

PhyreEngine™  
The new hotness

Rich Forster  
SCEI Euro R&D

PHYREENGINE™

# Introduction

- What is it?
- It's a game engine including
  - Modular run time
  - Art pipeline
  - Example game templates
  - Samples and docs

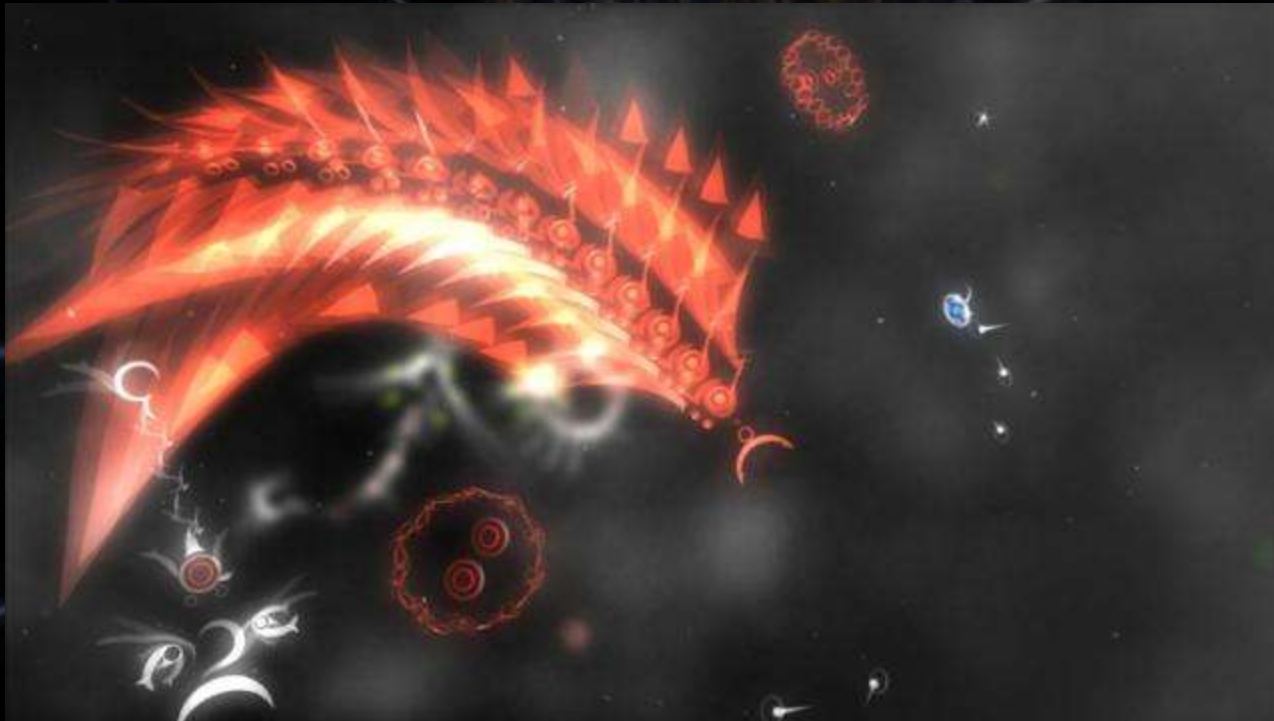
# PhyreEngine™ Titles

- GripShift



# PhyreEngine™ Titles

- fl0w + expansion pack



# PhyreEngine™ Titles

- Colin McRae: DiRT



# Agenda

- Overview
- Technology
- Terrain Engine
- Roadmap



# Overview

PHYREENGINE™

# PhyreEngine™ Genesis

- Built on pre-existing technologies
  - PSSG for core graphics
  - Multi-stream for sound
  - Bullet, Havok and AGEIA for physics
- Added value from R&D projects
  - Also using PSSG for rendering

