

[http://game.watch.impress.co.jp/docs/series/3dcg/20090613\\_283112.html](http://game.watch.impress.co.jp/docs/series/3dcg/20090613_283112.html)

西川善司为 3D 游戏粉丝的[MT Framework 2.0]的图形讲座  
CAPCOM 的次世代游戏引擎的真实一面！更快的把[失落星球 2]的高科技视觉公开！！

原文 西川善司



随着时代的更新，游戏开发规模变大，游戏平台的复杂性也增加了。为了对应这种变化，2004 年，从微软和索尼的次时代机的传言刚流出来的时候开始，先进的游戏工作室就进行了把游戏开发底层框架化的讨论。这里所谓的框架(Frame work)，是指把游戏程序的骨架部分做成通用程序库（common library）化的方案。

既然可以用[游戏引擎]的形式来归结这个潮流，那不用说大家都应该知道 EPIC GAME 的[Unreal Engine 3.0]。相比以原本的游戏工作室为单位，日本的游戏工作室更重视每个开发组的独立性和自主性，他们当初在接受这个潮流时，是有抵抗情绪的。

现在是 2009 年，虽然可以断言这时的日本和美国还没有技术差距，但即使这样，直到 2005 年和 06 年左右，对于[多核心 CPU (Multi-core CPU)][可编程着色器架构(Programmable Shader Architecture)]等新技术课题的基础研究，在很多游戏开发现场中，感觉都不是很充分。在这些技术面上，可以看到欧美阵营从一开始就决定冲刺了。

其中，CAPCOM 瞄准了这个时代的前端，进行了关于[游戏引擎][多核心 CPU][可编程着色器架构]的研究，这些的研究成果，就是 CAPCOM 引以自豪的[MT Framework]。这次，作为[生化危机 5]篇的特别追加，介绍[MT Framework]的最新情况。

---

#### ● MT Framework 2.0 也关注了向第三方做扩展

首先，我们向[生化危机 5]制作人竹内润先生询问了关于围绕 MT Framework 的战略



卡普空[生化危机 5]制作人竹内润先生



卡普空技术研究室程序员史田智史先生

竹内先生[如果用柔道战来比喻 MT Framework 的话，感觉是用先锋，中锋，大将的三阶段战术向成熟产品推进，先锋是[丧尸围城]和[失落星球]，中锋是[鬼泣 4]，大将是[生化危机 5]。使用这个战术，通过 MT Framework，在面向本世代机的游戏开发中获得了经验，也取得了成功。接下来，我们必须展开同样的战术，进一步挑战未来的次世代游戏开发。方法就是现在还在持续开发的 MT Framework2.0 (MTFW2.0)。通过各报道知道的人大概已经很多了。[MTFW2.0]的先锋战就是[失落星球 2]]

MT Framework，虽然没有对外通知，但在公司内用版本号为代号的形式来称呼。每个游戏开发项目中都在局部改良的层次对机能做强化，版本的编号就成了一号对一个游戏的形式，具体上就是下面的列表。

MTFW1.0---[丧尸围城]（2006 年）

MTFW1.1---[失落星球]（2006 年）

MTFW1.2---PC 版[失落星球]（2007 年）

MTFW1.3---[鬼泣 4]（2008 年）

MTFW1.4---[生化危机 5]（2009 年）

[MTFW1.x]，虽然在印象上是以 XBox360 和 PC 为主要目标来做出的基本设计，但感觉其对应到 PS3 的方法也越来越成熟了。这并不是“偏袒”，而是 CAPCOM 预研次世代开发技术后，进入实验研究阶段，存在的硬件首先是 PC，接着要做的开发环境是 XBox360，顺序有很大的影响。游戏机本体的发售也是这样，XBox360 是 2005 年底，PS3 是一年后的 2006 年底，如果考虑到这个发售顺序，现在的开发的顺序也是没办法的事。

随着时间的推进，对 XBox360、PS3，还有 PC 上，擅长和不擅长的地方也越来越了解，从把最初以 3 个平台作为目标的新构架游戏引擎（开发框架（Framework）），进行再设计。这就是 [MTFW2.0] 开发的开始。

竹内先生[我们也通过 [MYFW1] 的经验，对开发框架的重要性，以及它在游戏开发便利性上有了更深的理解。[MTFW2.0] 中，全部平台的通用性的进一步提高，每前进一步的战略都要讨论。那就是，我们 CAPCOM 的第三方也要使用 [MTFW2.0] 的战略。

根据竹内先生所说，[CAPCOM 通过 [MYFW1] 作为游戏的开发战略] 已经广泛的报道过了，而且进一步以 [MYFW1] 为基础的游戏也获得了成功，各公司询问关于 [MYFW1] 的事据说有很多。但是，[MYFW1] 是面向公司的内部引擎，因为完全没考虑到发给公司以外，作为 CAPCOM 人员的竹内先生也有很多痛苦的部分吧。

收到这种咨询后，并不是展开引擎商业化，而是针对开发能力高的游戏工作室，通过和 CAPCOM 合作关系，使用 [MTFW2.0] 的游戏开发的解决方案的讨论。

以具体的例子来说，CAPCOM 制作出版的游戏，或是 CAPCOM 人气系列的派生作品当中，也许会出现[虽然是 CAPCOM 品牌，但是由其他工作公司以 [MTFW2.0] 游戏引擎为基础开发] 的游戏。

顺便一说，万代南梦宫的游戏也有称为 [NU Library] 的优秀公司内制中间件，在人气游戏 [火影忍者终极风暴] 上也采用了。广泛的被第三方使用，现在，第三方利用大型工作室的内置引擎，本身已经不新奇了。但是，虽然当初卡普空一直说这 [MTFW1] 是只对应公司内制游戏的专用引擎，但从 [MTFW2.0] 开始战略似乎做了很大的改变，非常有兴趣的话题。

那么，最近次世代机的话题差不多一点点的开始了。在这个时期 [MTFW2.0] 的开发推进果然是要对应次世代机的完全改变架构吗？

竹内先生[要做出将来即使核心数量增加也可以灵活对应的设计。嗯，我对传闻的 Larrabee 也有兴趣（笑）]

