Shadowbuffers

Tom Forsyth
RAD Game Tools

Terminology

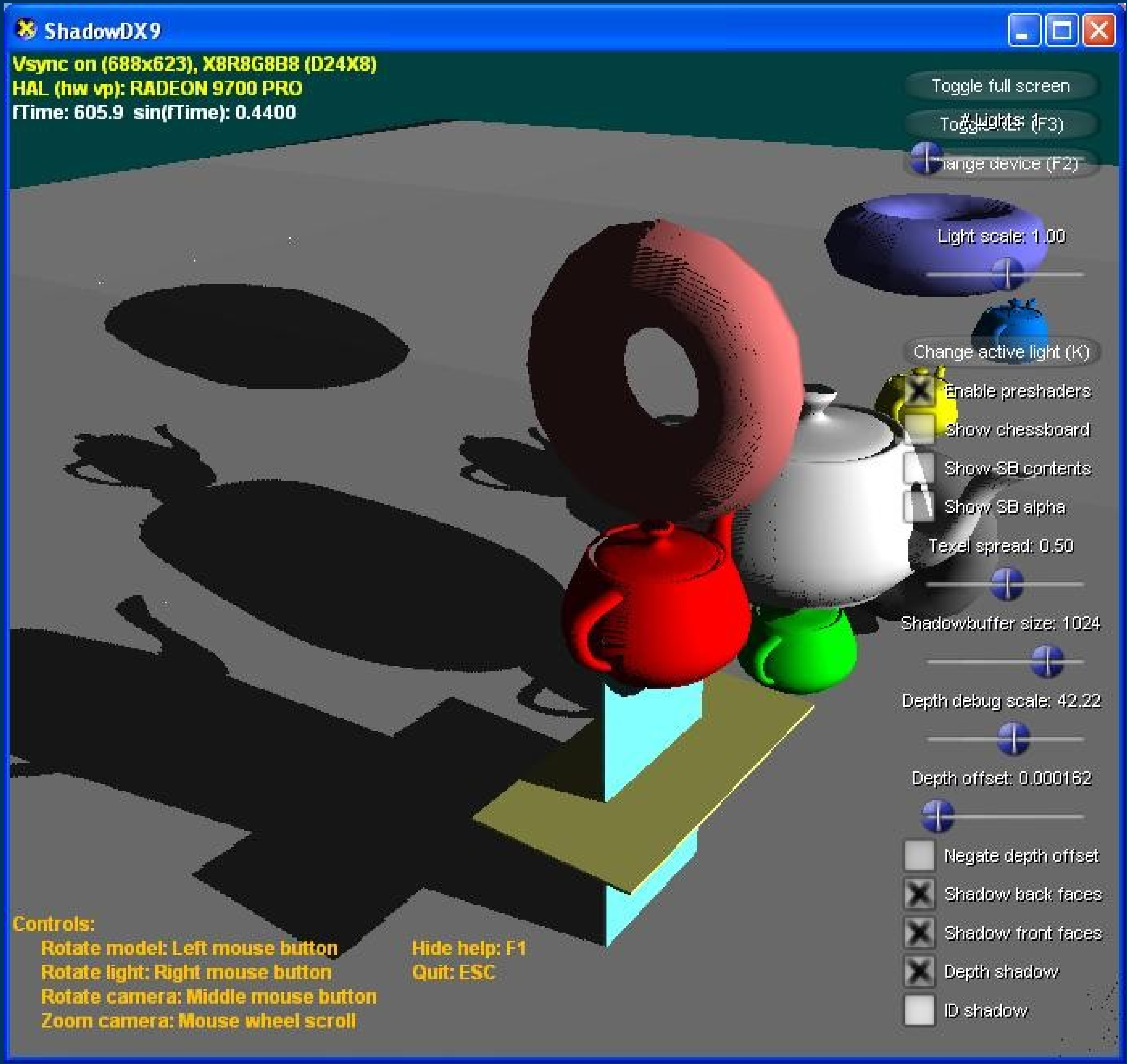
- View/viewspace = current D3D target
- Light POV = light point of view
  - Placing the viewer where the light is
- Camera POV = what the gamer sees
  - Conventional idea of a viewer
- Shadowbuffer/shadowmap
  - Both the same thing
  - To me, "buffer" = dynamic, "map" = static
- Other people use other conventions!
  - (so do I sometimes - oops!)

Principles

- Render scene from light point of view
  - Render to shadowbuffer texture
  - Store a surface identifier
    - Depth, ID, whatever
  - Use standard Z-buffer occlusion
  - Closest thing to light is stored in buffer
    - By definition, it can see the light = it is lit
  - All things behind it are invisible to light
    - Can't see the light = in shadow
  - So surface ID in shadowbuffer is the lit one

Principles 2

- Render scene from camera POV
  - Project shadowbuffer texture
  - Same projection maths as previous pass
    - Scale 0-512 pixels to 0-1 UV coords
  - Compute the surface ID the same way
  - Read the shadowbuffer
  - Compare computed & read IDs
  - If they match, this surface can "see" the light
    - So it's drawn lit
  - If not, something in the way, so draw shadowed



```
Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 605.9  sin(fTime): 0.4400
```

Lights: 1

Light scale: 1.00

☒ Enable preshaders

□ Show chessboard

□ Show SB contents

□ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 42.22

Depth offset: 0.000162

□ Negate depth offset

☒ Shadow back faces

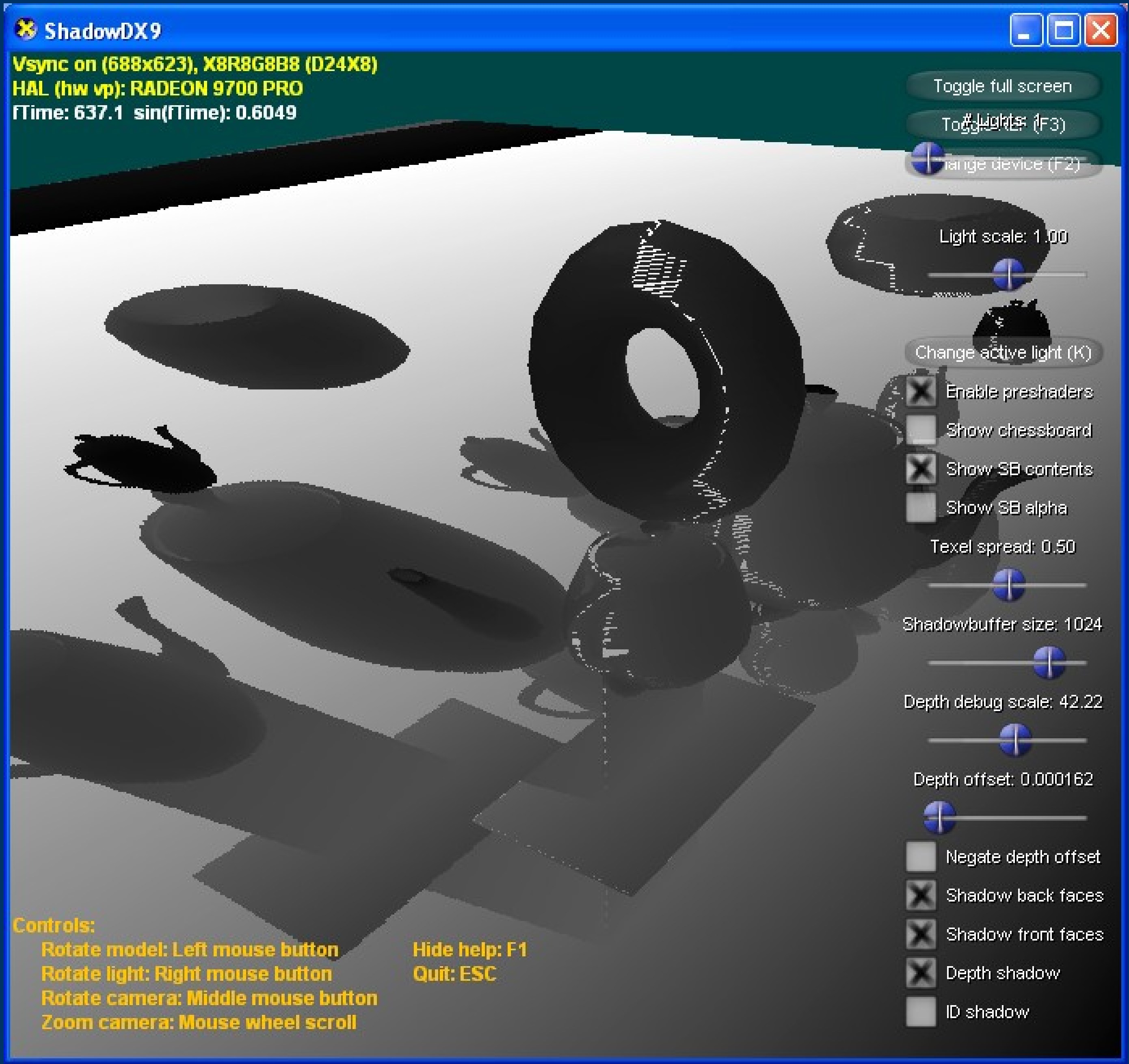
☒ Shadow front faces

☒ Depth shadow

□ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll

Hide help: F1
Quit: ESC



Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 637.1 sin(fTime): 0.6049

Lights: 1

Light scale: 1.00

☒ Enable preshaders
☐ Show chessboard
☒ Show SB contents
☐ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 42.22

Depth offset: 0.000162

☐ Negate depth offset
☒ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
☐ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll

Hide help: F1
Quit: ESC

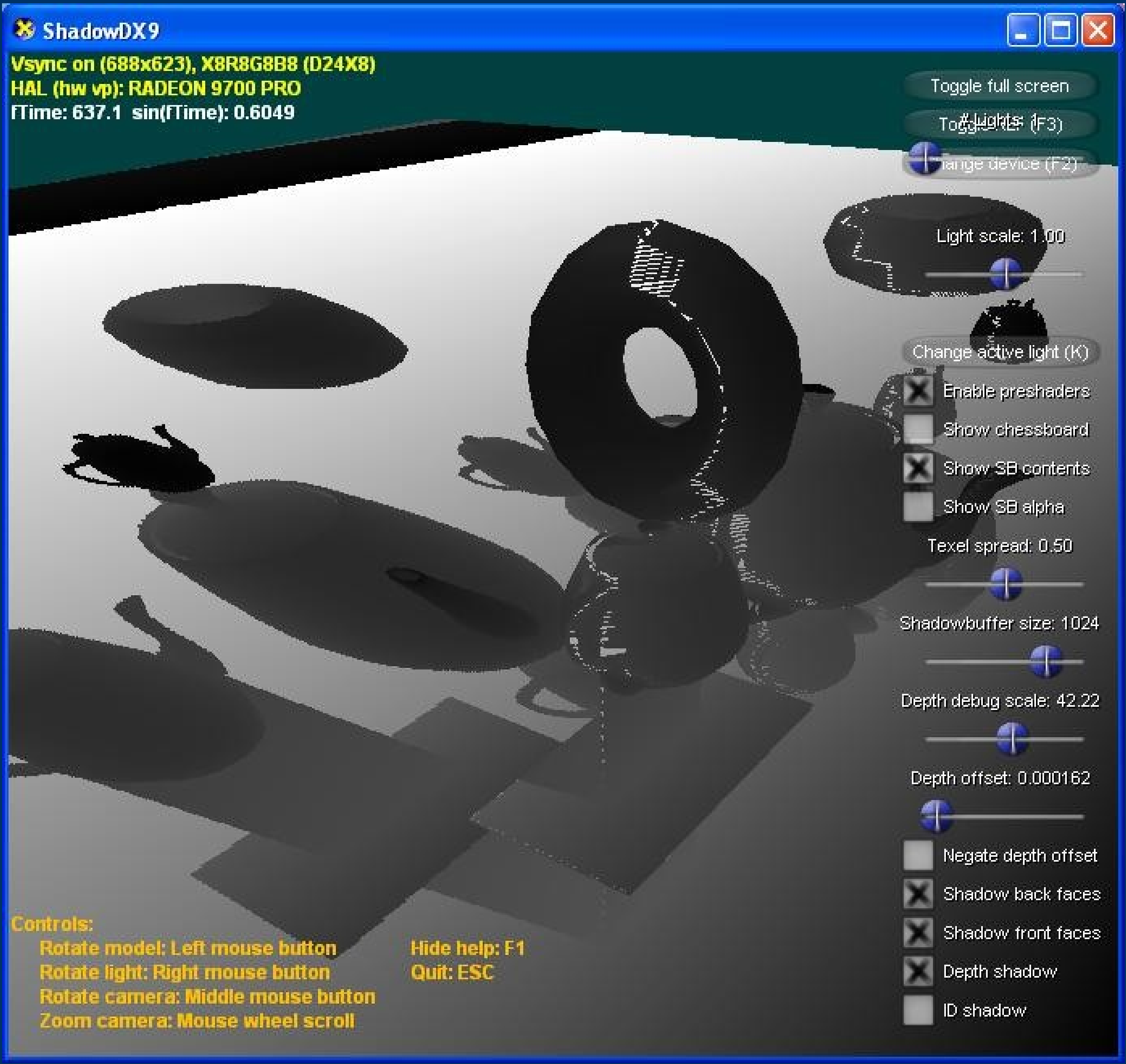
Major hurdles

- Which method of shadowbuffering?
  - What shader algorithm do we use?
  - What space/speed requirements does it have?
  - How robust is it?
- How do you choose the frustum?
  - Mitigating aliasing problems
  - Concentrating fillrate where it counts
- Making soft shadows
  - Look far better than hard shadows
  - But how?

