atf

authoring tools framework

Authoring Tools Framework

Open Source

Sony Computer Entertainment America

Agenda

- Who are you?
- What is ATF?
- Who uses ATF?
- Components of ATF
- Pros and Cons
- Lessons learned from shared code development
- Q & A

Authoring Tools Framework

- Create PC-based game development tools
- C#, .NET 4.0
- You choose the components you want, customize them, or add your own new ones
- Used by most Sony Computer Entertainment 1st-party studios
- Open source on GitHub! http://github.com/SonyWWS

Authoring Tools Framework

Adopters

(Partial List)

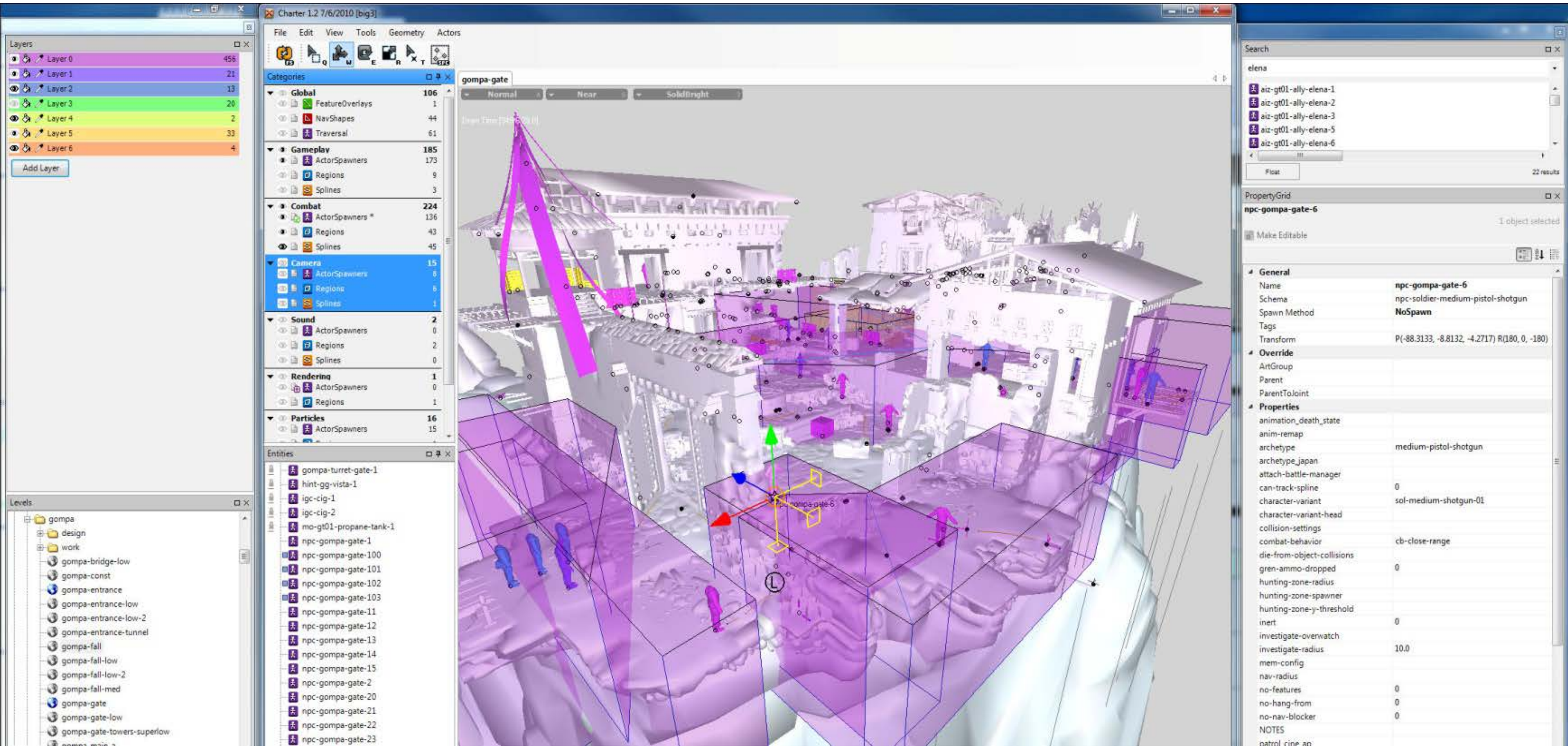
- Naughty Dog – *The Last of Us*
  - Charter Level editor
  - Surfer Shader Editor
- Guerrilla Games – *Killzone: Shadow Fall*
  - CoreText Editor – object and cinematic sequence editor
- Quantic Dream – *Beyond: Two Souls*
  - Four StateMachine-based tools
- Santa Monica Studios – *God of War*
  - Metrics – performance analyzer
  - CreatureEditor – animation blending tool
- Bend Game Studio – *Uncharted: Golden Abyss on PS Vita*
  - Level editor, etc.
- Zindagi
  - StateMachine, SLED, LiveEdit

Adopters

(Partial list)

- Cambridge Studio
  - LittleBigPlanet PSP's Level Editor and Moderation Viewer
- Home
  - Home Scene Editor
- ATG
  - Sulpha – sound visualization and editing
  - Nexus – Animation blending tool
- TNT
  - SLED – LUA IDE & debugger
  - StateMachine editor
  - SCREAM Tool – audio effects authoring tool
- Liverpool Studios
  - LevelEditor, StateMachine, SLED
- Zipper Interactive
  - Atlas – Level editor using ATF 3 and SlimDX

Naughty Dog's *Charter* Level Editor


Charter 1.2 7/6/2010 [big3]
File Edit View Tools Geometry Actors
gompa-gate
Normal
Near
SolidBright
Add Layer
Float

Layers

Layer 0	456
Layer 1	21
Layer 2	13
Layer 3	20
Layer 4	2
Layer 5	33
Layer 6	4

Categories

Global	**106**
FeatureOverlays	1
NavShapes	44
Traversal	61
Gameplay	**185**
ActorSpawners	173
Regions	9
Splines	3
Combat	**224**
ActorSpawners *	136
Regions	43
Splines	45
Camera	**15**
ActorSpawners	8
Regions	6
Splines	1
Sound	**2**
ActorSpawners	0
Regions	2
Splines	0
Rendering	**1**
ActorSpawners	0
Regions	1
Particles	**16**
ActorSpawners	15

Entities

- gompa-turret-gate-1
- hint-gg-vista-1
- igc-cig-1
- igc-cig-2
- mo-gt01-propane-tank-1
- npc-gompa-gate-1
- npc-gompa-gate-100
- npc-gompa-gate-101
- npc-gompa-gate-102
- npc-gompa-gate-103
- npc-gompa-gate-11
- npc-gompa-gate-12
- npc-gompa-gate-13
- npc-gompa-gate-14
- npc-gompa-gate-15
- npc-gompa-gate-2
- npc-gompa-gate-20
- npc-gompa-gate-21
- npc-gompa-gate-22
- npc-gompa-gate-23

Levels

- gompa
  - design
  - work
  - gompa-bridge-low
  - gompa-const
  - gompa-entrance
  - gompa-entrance-low
  - gompa-entrance-low-2
  - gompa-entrance-tunnel
  - gompa-fall
  - gompa-fall-low
  - gompa-fall-low-2
  - gompa-fall-med
  - gompa-gate
  - gompa-gate-low
  - gompa-gate-towers-superlow

Search

elena

- aiz-gt01-ally-elena-1
- aiz-gt01-ally-elena-2
- aiz-gt01-ally-elena-3
- aiz-gt01-ally-elena-5
- aiz-gt01-ally-elena-6

22 results

PropertyGrid

npc-gompa-gate-6

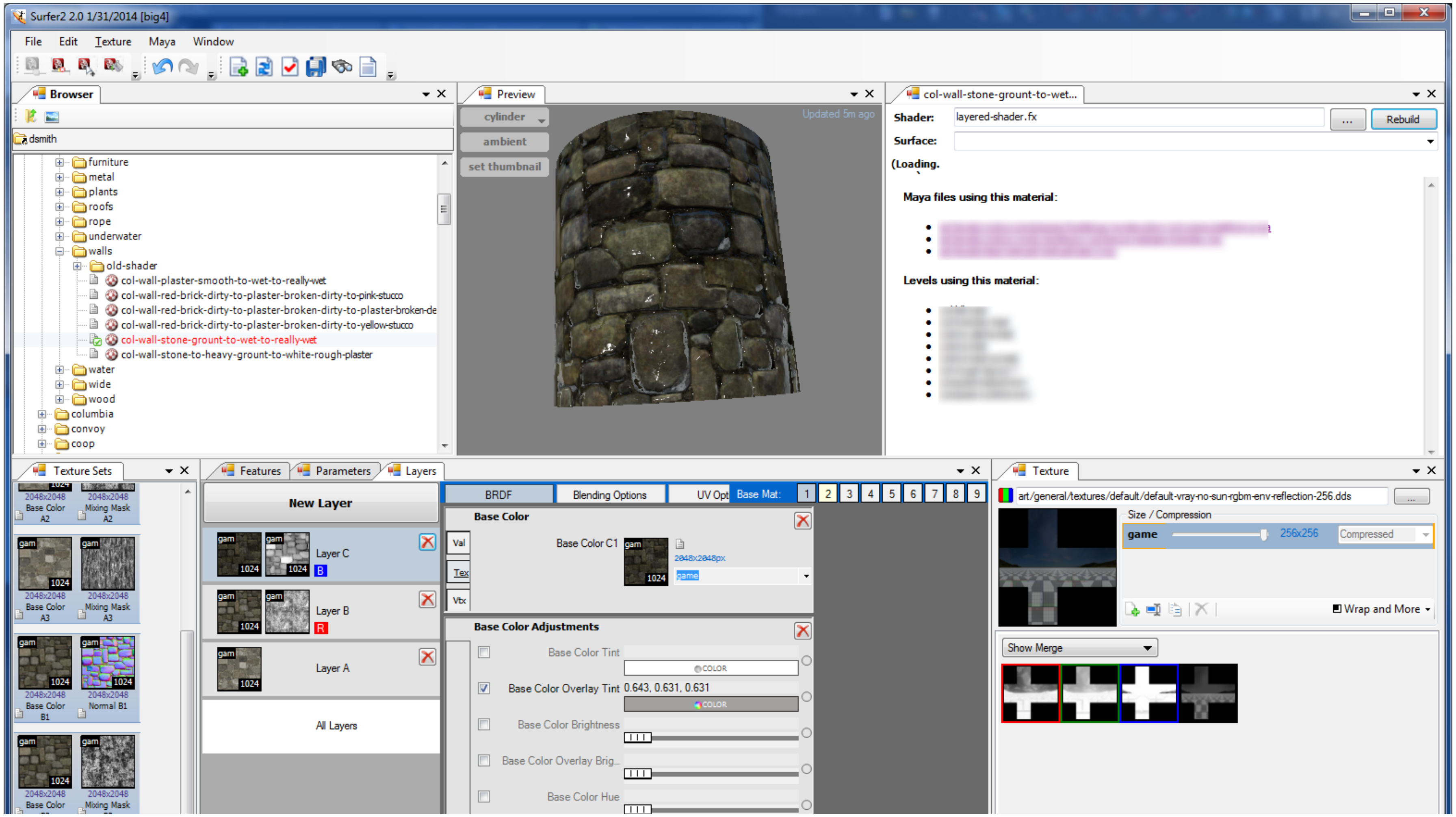
1 object selected

Make Editable

General	
Name	**npc-gompa-gate-6**
Schema	npc-soldier-medium-pistol-shotgun
Spawn Method	**NoSpawn**
Tags	
Transform	P(-88.3133, -8.8132, -4.2717) R(180, 0, -180)
Override	
ArtGroup	
Parent	
ParentToJoint	
Properties	
animation_death_state	
anim-remap	
archetype	medium-pistol-shotgun
archetype_japan	
attach-battle-manager	
can-track-spline	0
character-variant	sol-medium-shotgun-01
character-variant-head	
collision-settings	
combat-behavior	cb-close-range
die-from-object-collisions	
gren-ammo-dropped	0
hunting-zone-radius	
hunting-zone-spawner	
hunting-zone-y-threshold	
inert	0
investigate-overwatch	
investigate-radius	10.0
mem-config	
nav-radius	
no-features	0
no-hang-from	0
no-nav-blocker	0
NOTES	
patrol cine ap	

"There was ATF code running behind every shader tweak and enemy placement in *The Last of Us*." – Dave Smith, Naughty Dog tools programmer

Naughty Dog's *Surfer* Shader Editor



dsmith
- furniture
- metal
- plants
- roofs
- rope
- underwater
- walls
  - old-shader
  - col-wall-plaster-smooth-to-wet-to-really-wet
  - col-wall-red-brick-dirty-to-plaster-broken-dirty-to-pink-stucco
  - col-wall-red-brick-dirty-to-plaster-broken-dirty-to-plaster-broken-de
  - col-wall-red-brick-dirty-to-plaster-broken-dirty-to-yellow-stucco
  - col-wall-stone-grount-to-wet-to-really-wet
  - col-wall-stone-to-heavy-grount-to-white-rough-plaster
- water
- wide
- wood
- columbia
- convoy
- coop

Shader: layered-shader.fx

Surface:

(Loading.