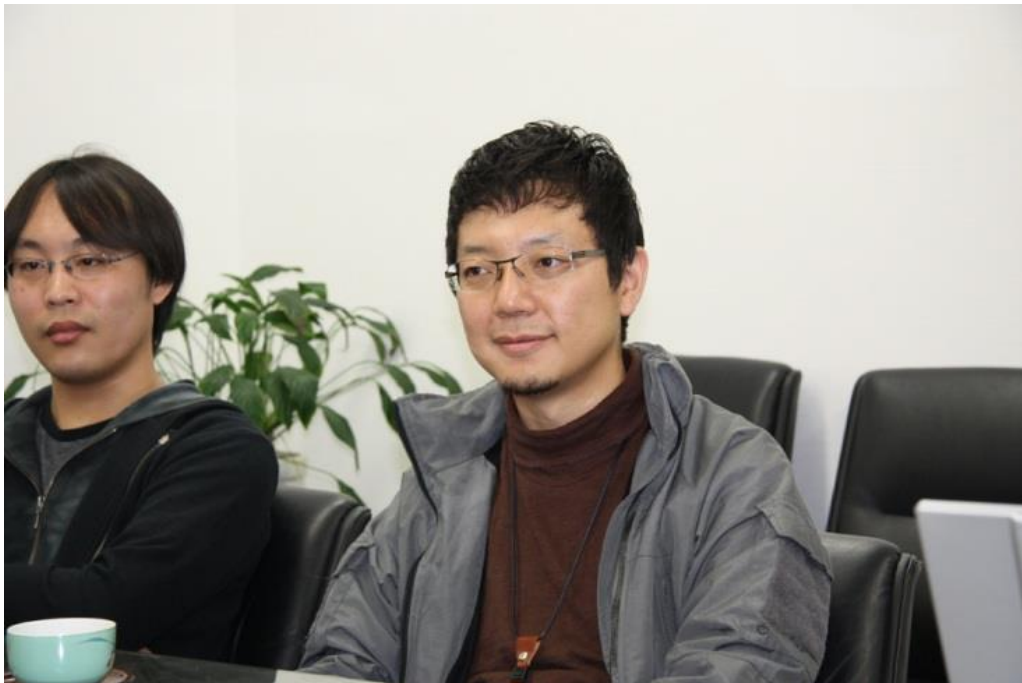
石田智史[是这样的，[MTFW2.0] 向 DirectX11 的对应就在积极的进行着。在已经发售的评估版 DirectX11 上已经 Build 完成了。是 Larrabee 吗？因为很早以前也写过软件渲染，

如果让我做，那真是享受啊（笑）]

【千里马肝注：Larrabee 是 Intel 已经取消研发的显卡，原预计在 2008 年第三季推出样本，在 2009 年正式发布，最终在 2010 年 5 月宣布中止。具体请见维基百科：[http://zh.wikipedia.org/wiki/Intel\\_Larrabee](http://zh.wikipedia.org/wiki/Intel_Larrabee)】

---

- 展望次世代机和次世代技术的[MTFW2.0]



公开发言说以后也会支持 PC 平台的竹内先生



卡普空技术研究室室长伊集院胜先生



在[生化危机 5]的取材中，笔者怎么没想到是从[MTFW2.0]的演示开始！真是幸运啊。

日本的游戏工作室，定期发售的 PC 版虽然有所减少，但 CAPCOM 的基于 MT Framework 的作品有比较快速发售 PC 版的倾向。这个流程也是继承了[MTFW2.0]吗？

竹内先生[不久前的 Larrabee 也是这样，游戏平台（技术方面）的最前端，现在是 PC 啊。为了在技术方面保持世界的最高端，这点是不能忽视的。关于 PC 平台的研究和 PC 版的提供，对[MTFW2.0]在战略层面都没有打算改变。但是，现在（2009 年 5 月）状况可以说时

机很微妙。在所谓的 Windows7 和 DirectX11 登场前，今年如果打算做出点什么就不能无视对它们的对应。这意味着我们开始具体的对应 PC 平台的公告是很难说出的（笑）]

根据竹内先生所说，关于 PC，难以解读今年以后的 PC 相关企业的动向。即使 [MTFW] 小组也对 GPGPU 的对应有兴趣，不过对于 CUDA，OpenCL(包括 ATI Stream)，DirectX Compute Shader 的情况，是各自对应，还是集中在一起，无法看出哪个会是主流，目前还没有决定。无论如何，[生化危机 5] 的 PC 版发售决定进行宣传。因为也表示了今后 CAPCOM 也要持续对应 PC 平台，所以 PC 用户暂时可以放心了吧。

有着[XBox360 和 PS3 之后]意味的[次世代机]，[MTFW]组是怎么关注的呢，很多的游戏开发者虽然希望是现在硬件的延长线上进化，但也不能否定像 PS2 到 PS3 那样做出激烈进化的可能性。

伊集院先生[次世代机吗？作为我们，感觉那是无论如何都要到来的事（笑）。与其说硬件没有急剧的进化，不如说以后的游戏形式不明确吧。如果用 Larrabee 作软件渲染，[就成了只有我们才能做出的画面了么...我感觉要积极的解决。]

竹内先生[我们作为开发一方，会注意硬件厂家的下一步。像开发环境的整備和新媒体的登场，很期待像 WII 那样有什么很厉害的变化。]

新硬件登场后，游戏开发者为了能活用硬件潜力，要反复的艰苦试验，会暂时迎来痛苦的时期。一旦这个痛苦时期结束，就会开始做出像是超越了硬件的潜力界限那样成熟的游戏了。回顾游戏史大概是反复这个模式没错的。

PS3 和 Xbox360，都是以 DirectX 9/10 为基础的本世代游戏平台，现在要进一步迎来成熟期，为了进行有优势有效的游戏开发，CAPCOM 的方法就是 [MTFW2.0] 吧。

而且很有意思的是，只要听到前面 MTFW 组的发言，就不会害怕下一代主机登场了。关于从 2005 年到 2006 年出现在本世代机上的激烈的技术革新的局面，虽然日本的开发现场晚于海外，不过 CAPCOM 的 MTFW 组也面对下一步进行秘密策略，用很自信的姿态表示[次世代机的技术革新啊，总算来了太好了]。至少在现在的本世代机上，MTFW 可以看作日本游戏开发场面的技术样本，而且还在世界战略意义上成了基准的存在。

---

- 面对[MTFW2.0]，回顾[生化危机 5]的开发





卡普空第二图形制作室设计师平林良章先生

被放入[MTFW2.0]的理念大体上清楚了，那么具体的看看[MTFW2.0]的样子吧。

首先，关于在[生化危机 5]中暂时告一段落的[MTFW1]项目，在[生化危机 5]的开发结束后，可以看到的课题，我想从这些话题开始。

石田先生[技术交流等事后的技术会议中有很好的话题，不过现在的主流着色器(Shader)管理方式，Shader 的变种过于庞大。现在[MTFW1]中，管理方式也是以[Shader Pipeline]为单位。把整个 Shader 作为一个管理单位，例如 Vertex Shader 的 skinning 设定了 4 个 weight，光源和法线贴图也是这样，按感觉来设计出来了。因此，只要改变光源的数量和法线贴图的指定，就成了其他 Shader，变化就增加了。在[生化危机 5]中，Shader 变化超过了 10 万种。感觉这些是无论如何都不能做到的(笑)]

结果，在[生化危机 5]中，只在 Shader 上就占有 16MB 左右的内存。还有，现在虽然是运用静态分支命令，指定 Shader 内的光源，控制了变种的增加，但因为在 PS3 的 GPU 的 RSX 上没有静态分支，PS3 版本上要全部作为变种来实现，使得 PS3 版上的 Shader 达到了 50 万种以上。