Which method?

Depth shadowbuffers

- Surface identifier = distance from light
- Can be same values as Z buffer
- Depth shadowbuffer can be the Z buffer
  - Depends on hardware support
- if computed.depth > texture.depth
  - Object must be further from light
  - Therefore shadowed
- else
  - Object visible to light
  - Therefore lit



```
Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 637.1  sin(fTime): 0.6049
```

Lights: 1

Light scale: 1.00

☒ Enable preshaders

□ Show chessboard

☒ Show SB contents

□ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 42.22

Depth offset: 0.000162

□ Negate depth offset

☒ Shadow back faces

☒ Shadow front faces

☒ Depth shadow

□ ID shadow

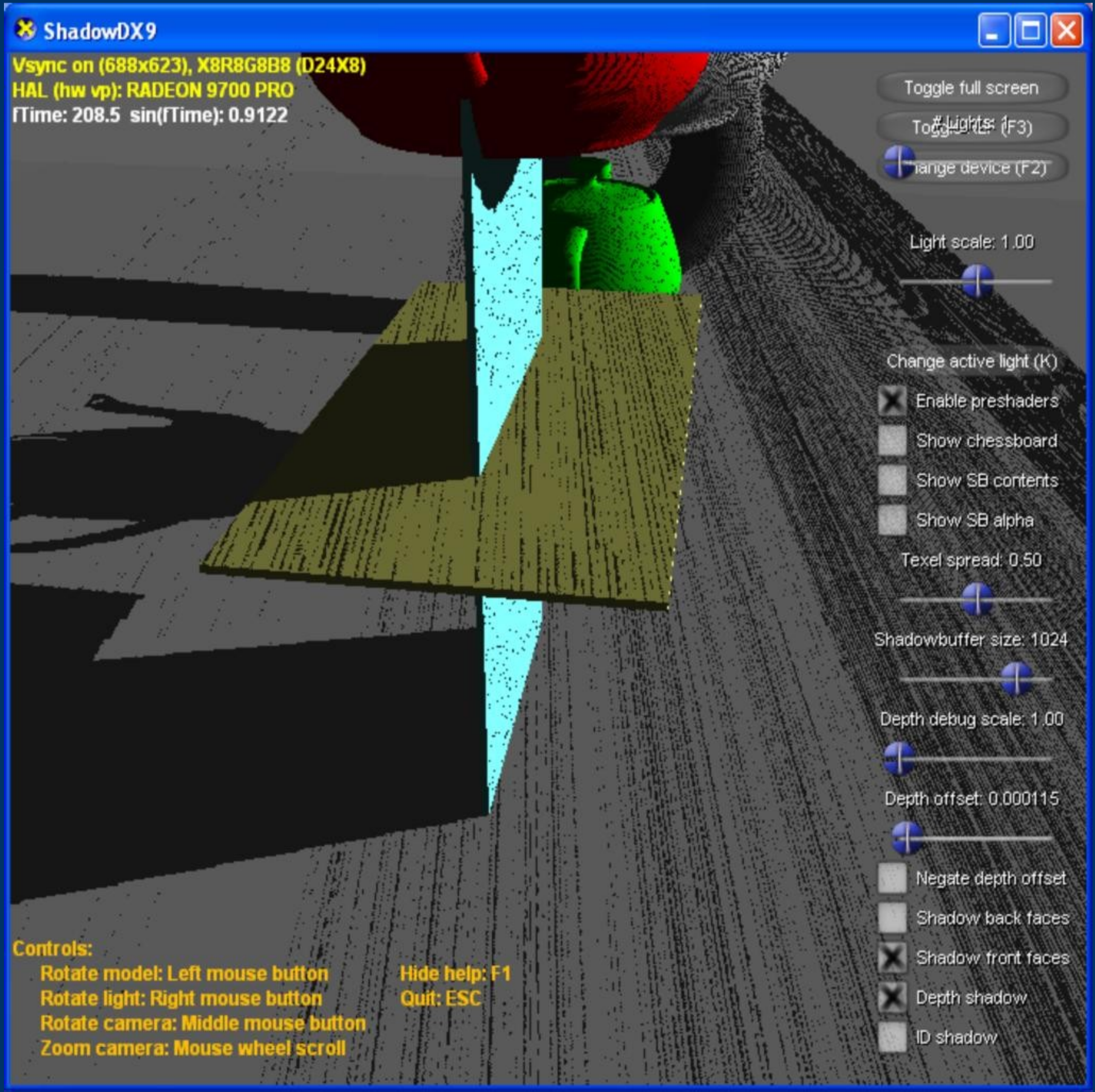
```
Controls:
    Rotate model: Left mouse button        Hide help: F1
    Rotate light: Right mouse button       Quit: ESC
    Rotate camera: Middle mouse button
    Zoom camera: Mouse wheel scroll
```

Depth shadowbuffers 2

- Simple in theory
- Lots of annoying problems in practice
- Incorrect self-shadowing
  - "Surface acne"
- Shadows detaching from objects
  - "Peter Pan" syndrome
- Hardware support is variable
  - Often needs high-precision buffers

Surface acne

- Incorrect self-shadowing on lit surface
- Caused by sampling differences
  - Value read from shadowbuffer is quantised
  - Computed value is not
- Frequency quantisation: not enough bits
  - 8 bits not enough
  - 24 bits = same as Z buffer = enough
- Spatial quantisation: small shadowbuffer
  - Gets very expensive very quickly
  - In practice, hardware & speed are limits



ShadowDX9

Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 208.5 sin(fTime): 0.9122

Light scale: 1.00

☒ Enable preshaders
□ Show chessboard
□ Show SB contents
□ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 1.00

Depth offset: 0.000115

□ Negate depth offset
□ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
□ ID shadow

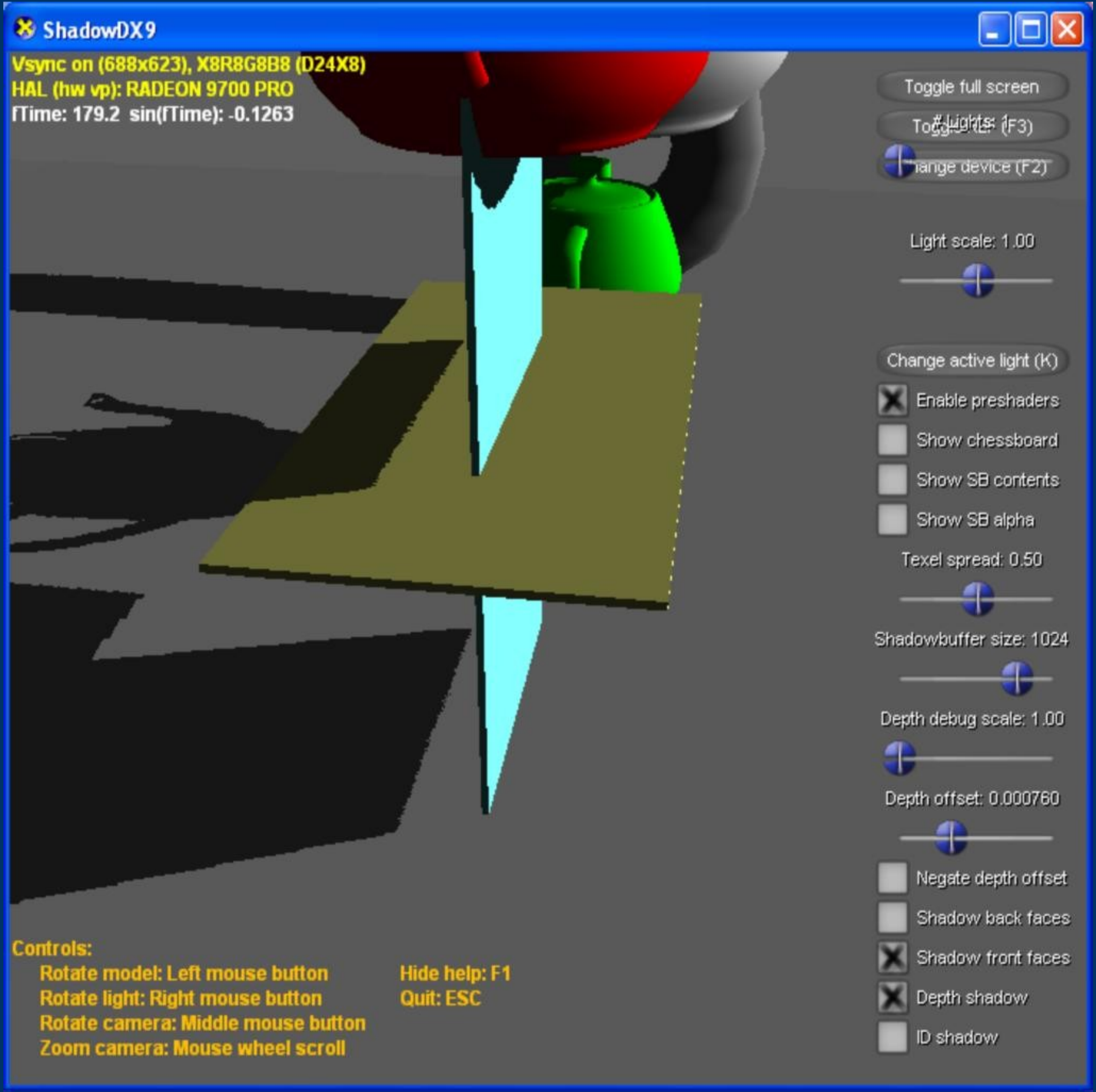
Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll
Hide help: F1
Quit: ESC

Surface acne - cures

- Surface acne is not a bug!
  - Fundamental side-effect of aliasing
  - Inherent in every image-based system
- Use more bits
  - 2x precision = 1 more bit
  - Fairly cheap, but careful of hardware support!
- Use bigger textures
  - Far more expensive
  - 2x precision = 4x memory & fillrate
- In practice, resolution is the limiter

Surface acne - cures 2

- Use a bias
  - Make it bigger than the quantisation errors
  - But errors are slope-dependent!
- Use a slope-dependent bias
  - Doesn't cope well with irregular surfaces
  - Bias can get very large for edge-on surfaces
  - Helps, but not very much
- Biases cause Peter Pan syndrome
  - Shadows detaching from their casters



```
Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 179.2  sin(fTime): -0.1263
```

Lights: 1

Light scale: 1.00

☒ Enable preshaders

□ Show chessboard

□ Show SB contents

□ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 1.00

Depth offset: 0.000760

□ Negate depth offset

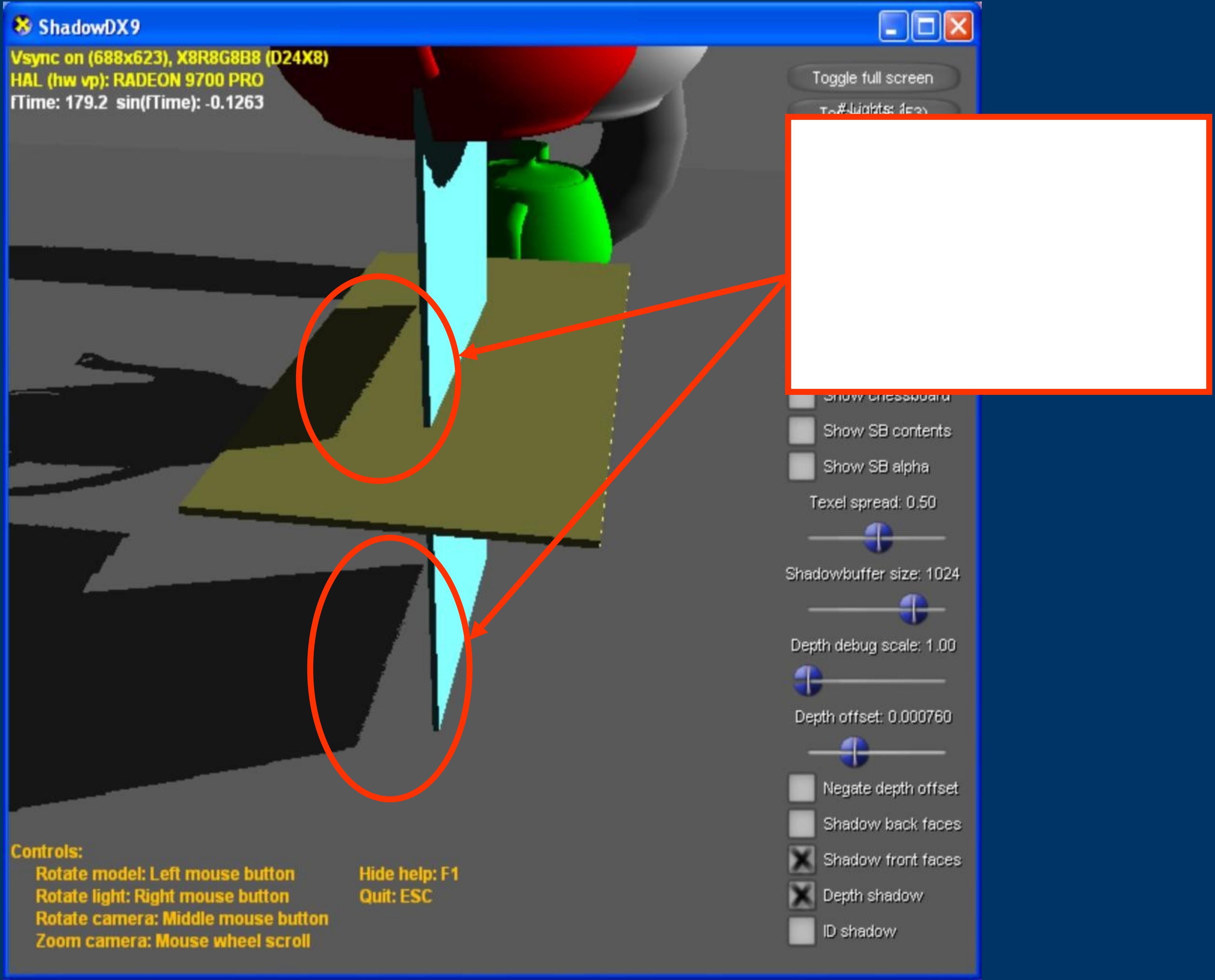
□ Shadow back faces

☒ Shadow front faces

☒ Depth shadow

□ ID shadow

```
Controls:
    Rotate model: Left mouse button        Hide help: F1
    Rotate light: Right mouse button       Quit: ESC
    Rotate camera: Middle mouse button
    Zoom camera: Mouse wheel scroll
```



