

http://game.watch.impress.co.jp/docs/series/3dcg/20120302_515505.html

西川善司为了 3D 游戏粉丝的[Child of Eden]讲座（后篇）
通过和[BISHAMON]结合来加速的“超越界限”的特效表现。



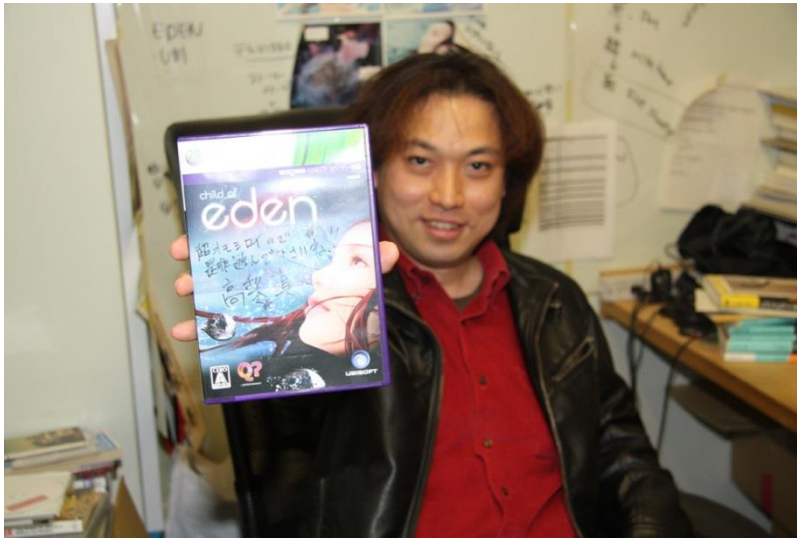
会场 Q ENTERTAINMENT 本社



水口哲也先生（Q ENTERTAINMENT COO）

水口哲也先生带领着 Q ENTERTAINMEN 亲手制作了 [太空频道 5] 系列和[REZ]还有[LUMINES] 系列等多个把影像和音乐的同步作为主题的崭新的名作。这次来解说他所带领的 Q ENTERTAINMENT 最新作[Child of Eden]的图形技术。

还有，虽然还不太被知晓，但[Child of Eden] 精心利用了本连载今年第一期介绍的 3D 特效插件[BISHAMON]的作品。因为本连载今年的主题是[游戏开发和中间件]，所以这次，也决定涉足以[实际的游戏开发现场关联的中间件]视点的话题



[著者近影]

[Child of Eden]因为据说彻底利用了特效中间件[BISHAMON]的游戏。所以这次也询问了很多这方面的话题。照片是在采访[Child of Eden]时，我借机坐在水口哲也先生工作现场的豪华椅子上拍摄的照片（笑）。

[Child of Eden]特有的图形表现的秘密 1—— 由 MIDI 控制运动的图形!?

[Child of Eden]的图形中，虽然非常独特，但各物体，并不只是规则的角色动作，游戏世界和素材的阴影变化，都是与游戏中乐曲（BGM）和声音效果同步的部分。这种表现在一部分音乐可视化和 megademo 里多次看到的，但在商用游戏题材中是非常少见的表现手法。

电脑游戏的大主题应该就是声音和影像，还有根据玩家的输入相互作用交织起来联姻，[Child of Eden]的这样的表现手法，感觉有某种[那不就是作为电脑游戏的一个正确的形态吗]的感觉。这个范围在技术方面上是用怎样的结构来实装的呢。

因为电脑游戏是以声音和影像，还有根据用户的输入互动交织起来的结合为大主题的。[Child of Eden]的这样的表现手法，有某种[那不是作为电脑游戏的一个正确形态吗]的感觉。这个范围在技术方面上是用怎样的结构来实现的呢？



高梨真先生（Q ENTERTAINMEN，平面设计部门主管）

高梨真先生「这样的影像，声音，还有玩家的输入，创造出的联觉（synesthesia 解释见上篇），正是基于水口所追求的[联觉理论]。这样的表现，用我们特有的架构实现了啊」

游戏中采用的乐曲，是由水口哲也先生亲自参与的音乐团队元气火箭（Genki Rockets）来制作，在音乐制作阶段乐曲的 MIDI 数据，要加入影响游戏内登场的背景物体和敌人角色动画的事件数据。

以前就认识笔者的人应该知道（笑），实际上因为 MIDI（Musical Instrument Digital Interface）也是笔者的专门领域，所以详细询问了，使用 MIDI 的控制变化（Control Change）作为给与图形影响的事件数据。

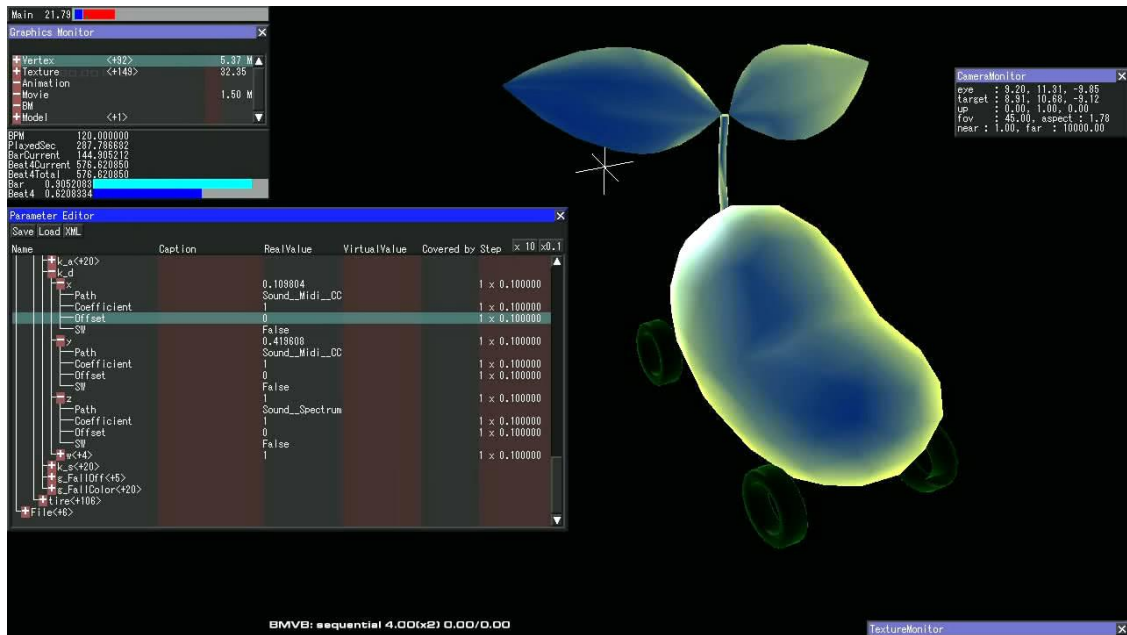
MIDI 的控制变化有全部 0~127，一共有 128 个（控制编号），把这个用 0~127（或者 0~16383）数据字节的方式来指定，如果假设无视 MIDI 规格的用途来使用，是把 128 种事件数据和 0~127（或者 0~16383）范围的数值参数嵌入到乐曲中的配置。

总之，是在预先的设计阶段，进行决定控制编号采用哪个图形表现参数的配置，再由图形小组和声音小组相互配合做出音乐和图形流程。

在设计小组一方，由乐曲方的控制变化对应图形表现的各个事件的设置，是由公司内部专用开发工具来进行的。还有通过控制变化的参数，来对几何体动画进行控制的利用。这个利用了背景整体乐曲的特定乐器和节拍同步，看起来像是旋转交织一样的活动效果。

厉害的是，通过控制变化的参数，直接加入到 Pixel Shader 的参数来使用，这个应用，可以让阴影自身的变化配合和乐曲配合，或者是让纹理动画与节拍或特定乐器同步。

[把 MIDI 参数分配给着 Shader 参数的工具画面]



乐曲的进行可以和动画或阴影表现同步的[Child of Eden]图形系统。

高梨先生[所以，我们这种通过修改 MIDI 数据来修正动画的情形，在其他的工作室中也有[不清楚意义]的状况啊（笑）]

还有，游戏中的乐曲播放不是用 MIDI 进行的。还是不能把基于 MIDI 的软件合成器在 Runtime 驱动，在最终的做成阶段(Audio Mastering)，由声音小组把乐曲在 PCM Stream data 上进行混音(mix down)。但是，前述的控制动画用的控制变化 MIDI 数据，是在 Runtime 时，用和乐曲 PCM Stream 同步的形式来实时的再生。即使是 Runtime，MIDI 驱动也会缩小规模，在暗中支持着[Child of Eden]的图形。

另外，在 Runtime 时，对播放的 PCM 声音 Stream 应用 FFT (Fast Fourier Transform 快速傅立叶变换)，也加入了取得复数乐队波谱数据(spectrum data)的结构。所以，就像[某个频率声音响起时，这个图形就会运动]一样，也具备了音乐可视化的机能。

高梨先生[因为 spectrum 抽出是有些沉重的机能，所以要叮嘱妥程序团队不要太积极的使用（笑）。但做出了使用了这个机能的音乐可视化应用后，如果和制作应用软件有关的业务会很有趣吧（笑）]

还有，[Child of Eden]是绝对的时间轴是由乐曲支配的。例如，敌人的出现等，不是用[从舞台开始后的几秒后]或[玩家到了某个地点]的管理手法，而是采用[乐曲到了哪个节拍]的那种管理系统。还有可以连射的 Balkan 炮(traser)的发射时机是一乐曲的 16 分音符为单位，锁定激光(Lock on laser)和节拍和命中敌人时机同步是在系统上自动的调整。但是，影响游戏得分的玩家评价，和敌人命中的时机无关，而是激光发射的时机和节拍一致才能得到高评价的结构。

[在[Child of Eden]中绝对的时间轴是乐曲]

