I'd Like To Teach The World To Code: Scripting In Second Life

Dr Jim Purbrick, Technical Director, Linden Lab

JAOO Aarhus 2008-10-01

Second Life

- 3D Virtual World
- Avatars
- Persistent
- Massively Multi User
- Resident Built
- Resident Textured
- Resident Scripted
- Resident Animated
- Resident Owned
- Linden Lab Sells Land And Services

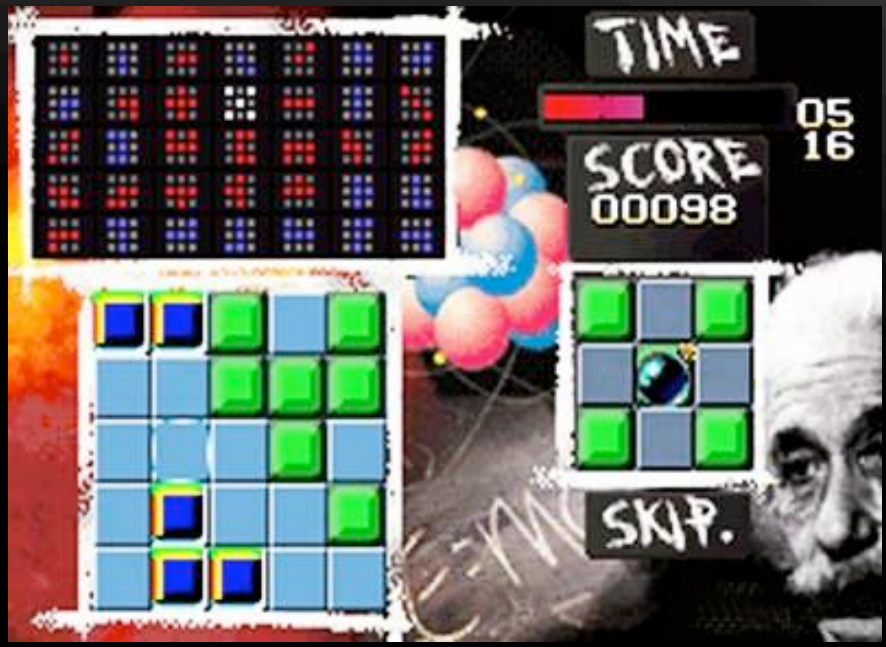
Not A Game

← World

← Game

SECOND LIFE

Tringo



SECOND LIFE

Live Performance

SECOND LIFE

Movie Making

SECOND LIFE

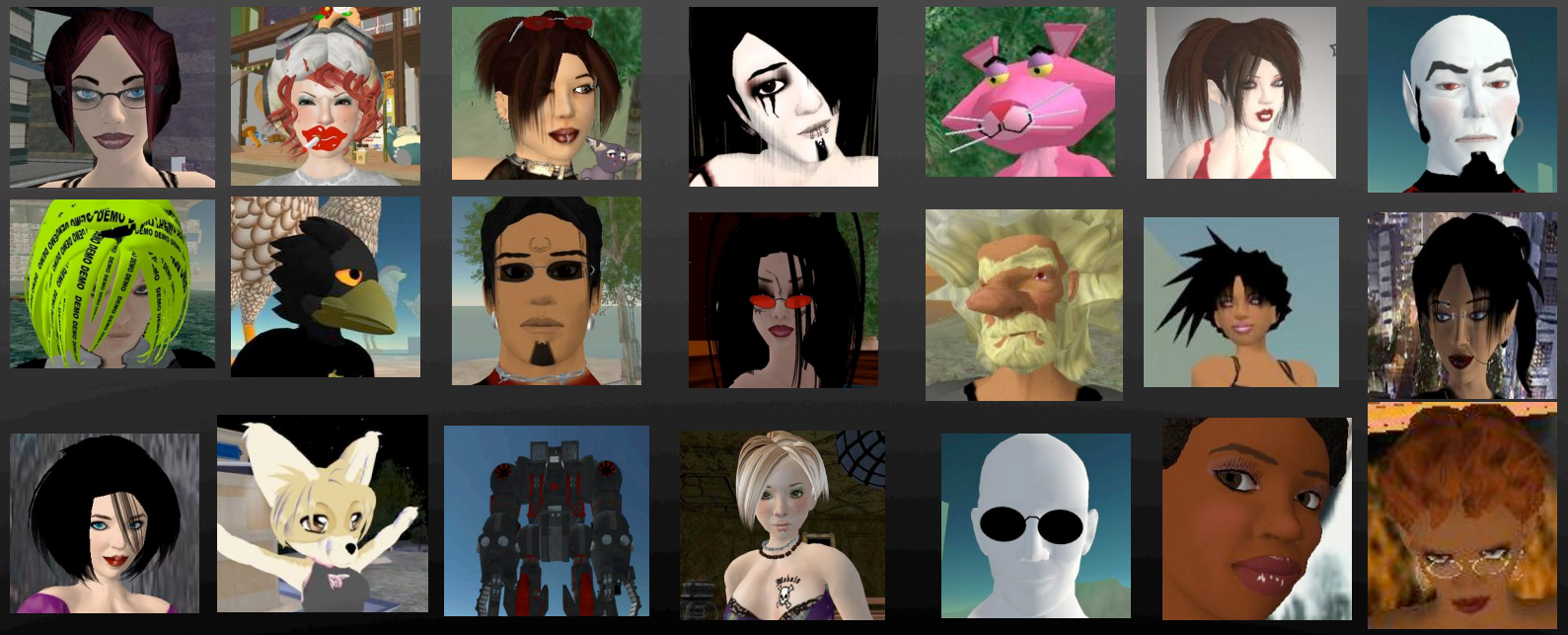
Conferences

Charity

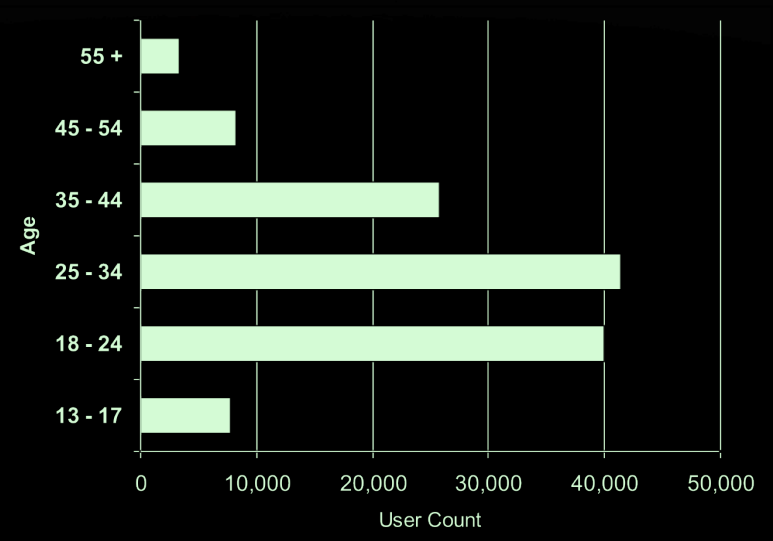
SECOND LIFE

Experimentation

Ordinary People

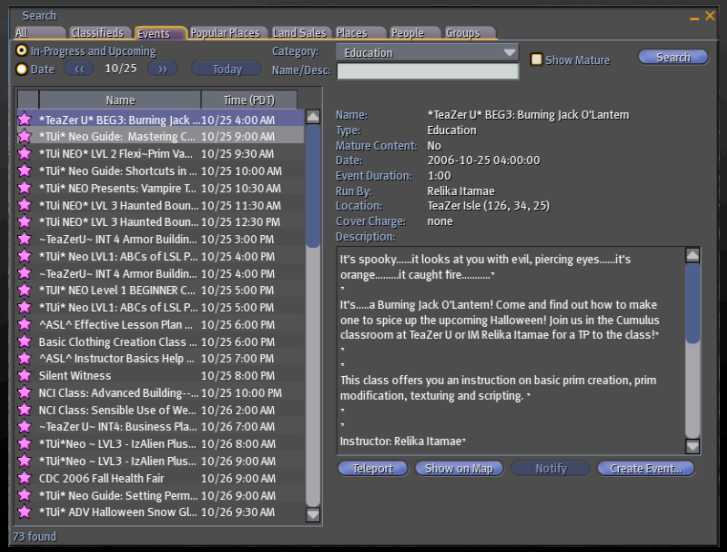


- 15M Residents
- 11K² Miles
- 43% Female
- Median Age 33
- 54% European
- 1M USD Per Day



SECOND LIFE

Second Life Taught The World To Code



- 30M Running Scripts
- 15% Residents Code
- 25M Source Files
- 2.5B Lines Of Code

