# Cubes Fill The Ubay Cour



#### FEZ Technical Post-Mortem By Renaud Bédard, Programmer (Polytron)

#### About Me

- Started messing with 3D programming in VB6 & Truevision3D around 2001
- Started working with Phil Fish on FEZ in April 2007
- FEZ gets 2 nominations and 1 award (Visual Arts) at IGF'08
- Bacc. in Computer Science at UQÀM in late 2008
- Worked full-time since then at Polytron to make the game
- FEZ is first commercial title and full-time "industry" "job"



- 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

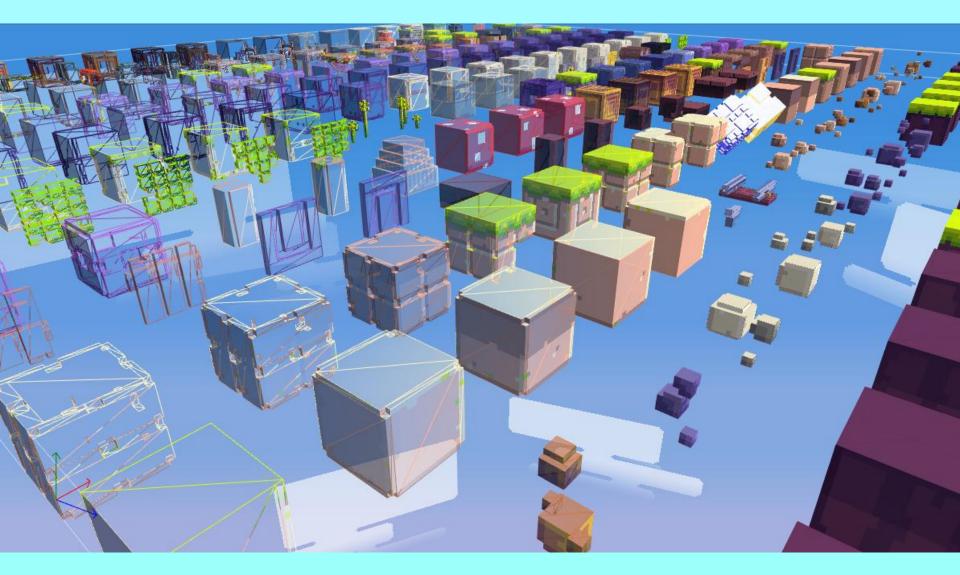
   Spanned 5 XNA versions, from 1.0 to 3.1!

# Long Screenshot

#### How is the world "cut up"?

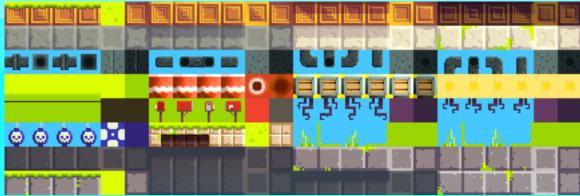
- Art and level production-wise, easier to work with tiles
- Plain old 2D tilemap doesn't work : we need 3D tiles
  - ...triles?
- 16x16x16 trixels chosen as an arbitrary fixed trile size

#### Noture Trile Set (soe triles)



#### Texturing Triles

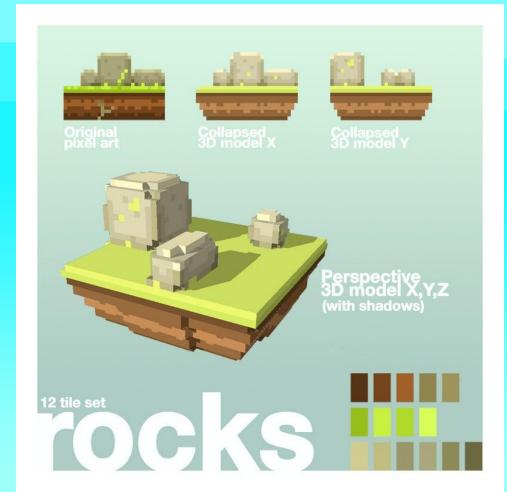
- Per-trixel coloring?
  - Implies good painting tools in modeler



- Cubemaps! (freebloods) F R B L U
  - Triles need to be convex
  - Cubemaps are 16x16x6 32bpp = 6Kb uncompressed!

