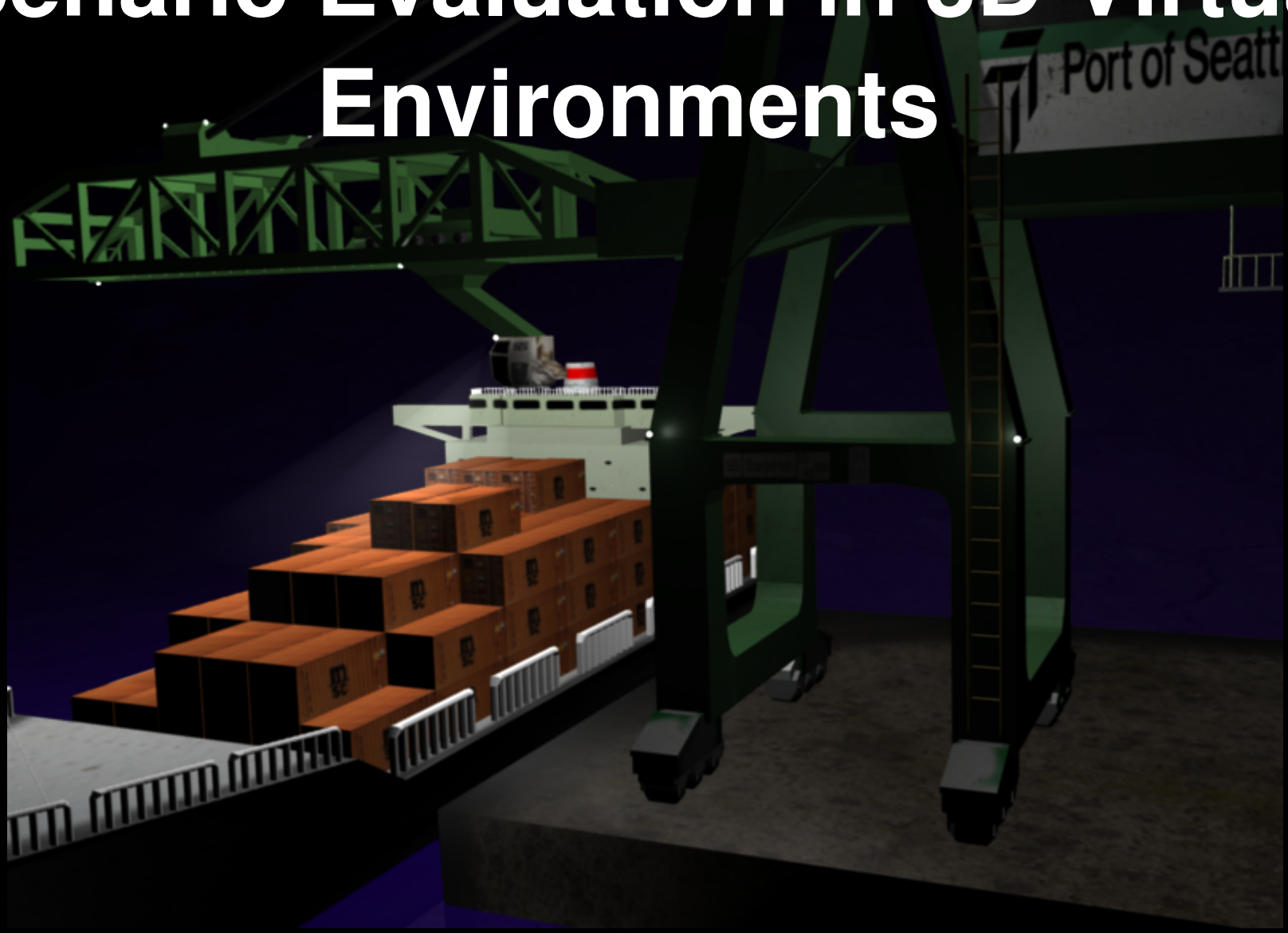
Port Locale Modeling and Scenario Evaluation in 3D Virtual Environments



Presentation by: Damian Johnson
Committee: Larry Holder, Diane Cook, and Bob Lewis

