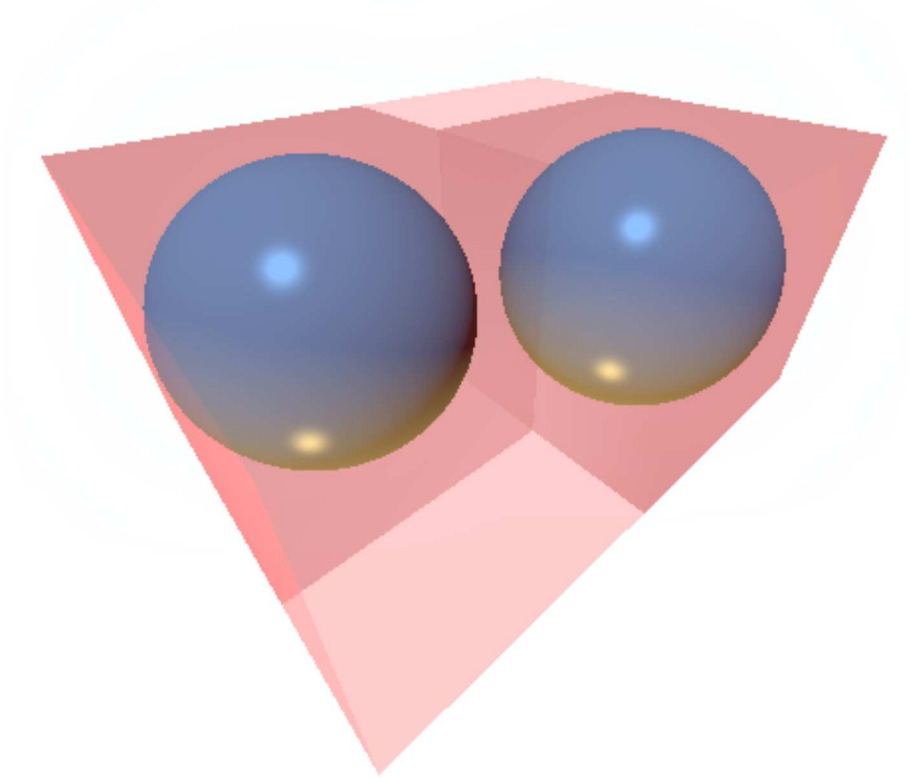
Scene Databases

Scene database: a data structure that makes *spatial* queries (e.g. point-inside) efficient



Scene Databases

Scene database: a data structure that makes *spatial* queries (e.g. point-inside) efficient



Most databases are bounding hierarchies, and almost all are *trees*

Mostly Functional Engine API

Computations

```
(make-shape ...)
(make-light ...)
(scene-union ...)
(scene-transform ...)
(draw-scene! ...)
```

3D Engine

Combinators

Algorithms

High-Powered Blender

Awesome! Let’s Try It!

```
>
```

Awesome! Let's Try It!

```
> (define unit-sphere
    (shape->scene
     (make-sphere-shape identity-flt3
                        (flvector 1.0 1.0 1.0 1.0)
                        (flvector 0.0 1.0 0.0 2.0)
                        (material 0.01 0.29 0.7 0.1)
                        #f)))

>
```

Awesome! Let's Try It!

```
> (define unit-sphere
    (shape->scene
     (make-sphere-shape identity-flt3
                        (flvector 1.0 1.0 1.0 1.0)
                        (flvector 0.0 1.0 0.0 2.0)
                        (material 0.01 0.29 0.7 0.1)
                        #f)))

> (scene-union unit-sphere
               (scene-transform
                unit-sphere
                (translate-flt3 (flvector 1.0 0.0 0.0))))
```

Awesome! Let's Try It!

```
> (define unit-sphere
    (shape->scene
     (make-sphere-shape identity-flt3
                        (flvector 1.0 1.0 1.0 1.0)
                        (flvector 0.0 1.0 0.0 2.0)
                        (material 0.01 0.29 0.7 0.1)
                        #f)))

> (scene-union unit-sphere
               (scene-transform
                unit-sphere
                (translate-flt3 (flvector 1.0 0.0 0.0))))

(scene-node
 (Nonempty-FlRect3 (flvector -1.0 -1.0 -1.0) (flvector 2.0 1.0 1.0))
 2
 (scene-leaf
  (Nonempty-FlRect3 (flvector -1.0 -1.0 -1.0) (flvector 1.0 1.0 1.0))
  1
  (sphere-shape (flidentity3) (flvector 1.0 1.0 1.0 1.0) (flvector 0.0 1.0 0.0 2.0)
                (material 0.01 0.29 0.7 0.1) #f))
 (scene-tran
  (Nonempty-FlRect3 (flvector 0.0 -1.0 -1.0) (flvector 2.0 1.0 1.0))
  1
  (flaffine3 (flvector 1.0 0.0 0.0 1.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0))
  (flaffine3 (flvector 1.0 0.0 0.0 -1.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0))
  (scene-leaf
   (Nonempty-FlRect3 (flvector -1.0 -1.0 -1.0) (flvector 1.0 1.0 1.0))
   1
   (sphere-shape (flidentity3) (flvector 1.0 1.0 1.0 1.0) (flvector 0.0 1.0 0.0 2.0)
                 (material 0.01 0.29 0.7 0.1) #f))))
```

A More Excellent Way

>

A More Excellent Way

```
> (require pict3d)
>
```

A More Excellent Way

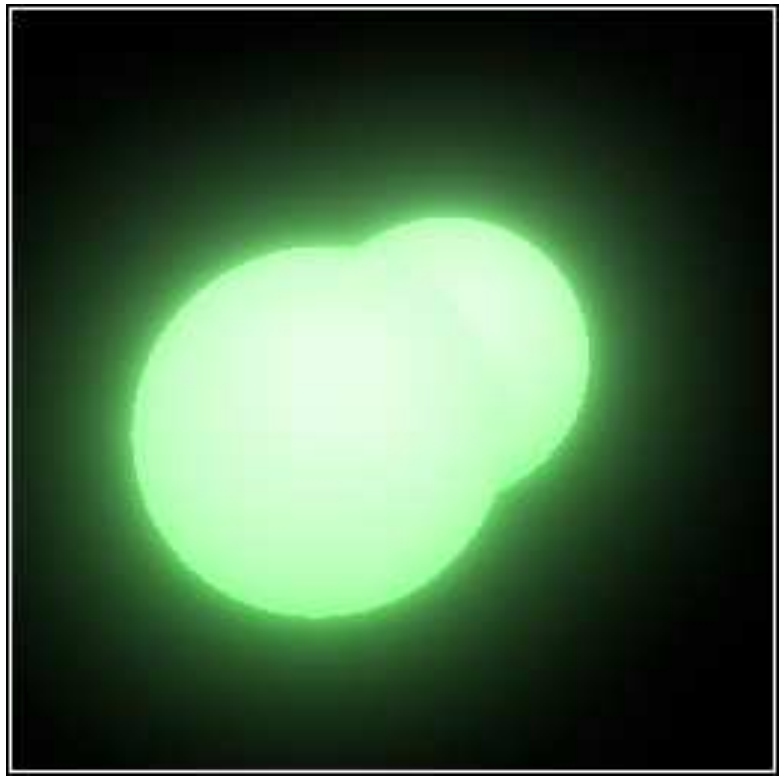
```
> (require pict3d)

> (with-emitted '(0 1 0 2)
    (combine (sphere '(0 0 0) 1)
             (sphere '(1 0 0) 1)))
```

A More Excellent Way

```
> (require pict3d)

> (with-emitted '(0 1 0 2)
    (combine (sphere '(0 0 0) 1)
             (sphere '(1 0 0) 1)))
```



Demos

Design Goals

- Make it fast
  - May require users to tweak, cache, give hints
  - Makes the high-powered blender unsafe and impure

Design Goals

- Make it fast
  - May require users to tweak, cache, give hints
  - Makes the high-powered blender unsafe and impure
- Use and allow modern rendering techniques (i.e. make it cool)