PS3 的 RSX 上，有动态形的分歧命令，活用这个会抑制变化的增加，在实际的尝试后，这个方法速度太慢无法使用。

关于[只要光源数不同，Shade 变化就会增加]的说法是在最近听到的问题点，解除这些的方法，也请注意在本连载的[杀戮地带 2]篇中介绍的 Deferred Shading。

石田先生[Deferred Shading 和 Xbox360 的匹配很差。Xbox360 的 GPU 上虽然有 10MB 的 EDRAW，但 720P 的分辨率做 MRT 的话，因为无法放入 10MB 里，只能使用 Tiled rendering，效率就变差了。虽然也有只把 Normal 和 Specular 写入 buffer 里，做出只有 Lighting 的 Deferred Shading : Light Pre-Pass Rendering，但这些在现在的实现中要符合状况的转换]

平林先生[还有，这个 Deferred Shading 的方法，和半透明物体的兼容性很差（笔者注：关于半透明处理需要另外渲染再做 post process 合成的方法）。CAPCOM 没有半透明吗，是因为不能考虑（笑）。要想取得同为半透明对象的 priority（前后关系），无论如何都会有的。我想其他的日本工作室也是这样，这个手法说不定在日本难以流行。

石田先生[Deferred Shading，能使用的材质表现也是有限制的啊。用 ID 管理的方法虽然也被考虑，但这样做时，动态分支就成了不可缺少的了。在 PC 上虽然很好，但在 PS3 上感觉变的恶梦一样（笑）。像 PS3 的 PHYRE ENGINE 那样，要是用 SPU 做出 Lighting 也很好吧。不论是 XBox360 还是 PS3 或 PC，感觉都很难采用。

[Trace 注:在 SIGGRAPH2011 上，EA 的 Digital Illusions CE 介绍了[战地 3]在 XBOX360 上的渲染方案：Tiled-Based Deferred Shading（把屏幕分为若干个 Tile, 来判断那些被光源影响，只有被光影响的像素才会做处理，从而减少了带宽），解决了因为只有 10MB EDRAM 无法用 MRT 的来做 Deferred Shading 的问题]

## How Does This Fit on Xbox 360?



- We don't have DirectCompute nor SPUs on 360...
- Fortunately, *Xenos* is powerful, and will crunch ALU
- For maximal throughput, data at rendering time has to be cleverly pre-digested
- If timed properly, we can also use the CPUs to help the GPU along the way...
- GPU is better at analyzing a scene than CPUs...
- Let's use it to classify the scene



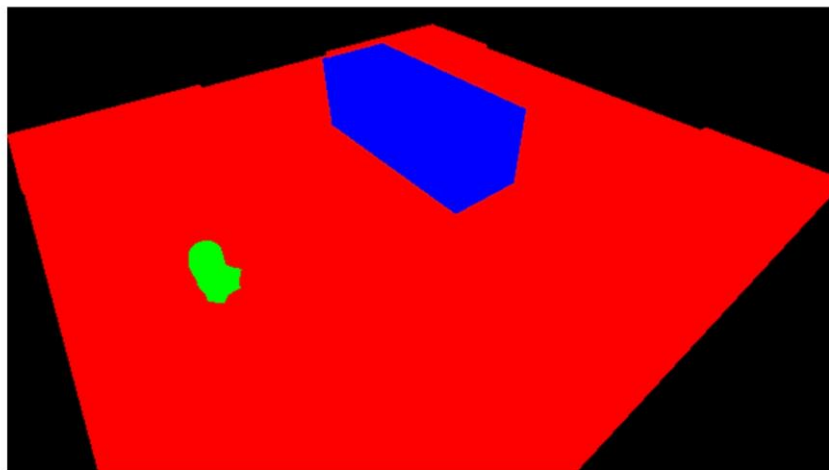
## GPGPU Culling (1/)



- Our screen is divided in 920 tiles of 32x32 pixels
- Downsample and classify the scene from 720p to 40x23 (1 pixel == 1 tile)
  - Find each tile's Min/Max depth
  - Find each tile's material permutations
  - Downsampling is done in multi-pass and via MRTs
- Similar to [Hutchinson10]

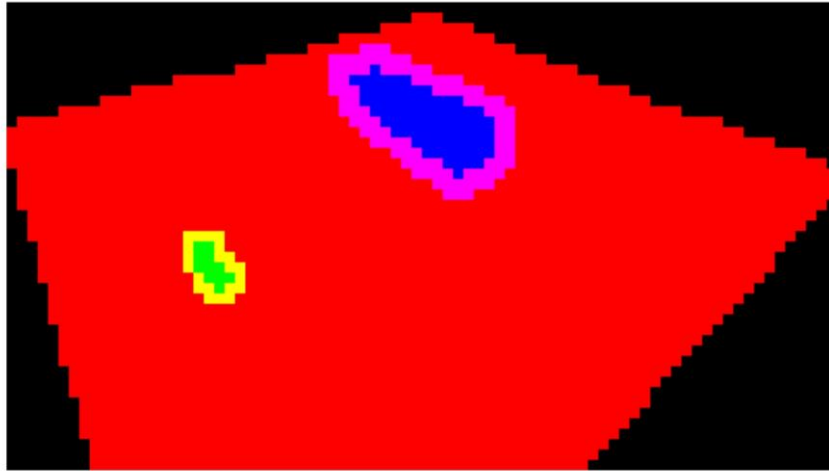
*Advances in Real-Time Rendering in Games*

## GPGPU Culling (2/)



*Advances in Real-Time Rendering in Games*

## GPGPU Culling (3/)



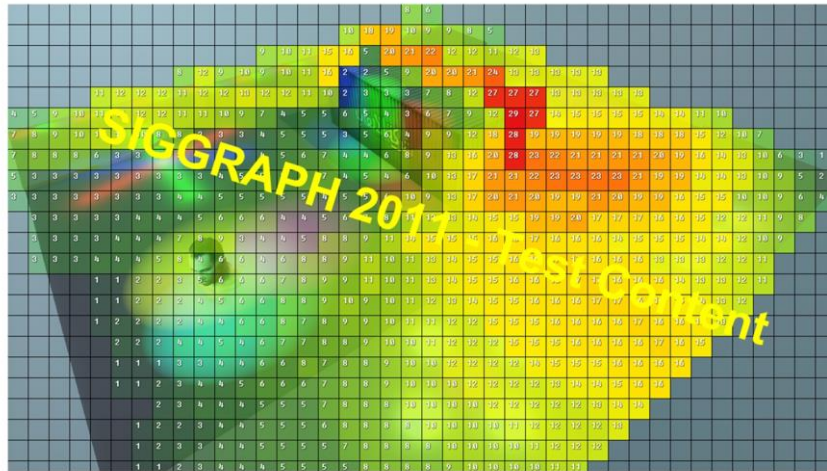
*Advances in Real-Time Rendering in Games*

## GPGPU Culling (4/)



*Advances in Real-Time Rendering in Games*

## Tiled-based Deferred Shading? (5/)



Advances in Real-Time Rendering in Games

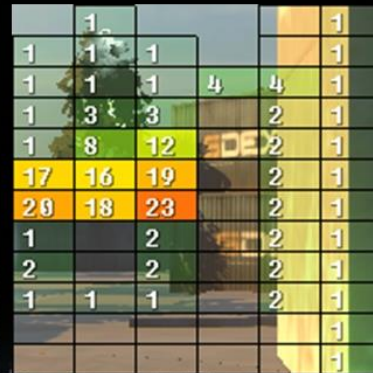
## Tile-based Deferred Shading

1. Divide screen into tiles and determine which lights affects which tiles

2. Only apply the visible light sources on pixels

- › Custom shader with multiple lights
- › Reduced bandwidth & setup cost

How can we do this best in DX11?