SECOND LIFE

Scripting Simulation

- Independent scripted agents
- Event queues
- Messaging between agents

Image The New Media Consortium

Scripting Second Life

- Untrusted code
- Migration
- Resource limits

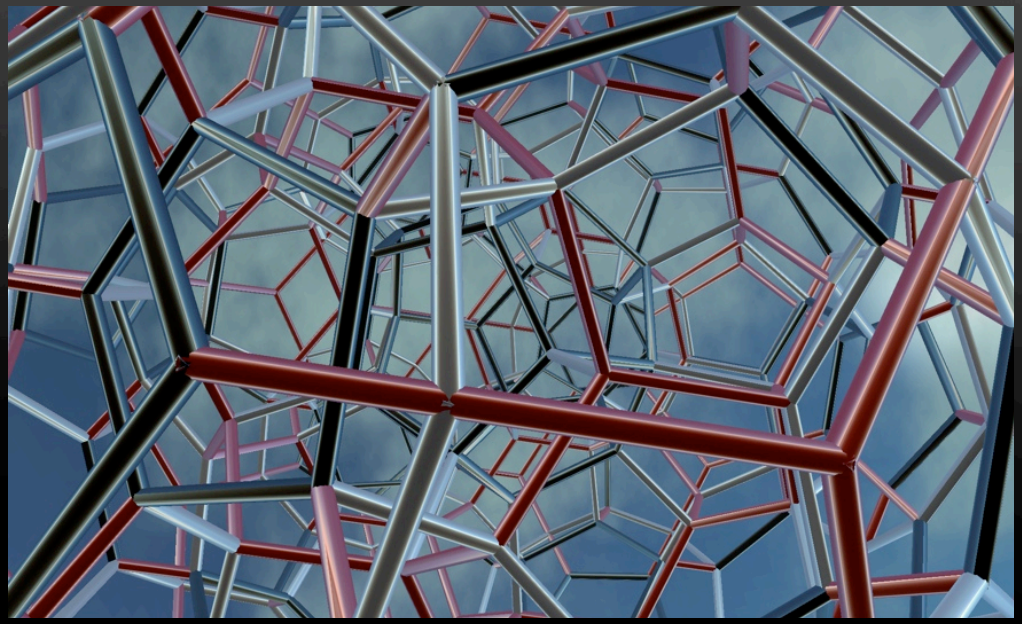


Image Suzanne Graves

SECOND LIFE®

LSL

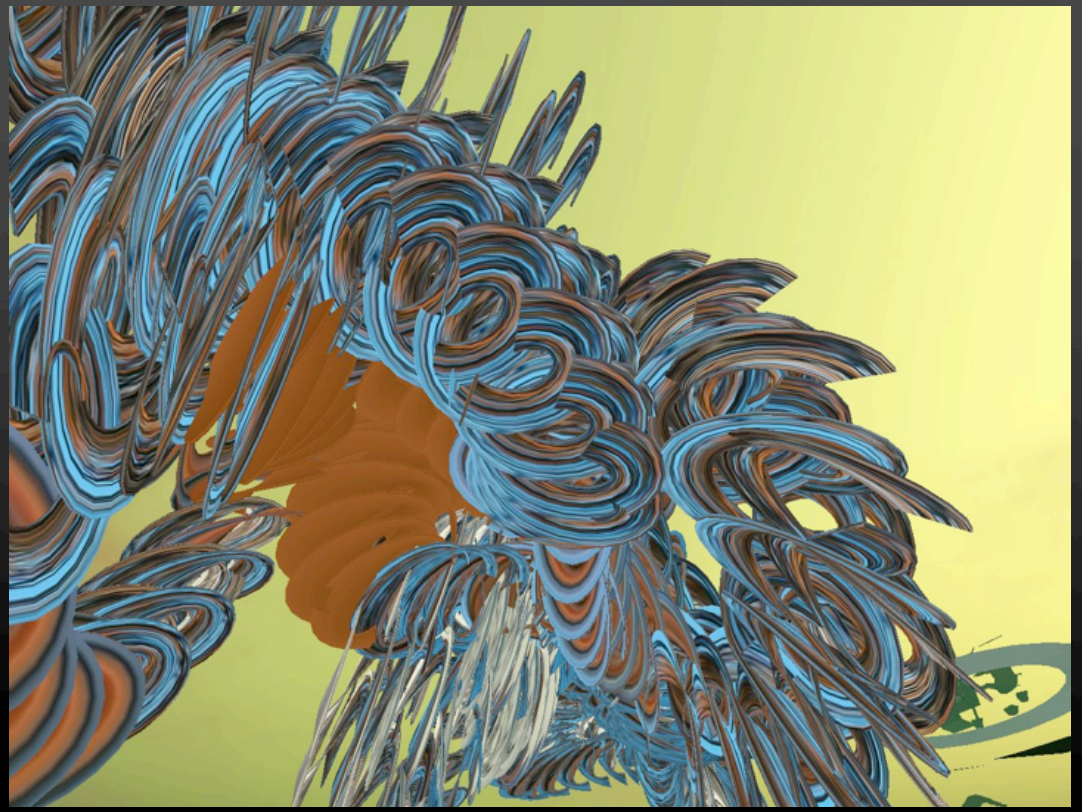


Image Bettina Tizzy

- Scripts in prims
- Prims linked to make objects
- Sensors and effectors via library calls
- Communication via IM, email, XMLRPC, HTTP
- Slow, proprietary C-like language
- No arrays, objects, classes, exceptions, libraries

SECOND LIFE

Mono

- Run LSL On CLI
- Allow CLI Languages
- Allow CLI Libraries
- Bytecode Verifier
- High Speed JIT
- Versioning
- Dynamic Linking
- Cross Platform
- Open Source