Maya files using this material:

[redacted]

Levels using this material:

[redacted]

Texture Sets	
2048x2048 Base Color A2	2048x2048 Mixing Mask A2
2048x2048 Base Color A3	2048x2048 Mixing Mask A3
2048x2048 Base Color B1	2048x2048 Normal B1
2048x2048 Base Color	2048x2048 Mixing Mask

Base Color

Base Color C1 — 2048x2048px — game

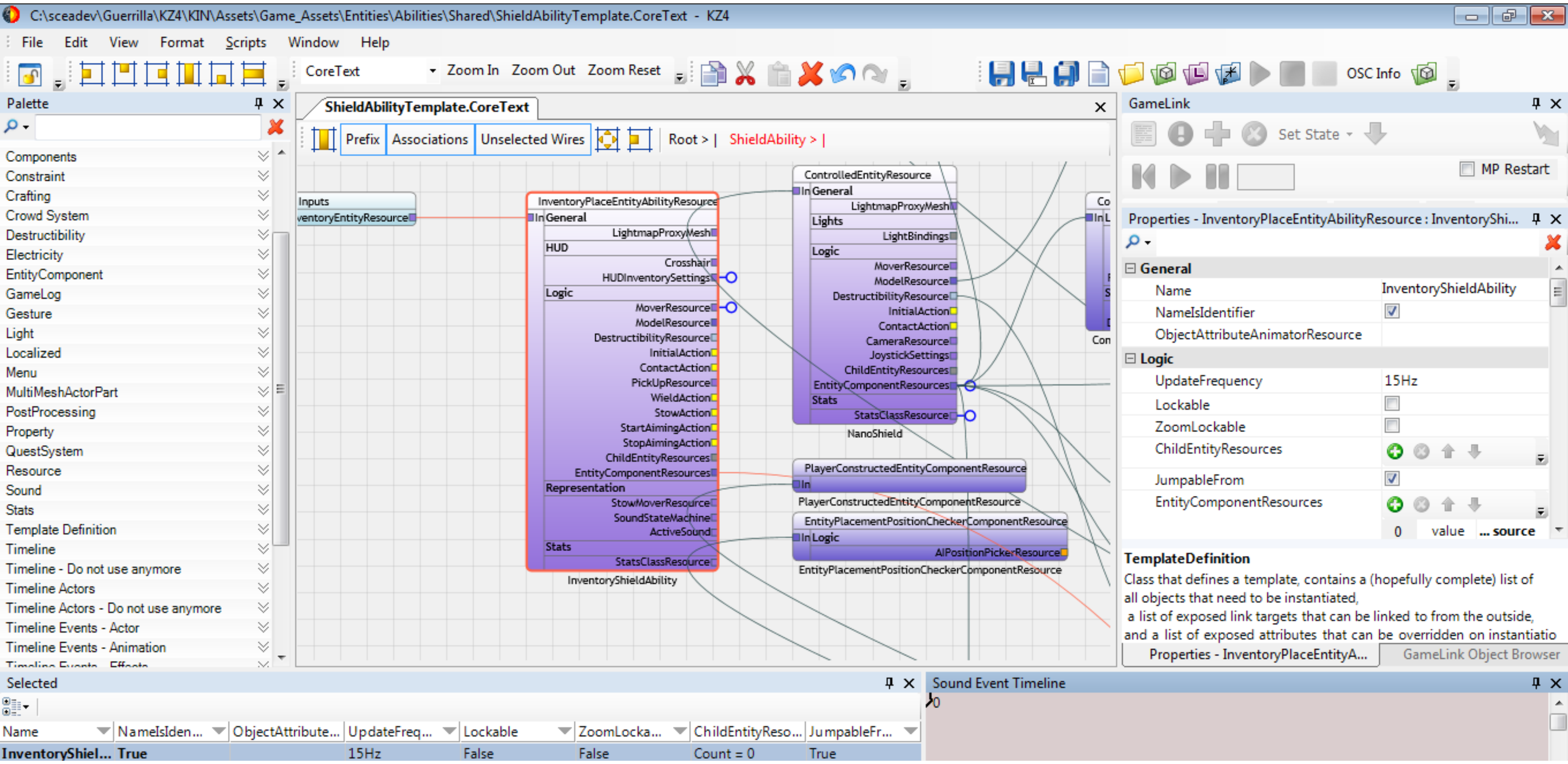
Base Color Adjustments

☐ Base Color Tint	COLOR
☑ Base Color Overlay Tint	0.643, 0.631, 0.631
☐ Base Color Brightness	
☐ Base Color Overlay Brig...	
☐ Base Color Hue	

art/general/textures/default/default-vray-no-sun-rgbm-env-reflection-256.dds

game — 256x256 — Compressed

Guerrilla's *CoreText Editor*



C:\sceadev\Guerrilla\KZ4\KIN\Assets\Game_Assets\Entities\Abilities\Shared\ShieldAbilityTemplate.CoreText - KZ4

Properties - InventoryPlaceEntityAbilityResource : InventoryShi...

General	
Name	InventoryShieldAbility
NameIsIdentifier	✓
ObjectAttributeAnimatorResource	
Logic	
UpdateFrequency	15Hz
Lockable	☐
ZoomLockable	☐
ChildEntityResources	
JumpableFrom	✓
EntityComponentResources	0 value ... source

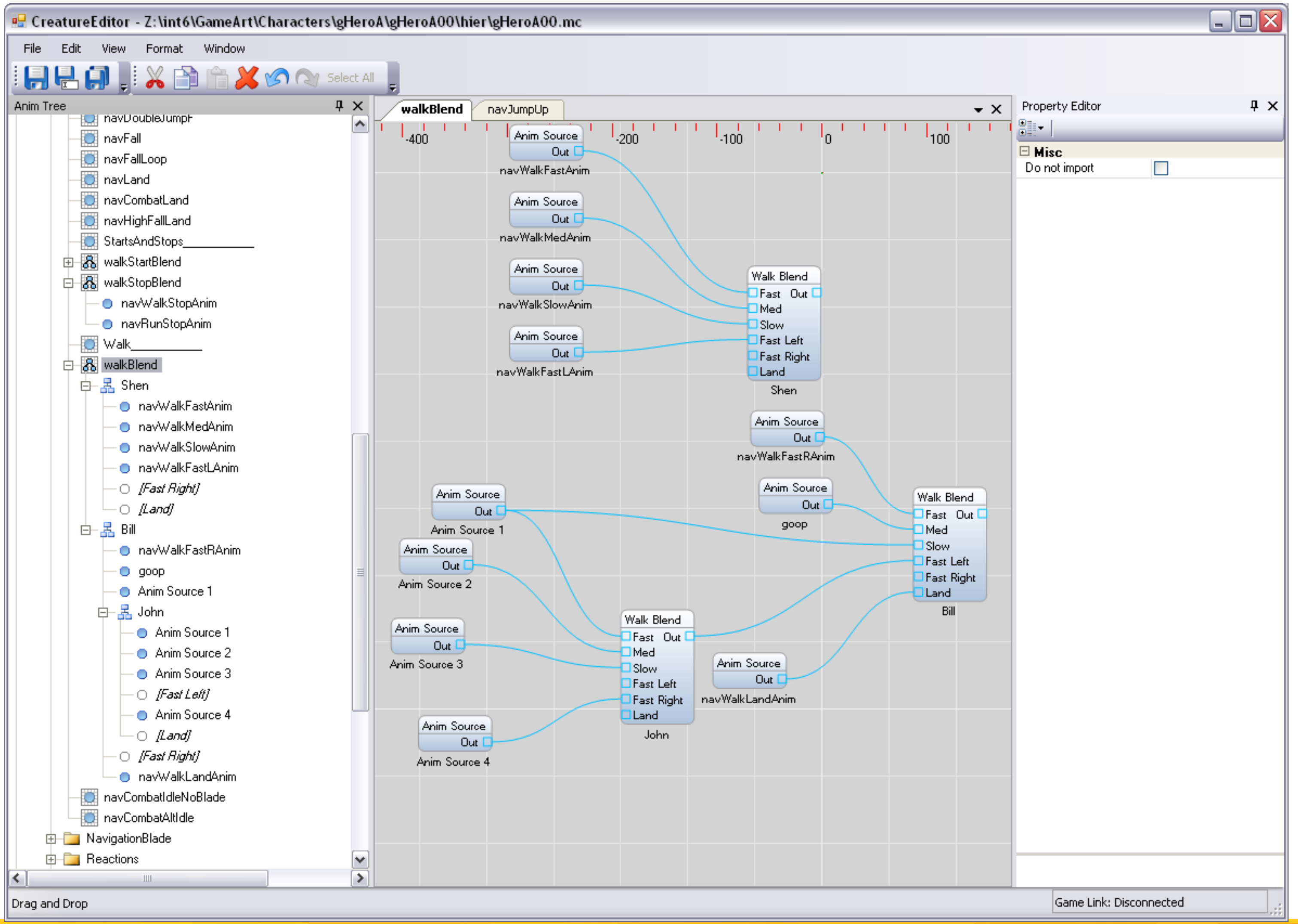
TemplateDefinition
Class that defines a template, contains a (hopefully complete) list of all objects that need to be instantiated,
a list of exposed link targets that can be linked to from the outside,
and a list of exposed attributes that can be overridden on instantiatio

Selected

Name	NameIsIden...	ObjectAttribute...	UpdateFreq...	Lockable	ZoomLocka...	ChildEntityReso...	JumpableFr...
InventoryShiel...	True		15Hz	False	False	Count = 0	True

Sequence and object editor for the Killzone series, including the *Killzone: Shadow Fall* PS4 launch title

