



虚幻引擎



《堡垒之夜》 | Epic Games

《堡垒之夜》 工作流程：
用 UnrealGameSync 进行
大团队协作

目录

	页码
1. 介绍	3
大规模游戏开发的挑战	4
2. 传统工作流程	5
构建与修复循环	5
内容指向型和代码指向型	6
版本控制	6
3. 案例研究	7
《Battle Breakers》	7
《虚幻争霸》	13
虚幻引擎	15
4. 在你的项目中使用 UnrealGameSync	16

介绍

Epic Games 于 2017 年发布了《堡垒之夜》的原始版本，随后又相继推出了“拯救世界”、“大逃杀”和“创意”模式，目前游戏拥有超过 2.5 亿的玩家群，并且持续推出全新的内容、玩法模式和功能，让玩家不断回归游戏。

为了与时俱进，我们逐渐形成了一套可有助于高效修改《堡垒之夜》并及时交付修订版的有效方法。许多外部的工作室都曾询问过我们到底是如何实现这一点的。

本文将介绍我们开发流程的演变过程、背后的原因以及《堡垒之夜》和 Epic Games 旗下游戏实现这一点的详细情况。我们的解决方案主要是通过切实可行的方案，利用虚幻编辑器和 UnrealGameSync 工具满足不断迭代的需求。

我们谨以此文献给技术总监、构建工程师、技术美术师和游戏设计师，希望能帮助他们用虚幻引擎向团队分配工作，精简大型项目的开发过程。



《堡垒之夜》 | Epic Games



《堡垒之夜》 | Epic Games

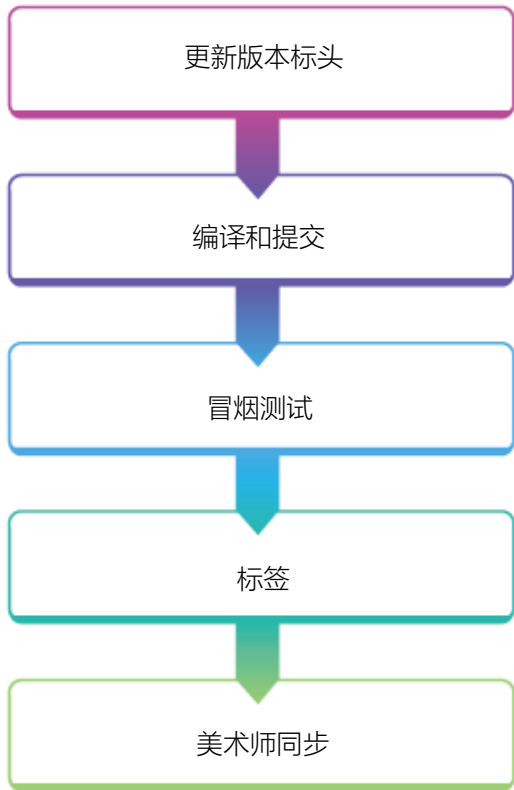
大规模游戏开发的挑战

在游戏开发过程中，程序员、设计师和美术师会通力合作，尽全力让游戏引人入胜，具有吸引眼球的视觉效果，并且避免出现错误。而随着游戏的不断更新迭代，开发者也将面临挑战，那就是要保证内容创作者构建和工具的稳定性的同时，继续为最新版本代码添加全新的玩法功能。

Epic Games 开发人员每天都会向《堡垒之夜》主线交付数百个改动，我们的基础构建和工作流程旨在保持这一改动量不变的情况下尽量确保进展顺利。不过事情也不总是尽如人愿。

传统工作流程

在《堡垒之夜》的早期开发过程中，内容创作者主要是依靠虚幻引擎 3 时间框架中的开发流程，名为“编辑器更新流程”。



图片 1: 传统编辑器更新流程

最新版的《堡垒之夜》源代码会被同步并且在用于构建的电脑上进行编译，然后得出的二进制文件会被提交到 Perforce 上。我们的质量测试团队会同步并运行这些二进制文件。如果它们通过了测试计划，团队就会给它们添加一个 Perforce 标签，将它们标记为可行。这个流程就叫做“更新”。然后内容创作者就可以通过 Perforce 标签来同步最近更新的二进制文件了。