在 PS3 上，也许 Deferred Shading 是最合适的，把多平台展开作为前提的游戏前提，并把半透明看的比其他都重要的 CAPCOM，没有理由[在 MTFW 上立即采用 Deferred Shading]

### ● [MTFW2.0]上新的 Shader 系统[Meta Shader System]是什么？

用了怎样的形式在[MTFW1]的基础上进行「MTFW2.0」的开发的呢？虽然简单来说应该是按着升级和「机能强化」的关键词走到现在，但实际的情况是怎样的呢？

伊集院先生[MTFW2.0，几乎是从最初开始完全重新构筑的东西，因为 MTFW1 如论怎样做，



最初的硬件都是 XBox360,所以基本设计不能否定是面向 XBox360 的。PS3 出现后走向了成熟,正是现在这个时期,为了要最大限度的活用全部的硬件而重新思考设计。]

MTFW 开发组从 2008 年 1 月左右之后开始了开发,开发初期做出连 3D 模型都不能显示的文字和窗口,几乎从零开始的项目启动了。初期开发成员是 2 个人。石田先生从 2003 年 PS2 的[鬼武者 3]的开发时代开始,就是一直从事渲染引擎设计行业的专家。在 MTFW 项目中担任的工作是领导基本设计的架构师。

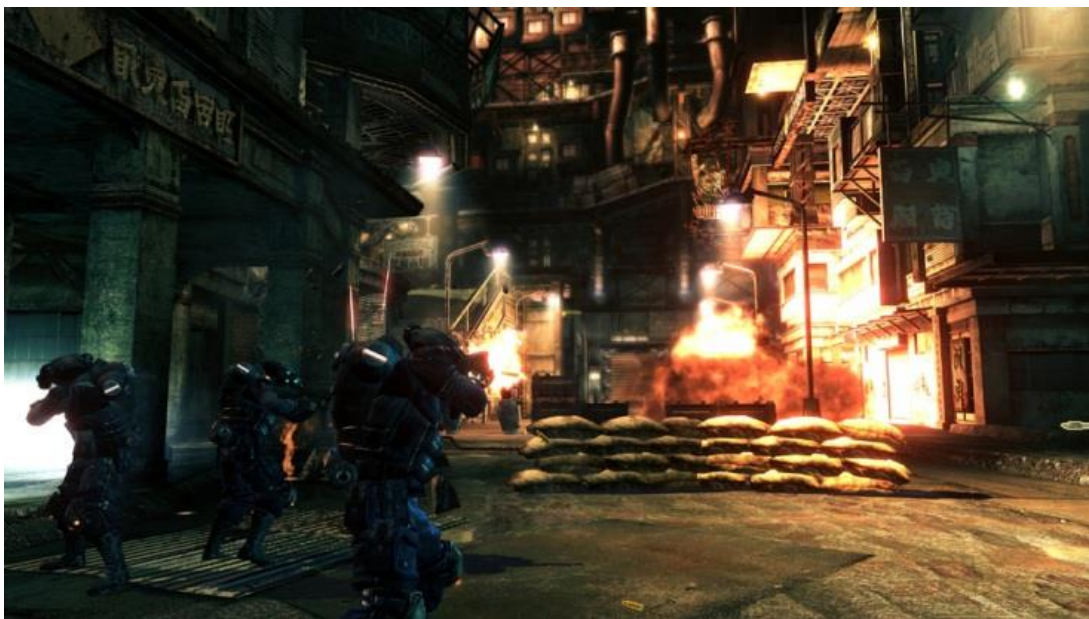
#### [MTFW2.0]支持的 HDR 渲染的变化]



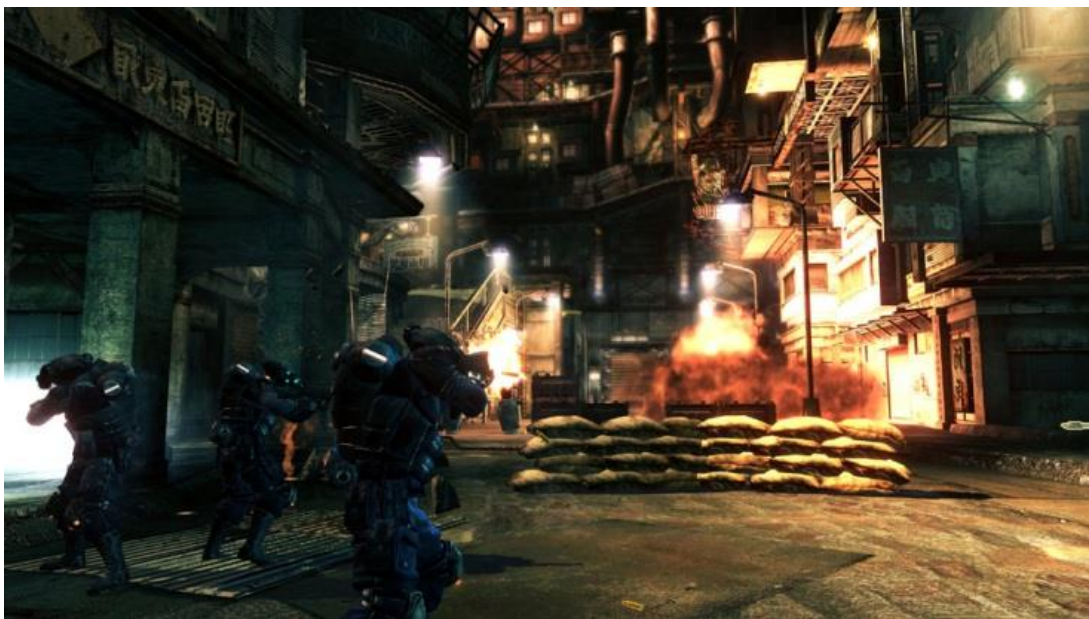
没有 HDR



和[女神侧身像 2]同样手法的范围压缩模拟



由于浮点数 buffer 得来的正确 HDR



保持色彩风格的同时施加范围压缩的模拟 HDR





非线性 Toon Mapping



和[MGS4]同样由 Blend buffer 得来的合成图例



在 MTFW 项目中担任着领导基本设计的架构师的石田先生。

根据石田先生所说，[MTFW2.0]的渲染引擎，是计划着从零开始重新制作的。要是只从表面的产品规格来说。迄今为止的[MTFW1]中，虽然对应了 XBox360 和 PS3 以及 DirectX9(SM3.0: ShaderModel3.0) 和 DirectX10 (SM4.0: ShaderModel4.0)，但在[MTFW2.0]中，只进行了到 DirectX11 (SM5.0: ShaderModel5.0) 的支持，感觉没有很大的变化。