Overview

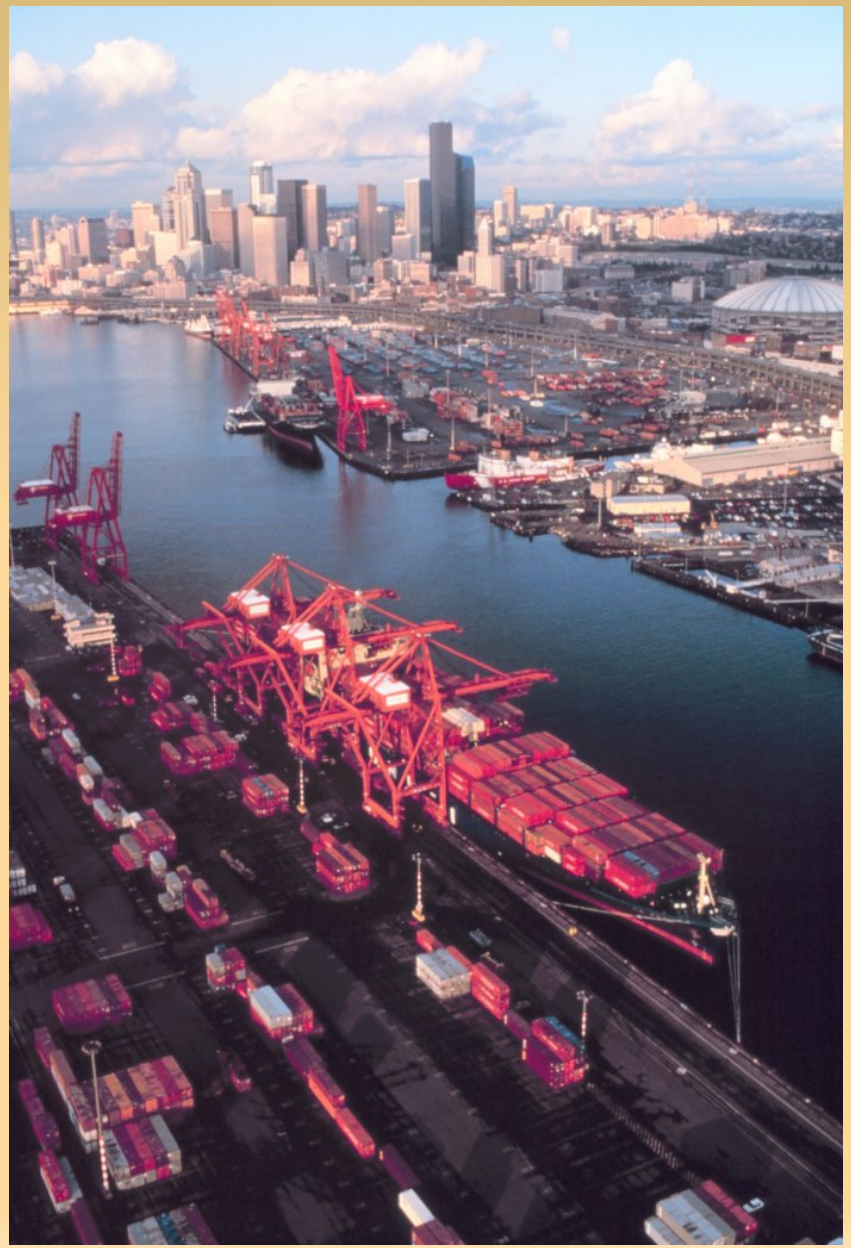
1. **Introduction / Motivation**
2. Game Development Pipeline
3. Results
4. Conclusions / Future Work

Benefits of Computer Simulations

- Avoids Interruptions
  - Downtime can be prohibitively costly
  - Interruptions in critical infrastructure can have widespread repercussions
- Computational Analysis
  - Opens potential for optimization
  - Machine learning to predict complicated behavior
- Fictional Scenarios
  - Prohibitive situations (dangerous, destructive, or costly)
- Visualization
  - Provides an easy way of understanding vast wealth of information

What are we trying to simulate?

- Port of Seattle
  - Satellite imagery
  - Aerial & high definition photos
- Scenario concerns the smuggling of radiological devices
  - i.e. nuclear armaments or dirty bombs



What are we trying to do?

- Simulation in a FPS engine that reflects a real world locale
- Torque Game Engine
  - cross platform
  - proprietary with engine source
  - substantial community support

Why this approach?

- Easier Simulation Development
  - Provides a decently mature framework to work in
- Basic Physics Emulation
  - Sense of space, gravity, air resistance, etc
- Real-Time Visualization
  - Comes for free! Just play the game.
  - Helpful for debugging and making results compelling for lay audience
  - White box model vital for mission critical decisions

What questions are going to be answered in this research?