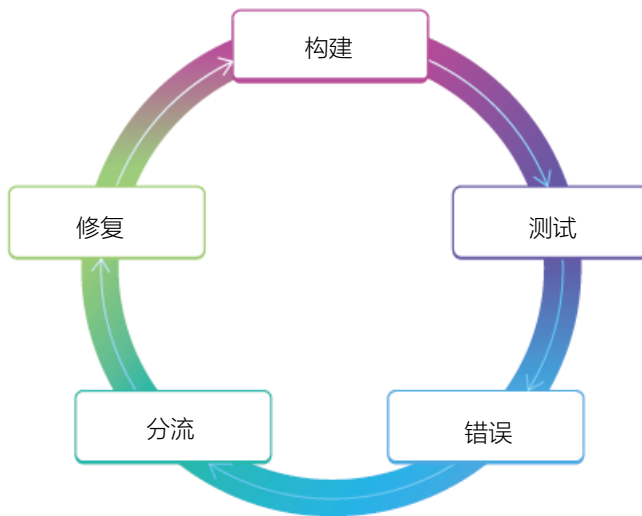
构建与修复循环

如果在编译或测试过程中发现了错误，质量测试团队就会将错误提交给开发人员，让他们可以分流和修复。

随着项目的规模和复杂程度不断提高，需要测试和查找错误的表层区域也不断增大。质量测试人员可能并不清楚错

误对特定项目的现行工作流程有多大的影响，所以他们经常因为过于谨慎而出现误判，将所有发现的错误都视为高优先级的问题，并输入我们的错误追踪软件。

图片 2: 构建与修复循环



构建与修复循环中的一次迭代最少需要花费半天的时间——如果开发人员无法重现问题或者离开了工作岗位，有时甚至需要花费几天的时间。但只有这样才能重复整个过程，而在第二次迭代的时候，很有可能会出现影响其它内容的新改动。

随着项目的发展，迭代的速度会降低，两次更新可能间隔一周的时间，这就对工程师和内容创作者的合作造成了挑战。

对于许多虚幻引擎项目来说，加快迭代速度至关重要，因为工程师和设计师经常要通过虚幻引擎的蓝图可视化脚本系统合作开发游戏功能。迭代对游戏开发非常重要，如果在向设计师交付新代码的流程中出现了瓶颈，就会导致审查和完善最终产品品质的机会减少。



《堡垒之夜》| Epic Games

内容指向型和代码指向型

这种工作流程还存在其它的问题。因为构建系统会不断向 Perforce 提交新的二进制文件，工程师经常会发现自己构建的和需要同步的二进制文件并不匹配。如果运气好的情况，操作系统会跳出一个模糊对话框，告诉他们无法找到损坏的符号名称。但如果运气不好，编辑器会直接加载，但是由于模块之间并不匹配，类布局会出现内存损坏，导致调用模糊堆栈而引起崩溃。

为了避免这种情况，工程师会在 Perforce 客户端配置上手动排除预编译的二进制文件，但是这个流程非常不可靠，同时也很难跟进新导入的文件。此外，使用大量的通配符来过滤文件会对 Perforce 服务器造成很大的负担，延长同步的时间。

如果工程师需要改动内容，他们就必须检入代码的改动，等待编辑器更新，然后才能提交他们改动的内容。如果他们尝试在本地构建中检入部分改动，这些改动就会与版本不符，使用更新编辑器的开发人员就无法加载它们。

因此要为你的团队部署虚幻引擎，理解版本控制至关重要。

版本控制

虚幻引擎的属性序列化模型对于持续开发而言非常方便灵活，你可以对编辑器上公开属性的代码进行迭代，并且无需重新保存已有的资产。

例如，如果你向一个类添加了一个属性，但之前保存的资产并没有这个属性，那么下次再载入这个资产时，引擎就会将该属性初始化为其默认值。如果属性的默认值没有被重写，则硬盘中加载的任何内容都会自动采用对默认值的修改。如果你移除了一个属性，任何拥有该属性值的资产都会在加载时忽略该属性。