Santa Monica Studios' *Creature Editor*



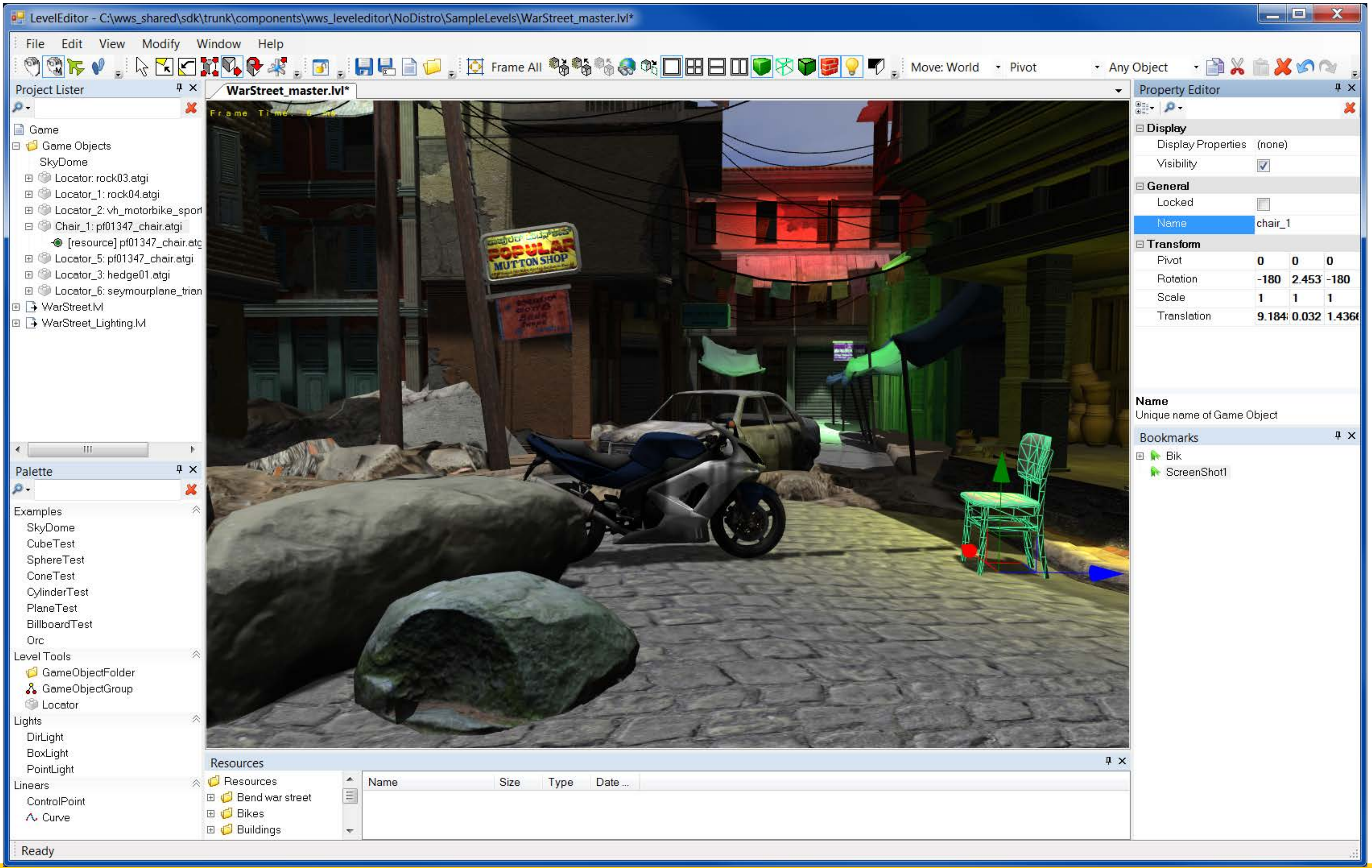
Anim Tree:

- navDoubleJumpF
- navFall
- navFallLoop
- navLand
- navCombatLand
- navHighFallLand
- StartsAndStops
- walkStartBlend
- walkStopBlend
  - navWalkStopAnim
  - navRunStopAnim
- Walk
- walkBlend
  - Shen
    - navWalkFastAnim
    - navWalkMedAnim
    - navWalkSlowAnim
    - navWalkFastLAnim
    - [Fast Right]
    - [Land]
  - Bill
    - navWalkFastRAnim
    - goop
    - Anim Source 1
    - John
      - Anim Source 1
      - Anim Source 2
      - Anim Source 3
      - [Fast Left]
      - Anim Source 4
      - [Land]
    - [Fast Right]
    - navWalkLandAnim
- navCombatIdleNoBlade
- navCombatAltIdle
- NavigationBlade
- Reactions

Graph (walkBlend):

- Walk Blend "Shen" (Fast, Med, Slow, Fast Left, Fast Right, Land; Out): Anim Source navWalkFastAnim → Fast; navWalkMedAnim → Med; navWalkSlowAnim → Slow; navWalkFastLAnim → Fast Left
- Walk Blend "John" (Fast, Med, Slow, Fast Left, Fast Right, Land; Out): Anim Source 1 → Fast; Anim Source 2 → Med; Anim Source 3 → Slow; Anim Source 4 → Fast Right
- Walk Blend "Bill" (Fast, Med, Slow, Fast Left, Fast Right, Land; Out): navWalkFastRAnim → Fast; goop → Med; Anim Source 1 → Slow; John → Fast Left; navWalkLandAnim → Land

LevelEditor (by Game Tech Group)



LevelEditor - C:\wws_shared\sdk\trunk\components\wws_leveleditor\NoDistro\SampleLevels\WarStreet_master.lvl*

Project Lister

- Game
- Game Objects
  - SkyDome
  - Locator: rock03.atgi
  - Locator_1: rock04.atgi
  - Locator_2: vh_motorbike_sport
  - Chair_1: pf01347_chair.atgi
    - [resource] pf01347_chair.atg
  - Locator_5: pf01347_chair.atgi
  - Locator_3: hedge01.atgi
  - Locator_6: seymourplane_trian
- WarStreet.lvl
- WarStreet_Lighting.lvl

Palette

- Examples
  - SkyDome
  - CubeTest
  - SphereTest
  - ConeTest
  - CylinderTest
  - PlaneTest
  - BillboardTest
  - Orc
- Level Tools
  - GameObjectFolder
  - GameObjectGroup
  - Locator
- Lights
  - DirLight
  - BoxLight
  - PointLight
- Linears
  - ControlPoint
  - Curve

Property Editor

Display			
Display Properties	(none)		
Visibility	✓		
General			
Locked	☐		
Name	chair_1		
Transform			
Pivot	0	0	0
Rotation	-180	2.453	-180
Scale	1	1	1
Translation	9.184	0.032	1.436

Name
Unique name of Game Object

Bookmarks

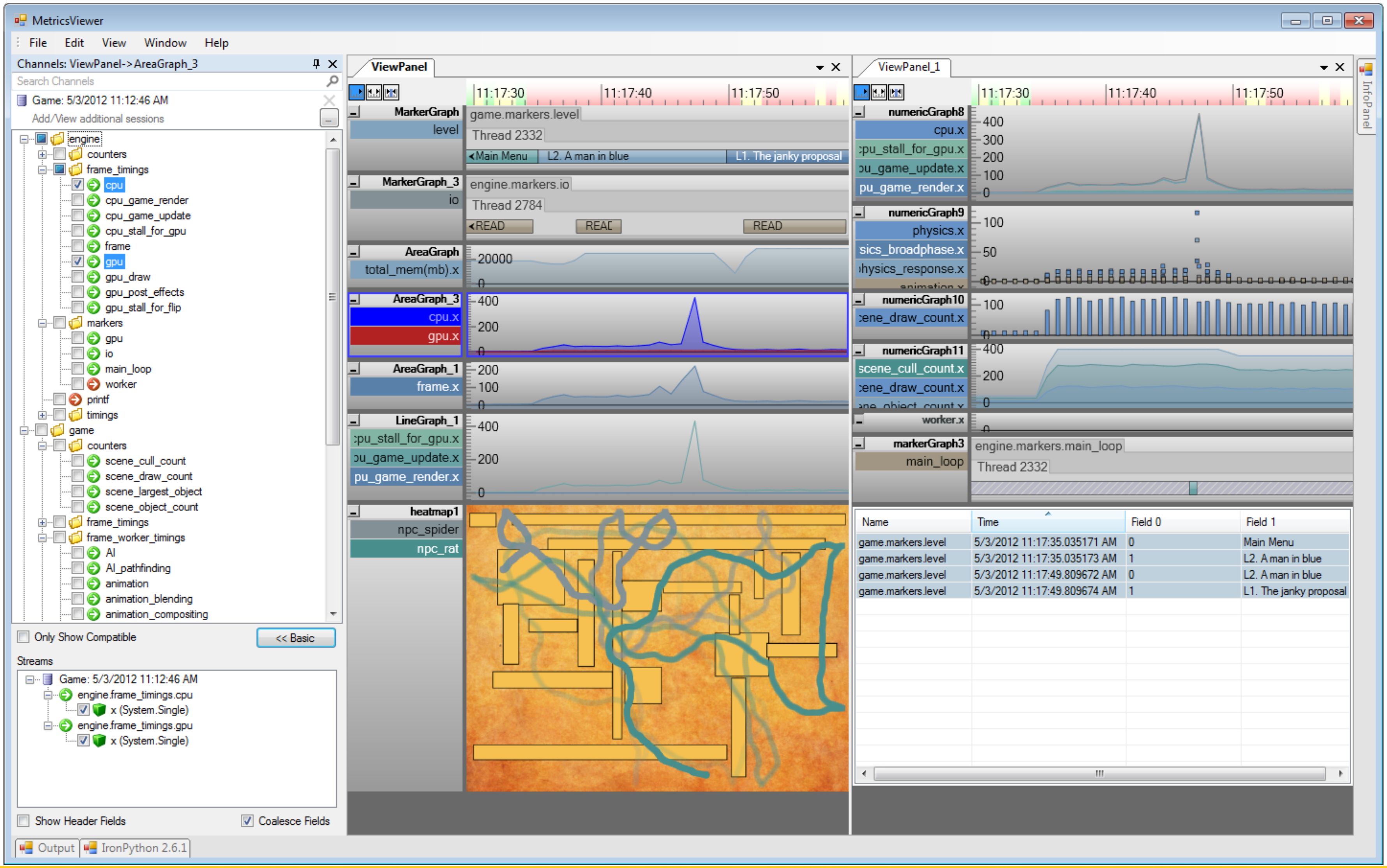
- Bik
- ScreenShot1

Resources

- Resources
  - Bend war street
  - Bikes
  - Buildings

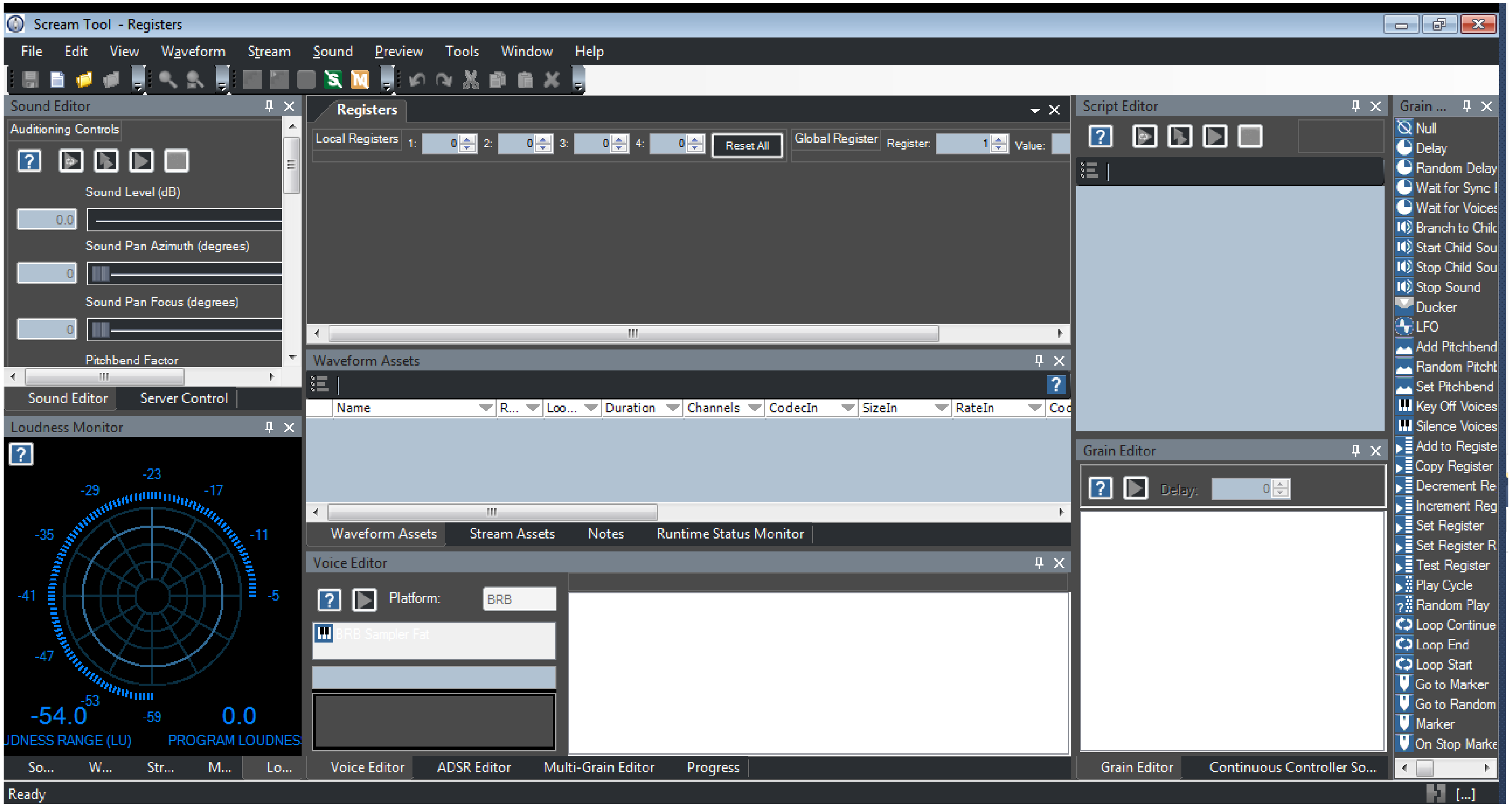
Name	Size	Type	Date ...