Design Goals

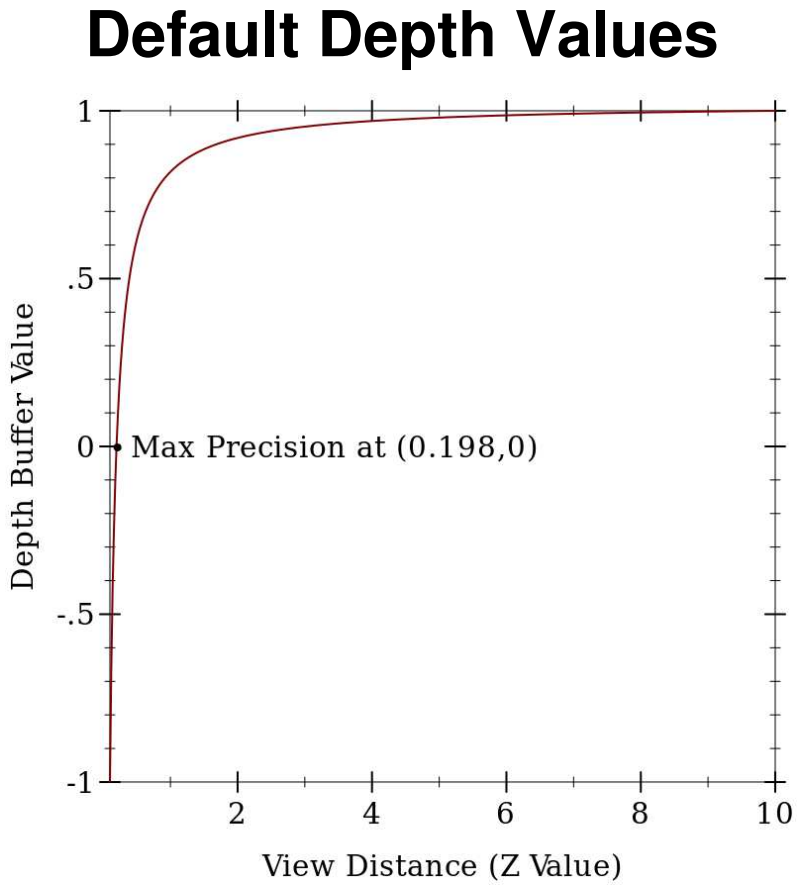
- Make it fast
  - May require users to tweak, cache, give hints
  - Makes the high-powered blender unsafe and impure
- Use and allow modern rendering techniques (i.e. make it cool)
- Minimize confusing surprises

Design Goals

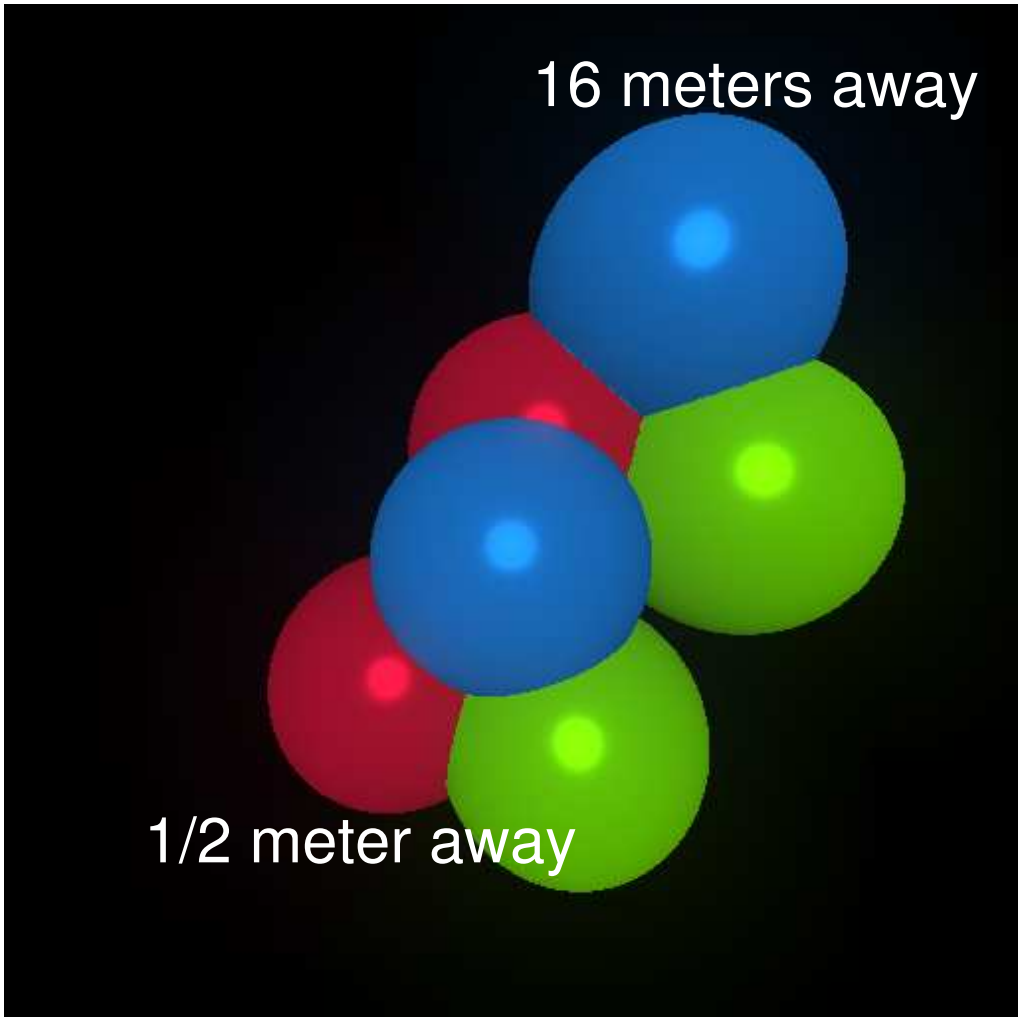
- Make it fast
  - May require users to tweak, cache, give hints
  - Makes the high-powered blender unsafe and impure
- Use and allow modern rendering techniques (i.e. make it cool)
- Minimize confusing surprises
- Last two goals are often complementary

Default Depth Buffer

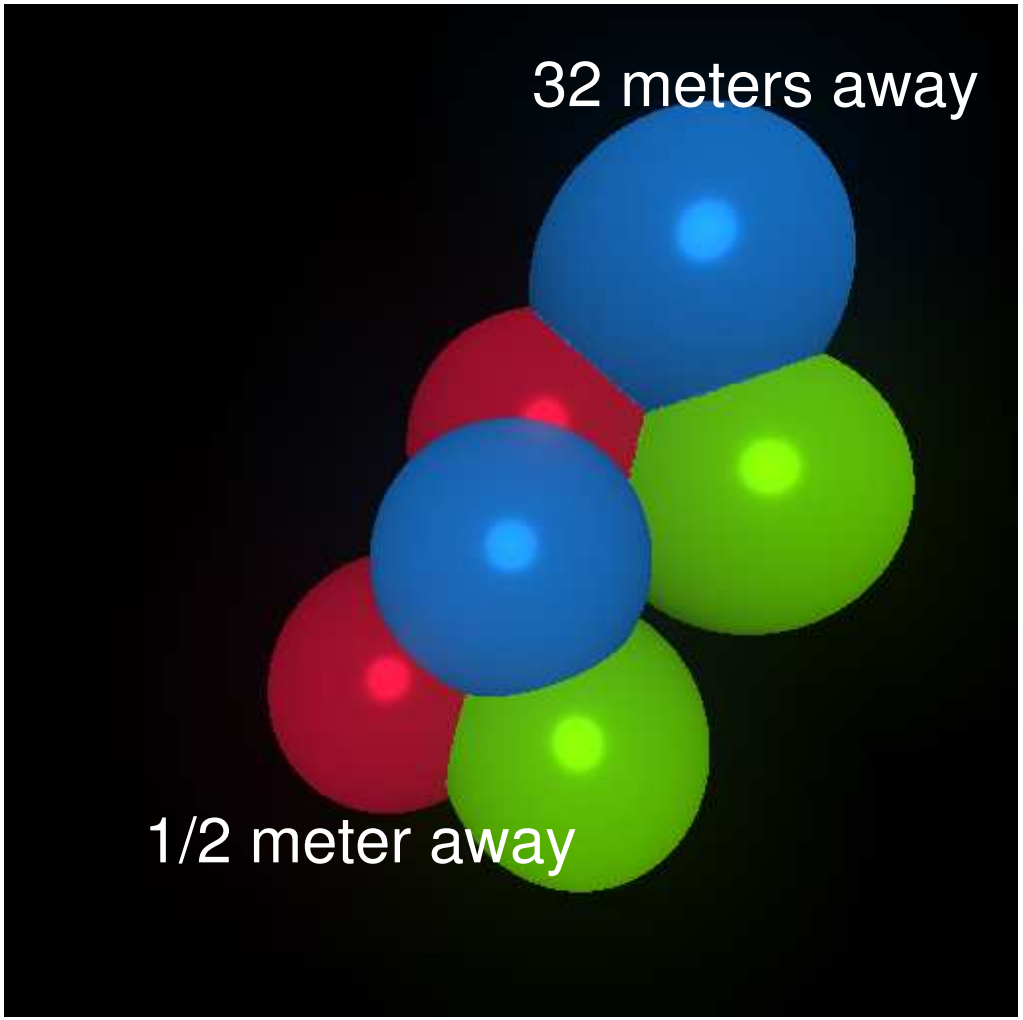
`z-near` = **`0.1`** and **`z-far`** = **`10.0`**