- How effective is a game engine, particularly Torque, for simulation purposes?
- What is the process for making a simulation and what issues arise when doing so?
- How feasible (in terms of persons, experience, and time) is simulation development with game engines?
- How large of a location can we simulate and at what level of complexity in terms of activities?
- What are the trade offs between realism and performance in simulating a port environment?

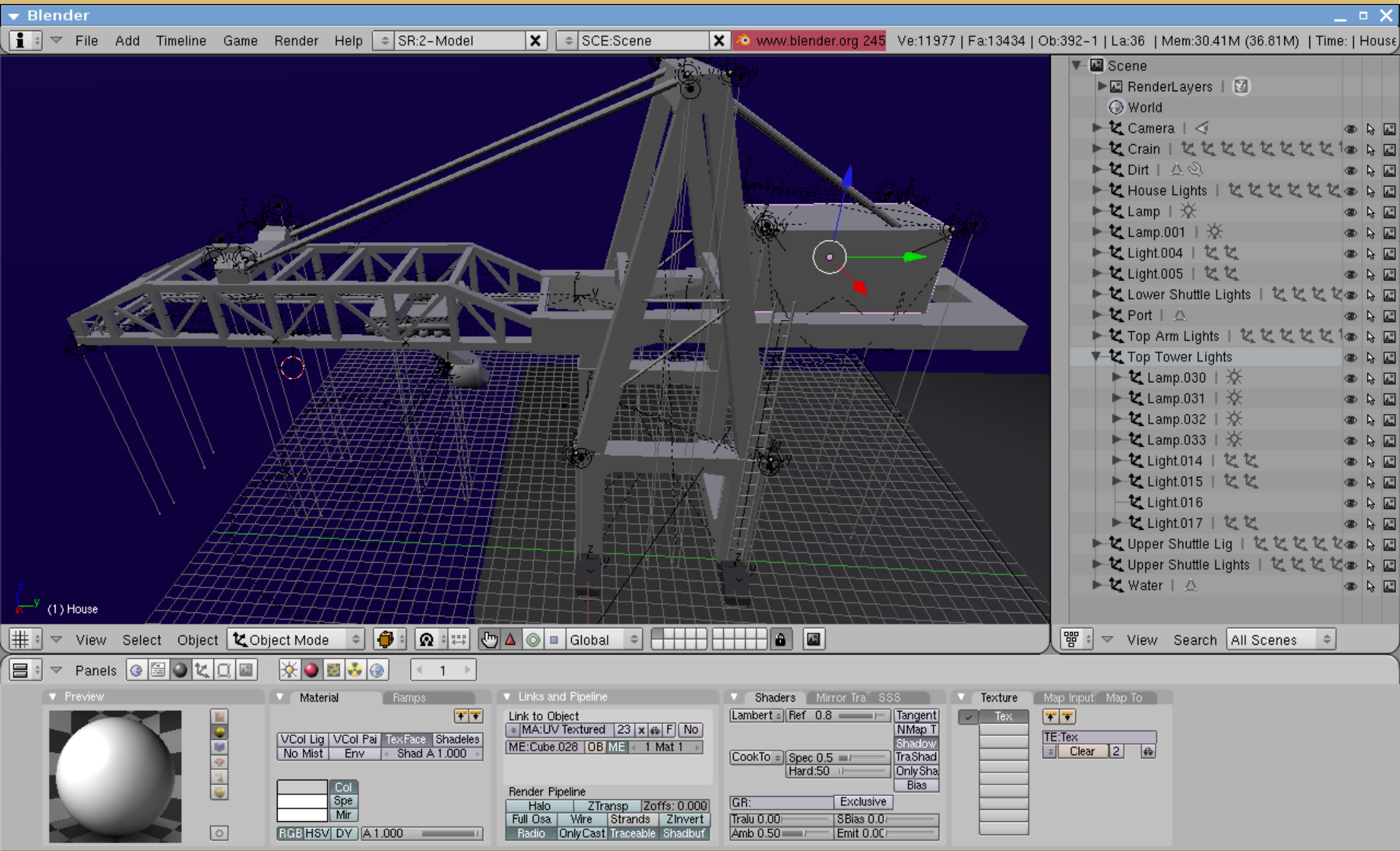
Overview

1. Introduction / Motivation
2. **Game Development Pipeline**
3. Results
4. Conclusions / Future Work

Development Steps

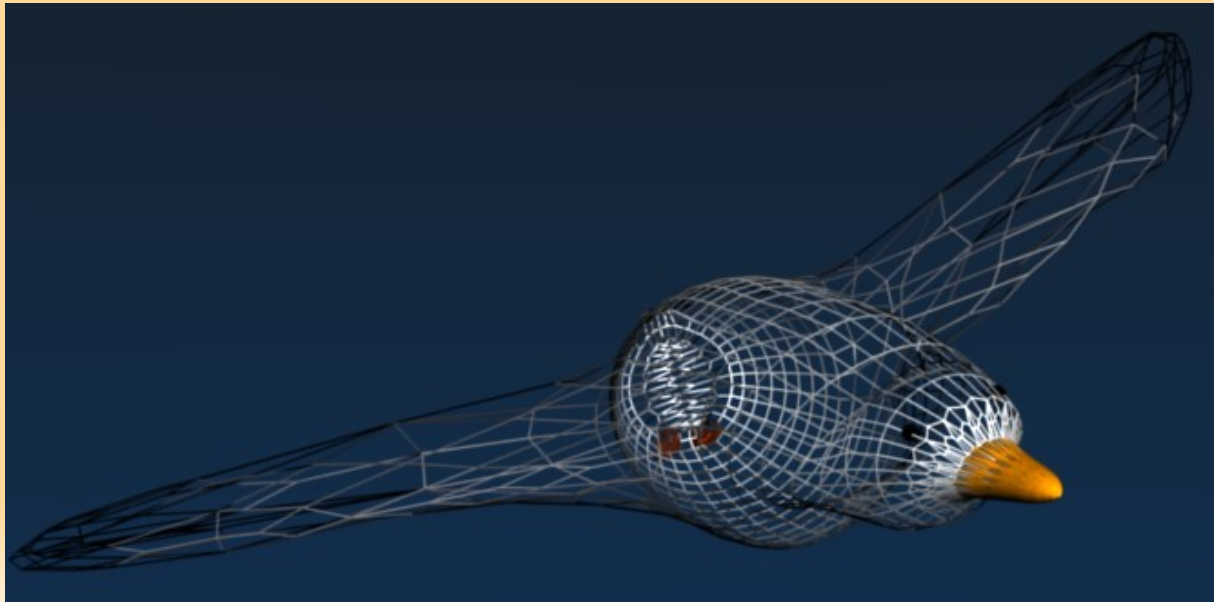
- The creation of each interactive component included three steps:
  - Modeling
  - DTS Export
  - Scripting

Modeling Environment



Modeling Steps

- Find detailed photos of the subject
- Decompose into basic geometry
- Vertex deformation (sculpting parts)
- Organize into logical hierarchy
- Scale to proper proportions
- Animations
- Texturing



DTS Export

- Dynamix Threespace Shape (DTS) conversion provided by Blender plugin
- Issues:
  - Error prone
  - Substantial fidelity loss, stripping:
    - Lighting
    - Subsurfacing
    - Multi-textured faces
    - Diminished texturing