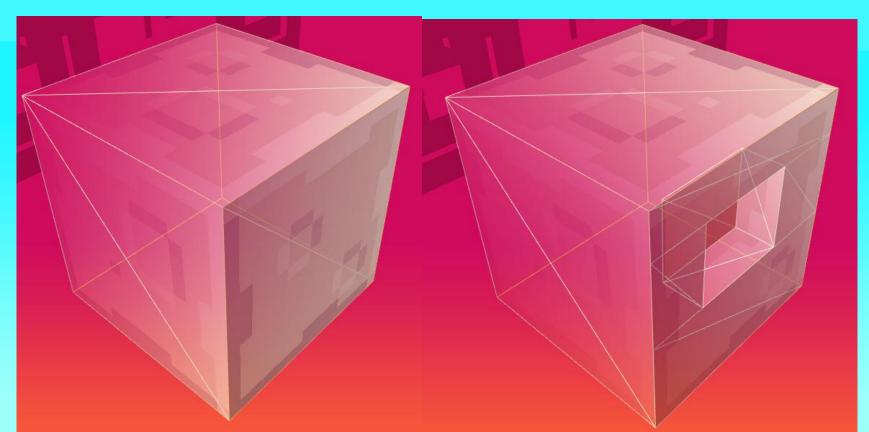
#### Modeling Triles

#### First trile concepts made in Google Sketchup



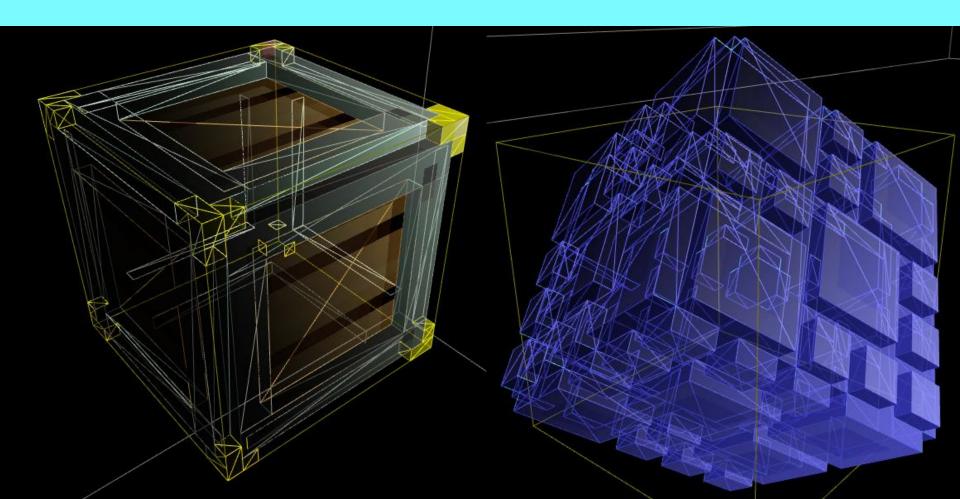
#### Modeling Triles

- Trixel sculpting
  - Push trixel faces in, pull faces out
  - Adaptative geometric complexity



