Cubes All The Way Down

Renaud Bédard
Programmer, Polytron

GAME DEVELOPERS CONFERENCE
SAN FRANCISCO, CA
MARCH 5-9, 2012
EXPO DATES: MARCH 7-9
2012

About Me

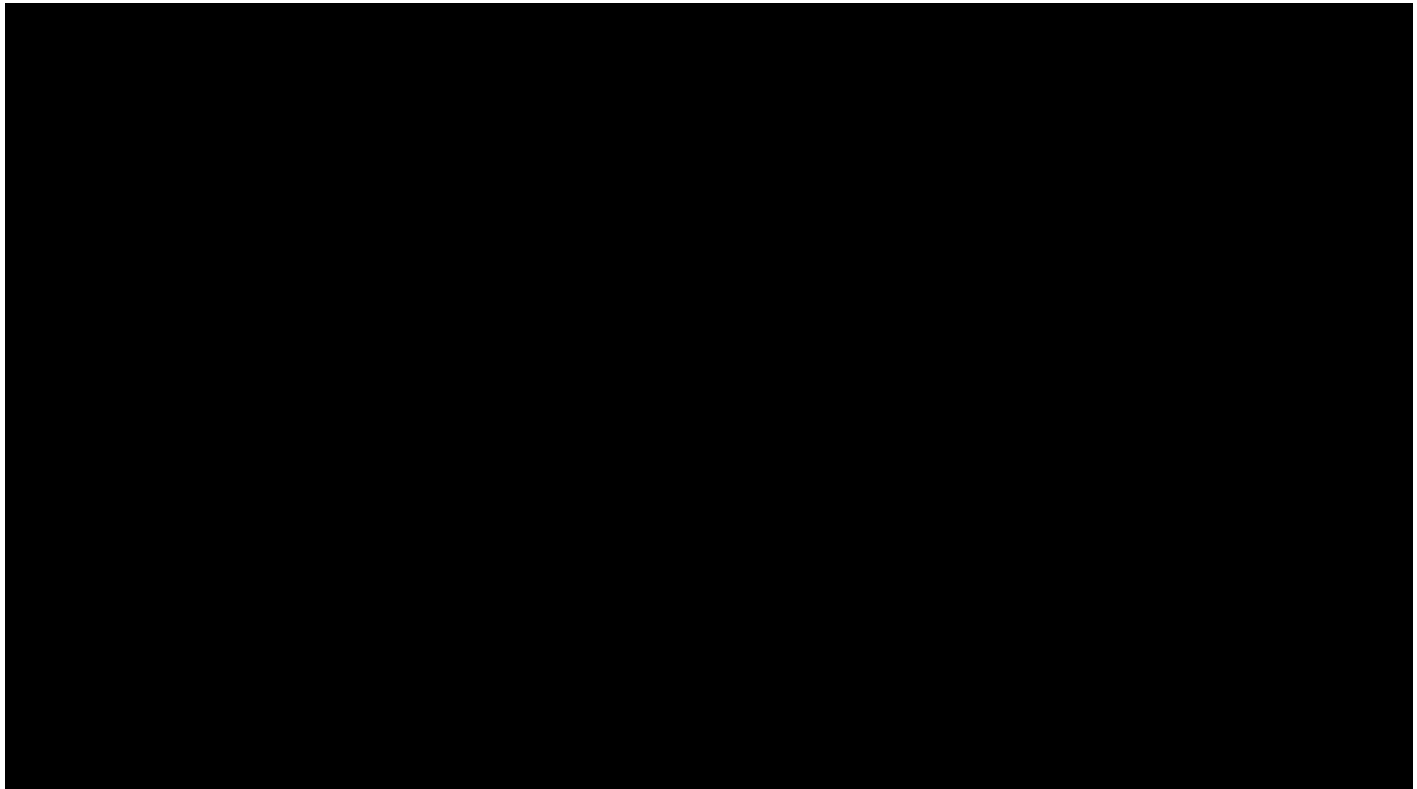
- Started 3D programming in VB6 & Truevision3D ~ 2001
- Started working with Phil Fish on FEZ in April 2007
- FEZ gets 2 noms and 1 win (Visual Arts) at IGF'08
- Bacc. in Computer Science at UQÀM in late 2008
- Worked full-time since then at Polytron
  - FEZ is first commercial title and full-time “industry” “job”

About FEZ

- 2D/3D Exploration Puzzle Platformer (“mystroidvania”)
- Pixel art where the pixels are 3D *trixels*
- Platforming in 2D, but across all 4 orthographic views

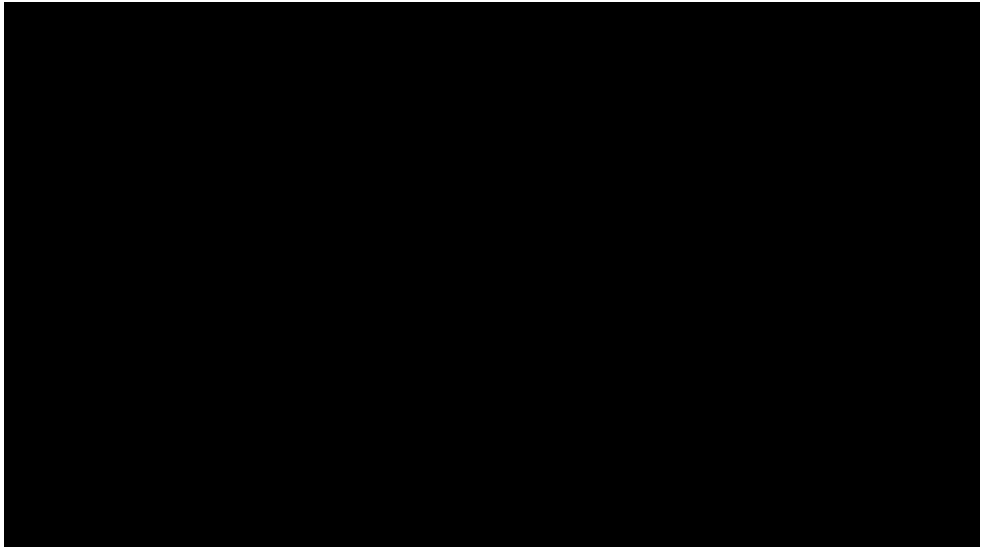
- Built in XNA/C# from the start
  - Spanned 5 XNA versions, from 1.0 to 3.1!

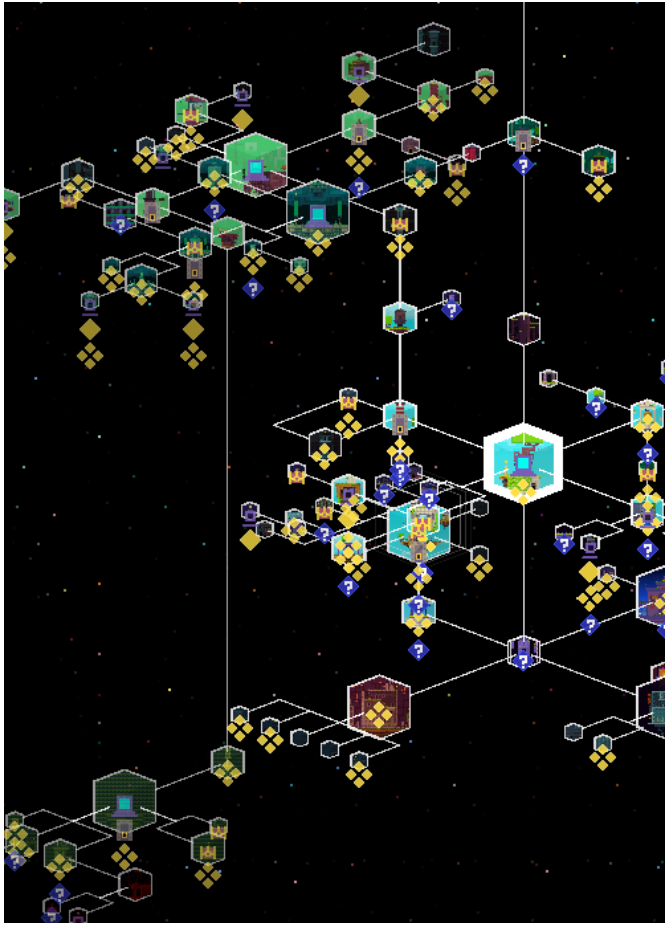
Game Footage



World Structure

- Big branching mess, 157 areas total
- Each is an isolated level
  - But level transitions are made “seamless”



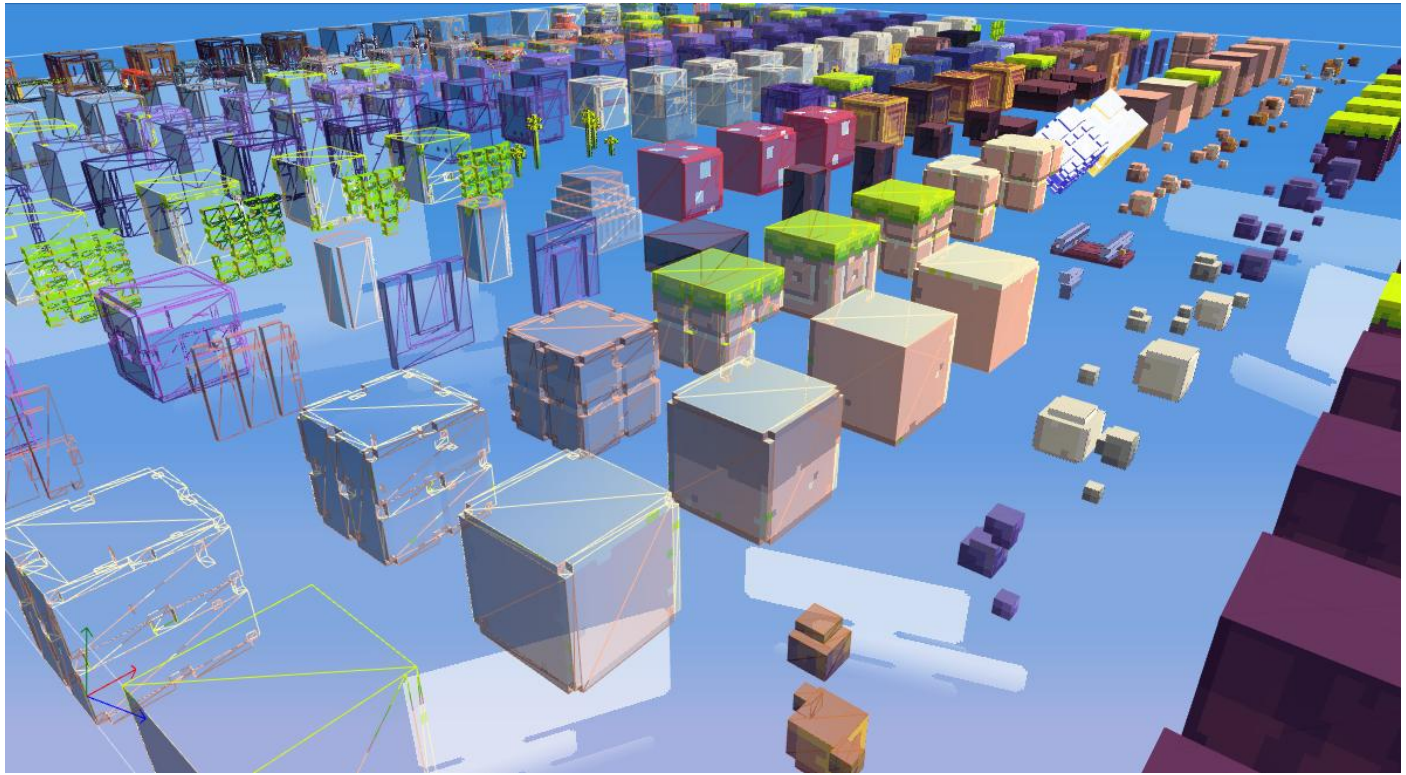


How are levels "cut up"?