缺乏显式的版本控制就会产生问题。如果你添加了新的属性，并且保存了相应的新资产，那么旧版本的编辑器就不知道该如何对其进行序列化，从而将其作弃置处理——也就是把它视作已经被移除了。

为了解决这个问题，我们将构建的改动列表编译到编辑器的执行文件中，把它保存到任何创建的资产中，然后防止编辑器加载新版本中创建的新内容。这种方法并不是对每条代码的改动进行微观上的版本控制，而是管理整个编辑器和其中保存的内容。



案例研究

我们在努力改善《堡垒之夜》繁琐的工作流程的同时，还在开发一个名为《Battle Breakers》的独立项目。它的游戏更新流程非常高效快速，于是我们就想要将这个流程延展到像《堡垒之夜》这样的大项目上。我们还研究了另一款游戏——《虚幻争霸》，了解它如何为自己的工作流程不断改进解决方案。

《Battle Breakers》

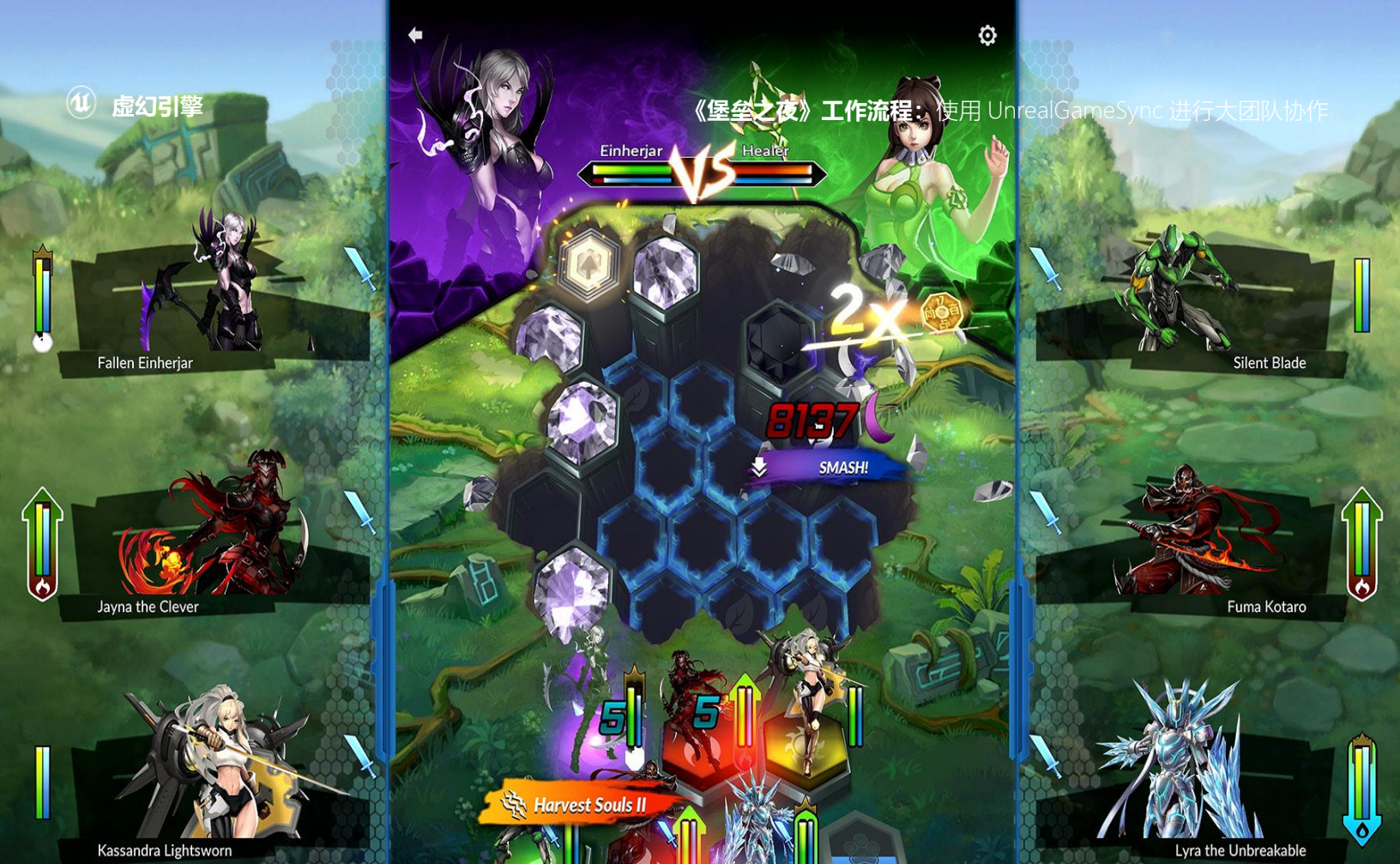
《Battle Breakers》拥有一支大约 10 人组成的团队，其中有美术师、设计师、工程师，他们都在一起办公。他们对自己的项目充满热情，努力想要让它大获成功，所以他们并不想被过分谨慎的流程束缚手脚。