Metrics Viewer (by Game Tech Group)

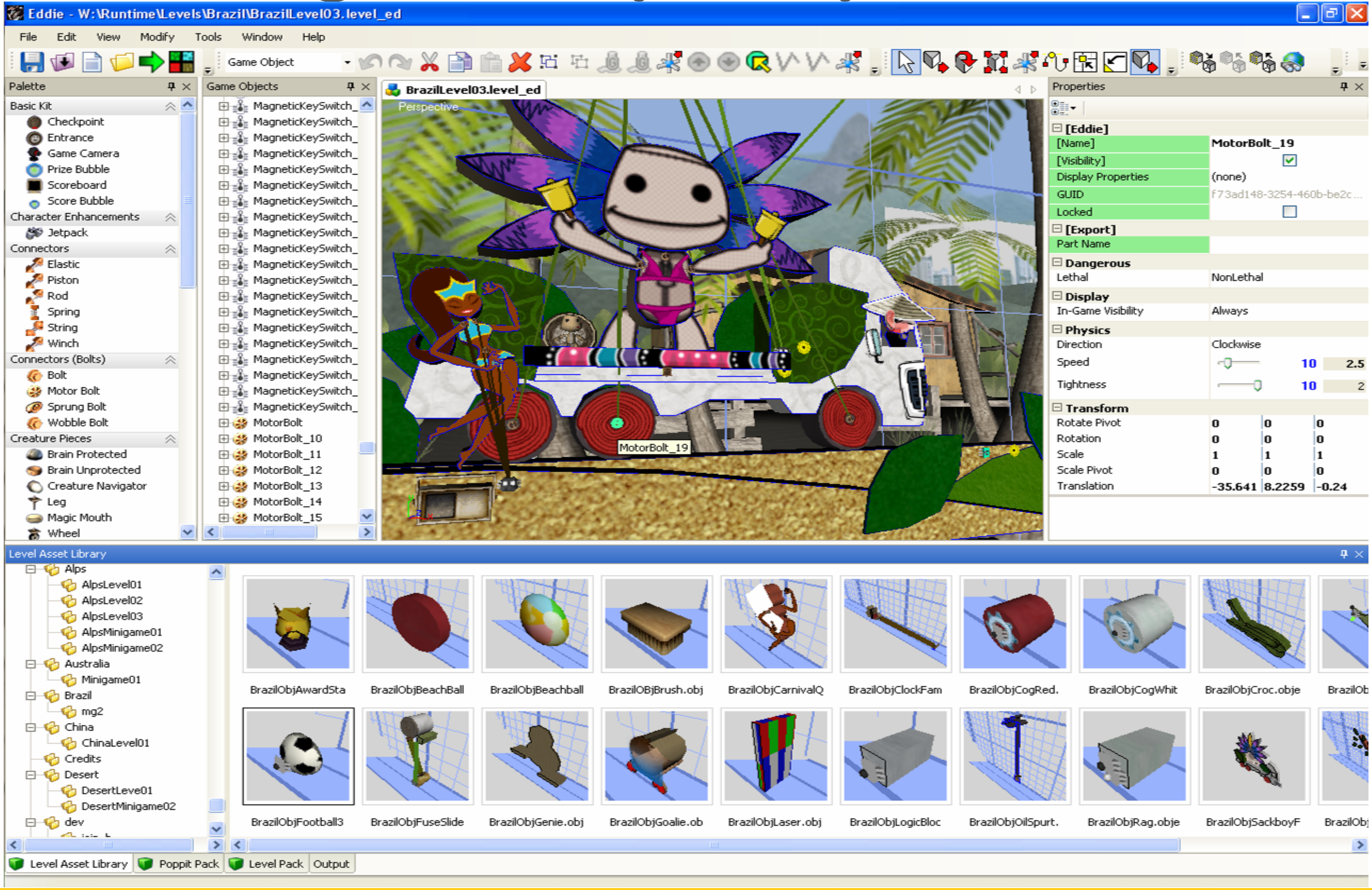


Name	Time	Field 0	Field 1
game.markers.level	5/3/2012 11:17:35.035171 AM	0	Main Menu
game.markers.level	5/3/2012 11:17:35.035173 AM	1	L2. A man in blue
game.markers.level	5/3/2012 11:17:49.809672 AM	0	L2. A man in blue
game.markers.level	5/3/2012 11:17:49.809674 AM	1	L1. The janky proposal

Scream Tool 7

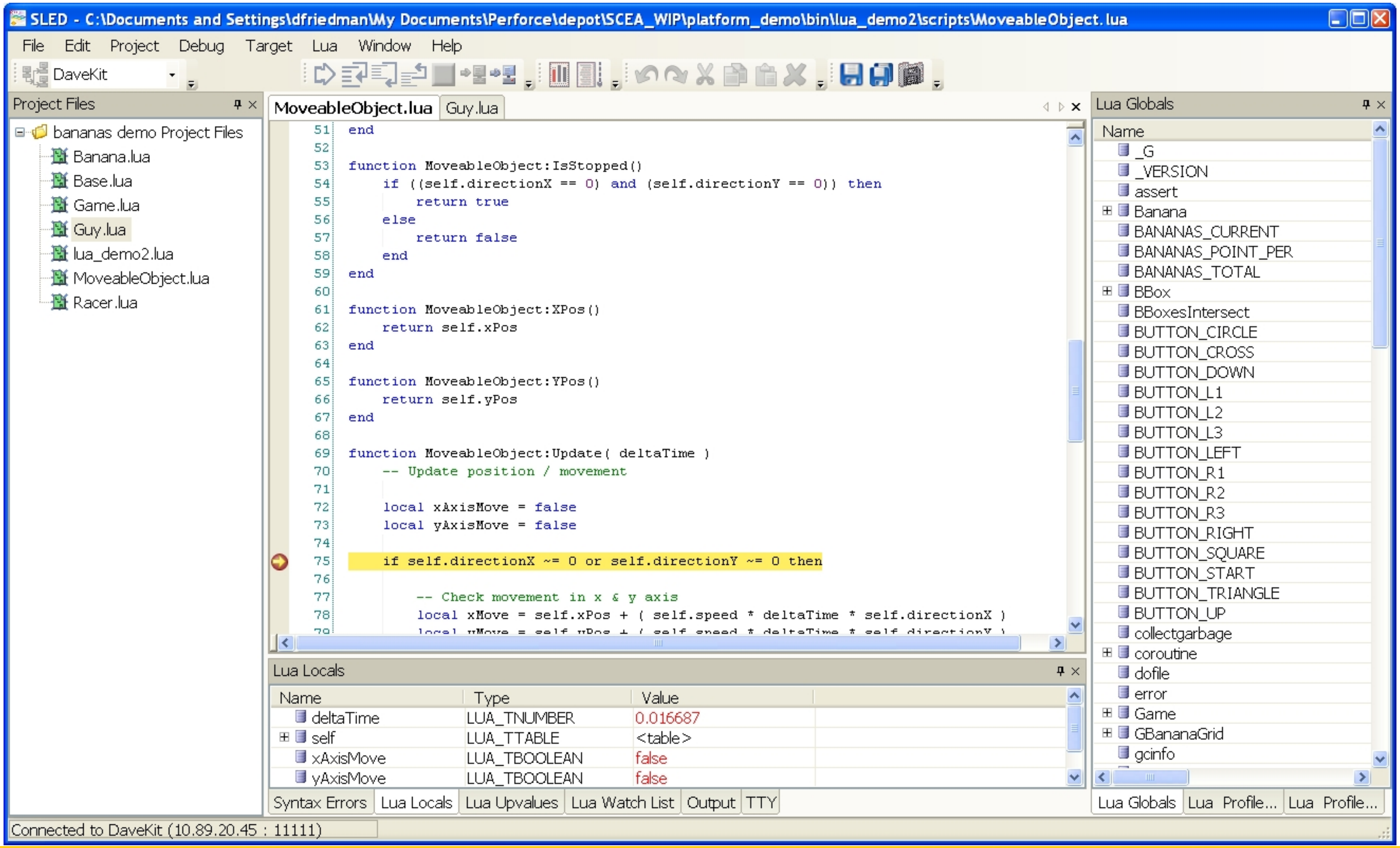


LittleBigPlanet (PSP®) Level Editor



[Eddie]			
[Name]	MotorBolt_19		
[Visibility]	✓		
Display Properties	(none)		
GUID	f73ad148-3254-460b-be2c...		
Locked	☐		
[Export]			
Part Name			
Dangerous			
Lethal	NonLethal		
Display			
In-Game Visibility	Always		
Physics			
Direction	Clockwise		
Speed	10	2.5	
Tightness	10	2	
Transform			
Rotate Pivot	0	0	0
Rotation	0	0	0
Scale	1	1	1
Scale Pivot	0	0	0
Translation	-35.641	8.2259	-0.24

SLED (by Game Tech Group)


SLED - C:\Documents and Settings\dfriedman\My Documents\Perforce\depot\SCEA_WIP\platform_demo\bin\lua_demo2\scripts\MoveableObject.lua

```lua
end

function MoveableObject:IsStopped()
    if ((self.directionX == 0) and (self.directionY == 0)) then
        return true
    else
        return false
    end
end

function MoveableObject:XPos()
    return self.xPos
end

function MoveableObject:YPos()
    return self.yPos
end

function MoveableObject:Update( deltaTime )
    -- Update position / movement

    local xAxisMove = false
    local yAxisMove = false

    if self.directionX ~= 0 or self.directionY ~= 0 then

        -- Check movement in x & y axis
        local xMove = self.xPos + ( self.speed * deltaTime * self.directionX )
        local yMove = self.yPos + ( self.speed * deltaTime * self.directionY )
```

Lua Locals

Name	Type	Value
deltaTime	LUA_TNUMBER	0.016687
⊞ self	LUA_TTABLE	<table>
xAxisMove	LUA_TBOOLEAN	false
yAxisMove	LUA_TBOOLEAN	false