Demonstration of Fidelity Loss

Demonstration of Fidelity Loss

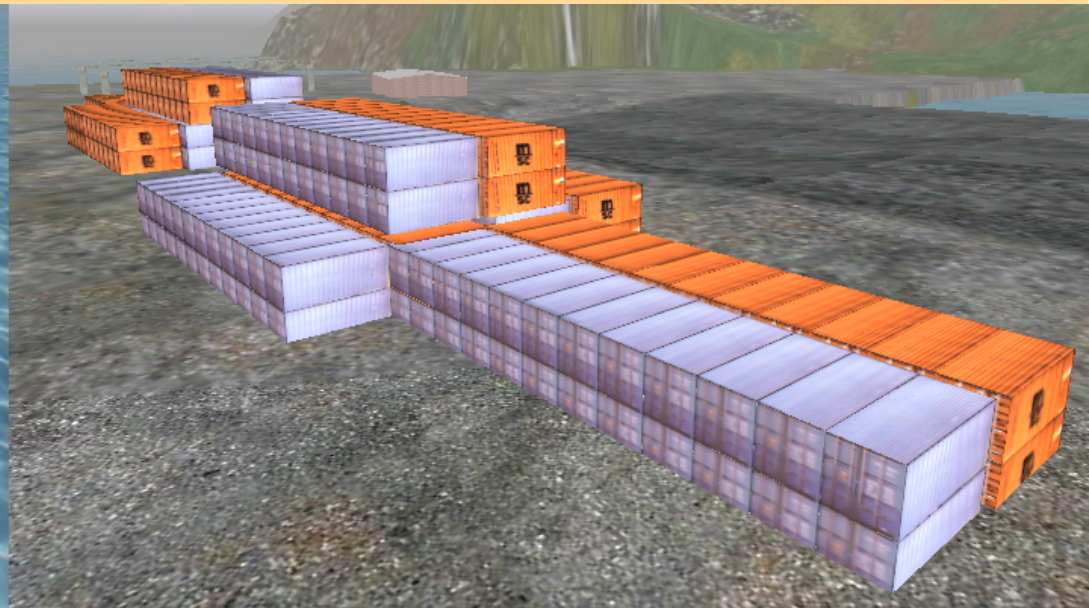
Scripting

- Conventionally included in game engines to:
  - Ease development
  - Decouple scene logic from engine

```
1  /*
2   *  Mobile cargo container. This generally behaves as a static shape, performing scheduled
3   *  affine transformations to move. When dropped this is replaced with an Item so gravity
4   *  can act on it.
5   */
6  datablock StaticShapeData(Cargo2S) {
7     category = "Items";
8     shapeFile = "~/data/shapes/cargo/cargo2.dts";
9  };
10
11 // Converts to an item to allow gravity to operate on it
12 function Cargo2S::drop(%this, %obj) {
13    %currentName = %obj.getName();
14    %currentPosition = %obj.getPosition();
15    %currentRotation = getWords(%obj.getTransform(), 3, 5);
16    %obj.delete();
```

Interactive Objects: Cargo

- Scene has two types:
  - Static – Unmoving cargo that litters the port
  - Dynamic – Mobile and possibly contains contraband



Interactive Objects: Cargo Ship