#### Modeling Triles

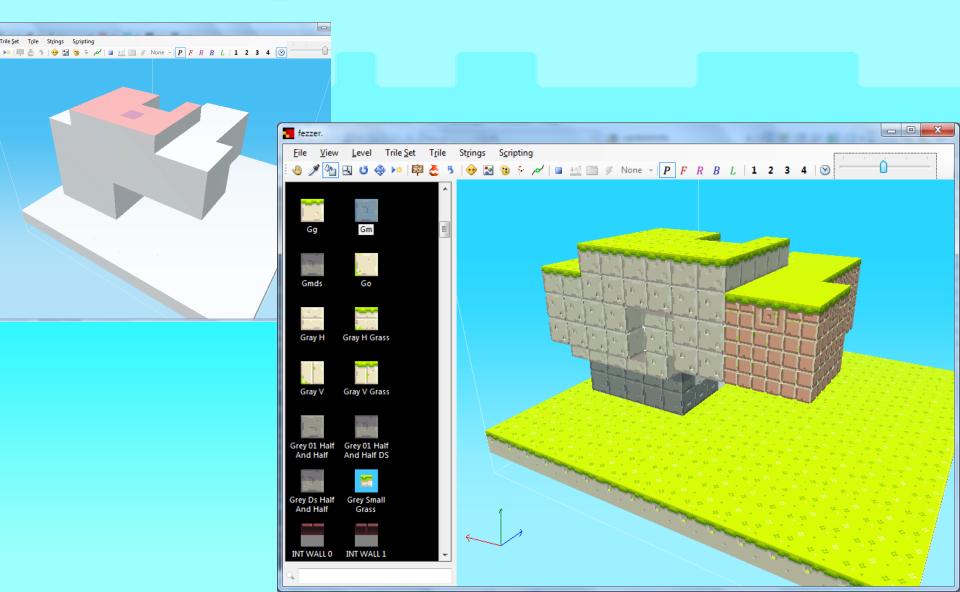
And things can get pretty complex



#### Mash Simplification

- Extrapolate contiguous surfaces from trixels
- For each surface,
  - Enumerate biggest rectangles in that surface
  - Each rectangle becomes a plane
- Triles have an average of 136 polygons
- Typical scene : 50,000 polygons on-screen
   Goes from anywhere from 5000 to 250,000

### Building Levels in "Fezzer"



#### Rendering Triles : Culling

- Orthographic views
  - Peel the frontmost layer
  - Keep offset and see-through triles
- During rotation transitions
  - Peel the two visible faces
- Always use frustum culling
  - Only cull the difference, leave what's still visible there

# Culling Disabled



# Culling Enabled



#### Rendering Triles : Instancing

- Lots of instances of the same trile in a level
- Few things specific to a single instance
   XYZ Position, Y Rotation : 4 floats

- vfetch Instancing on Xbox 360
   ≈ SM 3.0 Hardware Instancing on PC
- Max. 226 instances of same trile per draw call

### Instancing shader details

#### Instance data passed as VS constants

#define InstancesPerBatch 226

float4 InstancePositionPhiArray[InstancesPerBatch] : register(c30);

#### Reconstruct transformation matrix in VS

```
float sinPhi, cosPhi;
sincos(InstancePositionPhi.w, sinPhi, cosPhi);
float4x4 instanceMatrix =
{
    cosPhi, 0, -sinPhi, 0,
    0, 1, 0, 0,
    sinPhi, 0, cosPhi, 0,
    InstancePositionPhi.xyz, 1
};
```

#### Other World Objects : Flanes/Decals

Where sculpting doesn't matter, or sprite animations



#### other World Objects : Firt Objects

• For small overlapping details, or unique bigger landmarks





#### Collision Management

Triles are the collision map

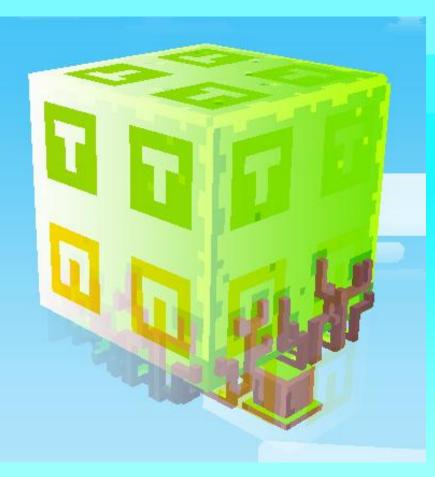


- Four types
  - Immaterial (e.g. blades of grass)
  - No collision (background elements)
  - Top-collide (most platforms)
  - All-collide (blocking boundaries)



#### Invisible Collision Triles

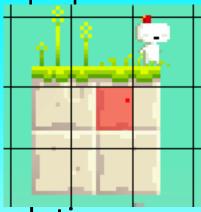
Art objects are filled with invisible collision triles



#### Collision Lookup

- Level is a large structure of triles indexable by (x, y, z)
   Actually a Dictionary<Int3, Trile>
- You collide with what you see
  - Get closest (x, y) or (z, y) row of triles
  - Traverse to nearest collideable trile and use its properties
- Gets tricky with offset triles

   Look up 4 closest neighbour rows and test



After that, normal AABB to line collision resolution

#### Offset/Non-Unit Triles



#### How Gomez Moves Around

Movement is in 2D according to the current viewpoint.