# Run-time

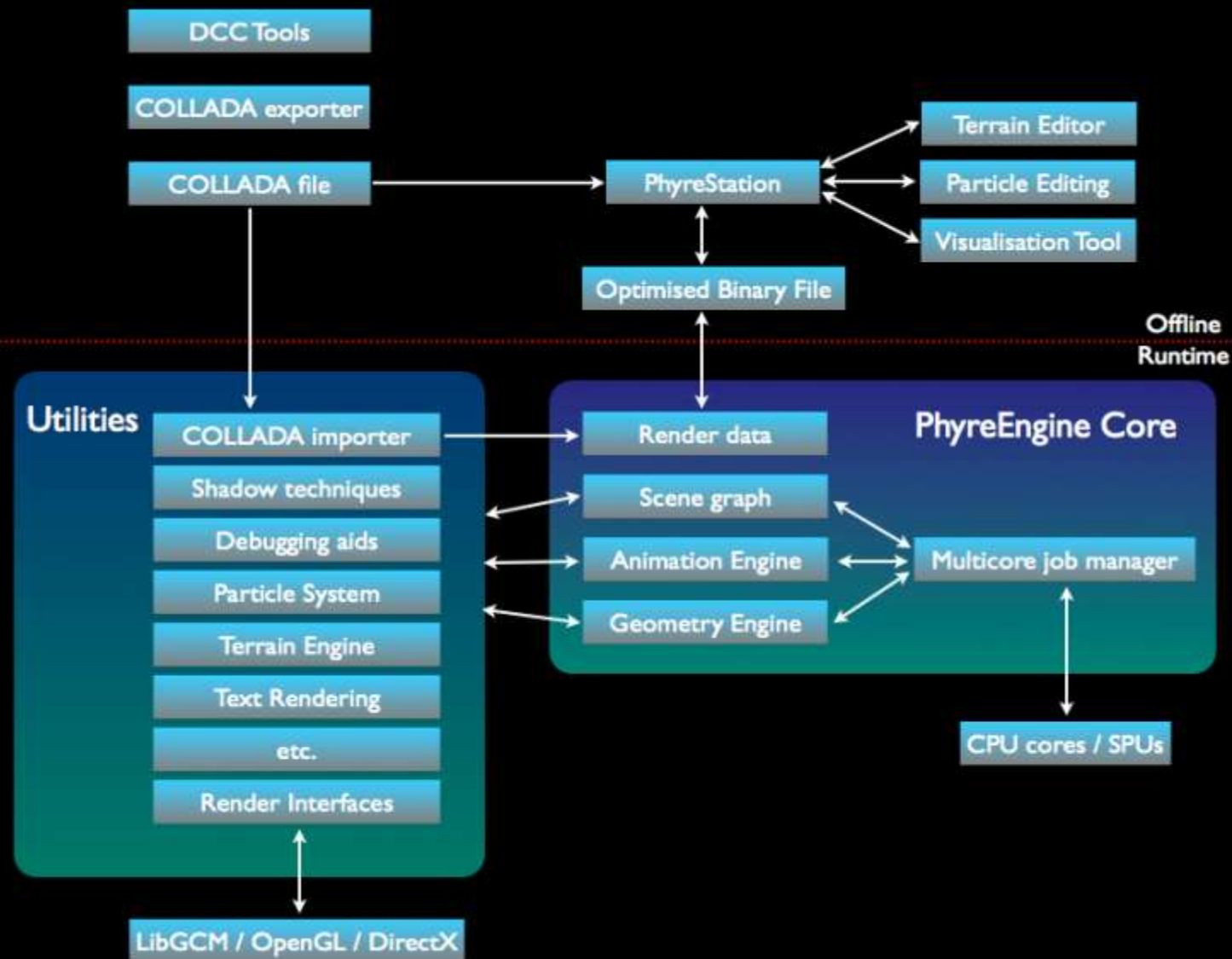
- Cross platform
  - Optimized for multi-core especially PS3
- Modular design
  - Extractable and reusable components

# Provided as source

- Gives control to the user
- Extend and customize
- Transparency
  - See how it works
  - Debugging

# Art pipeline

- Ships with exporters for Maya and Max
  - Supported by PhyreEngine™ team
- Compatible with most COLLADA tools
- PhyreStation asset processing tool
  - Workspace model for assets



# Example game templates

- Self-contained sample projects
  - Include full source and art work



# Samples and Documentation

- 70+ samples
  - Each documented at code level
- Full reference manuals
  - Inline API documentation
- Conference materials
- White papers

# Low-Cost Development Platform

- Enables PC development
  - OpenGL and Direct3D
  - Simple recompile for PLAYSTATION®3
- Run on Debugging Stations
  - With SN's tools
- Artist tools
  - Debug station viewer



Technology

PHYREENGINE™

# SPU or Multi-Thread

- Geometry processing
  - Skinning, morphing, culling, compression
- PhyreEngine™ handles streaming
- Can develop from non-SPU code
- Many already implemented
  - 60 stream, 31 packet
  - Each can be integrated into any other engine

# SPU or Multi-Thread

- Animation
  - Blending
  - Procedural generation
  - Controllable time flow
- Uses frame coherence
  - Persistent data stores playback position
  - Enables prefetch of required animation data
  - Prefetch handled by PhyreEngine™

# SPU or Multi-Thread

- Scene management
  - Graph traversal and update
  - Portal visibility
- Converts a scattered memory pattern
  - Into a format easily processed by an SPU
- SPU implementations of common nodes

# Lighting

- HDR
  - 8 or 16 bit with example HDR resolves
- Shadows
  - Maps (inc. dual paraboloid and parallel split)
  - Volumes – dynamic generation

# Physics

- Integration with existing physics engines
  - Bullet
  - Havok
  - AGEIA
  - Basis for your own