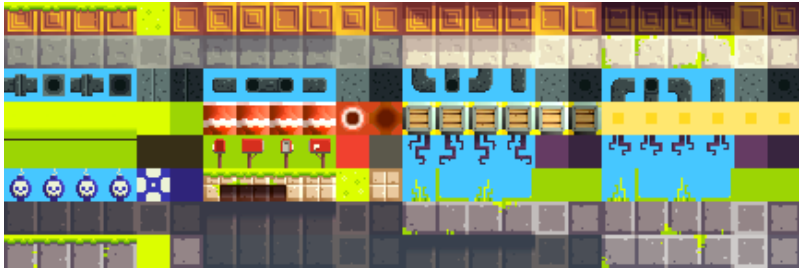
- Art- and level-production wise, tiles are easier
- Plain old 2D tilemap doesn't work : we need 3D tiles
  - …triles?

- 16x16x16 voxels (*trixels*) chosen as an arbitrary fixed size

"Nature" Tile Set (302 tiles)



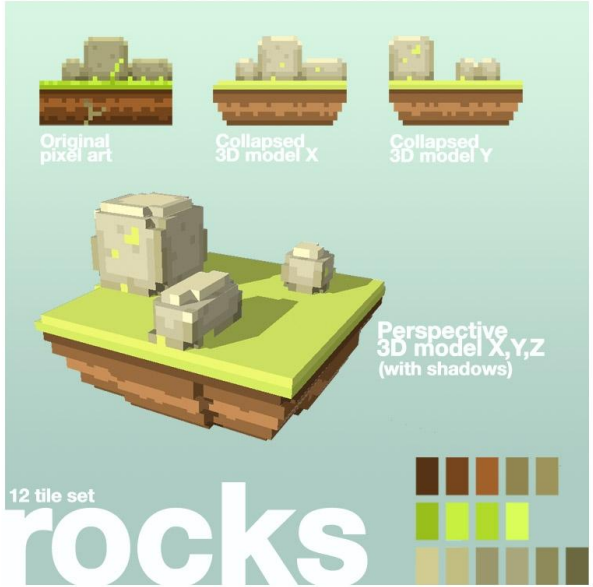
Texturing Triles



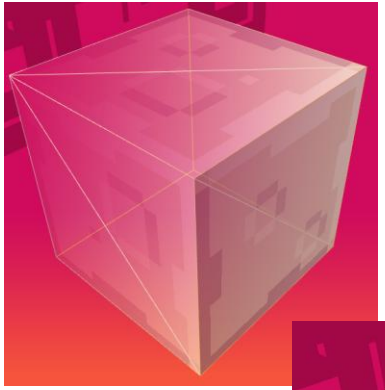
- Per-trixel coloring?
  - Implies good integrated painting tools
- Cubemaps! (*freebloods*) F R B L U D
  - Triles need to be convex
  - Cubemaps are 16x16x6 32bpp = 6kb uncompressed!

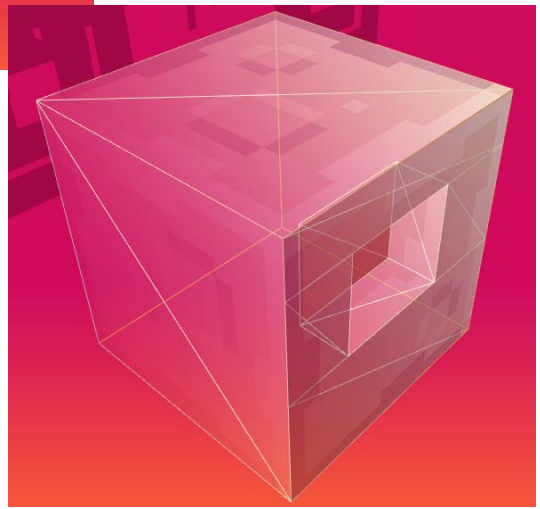
Modelling Trilas

- First trile concepts made in Google Sketchup (2007)



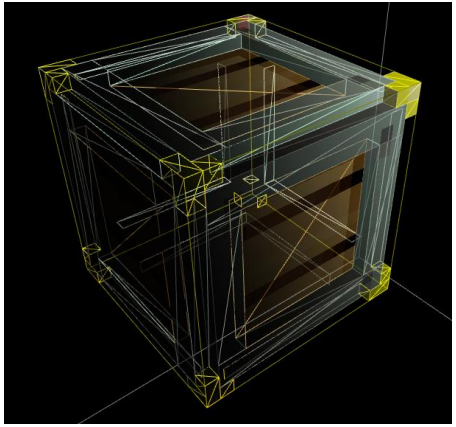
Trile Sculpting

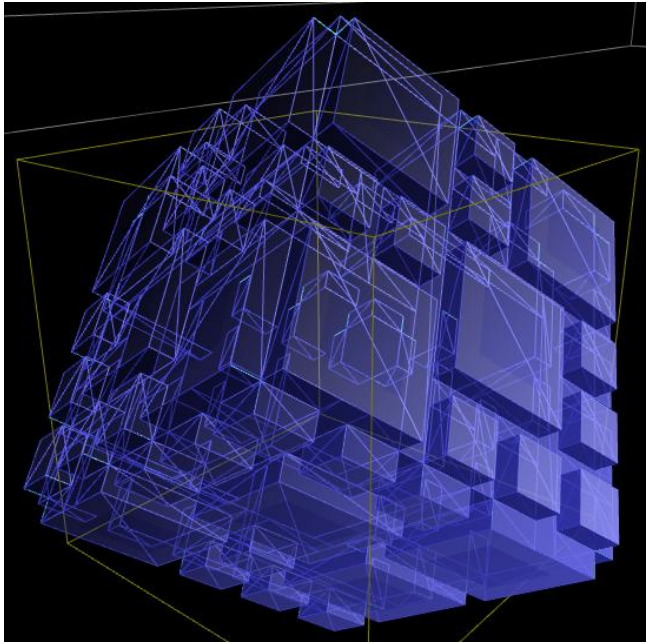




Blocky But Detailed

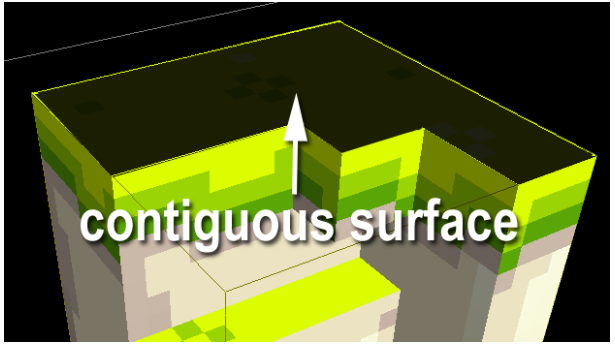
- Average of 136 polygons per trile
- Typical scene : 50,000 polygons
  - But anywhere from ~5000 to ~250,000

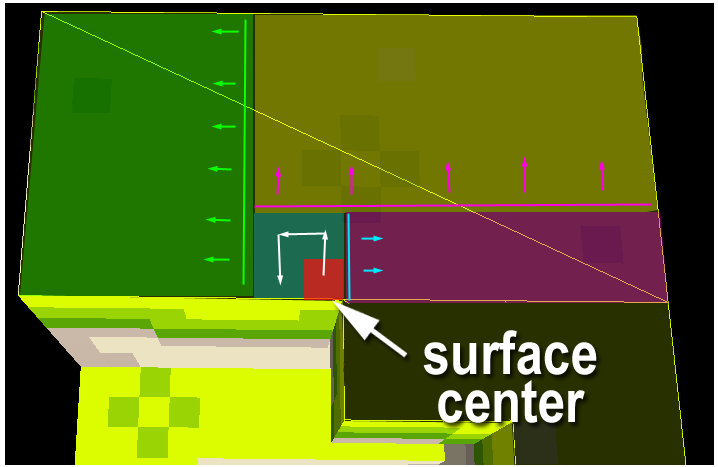




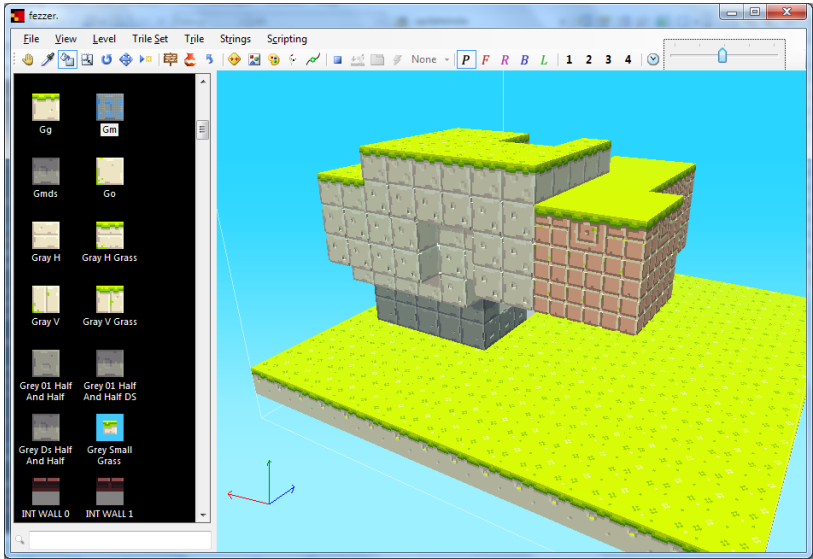
Mesh Simplification

- Extrapolate contiguous surfaces from trixels
- For each surface,
  - Enumerate biggest rectangles in that surface
  - Each rectangle becomes a plane