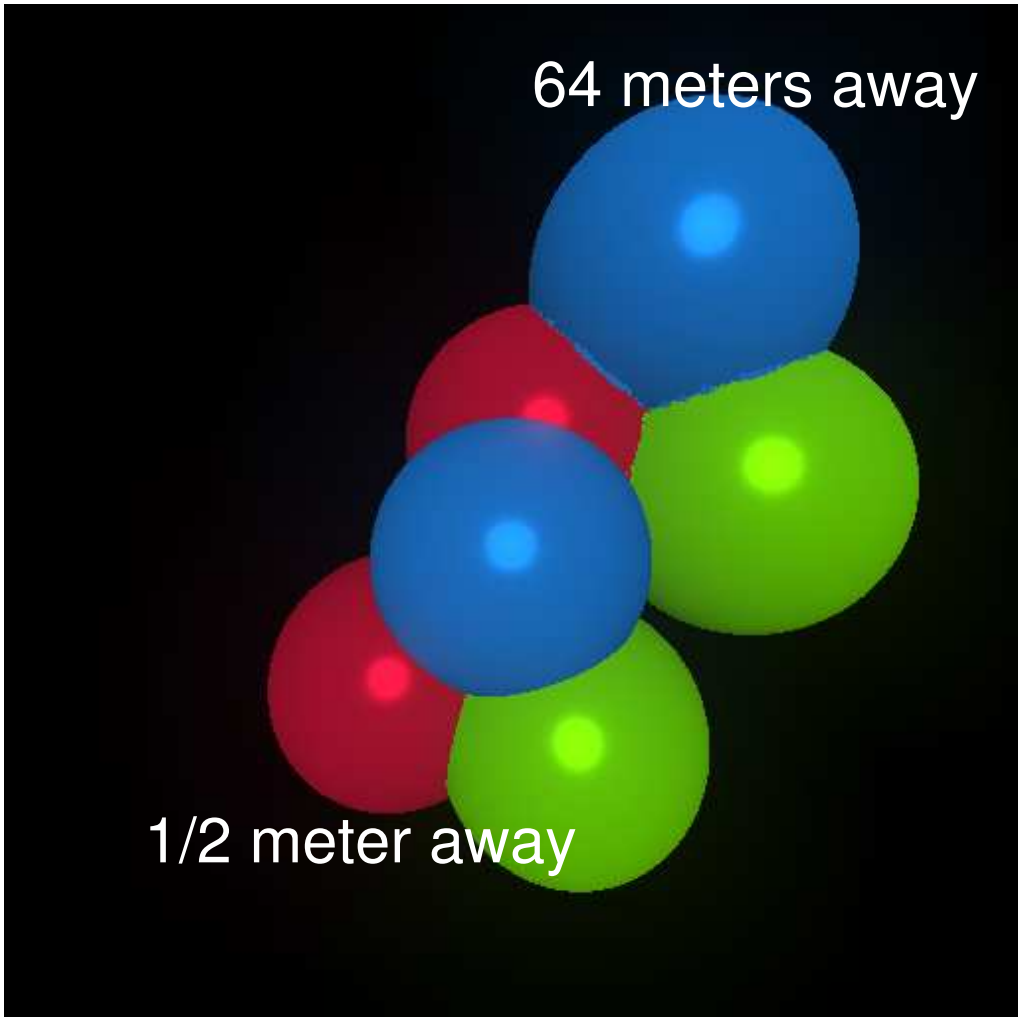
Default Depth Buffer: Confusing Surprises



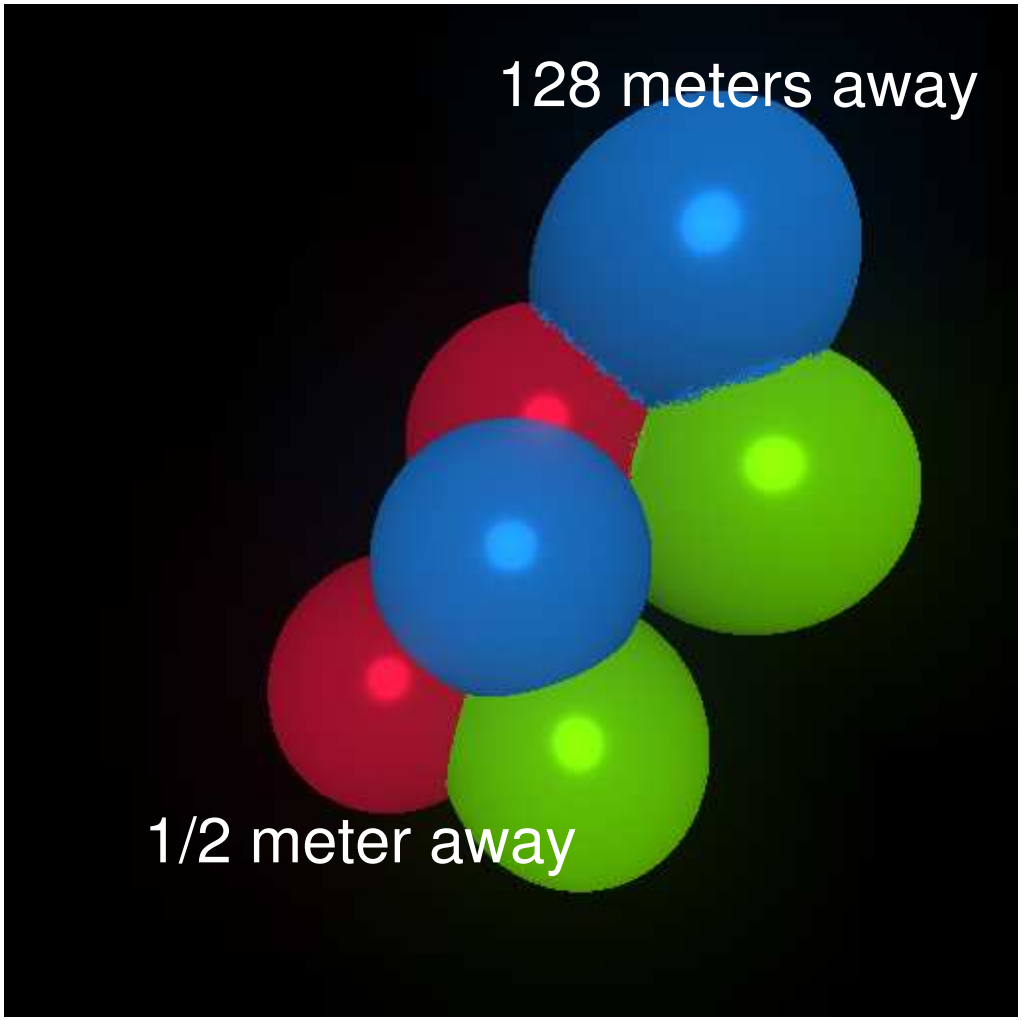
Default Depth Buffer: Confusing Surprises