他们并未采用标准的编辑器更新流程，而是每个开发人员，包括美术师和设计师的电脑都安装了 Visual Studio Express，并且学习如何在整合的开发环境下通过 Perforce 进行同步、生成项目文件、加载解决方案，并且构建自己的编辑器。

这种工作流程非常高效。

如果出现了编译错误或者编辑器崩溃，内容开发者可以立即向几步之外的工程师寻求帮助。他们的电脑上已经安装了 Visual Studio，并且可以使用完整的符号访问源代码，这样工程师就可以高效地调试编辑器，快速帮助他们重新开始工作。

此外，这样做还有一个意外收获，那就是美术师、设计师和工程师可以进行更广泛的沟通交流。他们更加熟悉彼此的工作流程，并且分享了关于如何改进编辑器和工具，以及如何完善游戏的想法。



我们绝不希望内容创作者花费时间去等待编辑器进行编译，但还是期望增量编译能够在合理的时间框架以内。只要我们确认构建成功的可能性较大，就可以把编译安排在停工的时间段，例如晚上或者早上的第一件事。此外，Epic 还使用 Xoreax IncrediBuild 来大幅加快编译的时间，尽可能减少对团队的影响。

界面复杂性

Visual Studio Express 等专业工具往往让新手用户望而生畏——他们在工作流程中发现这些工具的界面并不直观，还有许多令人困惑的选项，这些都不利于他们的工作。所以我们应该为用户创造简化的界面。