ShadowDX9

Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 179.2 sin(fTime): -0.1263

Toggle full screen

To#-lights: (F3)

□ Show chessboard

□ Show SB contents

□ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 1.00

Depth offset: 0.000760

□ Negate depth offset

□ Shadow back faces

☒ Shadow front faces

☒ Depth shadow

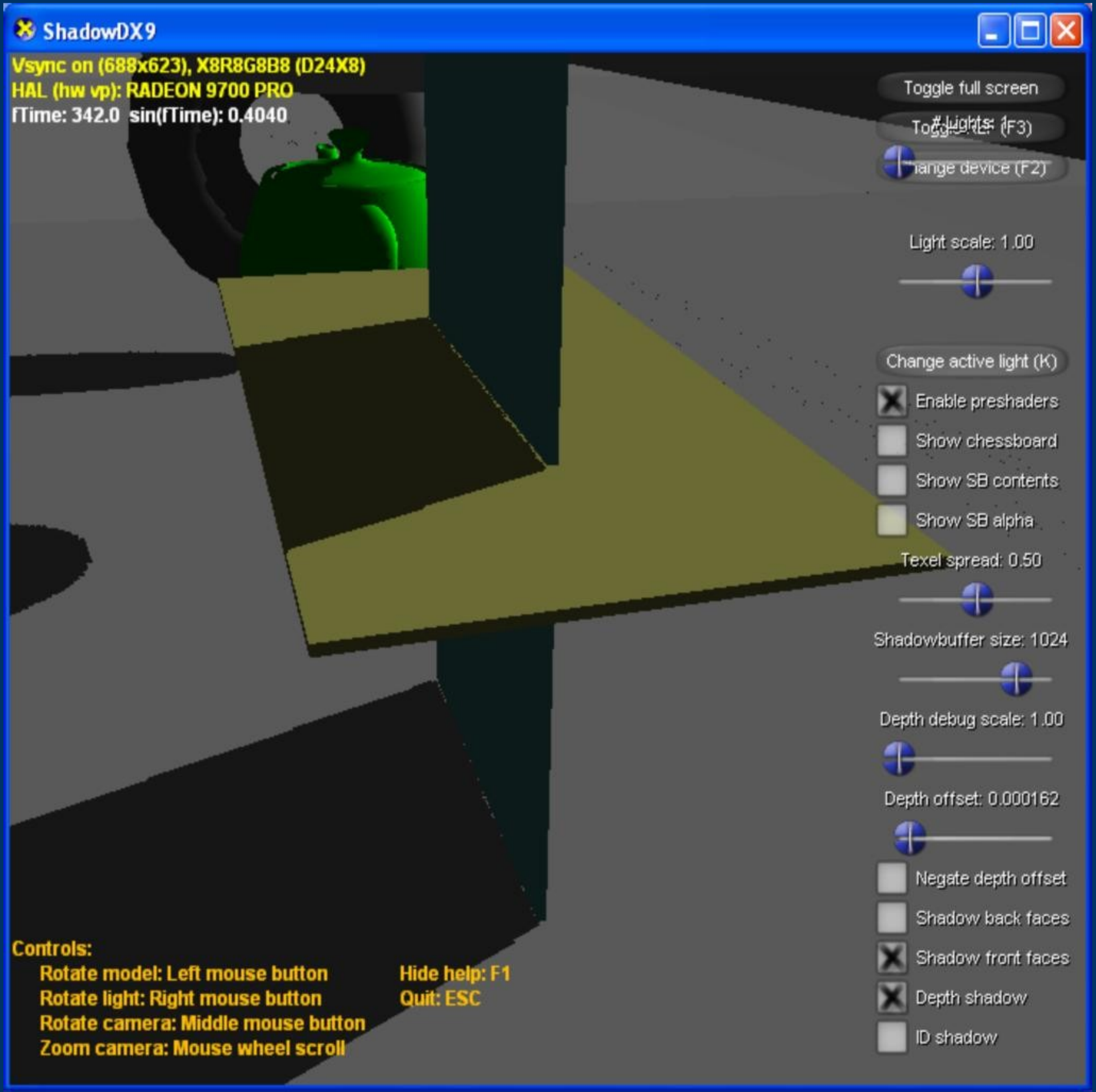
□ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll

Hide help: F1
Quit: ESC

Peter Pan vs Acne

- Large biases cause shadows to detach
  - Shadow values pushed through objects
- Small biases cause surface acne
- Some biases cause both!
  - Different areas of a scene show different ones
  - In practice, no bias causes neither :-(
- Could render backfaces to shadowbuffer
  - Acne is invisible on unlit backfaces
  - Error still large enough on thin objects
  - Relies on objects being closed
  - Causes even worse Peter Pan problems!



Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 342.0 sin(fTime): 0.4040

Light scale: 1.00

☒ Enable preshaders
☐ Show chessboard
☐ Show SB contents
☐ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

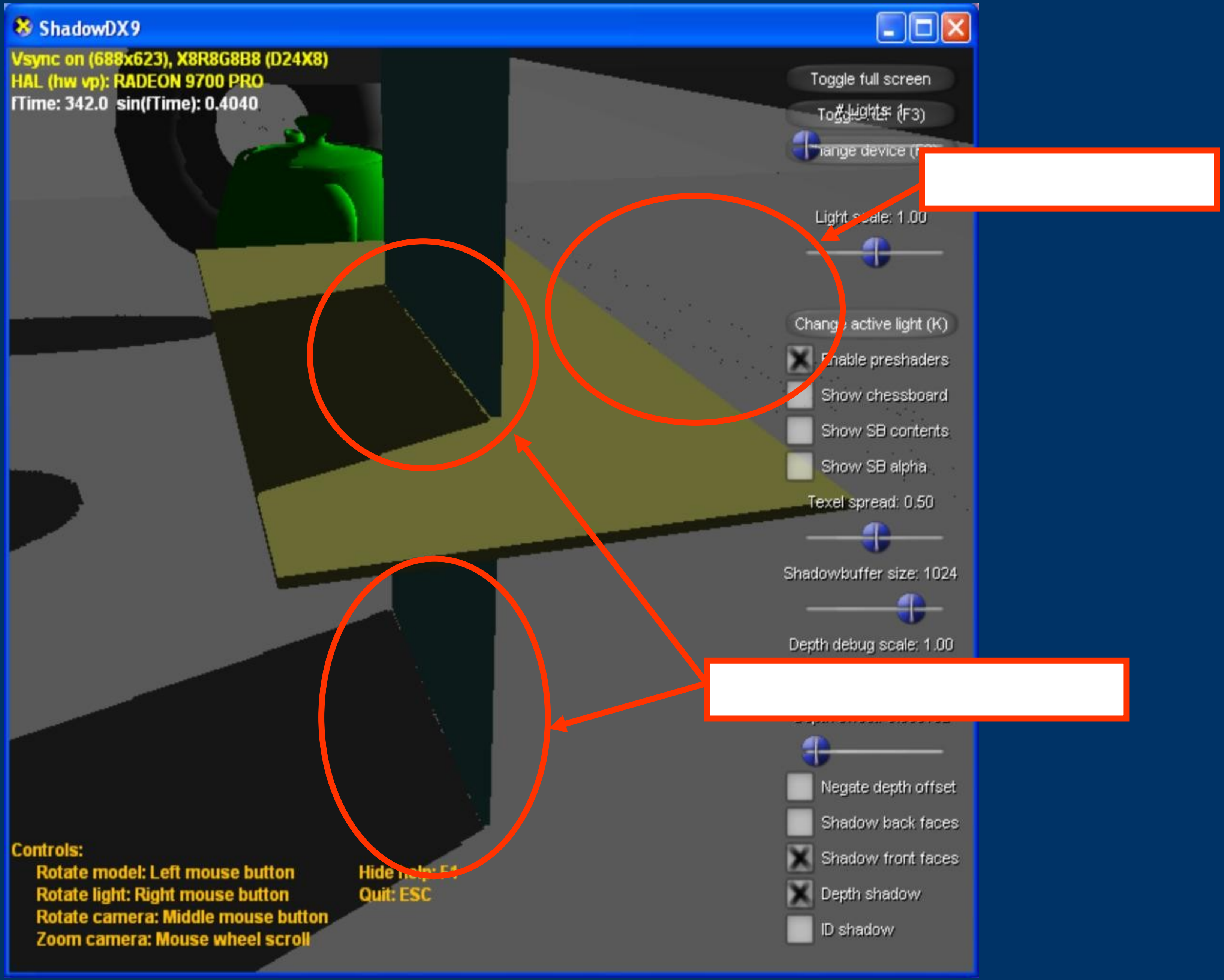
Depth debug scale: 1.00

Depth offset: 0.000162

☐ Negate depth offset
☐ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
☐ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll

Hide help: F1
Quit: ESC



Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 342.0 sin(fTime): 0.4040

Lights: 1

Light scale: 1.00

☒ Enable preshaders
□ Show chessboard
□ Show SB contents
□ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 1.00

□ Negate depth offset
□ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
□ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll

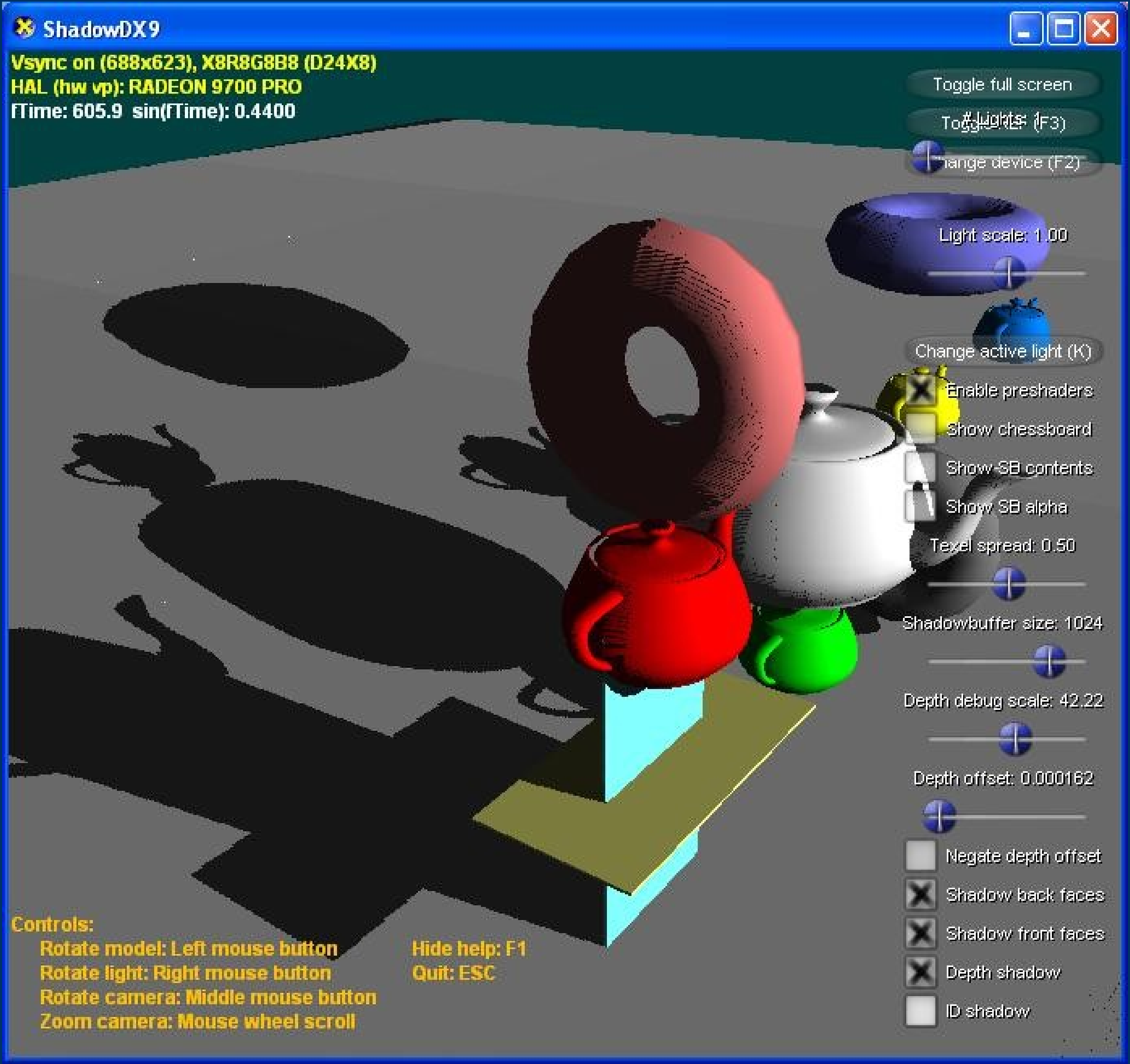
Hide help: F1
Quit: ESC

Depth shadowbuffers 3

- Lots of interesting research
- Smarter biases: Gradient Shadow Maps
  - Christian Schueler (ShaderX4, ShaderX5)
- Smarter sampling: Irregular Z buffer
- So far, nothing works for everything
  - Large flat areas at glancing angles
  - Curved or bumpy areas
  - Thin areas
  - Objects resting or interpenetrating
  - Combinatorial nightmare - tune it for one situation and it fails for another