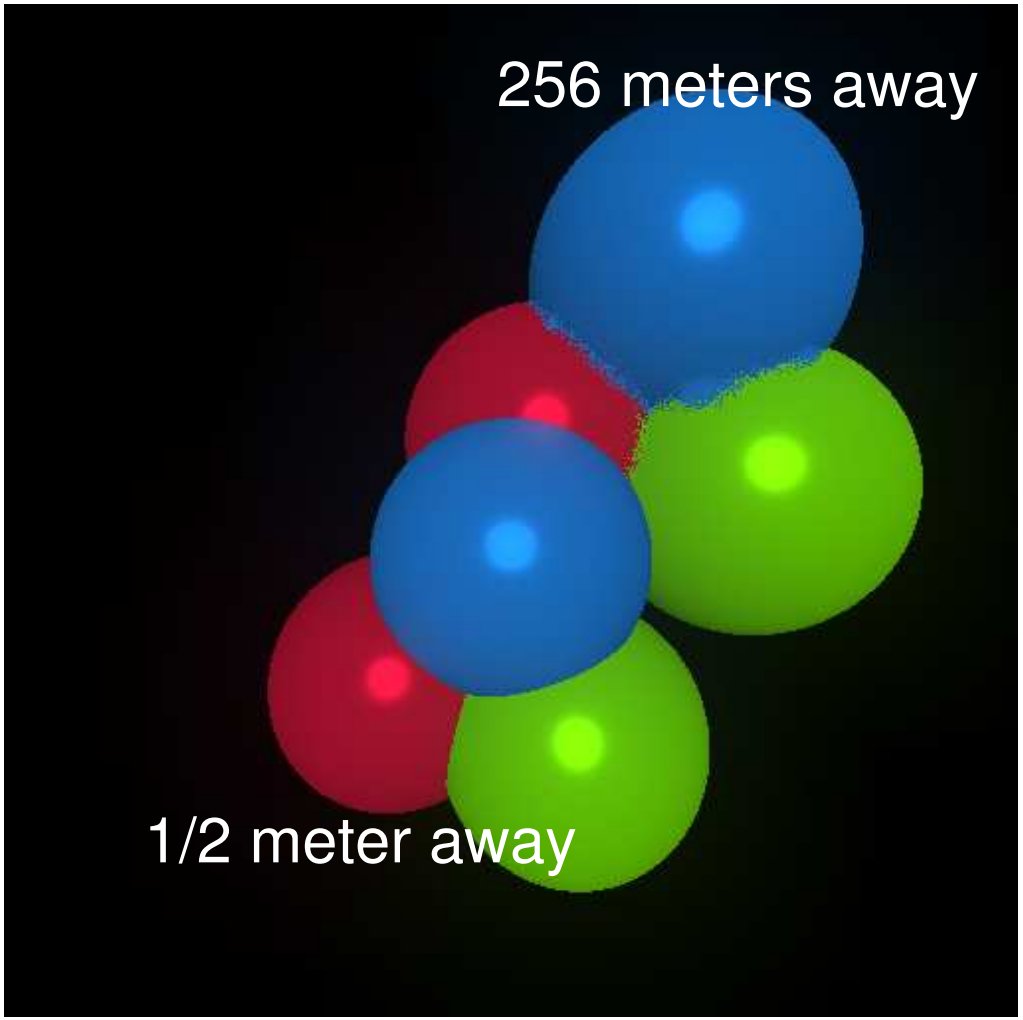
Default Depth Buffer: Confusing Surprises



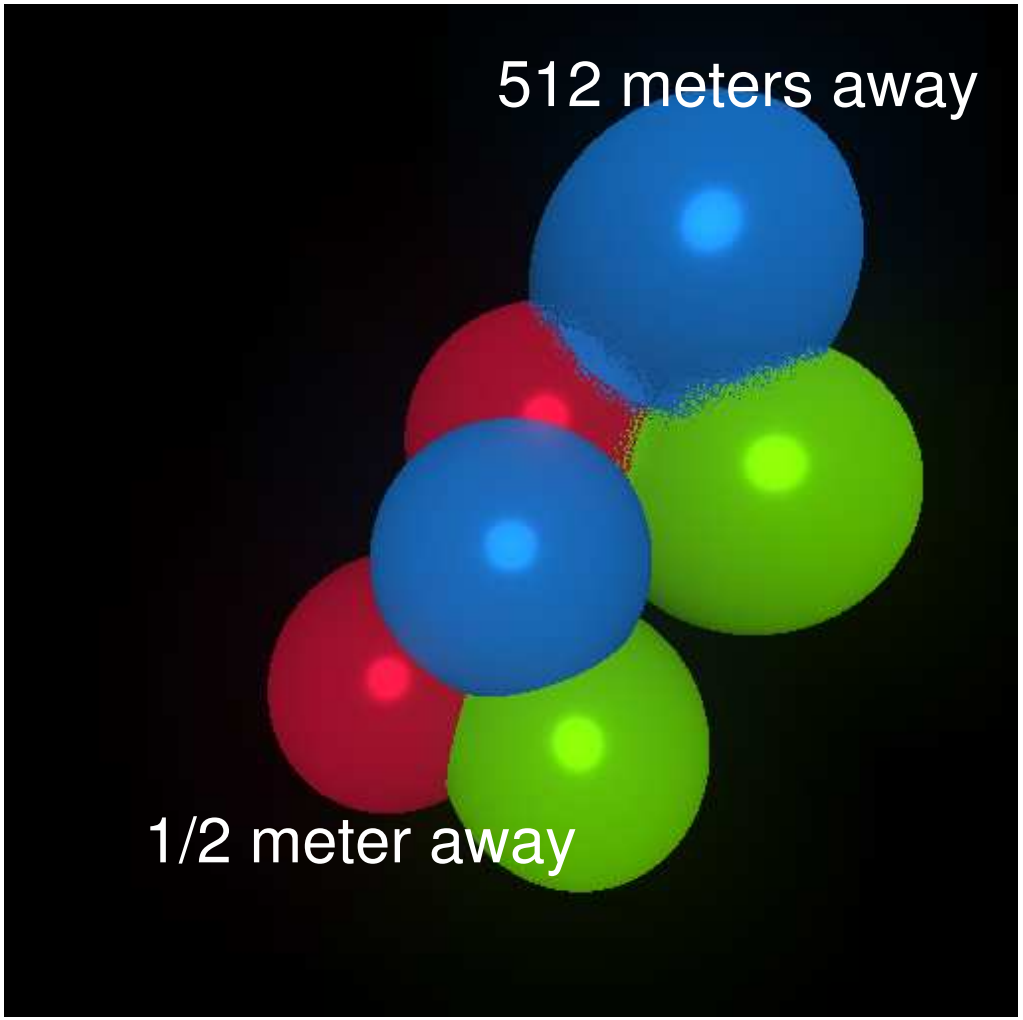
Default Depth Buffer: Confusing Surprises



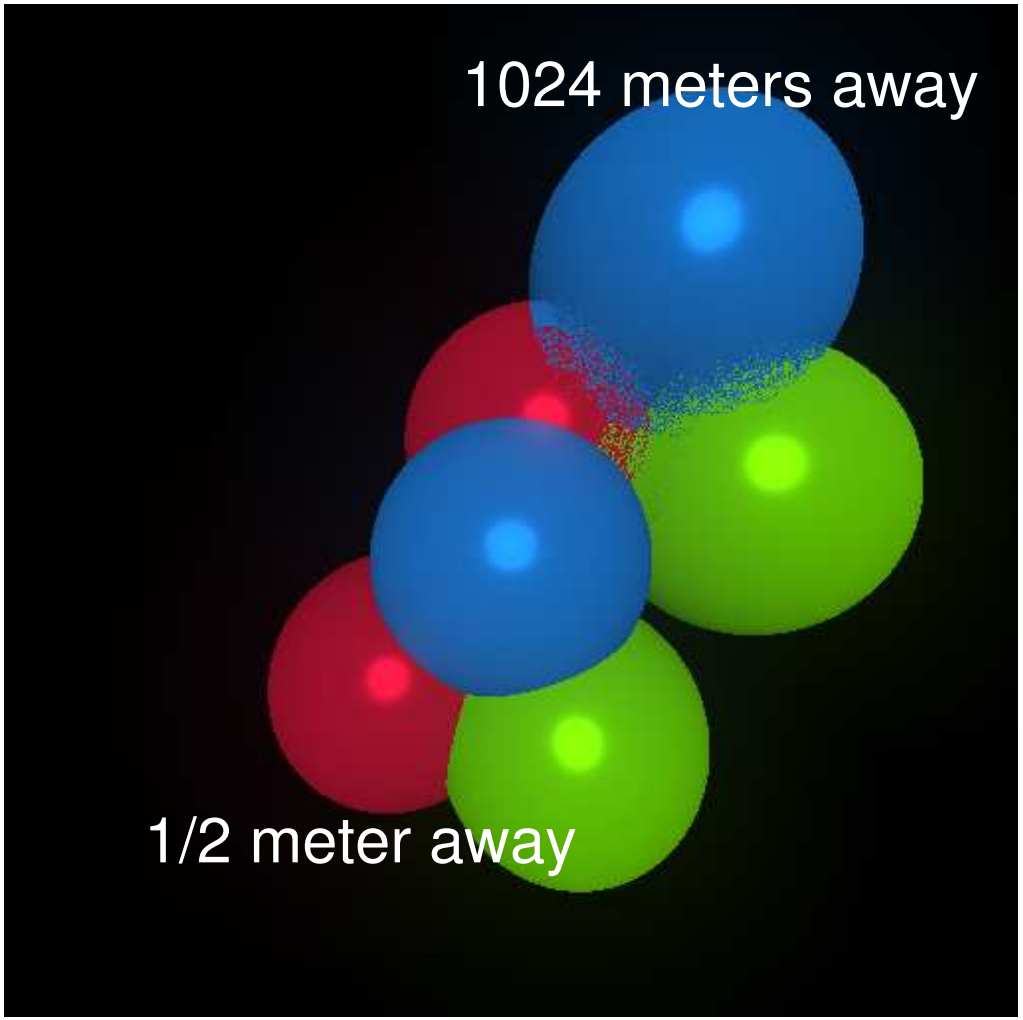
Default Depth Buffer: Confusing Surprises