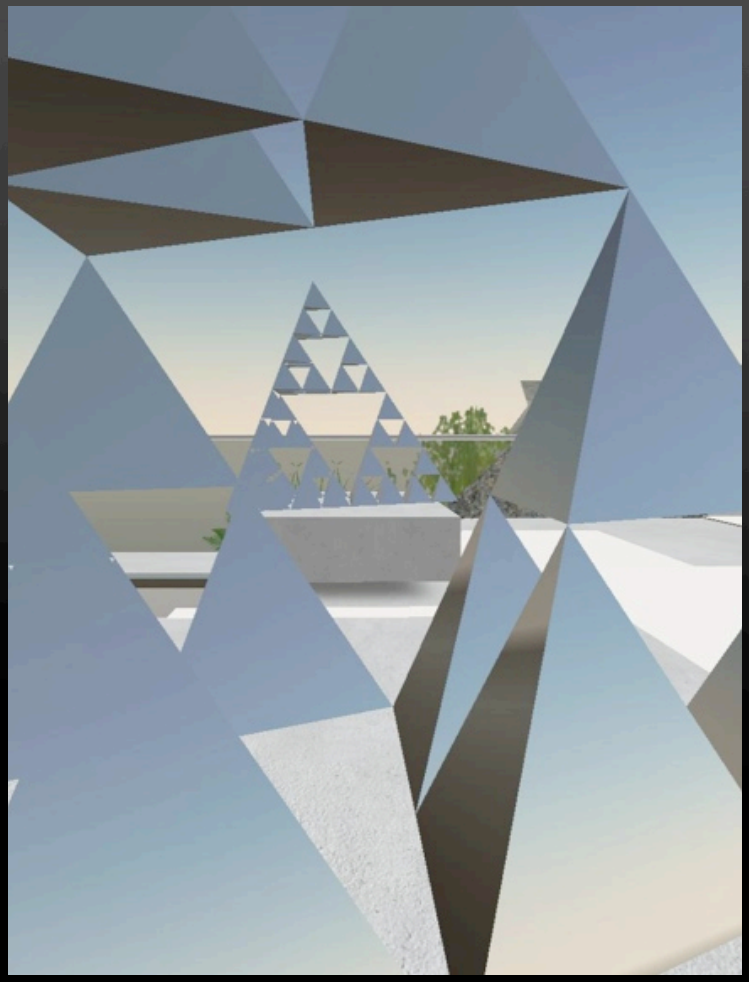


Image The New Media Consortium

SECOND LIFE

Problem 1: Concurrency/Mobility

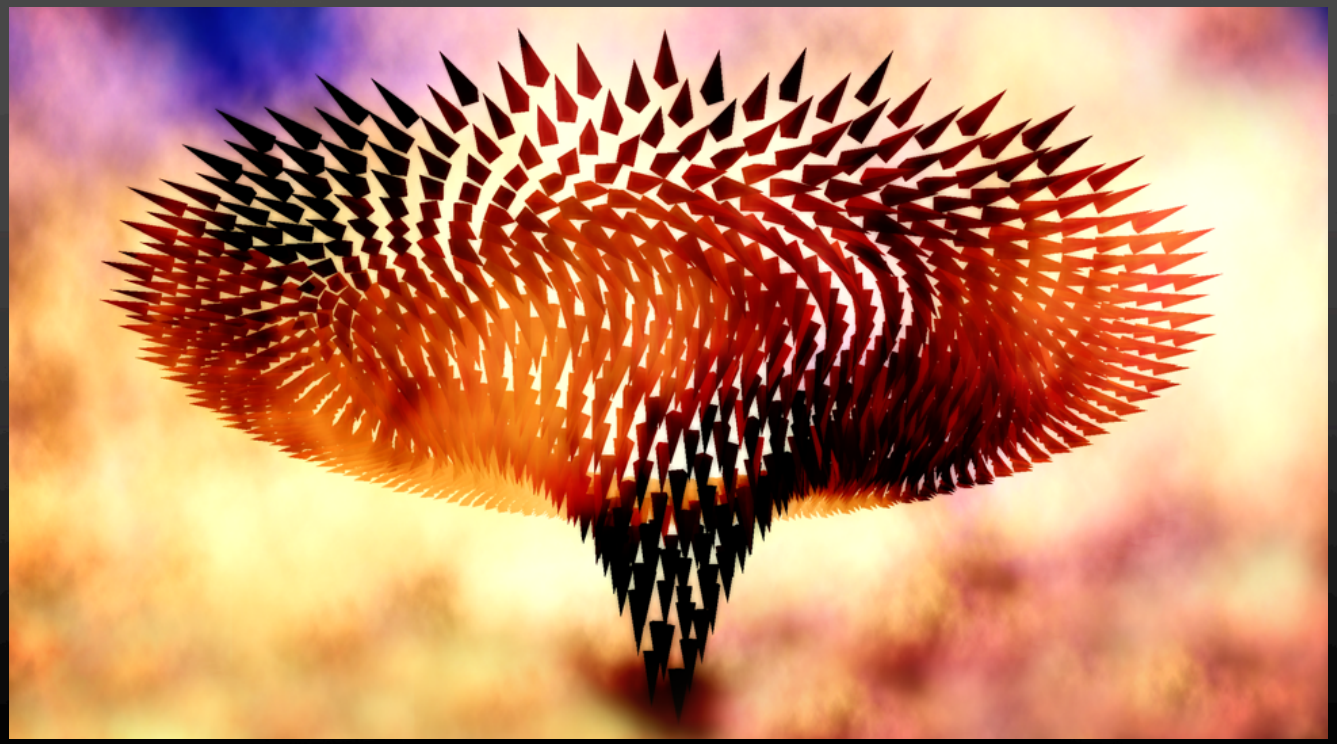


Image Ganymedes Costagravas

- Need to support 2000-10000 concurrent scripts
- Long running methods can be legitimate
- Need to avoid denial of service
- Mono uses (60-1000?) OS threads
- Need to migrate and persist running threads
- Need to migrate and persist stack