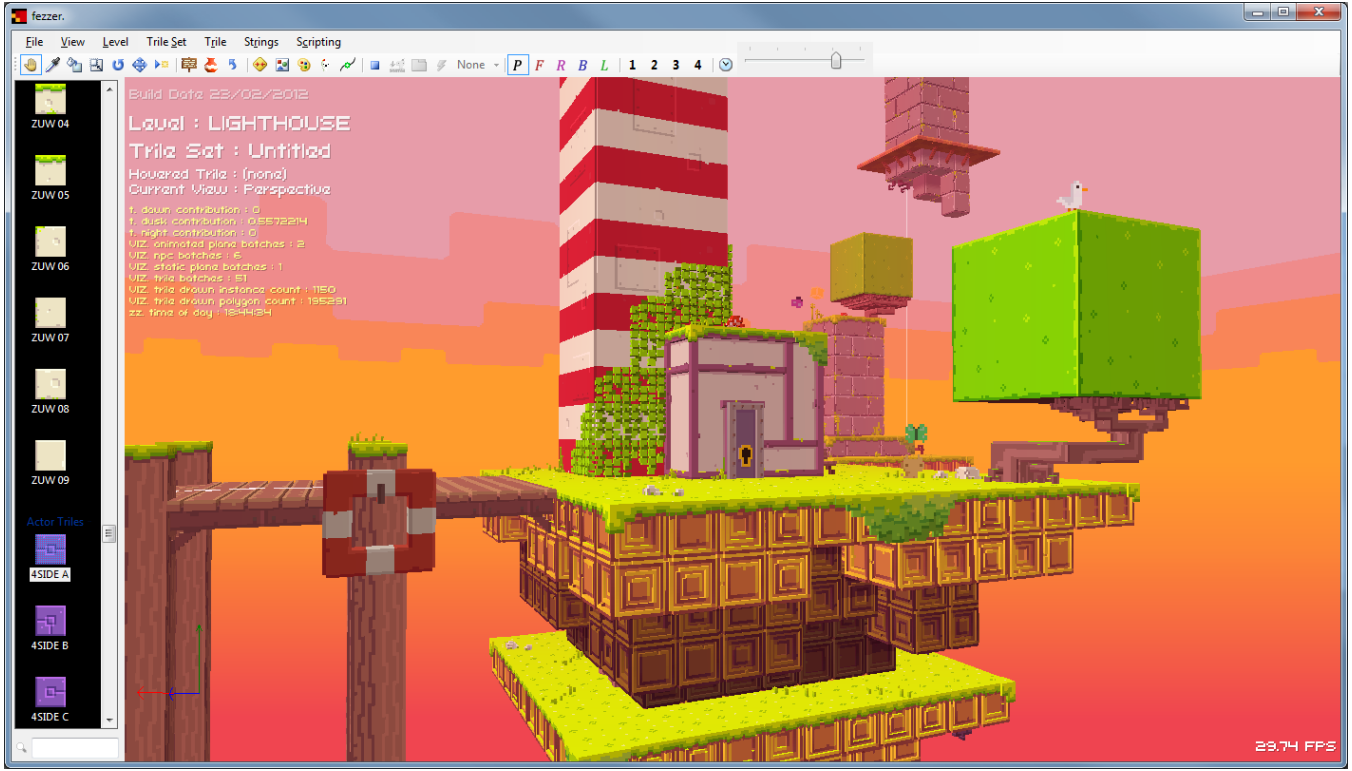




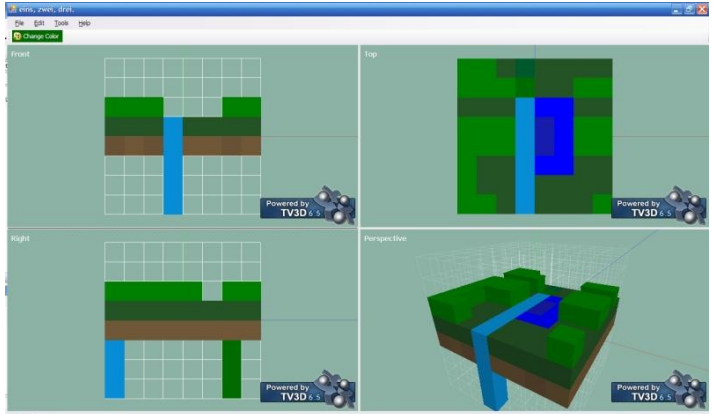
Building Levels in "Fezzer"

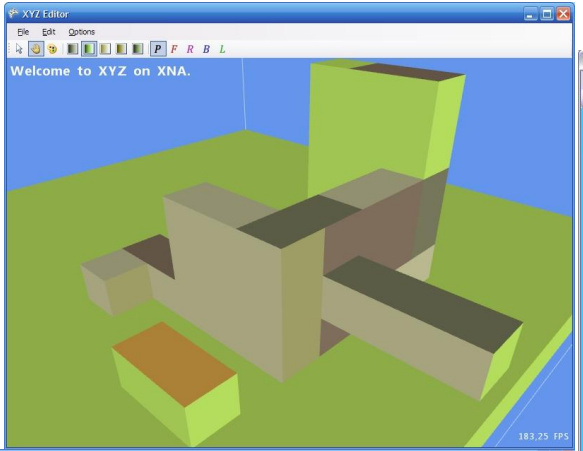


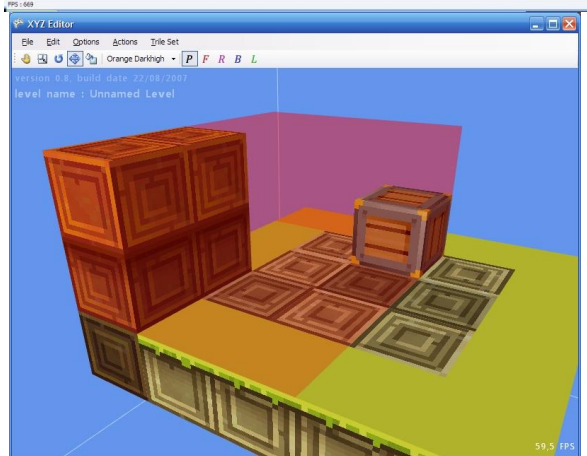
Completed level in final Fezzer

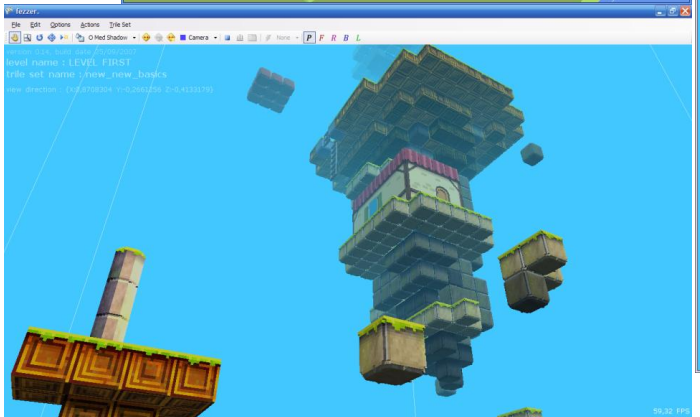


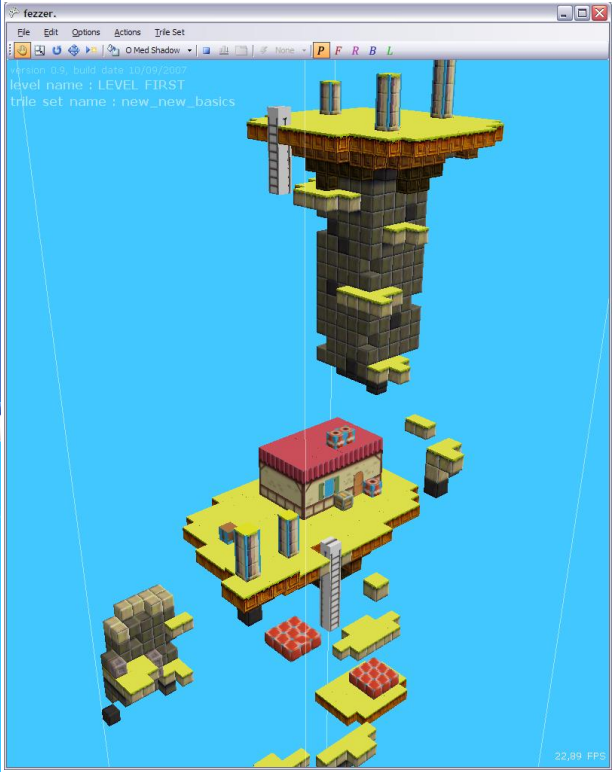
Building Fezzer...











Feature tracking

- Google Docs all the way!
- Sprints planned 1 week to 1 month ahead
  - Do basic feature first, revisit & polish later

	A	B	C	D	E
1	DESCRIPTION	PROGRESS	START	END	NOTES
2	Scriptable moving platforms	done	5/21/2009	5/27/2009	A group that moves along a path with speed/acceleration on a script e
3	Vines actor support	done	4/29/2009	4/30/2009	
4	Wrap around corners movement for vines	done	4/30/2009	5/1/2009	
5	Improved selection tool in editor	done	4/28/2009	4/29/2009	
6	One-click trile rotation randomizer	done	4/27/2009	4/28/2009	
7	Store surfaces in art object intermediate files	done	4/24/2009	4/25/2009	For faster loading in editor & compilation
8	Variable trile collision size (width + depth)	done	5/4/2009	5/11/2009	Including dynamic entities like crates (pickup/throw support, stacking)
9	Variable trile collision size (height)	done	5/11/2009	5/16/2009	Including dynamic entities too
10	Moving triles by the trixel in editor	done	5/12/2009	5/14/2009	
11	Remove/optimize LINQ queries for pickup objects	done	5/8/2009	5/8/2009	Only remaining garbage is caused by dictionary key/value iterators
12	Save/load trile groups in the editor	done	5/19/2009	5/20/2009	.grp.sdl files, act like selected triles but grouped (manipulated with the
13	Focus on selection on keypress in editor	done	5/6/2009	5/7/2009	Like the trile isolation mode, but just plays with camera (press Enter i
14	Store surfaces in trile set intermediate files	done			For faster loading in editor & compilation
15	Cleanup of stale code from last summer	done	April 2009 to July 2009		Composite trile instances (done), wierd switch support, probably other
16	Redesign/simplification of event system	done	Late June 2009		Especially the interface
17	Animated planes support	done	5/27/2009	6/3/2009	Source is an animated GIF, behaves like a planar art object
18	Basic content pipeline & editor support for NPCs	done	6/4/2009	6/10/2009	Properties dialog is still missing, but more pressing things await...
19	Vertically scrolling textures on triles & art objects	done			
20	Music areas in levels	done			Multiple music tracks in a level depending on "slice"
21	Level state persistence in a savefile	done	July 2009		
22	Heads-up display implementation	done	July 2009		
23	2D/generalized scriptable particle engine	done			
24	"Test hooks"	done			
25	Health/spawn point system implementation	done	July 2009		
26	Locked doors in editor	done	August 2009		
27	Allow sculpting of rotated triles	done	5/7/2009	5/8/2009	Rotated art objects are still problematic
28	Collision problems for stacked crates	done	5/8/2009	5/11/2009	Triles on top move faster than their grounds, passing through walls, et
29	Elastic collisions	done	5/11/2009	5/18/2009	Applied to some kinds of particles and non-collidable objects
30					If there are three trile layers and you are between two of them, the
	Three-layer background problems	done			background/foreground/wall-hugging goes berzerk