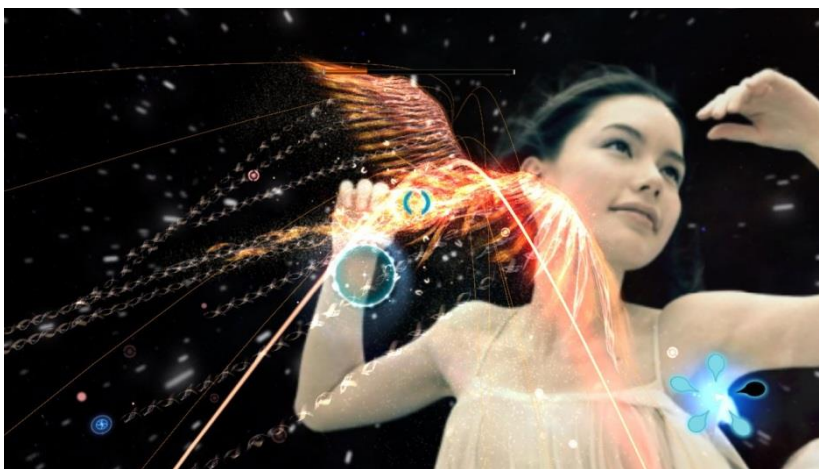


像上下广阔水面一样的背景的阴影色和太鼓的鼓点同步后产生变化。对于[Child of Eden]来说，乐曲才是绝对的时间轴。

[Child of Eden]特有的图形表现的秘密 2——“逆”的 Projection Mapping 的表现

[Child of Eden]的图形中非常具有象征性的是[Projection Mapping]的表现，所谓的Projection Mapping，是把CG用投影仪等，投射到人工建造物等现实世界的立体物体的数字艺术的新表现手法，现在在数字显示屏（Digital Signage）领域的应用上也是门值得期待的热门技术。

[Lumi]



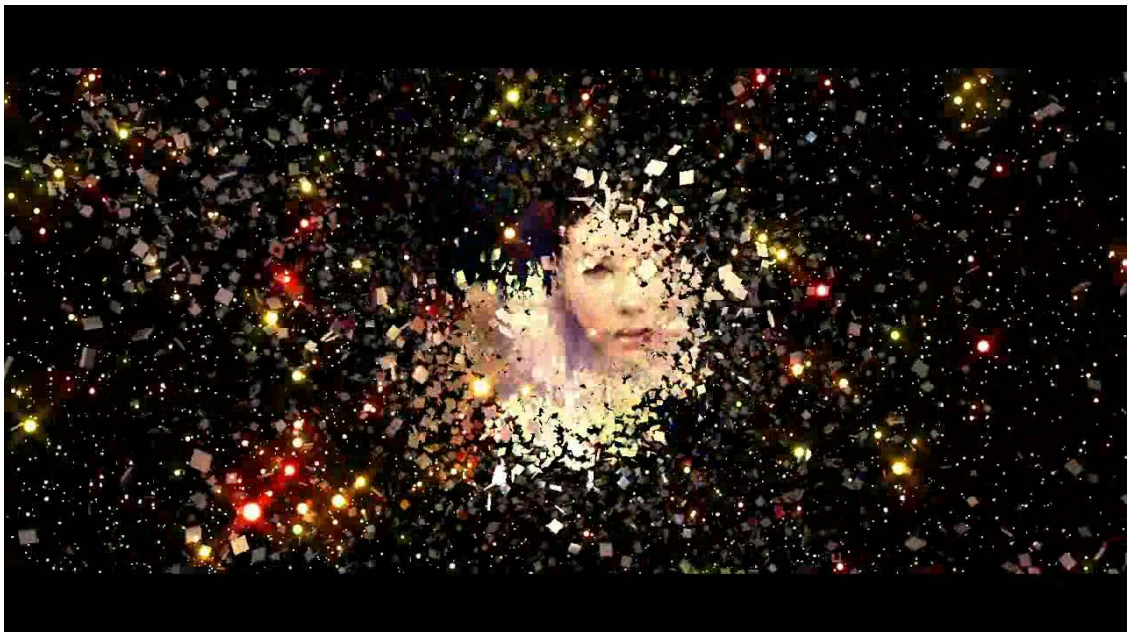


作为[Child of Eden]的女主角，元气火箭的主场和 Front Act（乐队前座奏乐）的 Lumi 登场了。

在游戏中，游戏进行和表演的一个环节。把游戏女主角 Lum 的表演摄影的写实电影 Projection Mapping 到，游戏场景内生成的各种各样的假想立体物体上。与把假想物的 CG 投射到现实立体物的一般 Projection Mapping 相反，把写实影像投射到假想物的 CG 对象上

现实世界 Projection Mapping，作为投射目标的立体物体是不动的，动的只是 CG 一方，而[Child of Eden]应该是“相反的” Projection Mapping 表现手段，因为映像（Movie）和透射目标的立体物体都在活动，所以可以看到非常幻想的有机表现。

[Projection Mapping]



在无数的粒子上 Projection Mapping 视频纹理的崭新表现

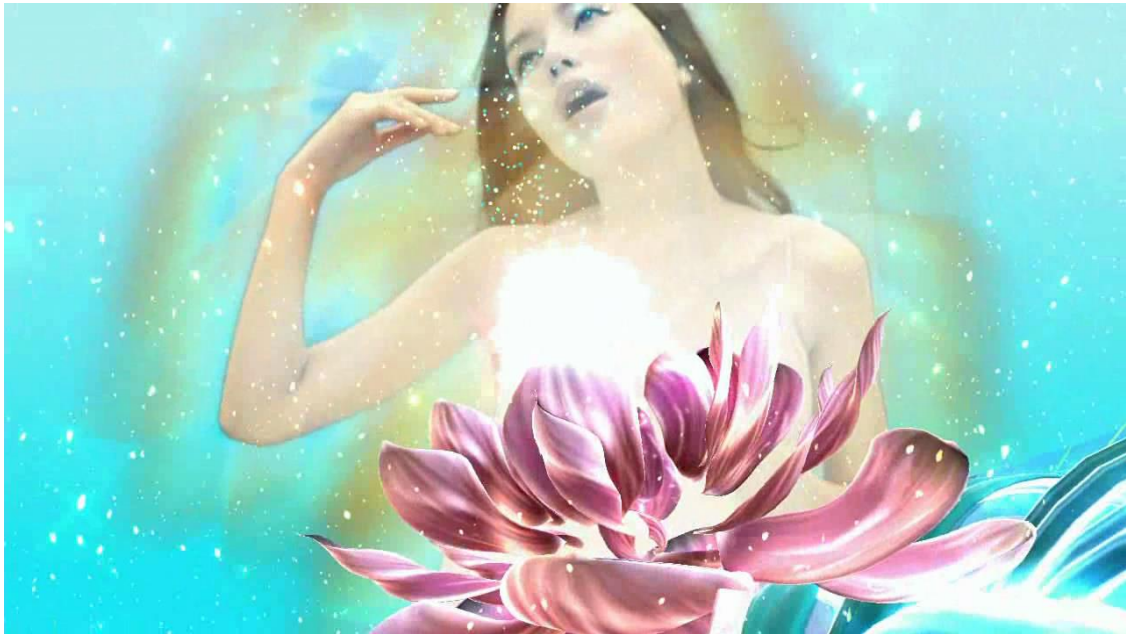
[Child of Eden]里叫做视屏材质的素材，大约有 170 个文件，总收录时间是 45 分钟，大约 500MB 以上。在迄今为止的游戏中，虽然有收录过面向事件场景的长时间影像的游戏，但

这种大量使用视屏纹理的游戏还没怎么看到过。

还有，贴付的影像纹理，基本上在 XBox360 上是 WMV，在 PS3 上是 divx，在贴付时，虽然也有只是简单的从视点投射纹理的场景，但也有些场合要加入扭曲贴付的纹理的技巧。WMV 素材，如果使用运行内存 (On Memory) 的状态，那么也会有一边缓冲一边从光盘媒体直接 Stream 播放的情况。

高梨先生[怎么说呢，稍微有一些像是音乐家制作宣传视频的感觉吧（笑）。实际的问题，[Child of Eden]的系统，可也可以这样认为吧。]

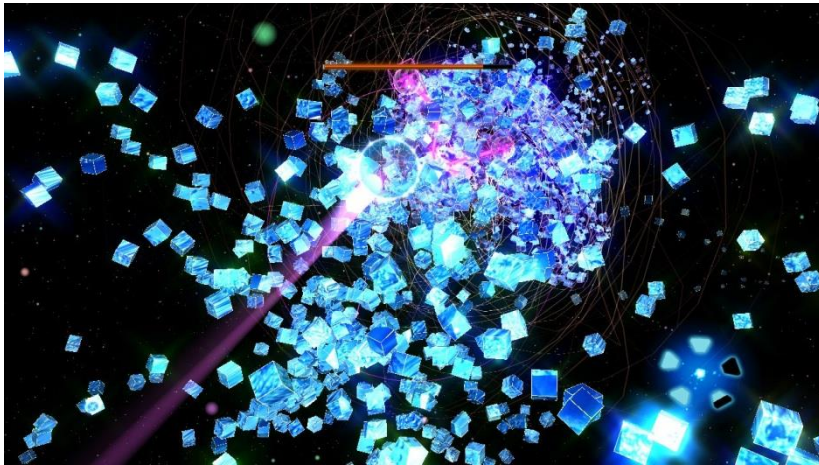
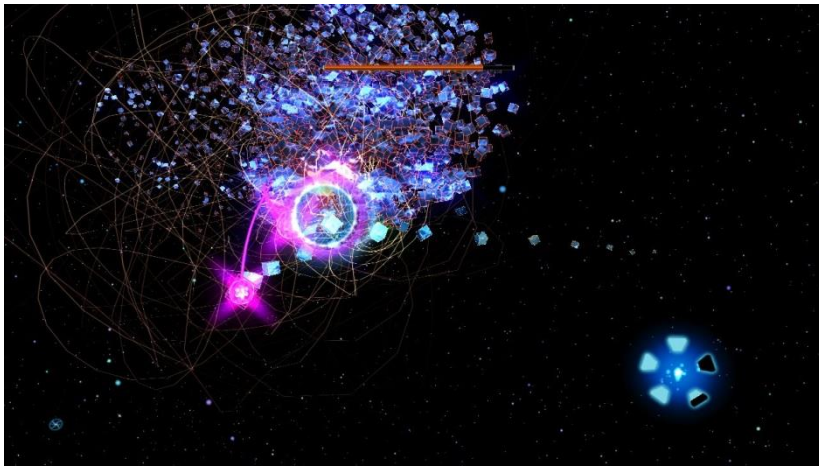
[利用视频纹理的音乐视频风格的表现]