SECOND LIFE

Hack 1: Microthreading

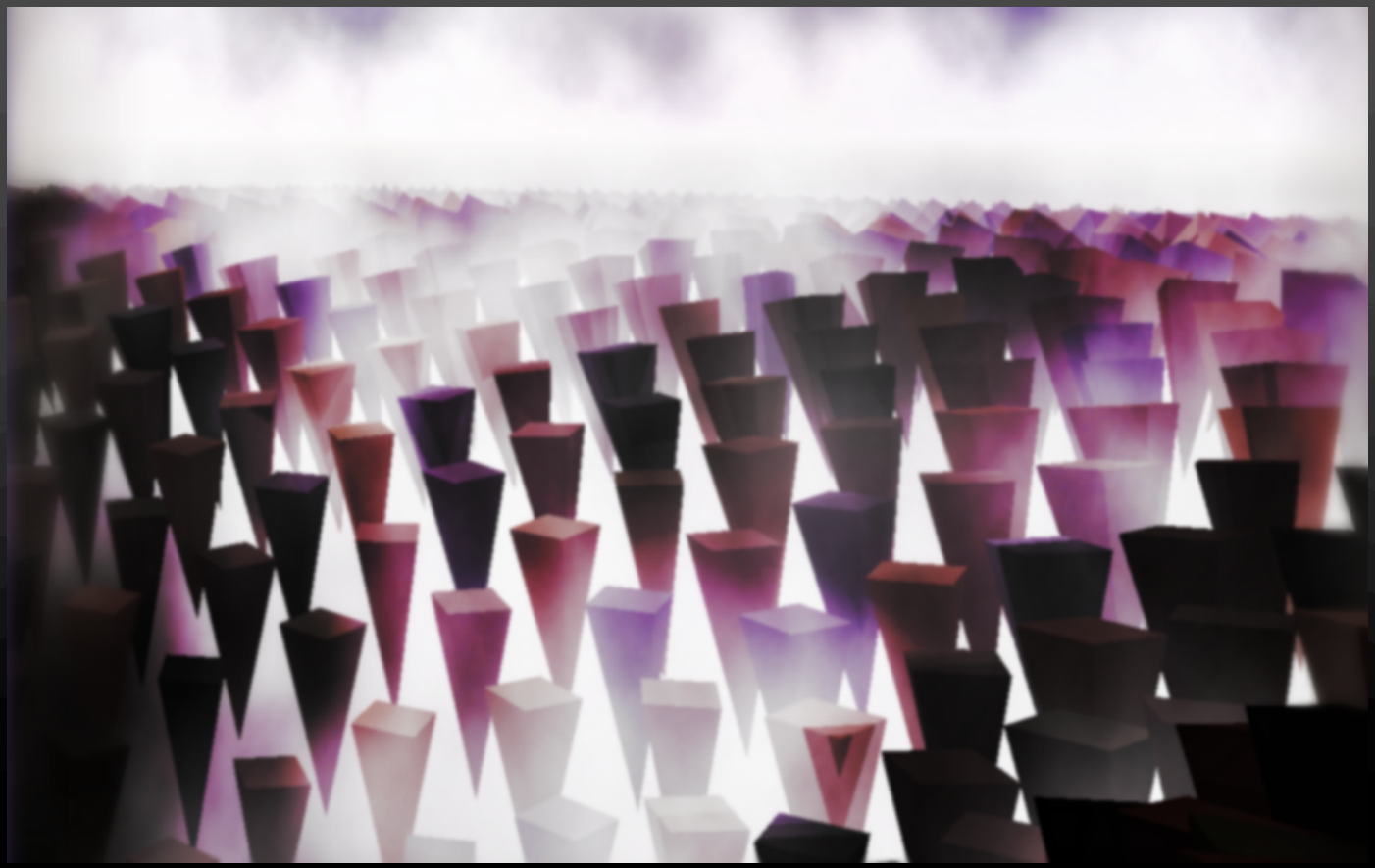


Image Ganymedes Costagravas

- Induce stack state (Verification)
- Rewrite bytecode (Reflection.Emit, RAIL, cecil)
- Inject microthreading (JavaGoX, Brakes)
- Serialize stack and heap, transfer assemblies

SECOND LIFE

```
.method public static hidebysig default int32 fib (int32 n)  cil managed
{
        IL_0000:  call bool class UThread::IsRestoring()
        IL_0005:  brfalse IL_006d

        IL_000a:  call class UThread.UThreadStackFrame class UThread::Pop()
        IL_000f:  castclass FibfibFrame

        IL_0024:  ldfld int32 FibfibFrame::pc
        IL_0029:  switch (
          IL_003a,
          IL_0053,
          IL_0082)
        IL_003a:  call class UThreadStackFrame class UThread::Peek()
        IL_003f:  callvirt instance object class UThreadStackFrame::Resume()
        IL_0044:  unbox [mscorlib]System.Int32
        IL_0049:  ldobj [mscorlib]System.Int32
        IL_004e:  br IL_0093

        IL_006d:  call bool class UThread::IsSaveDue()
        IL_0072:  brfalse IL_0082

        IL_0077:  ldc.i4 2
        IL_007c:  stloc.0
        IL_007d:  br IL_00ca


        IL_008e:  call int32 class Fib::fib(int32)
        IL_0093:  call bool class UThread::IsSaving()
        IL_0098:  brfalse IL_00a9

        IL_009d:  pop
        IL_009e:  ldc.i4 0
        IL_00a3:  stloc.0
        IL_00a4:  br IL_00ca

        IL_00cb:  ldloc.0
        IL_00cc:  ldarg 0

        IL_00d2:  ldloc.0
        IL_00d3:  ldloc.1
        IL_00d4:  newobj instance void class FibfibFrame::.ctor(int32, int32, int32, int32)
        IL_00d9:  pop
        IL_00da:  ldloc.1
        IL_00db:  ret

} // end of method Fib::fib
```

Restoring?

Resume Next Stack frame

Save Due?

Saving?