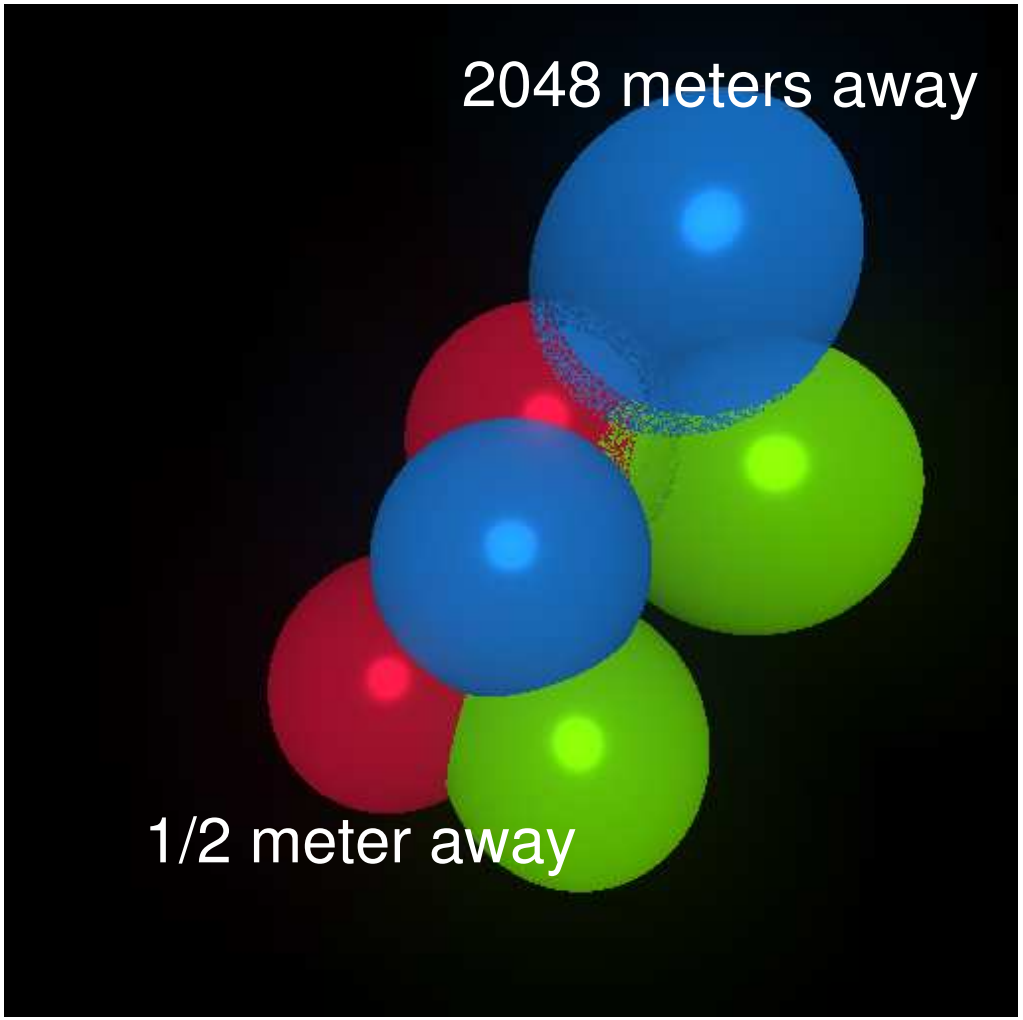
Default Depth Buffer: Confusing Surprises



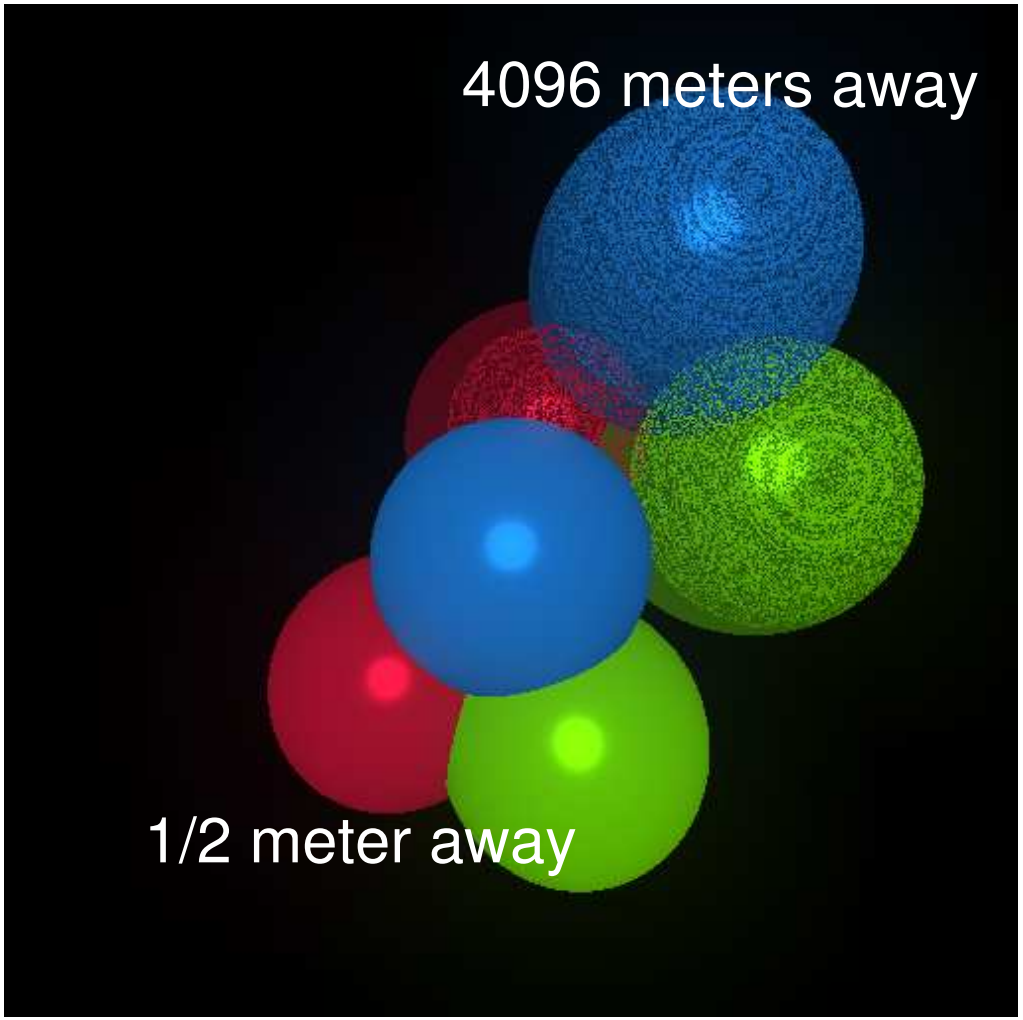
Default Depth Buffer: Confusing Surprises



Default Depth Buffer: Confusing Surprises



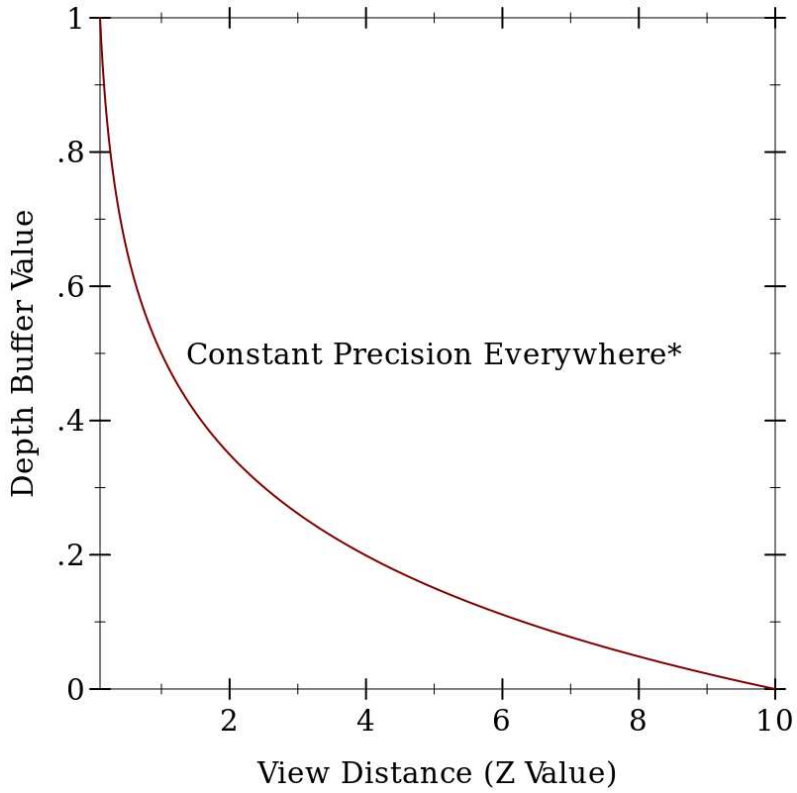
Default Depth Buffer: Confusing Surprises