ID shadowbuffers

- Don't **need** to use depth
- Shadowing asks simple question
  - "Is the surface in the shadowbuffer me?"
- Just need something to identify surface
- Can just pick an integer
  - Here shown as a shade of grey



Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 605.9 sin(fTime): 0.4400

Light scale: 1.00

☒ Enable preshaders
☐ Show chessboard
☐ Show SB contents
☐ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

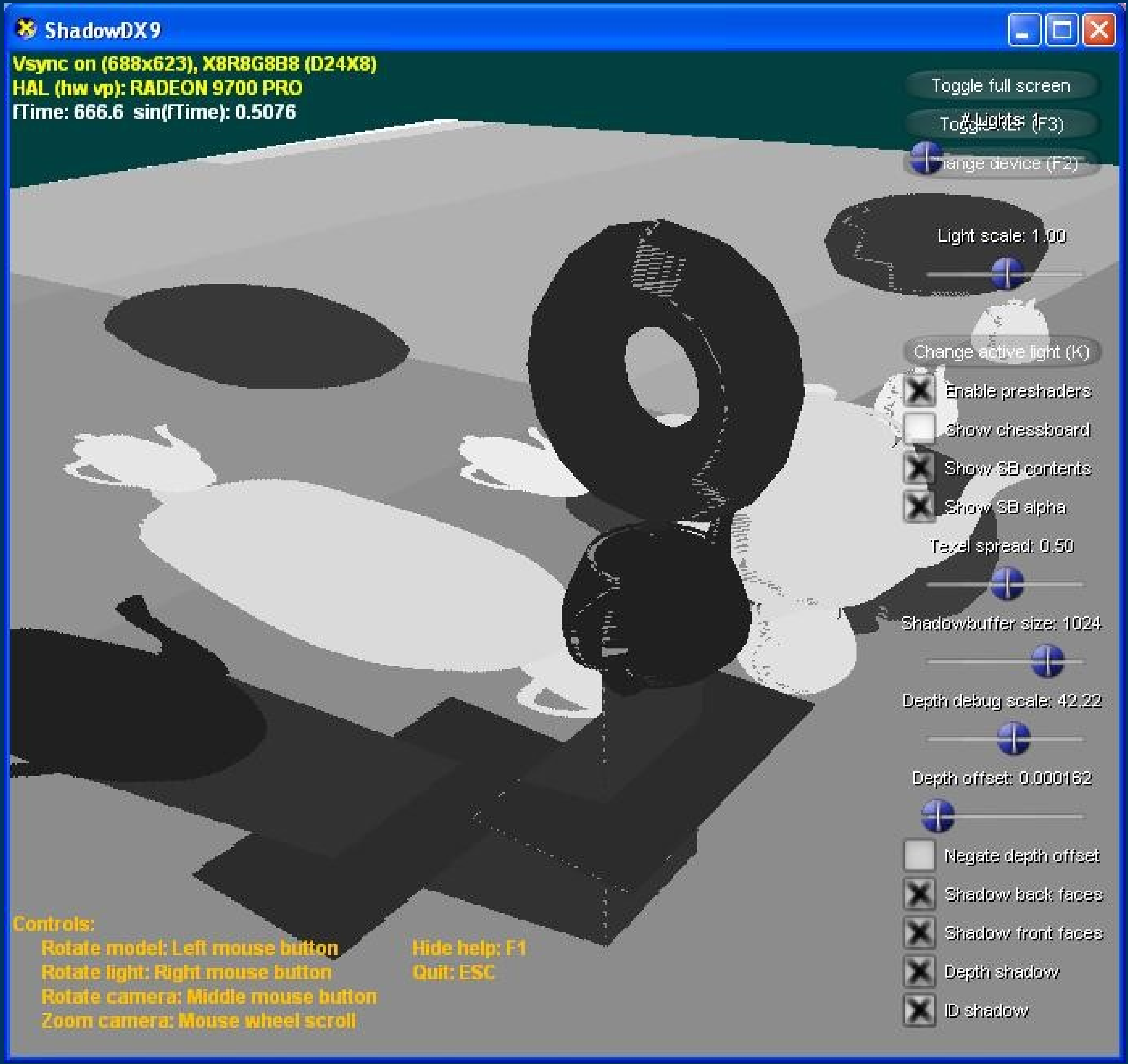
Depth debug scale: 42.22

Depth offset: 0.000162

☐ Negate depth offset
☒ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
☐ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll

Hide help: F1
Quit: ESC



ShadowDX9

```
Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 666.6  sin(fTime): 0.5076
```

Light scale: 1.00

☒ Enable preshaders
☐ Show chessboard
☒ Show SB contents
☒ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 42.22

Depth offset: 0.000162

☐ Negate depth offset
☒ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
☒ ID shadow

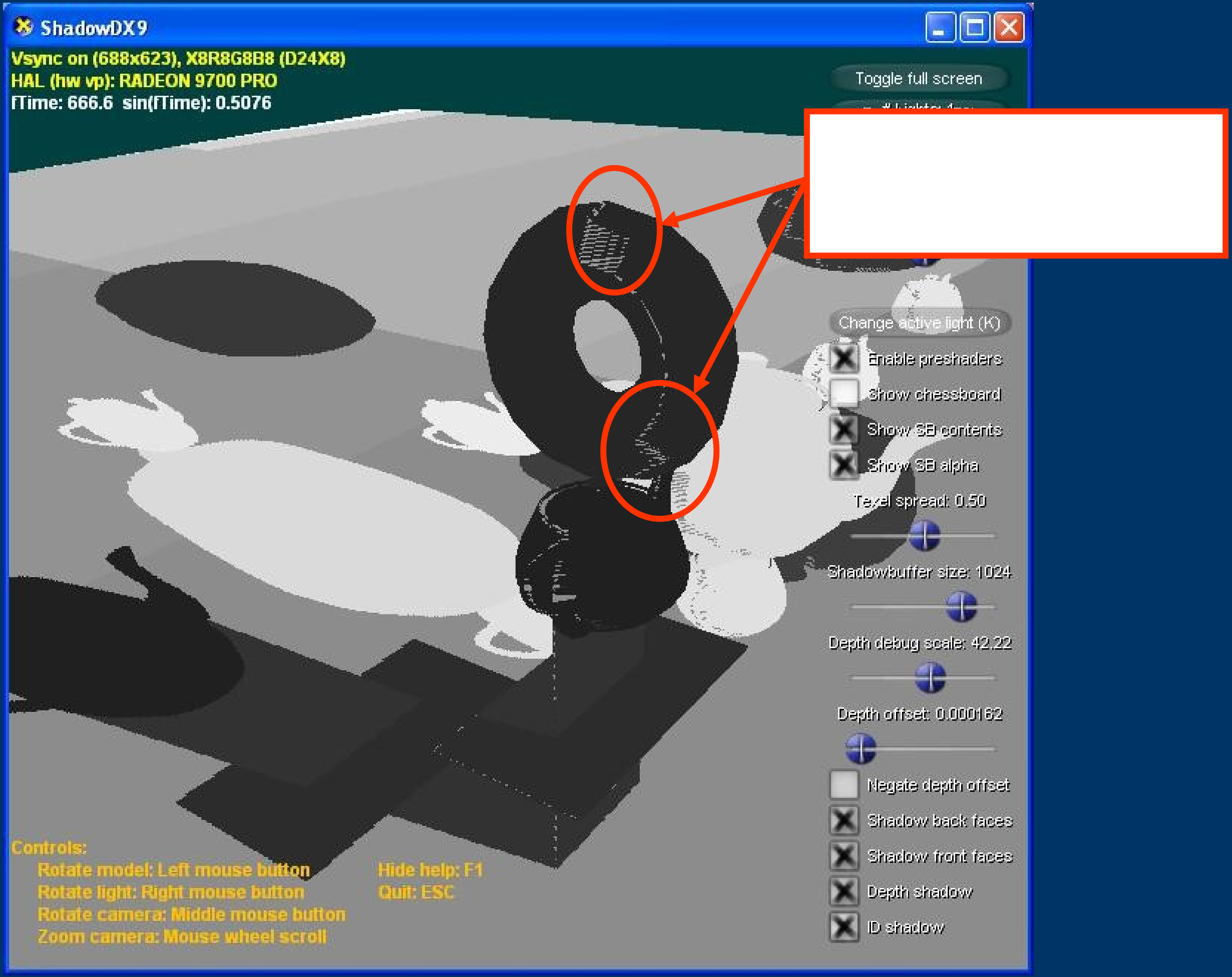
```
Controls:
    Rotate model: Left mouse button        Hide help: F1
    Rotate light: Right mouse button       Quit: ESC
    Rotate camera: Middle mouse button
    Zoom camera: Mouse wheel scroll
```

ID shadowbuffers 2

- Per-triangle integer
  - Needs hardware support (some ATI cards)
  - DX10 will have primitive ID as standard
  - Needs lots of integers - 16 bits+
  - Pixel-sized triangles can get “lost”
    - Causes acne
- Per-object integer
  - No hardware support needed
  - 8-bit ID will do fine - 256 objects in scene
    - For more, check light-space bounding boxes

ID shadowbuffers 3

- "Edge acne"
  - Sampling misses the edge of an object
  - Hits object behind
  - IDs don't match => shadowing
  - (depth shadowbuffers have an implicit order)
- Can sort objects back to front
  - Expensive, sometimes not possible
- Sample nearest four neighbour texels
  - Only shadow if all four don't match
  - Careful with user-set control-panel texel offsets!



ShadowDX9

```
Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 666.6  sin(fTime): 0.5076
```

☒ Enable preshaders

□ Show chessboard

☒ Show SB contents

☒ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 42.22

Depth offset: 0.000162

□ Negate depth offset

☒ Shadow back faces

☒ Shadow front faces

☒ Depth shadow

☒ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll

Hide help: F1
Quit: ESC

ID shadowbuffers 4

- No maths is done on the IDs
  - No frequency aliasing problems
  - Simple shaders
  - No biases or tweaks needed
  - No acne or Peter Pan problems
  - Robust & predictable
- But no self-shadowing
  - Whole of an object is same ID
  - Can't cast shadows on itself
  - Not critical in some games

Depth vs ID

- Depth
  - Surface acne and Peter Panning
  - Large shadowbuffer surfaces
  - Bias is fiddly & scene-dependent
- IDs
  - Small surfaces = fast
  - Robust - write once, works everywhere
  - No self-shadowing

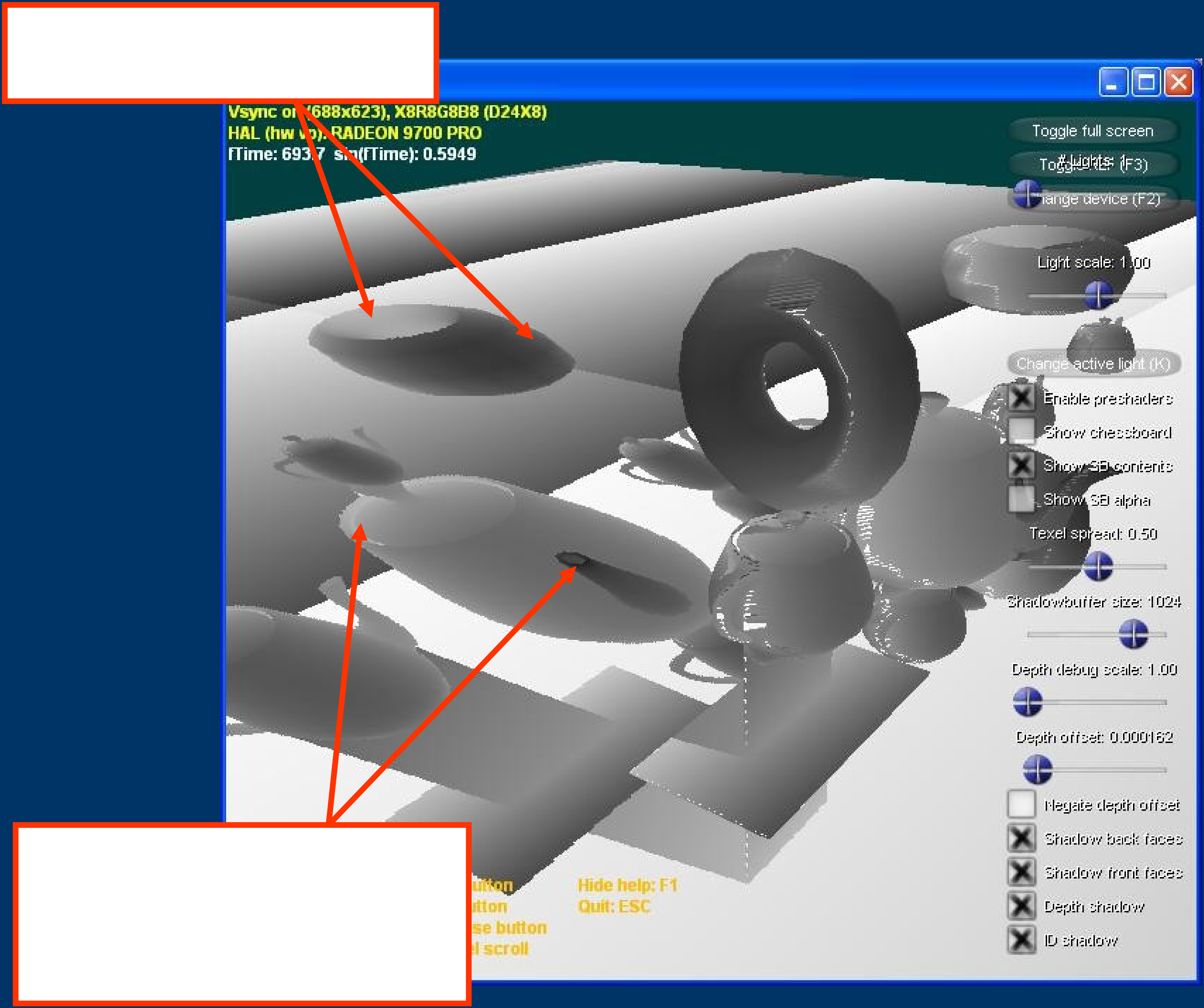
Depth + ID

- So use both!
- ID for inter-object (non-self) shadowing
  - No Peter Pan problems
  - No acne by definition
  - Works whatever the scene
- Depth for intra-object (self) shadowing
  - 0-1 depth only covers one object at a time
    - So can use smaller buffer - 8 bit usually fine
  - Bias tweaked for that object
  - Can have different biases for different objects