Build Stack Frame

Problem 2: Code Unloading

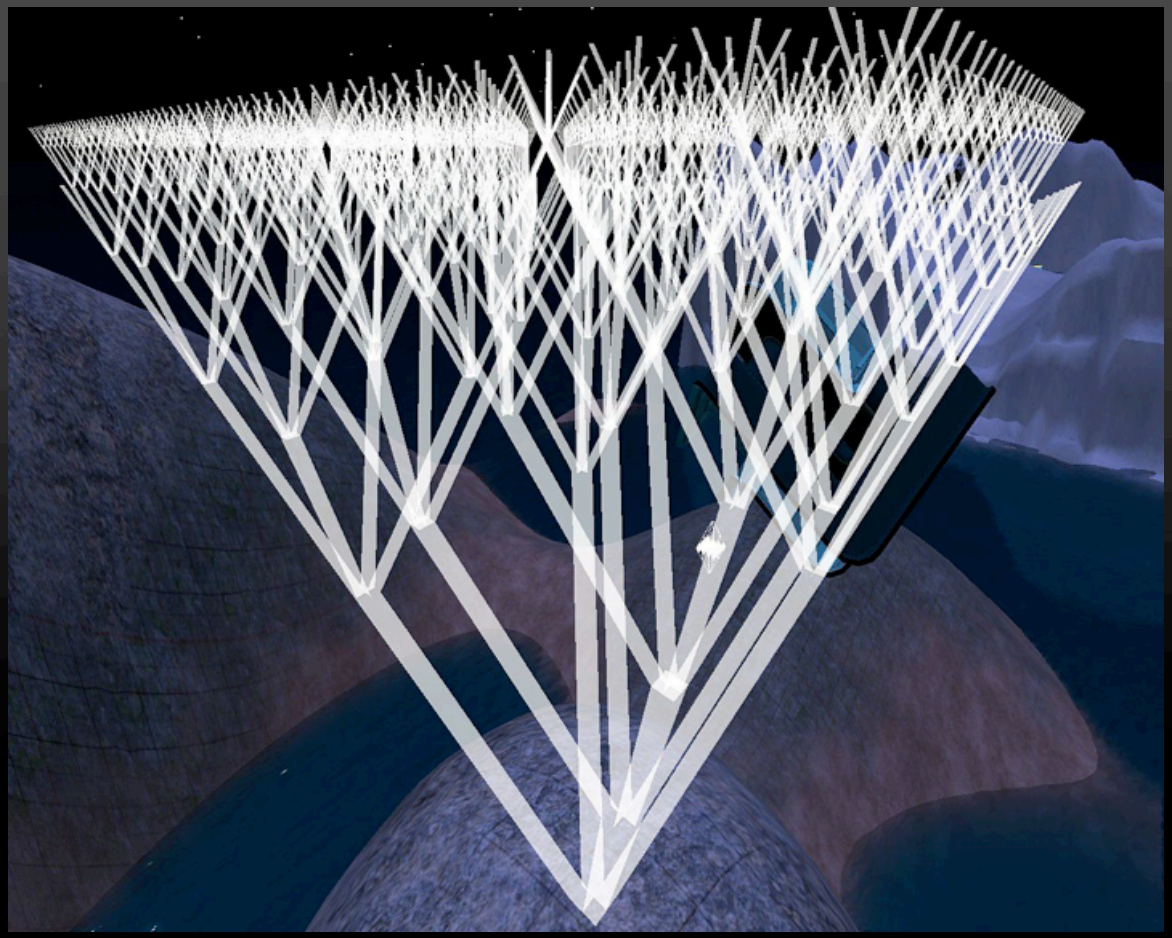


Image Bettina Tizzy

- Long running simulator
- 8000 scripts/day created in sandpits