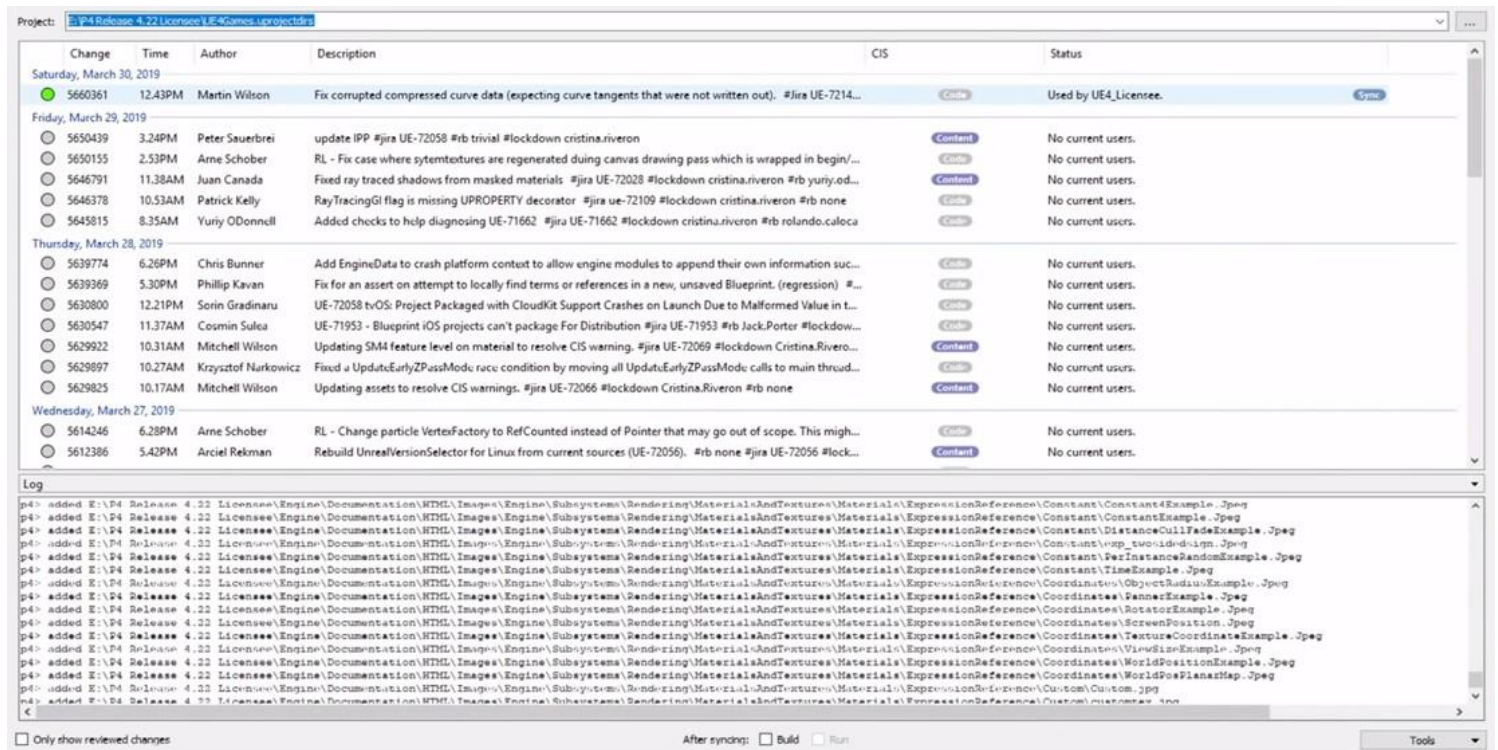
标记问题

如果我们要为下游的开发人员减少一些查错的流程，就需要在出现问题时能够简单快速地进行标记。就《堡垒之夜》而言，我们的团队遍布全球多个时区，在不同的工作空间分工协作，所以我们对《堡垒之夜》采用的任何解决方案都必须包含一套适用于这种类型团队的通知系统。

推行解决方案

向《堡垒之夜》这种规模的团队推行这样的工作流程需要经历一次剧烈的文化转变，而且必须是一次“孤注一掷”式的转变。不过当时采用的流程既缓慢又繁琐，所以我们也不会有什么太大的损失。

我们的 IT 部门利用一个周末为《堡垒之夜》内容创作者部门的电脑安装了 Visual Studio Express，还有一个内部的自定义工具，名叫 UnrealGameSync (UGS)。最近 Epic 专门为这种全新的工作流程开发了 UGS。



图片 3: UnrealGameSync 可以提供用于同步 UE 项目的图形前端，也可以用微软的 Visual Studio 编译器构建这些项目。此图片来自早期版本的 UGS。

在概念上，UGS 可以促进合作环境中代码和内容的整合。它可以显示提交到源码管理中的实时改动流，并且提供精简的界面，方便同步和构建改动，看到其他用户在做什么，同时标记出发现的问题。

在 UGS 中，内容创作者可以从自己的本地硬盘中选择项目。UGS 会自动检测到项目的 Perforce 设置和使用的工作空间等等，然后获取正在检入并会影响到项目的改动列表。

