
Introduction
To Principia
Principia v3.0
Reference Series
(c) Western Star
Entertainment Ltd, 2003-2006

Why Principia ?

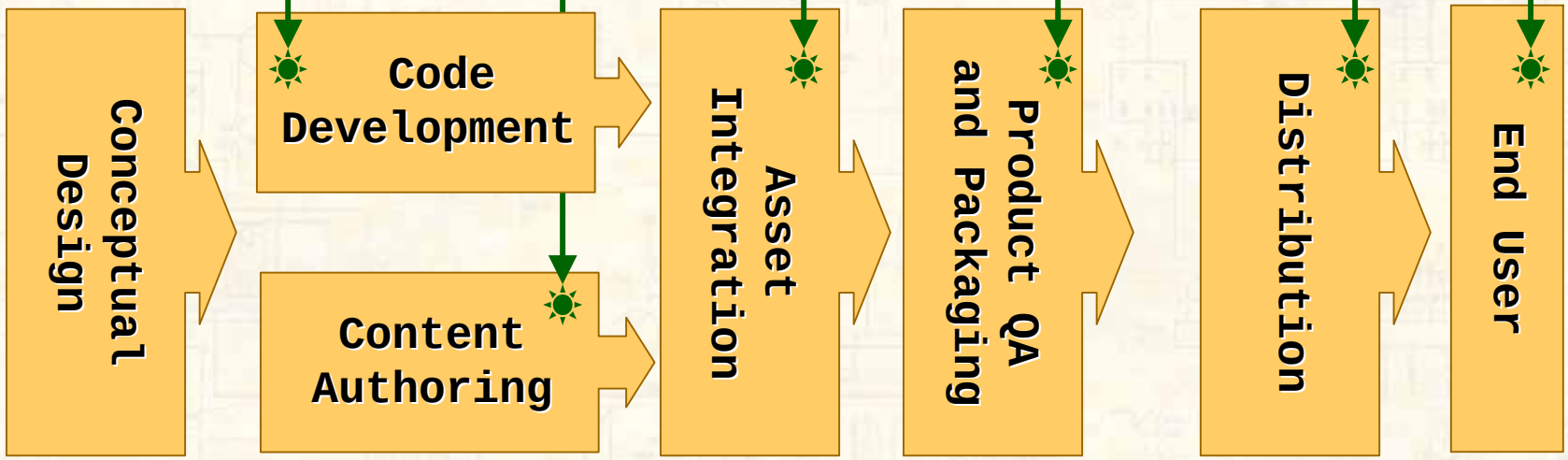
- **Cheaper**
  - Using Principia can reduce the end-to-end production cost of digital entertainment software by up to 40%
- **Faster**
  - Principia can cut development time by up to 70% and multimedia content authoring time by up to 30%
- **Better**
  - With Principia, product is up to 50% less likely to contain critical bugs, and offers a much deeper multimedia experience.

Why Principia ?

- **Example: Create a complex 3D overhead world with interacting, varied characters over a convincing terrain**
  - **Develop in house: weeks to months**
  - **With Principia: ~1-3 days!**
- **Example: Implement blended skeletal animation renders:**
  - **Develop in house: one to several days**
  - **With Principia: ~1 minute!**
- **Example: Implement fast quality path-finding in a complex interactive environment with dynamic obstacles:**
  - **Develop in-house: several days**
  - **With Principia: ~ 15 minutes!**
- **Example: Implement a particle system with instancing:**
  - **Develop in house: hours**
  - **With Principia: ~1-30 minutes!**

What Is Principia ?

Principia is an *end-to-end* toolbox for producing entertainment software.



Simplified Entertainment Software Production Pipeline

What Is Principia ?

Principia is an *integrated* toolbox for producing entertainment software.

Interfaces (APIs, scripts, events…)

Multiple engines

Product assembly components

Content generation procedures

Demos, utilities and assets collection

Principia Engines

- **Principia is not just a game engine:**
  - Many pre-defined ready-to-use game/art engines
  - Developers can create original engines with minimal effort
- **Unprecedented breadth of capability**
  - Third person / First person / Cinematic / Arcade ...
  - Open landscape / Interior / Custom environments ...
  - Single player / Multiplayer / Instanced ...
  - Fully configurable via script interface ...
- **Powerful script and application interfaces**
- **Integrated asset management**
- **Embedded physics, process simulation and AI**
- **Best-of-breed performance and capability**

Principia Components (1)

- **Application components implement nearly all elements of successful games without coding.**
  - **Controls**
  - **Multimedia content**
  - **Characters and objects**
  - **Animators and behaviors**
  - **Worlds**
  - **Viewers**
  - **And much more …**
- **Custom engines can be created by assembling different world and viewer components !**

Principia Components (2)

- **Production components provide support for the market success of your product:**
  - Installation
  - Customization
  - Collaborative content authoring
  - Licensing and digital rights management
  - Electronic distribution
  - Performance tuning and QA
  - Data persistence and interchange
  - Security
  - Auditing, accounting and billing
  - And much more…

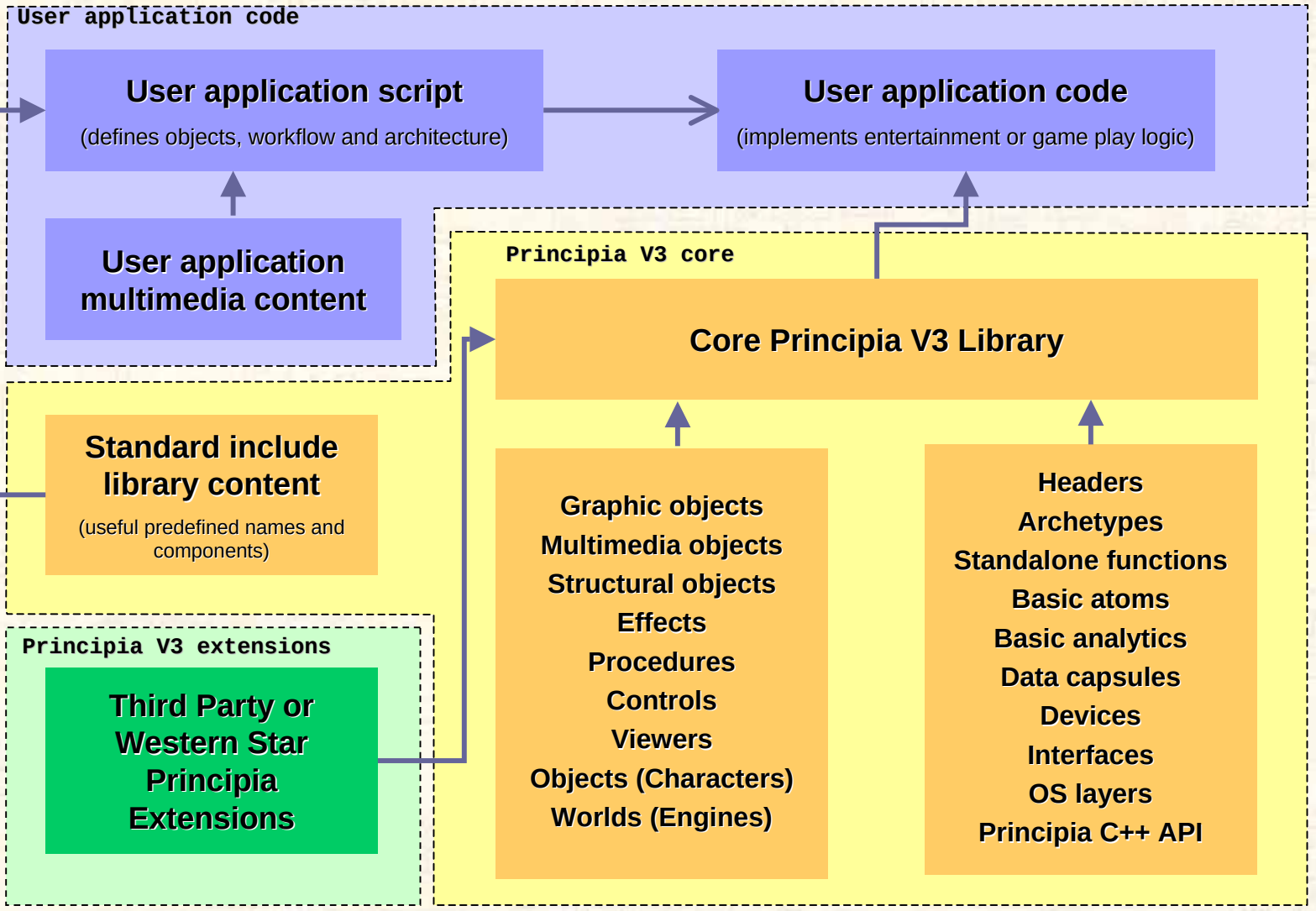
Principia Procedures

- **Procedures author and manage quality digital assets rapidly and effortlessly:**
  - **Textures and materials**
  - **Geometry**
  - **Animations**
  - **Objects**
  - **Characters**
  - **Particle systems**
  - **Terrains**
  - **Worlds**
  - **And much more...**

Principia Interfaces

- **Scripting interfaces**
  - Customize components and create complete applications with a minimum of code development.
- **Application interfaces**
  - Integrate user code with Principia components and engines to create custom applications.
- **Actor/event interfaces**
  - Implement hardware, player, AI, data and network interactions across multiple platforms.
- **Auto-scalable client-server architecture**
  - Run distributed multi-user applications on multiple CPUs and GPUs. Or play a simple game of solitaire on a legacy platform.

Principia V3 Application Structure



Why Use The Demos ?

Much easier to learn how to do things!

The Principia Demos Package provides examples of virtually any type of functionality commonly needed in commercial entertainment software.

The Package is an excellent companion for working with the Principia Reference Guide, which documents the full spectrum of Principia capabilities. It is also a great platform for doing your own experiments.

Demos Table of Contents

Note: Do not be mislead by the world “basic” you will see in Section I. Some of the basic demos take it from where the latest books on advanced game development leave.

I. Core Components

- Chapter 1 - Introduction to Principia
- Chapter 2 - Core Controls
- Chapter 3 - Basic Rendering
- Chapter 4 - Basic Shaders
- Chapter 5 - Basic Animation
- Chapter 6 - Structural Components
- Chapter 7 - Core Objects
- Chapter 8 - Core Worlds
- Chapter 9 - Core Viewers
- Chapter 10 - Audio and Video

II. Specialized Components

- Chapter 1 - Advanced Controls
- Chapter 2 - Advanced Rendering
- Chapter 3 - Advanced Animation
- Chapter 4 - Effects, CPU-Driven
- Chapter 5 - Effects, GPU-Driven
- Chapter 6 - Specialized Material Objects
- Chapter 7 - Specialized Logical Objects
- Chapter 8 - Specialized Viewers
- Chapter 9 - Specialized Worlds

III. Procedural Components

- Chapter 1 - Elemental Procedures (Algebraic)
- Chapter 2 - Elemental Procedures (Structural)
- Chapter 3 - Procedural Textures
- Chapter 4 - Procedural Geometry
- Chapter 5 - Procedural Objects
- Chapter 6 - Procedural Animation
- Chapter 7 - Procedural Particle Systems
- Chapter 8 - Procedural Behaviors
- Chapter 9 - Procedural Worlds
- Chapter 10 - Procedural Narratives
- Chapter 11 - Procedural Audio
- Chapter 12 - Data Procedures

IV. Play & Physics Components

- Chapter 1 - Behaviors I (characters)
- Chapter 2 - Behaviors II (items)
- Chapter 3 - Play data management
- Chapter 4 - Decision-making and AI
- Chapter 5 - Rigid single body dynamics
- Chapter 6 - Rigid skeletal system dynamics
- Chapter 7 - Soft single body dynamics
- Chapter 8 - Soft skeletal system dynamics
- Chapter 9 - Particle systems dynamics
- Chapter 10 - Diffusion processes
- Chapter 11 - Fluid motion
- Chapter 12 - Ecosystem simulation

V. Infrastructure Components

- Chapter 1 - Interfaces and devices
- Chapter 2 - Users and actors
- Chapter 3 - Scene and workflow design
- Chapter 4 - Networking and connectivity
- Chapter 5 - User and data persistence
- Chapter 6 - Configuration and requirements
- Chapter 7 - Performance and optimization
- Chapter 8 - Production packaging
- Chapter 9 - Encryption and rights protection
- Chapter 10 - Cross-platform development
- Chapter 11 - Distribution and installation

VI. Production Gallery

- Chapter 1 - Scenes
- Chapter 2 - Characters
- Chapter 3 - Creatures
- Chapter 4 - Things
- Chapter 5 - Landscapes
- Chapter 6 - Production VFX
- Chapter 7 - Production AFX
- Chapter 8 - Short movies
- Chapter 9 - Example Applications, Games
- Chapter 10 - Example Applications, Media

VII. Developer Utilities

- Chapter 1 - Component editors
- Chapter 2 - World editors
- Chapter 3 - Format converters
- Chapter 4 - Requirement testers
- Chapter 5 - Performance testers
- Chapter 6 - Script generators

How to Use This Document ?

- This User Guide is a collection of working demos. All demos are self-contained applications that illustrate a particular functional aspect of Principia V3.
- Each demo showcases:
  - Necessary Principia script to implement the demo
  - Optional Principia API custom code
  - Discussion of component properties and methods
  - Discussion of implementation techniques
- No demo provides a comprehensive record of the entire set of Principia commands, objects, methods, properties and capabilities. We recommend reading the relevant sections of the Principia Reference Manual each time a new item is introduced.

How to Use the Demos ?

- Each self-contained demo application is in a separate folder that holds all demo-specific files.
- Files common to multiple demos are placed in a special common folder at the head of the hierarchy
- The folders are organized in a directory tree that mirrors the sections of the user guide.
- The user guide is accompanies by a Microsoft Visual C++ (version 6 or above) project, providing access to the demo hierarchy, source code and scripts.
- Using this project, you can compile, run, modify and experiment with the demos.