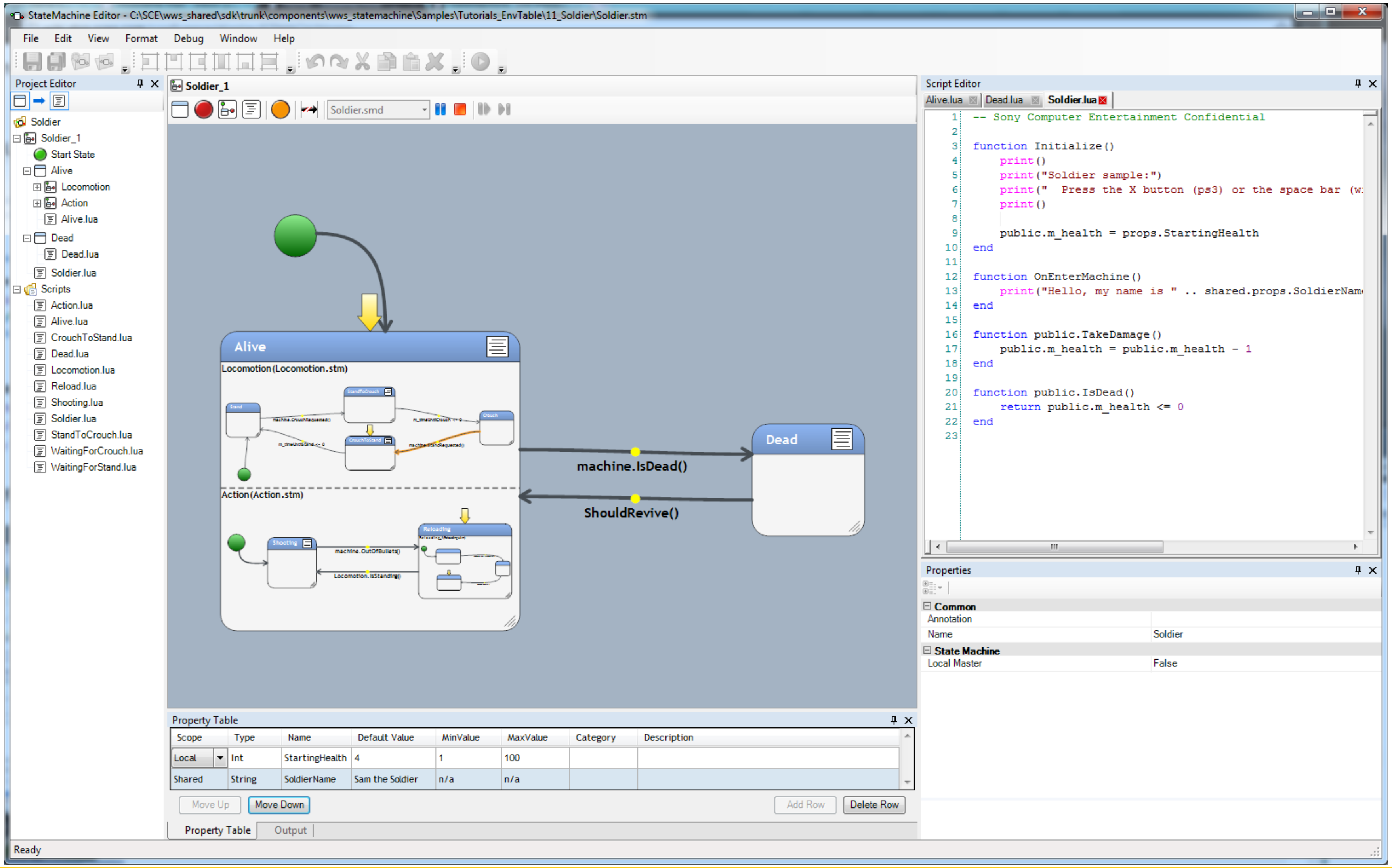
Syntax Errors | Lua Locals | Lua Upvalues | Lua Watch List | Output | TTY

Lua Globals

Name
_G
_VERSION
assert
⊞ Banana
BANANAS_CURRENT
BANANAS_POINT_PER
BANANAS_TOTAL
⊞ BBox
BBoxesIntersect
BUTTON_CIRCLE
BUTTON_CROSS
BUTTON_DOWN
BUTTON_L1
BUTTON_L2
BUTTON_L3
BUTTON_LEFT
BUTTON_R1
BUTTON_R2
BUTTON_R3
BUTTON_RIGHT
BUTTON_SQUARE
BUTTON_START
BUTTON_TRIANGLE
BUTTON_UP
collectgarbage
⊞ coroutine
dofile
error
⊞ Game
⊞ GBananaGrid
gcinfo

Connected to DaveKit (10.89.20.45 : 11111)

State Machine (by Game Tech Group)



Project Editor:

- Soldier
  - Soldier_1
    - Start State
    - Alive
      - Locomotion
      - Action
      - Alive.lua
    - Dead
      - Dead.lua
    - Soldier.lua
  - Scripts
    - Action.lua
    - Alive.lua
    - CrouchToStand.lua
    - Dead.lua
    - Locomotion.lua
    - Reload.lua
    - Shooting.lua
    - Soldier.lua
    - StandToCrouch.lua
    - WaitingForCrouch.lua
    - WaitingForStand.lua

Soldier.lua:

```
-- Sony Computer Entertainment Confidential

function Initialize()
    print()
    print("Soldier sample:")
    print("  Press the X button (ps3) or the space bar (w
    print()

    public.m_health = props.StartingHealth
end

function OnEnterMachine()
    print("Hello, my name is " .. shared.props.SoldierNam
end

function public.TakeDamage()
    public.m_health = public.m_health - 1
end

function public.IsDead()
    return public.m_health <= 0
end
```

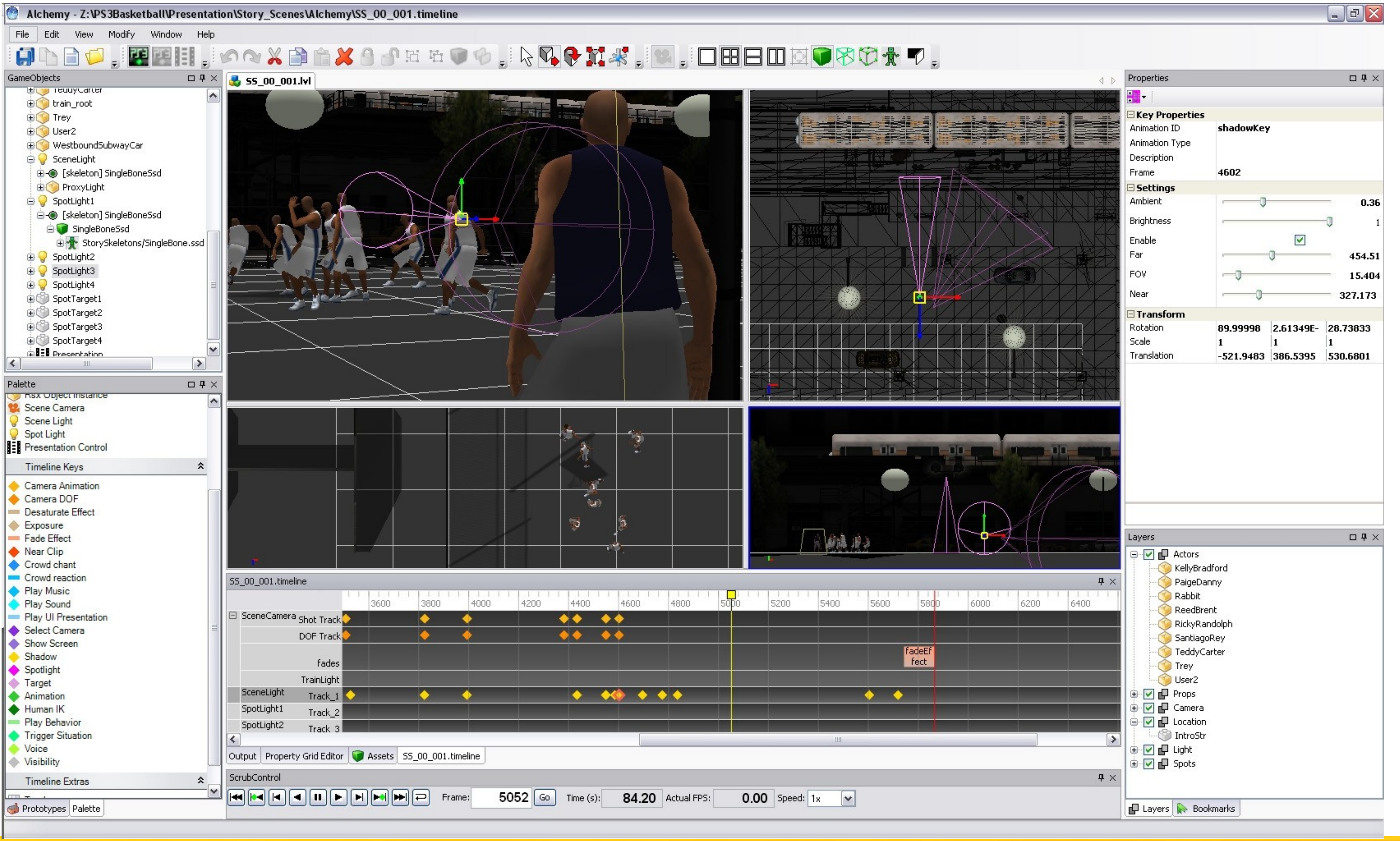
Properties:

Common	
Annotation	
Name	Soldier
State Machine	
Local Master	False

Property Table:

Scope	Type	Name	Default Value	MinValue	MaxValue	Category	Description
Local	Int	StartingHealth	4	1	100		
Shared	String	SoldierName	Sam the Soldier	n/a	n/a		

Alchemy



Key Properties

Animation ID	shadowKey		
Animation Type			
Description			
Frame	4602		

Settings

Ambient	0.36		
Brightness	1		
Enable	✓		
Far	454.51		
FOV	15.404		
Near	327.173		

Transform

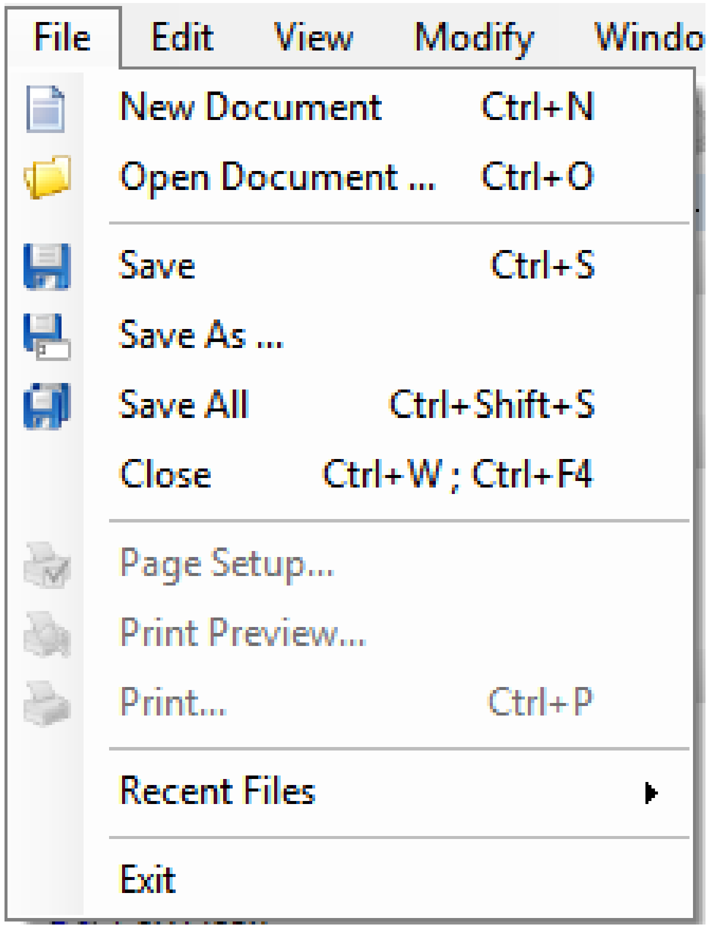
Rotation	89.99998	2.61349E-	28.73833
Scale	1	1	1
Translation	-521.9483	386.5395	530.6801