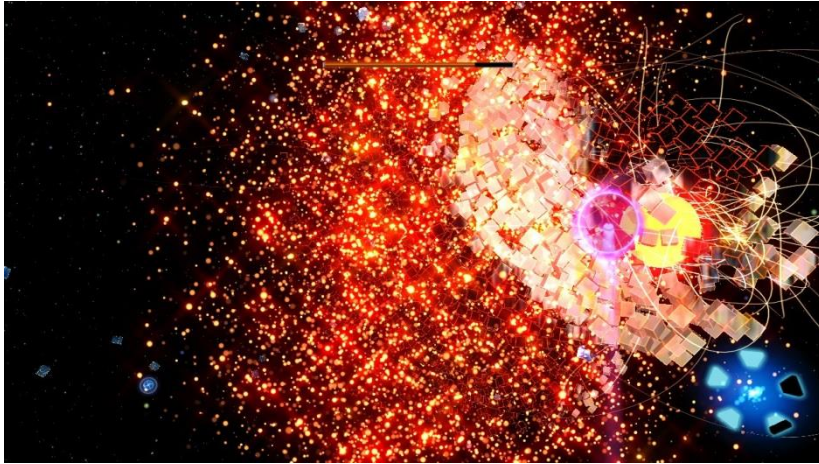


[Child of Eden]特有的图形表现的秘密 3- 没有使用 Geometry Shader 在 GPU 完成大量粒子生成绘制的方法。

要说起[Child of Eden]的图形，迄今为止没见到过大量的粒子在画面内立体的转动的表现是其特征。有时整齐，有时又杂乱，一边按照一定的力学，同时一边发出无数的粒子群转动。这个也可以说[动态丰富]的图形风格，特别在 3D 立体视觉模式中可以给与强烈的感动。即使那样，这么庞大的粒子，不会稍微超越了 PS3 和 XBox360 的 GPU 的性能吗。

[GPU 的粒子表现 之 1]





快要超越硬件规格的全局的粒子表现

高梨先生[开发出了几乎不介入 CPU 就可以在 GPU 内完成的粒子系统，用这个来驱动。这种技术一般成为[GPU 粒子(GPU Particle)], 把这个用工具做成的 BISHAMON 追加机能就是[Mass Particle]

GPU 粒子是在 DirectX10/ SM4.0(ShaderModel4.0)以后的 GPU 上实用化发展的技术，灵活利用把几何体信息的生成和销毁的第 3 个可编程 Shader[Geometry Shader]。PS3 和 Xbox360 的 GPU，因为是 DirectX9/SM3.0，所以不支持 Geometry Shader，那么到底是怎么实现的呢。

对在 DirectX9/SM3.0 实现 GPU 粒子有两个方法。一个是使用 Vertex Shader 和 Vertex Texture Fetching(VTF)的实现方式。。VTF 是在 DirectX9/SM3.0 上作为可选功能搭载的技术，可以在 Vertex Shader 参考纹理(reference Texture，只能读取)的机能。在 DirectX9/SM3.0 发布时，虽然是只能在 NVIDIA 系 GPU 才能利用的机能，但 AMD（旧 ATI）系列，也在 DirectX 10/SM4.0 以后的 GPU 上对应了，Xbox 360 上尽管是也 ATI 系 DirectX 9/SM3.0 GPU，但支持这个机能。

首先要提前准备的是，管理位置坐标 x、y、z，颜色，速度 vx、vy、vz，纹理 uv 等，要生成的大量粒子群的参数群。1 个种类的粒子群，要预先记录在最大 4 张每个纹素的 RGB α 元素上。DirectX 9/SM3.0 的 GPU，Pixel Shader 虽然可以 reference 的纹理有 16 张，但 Vertex Shader 的 VTF 只能 reference 4 张纹理，所以给出了最多 “4 张” 的限制。

还有，粒子是非四边形的多边形板状的几何体，都是同一个形状，所以，使用了在 DirectX9 的 SM3.0 以后的 GPU 上搭载的几何体实例(Geometry Instancing)，预先准备了大量的粒子绘制。Geometry Instancing 是[如果 3D 物体相同形状时，把 3D 模型的形状信息等先传到 GPU 一方，只传送想要的坐标，方向，纹理地址等显示的附带信息的数据，那么就能够实行实际的绘制处理。]的高效率的绘制技术。

之后 Vertex Shader，使用 VTF 机能，从纹理读出这些粒子群参数，一边应用 Geometry Instancing，一边设置在 Frame 描绘必要的粒子的绘制用几何体。

把例子群在个别上进行移动或转动，或是改变颜色，做出材质纹理动画等各种菜蔬的更新，

在 DirectX 9/SM3.0 代的 GPU 上，写入内存为了只用像素着色器进行，必然性的要使用像素着色器。

移动或转动个别的粒子群，颜色改变或者是纹理动画等各参数的更新，因为在 DirectX 9/SM3.0 的 GPU 上，除了 Pixel Shader 以外没法写入显存，所以必须要使用 PS。

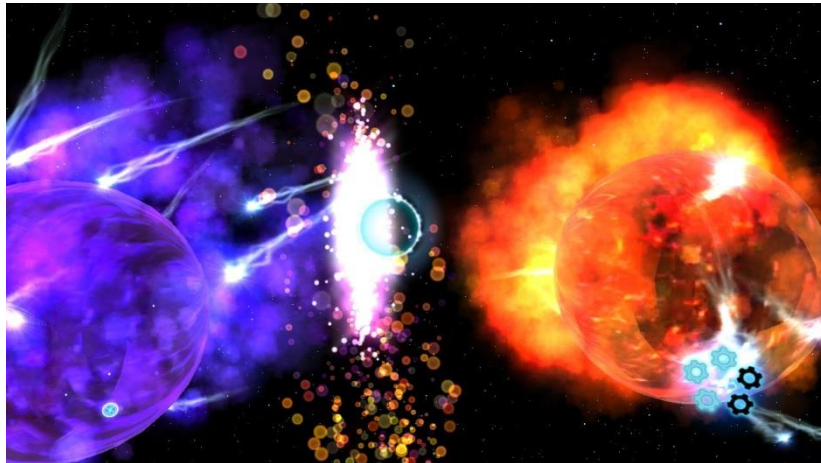
另外一个 GPU 粒子方法，是让 CU 辅助驱动，虽然不清楚 CPU 介入能否成为[GPU 粒子],但这个和游戏机稍微特有的硬件编码技术。

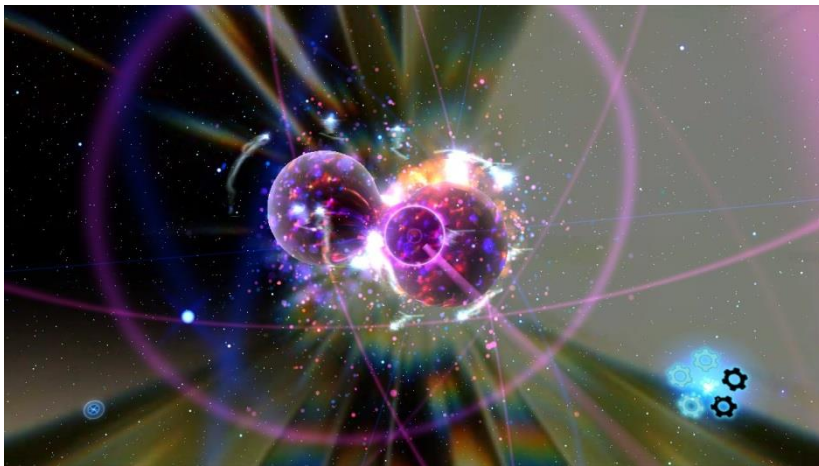
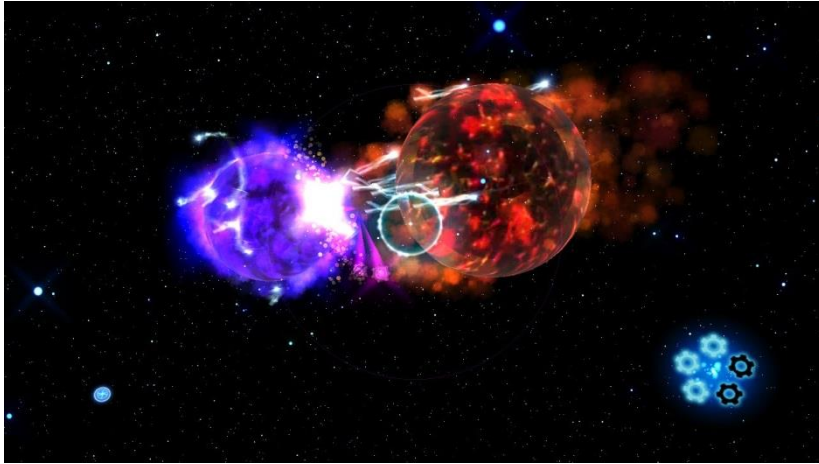
PS3 的 GPU 是 RSX，因为是 NVIDIA 系，虽然可以利用 VTF 的 GPU 粒子技术，但因为 RSX 的 Vertex Shader 比较慢，所以 PS3 版的[Child of Eden]，是使用 PS3 CPU CELL 处理器内的 SPE (Synergistic Processor Element：128 位的 SIMD 型向量 RISC 处理器) 来驱动，让它代替[Vertex Shader X 利用 VTF 机能]的阶段和[粒子参数更新]阶段。