- 128MB rubbish code/day/simulator
- CLI doesn't allow assembly unloading

SECOND LIFE

Hack 2: Scavenging AppDomains

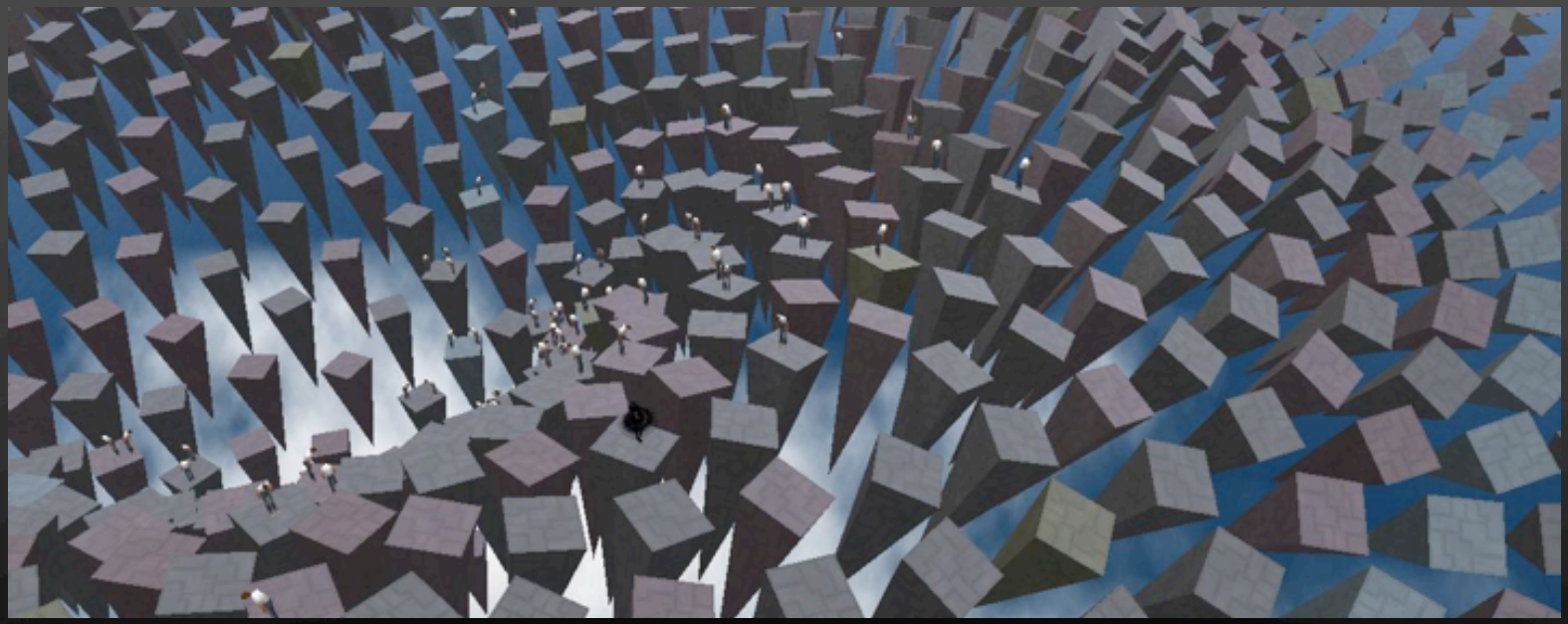
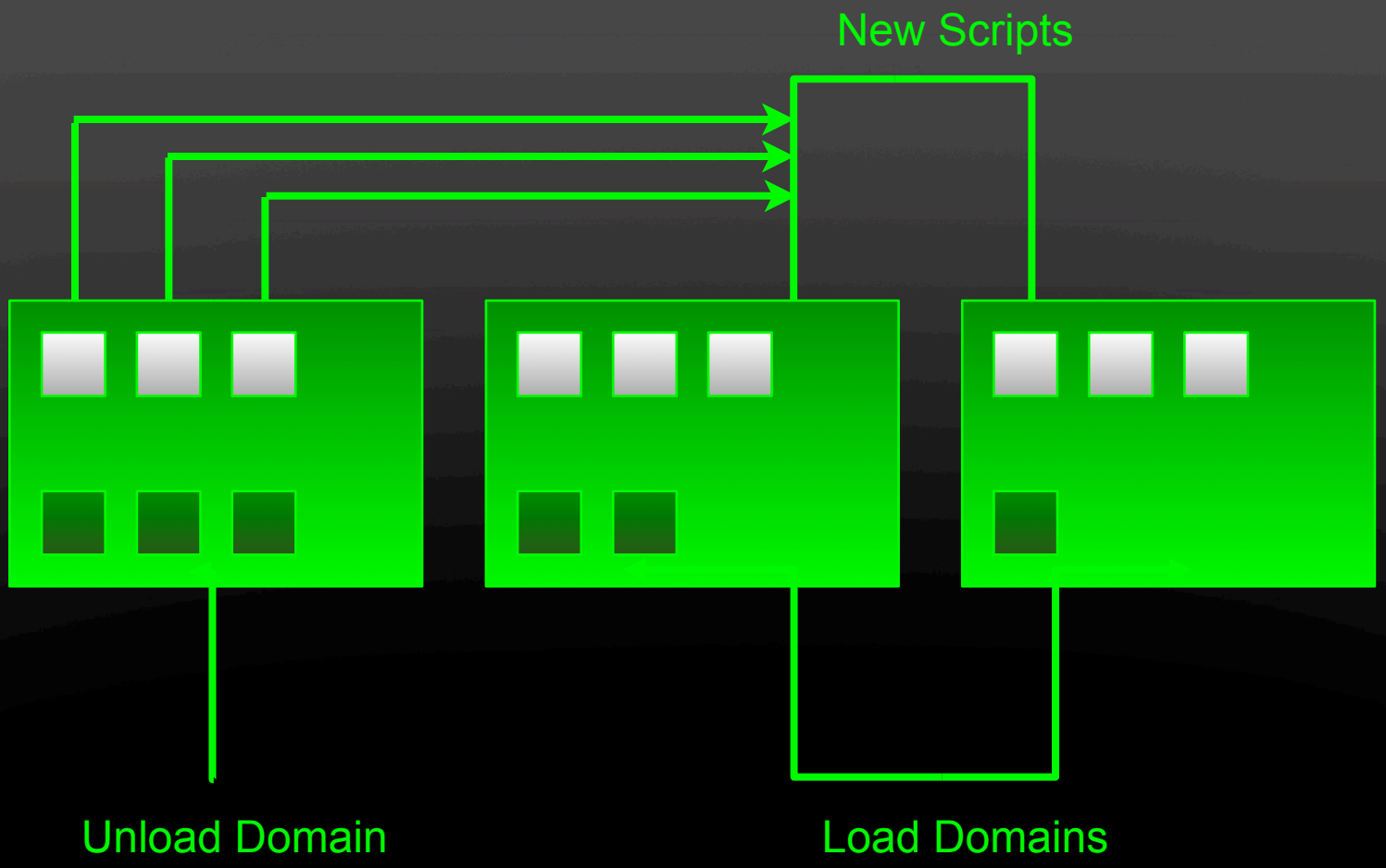