Frame: 5052 Time (s): 84.20 Actual FPS: 0.00 Speed: 1x

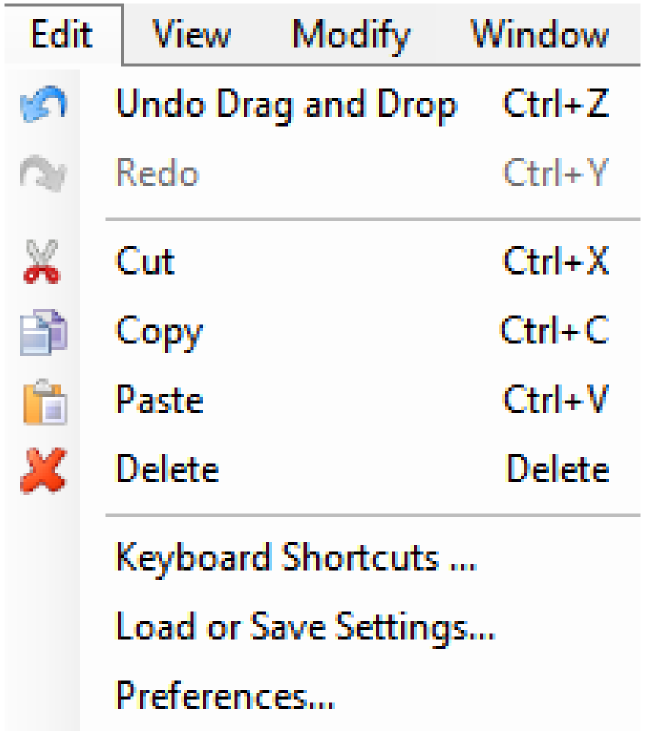
Main Components

- DOM (Document Object Model)
  - XML and Schema files can be used, but are optional
- Control Host Service with docking
  - WPF and WinForms
- Editor Infrastructure
  - Commands
  - Documents
  - Transactions
  - History
  - Contexts
  - Search & Replace
- Circuit, StateChart, Timeline with Direct2D
- Script editing with syntax highlighting
- Tree Control, Property Grid Editor, etc.
- Maya-like 3D Design View (OpenGL)
  - ATGI and Collada model support

We do the boring stuff…

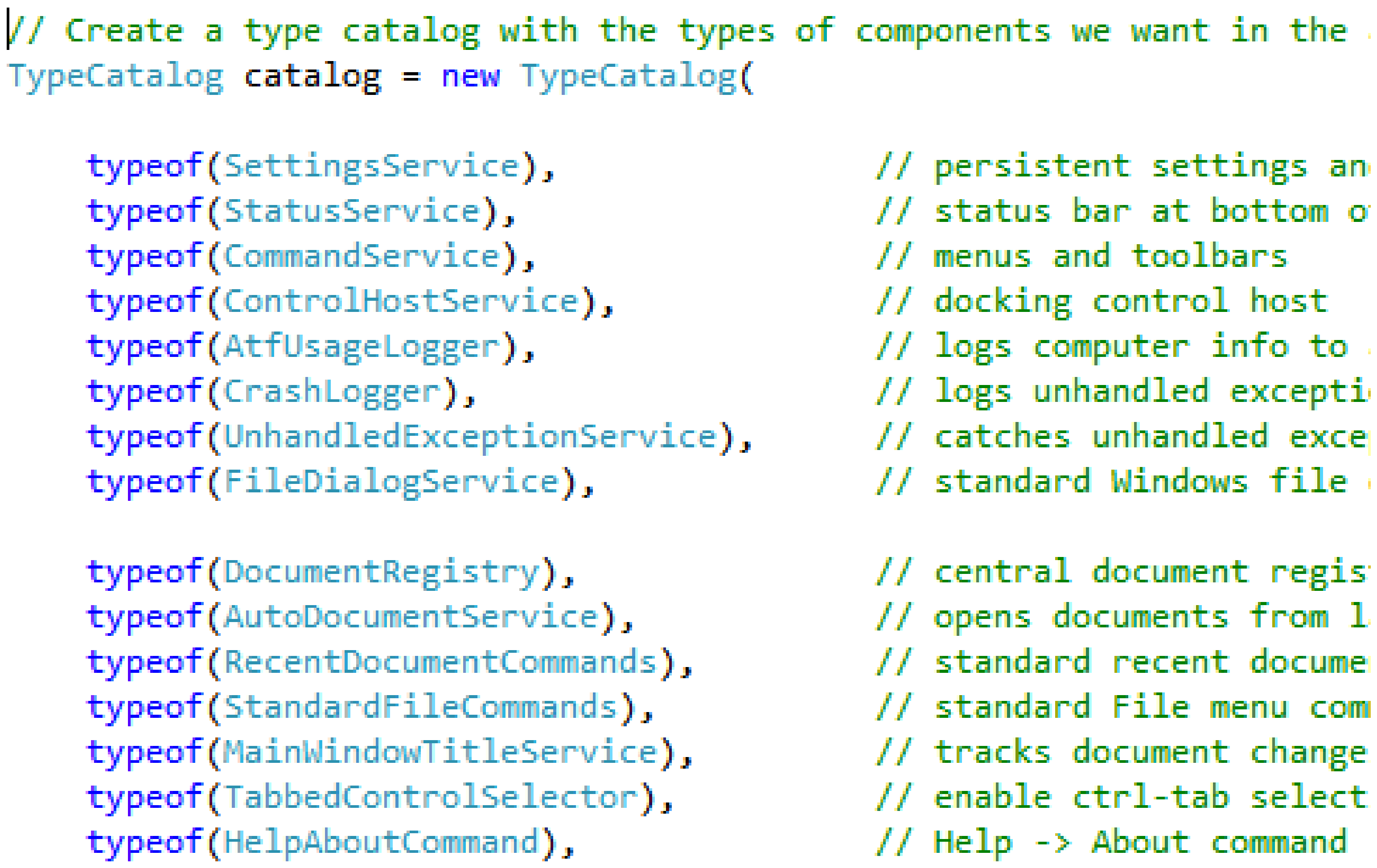


New Document	Ctrl+N
Open Document ...	Ctrl+O
Save	Ctrl+S
Save As ...	
Save All	Ctrl+Shift+S
Close	Ctrl+W ; Ctrl+F4
Page Setup...	
Print Preview...	
Print...	Ctrl+P
Recent Files	
Exit	



Undo Drag and Drop	Ctrl+Z
Redo	Ctrl+Y
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Delete	Delete
Keyboard Shortcuts ...	
Load or Save Settings...	
Preferences...	

You pick the parts you want…

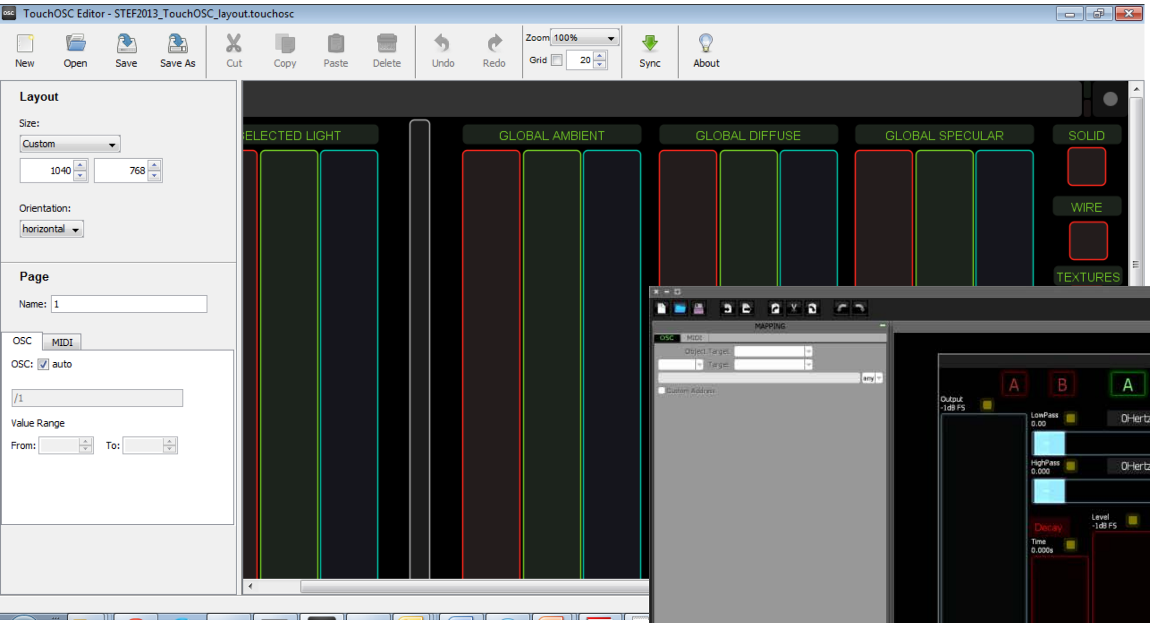


```
// Create a type catalog with the types of components we want in the
TypeCatalog catalog = new TypeCatalog(

    typeof(SettingsService),                    // persistent settings an
    typeof(StatusService),                      // status bar at bottom o
    typeof(CommandService),                     // menus and toolbars
    typeof(ControlHostService),                 // docking control host
    typeof(AtfUsageLogger),                     // logs computer info to
    typeof(CrashLogger),                        // logs unhandled excepti
    typeof(UnhandledExceptionService),          // catches unhandled exce
    typeof(FileDialogService),                  // standard Windows file

    typeof(DocumentRegistry),                   // central document regis
    typeof(AutoDocumentService),                // opens documents from l
    typeof(RecentDocumentCommands),             // standard recent docume
    typeof(StandardFileCommands),               // standard File menu com
    typeof(MainWindowTitleService),             // tracks document change
    typeof(TabbedControlSelector),              // enable ctrl-tab select
    typeof(HelpAboutCommand),                   // Help -> About command
```

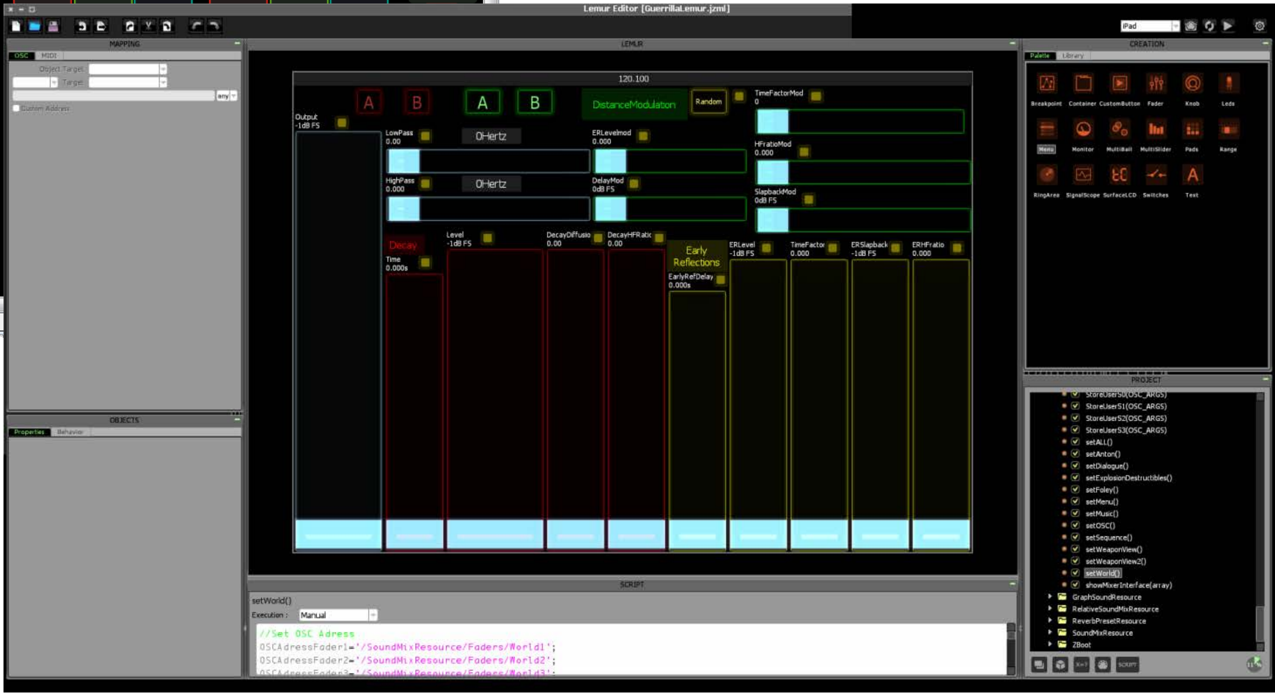
Open Sound Control

- Collaboration with Guerrilla Games
- Open standard, high-level network protocol
- Successor to MIDI
- Sends and receives name / value pairs
- Lemur ($50) for iPad works great
- TouchOSC for Android tablets works well, too
- Non-programmer can create GUI and tools