Demo Nomenclature

- **Demo nomenclature hierarchy**
  - General name: Demo
  - Section numeral: Demo_1
  - Chapter numeral: Demo_1.01
  - Demo folder topic: Demo_1.01.A_TopicName
  - Example within demo topic numeral: /Ex01
  - Component within example numeral: /Ex01A
- **Demo_1.02.B_Buttons/Ex04C**
  - Section 1, Chapter 2 (basic controls)
  - Topic B (buttons)
  - Example 4 (an example of particular selectable feature within the demo, such as a specific type of button)
  - Component C (third illustrative button in example)
- **Shorthand notations**
  - File names feature no dots, i.e. 1.02.B = 102B
  - Slide title references abbreviate Demo_1.02.B to D102B

Demo Requirements

- **Principia is *NOT***
  - A substitute for DirectX or OpenGL.
  - A substitute for the system low-level SDK.
- **The Principia V3 toolbox provided requires**
  - Microsoft Windows platform (Win98 or later)
  - Microsoft DirectX (9.0a or later)
  - An integrated C++ compiler
- **Several demos require a graphic adapter with advanced 3D rendering capabilities. If these capabilities are not present, the demo may generate an alert and terminate.**

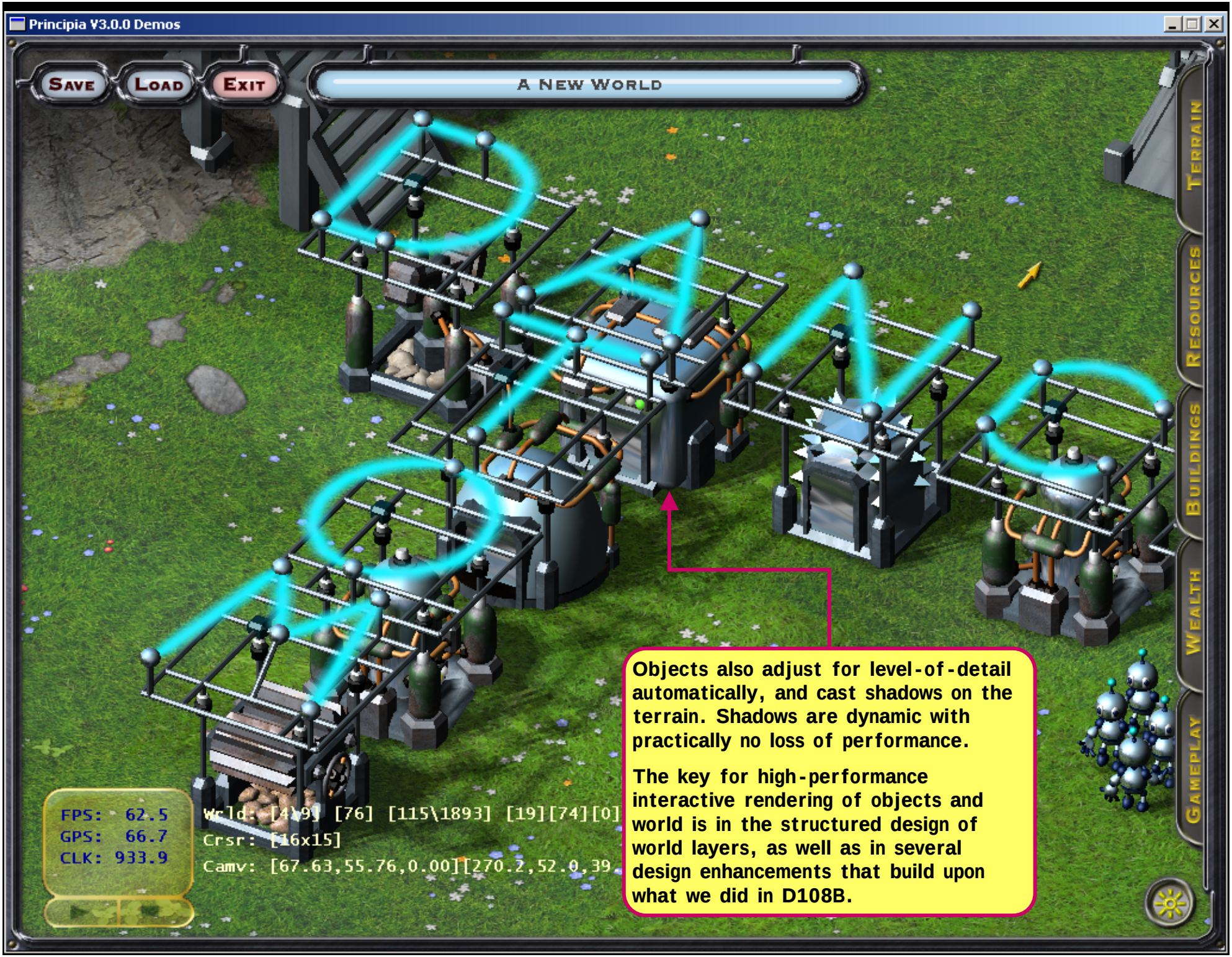
Why Is Principia Different?

- **Designed for real-time dynamic products**
  - Unlike all 3D modeling systems out there
- **Designed for content generation**
  - Unlike all game engines out there
- **Designed for multiple product types**
  - Unlike most game engines
- **End-to-end, integrated production**
  - Unlike all 3D modelers and game engines
- **Proven in high-performance quality apps**

Sample Demos Gallery

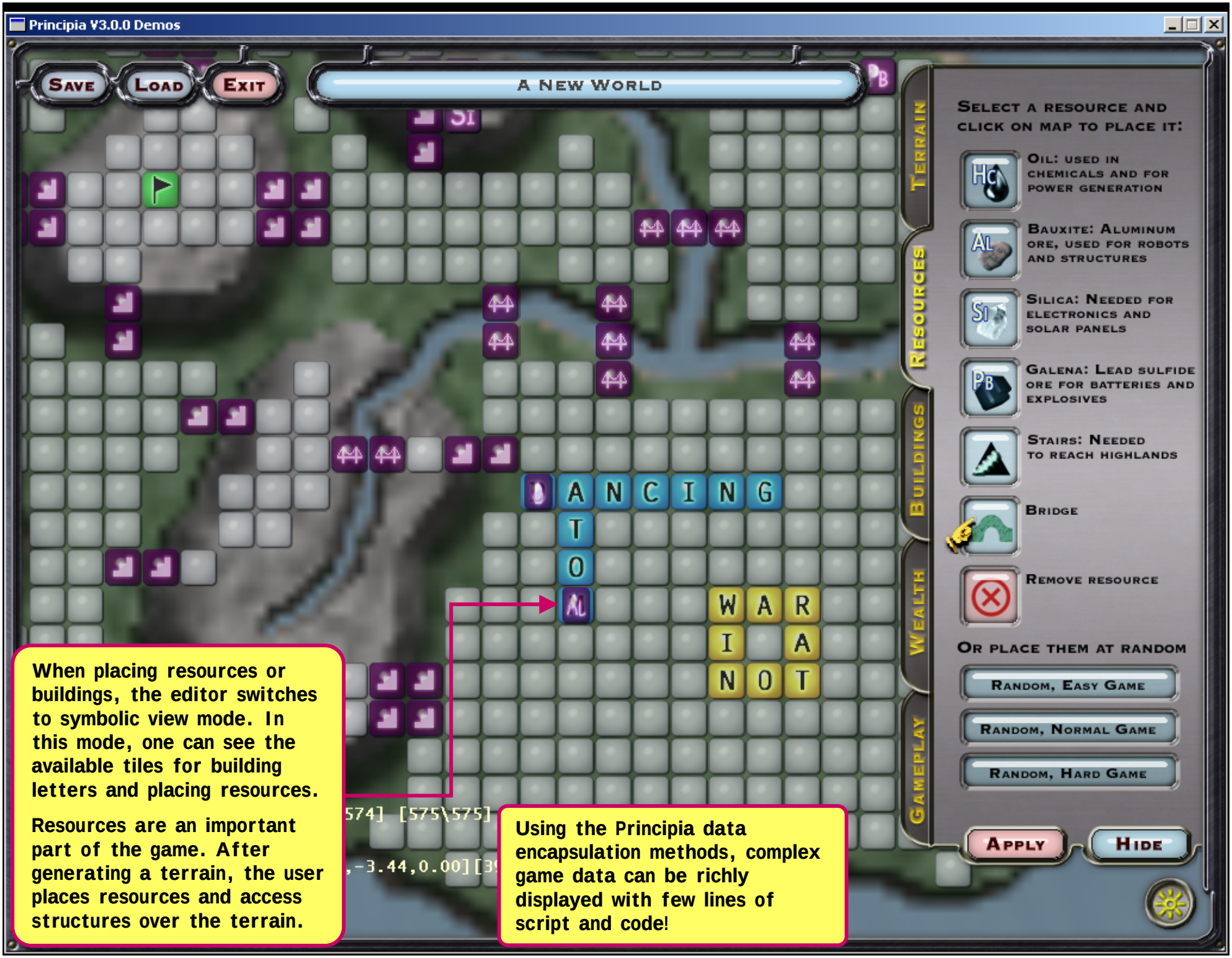
- **Enclosed are screenshots from few of the demos in the Principia Tutorials Demo User Guide**
  - The selection ranges from the simple to the arcane.
  - The selected topics cover rendering, controls, worlds and procedural generation.
  - Most demos do not require a single line of code!
  - Most demos are based on components that are easily configurable in script.

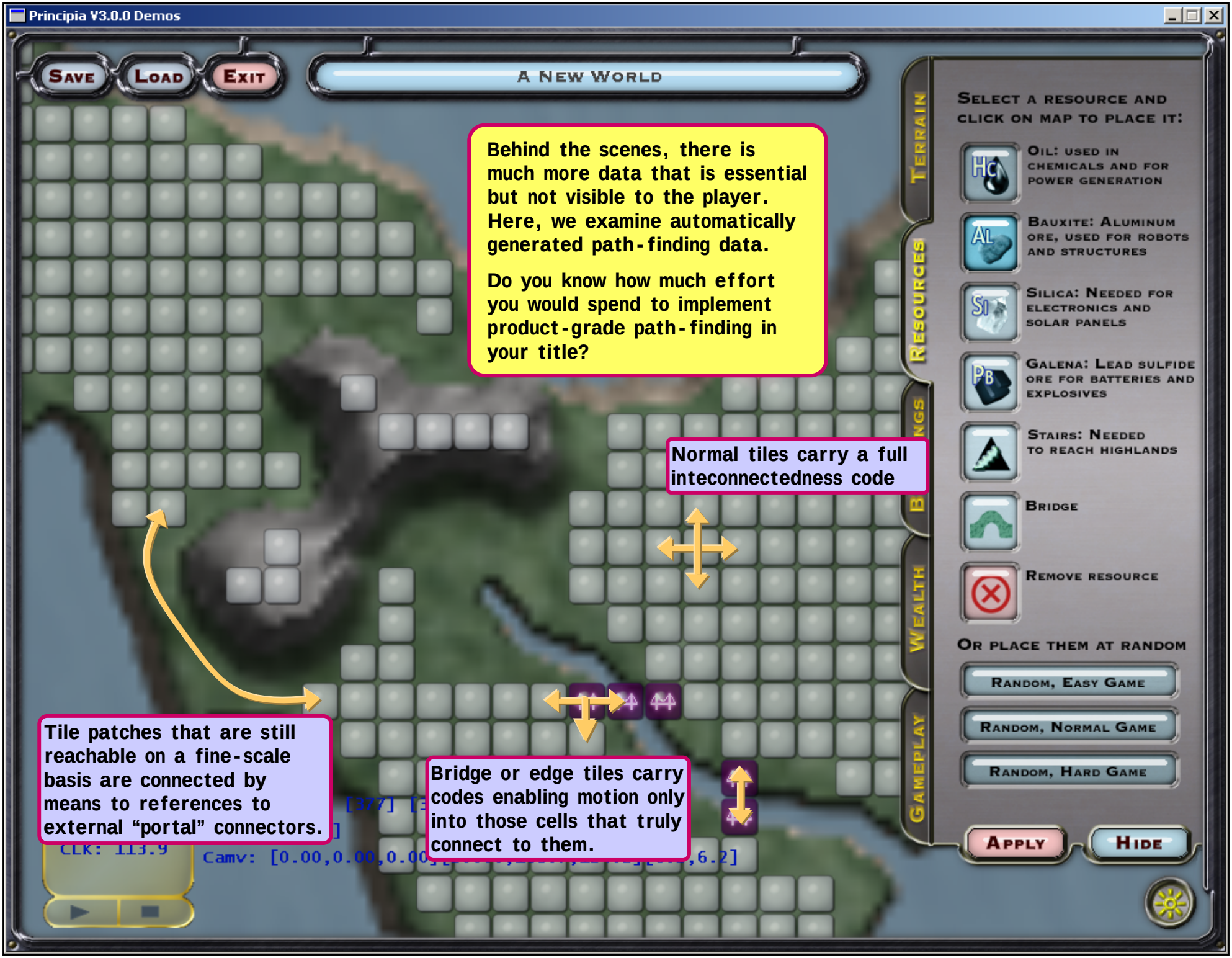

Principia ¥3.0.0 Demos
Save
Load
Exit
A New World
Terrain
Resources
Buildings
Wealth
Gameplay
DANGLING
MOMENT
WAR
NEAR
NOT
The world of D108C/D702A houses a wide variety of gameplay objects, which are placed using simple and intuitive GUI controls (no more object scatter procedures!).
FPS: 66.7
GPS: 31.3
CLK: 832.2
Wrld: [5\9] [80] [419\1905] [17][68][0]
Crsr: [13x16, 13x17, 14x17]
Camv: [76.35,40.66,0.00][370.6,205.6,154.1][5.0,5.0]


Principia ¥3.0.0 Demos
SAVE
LOAD
EXIT
A NEW WORLD
TERRAIN
RESOURCES
BUILDINGS
WEALTH
GAMEPLAY
FPS: 62.5
GPS: 66.7
CLK: 933.9
Wrld: [4\9] [76] [115\1893] [19][74][0]
Crsr: [16x15]
Camv: [67.63,55.76,0.00][270.2,52.0,39.
Objects also adjust for level-of-detail automatically, and cast shadows on the terrain. Shadows are dynamic with practically no loss of performance.
The key for high-performance interactive rendering of objects and world is in the structured design of world layers, as well as in several design enhancements that build upon what we did in D108B.


Principia V3.0.0 Demos
SAVE
LOAD
EXIT
A NEW WORLD
TERRAIN
RESOURCES
BUILDINGS
WEALTH
FPS: 32.3
GPS: 32.3
CLK:1112.4
Wrld: [4\9] [76] [204\1926] [38][80][0]
Crsr: [17x18]
Camv: [69.36,90.83,0.00][297.2,57.2,42.9][17.9,17.9]
Objects such as buildings can be interacted with via the pointer in an intuitive manner (even if our world does not implement a full gameplay yet). The objects are complex, multi-component constructs which execute complex animations and combine several different illumination effects such as reflective metal or glowing letters.


Principia V3.0.0 Demos
Save
Load
Exit
Terrain
Resources
Illumination
Luminosity Peak
Luminosity Midpoint (% of peak)
Luminosity Base (% of peak)
Specular Maxima (% of peak)
Specular Minima (% of maxima)
Specular Shapness, metals
Specular Shapness, earths
Atmospheric Balance
Lighting Model
Classic Blinn
Enhanced
Hide
FPS: 32.3
GPS: 32.3
CLK: 525.8
Complex controls are easily scripted and integrated with the user application. Here, we have a control panel that fine-tweaks a large collection of shaders.


Principia ¥3.0.0 Demos
Save
Load
Exit
A New World
Terrain
Resources
Buildings
Wealth
Gameplay
Select a resource and click on map to place it:
Oil: used in chemicals and for power generation
Bauxite: Aluminum ore, used for robots and structures
Silica: Needed for electronics and solar panels
Galena: Lead sulfide ore for batteries and explosives
Stairs: Needed to reach highlands
Bridge
Remove resource
Or place them at random
Random, Easy Game
Random, Normal Game
Random, Hard Game
Apply
Hide
DANCING
TO
WAR
IS
A
NOT
When placing resources or buildings, the editor switches to symbolic view mode. In this mode, one can see the available tiles for building letters and placing resources.
Resources are an important part of the game. After generating a terrain, the user places resources and access structures over the terrain.
Using the Principia data encapsulation methods, complex game data can be richly displayed with few lines of script and code!


Principia ¥3.0.0 Demos
Save
Load
Exit
A New World
Behind the scenes, there is much more data that is essential but not visible to the player. Here, we examine automatically generated path-finding data.
Do you know how much effort you would spend to implement product-grade path-finding in your title?
Normal tiles carry a full inteconnectedness code
Tile patches that are still reachable on a fine-scale basis are connected by means to references to external “portal” connectors.
Bridge or edge tiles carry codes enabling motion only into those cells that truly connect to them.
Terrain
Resources
Buildings
Wealth
Gameplay
Select a resource and click on map to place it:
Oil: used in chemicals and for power generation
Bauxite: Aluminum ore, used for robots and structures
Silica: Needed for electronics and solar panels
Galena: Lead sulfide ore for batteries and explosives
Stairs: Needed to reach highlands
Bridge
Remove resource
Or place them at random
Random, Easy Game
Random, Normal Game
Random, Hard Game
Apply
Hide

PrincipiaV.3.0.0

Button2C with persistent state and gradual transitions
between icons upon mouse click

Lets shift to something simple: Controls. Here are two buttons with programmable graphic and logic behavior...