Image Ka Rasmuson

- CLI does allow domain unloading
- Domain/Assembly -> 2000-10000 domains
- N Domains
- Load assemblies in to all domains
- Track unused assemblies in domains
- Migrate running scripts to new domain
- Unload domain
- Create new domain

SECOND LIFE


New Scripts
Unload Domain
Load Domains
App Domain
Running Script
Unused Assembly

Problem 3: Resource Accounting

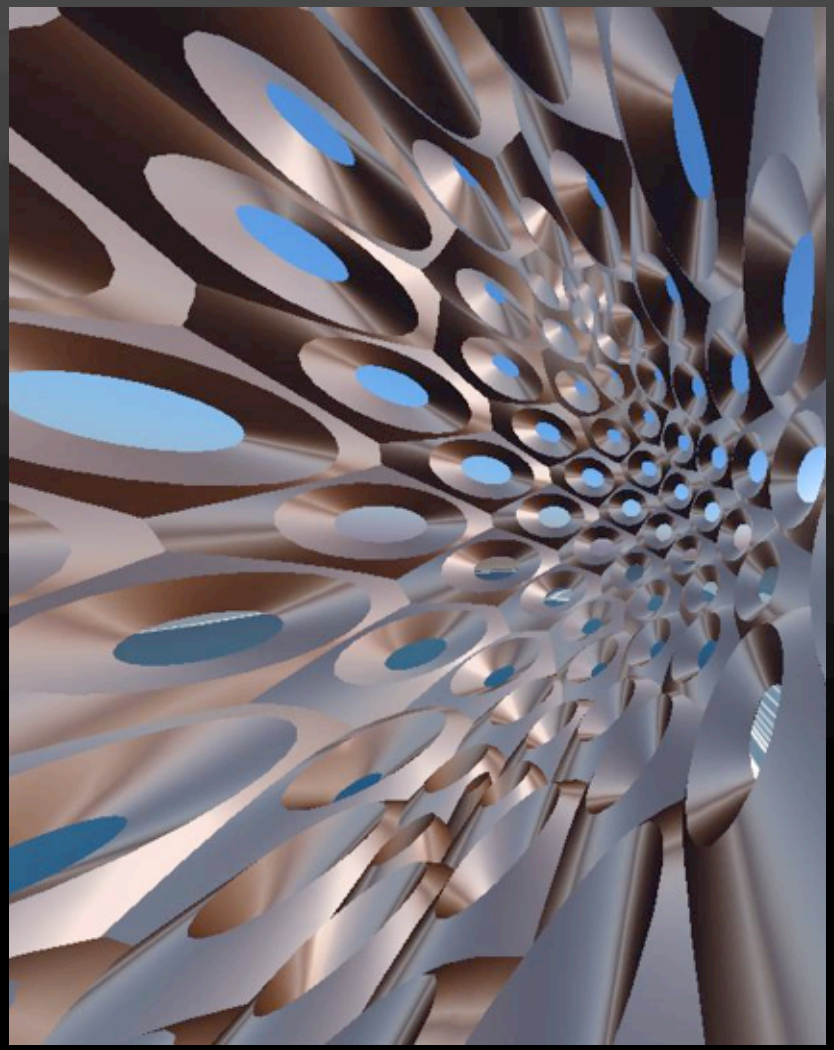


Image juria yoshikawa

- Limit resources per script
- Avoid non-CPU DOS
- HTTP Calls
- Emails
- (Self) Replication
- Memory
- (Stack + Heap + Text) <= 16KB