GLOBAL | VILLAGE | NATURE | INDUSTRIAL | ZU | UNDERGROUND | AQUADUCTS | DATES | TECHNOLOGY

	A	B	C	D	E	F	G
1	TYPE	NAME	DESCRIPTION	PROGRESS	START	END	WHO
2	Art	basic tileset	basic world geometry set	in progress	4/1/2009		phil
3		support beams	wooden support beams	done	12/5/2008	12/6/2008	phil
4		wooden wall	wooden support walls	in progress	12/6/2008		phil
5		mine cart	motor block variation	in progress	5/20/2009		phil
6		breakable blocks	mine level variant	done	12/4/2008	12/4/2008	phil
7		crystals	glowing crystals	done	12/4/2008	12/4/2008	phil
8		stalagmites	falling stalagatites	done	4/20/2009	4/20/2009	phil
9		giant bomb	really big bomb	done	4/20/2009	4/20/2009	phil
10		chain reaction block	block that can cause chain reactions when exploded	in progress	4/20/2009	4/20/2009	phil
11							
12	Design	level design		to do			phil
13							
14	Animations	dripping water		to do			annabelle
15		crumbling rock		to do			annabelle
16		big explosion		to do			annabelle
17		falling stalagtite		to do			annabelle
18							
19	Sound effects	dripping water		to do			jason
20		big explosion		to do			jason
21		echo		to do			jason
22							
23	Music	mine theme		to do			jason
24							
25	Technical	carts on rails	based on moving platforms, maybe specific changes?	to do			renaud
26		giant bombs	more dangerous than a standard bomb	to do			renaud
27		tnt block	basically a block that acts like a bomb	to do			renaud
28		motor blocks		to do			
29		tnt crates					
30							
31							

VILLAGE | NATURE | MINE | ZU RUINS | INDUSTRIAL | CRYSTAL CAVE | TRANSITION CAVE 02 | GRAVEYARD | TECHNOLOGY | GEEZER'S HOUSE | Bugs/To-

Designer vs. Programmer Dynamic

- Phil dictated design, but I implemented
  - As the programmer, YOU know what's possible
    - Time-wise, performance-wise, and in regards to your abilities
- You can always say no, but it's a discussion
  - FEZ is a "game d'auteur", designer associates with it deeply
  - It's all about mutual respect

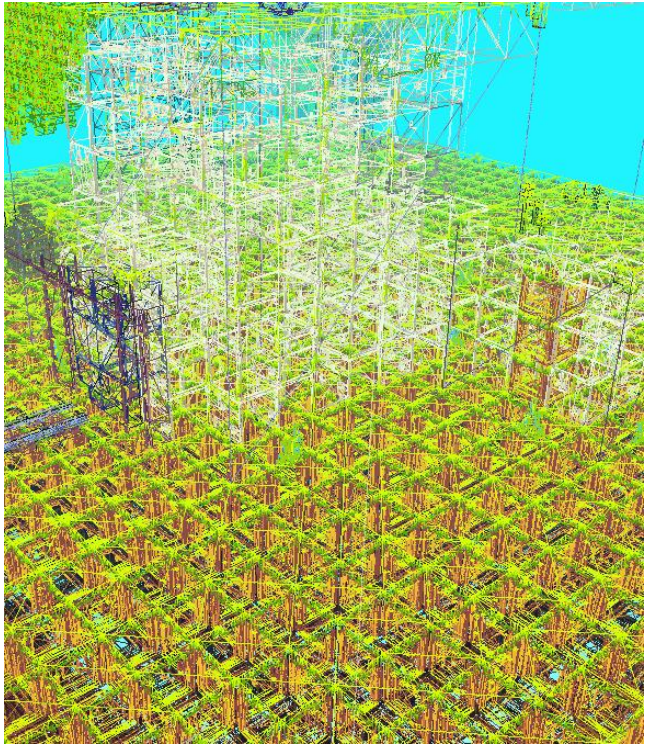
A note on Version Control

- Holy shit you guys, use version control.
- Used local NAS with batch file madness for whole 2009
  - Lost content, overwritten files, flaky backups
  - “I’m just one guy” and “I have infinite undo levels” are *not* good reasons
  - Remote SVN/Git/Hg servers are cheap, and they WILL save your ass

Thanks Nathan @ Capy for convincing us to make the jump!

How do we draw all that stuff?

- Lots of tiny blocks
- Geometry gets pretty dense
- Potentially lots of overdraw in 2D views
- Ideally 60 FPS on Xbox 360



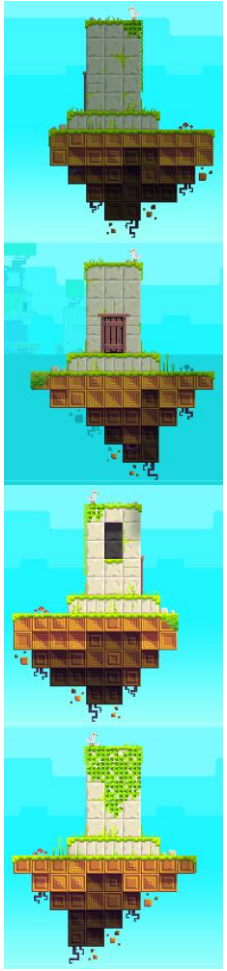
⇒ By **culling** and **batching** efficiently

Rendering Triles : Culling

- Orthographic views, usually looking into an axis
  - Peel the frontmost layer(s)
  - At most two layers if mid-rotation

- Cull within view frustum
- When moving, only invalidate triles for the screen's moving border

  (don't recull everything unless necessary)



How Culling Works

- Cell content is cached
- For each screen cell
  - Start at camera depth
  - Mark trile to be drawn
  - Walk into screen
  - Stop at solid, non-seethrough & non-offset trile

Culling Disabled

Culling Enabled (~3X less tiles)

Rendering Triles : Batching

- Draw calls are expensive
- All triles are independent (*so indie*)
  - Can't just throw everything at the GPU

⇒ Batching is a must

- But dynamic culling means dynamic batching...
  - Is there a simple way?

Rendering Trilos : ~~Batching~~ Instancing

- Lots of instances of a single trile type in a level
- Few things specific to an instance
  - XYZ Position, Y rotation : 4 single-precision floats
- vfetch instancing on Xbox 360

  ≈ Shader Model 3.0 Hardware Instancing on PC
- Max. 256 instances of the same trile per draw call

How-to : GPU Instancing (Xbox)

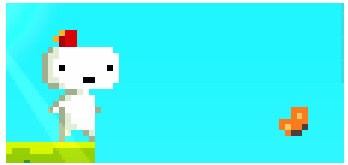
- A static vertex buffer
  - Vertex data of the template trile (lightweight)
- A dynamic index buffer
  - One set of indices for each instance
  - Offset indices as if you'd have as many clones in vertex buffer
- An instance data buffer
  - Actually a 4D vector or 4x4 matrix array in vertex shader constants
  - Contains the instance-specific data

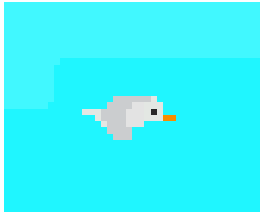
Xbox Instancing Vertex Shader

```
int vertexIndex = mod(IN.Index, VertexCount);            // Index is an automagic vertex input semantic
int instanceIndex = IN.Index / VertexCount;              // VertexCount is uniform parameter fed by app
asm
{
       vfetch position,          vertexIndex, position0
       vfetch normal,            vertexIndex, normal0
       vfetch textureCoordinate, vertexIndex, texcoord0
};                                                       // vfetch magic gets appropriate vertex data

float4 InstanceData = InstanceDataArray[instanceIndex];

float sinPhi, cosPhi;
sincos(InstanceData.w, sinPhi, cosPhi);                  // W component contains Y-axis rotation angle
float4x4 instanceMatrix =                                // Recompute instance transformation matrix
{
              cosPhi,        0,             -sinPhi,       0,
              0,             1,             0,             0,
              sinPhi,        0,             cosPhi,        0,
              InstanceData.xyz,                            1
};
```

Other Stuff : Planes/Decals

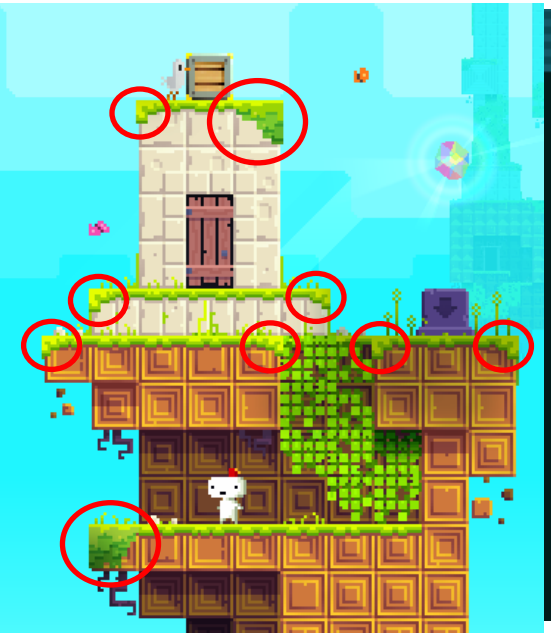
- Where sclupting doesn’t matter, or sprite animations