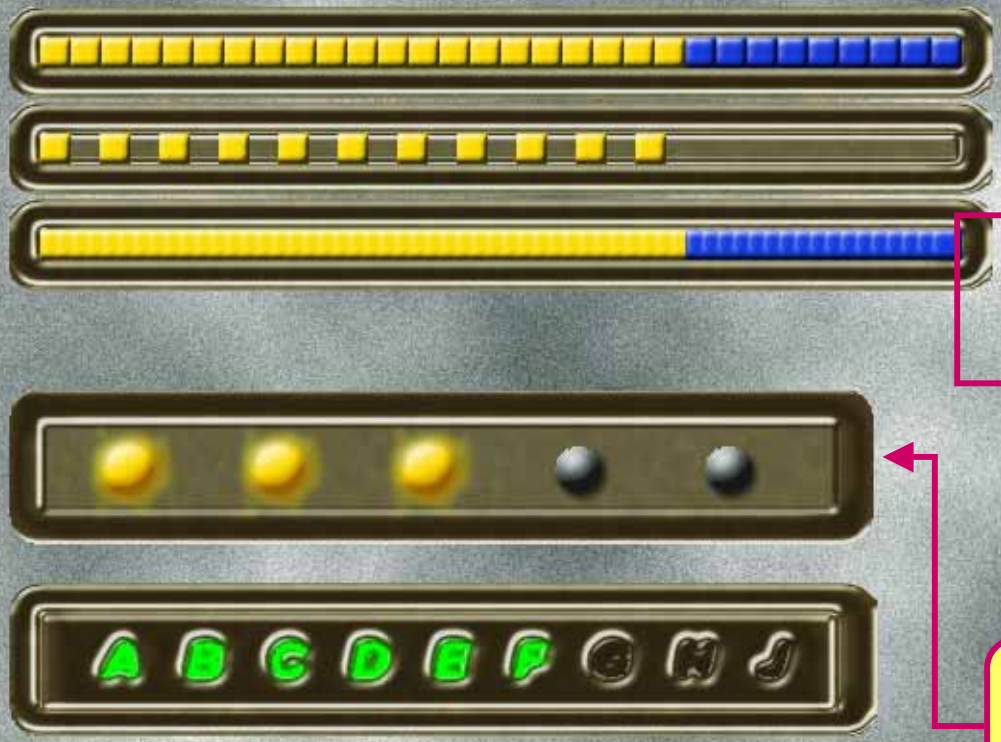
In neutral state, both buttons are red. When clicked, the topmost button smoothly shifts to a yellow “on” icon. When clicked again, it shifts smoothly back to the neutral red icon.

When clicked, this button smoothly shifts to yellow and then back to red without a second click. The button is non-persistent.

FPS: 33.3
GPS: 33.3

PrincipiaV.3.0.0

Ex4: Indicator2A variants using non-counting icon repetition



This is the same control component (Indicator2A) configured differently.

Principia has a vast library of control components.

Examples 4A-C show progress by repeating two simple icons with different spacing

Example 4D shows progress by illuminating a row of lightbulbs

Example 4E lights each character as the indicator progresses… but wait…these are not series of identical images! How on earth do we do that?

Principia V3.0.0 Demos

Ex01: Basic Map2S. Method=Data mapped textures, Data=Z(X,Y)
DataTex=4, WaterTex=Y, WaterAlpha=Y, Shading=Model, TexDim=MapDim

A minimap control, configured with a dozen script lines and not a single line of code !!!

FPS: 62.5
GPS: 66.7
CLK: 6.8

Lighting on
Specular on
Perspective
Orthogonal

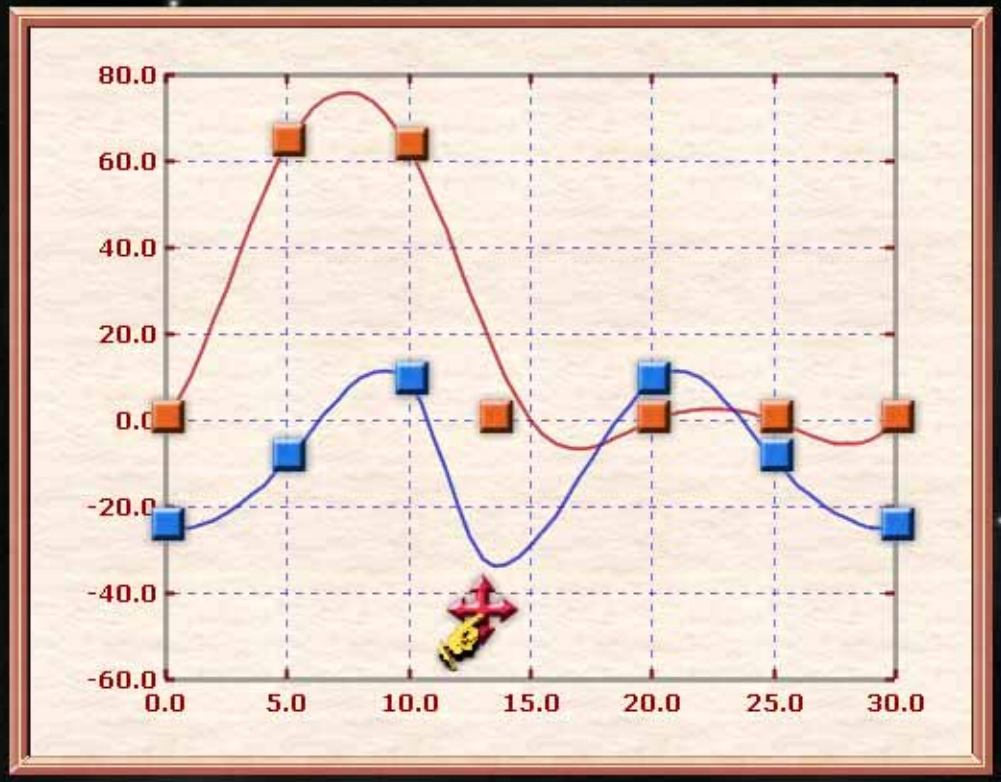
View
Lite1
Lite2
Lite3
Matl

Camera Distance 6.0
Camera Elevation 25.0

This is a simple use case of Map2S to generate a geomorphic terrain map. It maps different textures to different scalar grid elevations, applies a water level with depth-based transparency, and performs user-configured terrain slope shading.

Principia V3.0.0 Demos

Ex02: Basic Graph2A. Object-drawn axis box, no autoranging, points and point methods.
DataX=Var(Lineal(Frame)), DataY=Var(Lineal(XYZ)), data defined in script.



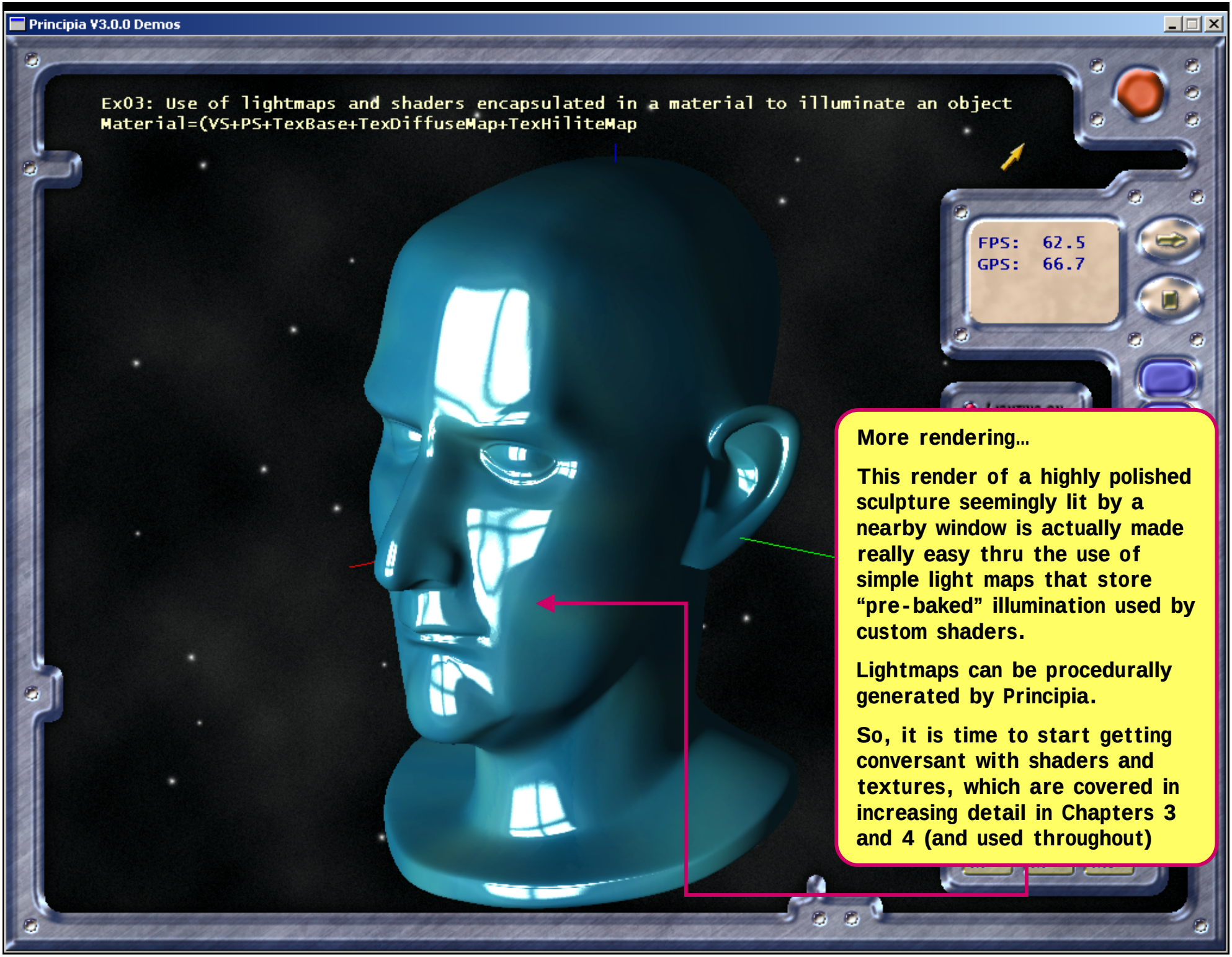
Same for interactive graphs !!!

Lighting On
Specular On
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance 4.0
Camera Elevation 25.0
Camera Azimuth 45.0
Camera FoV 45.0
Look At Point 0.0 0.0 0.6

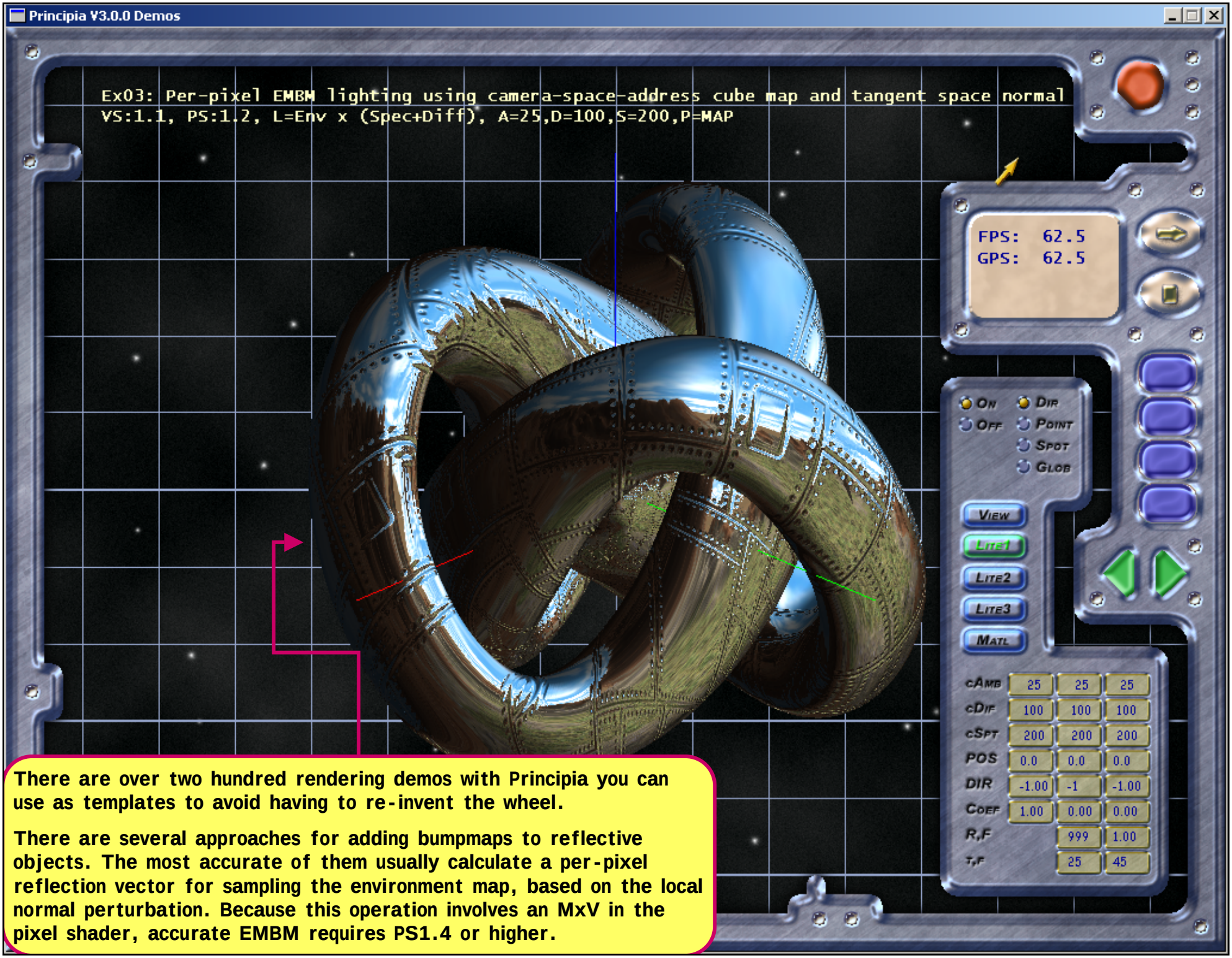
Lineals are essential generating objects in Principia. CX_Graph2A allows not only to plot their full trace, but also to visualize the discrete, defining lineal points, and interactively moving them, thereby updating the entire variables and lineals associated with the plot. This CX_Graph2A functionality is essential for interactive graphic input.

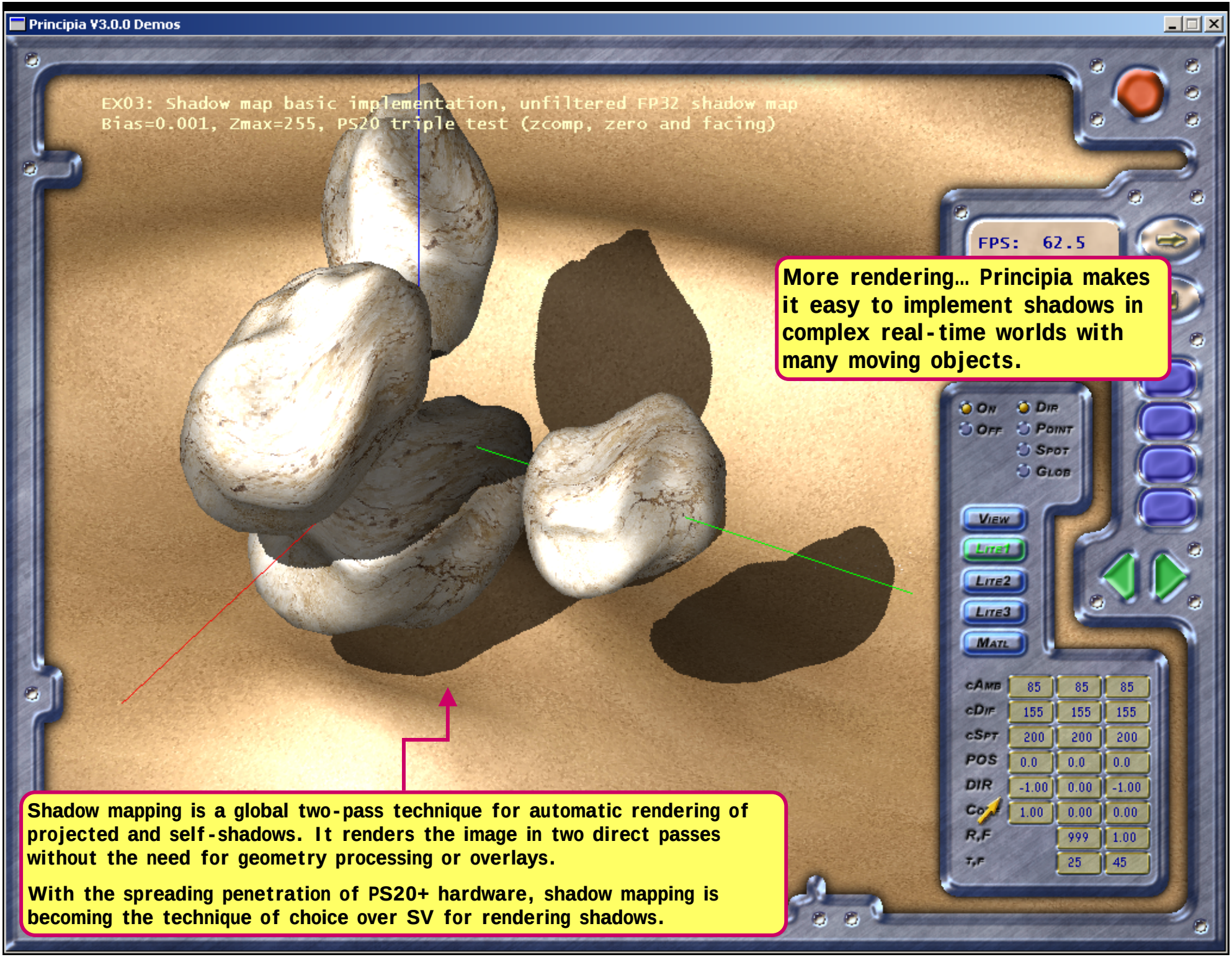
Note how changing the axial position of a point updates both traces. This is because they all share the same independent variable lineal.