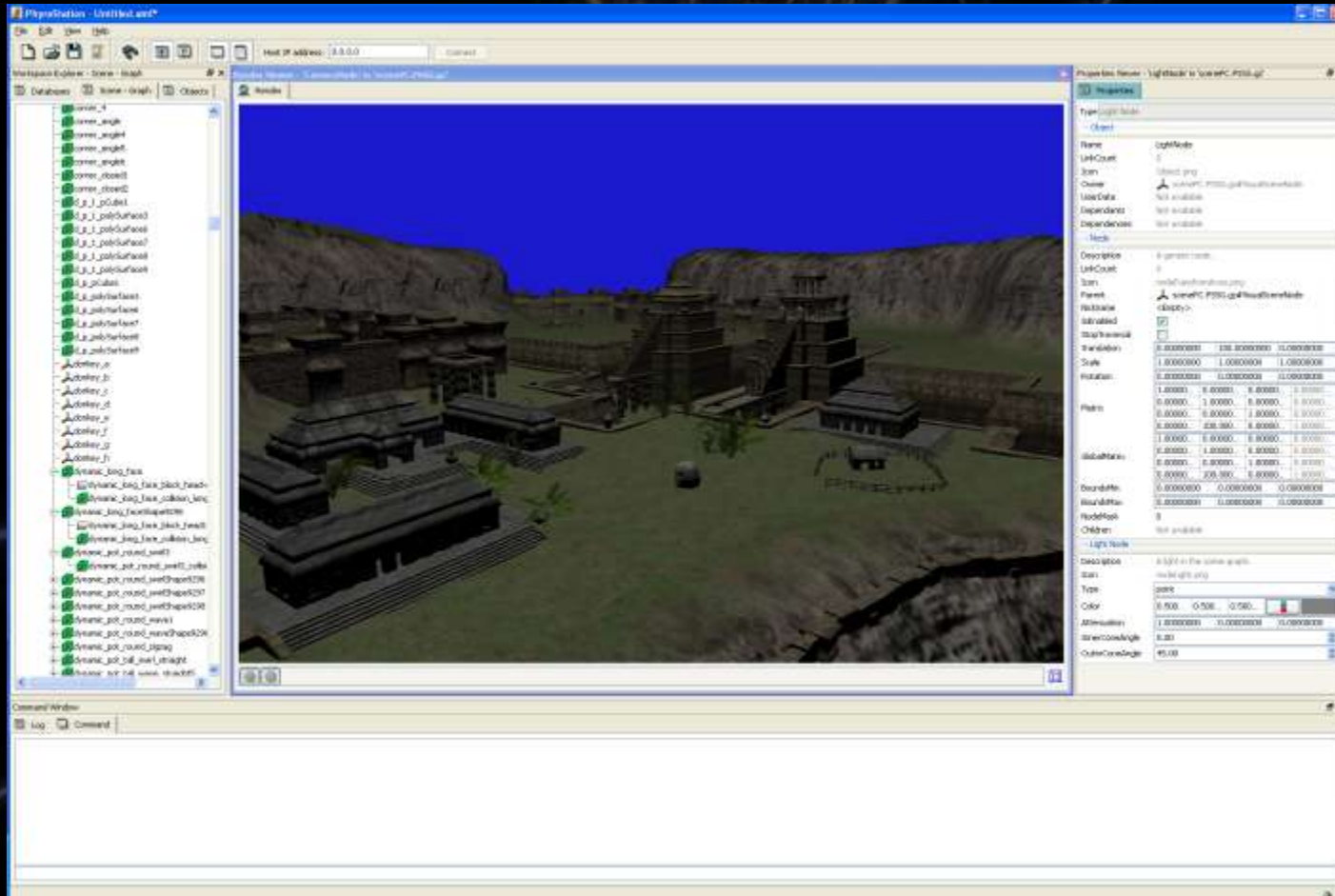
# LOD and Effects

- Level of detail
  - Support for geometry and shaders
- Particle Systems
  - Based on XML description
  - Offline code generation for SPU and PPU/PC
  - Author in PhyreStation

# PhyreStation

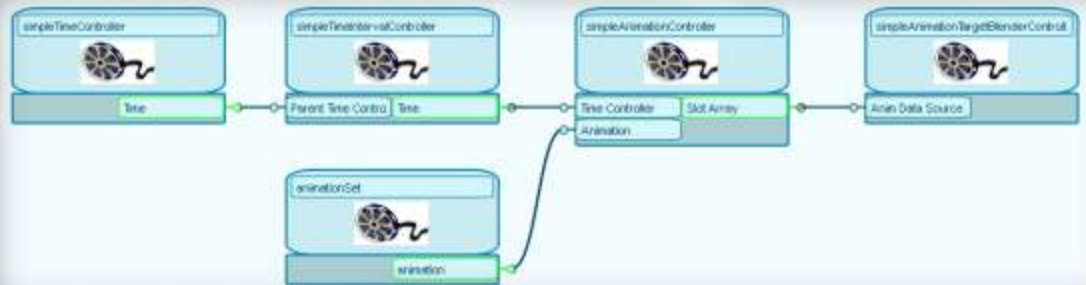
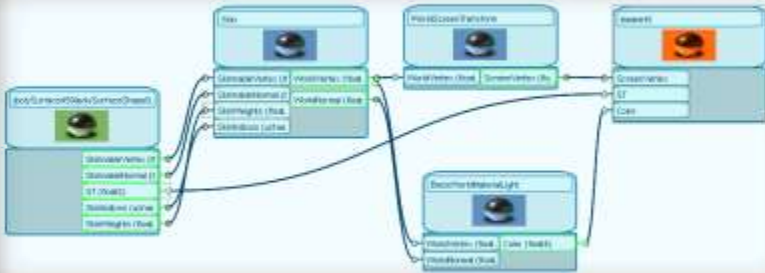
- PhyreEngine™ asset processing tool
  - Interactive
    - Preview, debugging
  - Automated
    - Refactoring, optimization