Interactive Objects: Freight Truck

Port Scenario Activity

- Initial scene includes a boat loaded with cargo and a crane with a truck under it.
  - Boat comes into the port and stops under the crane. Cargo may or may not contain emitters.
  - Cargo is transferred off one at a time to the waiting trucks.
  - Whenever cargo meets a truck it is driven away and a new truck comes in to take its place.

Demo!

Overview

1. Introduction / Motivation
2. Game Development Pipeline
3. **Results**
4. Conclusions / Future Work

Scalability

- Computational scalability - resources required to render and simulate, largely dominated by:
  - Model rendering
  - Particle physics
- Development scalability - scene complexity as objects, interactions, and details rise

Model Statistics

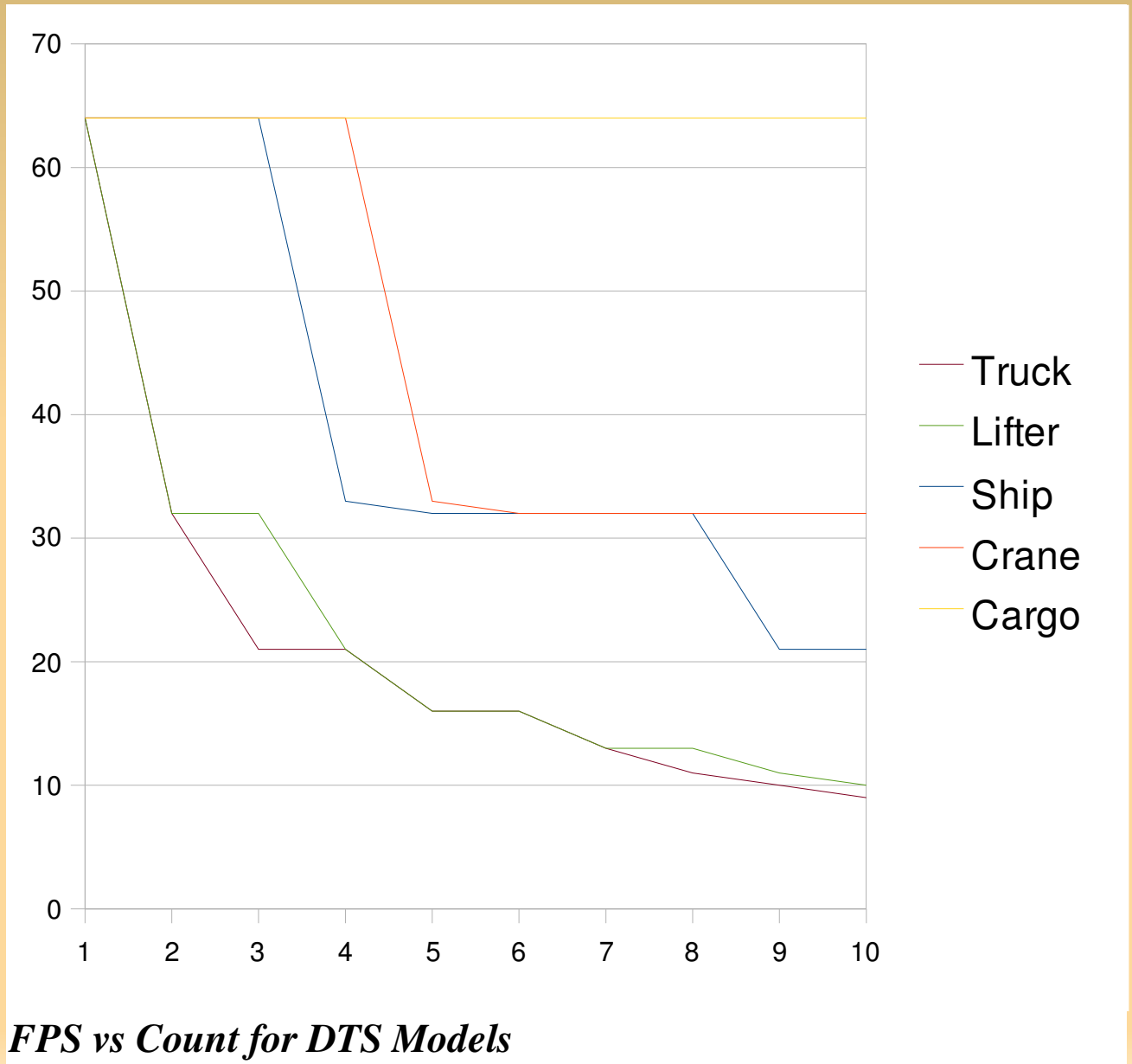
	Vertices	Faces	Objects	Model Size	Textures Size
Truck	27334	31930	163	3.7 MB	14.4 KB
Lifter	24554	27032	205	3.3 MB	530 KB
Ship	10219	11731	419	2.2 MB	3.2 MB
Crane	7302	8464	307	1.7 MB	3.2 MB
Seagull	2198	2206	3	225 KB	491 KB
Cargo	8	6	1	122 KB	104 KB