Principia V3.0.0 Demos
Ex01: Planar reflection implemented via rendering and testing the stencil buffer
Render sequence control of Alpha,Depth,Stencil buffers via material setting
FPS: 62.5
66.7
Simple transparent overlay of two images is the most typical use of the frame buffer.
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
6.0
Camera Elevation
17.0
Camera Azimuth
21.0
Camera FoV
25.0
Lets do some rendering now. You can define complex rendering sequences easily in script. Principia encapsulates content in a unique manner that is geared towards real-time game rendering, while enabling full range of complex VFX.

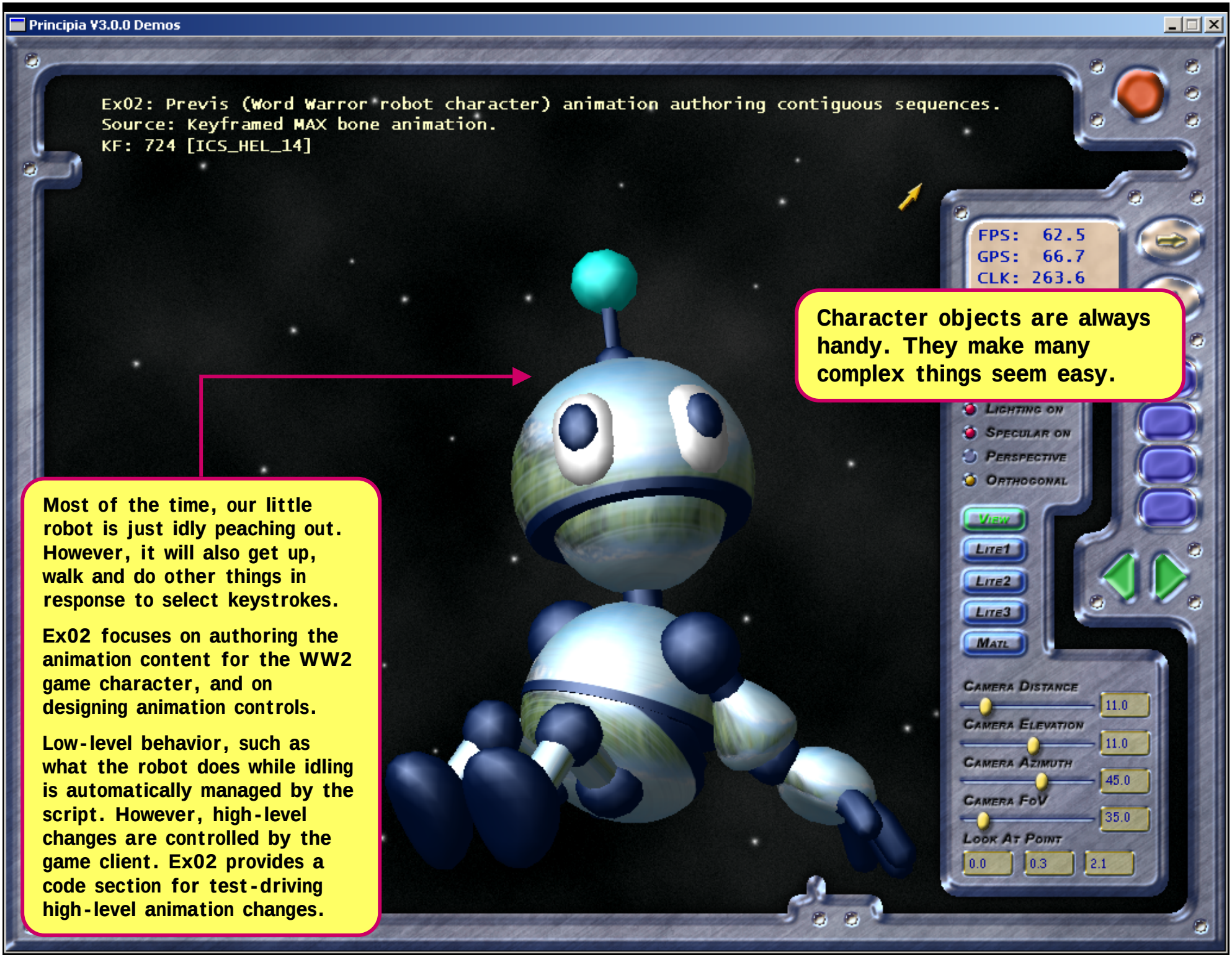

Principia V3.0.0 Demos
Ex03: Use of lightmaps and shaders encapsulated in a material to illuminate an object
Material=(VS+PS+TexBase+TexDiffuseMap+TexHiliteMap
FPS: 62.5
GPS: 66.7
More rendering…
This render of a highly polished sculpture seemingly lit by a nearby window is actually made really easy thru the use of simple light maps that store “pre-baked” illumination used by custom shaders.
Lightmaps can be procedurally generated by Principia.
So, it is time to start getting conversant with shaders and textures, which are covered in increasing detail in Chapters 3 and 4 (and used throughout)


Principia V3.0.0 Demos
Ex03: Render target encompassing a GUI render step results in a reflection-like effect.
RT Effect=Clr_Color_Depth. Pass1: Render GUI on RT, Pass2: Render OBJ with TEX=RT
FPS: 32.3
GPS: 66.7
On
Off
Dir
Point
Spot
Glob
View
Lite1
Lite2
Lite3
Matl
cAmb 100 100 100
cDif 200 200 200
Complex rendering tasks are simple to script…
This sphere seems to reflect the GUI, including the moving cursor and all. The effect is achieved by rendering the GUI onto a RT texture, which is then used as a skin for the scene objects.

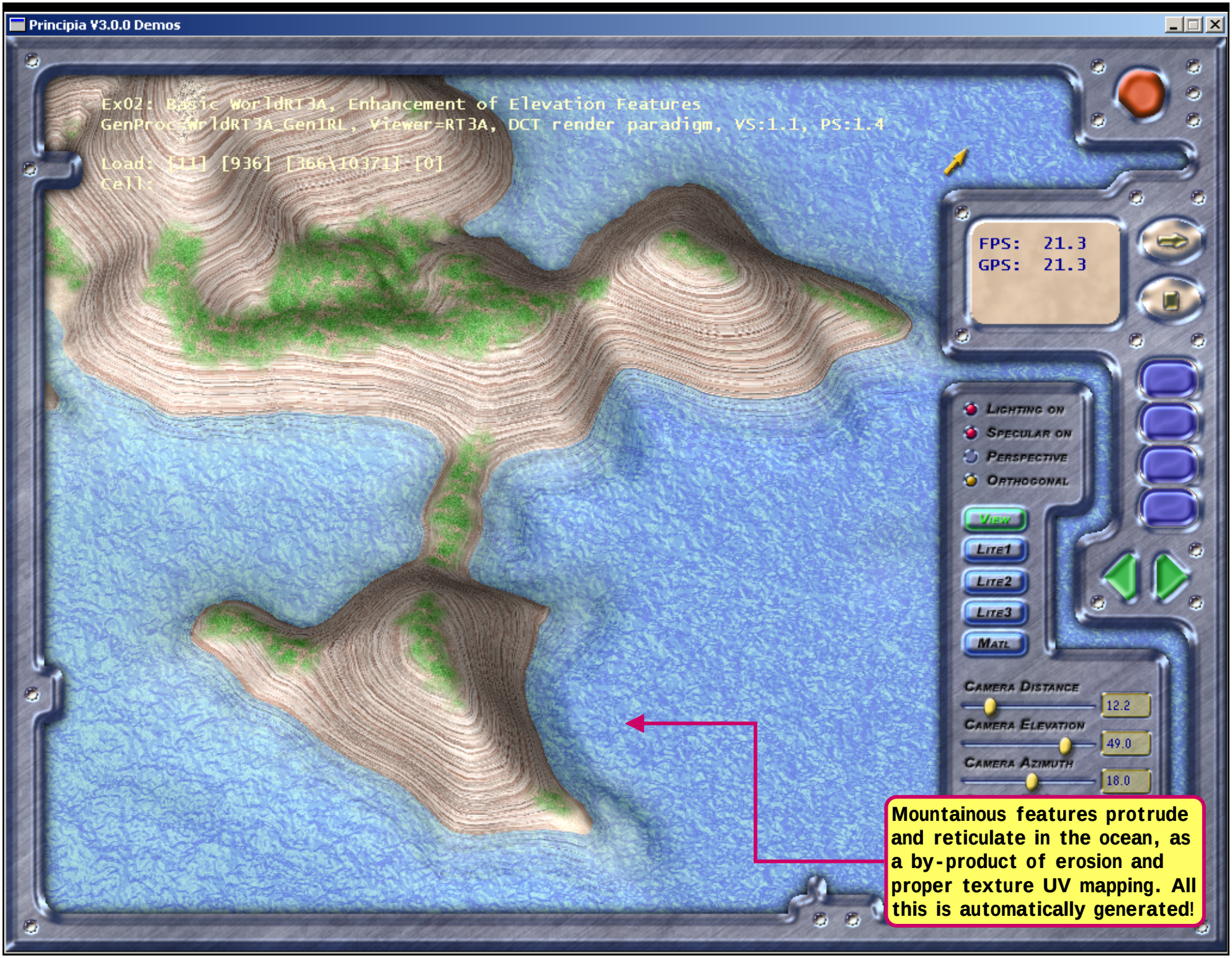

Principia V3.0.0 Demos
Ex03: Per-pixel EMBM lighting using camera-space-address cube map and tangent space normal
VS:1.1, PS:1.2, L=Env x (Spec+Diff), A=25,D=100,S=200,P=MAP
FPS: 62.5
GPS: 62.5
On
Off
Dir
Point
Spot
Glob
View
Lite1
Lite2
Lite3
Matl
cAmb 25 25 25
cDif 100 100 100
cSpt 200 200 200
POS 0.0 0.0 0.0
DIR -1.00 -1 -1.00
Coef 1.00 0.00 0.00
R,F 999 1.00
T,F 25 45
There are over two hundred rendering demos with Principia you can use as templates to avoid having to re-invent the wheel.
There are several approaches for adding bumpmaps to reflective objects. The most accurate of them usually calculate a per-pixel reflection vector for sampling the environment map, based on the local normal perturbation. Because this operation involves an MxV in the pixel shader, accurate EMBM requires PS1.4 or higher.


Principia V3.0.0 Demos
EX03: Shadow map basic implementation, unfiltered FP32 shadow map
Bias=0.001, Zmax=255, PS20 triple test (zcomp, zero and facing)
FPS: 62.5
More rendering… Principia makes it easy to implement shadows in complex real-time worlds with many moving objects.
On
Off
Dir
Point
Spot
Glob
View
Lite1
Lite2
Lite3
Matl
cAmb 85 85 85
cDif 155 155 155
cSpt 200 200 200
POS 0.0 0.0 0.0
DIR -1.00 0.00 -1.00
1.00 0.00 0.00
R,F 999 1.00
T,F 25 45
Shadow mapping is a global two-pass technique for automatic rendering of projected and self-shadows. It renders the image in two direct passes without the need for geometry processing or overlays.
With the spreading penetration of PS20+ hardware, shadow mapping is becoming the technique of choice over SV for rendering shadows.


Principia V3.0.0 Demos
Ex01: Generic3A object, States=1, Comps=1, Meshes=1,
BhF=N, Animation=N, Spec=N
Principia objects encapsulate content with kinematics and animation. Principia features many types of objects.
Each one of these shuttles is a simple Generic3A object. The kinematics of this collection of objects is managed by a short procedure, connected with this page.
FPS: 6.2
GPS: 6.2
A- M 1 2
D- M 1 2
S- M 1 2
E- M 1 2
View
Lite1
Lite2
Lite3
Matl
Matl-Ambient color
255 255 255 255
Matl-Diffuse color
255 255 255 255
Matl-Specular color
255 255 255 255
Matl-Emissive color
0 0 0 0
Matl-Power
1.5


Principia V3.0.0 Demos
Ex02: Previs (Word Warror robot character) animation authoring contiguous sequences.
Source: Keyframed MAX bone animation.
KF: 724 [ICS_HEL_14]
FPS: 62.5
GPS: 66.7
CLK: 263.6
Character objects are always handy. They make many complex things seem easy.
Most of the time, our little robot is just idly peaching out. However, it will also get up, walk and do other things in response to select keystrokes.
Ex02 focuses on authoring the animation content for the WW2 game character, and on designing animation controls.
Low-level behavior, such as what the robot does while idling is automatically managed by the script. However, high-level changes are controlled by the game client. Ex02 provides a code section for test-driving high-level animation changes.
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
11.0
Camera Elevation
11.0
Camera Azimuth
45.0
Camera FoV
35.0
Look At Point
0.0
0.3
2.1


Principia V3.0.0 Demos
Ex01: Basic WorldRT3A, Simplest Foundational Capabilities Demo
GenProc=WrldRT3A_Gen1RL, Viewer=RT3A, DCT render paradigm, VS:1.1, PS:1.4
Enough rendering for now. Lets look at procedures. An entire world populated with objects is generated on the fly using scripted procedures.
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Many static objects are spread thru our world. While there is some structure to their spread, we have some way to go before there is a truly realistic-looking distribution.
Keep in mind, that a realistic distribution may not be in the best interest for strategy games, where concentration and some stylistic distribution can enhance gameplay,


Principia V3.0.0 Demos
Ex02: Basic WorldRT3A, Enhancement of Elevation Features
GenProc=WrldRT3A_Gen1RL, Viewer=RT3A, DCT render paradigm, VS:1.1, PS:1.4
Load: [11] [936] [366\10371]-[0]
Cell:
FPS: 21.3
GPS: 21.3
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
12.2
Camera Elevation
49.0
Camera Azimuth
18.0
Mountainous features protrude and reticulate in the ocean, as a by-product of erosion and proper texture UV mapping. All this is automatically generated!