# Workspace Model



# Graph Editors

- Geometry and animation

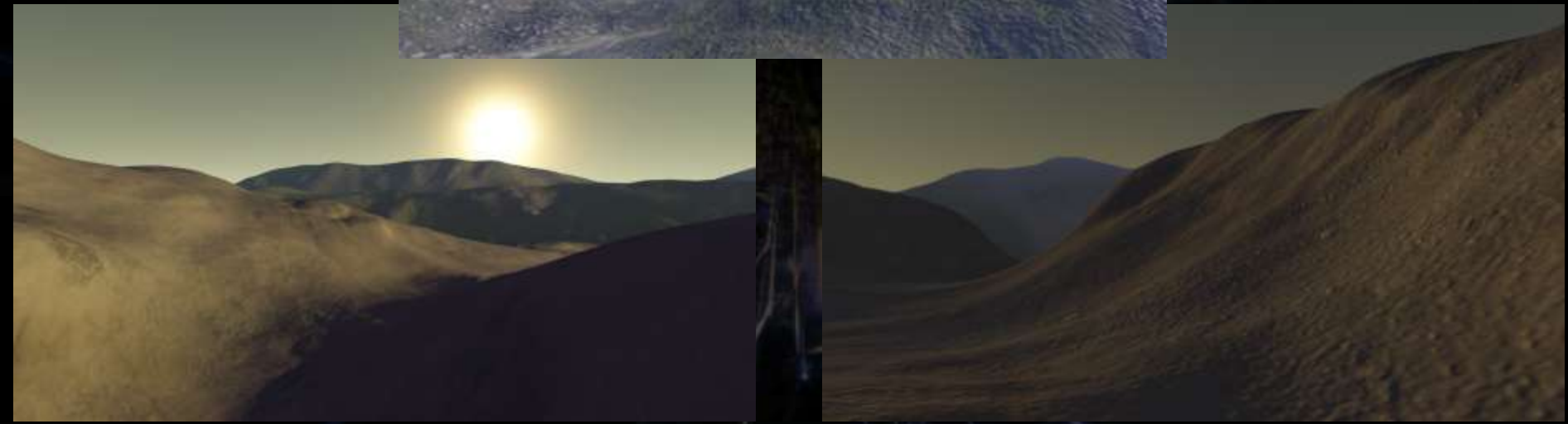




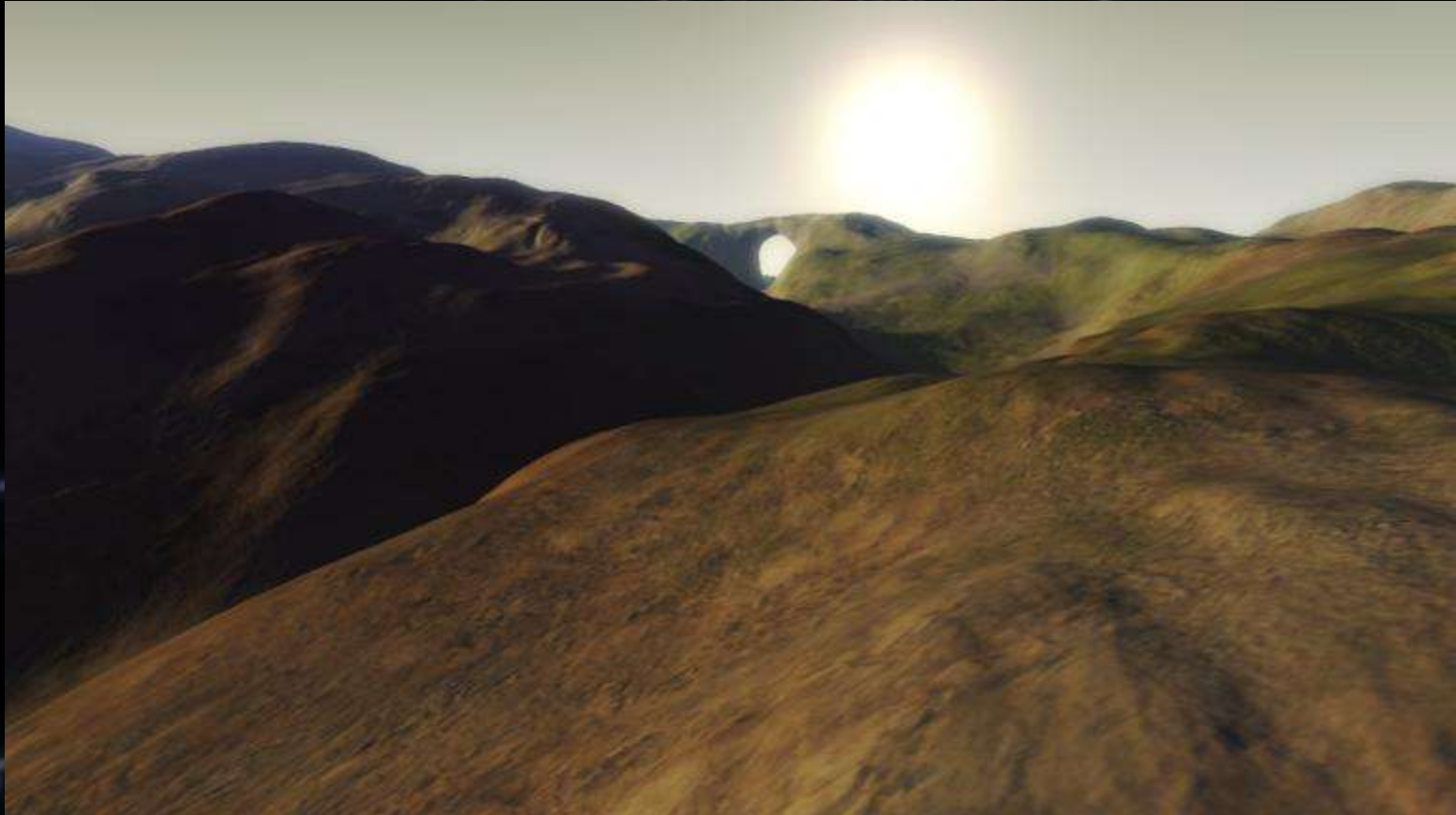
# Terrain Engine

PHYREENGINE™