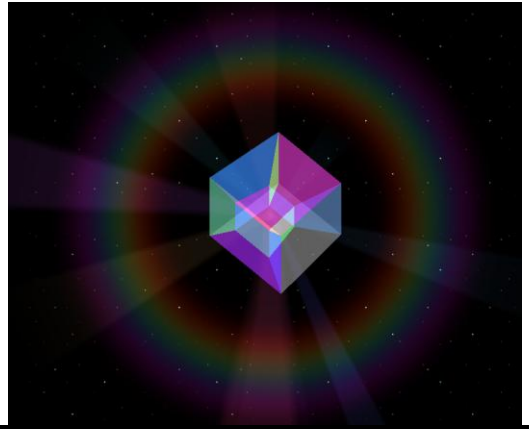


Other Stuff : Art Objects

- For small overlapping details, or unique bigger landmarks



DOT the Tesseract



- 4D hypercube fairy
- Continually rotates about the X-W plane
  - Done in C# (on the CPU)
  - 4x4 matrix works for rotation
- Faux 4D to 3D projection
  - Further out in W axis = smaller in 3D
- Rendered in orthographic projection like everything else
- 96 vertices, 144 triangles

  (no intersection in 4D space)

Collision Management

- Triles *are* the collision map
  - Each trile has a type
    - Can be per-face too!
- Four types
  - Immaterial (blades of grass)
  - No collision (background elements)
  - Top-collide (most platforms)
  - All-collide (blocking boundaries; rare)

Invisible Collision Triles

- Art objects need to collide too
  - But they're not made of triles!
- Filled with invisible triles
  - No collision, or
  - Top-only collision

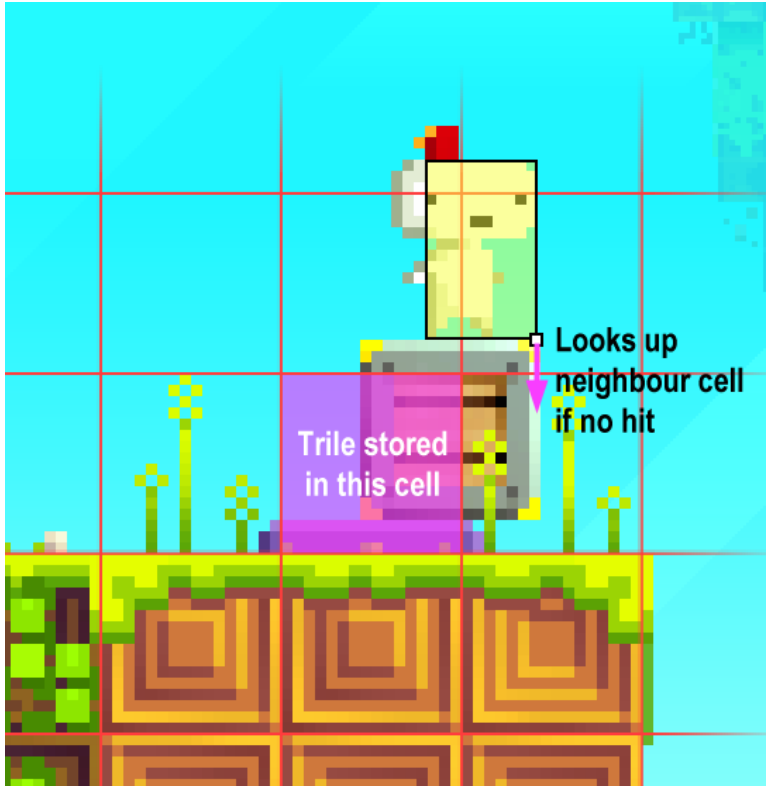
Collision Lookup

- Similar as the culling process
  - Collide with what you see!
  - Peel back from view
  - Keep peeling back if hitting
    - Immaterial triles (i.e. grass strands)
    - Non-grid-aligned triles

Otherwise, point-to-line with first solid trile edge found (three points), one axis at a time.

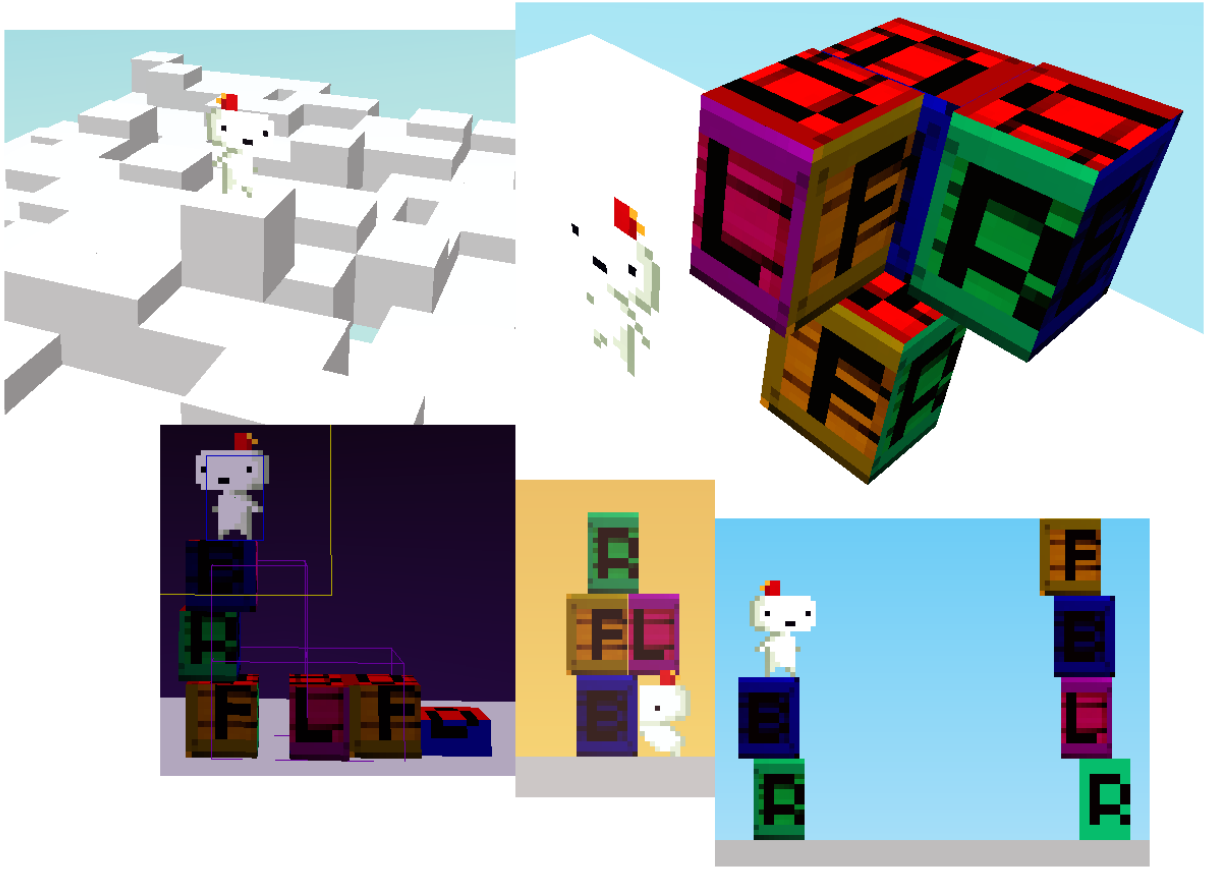
Non-Grid-Aligned?

- Triles can be moved arbitrarily
  - Stored in cell that contains its center
- Collision tests look up neighbours
  - Only one, away from cell's center
  - Only if no hit in current cell
  - Once found, point-to-line
    (using appropriate size & offset of the trile)



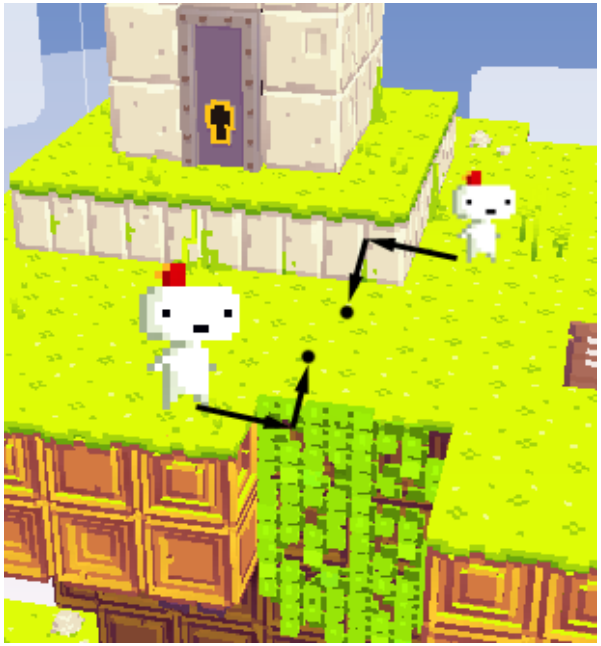
Tests...

- Scaling can also be arbitrary
- As long as triles are no smaller than half of Gomez's size
  (limitation of using point collision)



How Gomez moves around

- Movement is along on the view plane
- Depth correction rules
  - Gomez should stay visible, always
  - Gomez should never walk in mid-air
- Otherwise, don't touch his depth



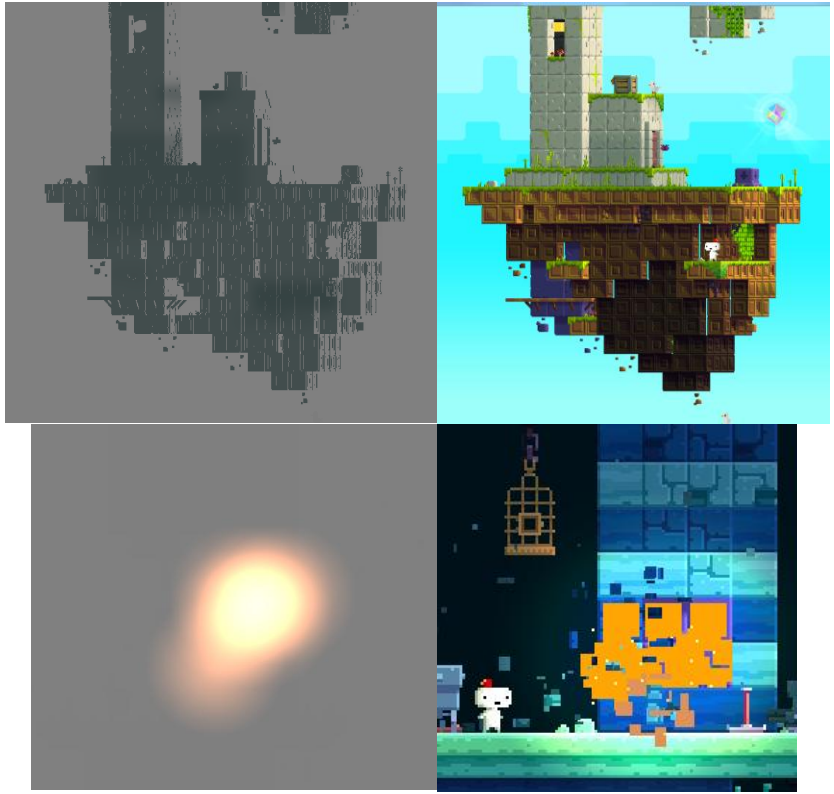
- During view rotations, movement & time are suspended