但是，并不是那样。实际上，在内部的大的架构上谋求了革新。[MTFW2.0]中，有着新开发的独立的统一 Shader 管理系统。各世代不同 GPU 配置的差异，ShaderModel 的差异，都可以进行处理。

虽然 PS3, XBox360 等家用机有着同样 SM3.0 世代的 GPU，但在这之外的部分，有着这些机种专业的 GPU 命令，DirectX9 和 DirectX10 中 SM3.0 和 4.0 存在着差异。前述的 Shader 变种的问题还有，游戏项目的程序员一旦打算写出实现某种表现的 Shader，为了得到最大的性能，有必要全部使用 PS3, XBOX360, DirectX9, DirectX10, DirectX11 的最合适指令。Framework 的目标是让程序员那边的负荷降低，可以说是作为以游戏引擎为基础的游戏开发必须克服的课题。

石田先生[为了应对这个问题，游戏程序员一方就是用最高级的 ShaderModel 进行 shader 设计就可以了。[MTFW2.0]的 Shader 系统的预处理实现了，在分析 Shader 的程序代码后，使用对应平台的专用命令来做最合适替换的最优化的结构。具体来说，程序员要是写出 DirectX10.1 的 SM4.1 Shader 代码就可以了，在向下级世代的平台输出代码时，[MTFW2.0]会替换成哪个平台 GPU 的专用命令。也就是说，用这个 shader 可以得到最大性能的形态，并向低级的 Shader Model 进行转换。也要让这个配置对应将来 DirectX11 的 SM5.0。]

程序员即使没有对每个目标平台做特殊优化技术的知识，也可以得到最大性能的 Shader

[illegible]

作为一个没有依赖平台的“方言” Shader 系统，也有着 EPIC GAMES 的[引擎 3 (UE3)]



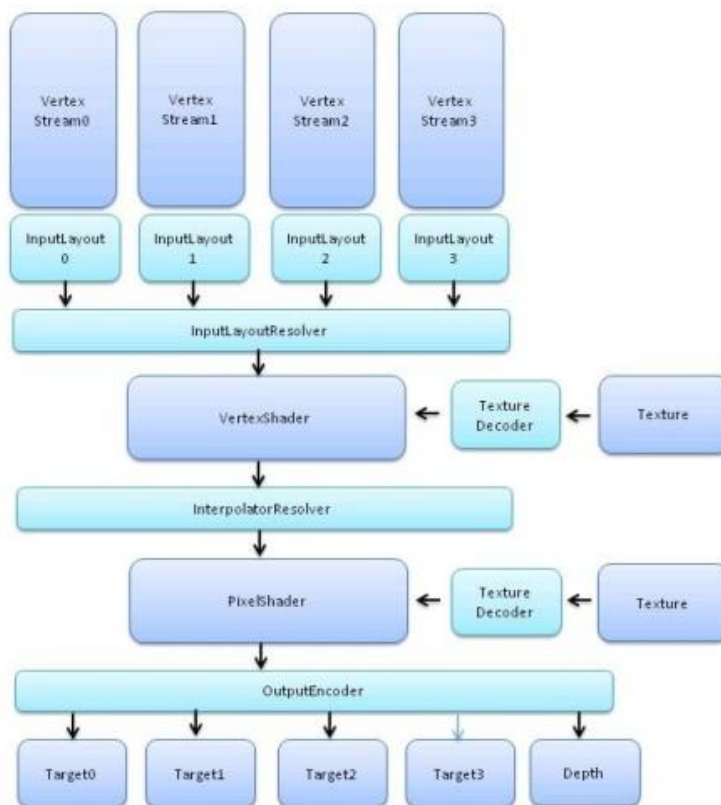
那样的材质工具。[UE3]用的是把每个平台中最优化的 Shader 效果的部件组合后，来设计想表现的材质 Shader 的方法。还有[UE3]也是以艺术家和设计师也可以设计 shader 这点来吸引买家的。

石田先生[不能像[UE3]的材质编辑器那样随意的把部件替换。[MTFW2.0]的 meta shader 系统，处理的对象都要是 shader 的源代码。日本的设计师和艺术家不太想做 shader 设计，相反，（日本的）程序员虽然没有那种零件为基础的 shader 设计工具，却很希望用源代码来写 shader（笑）。[MTFW2.0]的 meta shader 系统是把程序员的自由度作为最优先来制作的。

[MTFW2.0]的 meta shader 系统中，首先，基本的 shader 设计是在源代码等级由程序员进行的。这个 shader 的流程本身，即使在[MTFW2.0]的 meta shader 系统的 GUI 中，也不能改变。

但是，[MTFW2.0]的 shader 系统有着，解析 HLSL(High Level Shader Language) 基本的 shader 源代码，分析出程序中函数的依存关系，把各函数动态的替换成其他函数的结构。也就是说，因为替换了函数，就可以完成做成其他表现的 shader。

[meta shader 系统的结构图]



