



# Advanced Game Engine

Sean Lilley & Ian Lilley  
Advisors: Norman Badler  
Aline Normoyle

Senior Project Poster Day 2013 – Department of Computer and Information Science – University of Pennsylvania

## Abstract

We created a game engine that takes advantage of OpenGL and modern graphics hardware to show off some of the newest real-time rendering techniques for games. We also added physics, sound, AI, GUI, and more.

## Rendering

Cascaded Shadows

Reflections

Skeletal Animation

Transparency

Parallax Mapping

Decals

Ambient Occlusion

GPU Culling

## Physics

(Bullet Physics Integration)

Rigid Bodies

Terrain

Soft Bodies

Vehicles

Cloth

Characters

## Other

Audio

Scene Loading

Simple AI

User Interface

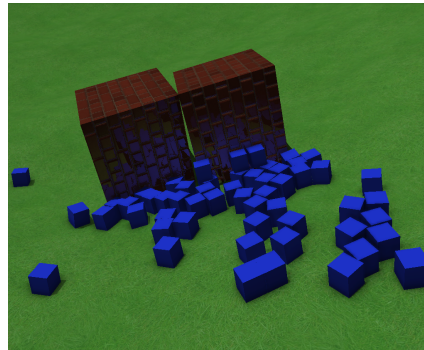
## Technique

We implemented a deferred rendering pipeline which lets us do a variety of screen space effects. Each pixel stores:

|                    |                    |                |              |         |
|--------------------|--------------------|----------------|--------------|---------|
| <b>D24S8</b>       | Depth              |                |              | Stencil |
| <b>R10G10B10A2</b> | Color R            | Color G        | Color B      | Unused  |
| <b>R10G10B10A2</b> | Normal X           | Normal Y       | Normal Z     | Unused  |
| <b>R8G8B8A8</b>    | Specular Intensity | Specular Power | Reflectivity | Unused  |

### Screen-Space Reflections

Shoot ray from pixel and determine intersection with the depth buffer



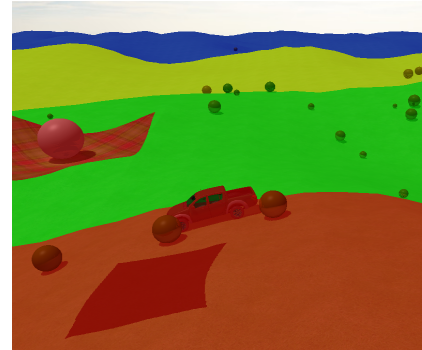
### Ambient Occlusion

For each pixel, determine how many nearby pixels overlap



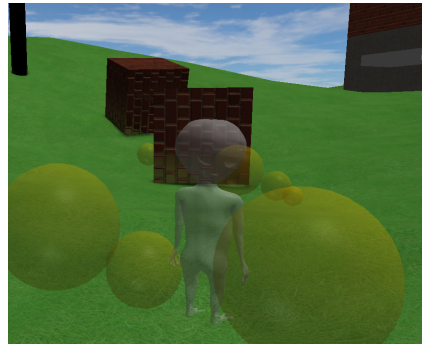
### Cascaded Shadow Maps

Adaptively decrease shadow detail based on distance from camera



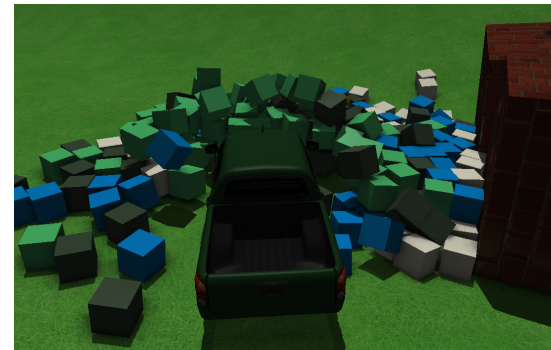
### Order Ind. Transparency

Use per-pixel linked list to combine and shade transparent fragments



## Results

We achieved over 60 fps for all scenes at 1080p. The main bottleneck is screen-space effects.



## Future Work

Global Illumination

Post-processing

GPU Profiling

Particle Effects

Level Editor

Scripting

And finally make a working game!