FPS vs Model Count

- Environment: Intel Core 2, Nvidia GeForce 7900 GS
- Measurements via FRAPS

	1	2	3	4	5	6	7	8	9	10	15
Truck	64	32	21	21	16	16	13	11	10	9	6
Lifter	64	32	32	21	16	16	13	13	11	10	6
Ship	64	64	64	33	32	32	32	32	21	21	16
Crane	64	64	64	64	33	32	32	32	32	32	21
Cargo	64	64	64	64	64	64	64	64	64	64	64
Bomb	64	64	64	64	64	64	64	64	64	64	64

FPS vs Model Count

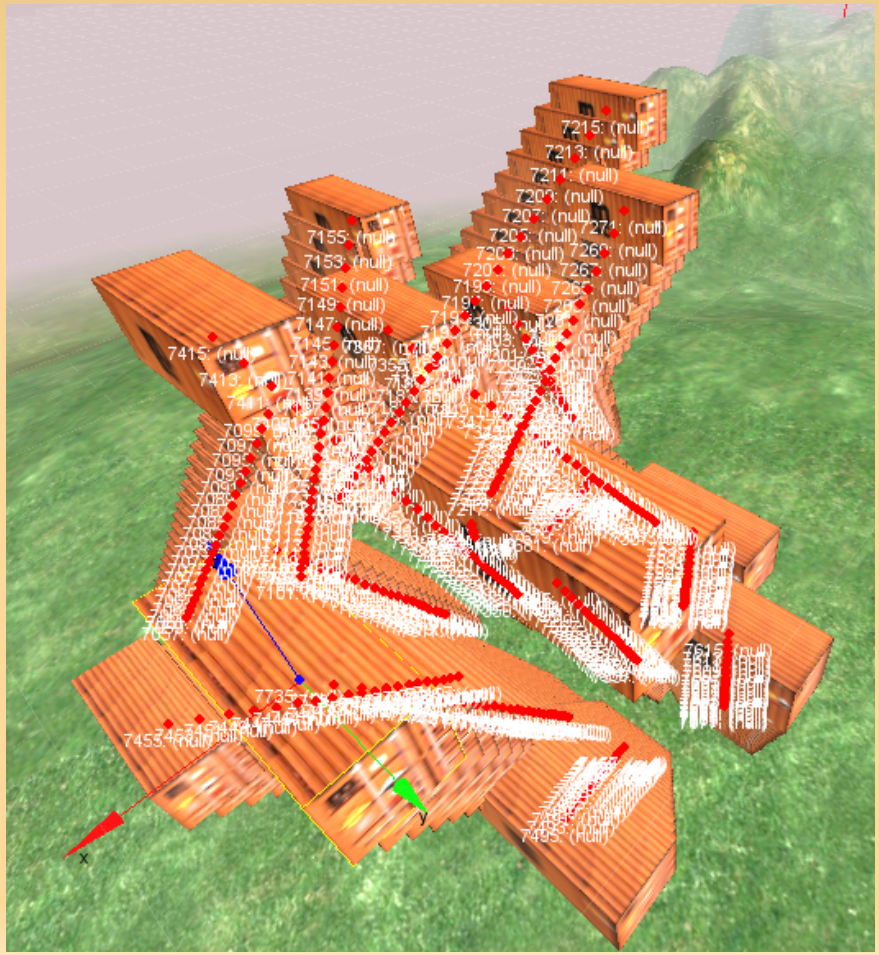


FPS vs Count for DTS Models

Scene Complexity at First Drop

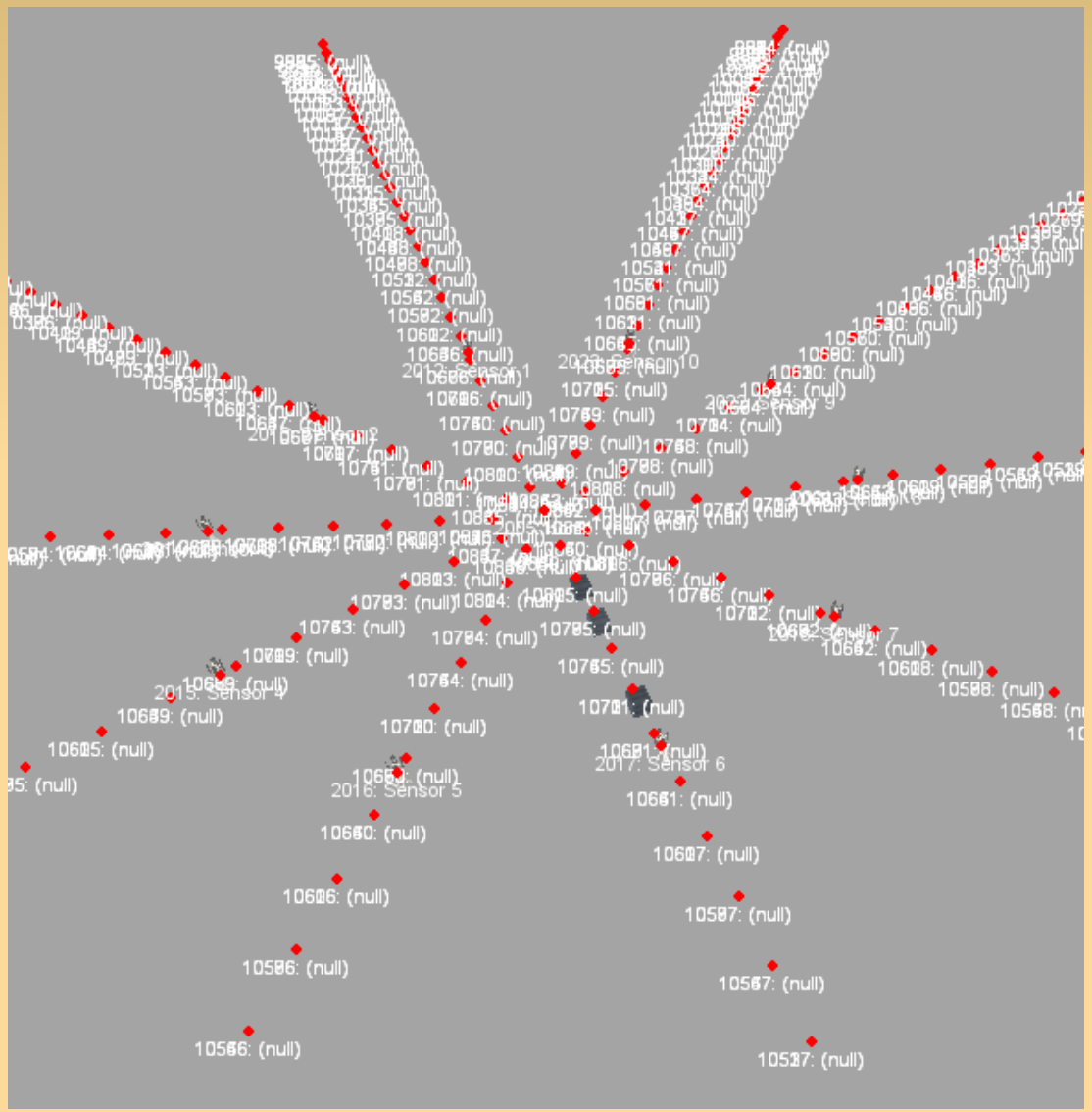
	Truck	Lifter	Ship	Crane	Cargo	Bomb	Sensor
Count in frame rate drop	2	2	4	5	340	93	55
Vertices being rendered	54668	49108	40876	36510	2720	---	---