# Terrain Engine

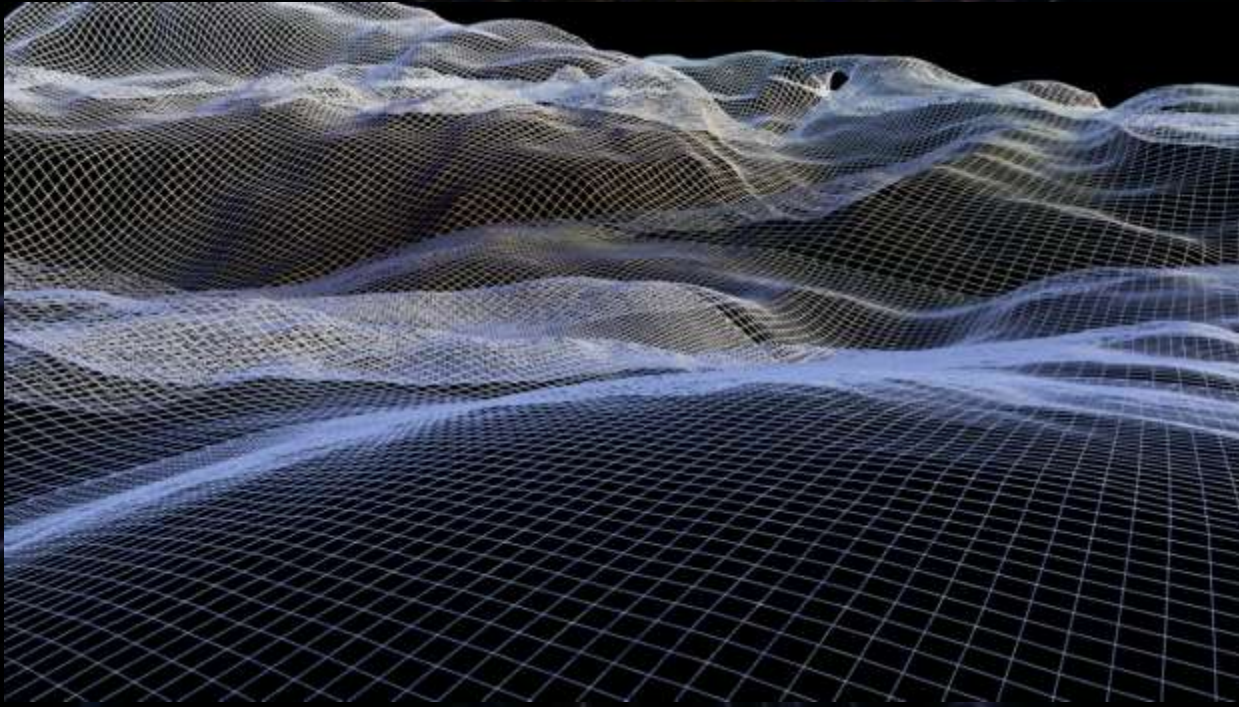


# Freeform mesh based terrain



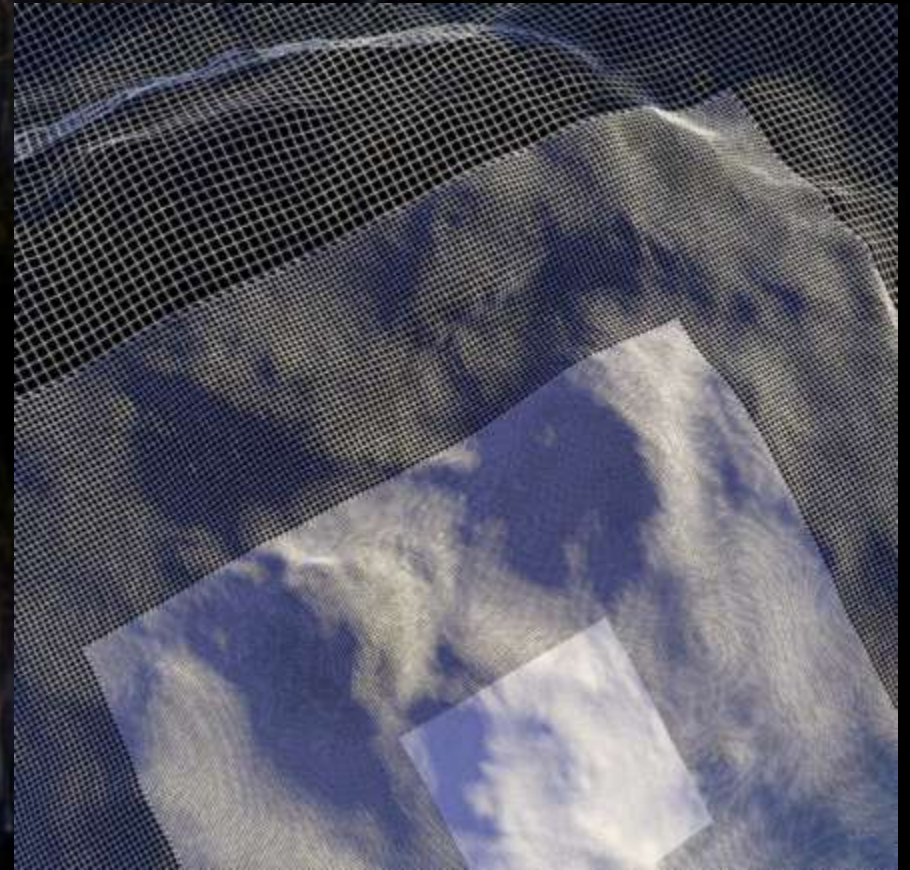
# Features

- Streaming textures and geometry
  - Large worlds - without tiled textures!