Background Mode

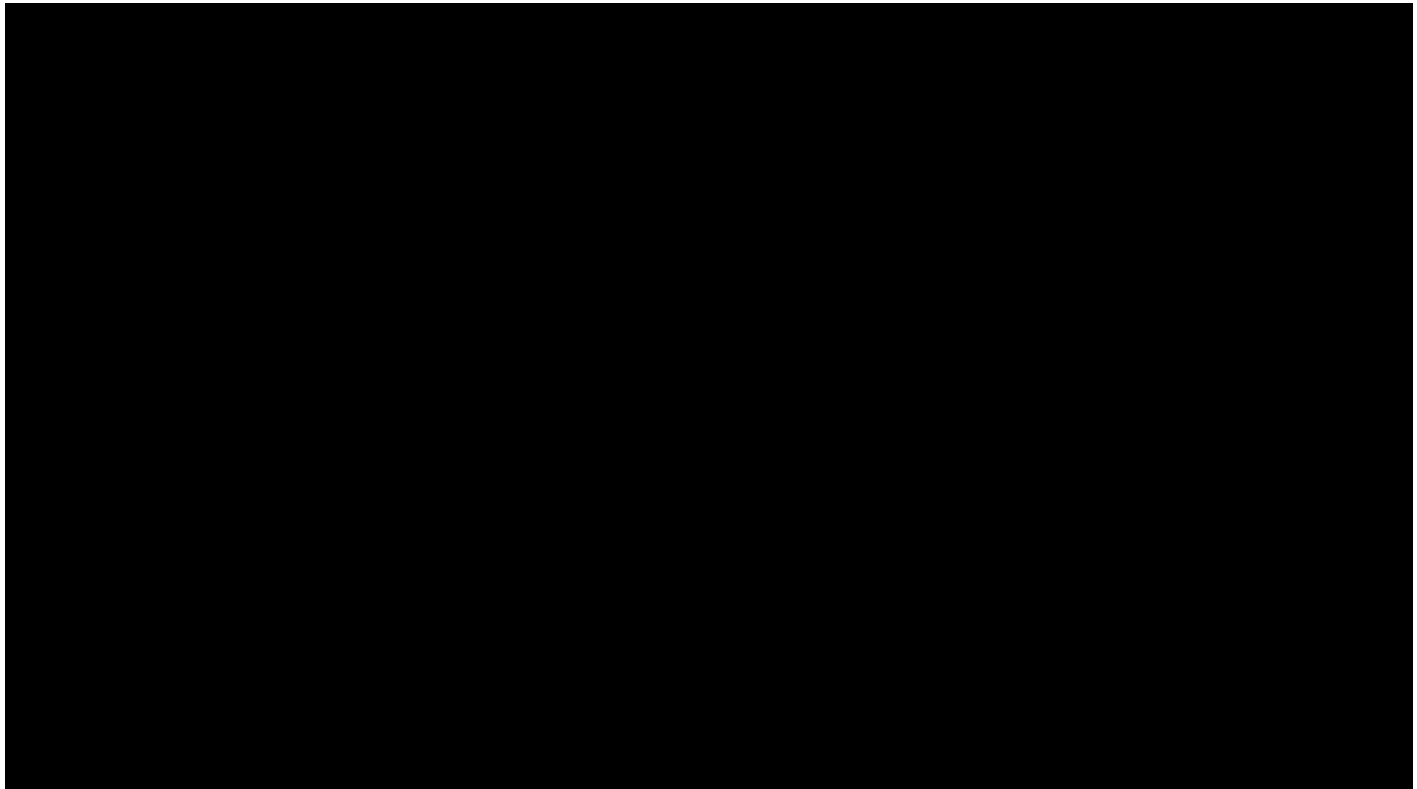
- If Gomez is behind the level post-rotation
  - IGF'08 build : Panic & QTE!
    - ⇒ Stressful, kills the mood, generally dumb
- Final build
  - Silhouette rendering
  - Low-pass music
  - Limited actions

Lighting Pre-Pass

- Per-face direct/diffuse light
- Ambient light = sky background color
  - Cloud shadows end up blueish
- Shadows and additional lights added
  (in screen-space)
- All done in a lighting pre-pass
- Blended in Modulate2X mode
  - so that it can light up and shadow



Time Of Day Lighting



World Interactions







- Gomez can :
  - Grab/push/pull objects
  - Rotate parts of the world independently
  - Make blocks crumble under his weight
  - Grab ledges all around platforms
  - Interact with one-off puzzle objects
  - Swim in water & drown in toxic liquids
  - AND MUCH MUCH MORE

- 56 different “action classes” control his behaviour

Action "Classes For States"

- All player action classes derive from a base abstract class
- Not all mutually exclusive
  - "Fall" is an action that is evaluated as long as there's gravity
- They know how to chain from state to state

```
protected virtual void TestConditions()
{
}

protected virtual void Begin()
{
}

protected virtual void End()
{
}

protected virtual bool Act(TimeSpan elapsed)
{
    return false;
}
```

Example : WalkRun.cs

```
protected override void TestConditions()
{
    switch (PlayerManager.Action)
    {
            case ActionType.Sliding:
            case ActionType.Idle:
            case ActionType.Teetering:
            case ActionType.Grabbing:
            case ActionType.Pushing:
            case ActionType.LookingAround:
                // If grounded and pressed movement keys
                if (PlayerManager.Grounded && InputManager.Movement.X != 0 &&
                    PlayerManager.PushedInstance == null)
                {
                        PlayerManager.Action = ActionType.Walking;
                }
                break;
    }
}
```

Example : WalkRun.cs

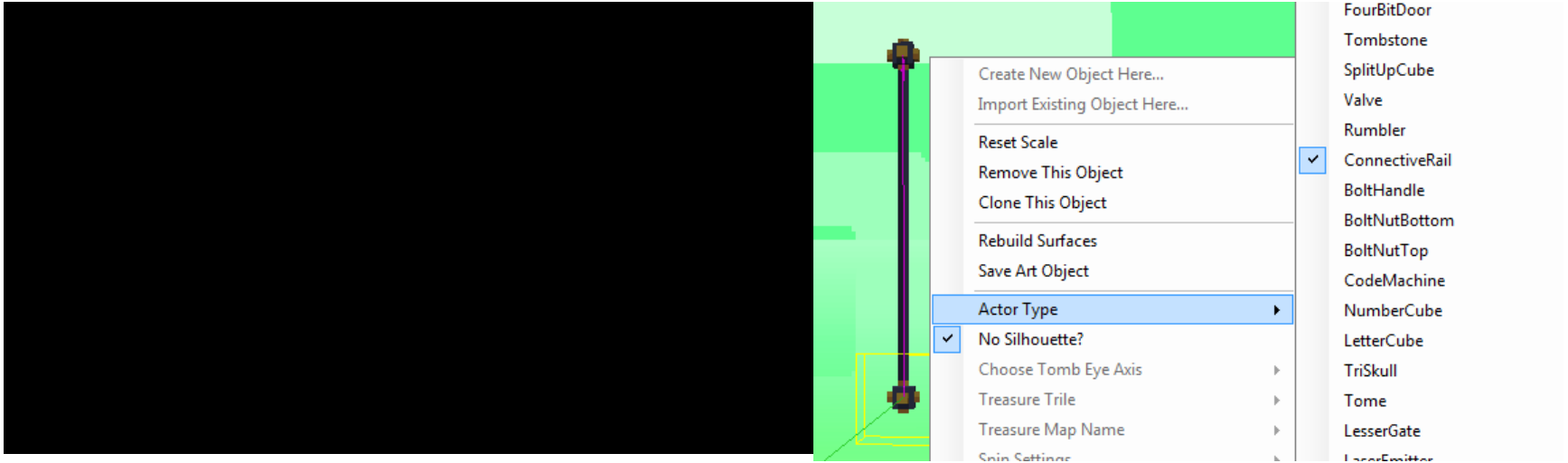
```
protected override void IsActionAllowed(ActionType type)
{
    return type == ActionType.Running || type == ActionType.Walking;
}

protected override bool Act(TimeSpan elapsed)
{
    // Transform input to physics impulses in a helper class
    MovementHelper.Update((float)elapsed.TotalSeconds);

    if (MovementHelper.Running)
            PlayerManager.Action = ActionType.Running;
    else
            PlayerManager.Action = ActionType.Walking;
}
```

Additional Flags for Actions

```
public static class ActionTypeExtensions
{
    public static bool IsAnimationLooping(this ActionType type)
    {
            switch (action) { /* ... */ }
    }

    public static bool DisallowsRespawn(this ActionType type)
    {
            switch (action) { /* ... */ }
    }

    public static bool PreventsRotation(this ActionType type)
    {
            switch (action) { /* ... */ }
    }
    /* ... */
}
```

Actors : Dynamic World Entities

- Spinning blocks, moving platforms, ladders, interactive structures, etc.
- Hardcoded behaviours with flags or parameters set in the editor
- Tradeoff for not having proper scripting support