Principia V3.0.0 Demos
Ex03: Basic WorldRT3A, Enhancement of Elevation Features
DataProc=WrldRT3A_Asm1RL, Viewer=RT3A, DCT render paradigm, VS:1.1, PS:
Load: [8] [479] [115\6695] [0]
Cell: [69,169, 1.101]
Note the good appearance of river beds and animated water flow. The water cascades down the terrain gradient steps naturally. Remember, all this is procedurally generated, using a sequence of terrain enhancement steps that started in Ex02.
GPS: 31.3
On
Off
Dir
Point
Spot
Glob
View
Lite1
Lite2
Lite3
Matl
cAmb 85 85 85
cDif 155 155 155
cSpt 100 100 100
POS 0.0 0.0 0.0
DIR -1.00 0.00 -1.00
Coef 1.00 0.00 0.00
R,F 999 1.00
T,F 25 45

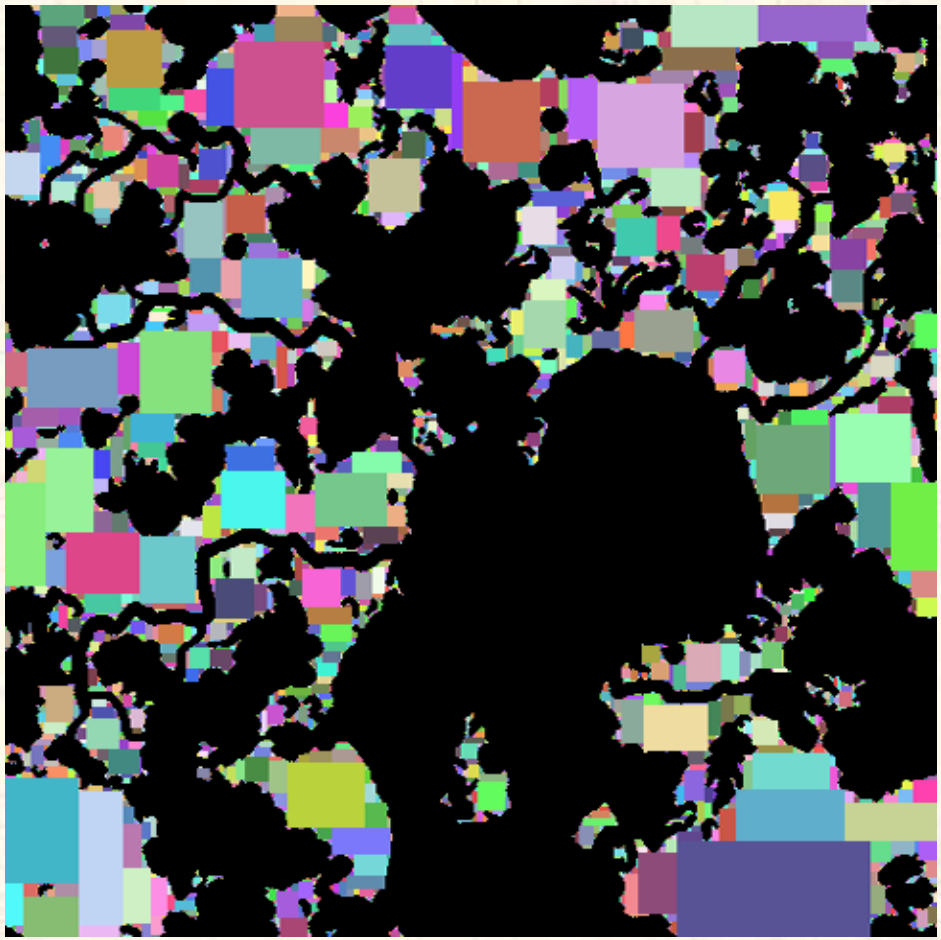

Principia V3.0.0 Demos
Ex04: Basic WorldRT3A, Enhancement of Elevation Features
GenProc=WrldRT3A_Asm1RL, Viewer=RT3A, DCT render paradigm, VS:1.1, P
Load: [3] [1945] [296\13634] [3][90][3]
Cell: [423,258, 0.265]
Character object can be selected by clicking on them, and then directed to walk to another location by means of another click. Selected walking characters will display their path (with automatic way-stations), which curve to avoid obstacles.
FPS: 62.5
GPS: 32.3
CLK: 274.2
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
6.0
Camera Elevation
39.0
Camera Azimuth
45.0
Camera FoV
Foundation code for object interaction, pathfinding and other world/user behaviors is built into Principia.

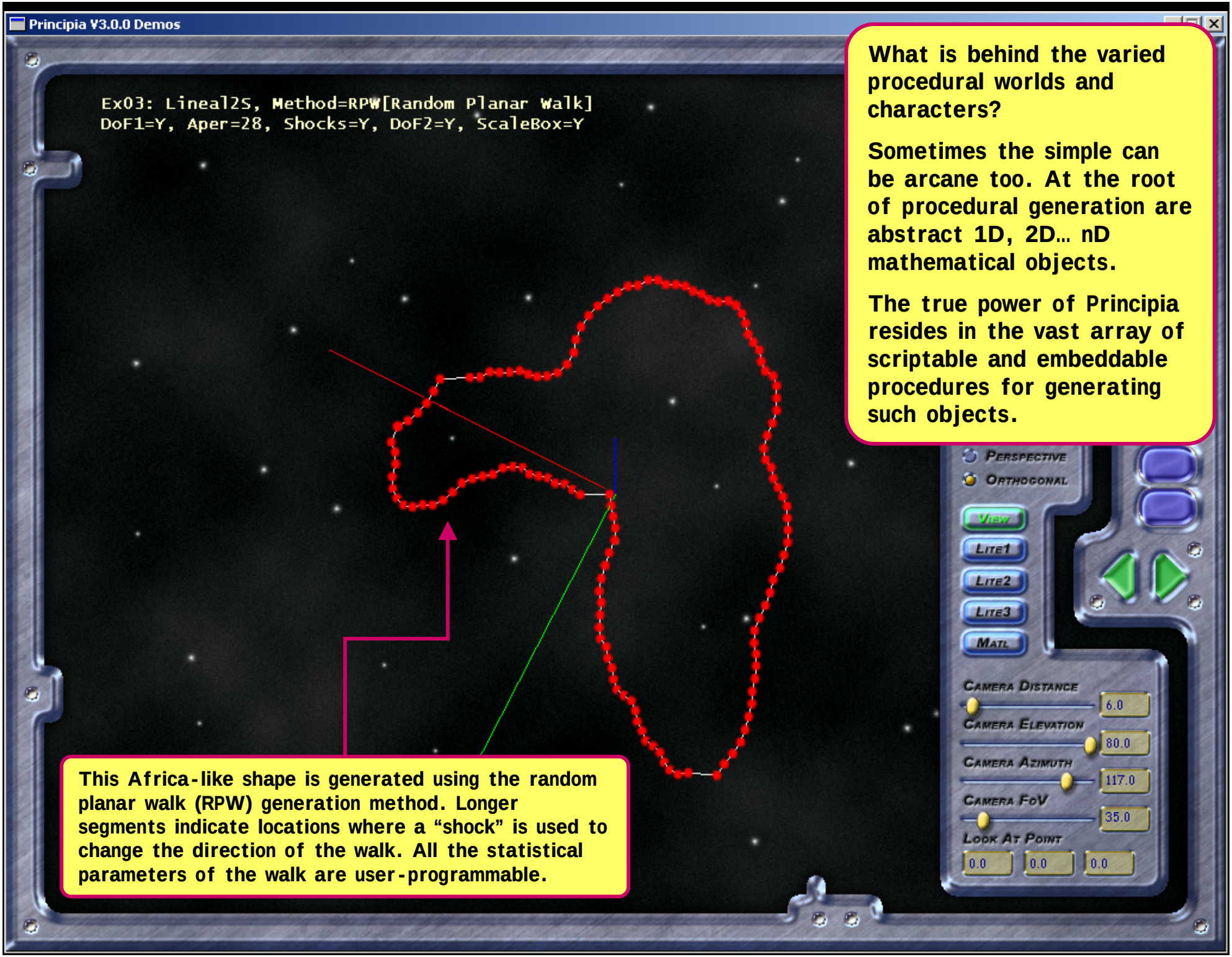
D108B:Ex04 – Worl

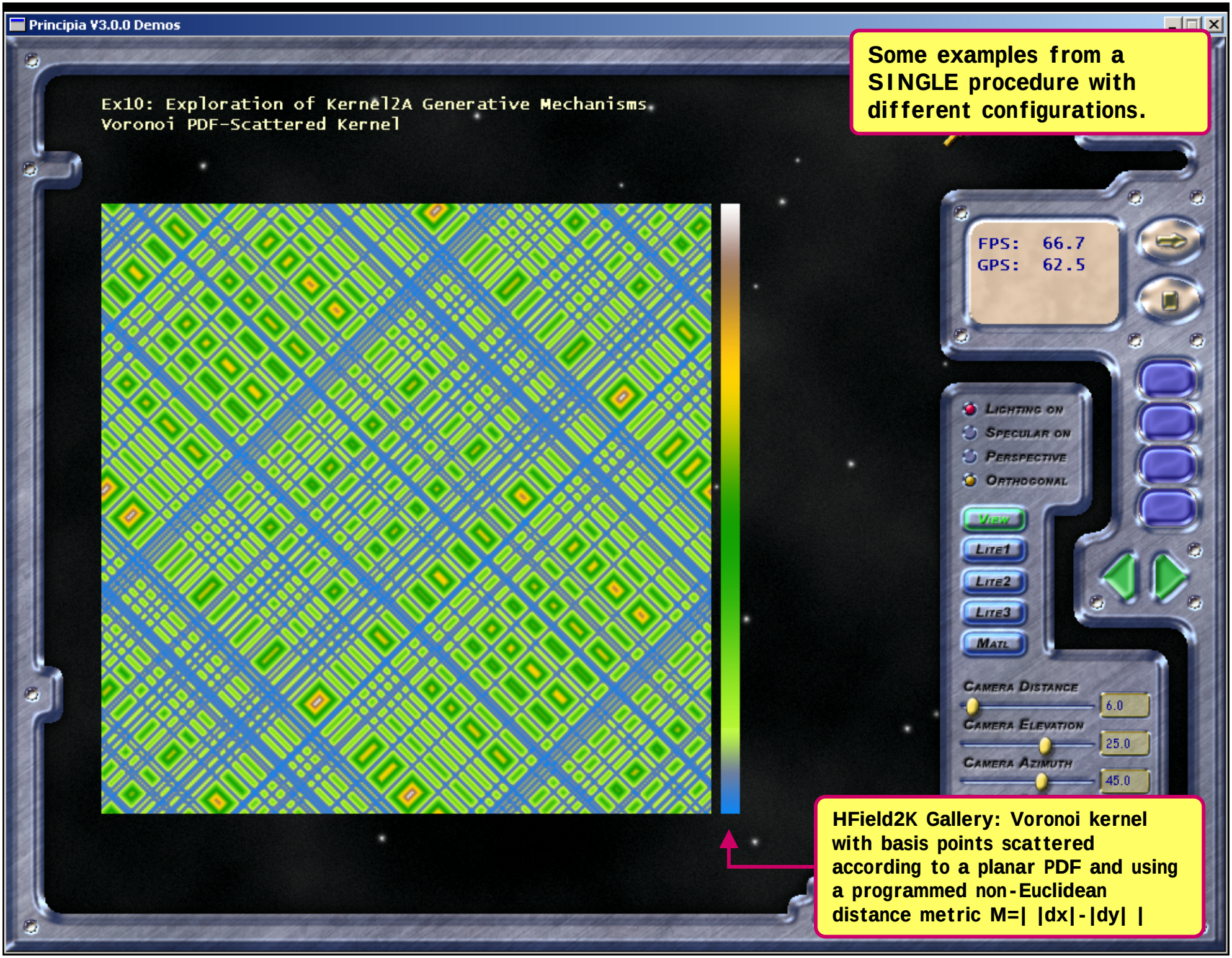
There are many path-finding methods. Principia has many procedures for generating path-finding data automatically for you.

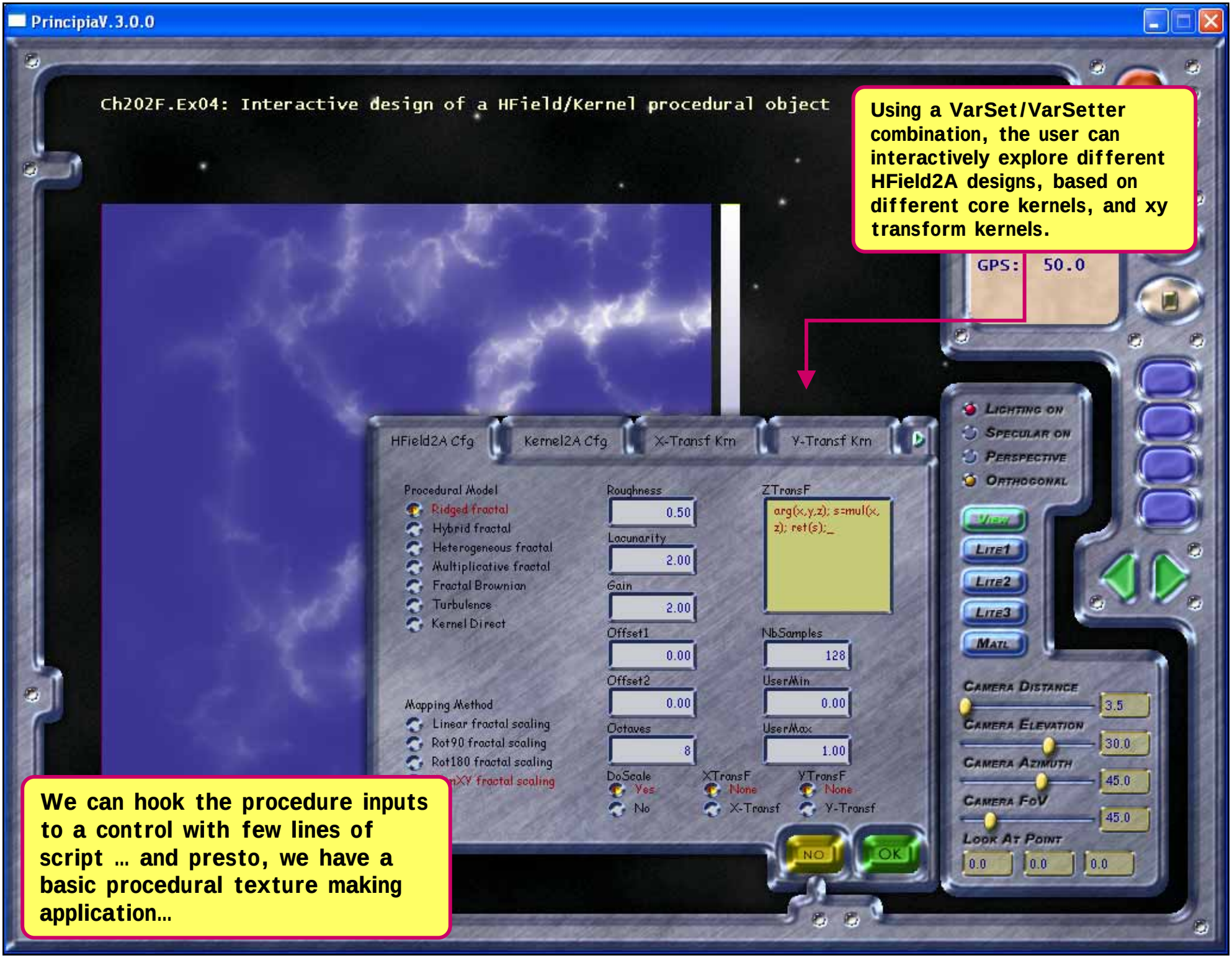
SVD design

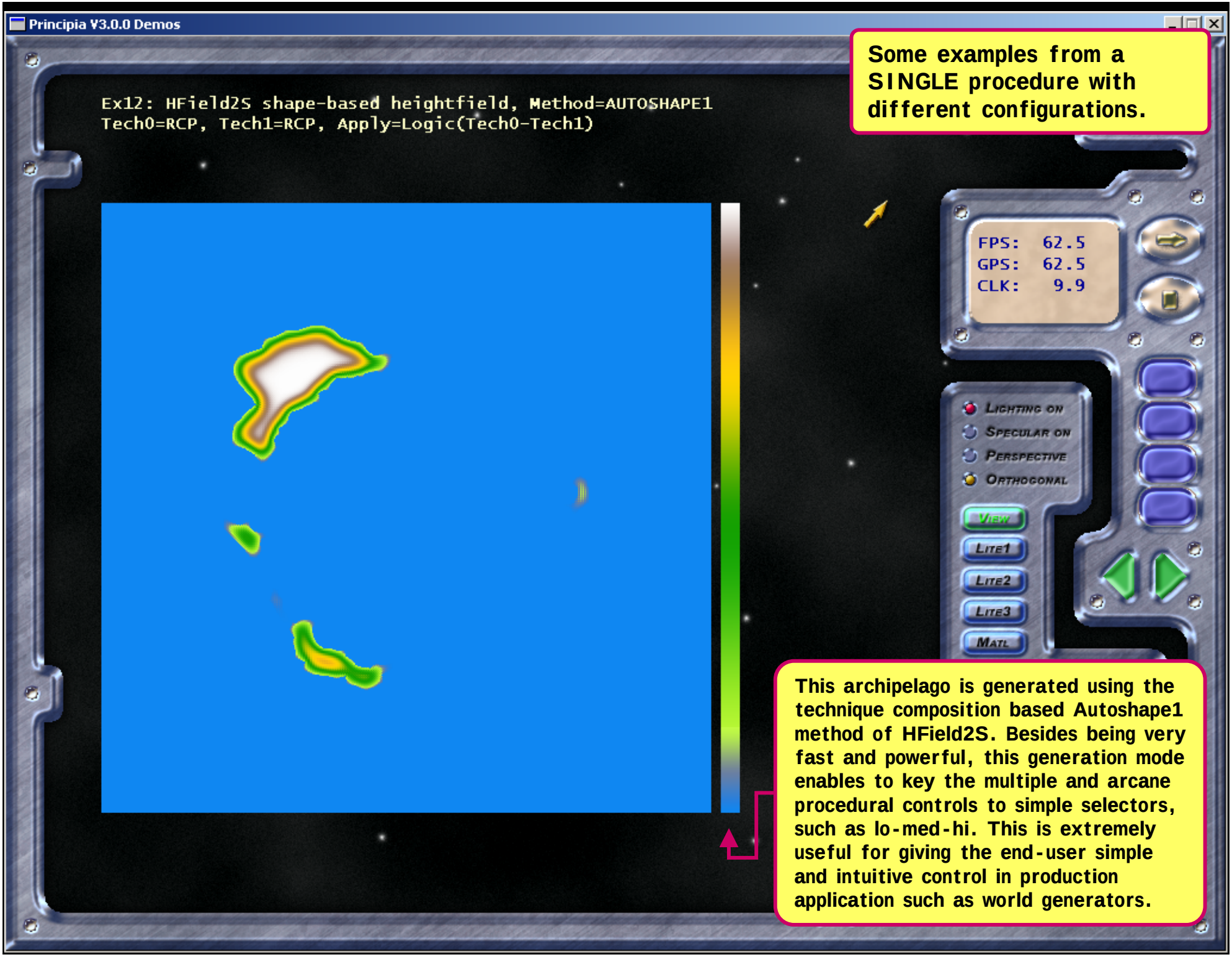
- Quilt of "walkable" rectangular regions, derived using GridT markings and Principia topology procedures.
- Allows lightning-fast and very optimal route finding of any length with or without waypoints or bridges.
- Good for diverse terrain with fixed obstacles.
- Bad for terrain with many small obstacles or many small navigability changes.