[MTFW2.0]中的 meta shader 系统的结构图]。系统中，对应自动生成的是图中淡蓝色的部分。







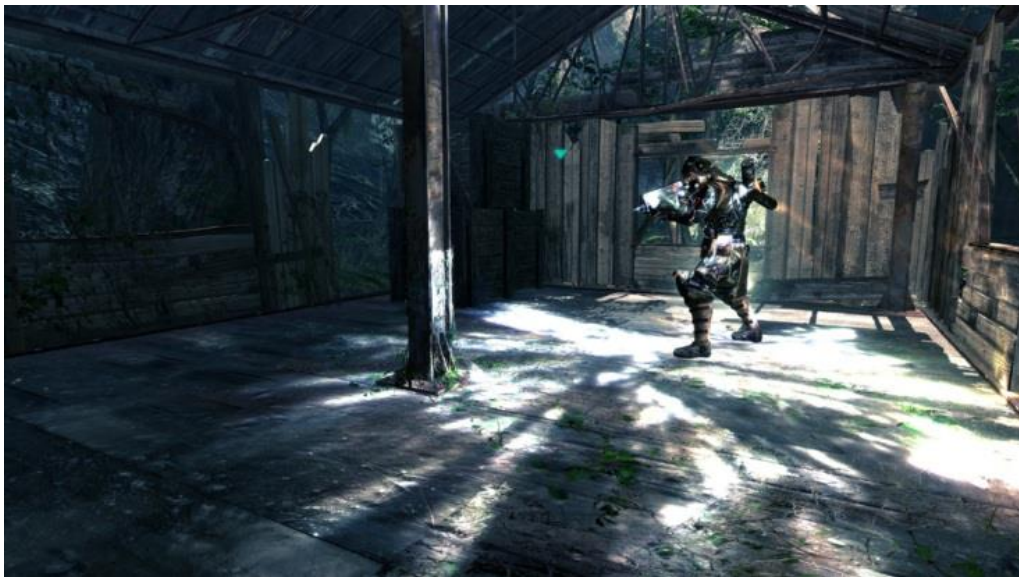
Blinn-Phong Shader



Skin Shader



Toon Shader



由于[MTFW2.0]的光源被做成一般化,如果把 Point light 替换成 Cubemap 的话,就可以实现全方面 Spot light 一样的效果。

还有,在这个 meta shader 系统中,要把 Vertex Shader 和 Pixel Shader 在全局解析后实现最优化的结构。迄今为止的 shader 编译器的最优化是要在 Vertex Shader 和 Pixel Shader 各自进行最优化。在 Pixel Shader 里没有必要的常量和变量参数,对应 Vertex Shader 里计算输出部分的代码要做处理,但这个系统中不会有这种的问题。

伊集院先生[由于新的 meta shader 系统,shader 的变种被控制在前作 1/10 的程度。容量上也控制在 3MB 字节左右。Shader 以区域为单位做阶层管理,作出了这个区域不需要的 Shader 就不读入的设计。

-----

- [MTFW2.0]上实现了物理模拟的内制化

关于在[MTFW1]上作为课题的物理模拟，在[MTFW2.0]上从基本的部分，缓慢的通过内制进行替换。

伊集院先生[不论哪个游戏，就像是必要的一样，最基本的刚体物理，布料物理，布娃娃等物理模拟在[MTFW2.0]上做成内制化。但是 HAVOK 为首的物理模拟中间件提供的高级物理模拟，具体上说有破坏系统，流体物理模拟等，还没做到内置。应用的弹簧物理，柔体物理模拟等已经实现了内制化。]

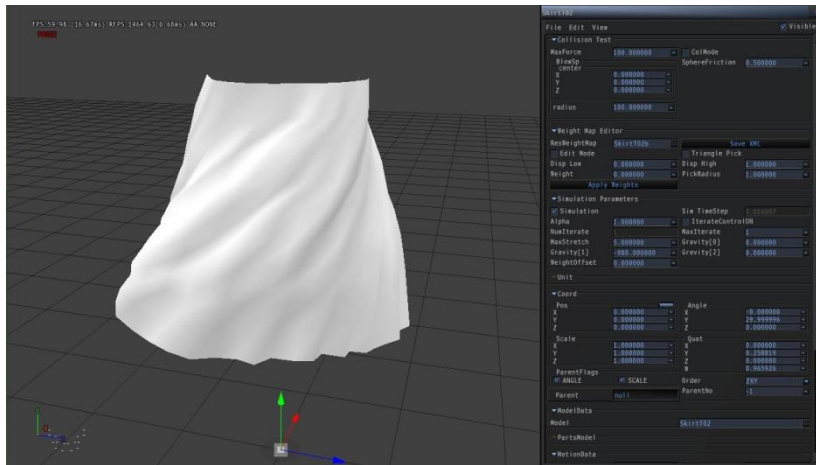
[在 MTFW2.0 上内制实现的各种物理模拟]



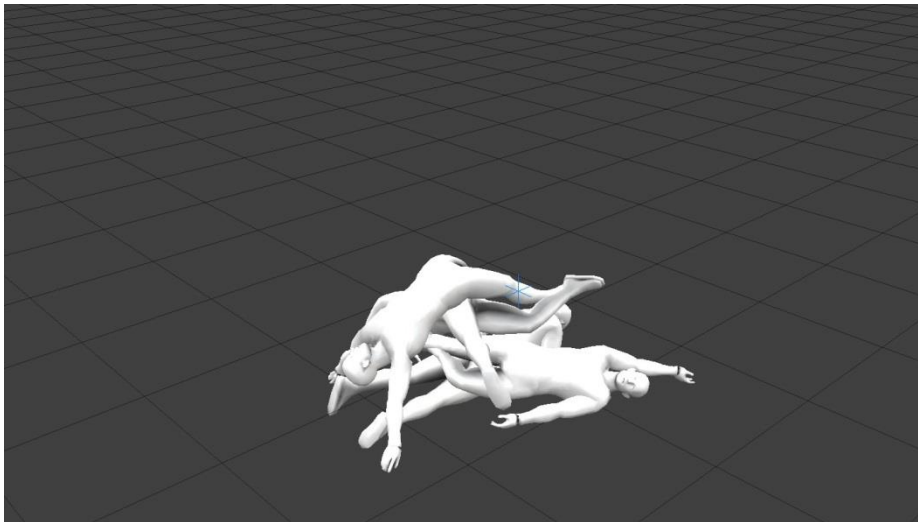
在[MTFW2.0]上的刚体（rigid body）物理模拟 DEMO



在[MTFW2.0]上的柔体（soft body）物理模拟 DEMO



在[MTFW2.0]上的布物理模拟 DEMO



在[MTFW2.0]上的布娃娃（Ragdo11） DEMO



解释技术革新关系到有效率的支持现场设计师工作的平林先生





担任[MTFW]项目指挥的伊集院先生

柔体物理在[UE3]上的实现，也是能称为话题的表现。是有着柔软的弹性 3D 物理的表现，这个一般作为生物的表皮使用。目前在[MTFW2.0]中实现的是，用 CPU 进行表皮的变形，skinning 操作时表皮上的顶点法线的再计算是用 GPU 实现的，计划最终全部用 GPU 来实现。

【千里马肝注：因为物理是在 CPU 上计算的，所以 skinning 就必须回归 CPU 计算，才能方便传入物理引擎以产生正确的碰撞。】

平林先生[从我们设计师的立场来看，这样做技术革新，不仅仅是唤起新的表现，工作效率也提高了。例如，要是把柔体物理用[MTFW1]来制作，必须加入大量的骨骼，这个柔体物理被加入到引擎，就会大量的减少骨骼数量，这样对于运行时的数据量也有好处。

现在实验性的挑战基于刚体物理的破坏系统的实现。不过无论如何，现在都是要把[今后 CAPCOM 的游戏中必要的物理模拟要自己制作]为重点，相反的，如果什么都要固执的自制，在自制过程中消耗的时间和人力成本，要考虑价值的平衡。例如，优秀的性能和各种要素一起做成高质量的表现，在需要自制不能支持的机能时，就要像以前一样把其他公司的物理模拟中间件加入，这个判断与其说是[MTFW]小组，不如说是在游戏的项目组中进行的。实际上[失落行星 2]的刚体模拟在考虑各种要素后使用了 HAVOK。

伊集院先生[生化危机 5 虽然是用 MTFW1 做的游戏，但破坏系统上使用了 HAVOK。要是说起生化危机 5，有游戏逻辑内置的碰撞判定和 HAVOK 管理的破坏系统的碰撞判定，他们都作为独立系统来运作。从数据的角度来说，内存里装载两份类似的破坏碰撞用形状模型是浪费了。如果能自己制作统一起来，就可以把浪费节省了。物理模拟内制化的出发点首先就是那里啊!]