- Points of Interest:
  - Regardless of model there is a similar vertex count at which performance first drops
  - This cutoff point can be approximated to 32768 vertices (2^{15})



Particle Emission Performance

- Particles have a few attributes:
  - Fire rate
  - Longevity
  - Collisions



Particle Fire Rates

- Testing for cost of particle construction/destruction

Fire rate (ms)	100	50	10	5	3	2
Particles per second	100	200	1000	2000	3333	5000
Frame rate (fps)	64	64	64	32	21	1

Particle Counts

Particle Count	10	100	1000	2000	5000	10000
FPS (1000 p/s)	64	64	21	16	0	0
FPS (200 p/s)	64	64	52	32	8	0
FPS (100 p/s)	64	64	64	32	11	0

- Points of Interest:
  - Loss in performance gradual
  - Fixed fire rates factor into the loss in performance

Particle Collisions

- Fire rate: 10 milliseconds (1000 particles/sec)
- Longevity: 1,000 milliseconds
- Emitter was wrapped in multiple layers of cargo

Collisions per particle	0	1	2	3	4	5
Frame rate (fps)	21	21	21	21	21	21

Runtime Performance Summary

- Models Rendering:
  - Results suggest the primary burden of models are the vertices (rather than textures, objects, etc)
  - Limited detail levels can improve this, but still a limiting factor in simulating a large port
- Particle Simulation:
  - Results indicate both particle counts and, to a lesser degree, fire rate can demand substantial resources (collisions were not noticeable)
- Loss in performance between these stack!

Development (Advantages)

- Game engines provide several desirable attributes, including:
  - Avoids Reinventing the Wheel
  - Obeys Basic Physical Constraints
  - Highly Efficient Rendering
  - Real Time Visuals
  - Scriptable

Development (Disadvantages)

- Realism, particularly in scripting arbitrary interactions quickly becomes time consuming and brittle. Issues include:
  - Limited Fidelity
  - High Abstraction
  - Repurposing
  - Poor Design and Incomplete Functionality
  - Antiquated Engine

Overview

1. Introduction / Motivation
2. Game Development Pipeline
3. Results
4. **Conclusions / Future Work**

Conclusions

- - Despite using an established framework development went slowly
- - Many issues raised are Torque specific, but for most the roots lie in the game industry at large
- - Experimental results show an inability to support large scenes with a decent level of graphic fidelity
- + The simulation demonstrates a working level of detail not possible if done from scratch
- + As both hardware and engine advancements are made the issues raised will likely become a thing of the past

Future Work

- Continuing investigation of game engines for simulations:
  - Detail levels
  - Introduction of animations
  - Point light sources
  - Add to environment (buildings, roads, etc)
- Instance based investigations:
  - How faithful can it be for reflecting the real world?
  - Crossover benefits for other locales?

Conferences & Presentations

- Johnson, Damian, Allen Christiansen, Lawrence Holder. "Game-Based Simulation for the Evaluation of Threat Detection in a Seaport Environment." Entertainment Computing - ICEC 7(2008): 221-224.
- Washington State University Showcase 2009

Related Works

- Sekine, Junko. "A simulation-based approach to trade-off analysis of port security." Winter Simulation Conference 38(2006): 521 - 528.
- Parry, Lucas. "L3DGEWorld 2.3 Input & Output Specifications." 22 Feb 2008 25 Mar 2009 <http://caia.swin.edu.au/reports/080222C/CAIA-TR-080222C.pdf>.
- Sanders, R. L.. "A simulation learning approach to training first responders for radiological emergencies." Summer Computer Simulation Conference (2007):
- Sharma, M.. "Transfer learning in real-time strategy games using hybrid cbr/rl." Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (2007):
- Maher, Mary. "Situated design of virtual worlds using rational agents." Proceedings of the second international conference on Entertainment computing 38(2003): 1 - 9.

Questions?