美术师只要双击一项改动，就可以在本地同步和编译需要的文件，然后启动生成编辑器。

合作

成功编译之后，项目内的改动列表会被绿色的标志标记，这样项目的其他人员也能够看到。如果编译失败，就会显示一个红色的标志。

如果编辑器在启动时崩溃或者出现其它问题，美术师可以返回 UGS 并将该改动标记为错误，然后备注具体情况。这样可以工程师团队知道该从哪里诊断问题。

工程师在调查别人标记的问题时，他们可以把它标记为“正在调查”。这样就可以将所有后续的改动都标记为错误，直到他们再次将其标记为可行。

安排构建

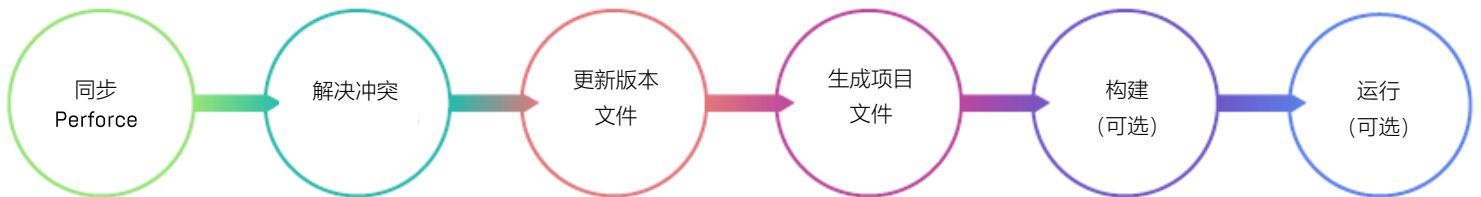
首要的功能之一就是要让用户可以自行制订时间表，从而同步分支，并且在数小时之后触发构建，这样就可以保证工作人员早上来到办公室后就可以使用最新并且稳定的编辑器。