Depth + ID 2

- Shadowbuffer is small
  - 8bit ID + 8bit depth
  - No special hardware required

```
if object.ID != buffer.ID
    inter-object shadow
else if object.depth - bias > buffer.depth
    object self-shadows
else
    lit
```



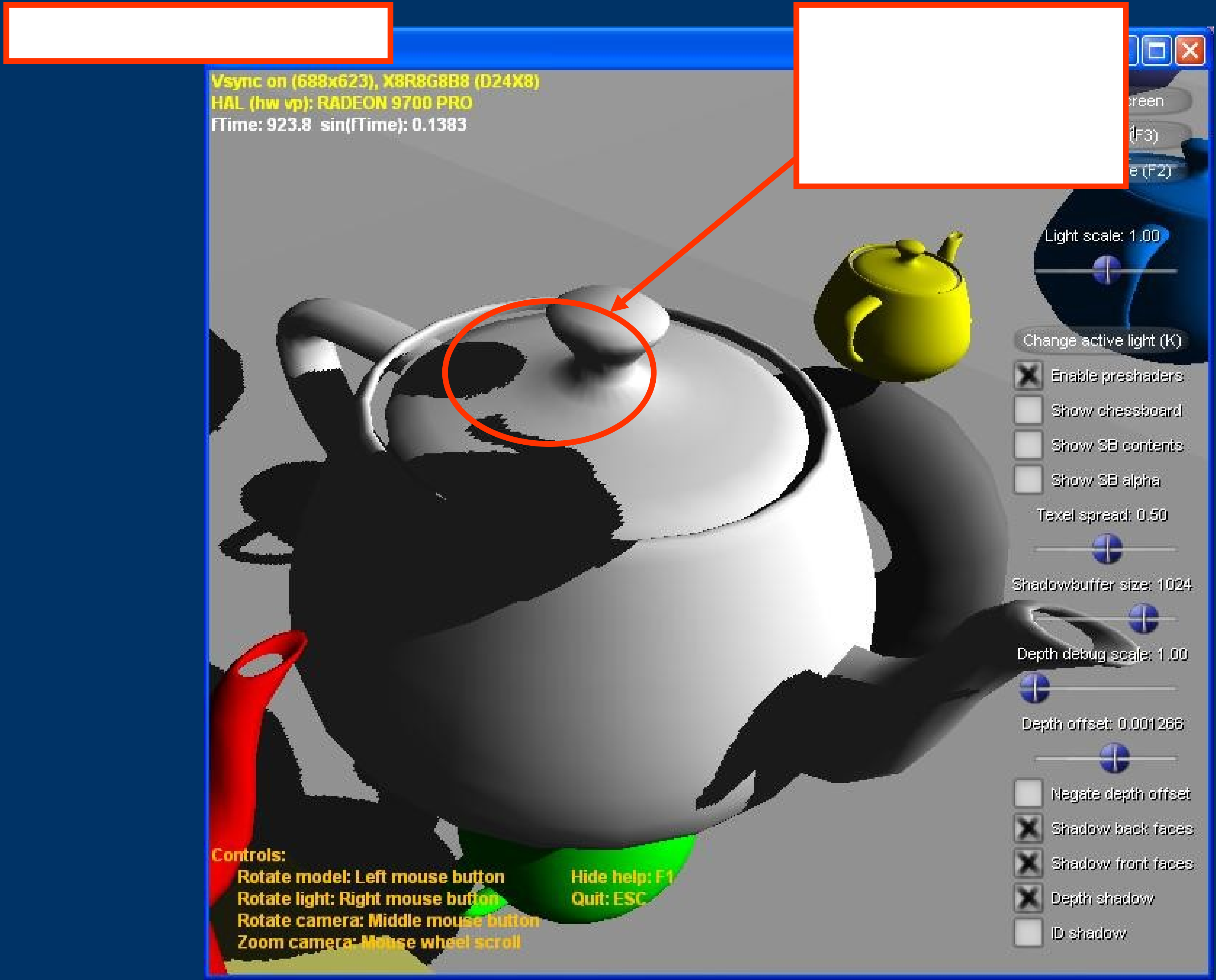
Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 693.7 sin(fTime): 0.5949

Light scale: 1.00

☒ Enable preshaders
☐ Show chessboard
☒ Show SB contents
☐ Show SB alpha

Texel spread: 0.50
Shadowbuffer size: 1024
Depth debug scale: 1.00
Depth offset: 0.000162

☐ Negate depth offset
☒ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
☒ ID shadow



Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 923.8 sin(fTime): 0.1383

Light scale: 1.00

☒ Enable preshaders
□ Show chessboard
□ Show SB contents
□ Show SB alpha

Texel spread: 0.50

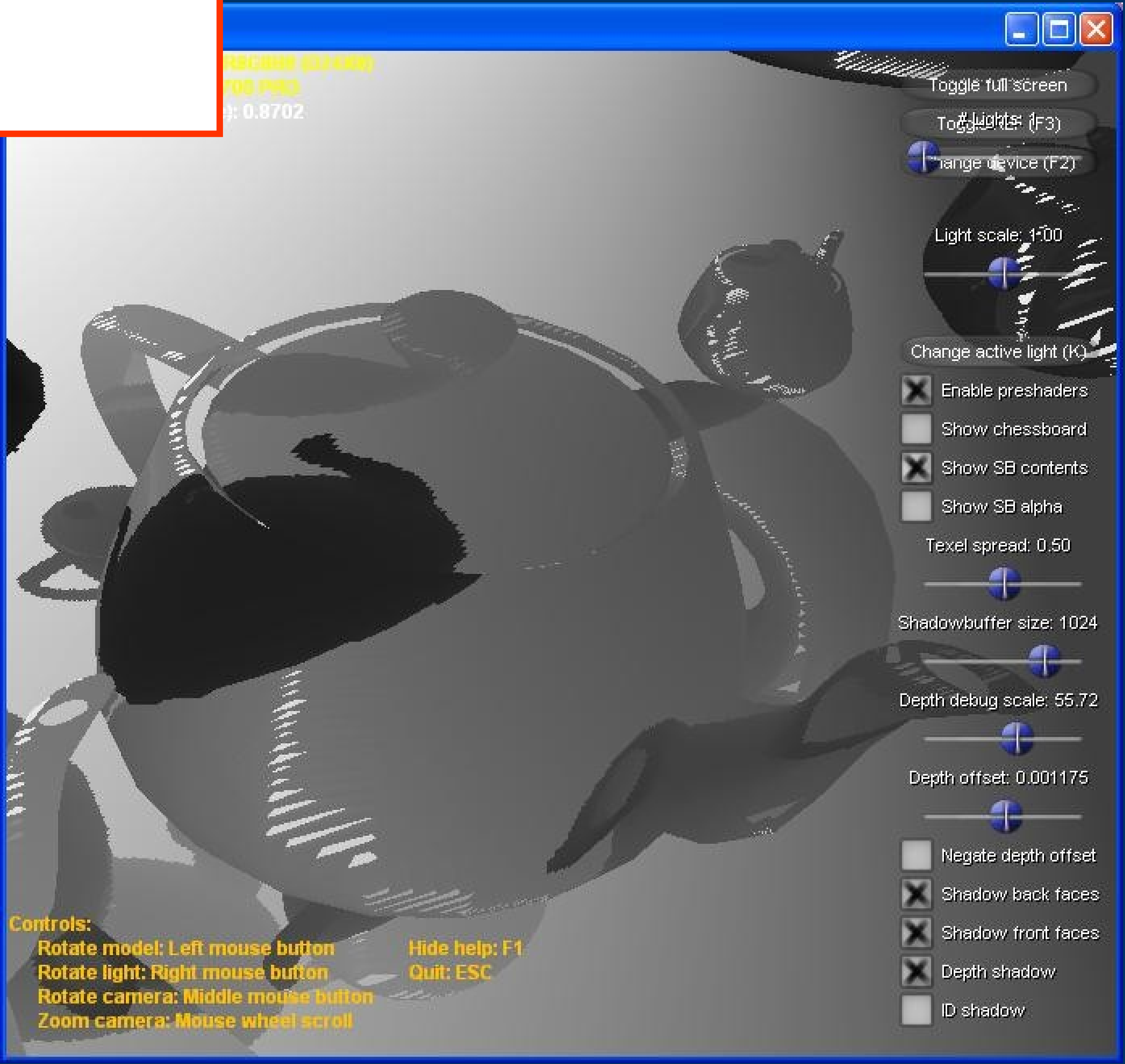
Shadowbuffer size: 1024

Depth debug scale: 1.00

Depth offset: 0.001266

□ Negate depth offset
☒ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
□ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll
Hide help: F1
Quit: ESC



R8G8B8 (D24X8)
700 PRO
: 0.8702

Light scale: 1.00

☒ Enable preshaders
☐ Show chessboard
☒ Show SB contents
☐ Show SB alpha

Texel spread: 0.50

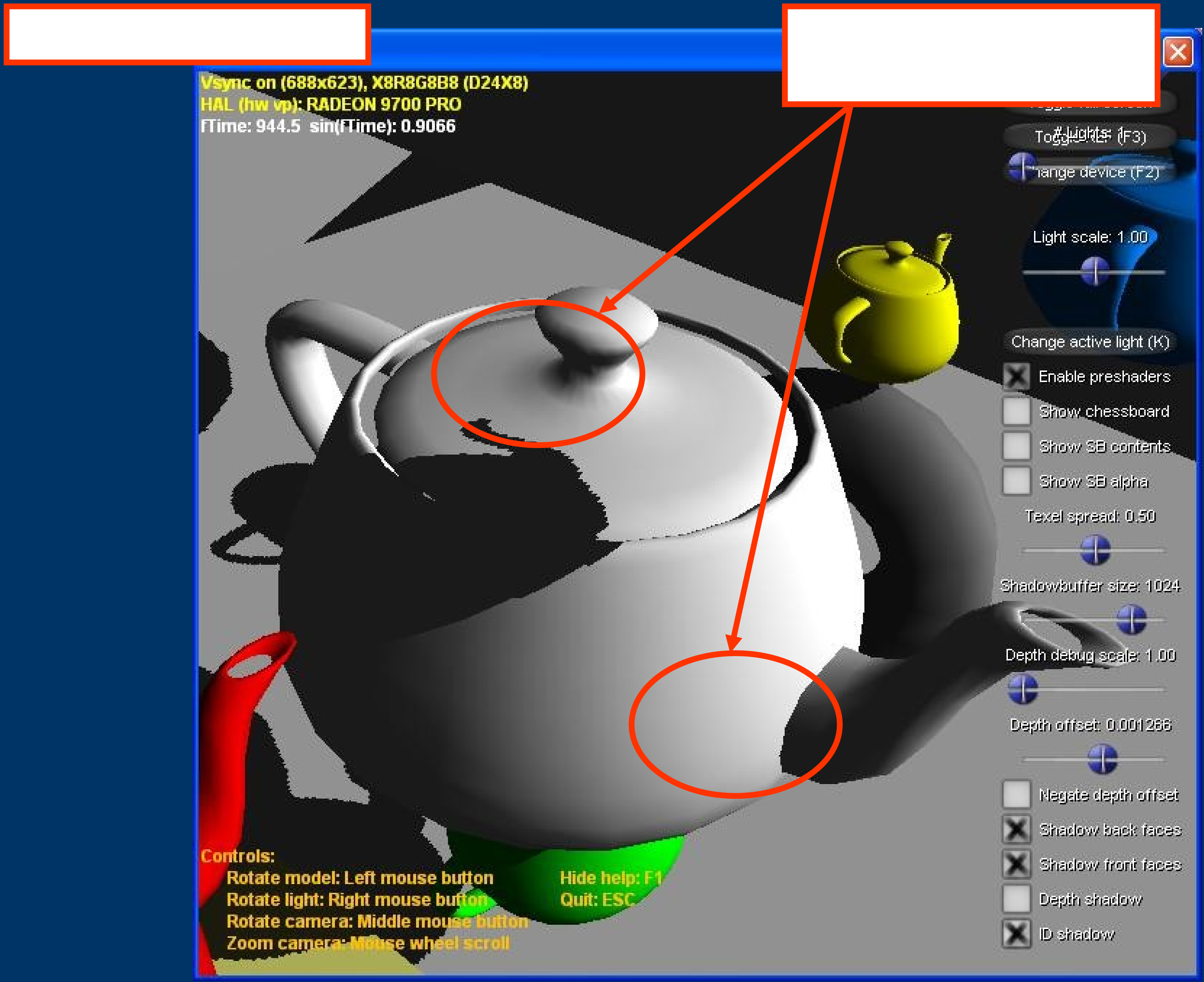
Shadowbuffer size: 1024

Depth debug scale: 55.72

Depth offset: 0.001175

☐ Negate depth offset
☒ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
☐ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll
Hide help: F1
Quit: ESC



Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 944.5 sin(fTime): 0.9066

Light scale: 1.00

☒ Enable preshaders
□ Show chessboard
□ Show SB contents
□ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

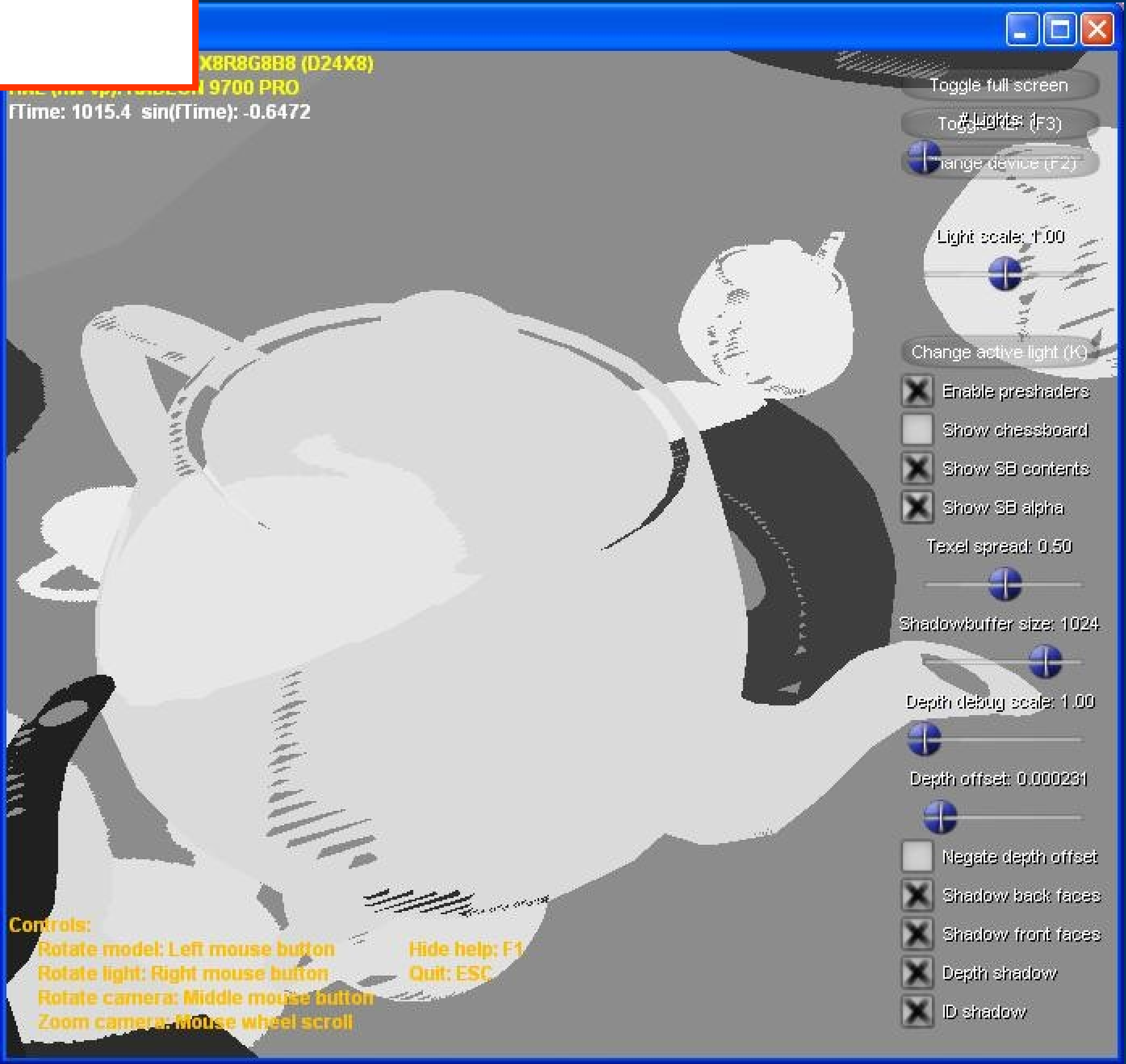
Depth debug scale: 1.00

Depth offset: 0.001266

□ Negate depth offset
☒ Shadow back faces
☒ Shadow front faces
□ Depth shadow
☒ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll

Hide help: F1
Quit: ESC



X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 1015.4 sin(fTime): -0.6472

Lights: 1

Light scale: 1.00

☒ Enable preshaders
☐ Show chessboard
☒ Show SB contents
☒ Show SB alpha

Texel spread: 0.50

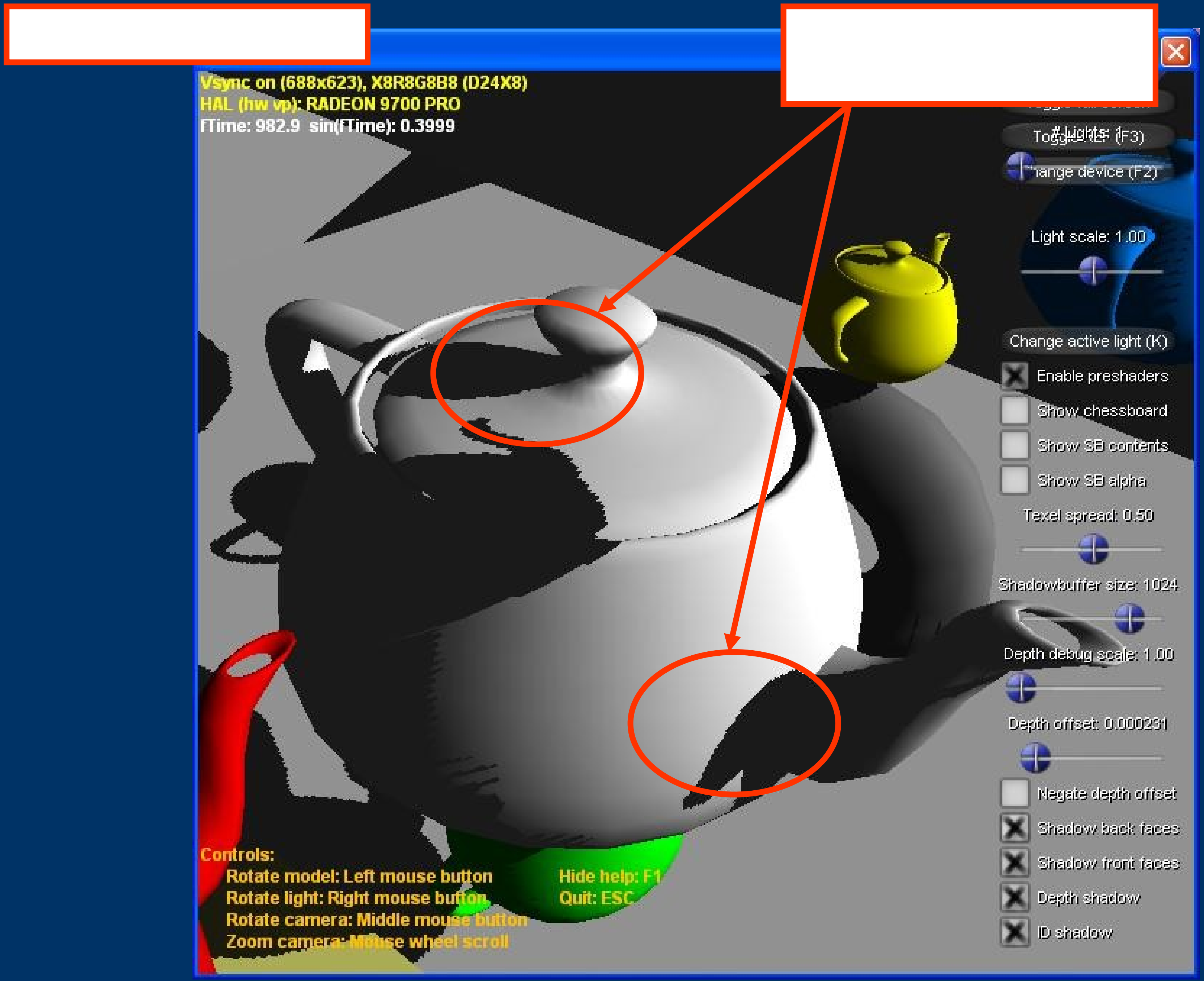
Shadowbuffer size: 1024

Depth debug scale: 1.00

Depth offset: 0.000231

☐ Negate depth offset
☒ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
☒ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll
Hide help: F1
Quit: ESC



```
Vsync on (688x623), X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 982.9  sin(fTime): 0.3999
```

Light scale: 1.00

☒ Enable preshaders

□ Show chessboard

□ Show SB contents

□ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 1.00

Depth offset: 0.000231

□ Negate depth offset

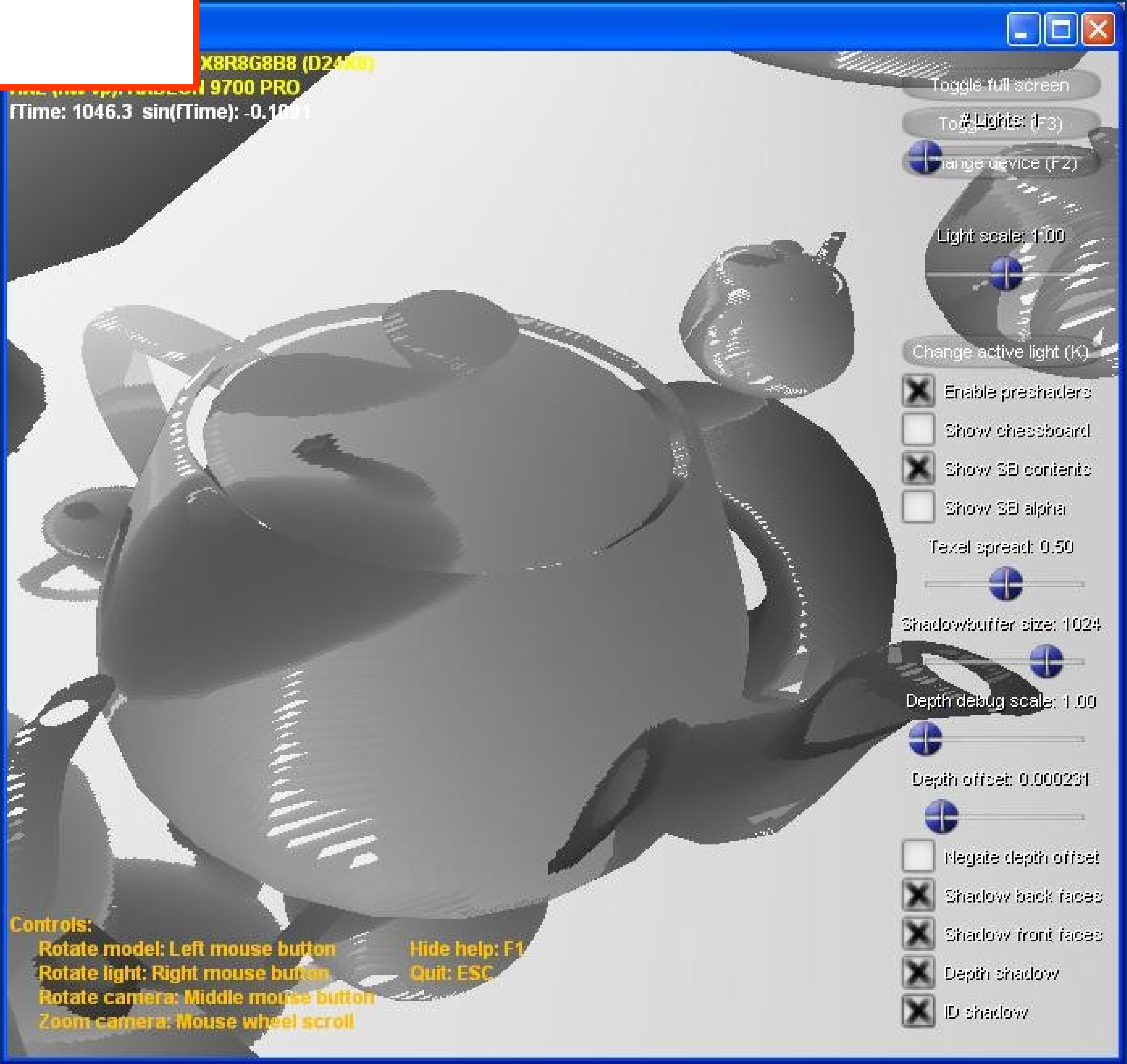
☒ Shadow back faces

☒ Shadow front faces

☒ Depth shadow

☒ ID shadow

```
Controls:
    Rotate model: Left mouse button        Hide help: F1
    Rotate light: Right mouse button       Quit: ESC
    Rotate camera: Middle mouse button
    Zoom camera: Mouse wheel scroll
```



X8R8G8B8 (D24X8)
HAL (hw vp): RADEON 9700 PRO
fTime: 1046.3 sin(fTime): -0.1001

Lights: 1

Light scale: 1.00

☒ Enable preshaders
☐ Show chessboard
☒ Show SB contents
☐ Show SB alpha

Texel spread: 0.50

Shadowbuffer size: 1024

Depth debug scale: 1.00

Depth offset: 0.000231

☐ Negate depth offset
☒ Shadow back faces
☒ Shadow front faces
☒ Depth shadow
☒ ID shadow

Controls:
Rotate model: Left mouse button
Rotate light: Right mouse button
Rotate camera: Middle mouse button
Zoom camera: Mouse wheel scroll
Hide help: F1
Quit: ESC

Demo...