Open Sound Control



TouchOSC for Android



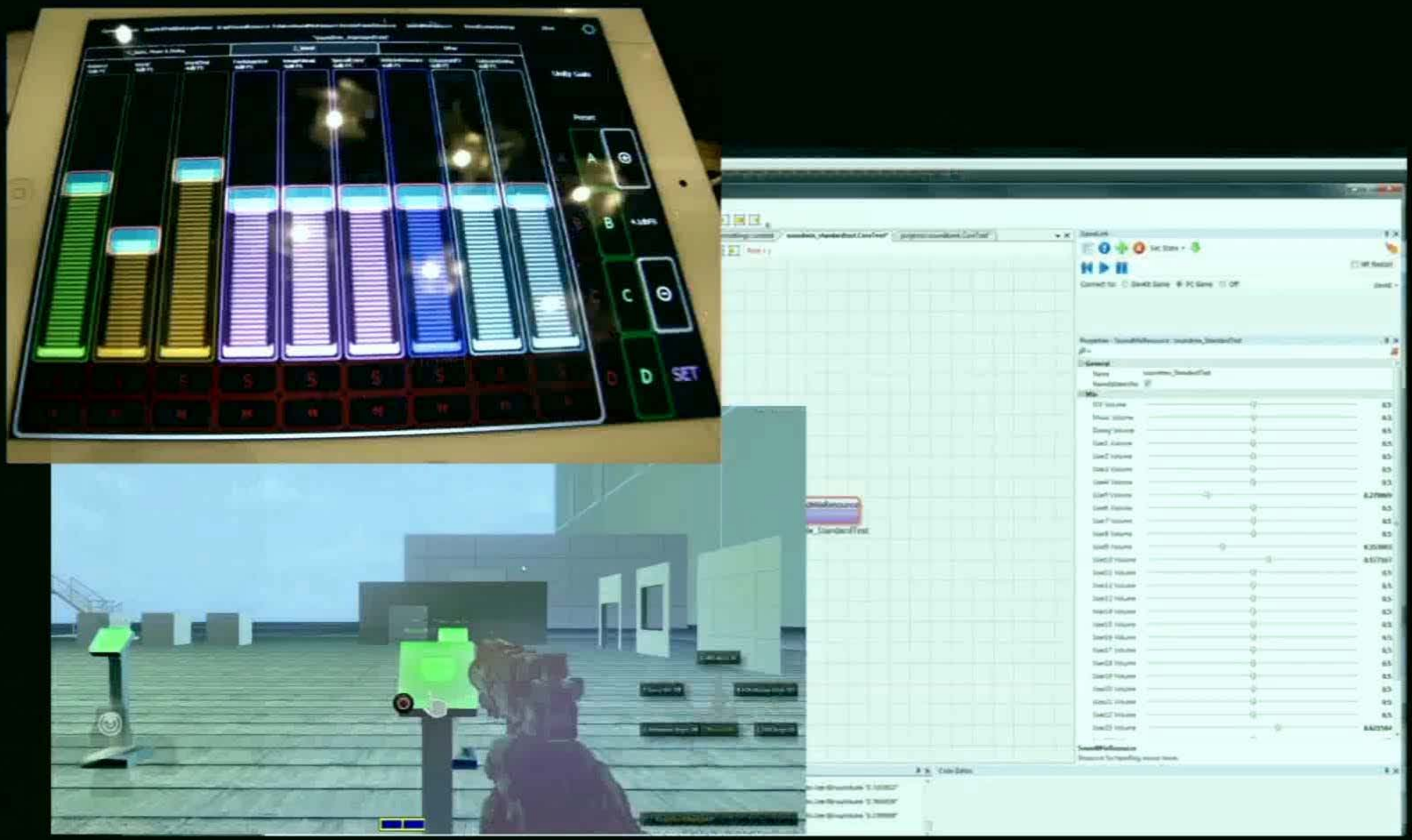
setWorld()

Execution : Manual

```
//Set OSC Adress
OSCAdressFader1='/SoundMixResource/Faders/World1';
OSCAdressFader2='/SoundMixResource/Faders/World2';
OSCAdressFader3='/SoundMixResource/Faders/World3';
```

Lemur for iOS
+ scripting
+ more controls
+ editing on iPad

Open Sound Control at Guerrilla Games

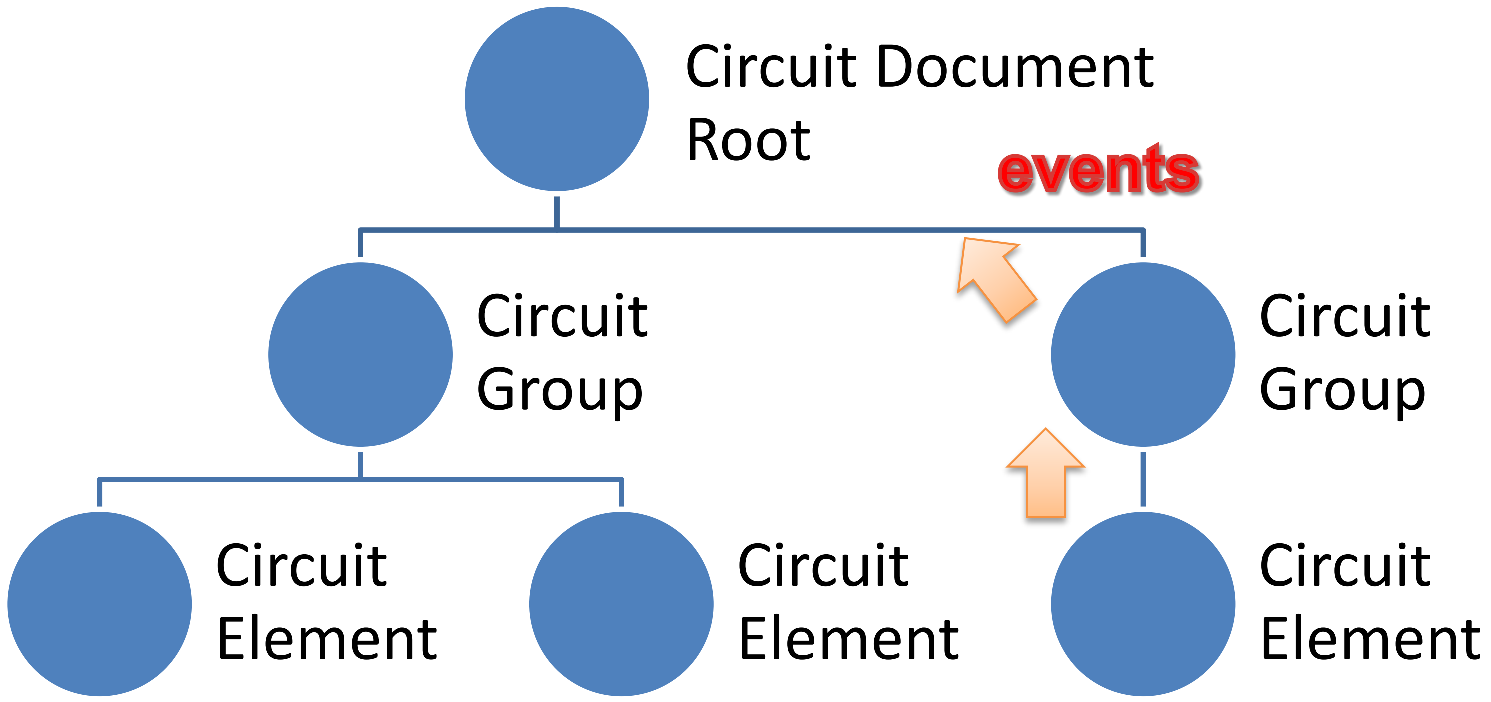


DOM (Document Object Model)

- In-memory observable XML-like database
- DomNode trees
  - Root of a DomNode tree is typically a document
- DomNodes have attributes and children
  - Specified by a DomNodeType (like a schema type)
  - Attributes, like in XML, are simple types (int, float, string, reference) or arrays of simple types.
- DomNodes are observable
  - Child Added event
  - Child Removed event
  - Attribute Changed event

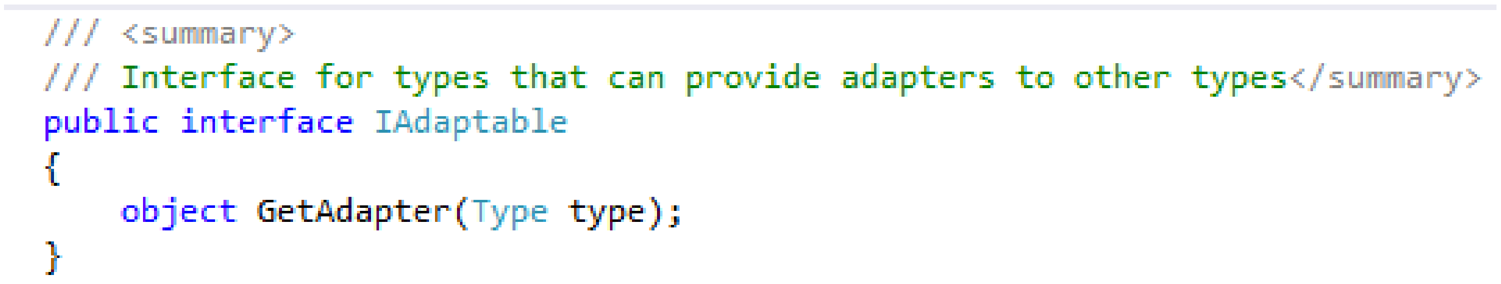
DomNode Hierarchy

- Each DomNode has certain attributes and children, specified by the DomNode's DomNodeType
- DomNodeTypes can be created programmatically or by loading schema file
- Events propagate from children to parents



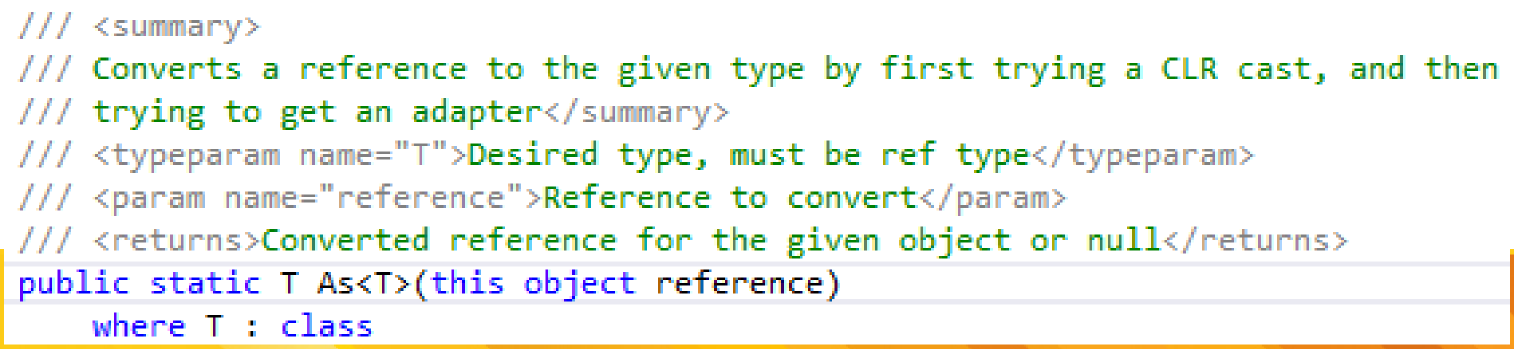
Adaptability

- IAdaptable
  - Implemented by DomNode and DomNodeAdapter



```
/// <summary>
/// Interface for types that can provide adapters to other types</summary>
public interface IAdaptable
{
    object GetAdapter(Type type);
}
```