# Continuous Level of Detail

- Continuous LOD
  - Similar to mesh-based clipmaps



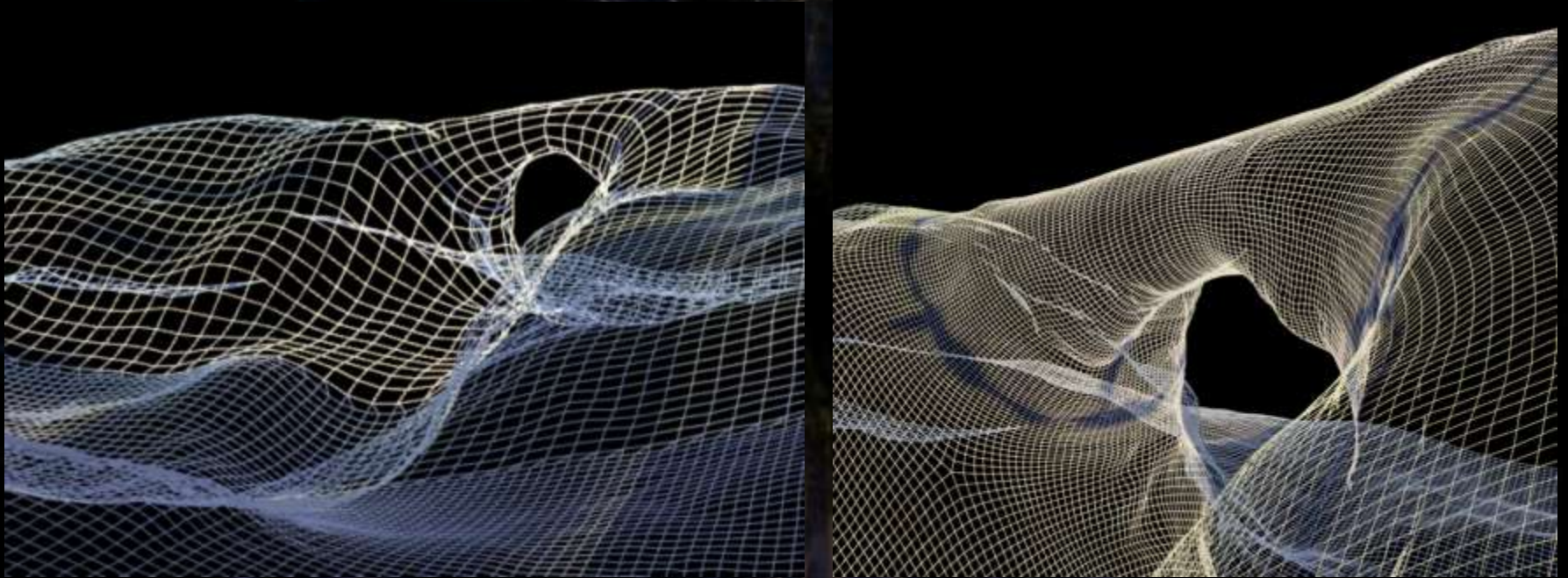
# Terrain blocks

- High level culling by scene graph
- PhyreEngine™ modifier system packets
- Polygon culling by modifier

# Levels of detail

- Specific modifiers for each LOD
- Input:
  - Streams defining required LOD data
- Outputs:
  - Tessellated geometry
  - Index data

# Levels of detail

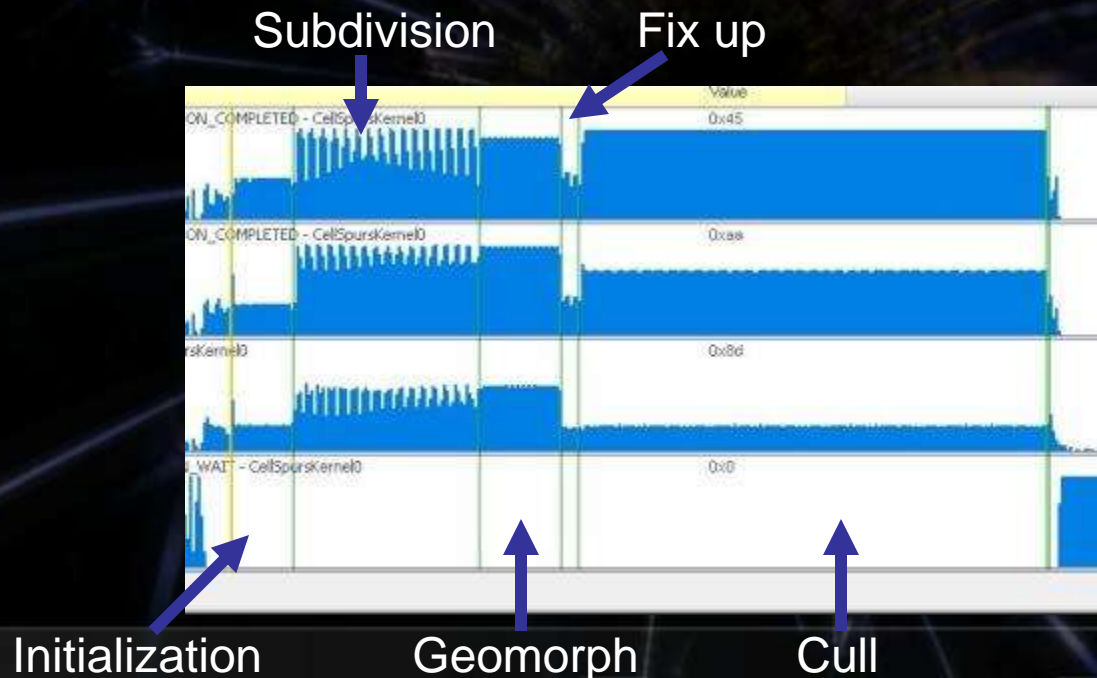


# Development

- Initially developed on Windows
  - Using Cell SDK Vectormath library
- SPU version = 5 minutes
  - Included code in SPU cpp file
  - Compiled...
  - Linked...
  - Ran

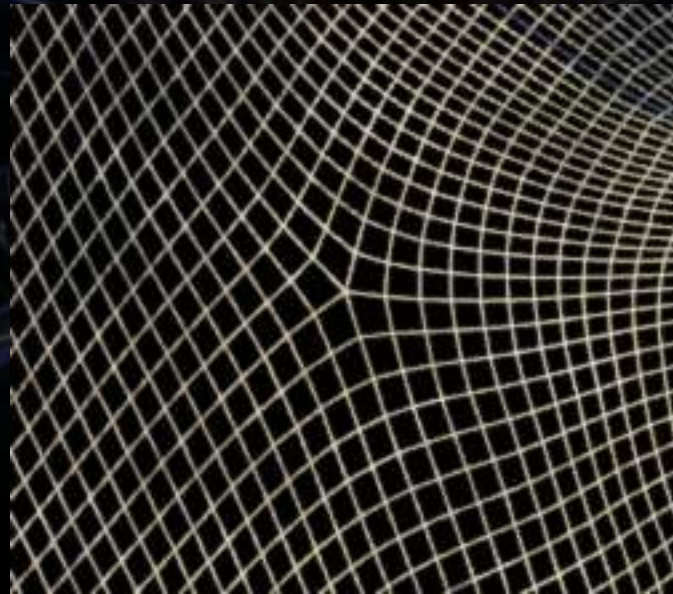
# Performance

- 1040 cycle quad generation loop
- 200M quads/SPU/s



# Workflow

- Model low res meshes – 100 tris/quads
  - Use as building blocks



# Workflow

- Import and arrange in terrain editor
  - Subdivided to produce hi res terrain
- Export sections for sculpting
  - Deform in 3D