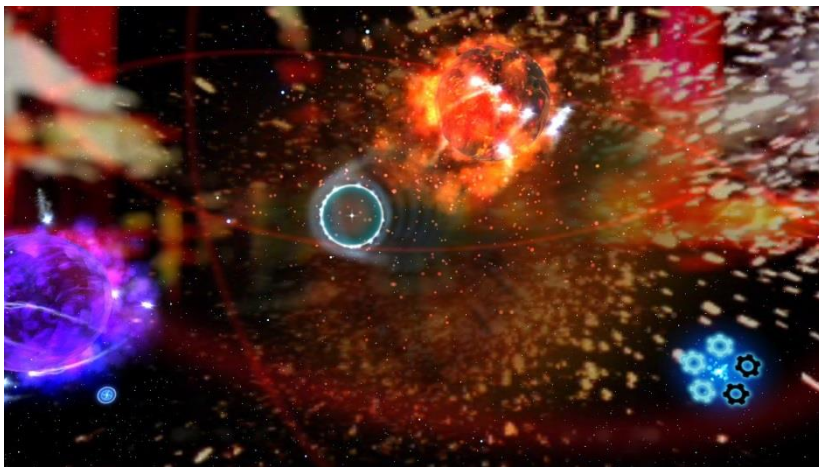
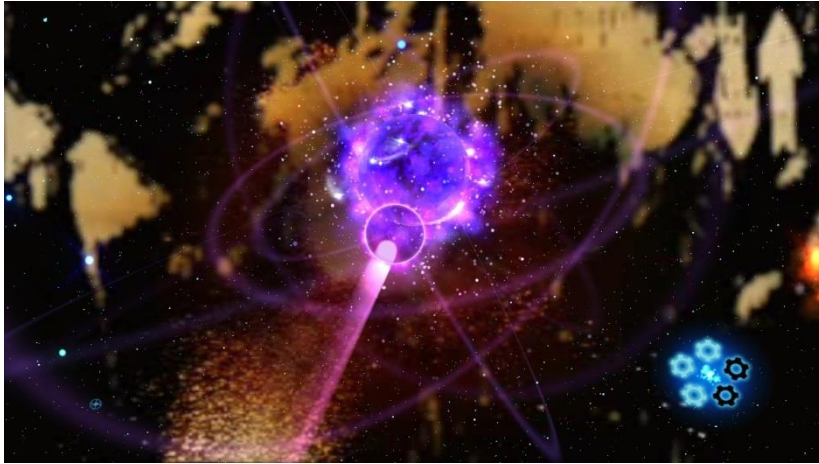
对于 XBox360，一部分特效是利用 3 核 6 线程 CPU 和 GPU 共有共通的物理内存空间的硬件特性，实现在 CPU 线程上支援 GPU 绘制的粒子生成，参数更新的结构。动态丰富的[Child of Eden]特有的图形表现，是由这样的高技术来实现的。

[GPU 粒子表现 之 2]

没有 CPU 支援的 GPU 内让粒子绘制完成的[GPU 粒子]机能，也可以说是支撑[Child of Eden]的粒子为主体图形的主干技术。







[Child of Eden]特有的图形表现的秘密 4——不可错过的 blow particle 表现！

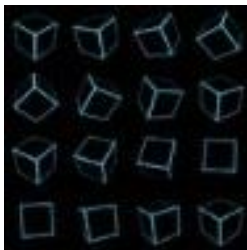
[Child of Eden]特有的庞大数量的粒子飞舞的 blow particle 表现虽然很清楚是用 GPU 粒子技术实现的，但根据粒子不同，不一定用板形多边形，也能在到处看到像是拥有立体的几何构造一样的东西。

[无数的立方体有 2D 也有 3D]



双重 Link 启动飞舞的无数青色立方体粒子实际上是以板形多边形为基础

高梨先生[不，实际上是板型多边形吗？（笑）。可以立体的看到几何体的构造是不是艺术小组的技术呢(笑)。虽然也是被图形感觉拯救的部分，但那只是纹理动画]



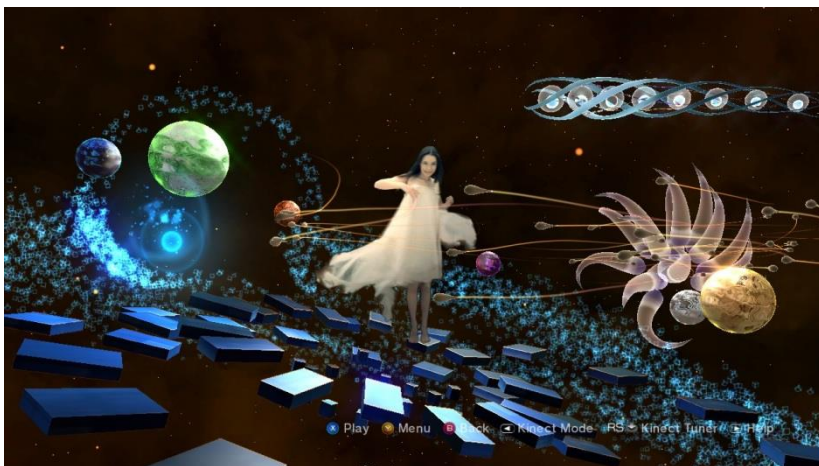
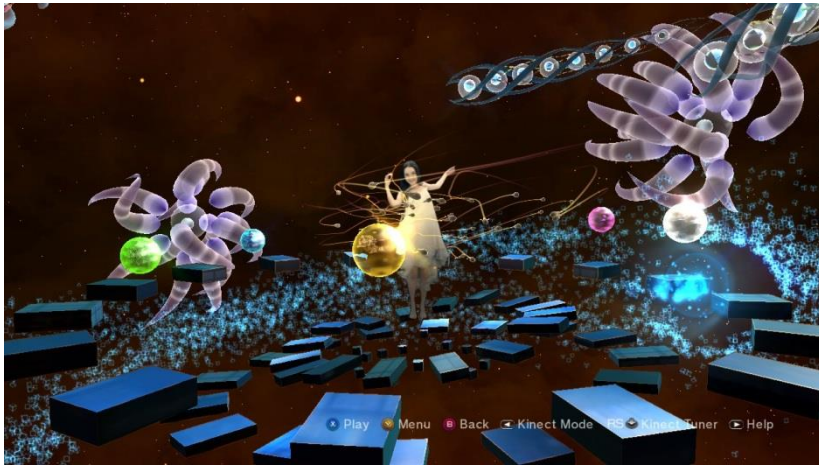
立方体的转动表现实际上是由 16 张纹理的动画来表现

首先，看起来好像有几何体构造的纹理动画，是把 3D 物体运动渲染的结构作为纹理在每帧渲染在单一的四边型上，或是作为纹理贴在多边形上。例如，菜单画面中出现的无数青色的立方图，就是由有这个技术表现的，青色立方图由 16 个纹理动画组成。这个纹理动画，是在某视点预渲染的图像，这里描绘的形状，虽然从玩家的视点看到的不是正确形状，但因为这些粒子是在远处，看到不会那么不自然。

高梨先生[非常稀少，这个附带纹理动画的粒子在离视点非常近的位置出现，这个时候还是会感觉到 16 张动画的抖动感和形状的不自然吧

在[Child of Eden]中，这样的质量粒子的表现的轨道本身，不是只被固定的物理法则支配，而是感觉由某种意志构成的形状。例如，前面的菜单画面，无数的（贴上了纹理动画的粒子）立方体虽然是在二重 link 的轨道上公转，但这里一旦被光标对着，就会发生排斥力，立方体群就会啪的一下飞散开，过会由会恢复到原来的公转轨道上。

[二重 link 的轨道上旋转的无数立方体粒子群]