Which frustums?

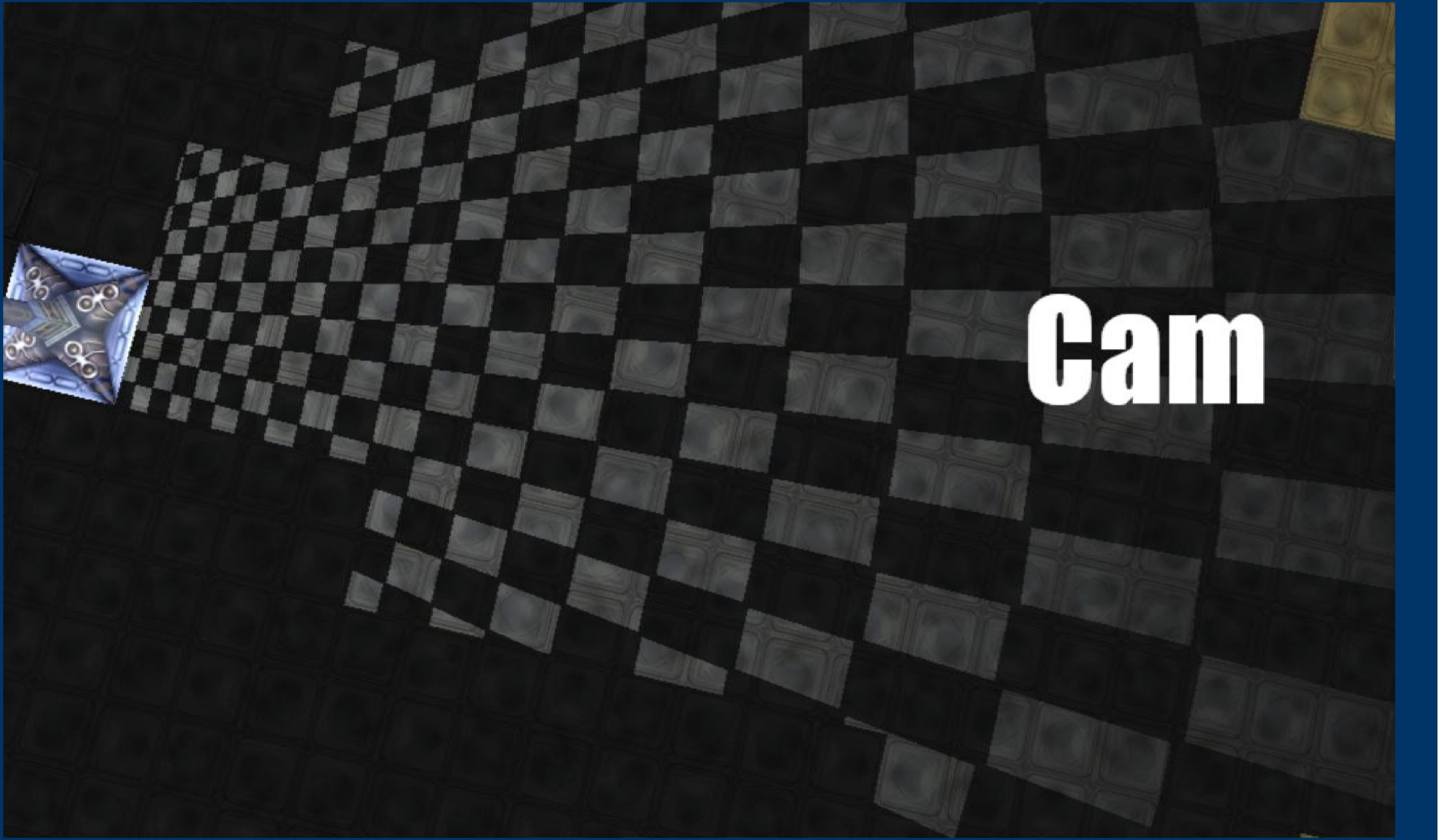
Shadowbuffer frustums

- Frustum = pyramid of rendered scene
- Position + direction + depth + FOV
- Shadowbuffers need frustums too
    - But they can sometimes get strange
- Position = light position
- Depth = usually not a big problem
- But direction + FOV are difficult

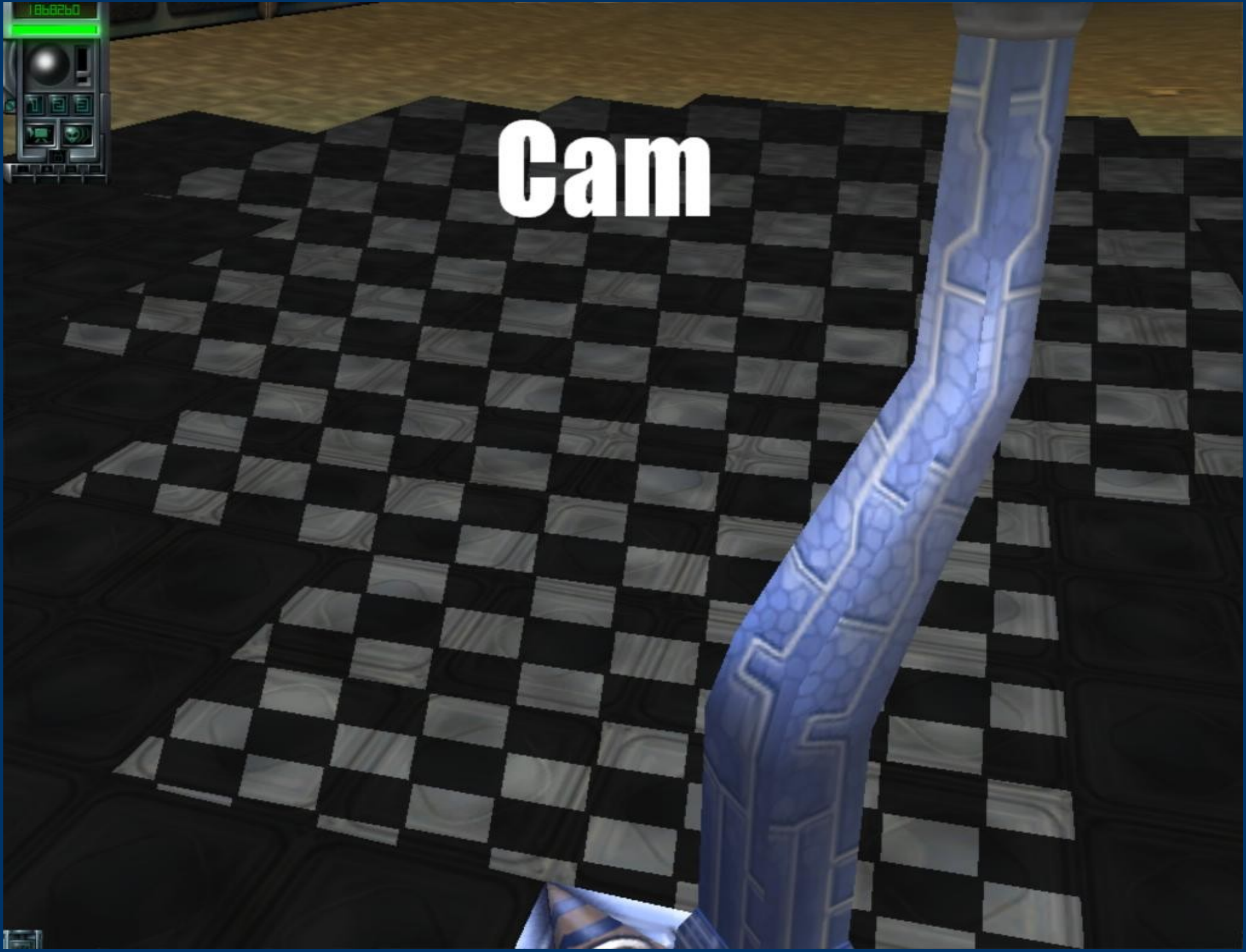
The projection problem

- Shadowbuffer rendered from light POV
- Then projected into camera POV
- The two do not agree
  - Can violently disagree - "duelling frustums"
- Too many texels in some places
  - Inefficient use of memory & fillrate
- Too few texels in others
  - More visible aliasing

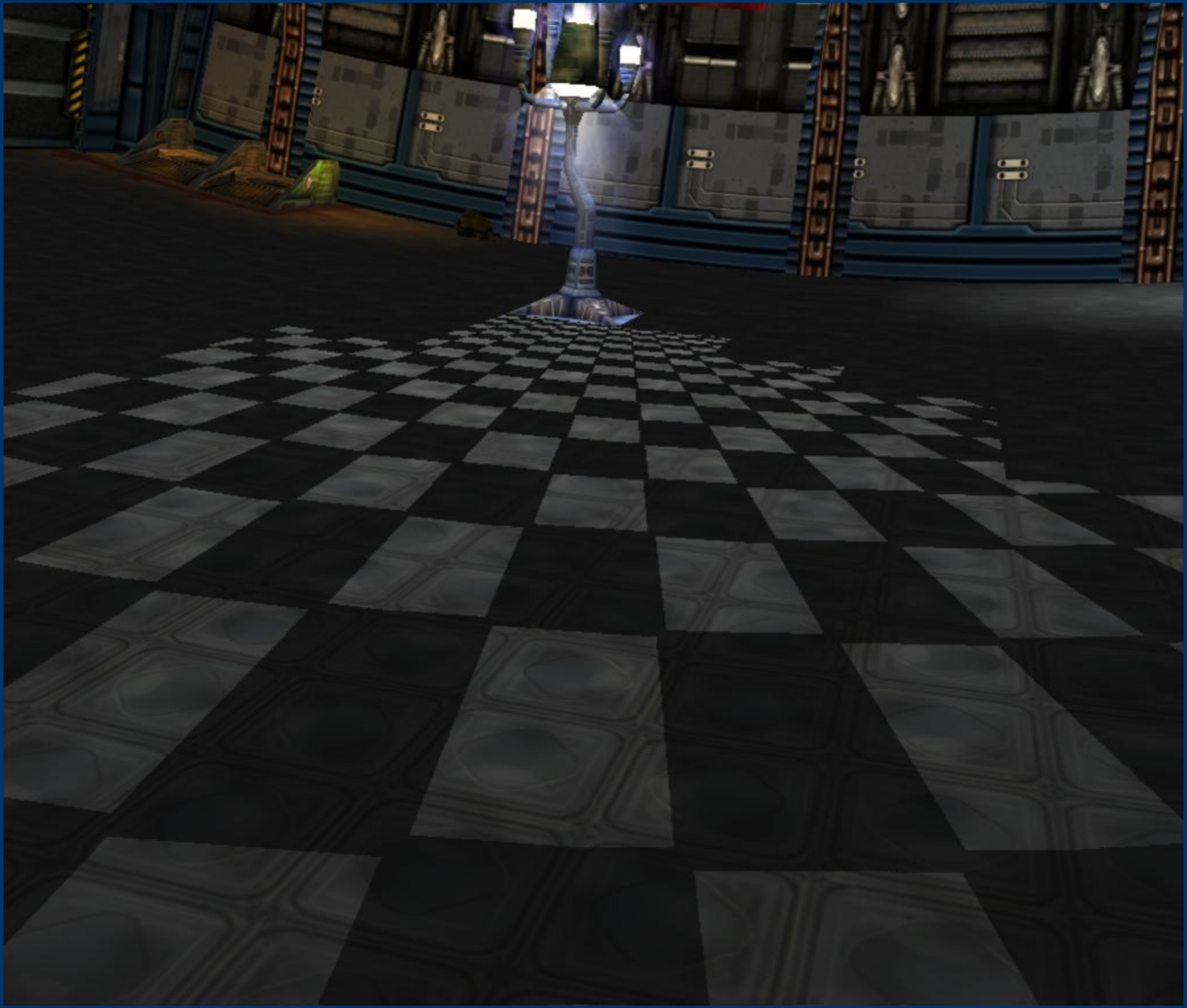
- Overhead view



- From the light POV



- From the camera POV



Smarter projection solutions

- Use extra freedom in the projection
  - Frustum doesn't have to look sensible!
- Perspective Shadow Maps
  - Flaky, full of special cases - avoid
- Light-space Persp. Shadow Maps
  - (LiSPSM)
  - More robust than PSM, but more complex
- Trapezoidal Shadow Maps (TSM)
  - Needs complex shader support
  - Tuned for terrains, not arbitrary worlds