Scripting System

- Designer-friendly UI (no code!)
- Event-based

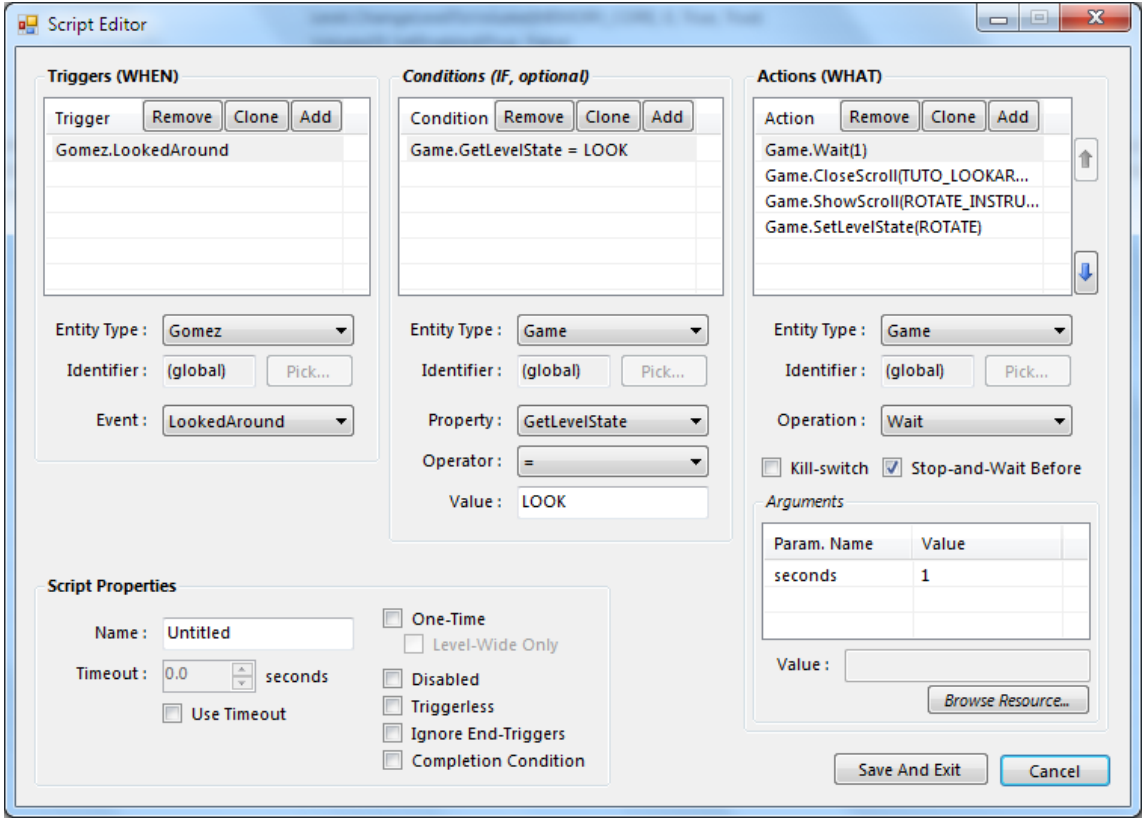
Scripts Browser

Id	Trigger	Condition	Action
10	Level.Start		Dot.SpiralAround(True, False), Dot.Say(DOT_TRIAL_A, True, False), Dot.Say(DOT_TRIAL_B, T
11	Volume[6].Enter	Game.GetLevelState != JUMP, Game.GetLevelState != LOOK, ...	Game.ShowScroll(TUTO_VINE, 5, True, True)
4	Volume[3].Enter		Level.ChangeLevelToVolume(MEMORY_CORE, 0, True, True)
6	BitDoor[38].Open		Volume[3].SetEnabled(True, False)
7	Level.Start		Camera.SetPixelsPerTrixel(3), Volume[3].SetEnabled(False, False)
8	Level.Start	Level.FirstVisit = False	Sound.ChangeMusic()
9	Level.Start	Level.FirstVisit = True	Sound.ChangeMusic(Clear Skies)
13	Gomez.Jumped	Game.GetLevelState = JUMP	Game.CloseScroll(TUTO_JUMP), Game.ShowScroll(TUTO_LOOKAROUND, 5, True, False), Ga
14	Gomez.LookedAround	Game.GetLevelState = LOOK	Game.Wait(1), Game.CloseScroll(TUTO_LOOKAROUND), Game.ShowScroll(ROTATE_INSTRU
15	Camera.Rotated	Game.GetLevelState = ROTATE	Game.Wait(1), Game.CloseScroll(ROTATE_INSTRUCTIONS), Game.SetLevelState()
23	Volume[7].Enter		Game.ShowScroll(TUTO_GRAB_LEDGE, 0, True, True)
20	Volume[3].Enter		Game.ShowScroll(TUTO_ENTER_DOOR, 0, True, True)
21	Gomez.EnteredDoor		Game.CloseScroll(TUTO_ENTER_DOOR)
22	Gomez.ClimbedVine		Game.Wait(1), Game.CloseScroll(TUTO_VINE)

Script Editor

- Triggers
- Conditions
- Actions

- And lots of WinForms controls



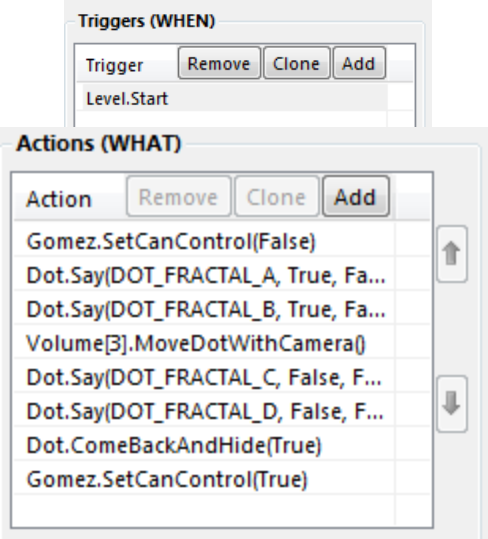
Script Example : DOT interaction

- Warn the player of a particular mechanic in a level using DOT

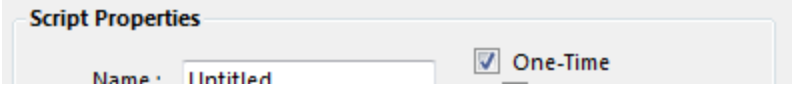
When Level Starts, (`Start` event on `Level` static entity)

(blocking actions executed in sequence)

- Remove player controllability
- DOT says a couple lines
- Move the camera to Volume #3 (point of interest)
- DOT says more stuff
- DOT comes back and hide
- Player regains control



This script happens once in the game :



Scripting Interfaces

```
[Entity(Static = true)]
public interface ILevelService : IScriptingBase
{
        // Events, for triggers
        [Description("When the level starts")]
        event Action Start;
        void OnStart(); // Called by the engine to trigger the event

        // Properties, for conditions
        bool FirstVisit { get; }

        // Operations, for actions (can be running over time à la coroutine)
        [Description("Smoothly changes to a new water height")]
        LongRunningAction SetWaterHeight(float height);
}
```

Level Format

SDL SIMPLE DECLARATIVE LANGUAGE
a simple universal language for describing typed data in lists, maps and trees

- At design-time, serialized from objects to SDL
  - Similar to YAML or JSON, but better integrated with .NET
  - Tweakable by hand
  - Error resilient
    - Ignores unknown elements
    - Elements can be marked as optional
- At compile-time, binary format for performance & filesize
  - No automatic serialization, reflection is too slow on Xbox

SDL Looks Like This

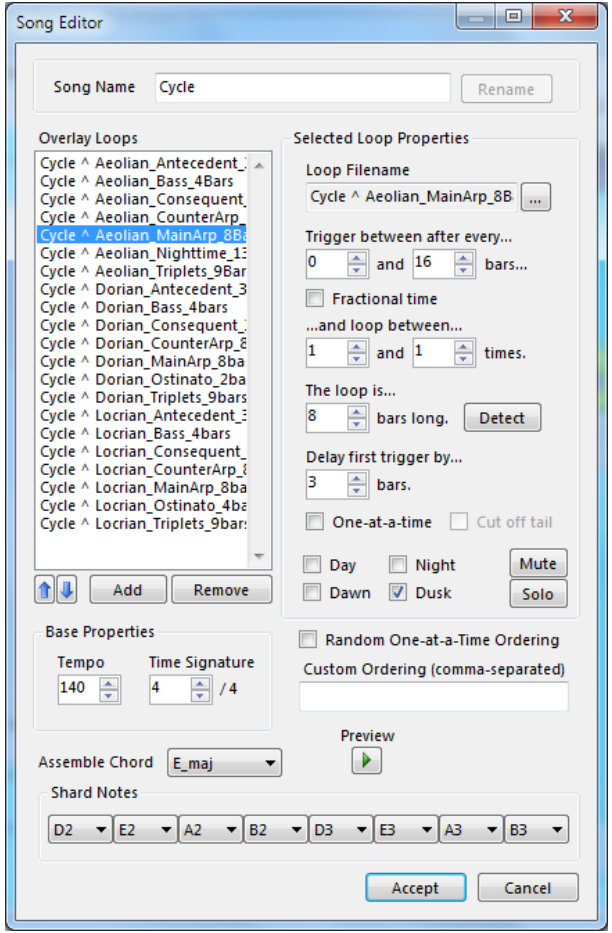
- Output much more concise than equivalent XML serialization
- Serialization tags in data objects

```
public string Name { get; set; }

public TrileFace StartingPosition { get; set; }

public Vector3 Size { get; set; }

[Serialization(Optional = true)]
public float BaseDiffuse { get; set; }

[Serialization(CollectionItemName = "Trile")]
public Dictionary<TrileEmplacement, TrileInstance> Triles;
```

```
level type="FezEngine.Structure.Level, FezEngine" {
    name "ARCH"
    startingPosition {
        face "Front"
        id 16 6 5
    }
    size 30F 49F 35F
    baseDiffuse 1F
    baseAmbient 0.35F
    haloFiltering true
    blinkingAlpha false
    waterHeight 11F
    skyName "WATERFRONT"
    trileSetName "Untitled"
    volumes {
        volume key=0 {
                orientations "Front"
                actorSettings {
                    farawayPlaneOffset 5F 1F
                }
                from 3F 26F 15F
                to 4F 28F 16F
        }
```

Music System

- Written on-top of XACT
- Allows infinite, dynamic track-based songs
- Scriptable
  - Level/player events can mute/unmute tracks
- Works with time of day



Music System In Action

- "Puzzle-solving music" @ daytime

Track	Initial Delay	Duration	Inter-Play Delay
Main Arp	None	8 bars	[0, 16] bars
Counter Arp	4 bars	8 bars	[0, 16] bars
Ostinato	8 bars	4 bars	[0, 16] bars
Bass	8 bars	4 bars	[0, 24] bars
Antecedent	16 bars	3 bars	[0, 16] bars
Triplets	16 bars	9 bars	[0, 16] bars
Consequent	20 bars	3 bars	[0, 16] bars

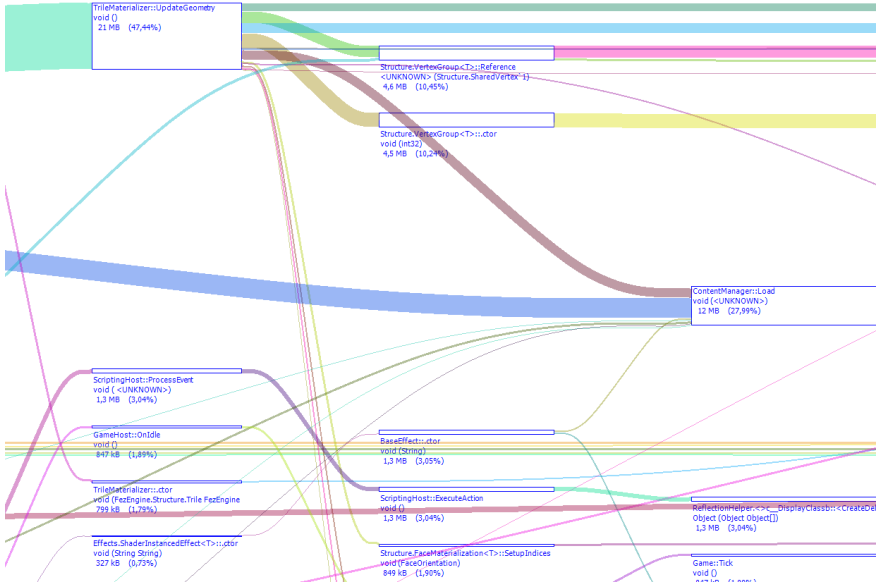
- Night is less dense, very different sounding and still randomized :