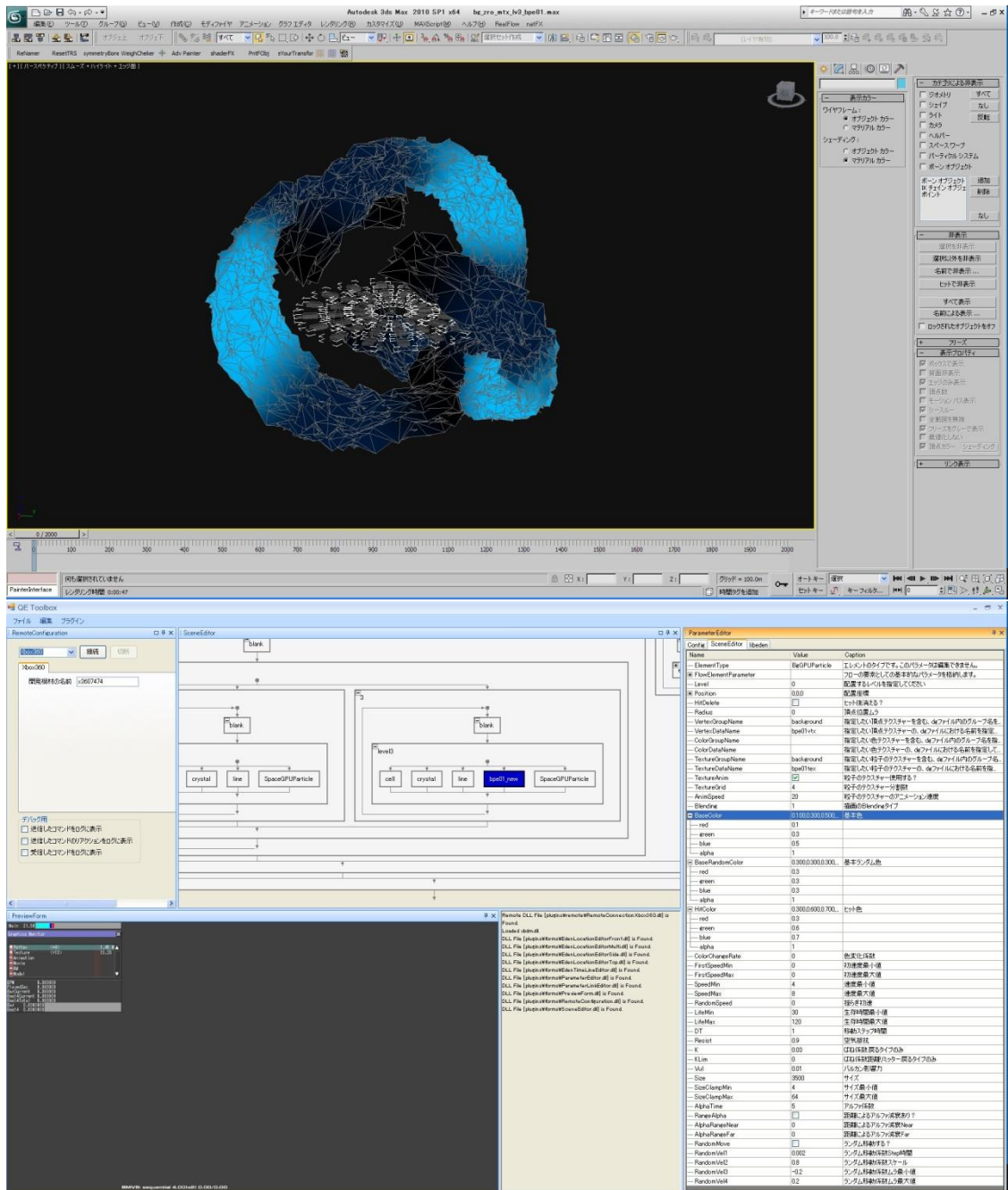


一旦对上光标，排斥力就会发生，立方体群就会啪的一下飞散开，一旦短时对上光标就会再次恢复到原来的双重锁链的公转轨道上。

高梨先生[把 GPU 粒子本身的轨道用几何形状做出设计。如果用这个菜单画面来说，是用二重的甜甜圈形状把 3D 模型设计出来，这个二重甜甜圈的各项点就成为 GPU 粒子的发生源，并拥有从那里产生的粒子的数量和颜色等的参数]

总之，blow particle 的公转，使用这个公转 3D 模型本身的转动和动画来表现的，发生排斥力等的情况，也是做出像是回到原来发生地一样的轨道控制。要是揭穿了秘密，虽然有种[算不了什么]的感觉，但做成影像的外观是强力的。是否完全可以说成想法 (idea) 产生的魔法表现呢。

[blow particle 的配置形状是 3D 模型作成的]



粒子的配置信息用 3dsmax 的 3D 模型来做成。粒子被配置在 3D 模型的各顶点上，是由各顶点的颜色反映粒子颜色的结构(上图)。各粒子的基本举动的算法是在称为 QEToolbox 的公司内部专用工具上设定的（下图）。

支撑[Child of Eden]开发的[BISHAMON]。[要做的更加有机!] (笑)

所以这样就明白了, [Child of Eden]的图形中, 以粒子为主题的特效担任了很大的作用。紧密联系游戏系统的特效群, 虽然是由设计团队的构思和艺术感觉的程序的相互配合才产生出来的, 但那以外庞大数量的特效群, 利用了在本连载中也刊登过的特效设计工具/中间件[BISHAMON]。

高梨先生[[Child of Eden]的图形, 因为是以特效为主题, 所以提高特效的开发效率就是当务之急。但是, 在预定好的开发期间内, 从零开始开发自己专用的工具是很难的。因为这样的流程, 才采用了[BISHAMON]



藤本文彦先生 (matchlock 董事长)

藤本文彦先生 [随着[Child of Eden]的开发, [BISHAMON]在一起剧烈的进化着啊。果然像彼此所说的互相共同开发的关系比较好吧(笑)。例如, 在BISHAMON上新搭载的Mass Particle机能等, 在[Child of Eden]的开发中可以说是互相提高而产生出新机能的代表案例。]

要说两个公司从最初就很圆满, 实际上不是这样。虽然[BISHAMON]确实可以很轻松的做出特效, 不过据说Q ENTERTAINMENT方面在开发初期就给出了[活动太僵硬]的严厉评价。视点(摄像机)和特效发动位置如果固定的话, 那还算能做出好看的特效, 在视点和特效都活动积极的「Child of Eden」里就会有脱节。具体的, 当初用[BISHAMON]制作的特效群怎样动都是算术的, 和游戏系统上直接实现的特效群之间似乎缺乏整体感

藤本先生[被屡次说到 [要更有机些] (笑)。因此, 特效部件的活动上可以附加引力和排斥力等物理参数, 在参数上不只是给与Functional Curve, 而且还能附加上噪声。根据这样, 做出了同一个特效, 每次活动都会不同的效果]。

高梨先生[在我们这边, 每次反复时, 因为BISHAMON的使用方法都会有进步, 所以渐渐的能制作出符合印象的东西了。即使同一个特效, 做成了父子阶层关系, 因为是父侧的活动和转

动中随机加入，所以变的相当的复杂。]

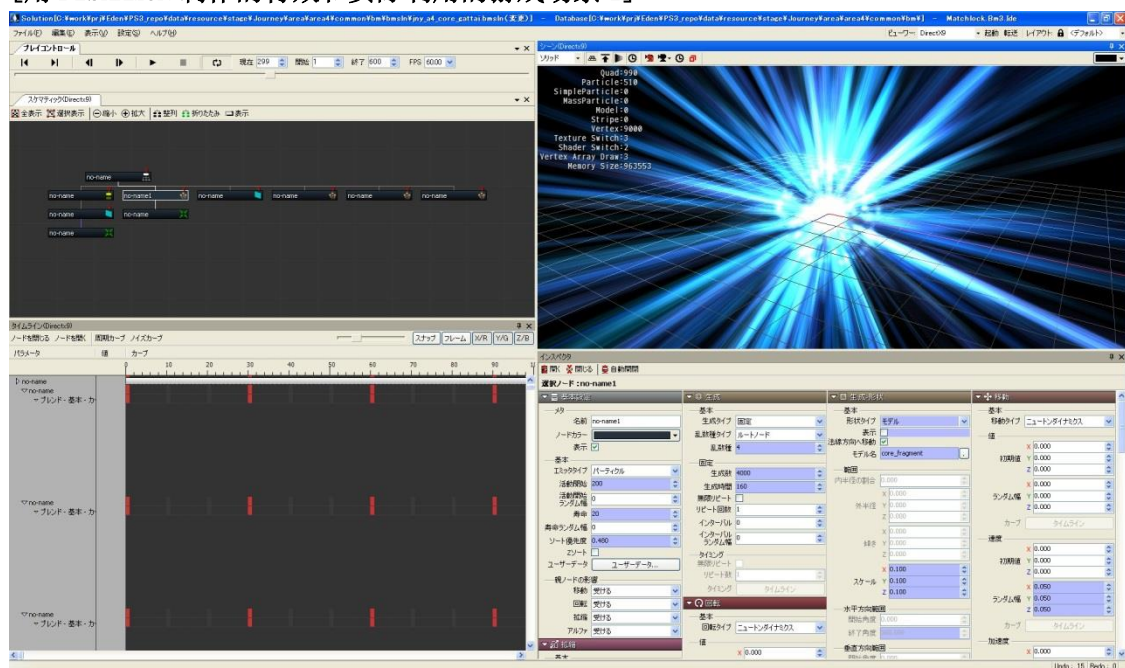
只要看到实际完成版的[Child of Eden],哪个是用[BISHAMON]制作的在[BISHAMON]Runtime上驱动的特效，那个是从游戏逻辑方面直接驱动的特效，已经很难区别了。实际上，据说即使在 Q ENTERTAINMENT 内部，除了一部分核心开发职员，也都很难判别。

但是，作为基本方针，大量特效基本上是由游戏程序作为直接驱动实现的，数量虽然还可以，不过活动的有丰富漂亮外观的特效要以[BISHAMON]为基础制作。Mass Particle 的机能，是为了回应对量的期待而从途中追加的机能。

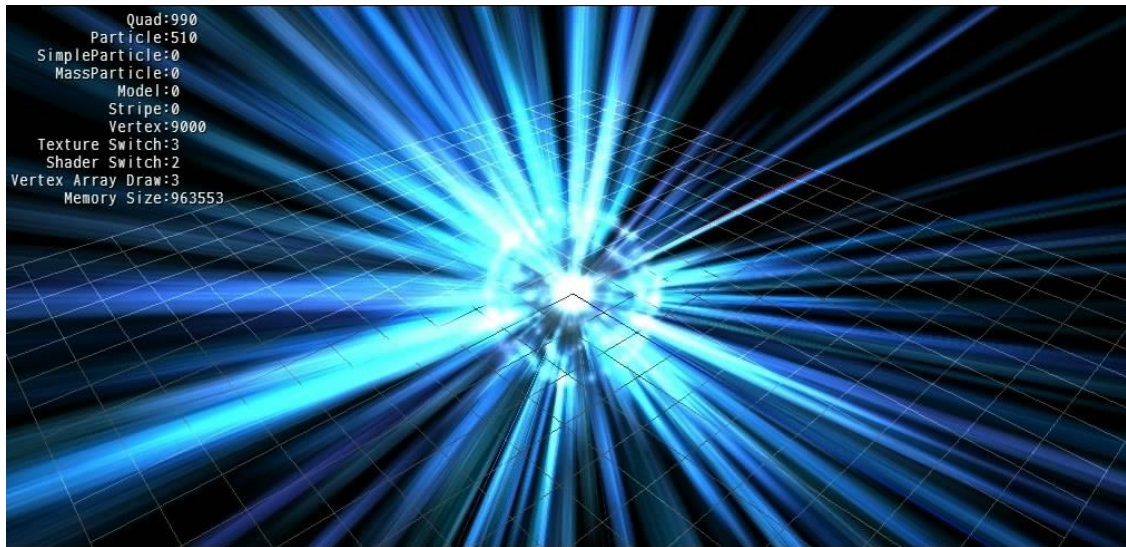
实际上，玩到产品版的游戏场景，虽然得到这个特效是由游戏程序直接驱动，那个特效是由 BISHAMON 制作的 BISHAMON 驱动形式的解说，但前述的基本方针实际在每个要素开发时是不是会遵守，要笔者在有限的杂志上列举[哪个是哪个]，感觉要解说相当困难。