Logarithmic Depth Buffer

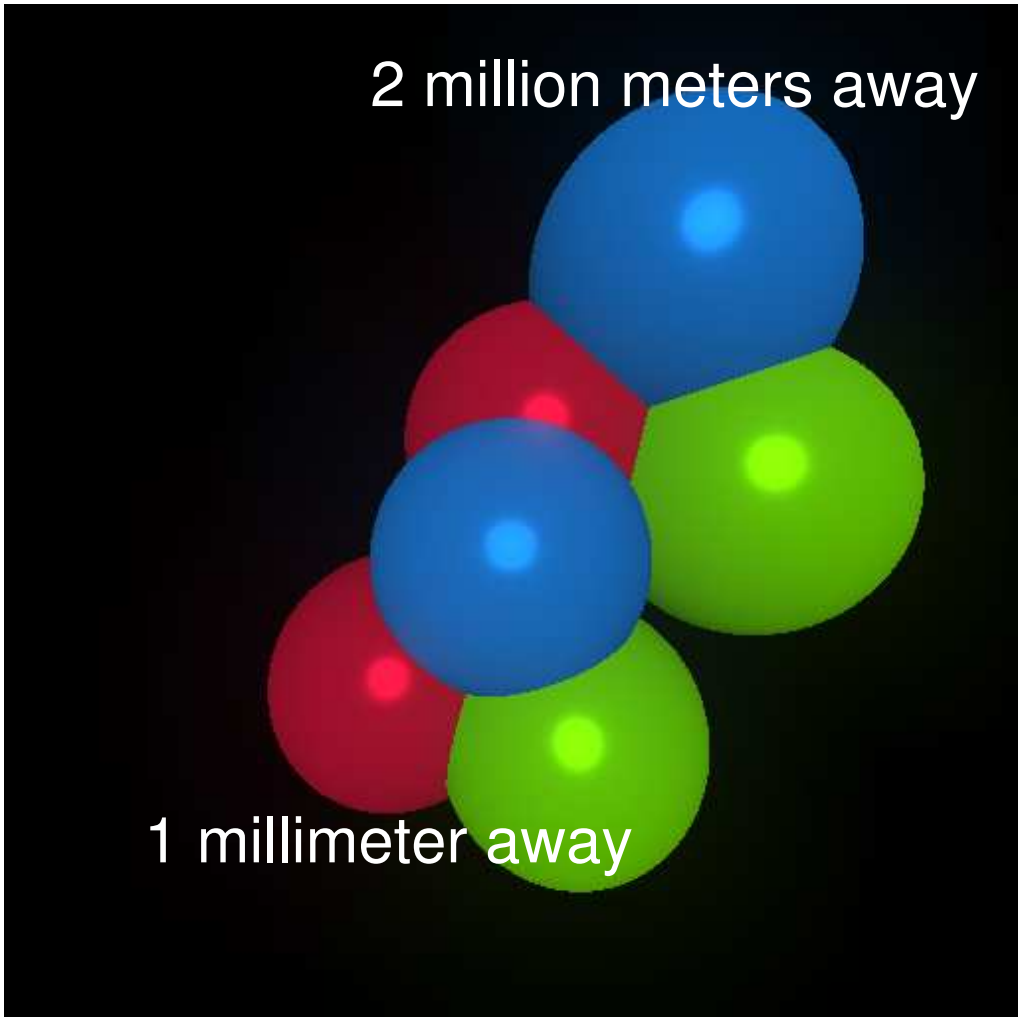
`z-near` = **`0.1`** and **`z-far`** = **`10.0`**

Logarithmic Depth Values



*** http://tulrich.com/geekstuff/log_depth_buffer.txt**

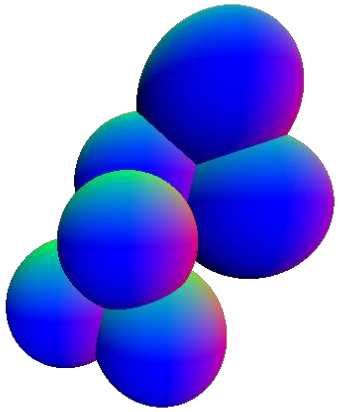
Logarithmic Depth Buffer: Results



Multi-Pass Rendering

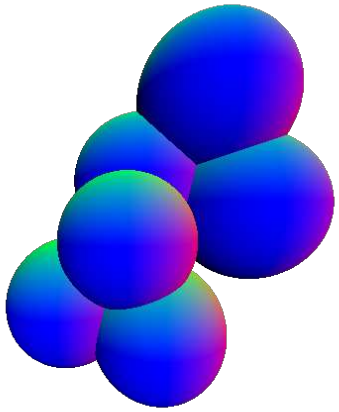
Multi-Pass Rendering

Material: draw shape normals

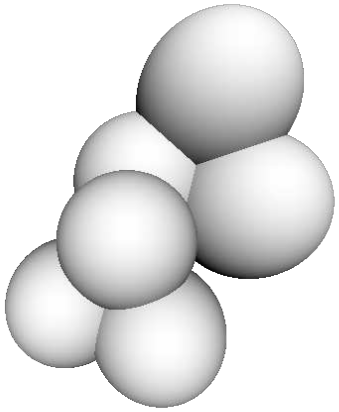


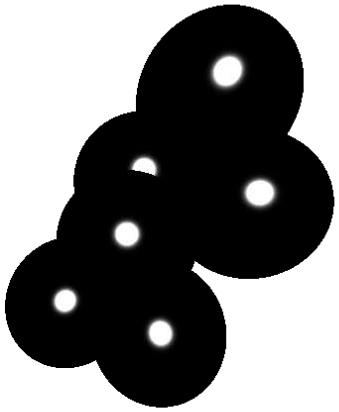
Multi-Pass Rendering

Material: draw shape normals



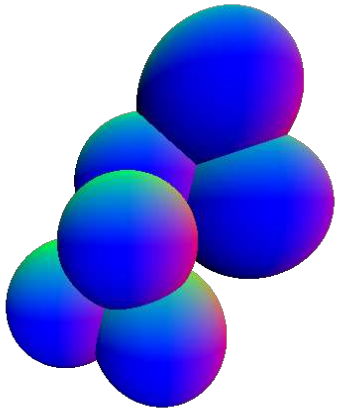
Lighting: draw illumination (diffuse and specular)



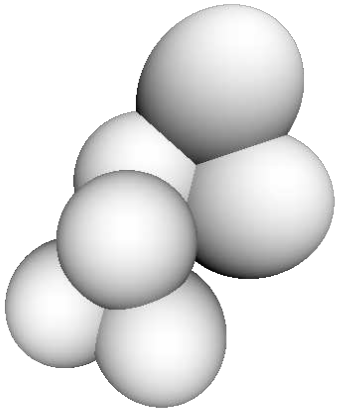


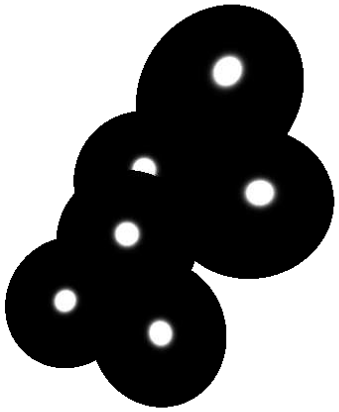
Multi-Pass Rendering

Material: draw shape normals

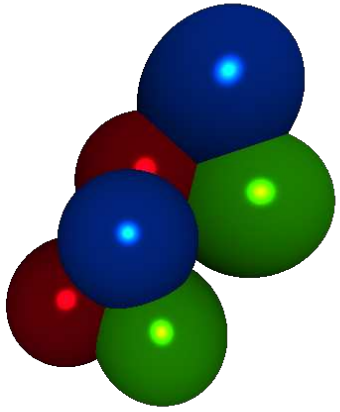


Lighting: draw illumination (diffuse and specular)