- 
- [MTFW2.0]上的第一个游戏[失落星球 2]怎样了？



四月时闪电性的公开了预告影像[失落星球 2] (LP2)，作为[MTFW2.0]的第一弹作品来吸引购买者。实际上能让我们看到怎样的视觉体验呢？

[LP2]在开发初期，是用[MTFW1]来开发的，在开发途中改换成以[MTFW2.0]为基础。在这个过程中，[LP2]中的图形表现上的材质，以[MTFW1]为基础的设计几乎全部结束了。

因此，在[LP2]里，在[MTFW2.0]采用的方针是使用可以运行「MTFW1」材质的结构。[LP2]上虽然到处都有值得一看的地方，但在技术上非常有趣味的是比[生化危机 5]更先进的活动角色与植物的交互。

因为[LP2]的时间设定是前作[失落星球] (LP)之后，所以[LP1]的舞台[极寒地带]EDN-3<sup>rd</sup>设定成进入了温暖化，到处都生长着植物。

这些植物，在[LP2]中会由于枪击和轰炸而产生的爆炸引起的风而摇动着。当然，活动角色和植物也有碰撞，作为背景物体的植物看起来会非常活灵活现。

植物模型在设计阶段被设计成固定模型，为了驱动，开发了解析模型构造，并把植物“手臂”信息按顺序生成结构。结果，植物可以实时的和游戏内的场景中的活动角色柔软的交互了。就想CRYTEK的CRY ENGINE3一样，实现出通过枪击，树木树枝从中弹地点折断等表现。

[在[失落星球 2]中和竖立的植物产生的交互]



通过软体做出的树（PC 和 XBox360 是用 GPGPU，PS3 是用 SPU 驱动）



草的表现



程序生成的背景 weight map（要表现出受到某种程度的风的影响）



草木的最终合成场景

石田先生[现状是因为把 XBOX360 版作为预想来制作，植物的影生成只对主要部分进行生成，但在引擎上也能把生长在地面上的杂草一根根全部对应的投射影子。也许在 PC 版上有设置这种选项的自由度吧]

还有，倒下的树木在撞到地面时，不是用刚体物理而是应用了柔体物理。倒下的树枝由于树叶的缓冲效果，绘制出了弹簧那样一下弹起的效果。

在[LP2]中，像[汪达和巨像]那样，跑到巨大的 BOSS 上面进行直接攻击的动态战斗表现也被导入，不过，这时在 BOSS 背上的表皮和褶皱表现上，采用了新开发的柔体物理表现。这次在[MTFW2.0]上的物理模拟，就是理解自制化好处的最佳成果。



体验到[LP2]的 BOSS 角色的后背的柔软感！





由于全方位 Variance Shadow Map ，使[LP2]场景中影的效果很有印象。把点光源生成的动态影子在全方位生成，还要加入 Soft Shadow 化的处理。

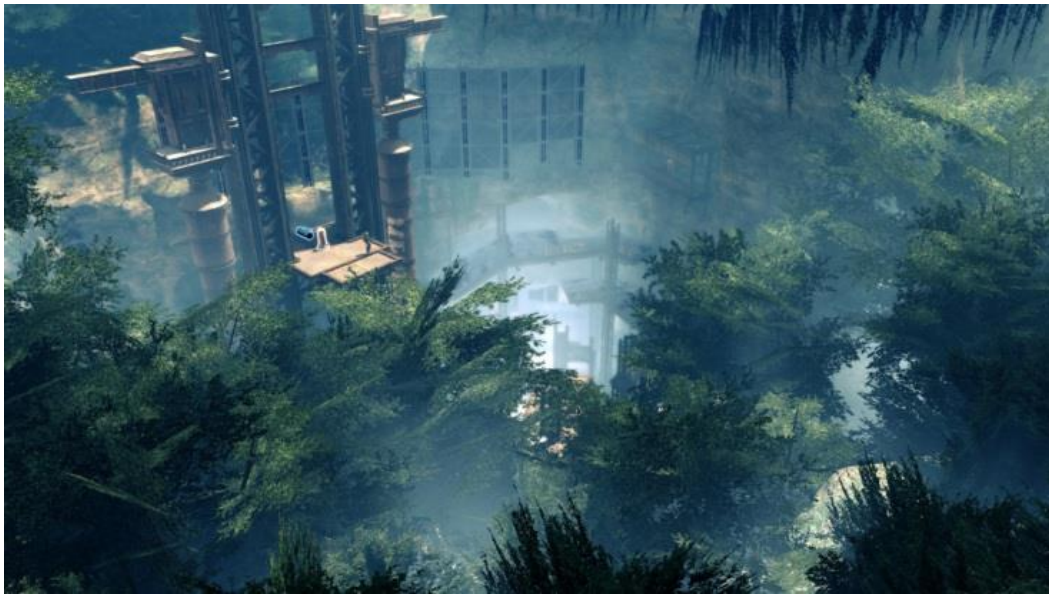
今后，在图形表现方面，有考虑导入到[MTFW2.0]或者是[LP2]里面的技术吗。

石田先生[现在不管怎样，都要力图把基本部分充实化，但 HDR 的 DOF 模拟等都在讨论中。之后屏幕空间效果和动态 Ambient Occlusion 等，都是很有趣的试验。]

关于石田先生的发言内容做个补充说明。。一般的 HDR（High Dynamic Range）渲染只在高亮度部分产生 Bloom 的表现一直被议论着。但实际的 HDR 渲染上，隐藏着更高照片真实感的可能性。拍摄照片时，焦点不一致的部分如果包含着高亮度的部分，那么就会出现在镜头光圈形状的斑点（Bokeh）。做出这样表现，需要在 HDR 渲染结果里，加入摄像机和镜头结构的模型化的后期处理。只有高亮度的模糊表现是不能表现 HDR 的。石田先生应该是想做这个。

[Trace 注：UE3 和 CE3 都已经实现了 Bokeh DOF 这个特性]





[LP2]的一个场景。雾效果是由[MTFW2.0]的 volumetric height fog 做出的。考虑了从视线方向的空气层的线性厚度生成的雾。





[结果，实现的图像]（石田先生）最后说。[失落星球 2]的测试图像。基于摄像机模型的 DOF 表现。确实出现了六角形的摄像机光圈形状的光斑。

所谓的动态 Ambient Occlusion，是在把全局照明表现在屏幕空间系的后期处理上实现的技术，别名被称为[Screen Space Global Illumination]（SSGI）的技术。基本的概念和在本连载的[Gears of War 2]篇中介绍的 Screen Space Ambient Occlusion（SSAO）非常相似。SSAO 是注重检查像素周围的，计算出遮挡度来添加阴影，而 SSGI 是注重像素周围的颜色，反映在阴影处理的技术。根据这个可以做出简易型的互相反射的表现。据说 CRY ENGINE3 也是采用了这个 SSGI。