# Workflow

- Export grid sections for texture editing
  - Simultaneously edit diffuse, bump and gloss
  - Can also bake lighting due to unique texturing



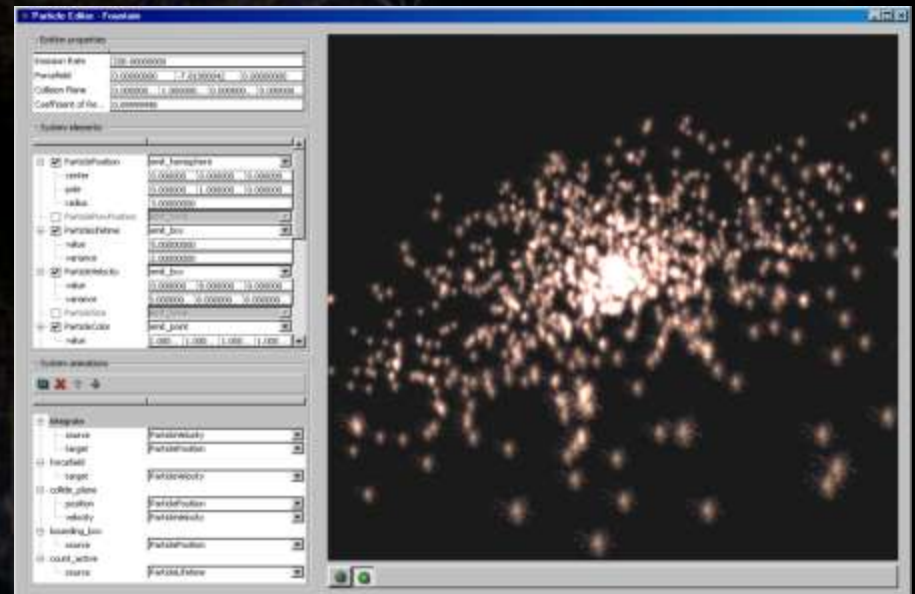
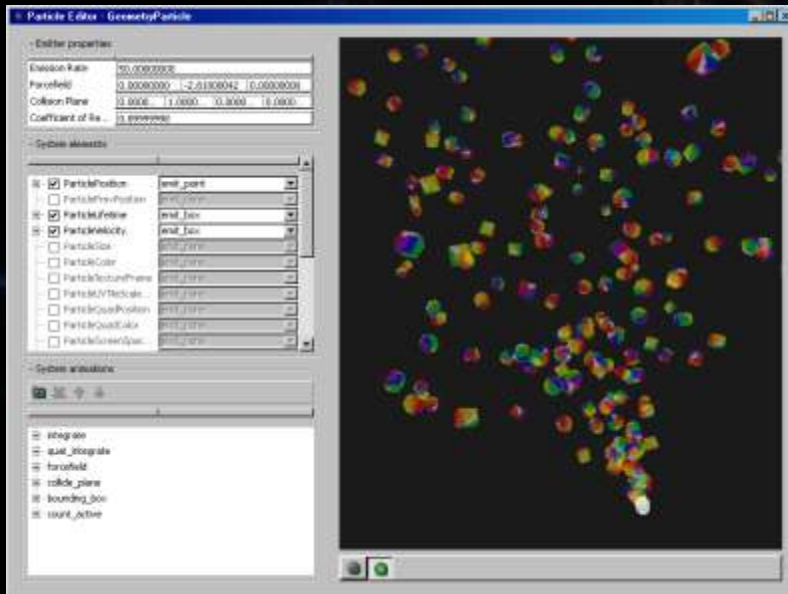


# Recent Additions

PHYREENGINE™

# PhyreEngine™ 2.0

- Particle systems in PhyreStation



# PhyreEngine™ 2.0

- SPU render post processing examples



SSAO



Depth Of Field

# PhyreEngine™ 2.0.1

- Initial terrain engine example
- Bullet physics integration
- HLSL example in Direct3D renderer