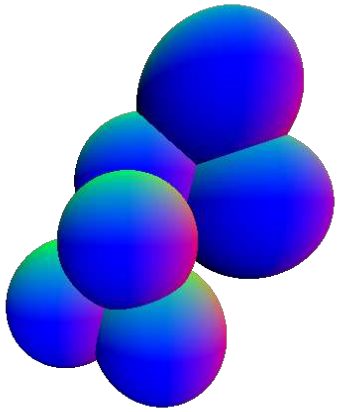


Color: draw shape colors

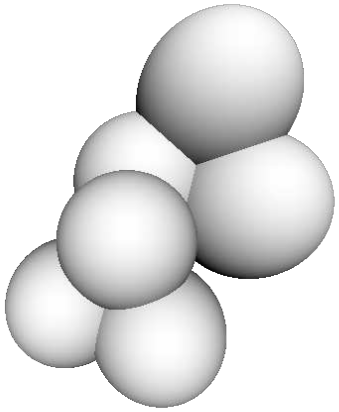


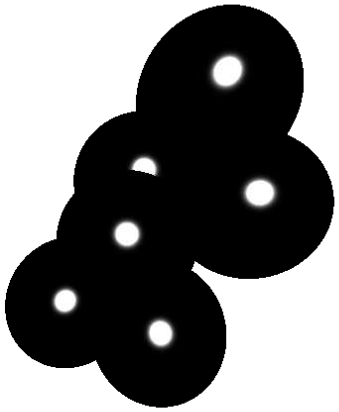
Multi-Pass Rendering

Material: draw shape normals

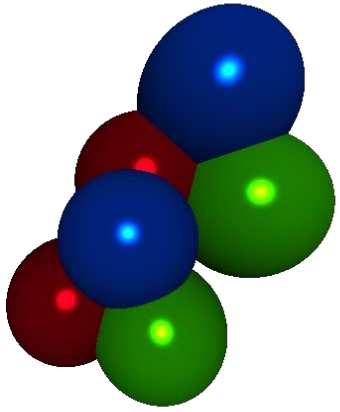


Lighting: draw illumination (diffuse and specular)

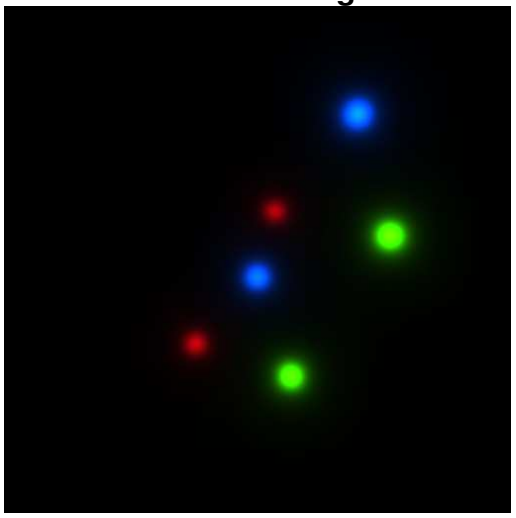




Color: draw shape colors

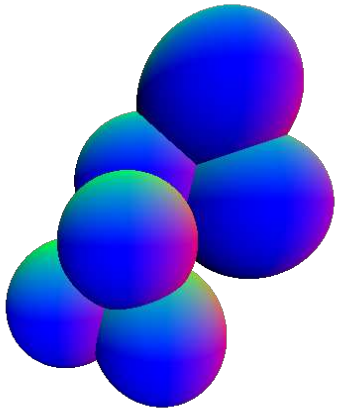


Bloom: extract overbright and blur

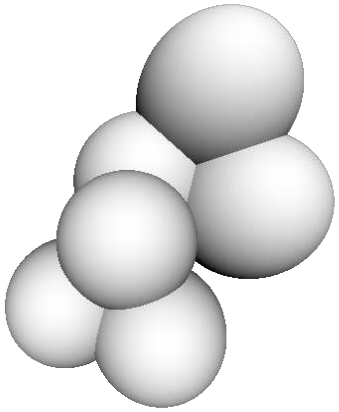


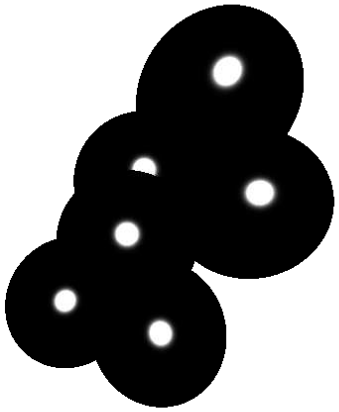
Multi-Pass Rendering

Material: draw shape normals

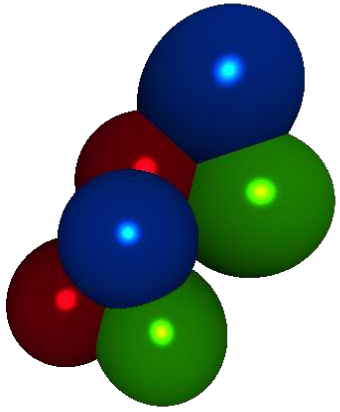


Lighting: draw illumination (diffuse and specular)

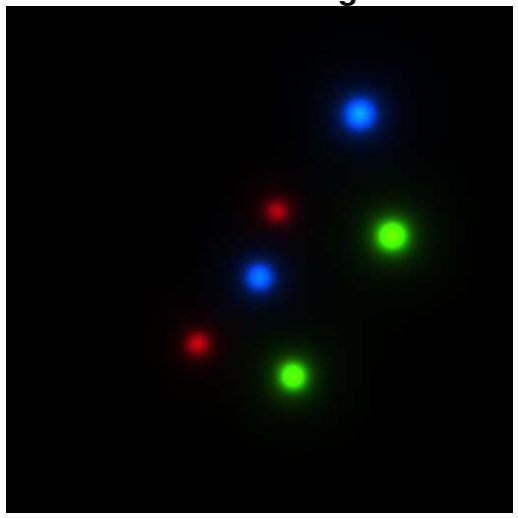




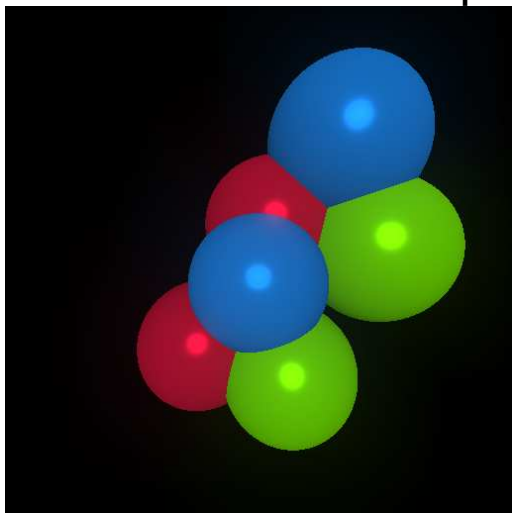
Color: draw shape colors



Bloom: extract overbright and blur

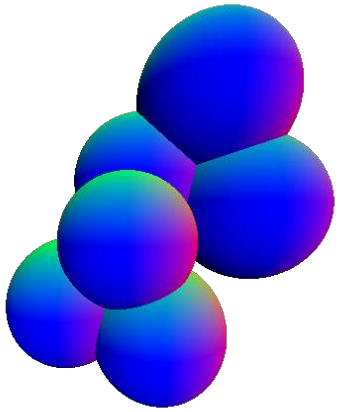


Final: combine and tone-map

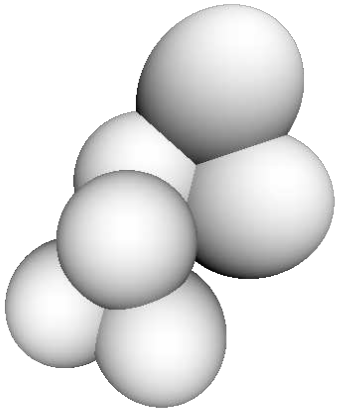


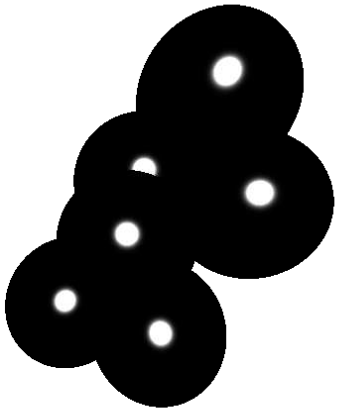
Multi-Pass Rendering

Material: draw shape normals

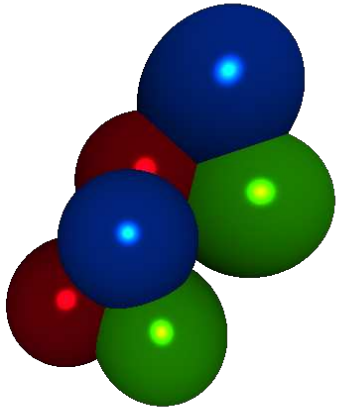


Lighting: draw illumination (diffuse and specular)