安排的同步版本通常要使用最新版本的代码，并且要有绿色的标志，表明这是一个成功的构建。

其它应用

工程师还可以使用 UnrealGameSync 来更新用于构建编辑器的版本元数据。

这样，他们就可以使用自己本地编译的二进制文件来修改内容，而不是提交自己的代码并等待更新构建之后再修改内容。



图片 4: UnrealGameSync 工作流程

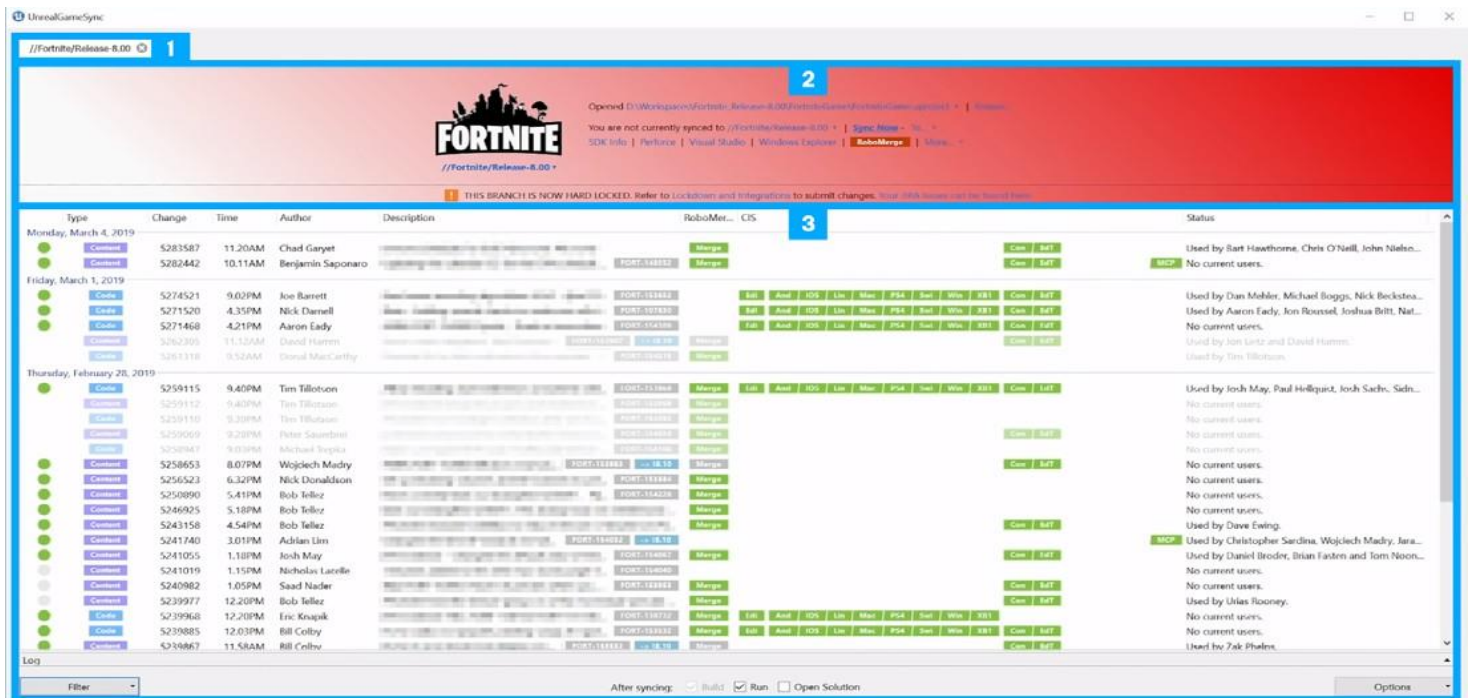


《堡垒之夜》 | Epic Games

集中反馈

UGS 很快就成为了工作流程的主要部分，它确实可以帮助恢复快速迭代周期，这对快速生产来说至关重要。

虽然那还是四年之前，但直到现在《堡垒之夜》团队依然在使用 UnrealGameSync。



图片 5: 《堡垒之夜》团队自定义的 UGS (详解见下表)

随着我们不断和《堡垒之夜》团队完成迭代，我们还添加了其它的几个功能。下表中的内容为图片 5 中数字区域的详解。

UGS 区域	功能详解
1	工作人员可借助多个选项同时查看不同的项目和流送。
2	顶部的状态面板显示的是选择的项目，并且可以使用其它的工具。这个面板还会显示当前分支需要的 SDK 版本。 用户可以轻松切换不同的流送。Perforce 的快速换流功能一直被诟病很容易出现错误，但是通过这种工具就可以更加稳定地运行。
3	右边的标记来自构建系统，例如显示是否在验证构建、是通过还是失败。用户只需点击标记即可直接打开构建日志。 因为 UnrealGameSync 会一直在后台查询分支中的改动，所以它知道何时检入了新内容。如果标记从绿色变为红色，UGS 会跳出提示，告诉开发人员提交的改动列表破坏了构建。

表格 1: UGS 区域和为《堡垒之夜》团队添加的功能

如需深入了解这些功能的详情，请查看 [UGS 参考](#)。

针对《堡垒之夜》，我们会在构建农场中持续进行构建循环，生成带有标记的 UnrealGameSync。

- 每过 10 分钟，我们会递增式编译编辑器。它禁用了统一构建功能，从而减少各个改动的表层区域，这样会抓到提交的代码中丢失的 `#include` 指令。
- 当编辑器被编译之后，指令会加载自上一次运行以来的改动内容。我们发现一个很常见的错误源头就是美术师会忘记提交一段内容，而尝试载入时就会快速生成描述此类问题的错误和警告。
- 在检查内容之后，我们会进行一次快速的自动测试，生成编辑器两次——一次带有 `-game` 参数，还有一次带有 `-server` 参数。这样的测试可以检测客户端和服务器是否可以彼此沟通，确认登录流程和匹配过程是否正确运行，以及玩家能否顺利开始对局。它还会检查玩家在《堡垒之夜》所有四个基本的动作：移动、射击、建造和收集。
- 还有第二台电脑会为所有目标平台上的游戏进行增量编译。

事实证明，用两台电脑运行这些测试极具价值，并且可以在快速反馈和广泛覆盖之间找到很好的平衡。



《虚幻争霸》 | Epic Games

《虚幻争霸》是我们尝试使用 UnrealGameSync 的下一个项目。《虚幻争霸》是一款多人在线竞技游戏 (MOBA)，它拥有可以快速延展到大型美术团队和大量内容的工作流程。设计师使用预设的蓝图功能完成了大部分的游戏玩法逻辑编写，所以设计和工程之间的迭代就不像《堡垒之夜》那么重要。