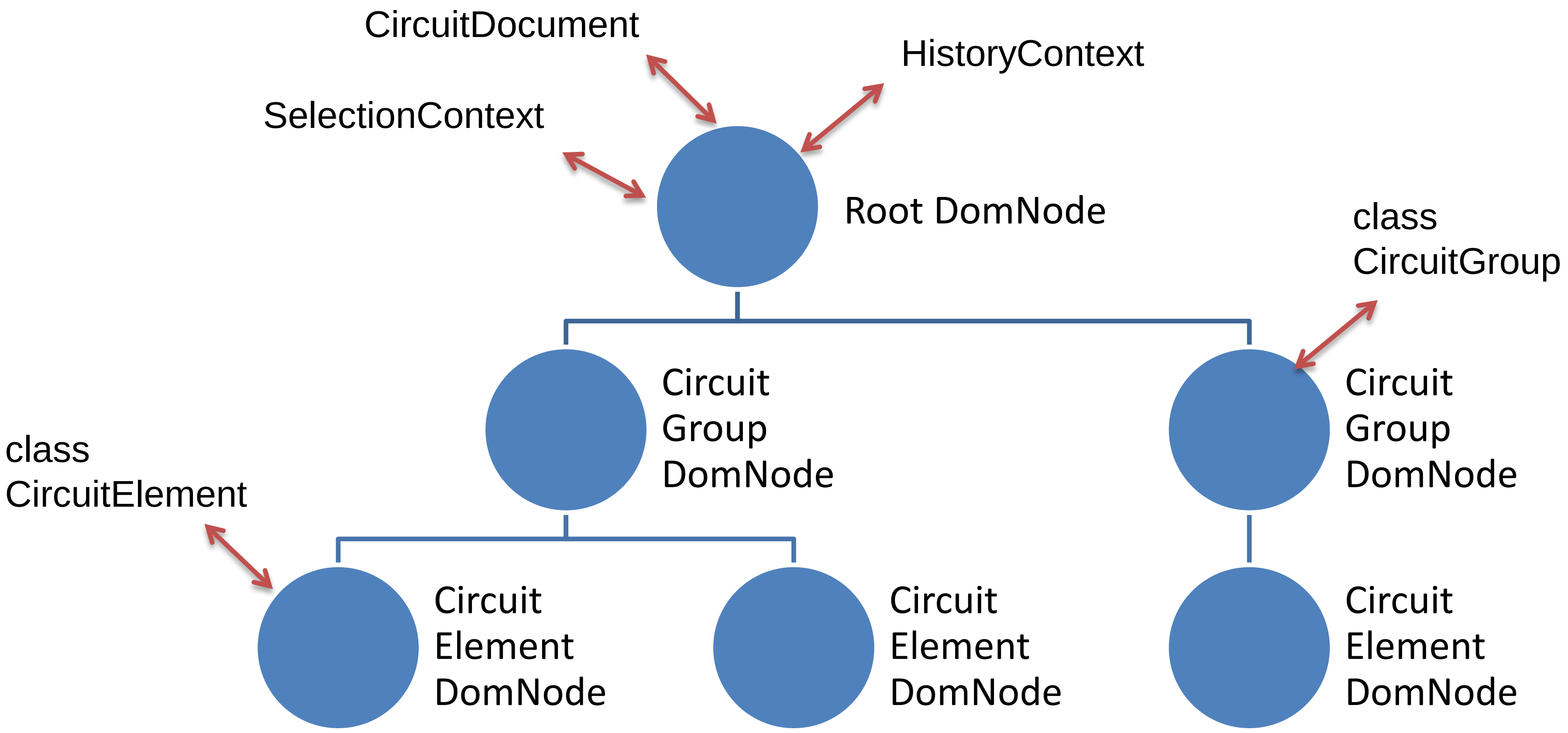
- As<T> extension method on object
  - First does C# ‘as’, then checks for IAdaptable



```
/// <summary>
/// Converts a reference to the given type by first trying a CLR cast, and then
/// trying to get an adapter</summary>
/// <typeparam name="T">Desired type, must be ref type</typeparam>
/// <param name="reference">Reference to convert</param>
/// <returns>Converted reference for the given object or null</returns>
public static T As<T>(this object reference)
    where T : class
```

DomNodeAdapters

Clients' "business classes" derive from DomNodeAdapter and are defined for particular DomNodeTypes. A DomNode is created first and then its DomNodeAdapters are created automatically but are initialized on demand. Call InitializeExtensions on root DomNode to initialize all DomNodeAdapters for the whole tree.



Contexts

Typically one of each per document

- SelectionContext
  - Tracks user’s selection and has change events
- HistoryContext
  - Tracks DOM changes to sub-tree for undo/redo
- TransactionContext
  - Base class of HistoryContext. Tracks when a set of changes begins and finishes, so that validation logic can be executed at correct time.
- InstancingContext
  - Implements copy, paste, and delete

Registries

One of each of per app

- DocumentRegistry – tracks documents
  - List of open documents
  - Adds and removes documents
  - Active document
- ContextRegistry – tracks contexts
  - List of available “contexts”
  - Adds and removes contexts
  - Active context
- IControlRegistry, IControlHostService
  - Clients register Controls, so that they appear in docking framework. Active Control is tracked.

Services

One of each per app. Provides functionality to other components.

- ControlHostService
  - Docking framework
- CommandService
  - Menus and toolbars
- SettingsService
  - User and app settings GUI and persistence
- PerforceService
- SkinningService
- *Etc.*

Editors

One of each per app; work with active context

- PropertyEditor
  - 2-column property editor with names and values
- GridPropertyEditor
  - Spreadsheet-style multi-object property editor
- TimelineEditor
- CircuitEditor
- CurveEditor
- *Etc.*

ATF Pros and Cons

- Pros
  - Easy to create editing tools with all of the standard features -- copy & paste, undo & redo, windows docking, user settings, document persistence (if using XML files), etc.
  - Powerful components for specific tasks
    - Circuit editing
    - Timeline editing
    - Property editing
    - Direct2D wrappers

ATF Pros and Cons

- Cons
  - Connections between components are usually abstract and use C# interfaces and Adaptability. It can be difficult to know which components are working with each other. Tip: use debugger.
  - Steep learning curve. We’ve tried to address this with well-written and thorough docs.
  - The DOM is difficult to debug. Use DomNodeAdapters, DOM Recorder, and DOM Explorer.

Tip

- “Features are an asset. Code is a liability.” – Bill Budge

Tip

- Creating shared code is 2x to 3x slower.
  - Avoiding breaking changes
  - Difficult to know how clients are using your code

Tip

- Clients want to customize everything!
  - Expect to need to make class members public or protected.
  - If you're unsure, keep it private and then make it public upon request.

Tip

- Code Reviews?
  - Always: for new C# interfaces
  - Always: for significant new features
  - “It Depends”: for more minor changes

Tip

- Have written coding standards
  - For C#, see “Framework Design Guidelines” on MSDN

Tip

- Build “orthogonally”.
  - Try to have minimal well-defined dependencies on other classes.
  - Program against interfaces instead of concrete classes where possible.

Tip

- Leave yourself a backdoor with the ‘info’ object

```
public interface IDocumentClient
{
    DocumentClientInfo Info
    {
        get;
    }
…
```

Tip

- Prefer IEnumerable<T> over IList<T> in APIs
- Never use List<T>

Tip

- When developing a large new piece of tech, try to find a client to work with.
  - This validates your approach.
  - When finished, you’ll have at least one client.

Tip

- Write the release note for a breaking change, before making the breaking change.
  - What is this breaking change?
  - Why is this breaking change necessary?
  - How do clients fix their code?

Tip

- Make C# interfaces be as small as possible.
  - If it has > 6 completely different kinds of members, that’s a code smell
  - Use extension methods to provide utility methods.

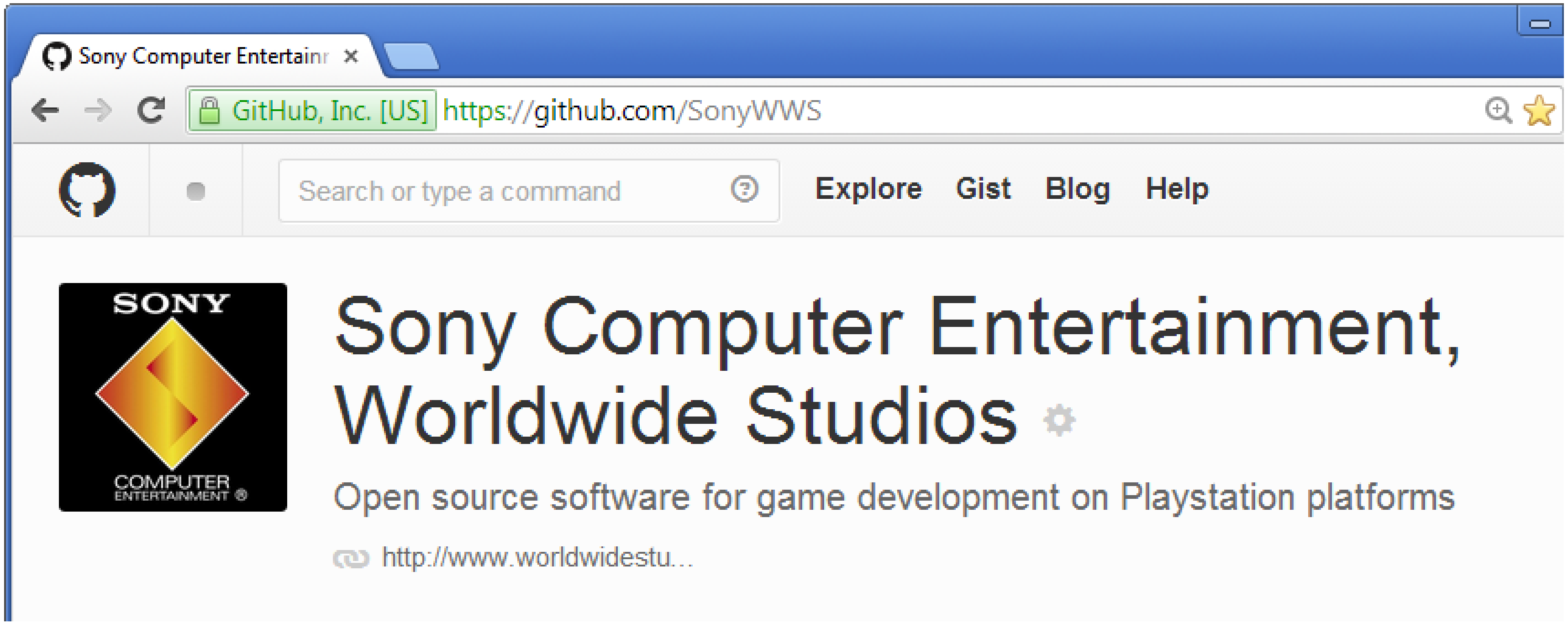
Tip

- Visit clients once or twice a year for a “road show”.
  - Show off your latest work.
  - See what they’re up to.
  - Get ideas for future projects.
  - Spread knowledge between clients.

Resources

- Full Featured Examples
  - Circuit Editor
  - Statechart Editor
  - Timeline Editor
  - Using Direct2D
  - Model Viewer
  - …
- Massive wiki documentation
- Issue tracker
- Responsive staff ☺

github.com/SonyWWS


Sony Computer Entertainm ×

GitHub, Inc. [US] https://github.com/SonyWWS

Search or type a command

Explore Gist Blog Help

Sony Computer Entertainment, Worldwide Studios

Open source software for game development on Playstation platforms

http://www.worldwidestu...

Questions? Thank you!