# Depth correction Gomez should stay visible at all times Gomez should never walk in mid-air Otherwise, don't change Gomez's depth arbitrarily

• When the camera rotates, movement (and time) is suspended.

#### Background Mode

 If Gomez is behind geometry after a rotation, don't panic!



- Silhouette rendering
- Low-pass music
- Limited actions

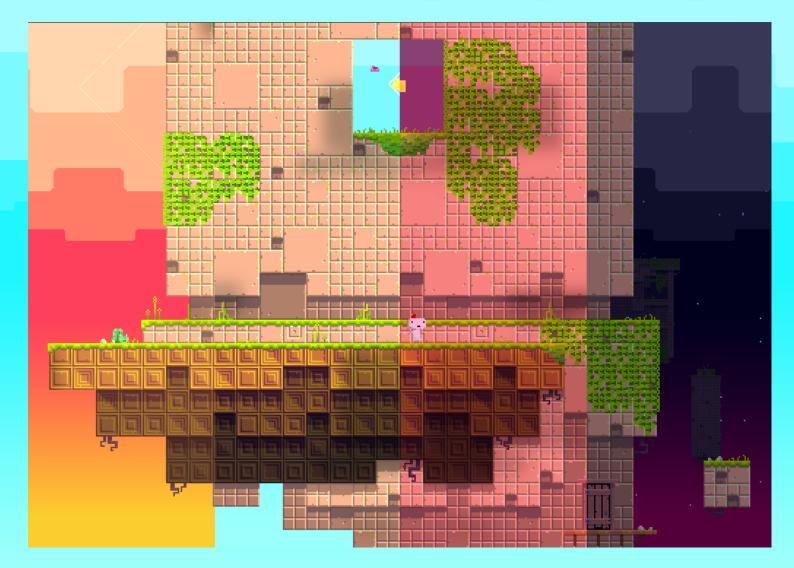


# Lighting

- Per-face direct/diffuse light
- Ambient light based on sky background color
  - Also tints transparent cloud layers
- Shadows and additional lights added in screen-space
- All done in a lighting pre-pass (render-to-texture)
  - Blended in Modulate2X mode so that it can light up and shadow



#### Time of Day Lighting



#### Dynamic World Interactions

- FEZ @ IGF'08 was 99% static
- Now 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 "Objects For States"

ł

}

ł

}

{ }

}

- 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
- Tied to sprite animations, player controllability, etc.

protected virtual void TestConditions()

```
protected virtual void Begin()
```

```
protected virtual void End()
```

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

#### Dynamic World Entities = Fictors

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

					Tombstone
- T		Create New Object Here			SplitUpCub
		Import Existing Object Here			Valve
- 1		Reset Scale			Rumbler
		Remove This Object		*	Connective
		Clone This Object			BoltHandle
_					BoltNutBott
		Rebuild Surfaces			BoltNutTop
		Save Art Object			CodeMachi
		Actor Type	•		NumberCul
	~	No Silhouette?			LetterCube
		Choose Tomb Eye Axis	•		TriSkull
- I 🔶 I		Treasure Trile	►		Tome
¥-		Treasure Map Name			LesserGate
		Spin Settings			LacorEmitta

nbstone itUpCube ve nbler nnectiveRail tHandle tNutBottom tNutTop deMachine mberCube terCube skull ne serGate

FourBitDoor

## Scripting System

- Designer-friendly UI (no code!)
- Event-based (no continuous evaluation)
- Extensible and backwards-compatible

#### 🖳 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,
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), G
14	Gomez.LookedAround	Game.GetLevelState = LOOK	Game.Wait(1), Game.CloseScroll(TUTO_LOOKAROUND), Game.ShowScroll(ROTATE_INSTR
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)