敌人的破坏动画是以 BISHAMON 的基础的，中 BOSS，大 BOSS 发出的放射状巨大的特效，有以程序为基础的，也有以 BISHAMON 为基础的。不只是特效，出现的敌人本身也有是以 BISHAMON 基础开发后运动的情况。[长时间在画面存在的是用程序直接驱动]，[在同一个场景内反复再生的特效群是基于 BISHAMON 的]，笔者有这种感觉的倾向。

[用 BISHAMON 制作的特效和实际利用的游戏场景 1]



BISHAMON の工具画面

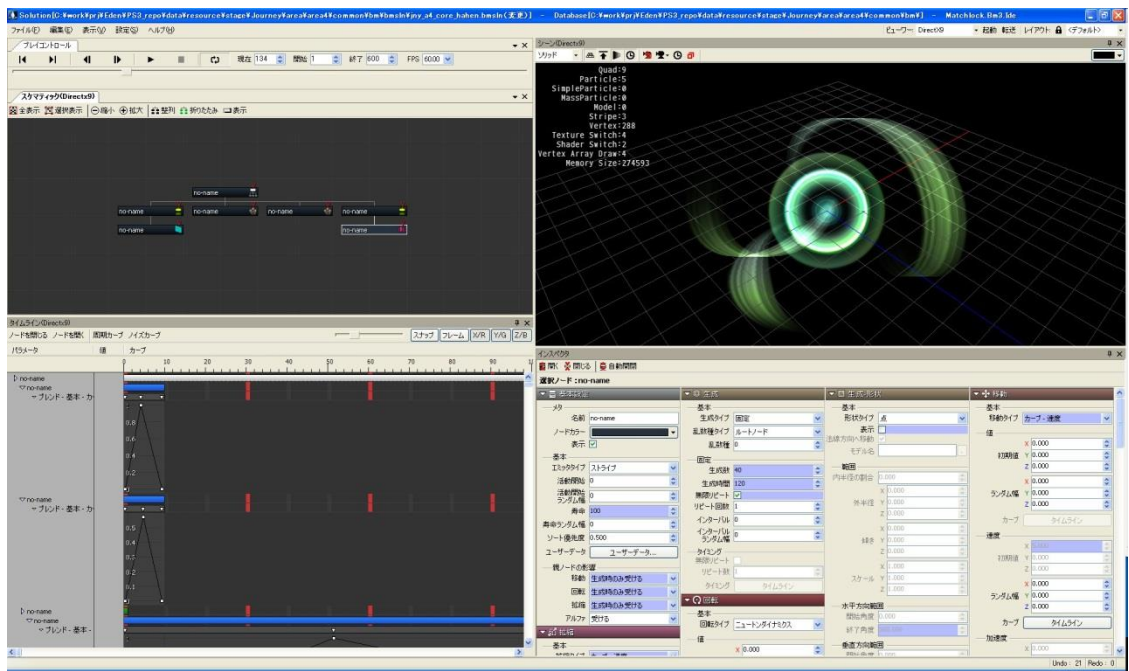


在 BISHAMON 上的预览

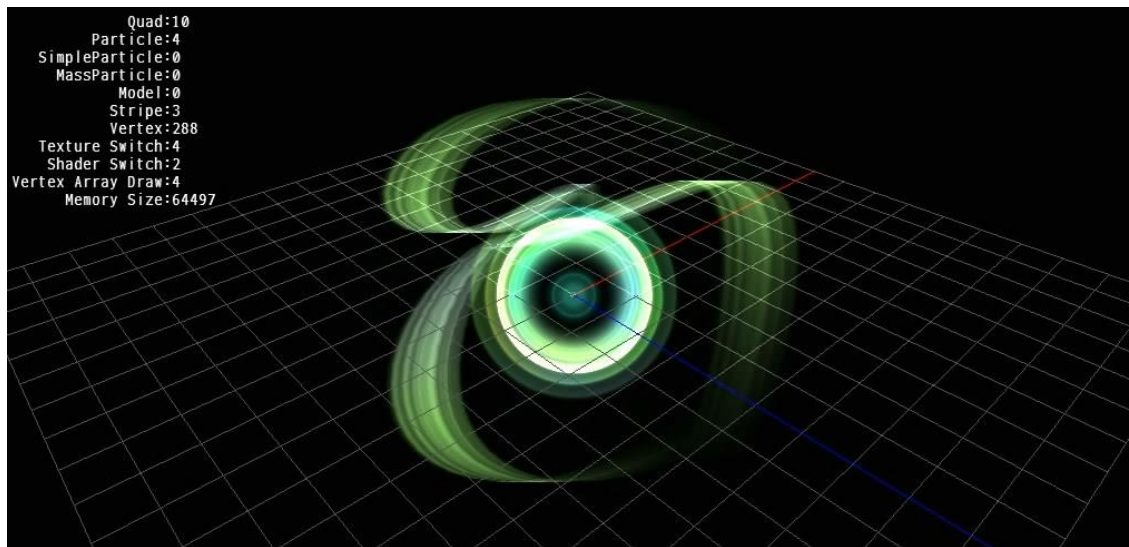


实际的游戏场景

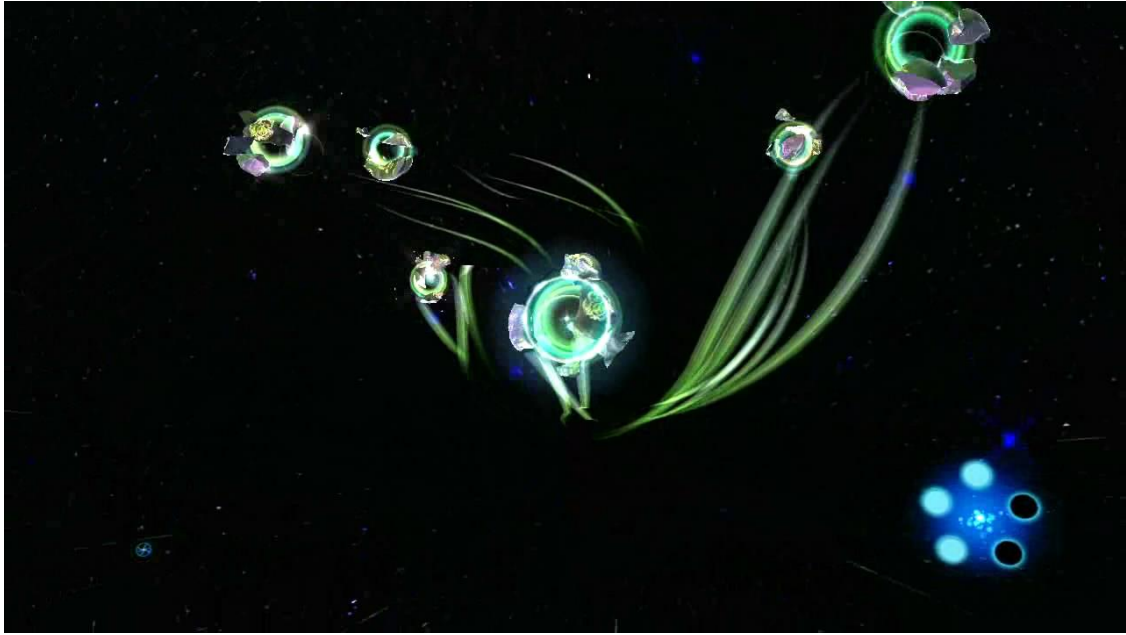
[用 BISHAMON 制作的特效和实际利用的游戏场景 2]