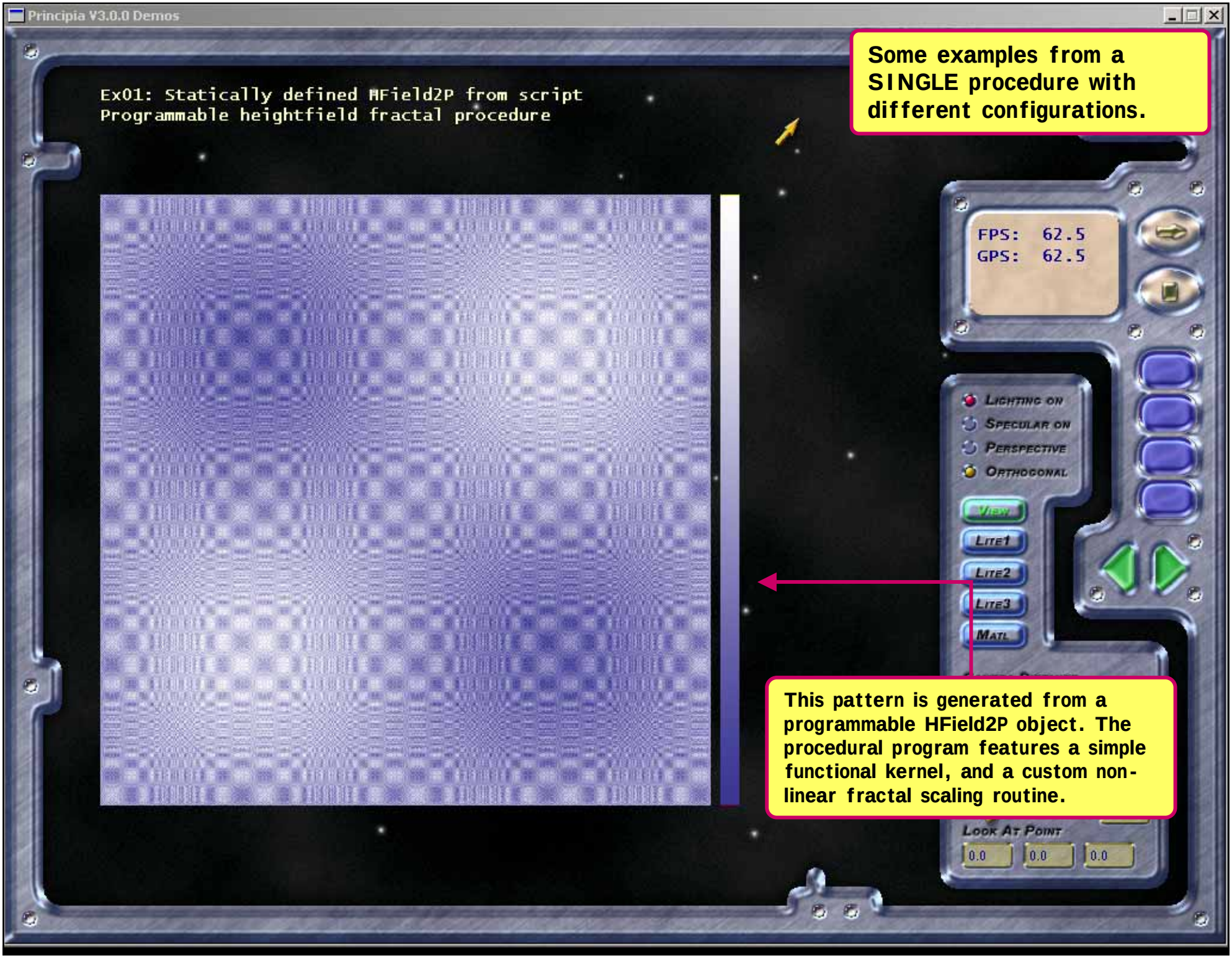


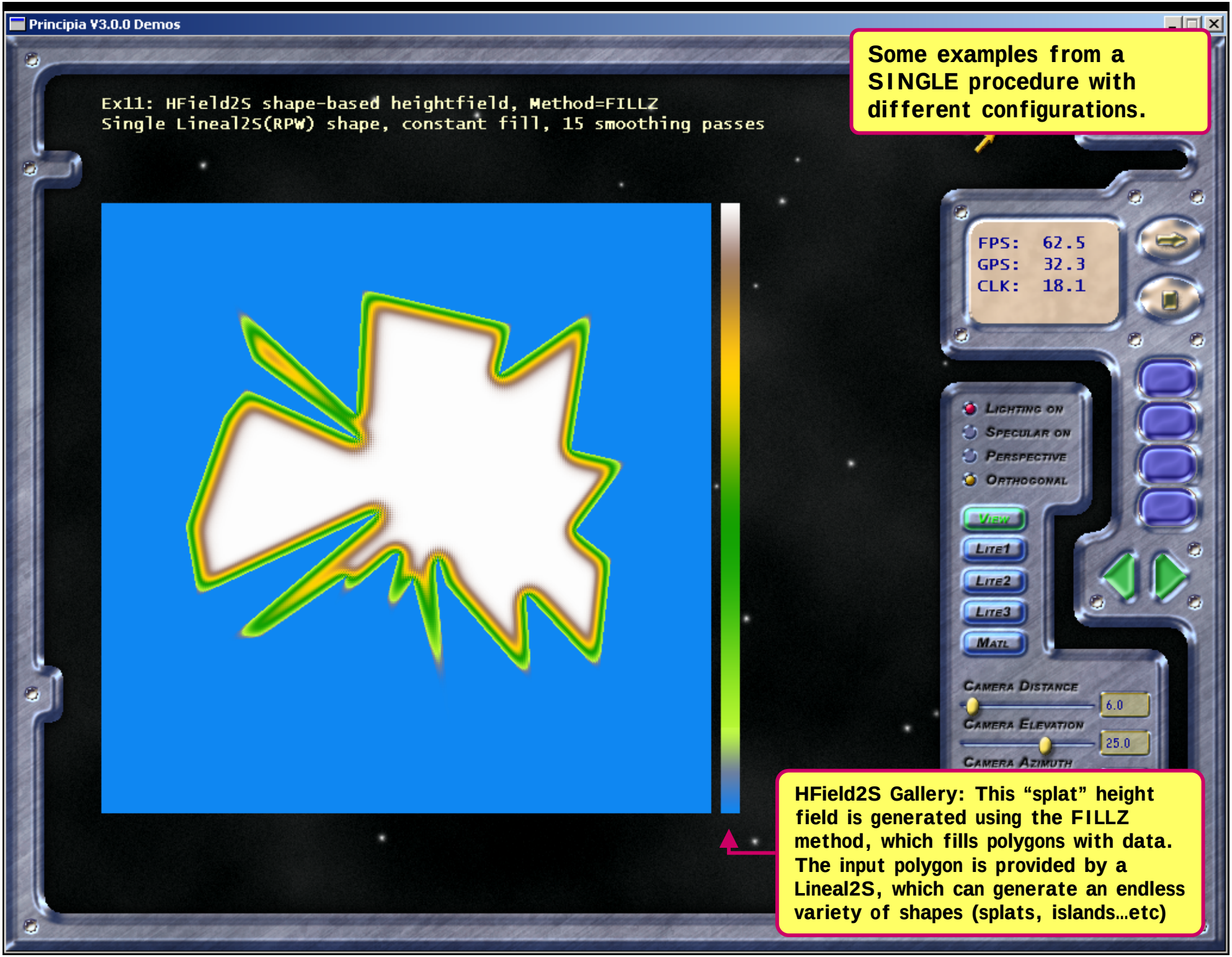

Principia V3.0.0 Demos
Ex03: Lineal2S, Method=RPW[Random Planar Walk]
DoF1=Y, Aper=28, Shocks=Y, DoF2=Y, ScaleBox=Y
What is behind the varied procedural worlds and characters?
Sometimes the simple can be arcane too. At the root of procedural generation are abstract 1D, 2D… nD mathematical objects.
The true power of Principia resides in the vast array of scriptable and embeddable procedures for generating such objects.
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
6.0
Camera Elevation
80.0
Camera Azimuth
117.0
Camera FoV
35.0
Look At Point
0.0
0.0
0.0
This Africa-like shape is generated using the random planar walk (RPW) generation method. Longer segments indicate locations where a “shock” is used to change the direction of the walk. All the statistical parameters of the walk are user-programmable.


Principia V3.0.0 Demos
Ex10: Exploration of Kernel2A Generative Mechanisms.
Voronoi PDF-Scattered Kernel
Some examples from a SINGLE procedure with different configurations.
FPS: 66.7
GPS: 62.5
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
6.0
Camera Elevation
25.0
Camera Azimuth
45.0
HField2K Gallery: Voronoi kernel with basis points scattered according to a planar PDF and using a programmed non-Euclidean distance metric M=| |dx|-|dy| |


PrincipiaV.3.0.0
Ch202F.Ex04: Interactive design of a HField/Kernel procedural object
Using a VarSet/VarSetter combination, the user can interactively explore different HField2A designs, based on different core kernels, and xy transform kernels.
GPS: 50.0
HField2A Cfg
Kernel2A Cfg
X-Transf Krn
Y-Transf Krn
Procedural Model
Ridged fractal
Hybrid fractal
Heterogeneous fractal
Multiplicative fractal
Fractal Brownian
Turbulence
Kernel Direct
Roughness
0.50
Lacunarity
2.00
Gain
2.00
Offset1
0.00
Offset2
0.00
Octaves
8
ZTransF
arg(x,y,z); s=mul(x, z); ret(s);_
NbSamples
128
UserMin
0.00
UserMax
1.00
Mapping Method
Linear fractal scaling
Rot90 fractal scaling
Rot180 fractal scaling
DoScale
Yes
No
XTransF
None
X-Transf
YTransF
None
Y-Transf
NO
OK
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
3.5
Camera Elevation
30.0
Camera Azimuth
45.0
Camera FoV
45.0
Look At Point
0.0
0.0
0.0
We can hook the procedure inputs to a control with few lines of script … and presto, we have a basic procedural texture making application…


Principia V3.0.0 Demos
Ex12: HField2S shape-based heightfield, Method=AUTOSHAPE1
Tech0=RCP, Tech1=RCP, Apply=Logic(Tech0-Tech1)
Some examples from a SINGLE procedure with different configurations.
FPS: 62.5
GPS: 62.5
CLK: 9.9
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
This archipelago is generated using the technique composition based Autoshape1 method of HField2S. Besides being very fast and powerful, this generation mode enables to key the multiple and arcane procedural controls to simple selectors, such as lo-med-hi. This is extremely useful for giving the end-user simple and intuitive control in production application such as world generators.


Principia V3.0.0 Demos
Ex01: Statically defined HField2P from script
Programmable heightfield fractal procedure
Some examples from a SINGLE procedure with different configurations.
FPS: 62.5
GPS: 62.5
Lighting On
Specular On
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Look At Point
0.0
0.0
0.0
This pattern is generated from a programmable HField2P object. The procedural program features a simple functional kernel, and a custom non-linear fractal scaling routine.


Principia V3.0.0 Demos
Ex11: HField2S shape-based heightfield, Method=FILLZ
Single Lineal2S(RPW) shape, constant fill, 15 smoothing passes
Some examples from a SINGLE procedure with different configurations.
FPS: 62.5
GPS: 32.3
CLK: 18.1
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
6.0
Camera Elevation
25.0
Camera Azimuth
HField2S Gallery: This “splat” height field is generated using the FILLZ method, which fills polygons with data. The input polygon is provided by a Lineal2S, which can generate an endless variety of shapes (splats, islands…etc)