### Final Script Editor UI

Covint Editor

#### ... is scary.



riggers (WHEN)	Conditions (IF, optional)	Actions (WHAT)		
Trigger Remove Clone Ad Gomez.LookedAround	d Condition Remove Clone Add Game.GetLevelState = LOOK	Action Remove Clone Add Game.Wait(1) Game.CloseScroll(TUTO_LOOKAR Game.ShowScroll(ROTATE_INSTRU Game.SetLevelState(ROTATE)		
Identifier : Gomez Identifier : (global) Pick. Event : LookedAround	<ul> <li>Entity Type : Game</li> <li>Identifier : (global)</li> <li>Property : GetLevelState</li> <li>Operator : =</li> <li>Value : LOOK</li> </ul>	Entity Type : Game Identifier : (global) Pick Operation : Wait Kill-switch Stop-and-Wait Before Arguments		
		Param. Name Value seconds 1		
icript Properties Name : Untitled Timeout : 0.0 🗼 second	<ul> <li>One-Time</li> <li>Level-Wide Only</li> <li>Disabled</li> <li>Triggerless</li> <li>Ignore End-Triggers</li> <li>Completion Condition</li> </ul>	Value : Browse Resource Save And Exit Cancel		

#### Serialized Scripts

}

- Scripts are objects, serialized as text inside the level file
- Tweakable by hand
  - If you are careful with braces
- Probably should've written a better formatter like :

```
script key=0 {
  code "Level.Start =>
   Camera.SetPixelsPerTrixel(2)"
}
```

```
script key=0 {
  name "Untitled"
 triggers {
    trigger {
      event "Start"
      object {
        type "Level"
      }
  actions {
    action {
      operation "SetPixelsPerTrixel"
      arguments "2"
      object {
        type "Camera"
```

#### Scripting Interfaces

Reflected by the editor to fill dropdowns

#### Music System

- Written on-top of XACT
- Allows dynamic track-based songs
- Works with time of day
- Horribly intricate and complicated, as requested by Disasterpeace!
  - ...but allows for kickass songmaking

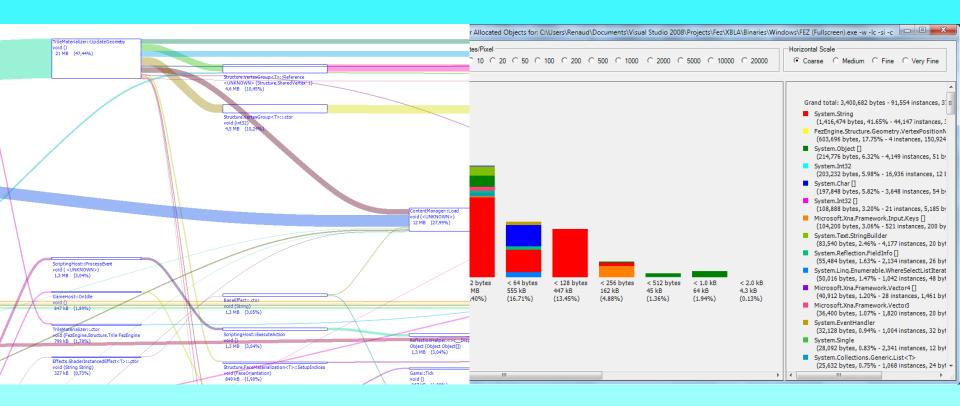
Song Editor				
Song Name Cycle	Rename			
Overlay Loops Cycle ^ Aeolian_Antecedent_i Cycle ^ Aeolian_Bass_4Bars Cycle ^ Aeolian_Consequent Cycle ^ Aeolian_CounterArp Cycle ^ Aeolian_MainArp_8Ba Cycle ^ Aeolian_Mightime_1: Cycle ^ Aeolian_Mightime_1: Cycle ^ Dorian_Antecedent_3 Cycle ^ Dorian_Consequent_i Cycle ^ Dorian_Consequent_i Cycle ^ Dorian_CounterArp_8 Cycle ^ Dorian_CounterArp_8 Cycle ^ Dorian_CounterArp_8 Cycle ^ Dorian_MainArp_8ba Cycle ^ Dorian_MainArp_8bars Cycle ^ Dorian_CounterArp_5 Cycle ^ Locrian_CounterArp_5 Cycle ^ Locrian_CounterArp_6 Cycle ^ Locrian_CounterArp_6 Cycle ^ Locrian_CounterArp_6 Cycle ^ Locrian_CounterArp_6 Cycle ^ Locrian_CounterArp_6 Cycle ^ Locrian_CounterArp_6 Cycle ^ Locrian_Ostinato_4ba Cycle ^ Locrian_Triplets_9bars	Selected Loop Properties Loop Filename Cycle ^ Aeolian_MainArp_8B Trigger between after every 0 and 16 bars Fractional time and loop between 1 and 1 times. The loop is 8 bars long. Detect Delay first trigger by 3 bars. One-at-a-time Cut off tail			
Add Remove	Day Night Mute Dawn V Dusk Solo			
Base Properties Tempo Time Signature 140 (*) 4 (*) / 4	Random One-at-a-Time Ordering Custom Ordering (comma-separated)			
Preview Assemble Chord $\boxed{E_maj}$ $\checkmark$ $\boxed{\blacktriangleright}$ Shard Notes $\boxed{D2}$ $\checkmark$ $\boxed{E2}$ $\checkmark$ $\boxed{A2}$ $\checkmark$ $\boxed{B2}$ $\checkmark$ $\boxed{D3}$ $\checkmark$ $\boxed{E3}$ $\checkmark$ $\boxed{A3}$ $\checkmark$ $\boxed{B3}$ $\checkmark$				
D2 • E2 • A2 • B2	→ D3 → E3 → A3 → B3 → Accept Cancel			