BISHAMON の工具画面

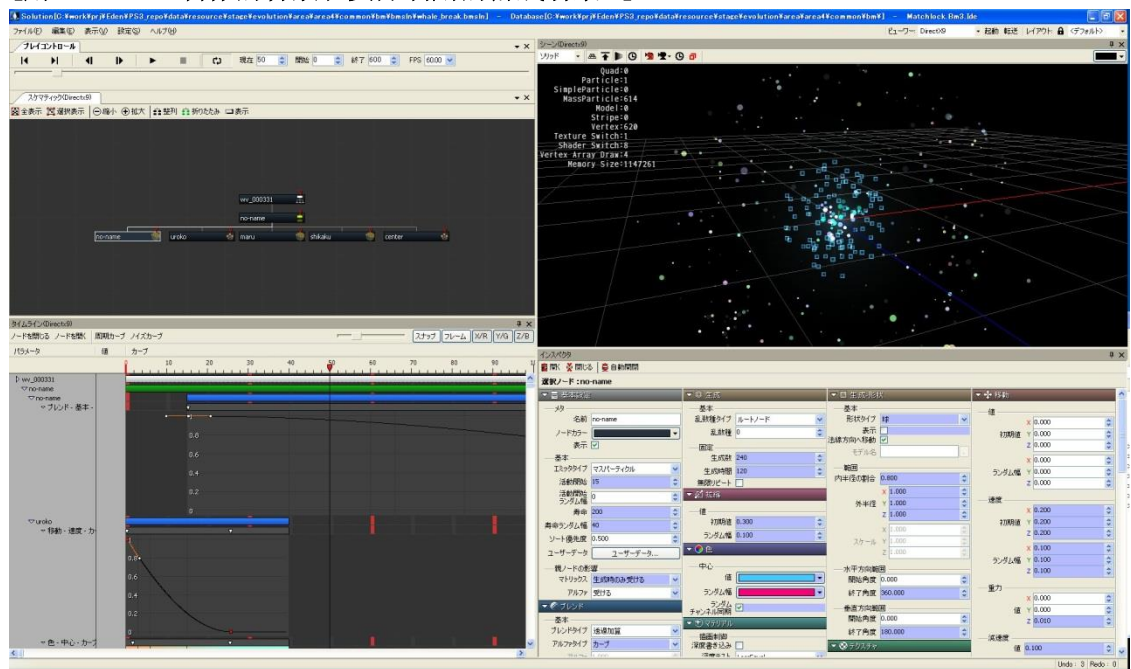


在 BISHAMON 上的预览

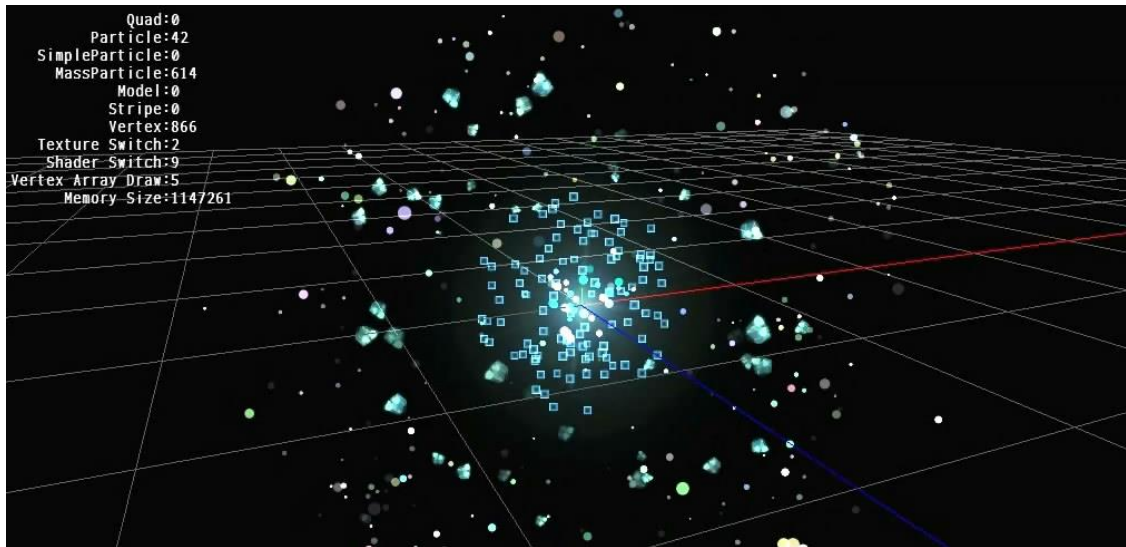


实际的游戏场景

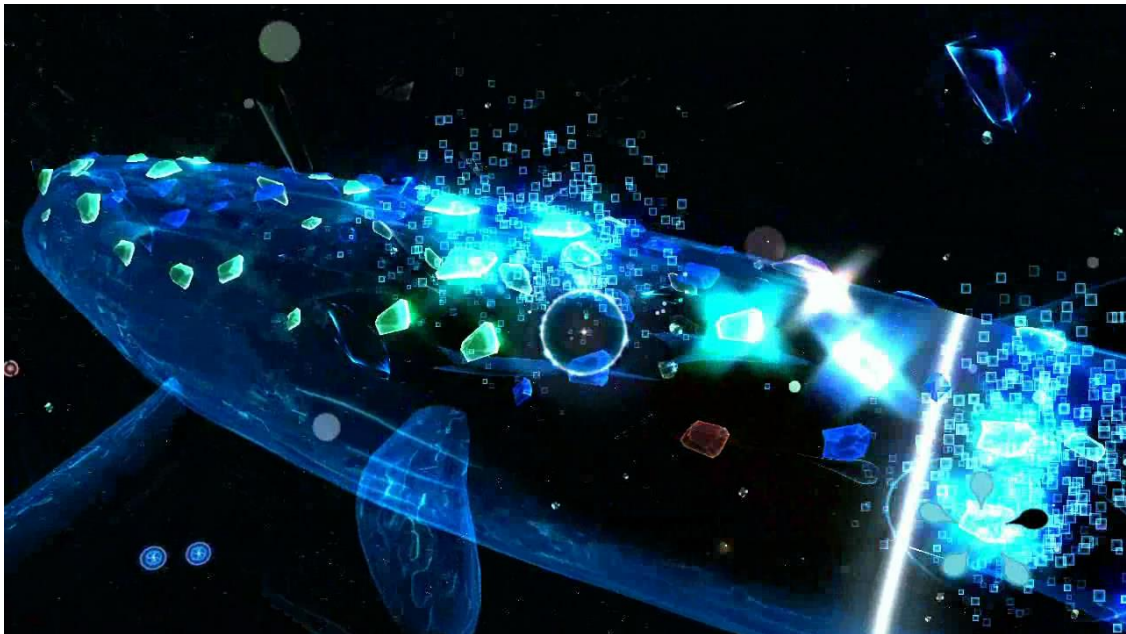
[用 BISHAMON 制作的特效和实际利用的游戏场景 3]