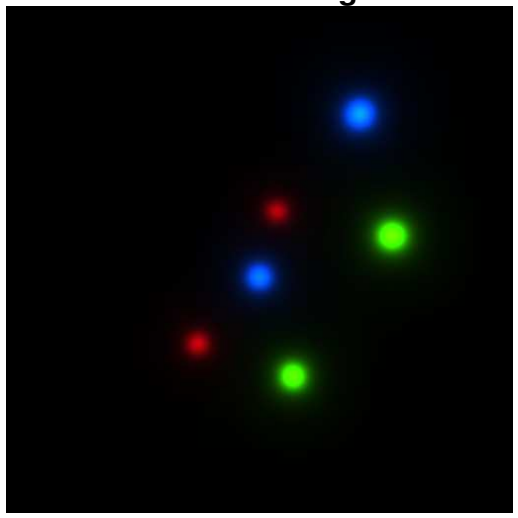




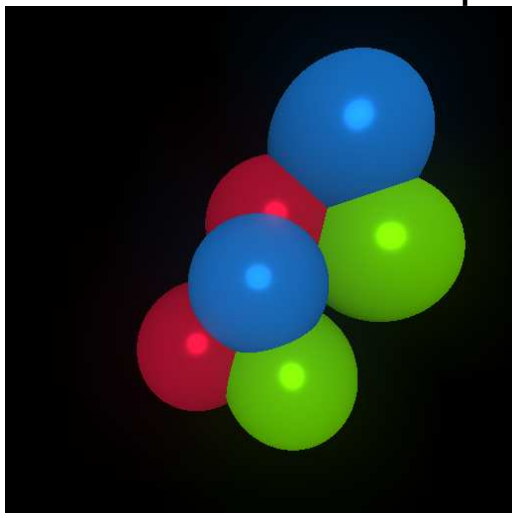
Color: draw shape colors



Bloom: extract overbright and blur



Final: combine and tone-map



O(m+n) in the number of shapes and lights

Best Minimizer of Confusing Surprises

Best Minimizer of Confusing Surprises

It's purely functional!

Best Minimizer of Confusing Surprises

It's purely functional!

- Don’t need to worry about the state of the scene or engine

Best Minimizer of Confusing Surprises

It's purely functional!

- Don’t need to worry about the state of the scene or engine
- Other advantages to full persistence
  - `printf` debugging is crazy useful
  - Store scenes to rewind a game or record a demo

Now What?

- Make **`pict3d`** work on the three major platforms

Now What?

- Make **`pict3d`** work on the three major platforms
- Make **`pict3d`** work in untyped Racket

Now What?

- Make **`pict3d`** work on the three major platforms
- Make **`pict3d`** work in untyped Racket
- Game canvases and **`big-bang-3d`** (very close)

Now What?

- Make **`pict3d`** work on the three major platforms
- Make **`pict3d`** work in untyped Racket
- Game canvases and **`big-bang-3d`** (very close)
- Other shapes and shape combinators, textures, user GPU programs

Now What?

- Make **`pict3d`** work on the three major platforms
- Make **`pict3d`** work in untyped Racket
- Game canvases and **`big-bang-3d`** (very close)
- Other shapes and shape combinators, textures, user GPU programs
- More modern techniques: shadow mapping, screen-space ambient occlusion, dynamic occlusion culling

Now What?

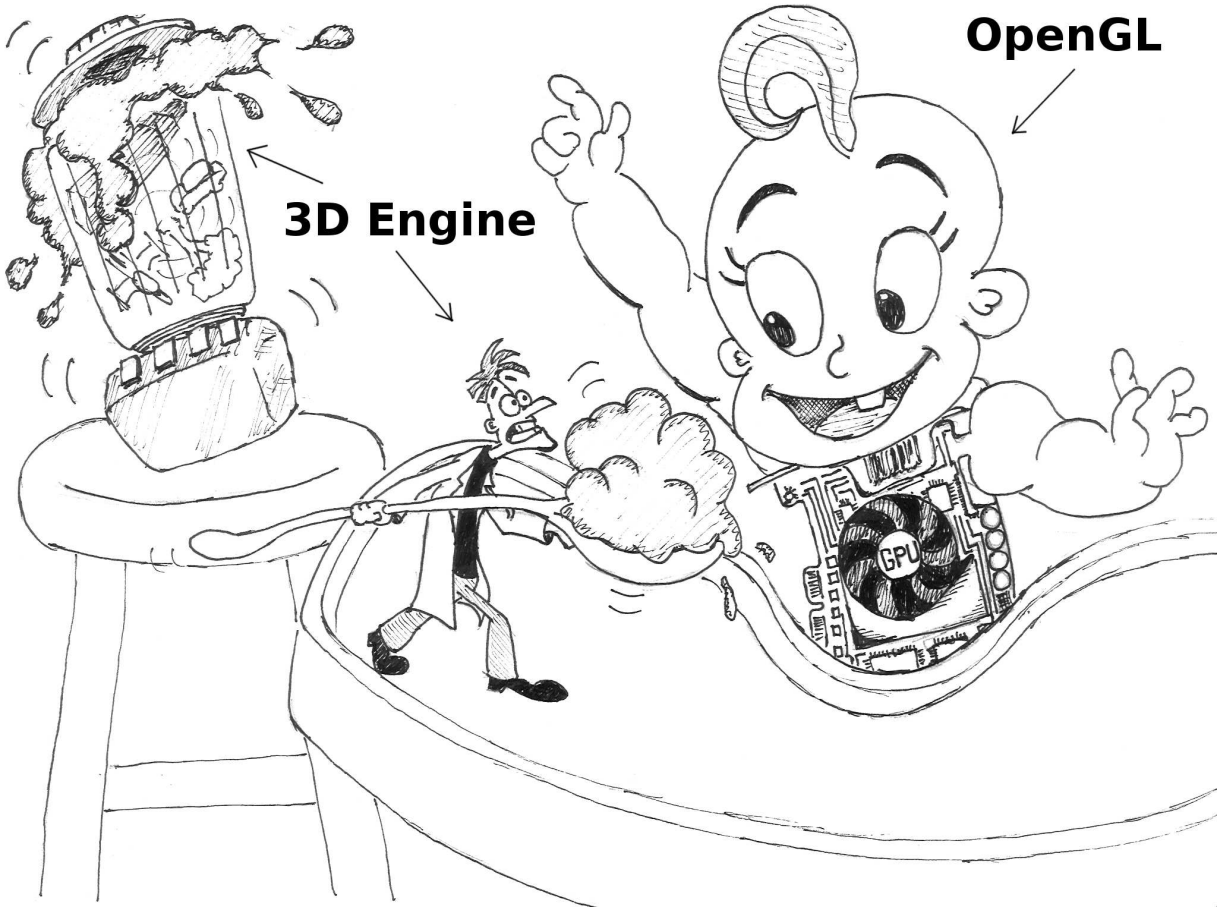
- Make **`pict3d`** work on the three major platforms
- Make **`pict3d`** work in untyped Racket
- Game canvases and **`big-bang-3d`** (very close)
- Other shapes and shape combinators, textures, user GPU programs
- More modern techniques: shadow mapping, screen-space ambient occlusion, dynamic occlusion culling
- Cairo rendering (for **`plot3d`** output)

Now What?

- Make **`pict3d`** work on the three major platforms
- Make **`pict3d`** work in untyped Racket
- Game canvases and **`big-bang-3d`** (very close)
- Other shapes and shape combinators, textures, user GPU programs
- More modern techniques: shadow mapping, screen-space ambient occlusion, dynamic occlusion culling
- Cairo rendering (for **`plot3d`** output)
- Faster faster faster faster faster

Install It Today!

Let **`pict3d`** spoon-feed the giant baby for you




```
raco pkg install pict3d
```