SECOND LIFE

Hack 3: Subvert The Profiler

Image The New Media Consortium

- Mono profiling API
- On script creation add size to current script size
- On script size >= 16KB walk object tree

SECOND LIFE

Future

- C#/F#
- IronPython/Ruby
- Resource pools
- HTTP servers
- Libraries

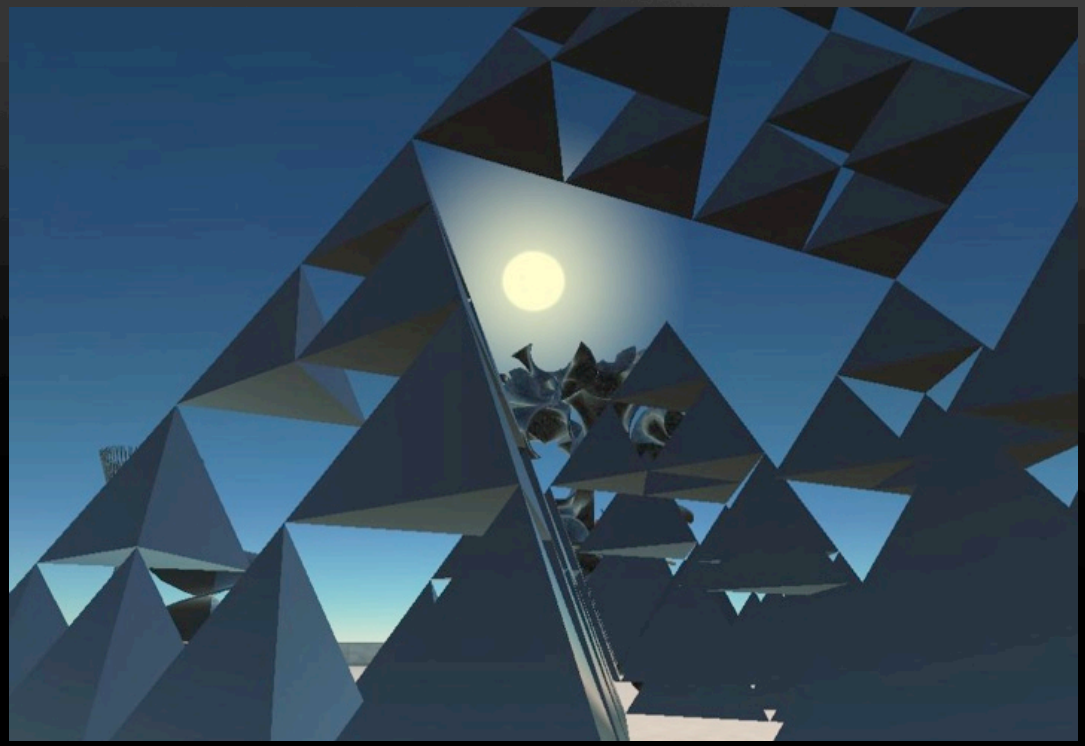


Image The New Media Consortium

SECOND LIFE

Conclusions

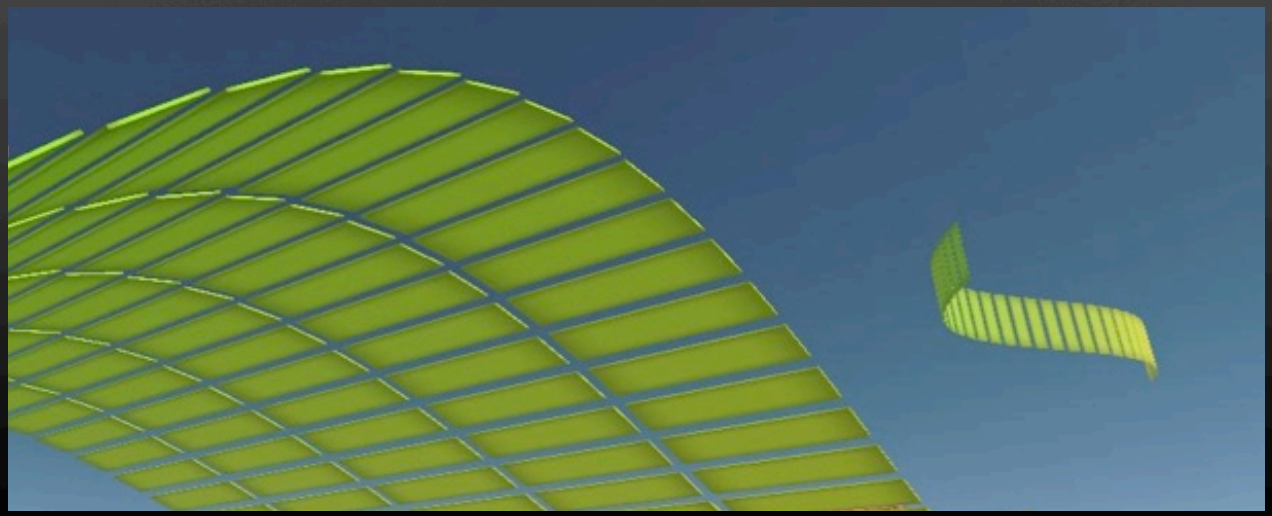


Image Atomo Hosho

- **Simulations are different**
- **Web sandboxes**
- **Adapt**