#### xbox 350 Optimizations

- XNA on the Xbox 360 = .NET Compact Framework
  - Garbage collection every 1Mb allocated, unpredictable, blocking
  - Draw calls are expensive : batching is essential, instantiate ALL THE THINGS
  - Defer as many draw calculations as possible to vertex shaders instead of CPU
  - Otherwise, multithread; 5 (slow) cores at your disposal
- Lots of content on the Web about this
  - ...especially since XBLIG games have the same issues
  - Rules of thumb : avoid LINQ, dynamic allocations

#### Tools : CLR Profiler

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



#### Tools : CPU Profiler

- Any one you want (AQTime, dotTrace...)
  - 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							
<u>F</u> ile <u>V</u> iew Time <u>l</u> ine <u>I</u> ools <u>H</u> elp							
Performance         Performance         00:10         00:20         00:30         00:30	:40	00:50	Select All				
U SQL Server ▶		~ ?	Bookmark Region				
File I/O	-	Stop Profiling					
			र् 🔍 🗖				
🛓 Tree View Display Options 🔉 Top-down (methods with source) 💌 (All Threads) 💌 CPU time 💌 🕅 Hide insignificant methods 🔮							
Method	Time (%)	Time With Children (%) 🛛 🔻	Hit Count				
EzGame.Program.Main(string[] args)	<0.001	99.872	1				
Common.Logger.Try(Action action)	<0.001	99.872					
😑 🚥 FezGame.Program.MainInternal()	<0.001	99.872					
😑 🚥 (Collapsed methods without source. To view these methods, change the display option to show all methods)	7.436	99.872					
🗄 🚥 FezGame.Fez.Draw(GameTime gameTime)	0.379	50.070					
FezGame.Components.PlayerActions.Update(GameTime gameTime)	0.169	11.998					
FezGame.Components.PlayerActionsHactionsManager.Update(GameTime gameTime)	0.171	4.599					
Image: Book State (Game Time game Time)       0.005       3.477       385							
Find: P V I Previous 4 Next			×				
C: Users Renaud Documents Visual Studio 2008 Projects Fez V& LA Source Fez Game Components Player Actions.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);							
180 303 0.027 11.03 Physicshanager.update(Playerhanager);							
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 : xbWatson, PlX

- Come with the Xbox Development Kit
- xbWatson
  - Make your own measurements and output to debug
- PIX
  - Frame-by-frame teardown of what's costly
  - Excellent for inspecting the draw calls & shaders
- CPU Performance Profiler in the XDK ultimately not that useful
  - Made for native code, not .NET games

#### WHAT TOOK YOU SO LONG?

- Game started in April 2007
- Still in QA in November 2011



• What the hell is going on??



#### Timeline

- **2007** : Groundwork, engine, editor, "level one"
  - ... for the IGF'08 prototype, eventually scrapped (built in 100 days)
- **2008** : Celebration, setting up Polytron, art re-draw, engine work while finishing studies, more scrapped levels
- **2009** : Full-time work begins, world design, art & more engine work
- **2010** : Design dialogue with Microsoft, internal drama, DOT and world map come in, BMcC & Disasterpeace come in (October)
- **2011** : Puzzle & cutscene work, more levels, QA, optimization and polishing

#### That's all, folks!

Thanks for coming!

Questions?