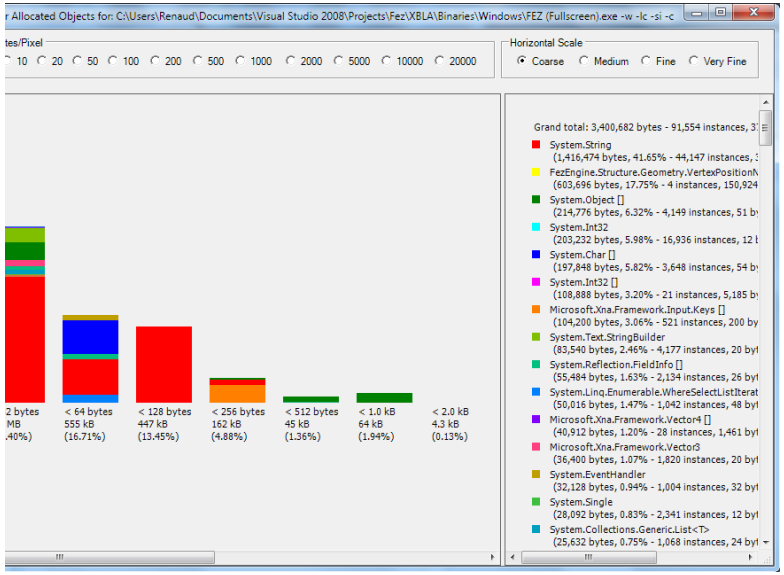
Xbox-specific Optimization

- XNA on the Xbox 360 = .NET Compact Framework
  - Garbage collection every 1Mb allocated, unpredictable, blocking
  - Rules of thumb : avoid LINQ, dynamic allocations
- Draw calls are expensive : batching is essential, instantiate ALL THE THINGS
  - Defer as many operations as possible to vertex shaders instead of CPU
  - Otherwise, multithread; 5 cores at your disposal
- HDD access is slow, flash memory access is worse!
  - Pre-load all content, avoid disk access later on
  - You probably have more RAM than content (in FEZ, totally)

Tools : CLR Profiler

- Excellent, free memory profiler
- Allocations graphs, memory usage by class
- Good for identifying real-time garbage & why load-times stall





Tools : CPU Profiler

- Any one you want (AQTime, dotTrace...), but I like ANTS Performance Profiler a lot
- Absolutely essential for finding bottlenecks

[.NET Executable] C:\Users\Renaud\Documents\Visual Studio 2008\Projects\Fez\XBLA\Binaries\Windows\Fez (fullscreen).exe* (Line-Level; Only Methods With Source) - ANTS Performance Profiler 6.2.0.13 Standard

File View Timeline Tools Help

Performance
SQL Server
File I/O

% Processor Time
100%
00:10 00:20 00:30 00:40 00:50

Select All
Bookmark Region
Stop Profiling

Tree View Display Options | Top-down (methods with source) | (All Threads) | CPU time | Hide insignificant methods

Method	Time (%)	Time With Children (%)	Hit Count
FezGame.Program.Main(string[] args)	<0.001	99.872	1
Common.Logger.Try(Action action)	<0.001	99.872	1
FezGame.Program.MainInternal()	<0.001	99.872	1
(Collapsed methods without source. To view these methods, change the display option to show all methods)	7.436	99.872	1
FezGame.Fez.Draw(GameTime gameTime)	0.379	50.070	347
FezGame.Components.PlayerActions.Update(GameTime gameTime)	0.169	11.998	385
FezGame.Components.PlayerActions+ActionsManager.Update(GameTime gameTime)	0.171	4.599	385
FezGame.Components.DotHost.Update(GameTime gameTime)	0.005	3.477	385

Find: Previous Next

'C:\Users\Renaud\Documents\Visual Studio 2008\Projects\Fez\XBLA\Source\FezGame\Components\PlayerActions.cs' Line-Level Timings

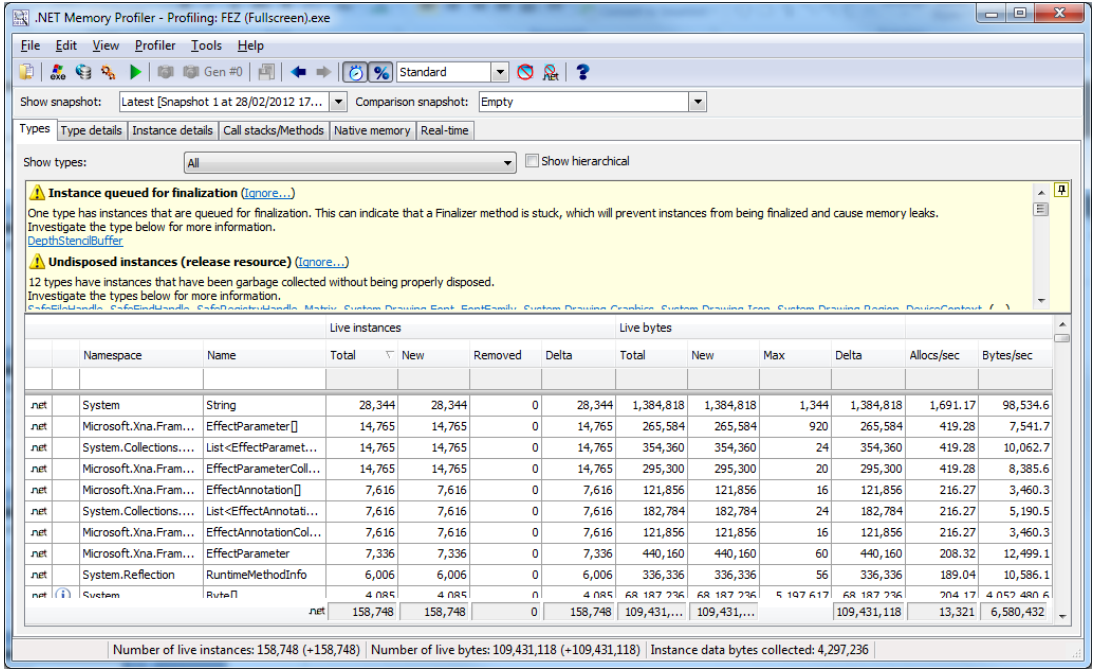
Line	Hit Count	Avg Time (%)	Time (%)	
184	122	0.000	0.000	}
185				
186				// Update physics on it
187	385	0.000	0.005	var oldVelocity = PlayerManager.Velocity;
188	385	0.029	11.105	PhysicsManager.Update(PlayerManager);
189				
190				// Can't hang by the 4 front pixels of his body alone
191	385	0.000	0.016	if (PlayerManager.Grounded && PlayerManager.Ground.NearLow == null)

Tools : Analyzing Memory Profiler

- CLR Profiler is good for garbage generation, but isn't very helpful for leaks

- I used the SciTech .NET Memory Profiler

- Heap snapshot comparisons
- Insight on possible problems
- Leaky objects are identified and their creation point given



		Live instances				Live bytes					
Namespace	Name	Total	New	Removed	Delta	Total	New	Max	Delta	Allocs/sec	Bytes/sec
.net System	String	28,344	28,344	0	28,344	1,384,818	1,384,818	1,344	1,384,818	1,691.17	98,534.6
.net Microsoft.Xna.Fram...	EffectParameter[]	14,765	14,765	0	14,765	265,584	265,584	920	265,584	419.28	7,541.7
.net System.Collections....	List<EffectParamet...	14,765	14,765	0	14,765	354,360	354,360	24	354,360	419.28	10,062.7
.net Microsoft.Xna.Fram...	EffectParameterColl...	14,765	14,765	0	14,765	295,300	295,300	20	295,300	419.28	8,385.6
.net Microsoft.Xna.Fram...	EffectAnnotation[]	7,616	7,616	0	7,616	121,856	121,856	16	121,856	216.27	3,460.3
.net System.Collections....	List<EffectAnnotati...	7,616	7,616	0	7,616	182,784	182,784	24	182,784	216.27	5,190.5
.net Microsoft.Xna.Fram...	EffectAnnotationCol...	7,616	7,616	0	7,616	121,856	121,856	16	121,856	216.27	3,460.3
.net Microsoft.Xna.Fram...	EffectParameter	7,336	7,336	0	7,336	440,160	440,160	60	440,160	208.32	12,499.1
.net System.Reflection	RuntimeMethodInfo	6,006	6,006	0	6,006	336,336	336,336	56	336,336	189.04	10,586.1
.net System	Byte[]	4,085	4,085	0	4,085	68,187,236	68,187,236	5,197,617	68,187,236	204.17	4,052,480.6
.net		158,748	158,748	0	158,748	109,431,...	109,431,...		109,431,118	13,321	6,580,432

XDK Tools

All other tools worked with a PC build; what if stuff only happens on Xbox?

- xbWatson
  - Make your own measurements and output to console
- PIX
  - Frame-by-frame teardown of what's costly
  - Excellent for inspecting the draw calls & shaders

- CPU Performance Profiler in the XDK ultimately useless
  - Made for native code, not .NET games

XNA on XBLA : My Experience

- Long dev cycle meant struggle with upgrades
- No native library allowed, only .NET : can be problematic
- Some boilerplate TCR stuff handled, but still a lot to think about
- No symbols for debugging .NET assemblies in Release builds

But...

- .NET, C#, XNA and WinForms make engine & tools dev way easier
- Transition from PC to Xbox all in all fairly painless
- It’s all about comfort : I couldn’t have done FEZ in C++

A lesson is learned...?

- I don't think there's a way around it : a first game is HARD to finish
  - Especially if you care a lot about it
  - And let's face it, FEZ is a huge game
- Early showing was a double-edged sword
  - But later PAX/Fantastic Arcade showing were great motivators & feedback tools
- Feature creep, constant feeling that finish line is 3-6 months away
  - Making short-form "game jam" games helped learn scope control

If I had to do it again...

- Use middleware (Unity?), or hire an engine programmer
- Have real scripting support and educate artists about it
- Hot-reloadability of scripts and content edits
  - Even if C# compilation is fast, back & forth is huge waste of time
- Don’t be afraid to scrap prototype code
  - 4-year-old bugs coming back to haunt you : it sucks
  - Realize you’re doing a big, long project, and that it’s worth the effort

That's all, folks!

- Thanks for coming!
- Questions?