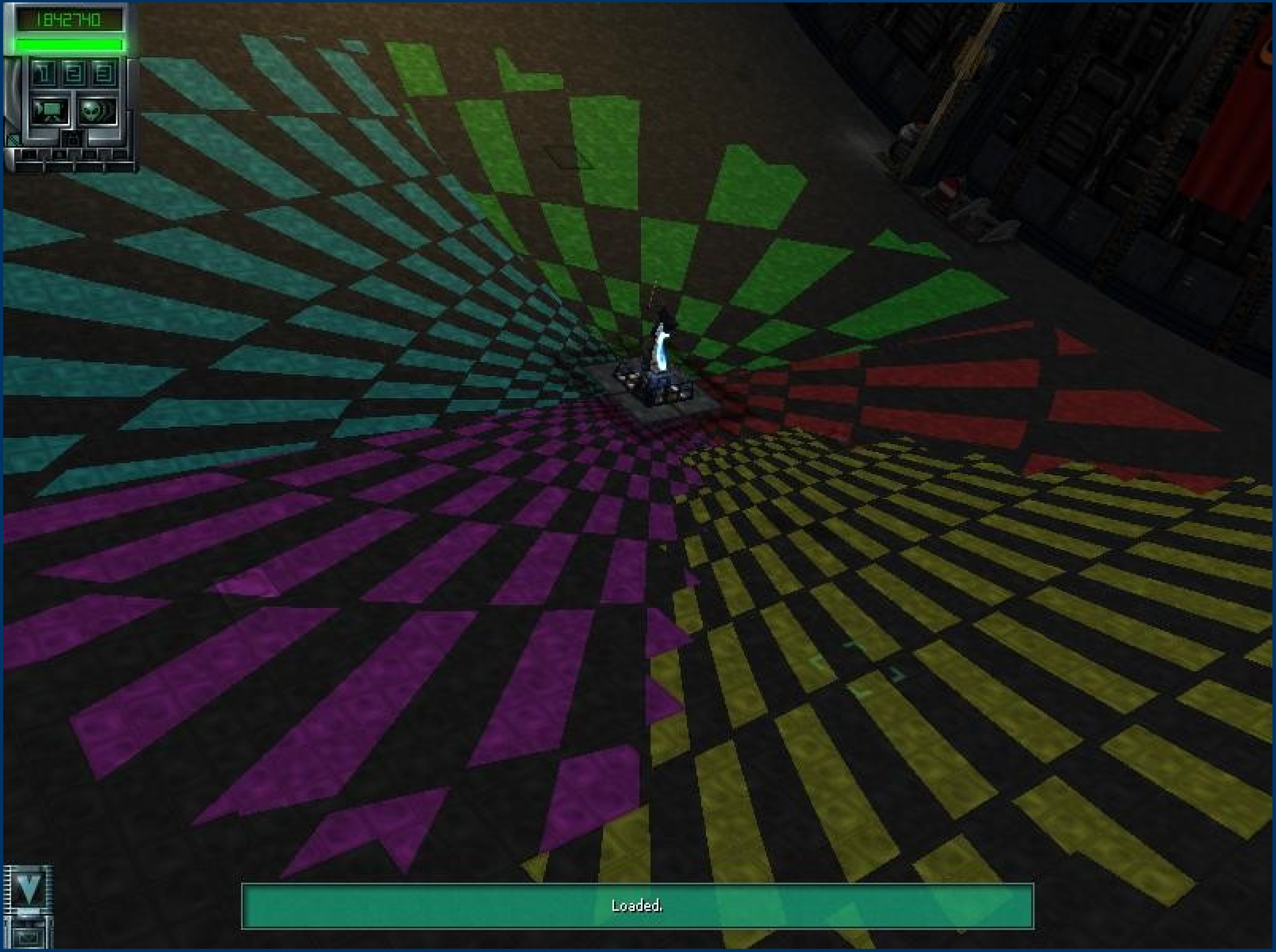
Smarter projection problems

- None of them solve duelling frustums
  - All degenerate to standard projection
- They can cause worse depth aliasing
  - Flexibility traded for spatial aliasing
- None of them solve omni-lights
  - Omni lights have 360-degree FOV!
  - Frustum cannot have >180
  - Practical limit is around 120 degrees
  - Also, guaranteed duelling frustum
    - Some part of the light is "facing" the camera

Multi-frustum partitioning

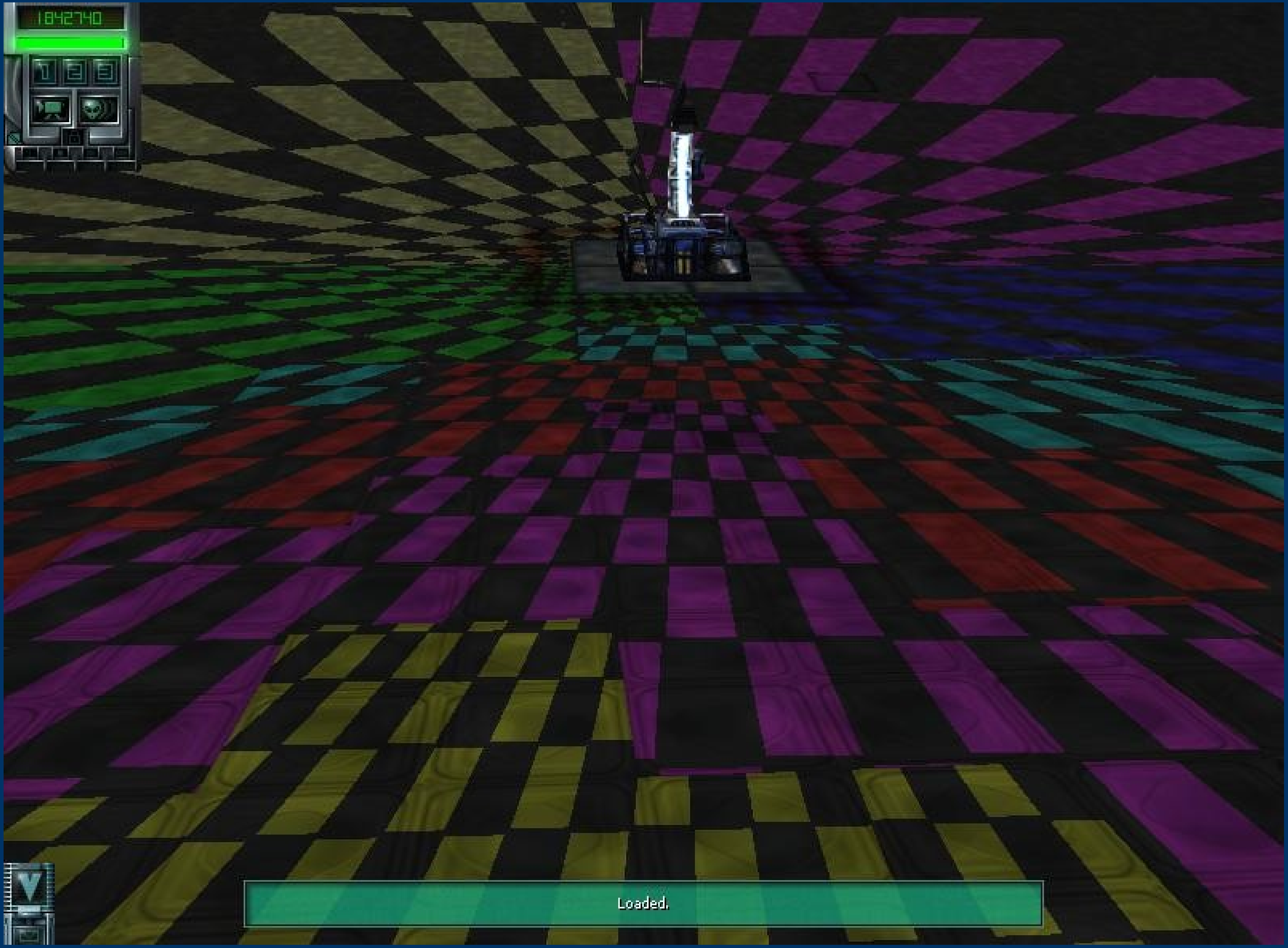
- Splits scene into multiple frustums
    - Each frustum rendered separately
    - Conventional frustum for each section
- Solves the two big problems
    - Duelling frustums
    - Omni lights
- Helps in other ways
    - Copes gracefully when smart projection fails
    - ...allows "dumber" smart projection
    - Can alleviate depth aliasing problem

- Omni with multiple frustums



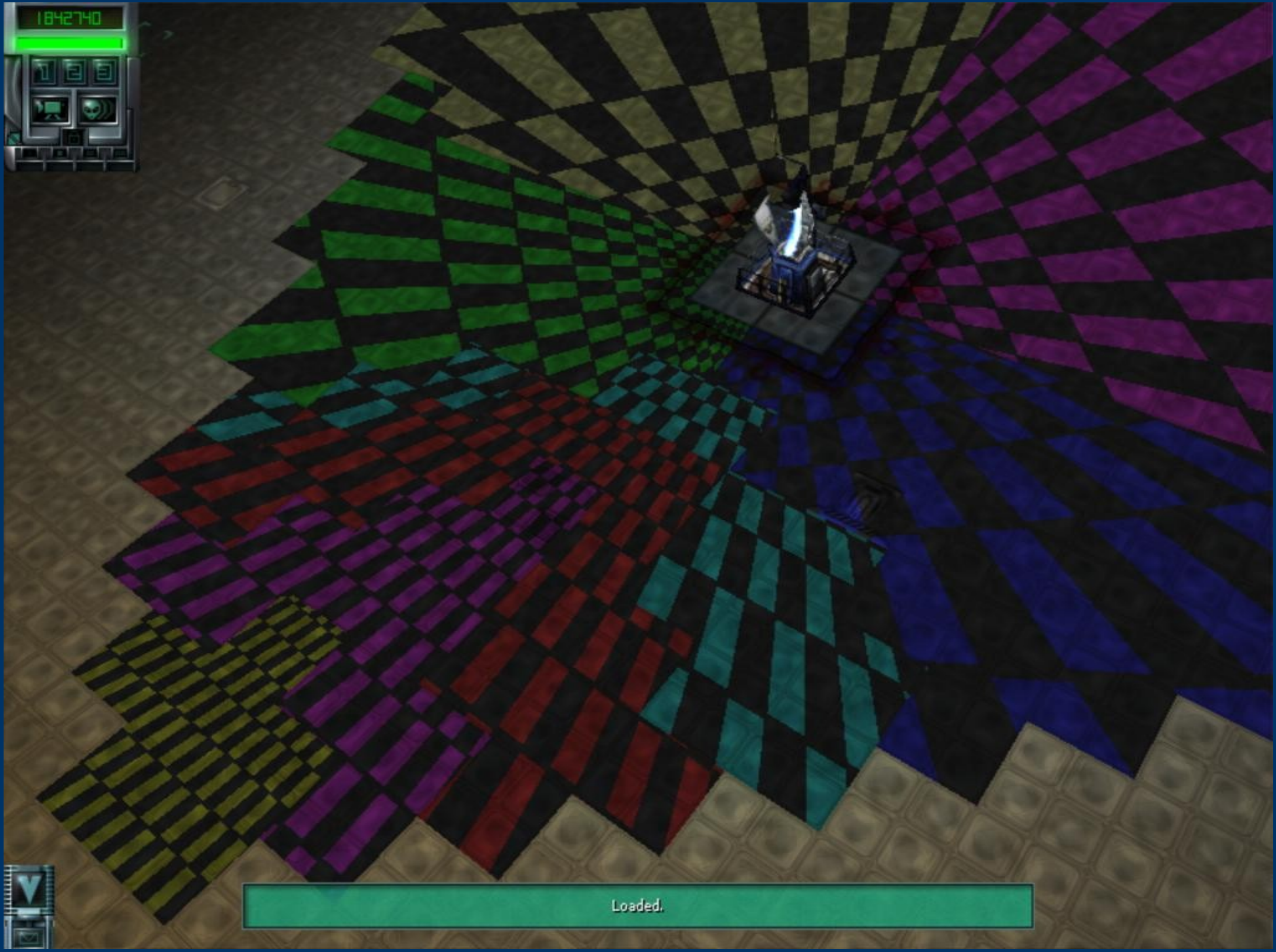
Loaded.

- Omni with duelling frustums



Loaded.

- Same, from above



MFP + smart projection

- MFP simply partitions the scene
- Each frustum can be "smart"
  - Different frustums can be differently smart
  - Where one has a problem, use another
- MFP can partition to avoid problems
  - Can dumb down the smart projections
  - Just solve problems by more partition
- MFP can just work with naïve frustums
  - Reliable fallback

MFP results

- Retrofitted to StarTopia (2001)
  - RTS/"god game"
  - Player-built world - no preprocessing possible
- No gameplay or artwork changed
  - Already had local lights (but no shadows)
  - All lights are omnis - rampant duelling!
  - Truly robust
- More details at ww.eelpi.gotdns.org
  - Subject far too big to cover in the time

Soft Shadows

Soft shadows

- More realism
- Hides aliasing
  - Allows use of lower-rez shadowbuffers
- Gives depth cues
  - Further from lightsource = softer

Simple blurring

- Not for realism, just to reduce aliasing
- Percentage Closer Filtering
  - Make multiple samples from shadowbuffer
  - Test each sample for shadowed/lit
  - Result is the percentage of lit
- Requires a lot of samples
  - 64 samples = 6 bits of grey
- Can make it adaptive
  - Only sample lots at shadow edges

Depth-dependent softness

- Simulates an area light
  - Objects can occlude all or part of the light
  - Penumbra formed at partial occlusion
- Penumbra wedges (Assarson)
  - Fusion of shadowbuffers and volume shadows
  - Needs watertight manifold meshes
  - Needs lots of fillrate

Depth-dependent softness

- Smoothies (Chan, Durand)
  - Not physically correct
    - Shadows only blur outwards, not inwards
  - Less fillrate demand than Penumbra Wedges
  - Hides spatial aliasing really well
  - But still needs watertight manifolds
    - Uses them to find silhouette edges
- Willem de Boer's work
  - Similar to Smoothies
  - But generates edges in image-space
  - So no geometric restrictions

Summary

- ID+depth = best of both world
  - This seems like the right solution - solved!
- Frustum choice is tricky
  - There are solutions, but they're all complex
  - Still some engineering problems to solve
- Soft shadows are very tricky
  - Lots of interesting research
  - None works completely yet
  - Still expensive
  - But progress is swift!

Questions?

More available from www.eelpi.gotdns.org