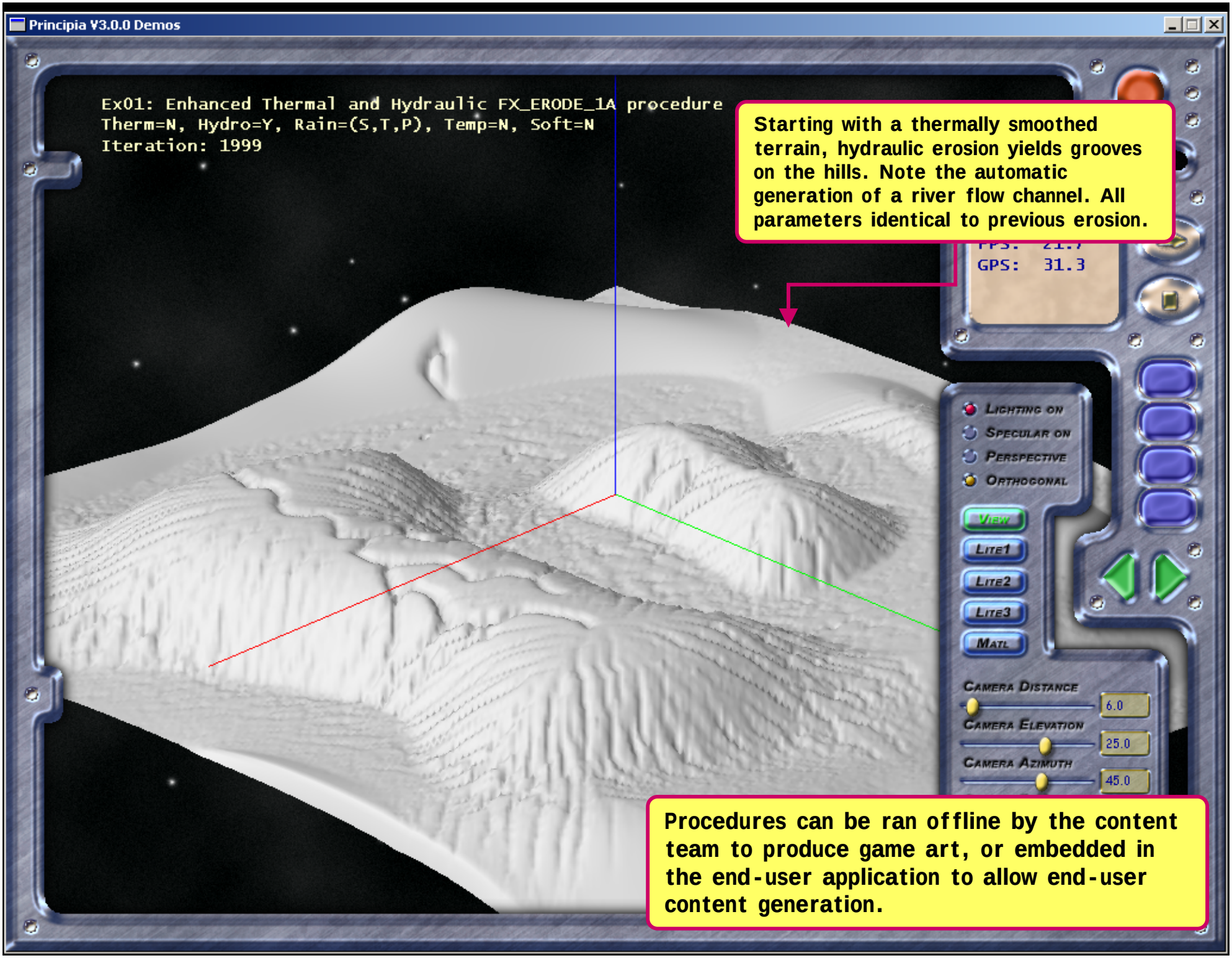
D303D:Ex02 – TexGenDct1A

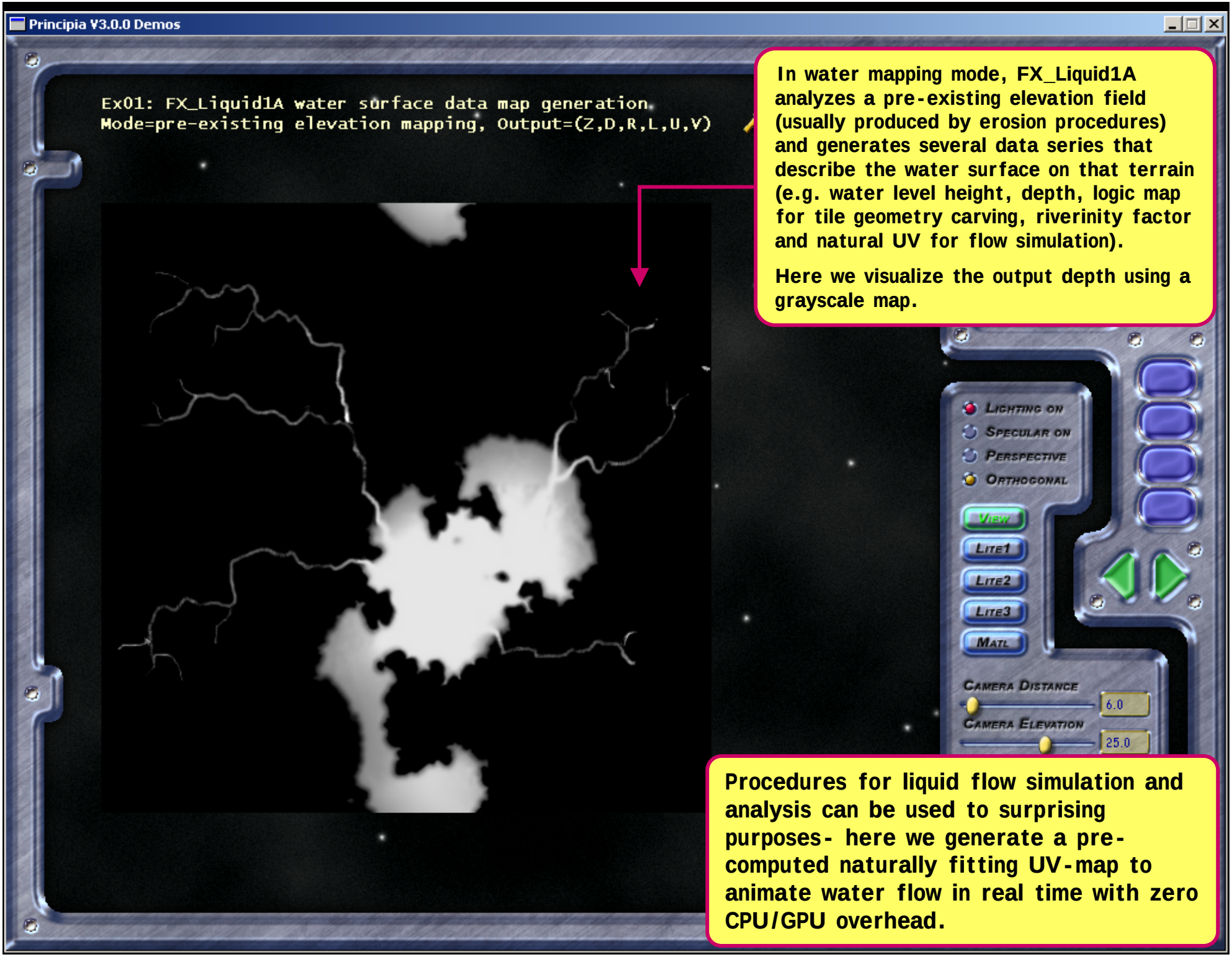
Slope distribution, clearly showing

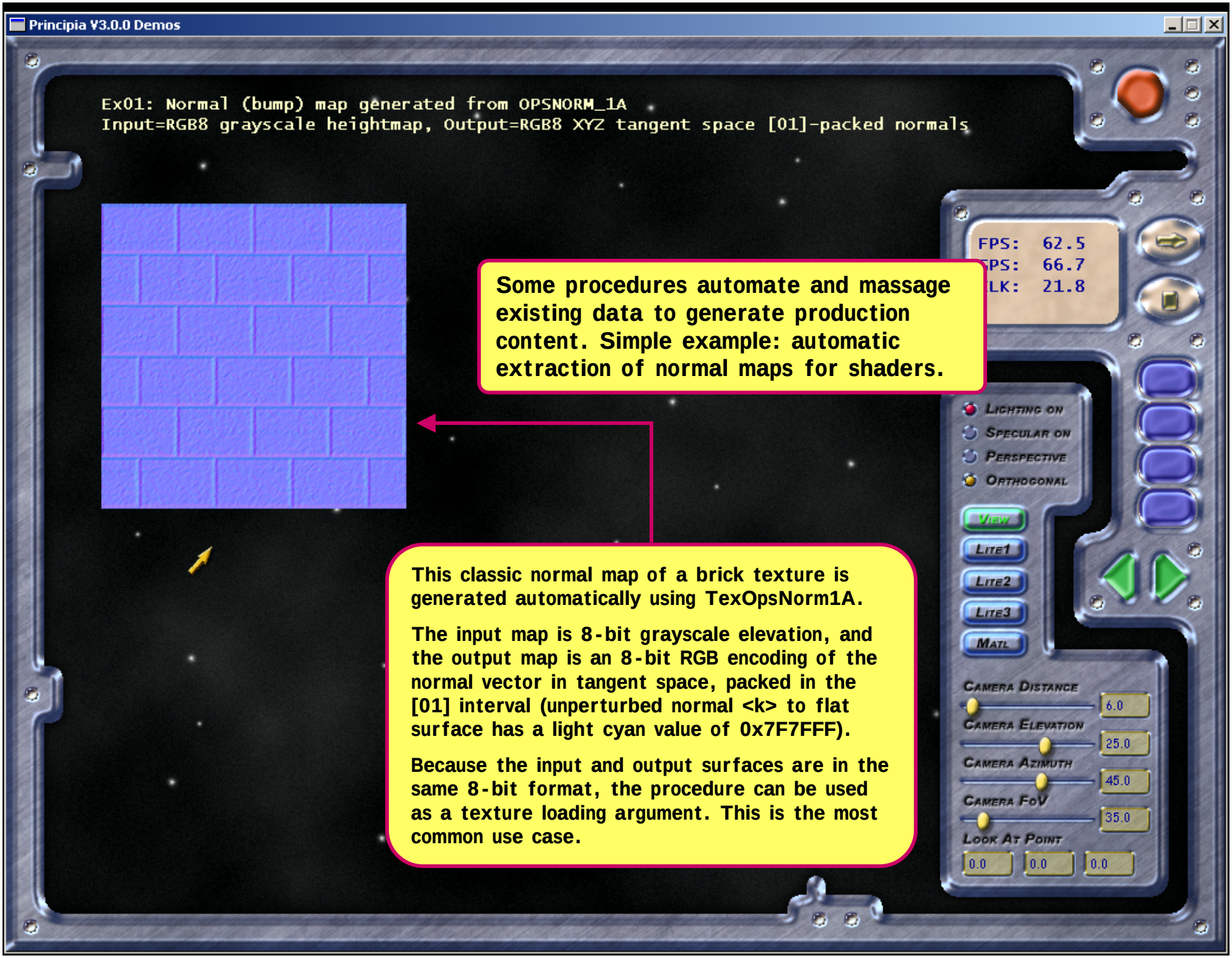
- Taluses
- Canyons
- Riverbeds
- Erosion fans
- Sedimentary plains
- High plateaux

The elevation is produced using FX_Erode1A proc.

Not all procedures are simple. Here is a terrain generation procedure that emulates terrain erosion and produces an elevation map with realistic geologic features.

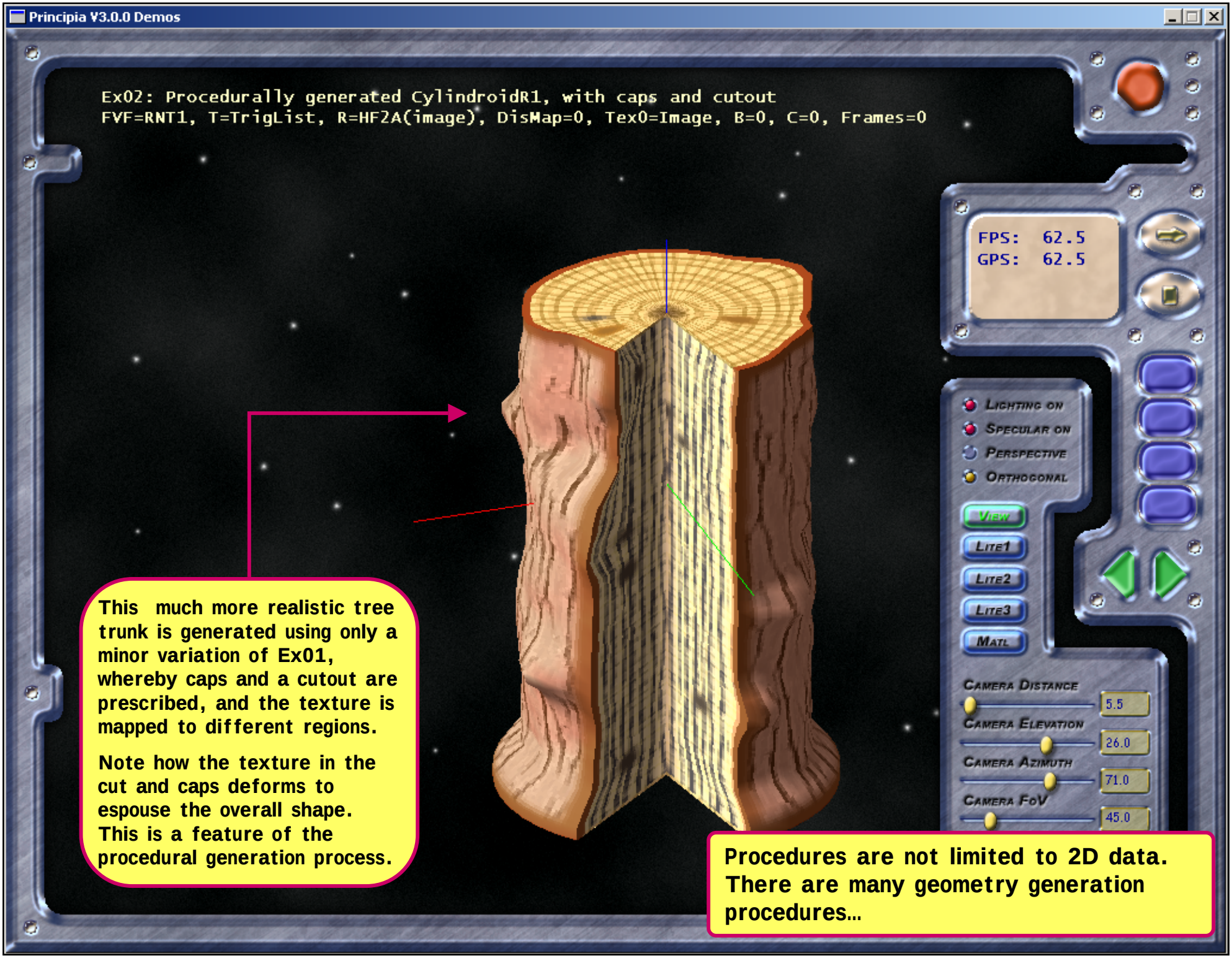

Principia V3.0.0 Demos
Ex01: Enhanced Thermal and Hydraulic FX_ERODE_1A procedure
Therm=N, Hydro=Y, Rain=(S,T,P), Temp=N, Soft=N
Iteration: 1999
Starting with a thermally smoothed terrain, hydraulic erosion yields grooves on the hills. Note the automatic generation of a river flow channel. All parameters identical to previous erosion.
GPS: 31.3
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
6.0
Camera Elevation
25.0
Camera Azimuth
45.0
Procedures can be ran offline by the content team to produce game art, or embedded in the end-user application to allow end-user content generation.

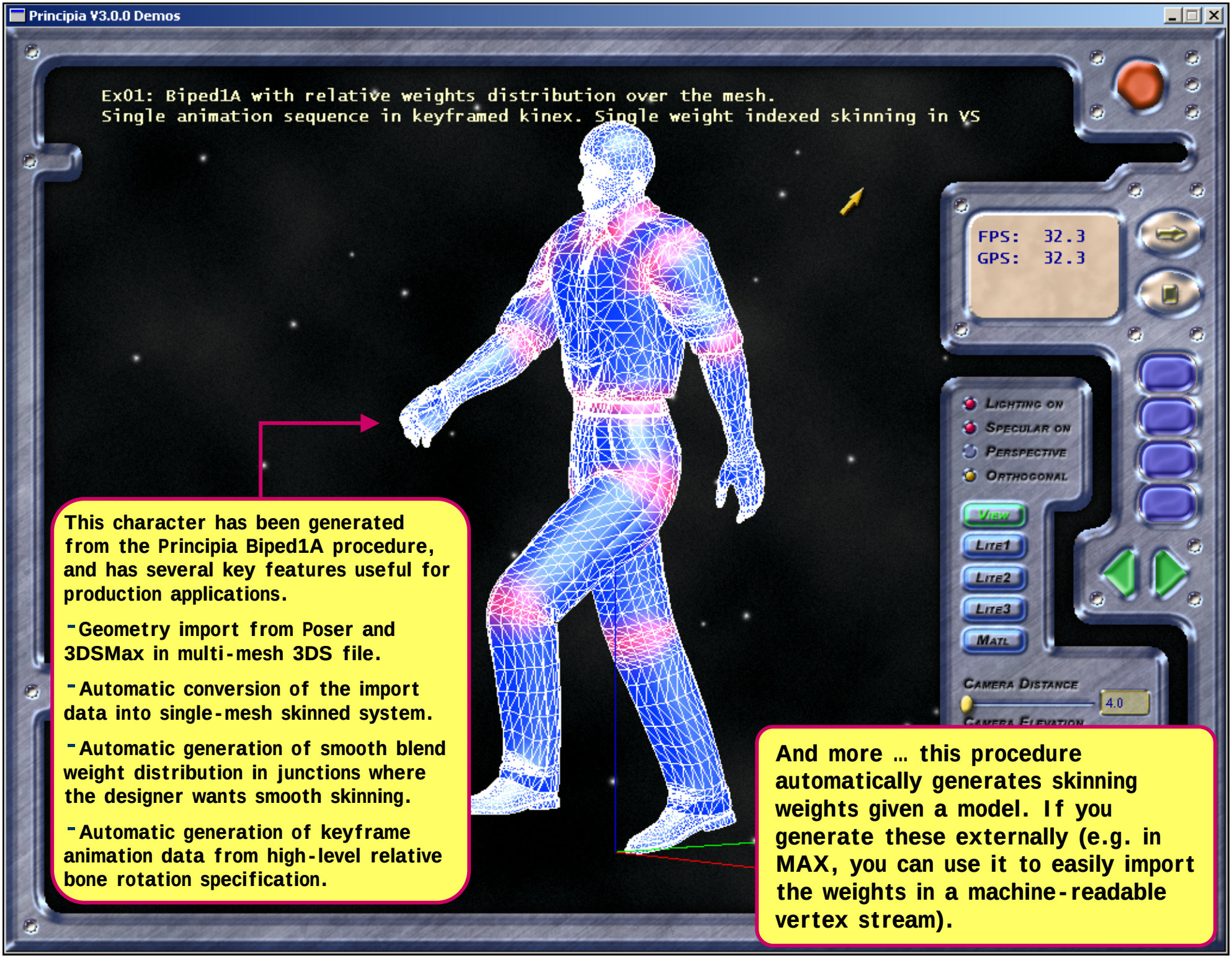

Principia V3.0.0 Demos
Ex01: FX_Liquid1A water surface data map generation.
Mode=pre-existing elevation mapping, Output=(Z,D,R,L,U,V)
In water mapping mode, FX_Liquid1A analyzes a pre-existing elevation field (usually produced by erosion procedures) and generates several data series that describe the water surface on that terrain (e.g. water level height, depth, logic map for tile geometry carving, riverinity factor and natural UV for flow simulation).
Here we visualize the output depth using a grayscale map.
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
6.0
Camera Elevation
25.0
Procedures for liquid flow simulation and analysis can be used to surprising purposes- here we generate a pre-computed naturally fitting UV-map to animate water flow in real time with zero CPU/GPU overhead.