BISHAMON 的工具画面

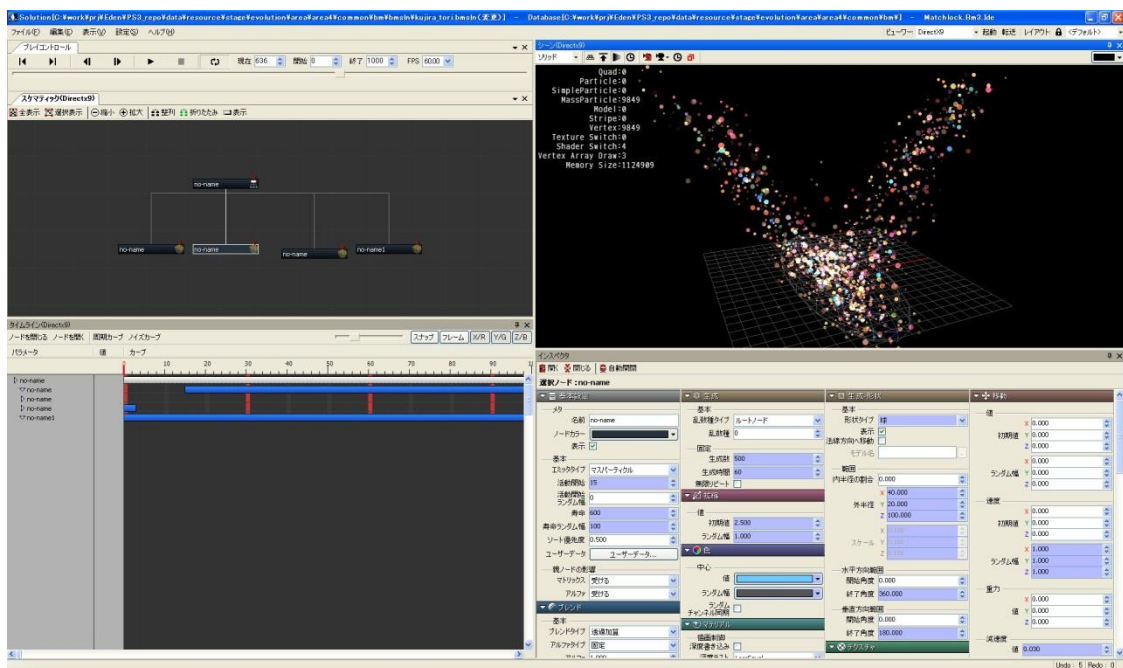


在 BISHAMON 上的预览

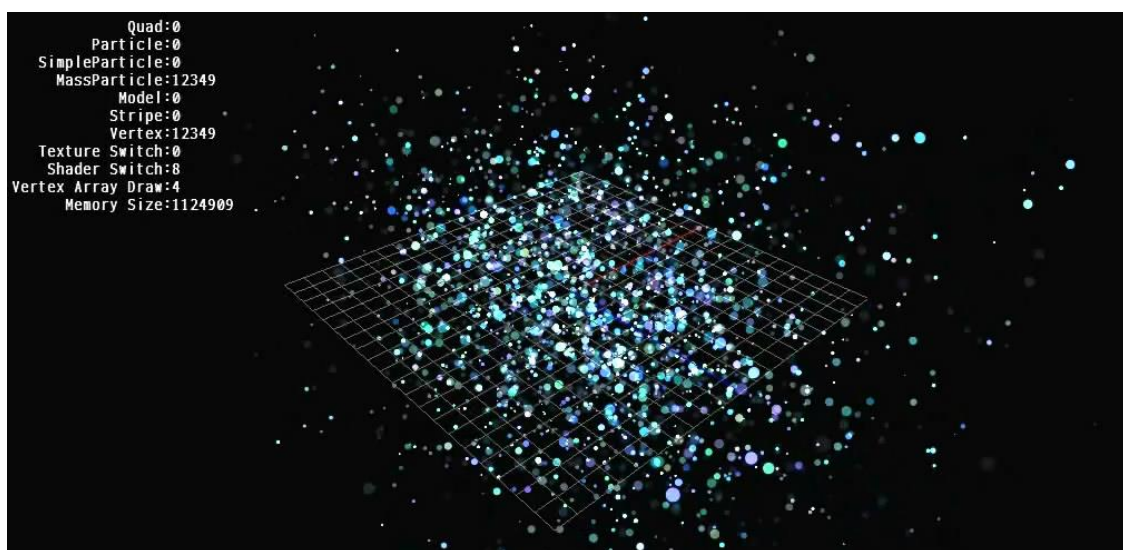


实际的游戏场景

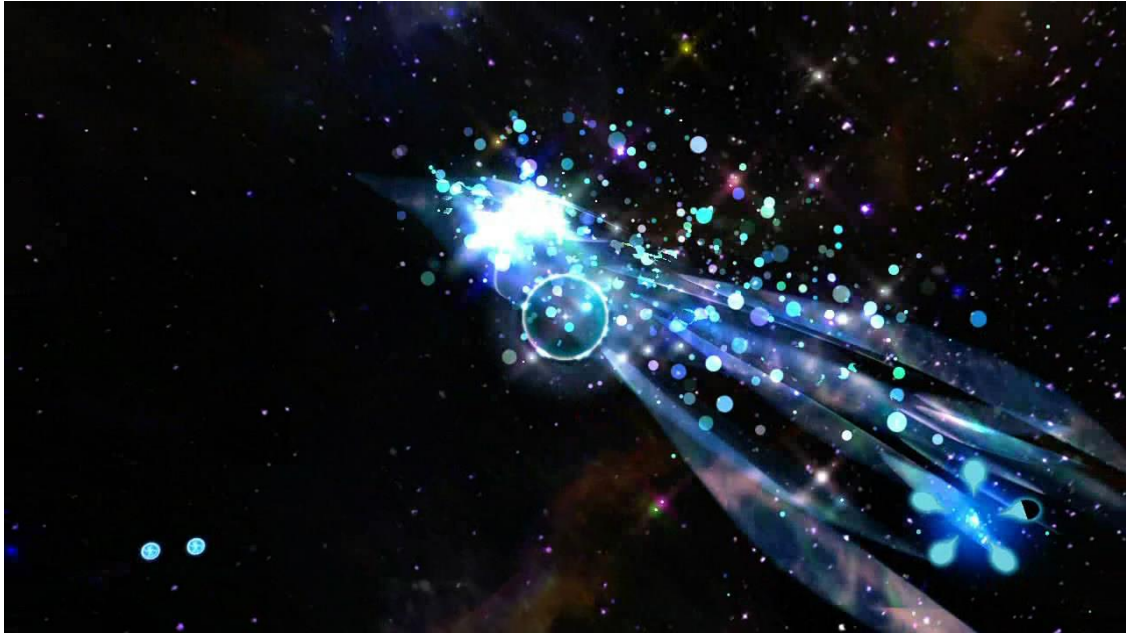
[用 BISHAMON 制作的特效和实际利用的游戏场景 4]



BISHAMON の工具画面

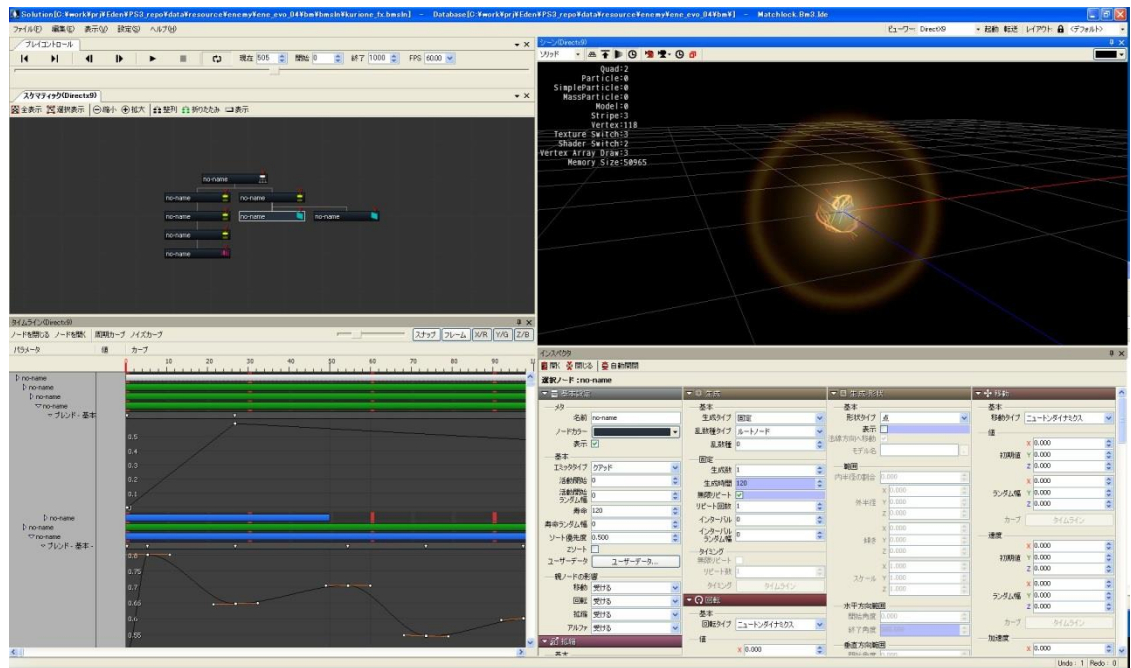


在 BISHAMON 上的预览

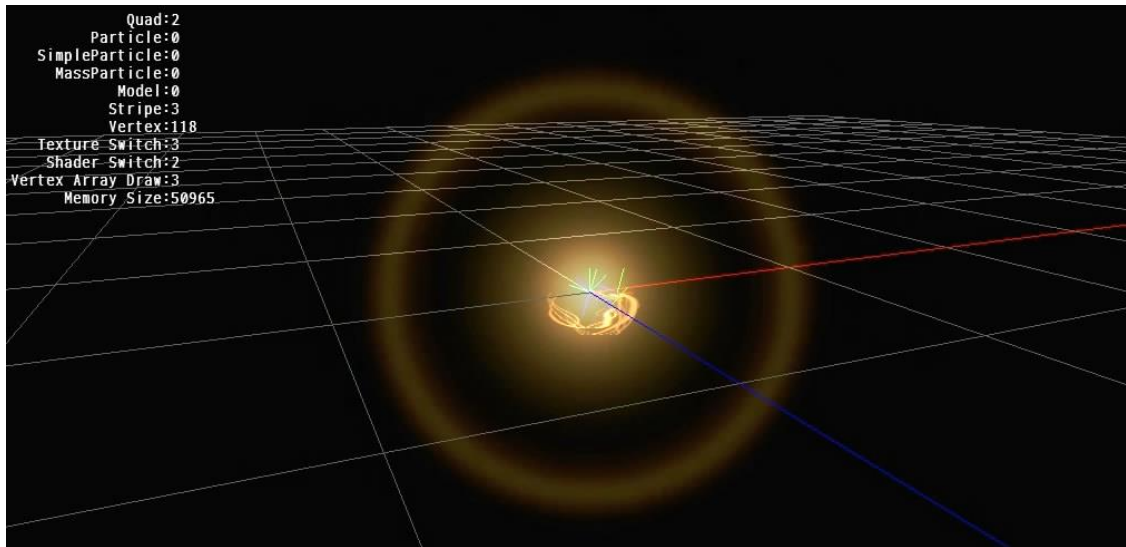


实际的游戏场景

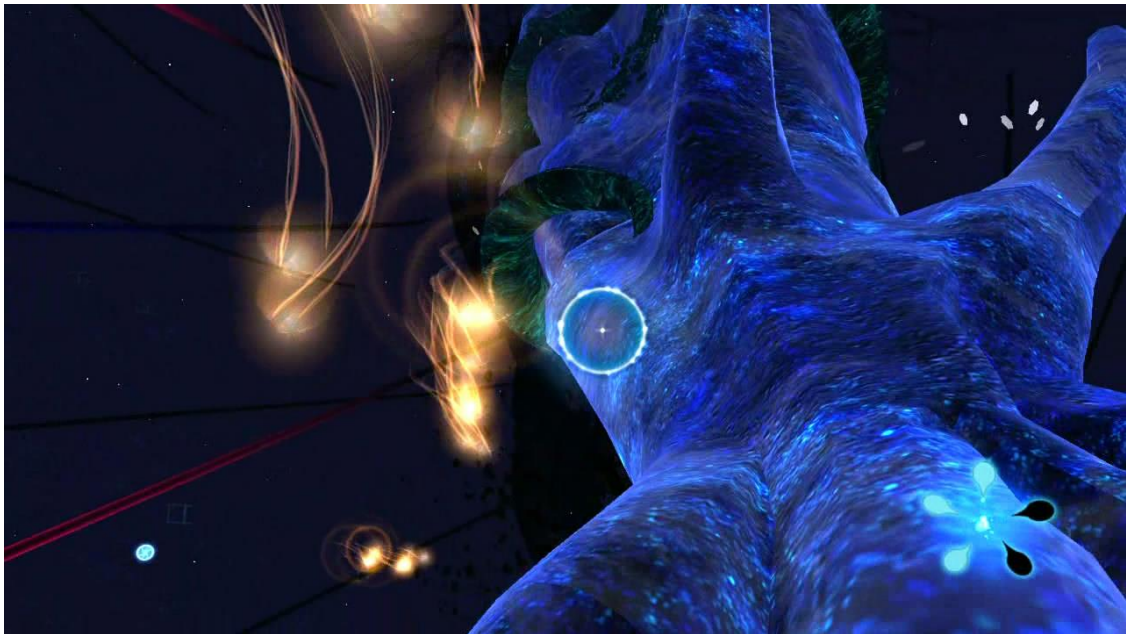
[用 BISHAMON 制作的特效和实际利用的游戏场景 5]



BISHAMON 的工具画面



在 BISHAMON 上的预览



实际的游戏场景

结语

对水口哲也先生率领的 Q ENTERTAINMEN 公司的作品，笔者感觉每个都有两点不偏离的方向性。第一是[玩时的心情舒畅]。在[Child of Eden]里是[瞄准射击]的射击游戏带来的[破坏的感情净化]，还有与玩家的动作的同步的影像和音乐，体现了新感觉的好心情。

还有另一点就是不畏惧的向新技术挑战的精神。这次的[Child of Eden]的开发，在很短的开发期间里，加入了 Kinect 的对应，PS Move 的对应，3D 立体视觉的对应，还有中间件

[BISHAMON]的采用等，本作的开发团队作为艺术团体的同时，也让人感觉是先端的技术专家。



最后，试着询问高梨先生，今后的 Q ENTERTAINMEN 公司作为目标的游戏方向性。

高梨先生[在[Child of Eden]中，因为没有对应声音输入，所以有个模糊的想法要在今后加入这样的技术。让玩家唱歌也是很有趣啊，用 Kinect 的 Microphone Array. 取得音场的定位，把这个做成某种应用也很有趣。在 Kinect 和 PS Move 的前面，绝对有我们想使用的技术在等候着。在[看到影像，听到声音]的视屏游戏的两大要素的前方，也许就是气味和触感等诉诸于五感的方式。并不是只能看到立体视觉，如果能触摸到凹凸也许更快乐吧]