[LP2]因为还在专心开发中，绝对会让你看到包括这样做出的新表现和还没看到过的视觉。从现在开始期待发售吧。

[Screen Space Global Illumination]（SSGI）



通常的渲染结果



SSAO 应用状态。脚下的淡影，墙壁和地面交叉点附近的“阴”是由 SSAO 做出的。

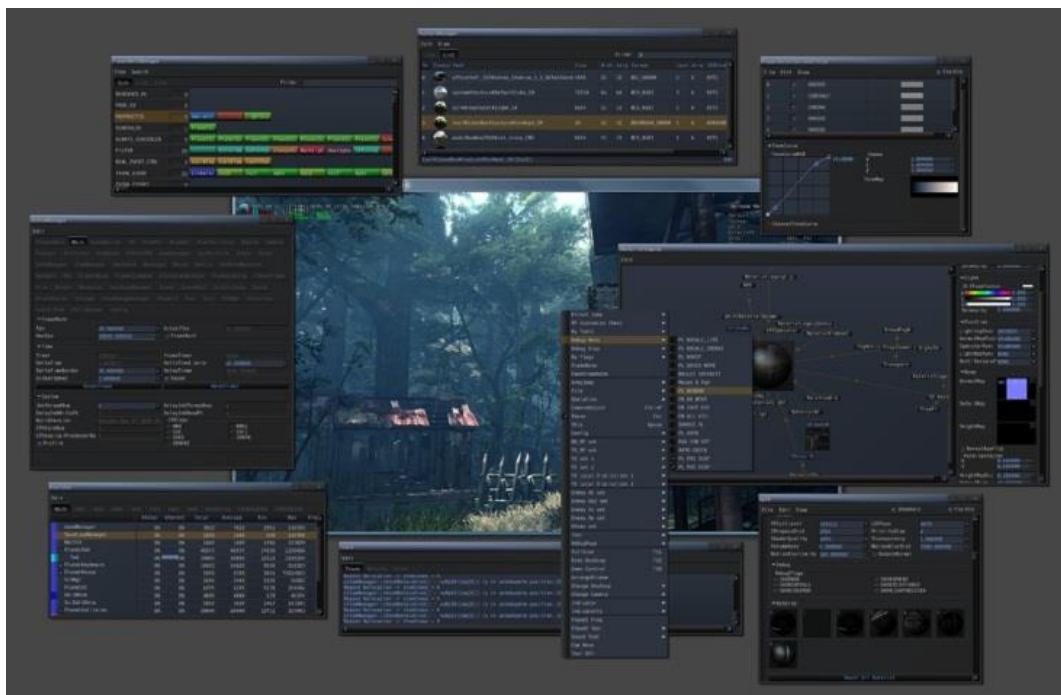
[译者注：在日本把“阴”和“影”分开解释，阴是自遮蔽产生，影是由其他物体投射产生的]



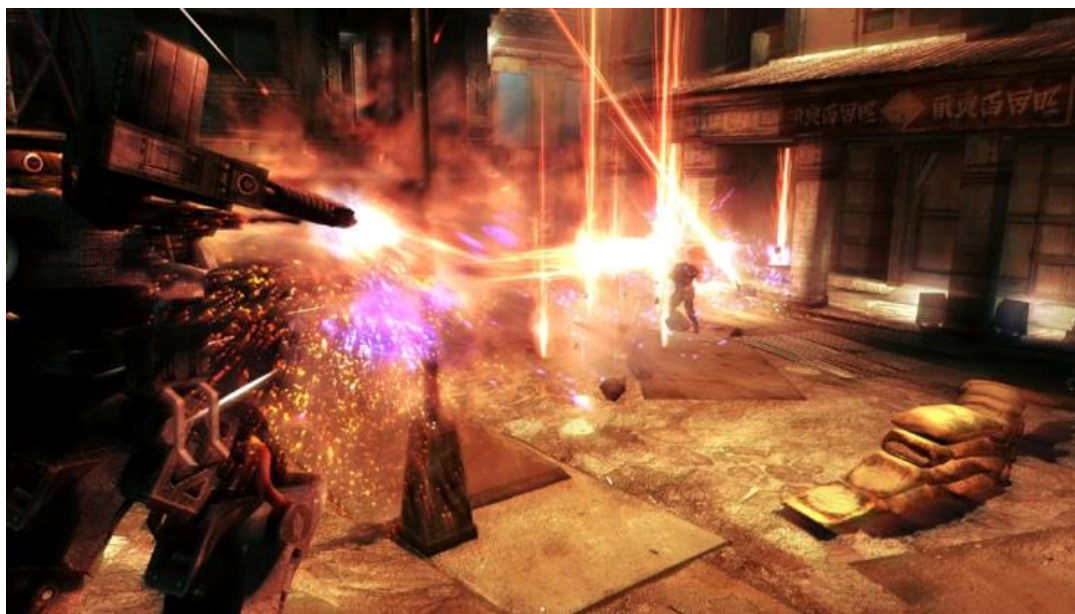
结果，在采访的最后，SSGI 也在[MTFW2.0]上实现。里面的梯子，因为从墙壁的反射的光被照亮了吧。这就是疑似性再现互交反射或二次光源照明的 SSGI 效果。

- 
- 结尾——在次世代机登场前，CAPCOM 游戏引擎的次世代化





一张[MTFW2.0]的开发画面图例。各窗口出现在了主窗口外面。



[失落星球 2]上半透明合成处理的最终画面

石田先生也那么说，这次的[MTFW2.0]的主题是的[很高的自由度]。

[用同一个游戏引擎制作的影像也会相似]虽然经常被提到，但在[MTFW2.0]上，程序员因为能够自由的追加也可以说成是着色器部件的函数，所以不会有那样的担心。CAPCOM 自制的游戏项目就不用说，就连和 CAPCOM 关联的游戏工作室，要是使用了[MTFW2.0]，各工作室做出的函数都会共享。[MTFW2.0]的图形表现绝对会做成多样化。

伊集院先生[本世代也迎来了成熟期，感觉今后的游戏引擎只有[看着外观很厉害]是不

行的。感觉是到了一个不能只靠厉害机能，内存占用量小，性能快和新奇感的时代了。在 [MTFW2.0] 中，我们为了高机能和容易处理而努力着。在今后，要关注 CAPCOM 关联的工作室也使用到 [MTFW2.0] 这点，变的要比 [MTFW1] 更重要。关于 [迷失行星 2] 游戏的有趣度就不用说了。不过我们为了能达到用户期待以上的影像表现而继续努力。也请继续期待 [失落星球 2] 的发售，还有 [MTFW2.0] 今后的进化。

在 [失落星球 2] 上的半透明合成处理的结构



Blend buffer RGB



Blend buffer alpha



缩小 blend buffer RGB



缩小 blend buffer alpha