Principia V3.0.0 Demos
Ex01: Normal (bump) map generated from OPSNORM_1A
Input=RGB8 grayscale heightmap, Output=RGB8 XYZ tangent space [01]-packed normals
Some procedures automate and massage existing data to generate production content. Simple example: automatic extraction of normal maps for shaders.
This classic normal map of a brick texture is generated automatically using TexOpsNorm1A.
The input map is 8-bit grayscale elevation, and the output map is an 8-bit RGB encoding of the normal vector in tangent space, packed in the [01] interval (unperturbed normal <k> to flat surface has a light cyan value of 0x7F7FFF).
Because the input and output surfaces are in the same 8-bit format, the procedure can be used as a texture loading argument. This is the most common use case.
FPS: 62.5
PS: 66.7
LK: 21.8
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
6.0
Camera Elevation
25.0
Camera Azimuth
45.0
Camera FoV
35.0
Look At Point
0.0
0.0
0.0


Principia V3.0.0 Demos
Ex03: TexWrk_WangExtract3A procedure, automatic atlas-based generation of Wang set from pri
WangMode=Corner, WangPrimaries=3, Layers=3, Masks=2(from Photoshop), AtlasEx=Y
In combination with FX_WangPasm3A from Ex02, the FX_WangExtract3A procedure has a layer mode, which can completely automate the generation of Wang texture atlases and the extraction of members, given only the grayscale atlas masks from Photoshop. This saves a lot of time…
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
6.0
Camera Elevation
25.0
Camera Azimuth
45.0
Camera FoV
35.0
Some data production procedures may seem arcane, but they can save days and even weeks of work !!!

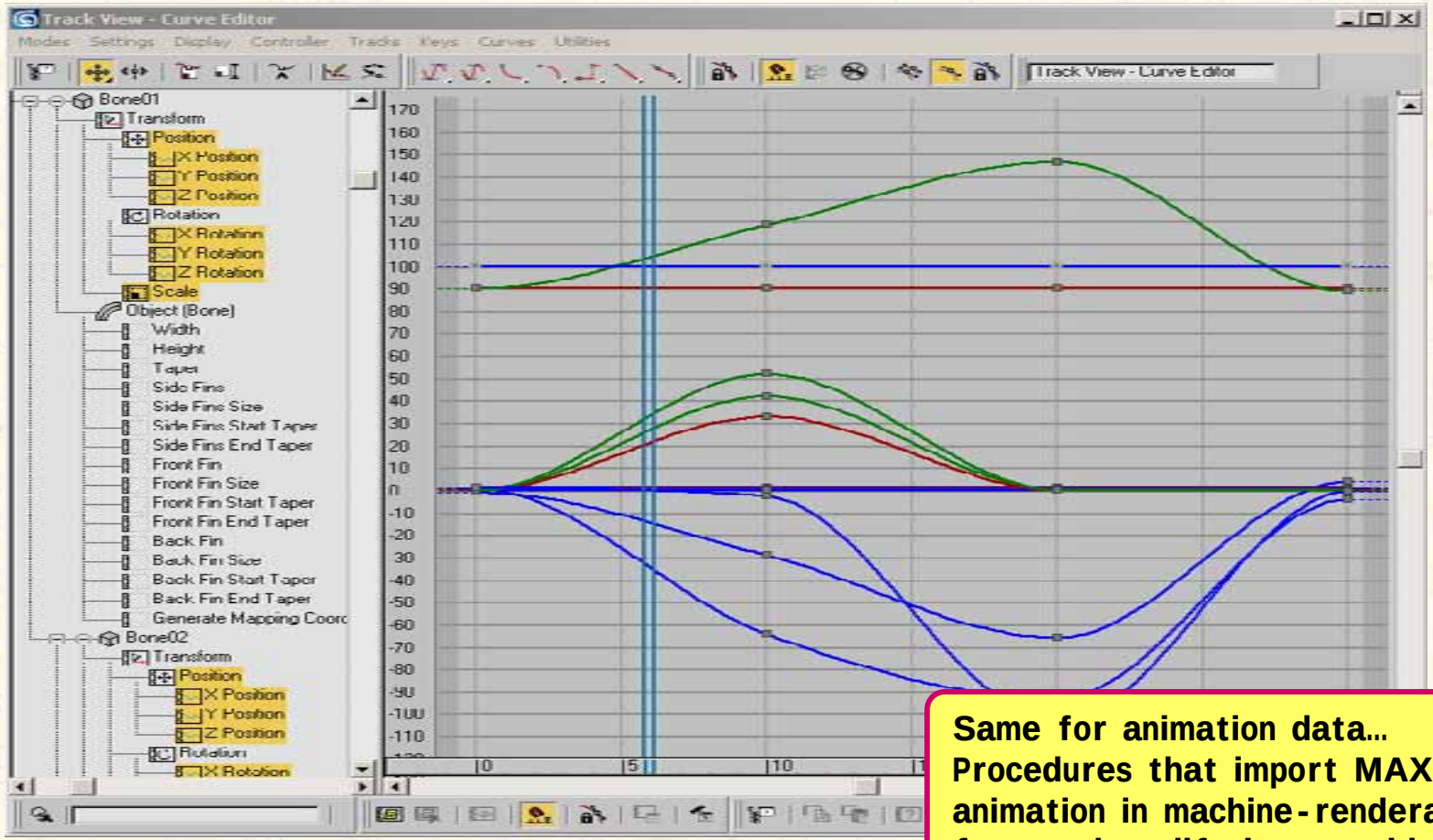

Principia V3.0.0 Demos
Ex04: TexWrk_MaskExtract3A procedure, true Alpha mask extraction from Photoshop samples.
Blend mode=Normal, Samples at 0,50,100 base
Ever made a nice interface in Photoshop with glass buttons, shadows…etc.; and wasted hours trying to get the alpha and color channels in your textures right so that the game engine renders a result that looks like the original design?
The Principia *MaskExtract3A procedure is designed to take series of Photoshop-ed elements sampled with different backgrounds, and generate an RGBA texture mask that replicates the shadows, glows and glints you see in Photoshop.
It is a HUGE time saver!!!
DIFFICULTY
SIZE
PLAYERS
VICTORY CONDITIONS
MATL
CAMERA DISTANCE
11.0
CAMERA ELEVATION
0.0
CAMERA AZIMUTH
0.0
CAMERA FOV
35.0
LOOK AT POINT
0.0
0.0
2.1


Principia V3.0.0 Demos
Ex02: Procedurally generated CylindroidR1, with caps and cutout
FVF=RNT1, T=TrigList, R=HF2A(image), DisMap=0, Tex0=Image, B=0, C=0, Frames=0
FPS: 62.5
GPS: 62.5
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
5.5
Camera Elevation
26.0
Camera Azimuth
71.0
Camera FoV
45.0
This much more realistic tree trunk is generated using only a minor variation of Ex01, whereby caps and a cutout are prescribed, and the texture is mapped to different regions.
Note how the texture in the cut and caps deforms to espouse the overall shape. This is a feature of the procedural generation process.
Procedures are not limited to 2D data. There are many geometry generation procedures…


Principia V3.0.0 Demos
Ex01: Biped1A with relative weights distribution over the mesh.
Single animation sequence in keyframed kinex. Single weight indexed skinning in VS
FPS: 32.3
GPS: 32.3
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
4.0
This character has been generated from the Principia Biped1A procedure, and has several key features useful for production applications.
- Geometry import from Poser and 3DSMax in multi-mesh 3DS file.
- Automatic conversion of the import data into single-mesh skinned system.
- Automatic generation of smooth blend weight distribution in junctions where the designer wants smooth skinning.
- Automatic generation of keyframe animation data from high-level relative bone rotation specification.
And more … this procedure automatically generates skinning weights given a model. If you generate these externally (e.g. in MAX, you can use it to easily import the weights in a machine-readable vertex stream).

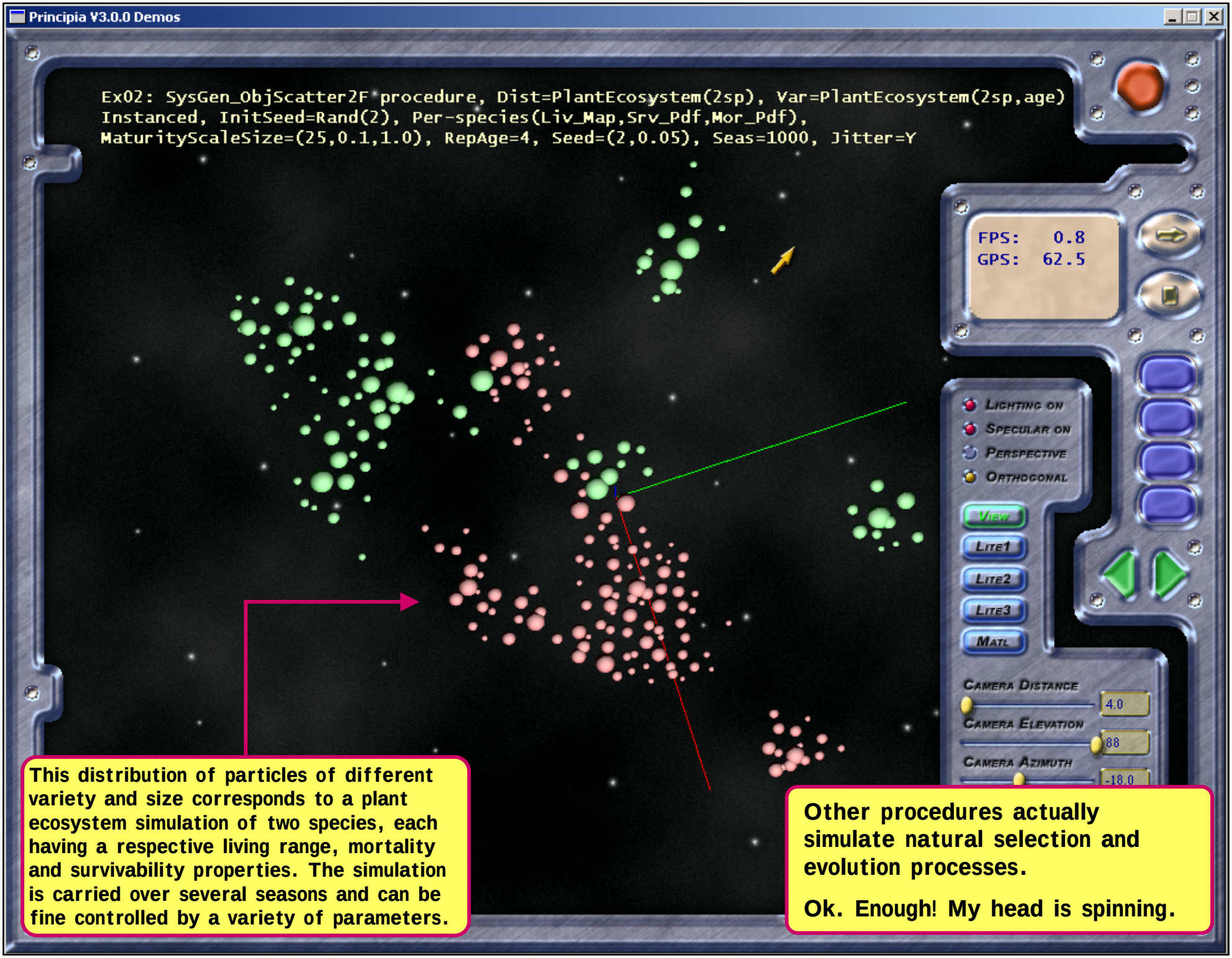
D306A:Ex06 – DataBoned1A

- **Multiple bone rotation keys in MAX curve sheet**



Same for animation data… Procedures that import MAX animation in machine-renderable form and modify it are a big time-saver for your production.


Principia V3.0.0 Demos
Ex01: SysGen_ObjScatter2F procedure, Dist=2D_Pos_PDF(sX,sY), Var=1D_Freq_PDF(X,Y,I)
Instanced, Samp(XY)=PrevPos+Rnd, Grid=.00/.00, MinMaxR=.00/.00, Jitter=N, MaxCell=1, MaxTem
More procedures…this time on particle generation. Most Principia procedures are built upon a common mathematical framework of plug-and-play mathematical objects. This makes them extremely powerful and flexible.
FPS: 1.0
GPS: 62.5
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
4.0
Camera Elevation
85
Camera Azimuth
45.0
Camera FoV
45.0
Look At Point
-0.5
-0.3
0.0
The functional form of the distribution maps is a powerful Principia paradigm for designing almost any type of scatter. Here, we use the fact that the position sampling functions PDF_Pos() take as argument the last point position, to generate a scatter based on 2D random walk.


Principia V3.0.0 Demos
Ex02: SysGen_ObjScatter2F procedure, Dist=PlantEcosystem(2sp), Var=PlantEcosystem(2sp,age)
Instanced, InitSeed=Rand(2), Per-species(Liv_Map,Srv_Pdf,Mor_Pdf),
MaturityScaleSize=(25,0.1,1.0), RepAge=4, Seed=(2,0.05), Seas=1000, Jitter=Y
FPS: 0.8
GPS: 62.5
Lighting on
Specular on
Perspective
Orthogonal
View
Lite1
Lite2
Lite3
Matl
Camera Distance
4.0
Camera Elevation
88
Camera Azimuth
-18.0
This distribution of particles of different variety and size corresponds to a plant ecosystem simulation of two species, each having a respective living range, mortality and survivability properties. The simulation is carried over several seasons and can be fine controlled by a variety of parameters.
Other procedures actually simulate natural selection and evolution processes.
Ok. Enough! My head is spinning.

Why Principia ?

- **Cheaper**
  - Using Principia can reduce the end-to-end production cost of digital entertainment software by up to 40%
- **Faster**
  - Principia can cut development time by up to 70% and multimedia content authoring time by up to 30%
- **Better**
  - With Principia, product is up to 50% less likely to contain critical bugs, and offers a much deeper multimedia experience.

What Is Principia ?

An essential ingredient for your success as a producer of commercial entertainment software and content

Legal Notices

- **A General Principia V3 End-User License Agreement (EULA) has been provided as attachment to the present documentation and digital media. *If this EULA is missing, do not proceed further, and do advise Western Star Entertainment Ltd of the situation.***
- **The present material is provided to you under the terms of the General Principia V3 EULA. In particular, you may not:**
  - **Disseminate this material in any form or fashion without Western Star's express written permission.**
  - **Use Principia V3 or any part of this material in developing commercial products without Western Star's express written permission and specific development license.**
- **Principia is fully compliant with all Microsoft DirectX, Microsoft SDK and OpenGL EULAs. If you do not intend to abide by all relevant third-party EULAs, do not use Principia and please do dispose of all attached software and documentation materials.**


Introduction
To Principia
Principia v3.0
Reference Series
(c) Western Star
Entertainment Ltd, 2003-2006