# PhyreEngine™ 2.0.1 - April

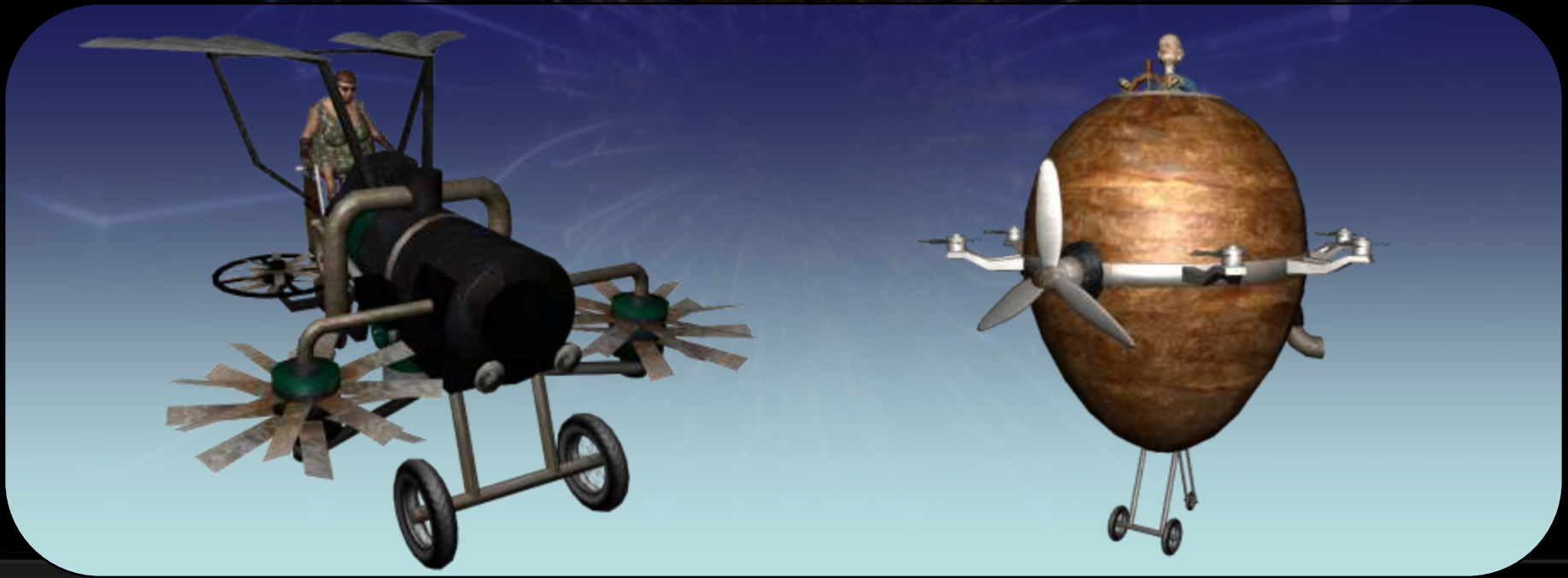
- White Papers
  - Polygon culling and modifiers in general
  - GCM memory management
  - SPU features
    - Scene management and culling
    - Job framework
    - Fragment program patching
    - Texture post processing



# Roadmap

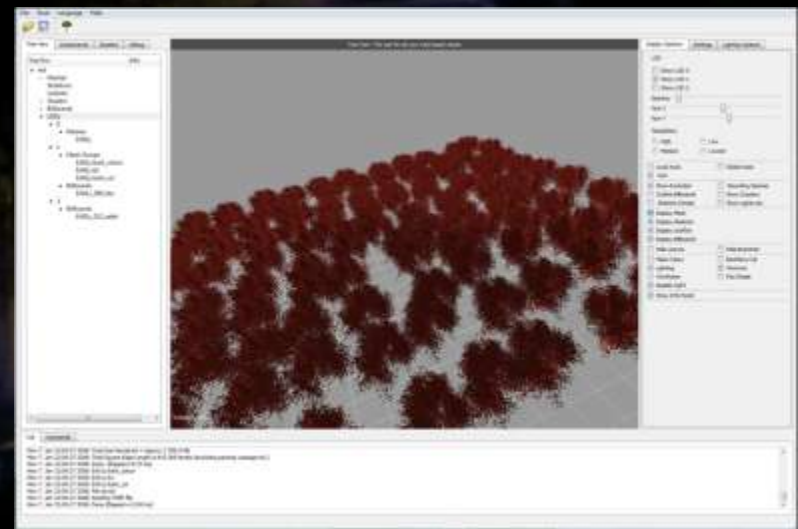
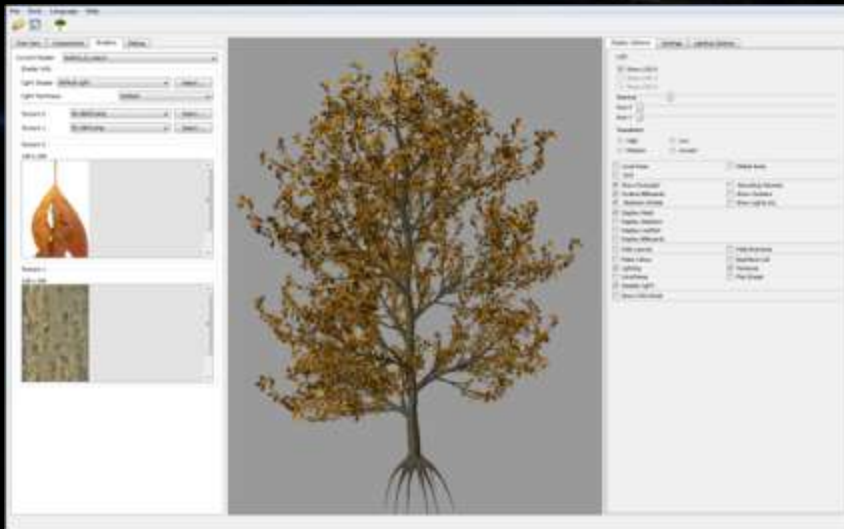
# PhyreEngine™ 2.1 - June

- New game template
  - Racing game using terrain engine



# PhyreEngine™ 2.1 - June

- Full terrain engine and tools
- First release of foliage technology
  - Examples and initial tools



# PhyreEngine™ 2.2 - August

- Further integration of foliage system
- PhyreStation updates
  - Advanced level editing
  - Shader editing & optimization

# PhyreEngine™ 2.3 - November

- Scripting Engine beta
  - SPU based bytecode VM
  - PC tools and PPU library
  - Simple visual editor



# Summary

PHYREENGINE™

# Summary

- Actively developed and supported
- Already proven in chart-topping titles
  - Across several genres and multiple platforms
- Freely available for PLAYSTATION®3 developers
  - With cross platform support

# Questions?