在如此庞大的美术团队中，让美术师去编写自己的二进制文件肯定不利于产出。但是我们想要保留 UGS 的其它几个优势：工程师可以保存版本管理良好的内容、显示构建运行状况的反馈，并且让预构建的二进制文件留在工作空间之外。

压缩的二进制文件

这个解决方案是一种模式，来自构建服务器的编辑器二进制文件会被自动压缩成 ZIP 格式的文件，并且提交到分支以外的位置。提交压缩文件之后，内容创作者可以选择使用 UGS 下载并解压，无需管理第二个工作空间或进行本地构建。

这些文件是针对特定匹配的改动列表编号发布的，它会有效地匹配源构建的工作流程。任何没有匹配的预编译二进制文件的改动都会在界面中显示为灰色。拥有匹配的二进制文件的改动在获得准许之后会同步仅限内容的改动。

如果用户不想要预编译的二进制文件，可以选择同步和编译。

客户端过滤

《虚幻争霸》项目中，我们和许多非现场的承包商和外包工作室展开合作，这本身就会带来挑战。《虚幻争霸》有许多高分辨率的角色资产，而开发人员通常一次只需要一部分资产来完成他们的工作。如果工作人员要使用 VPN 连接，同步完整的角色资产，那肯定会浪费带宽，并且也非常耗时。



《虚幻争霸》 | Epic Games

为了解决这个问题，我们首次尝试在 Perforce 上创建虚拟流送，这样可以将用户的工作空间缩小到目录的子集，而不是整个分支。这在小规模工作中效果不错，但是 Permutation 的数量突飞猛涨。不断发布更新也意味会有大量的分支，而内容创作者会想要每个分支都有自己的定制内容。

UnrealGameSync 不是在 Perforce 服务器上管理这些 Permutation，而是支持客户端过滤，它会向服务器查询流送的内容，并且选择性地同步（和移除）文件，只留下用户需要的内容。

这个界面会显示可自定义选择的一系列定制组件，用户可以使用自己的过滤器添加或排除补充的组件。例如，用户可以排除当时不相关的所有平台，或者过滤掉动画内容或特定的本地化资产。

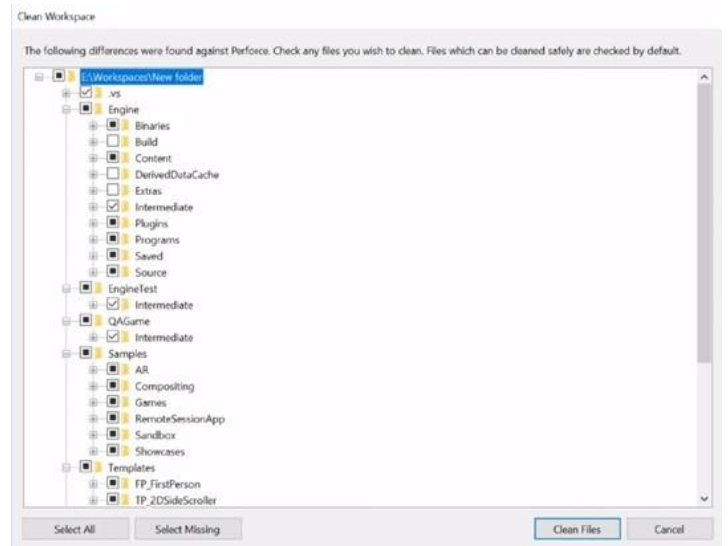
传统的 Perforce 客户端配置会对每个组件都采取集中化配置，相比之下，这些可选择的组件更具优势。因为定义组件的配置文件会提交到源码管理上，这样就可以为项目中的所有人自动更新过滤器，而无需针对个别用户进行显式配置改动。

清理工作空间工具

虚幻引擎的目录布局拥有中间文件，主要是针对插件和散布在整个源代码树中的游戏项目。UGS 中的工作空间清理工具可以用来清理中间文件，还能找到工作空间中丢失的文件。

所有本地工作空间和服务器中的差异都会显示在源代码树中，并且默认情况下选择的都是可以安全删除的中间文件。这就意味着

该工具不会意外删除你已经创建但却忘记检入的工作成果。



图片 6: 工作空间清理工具

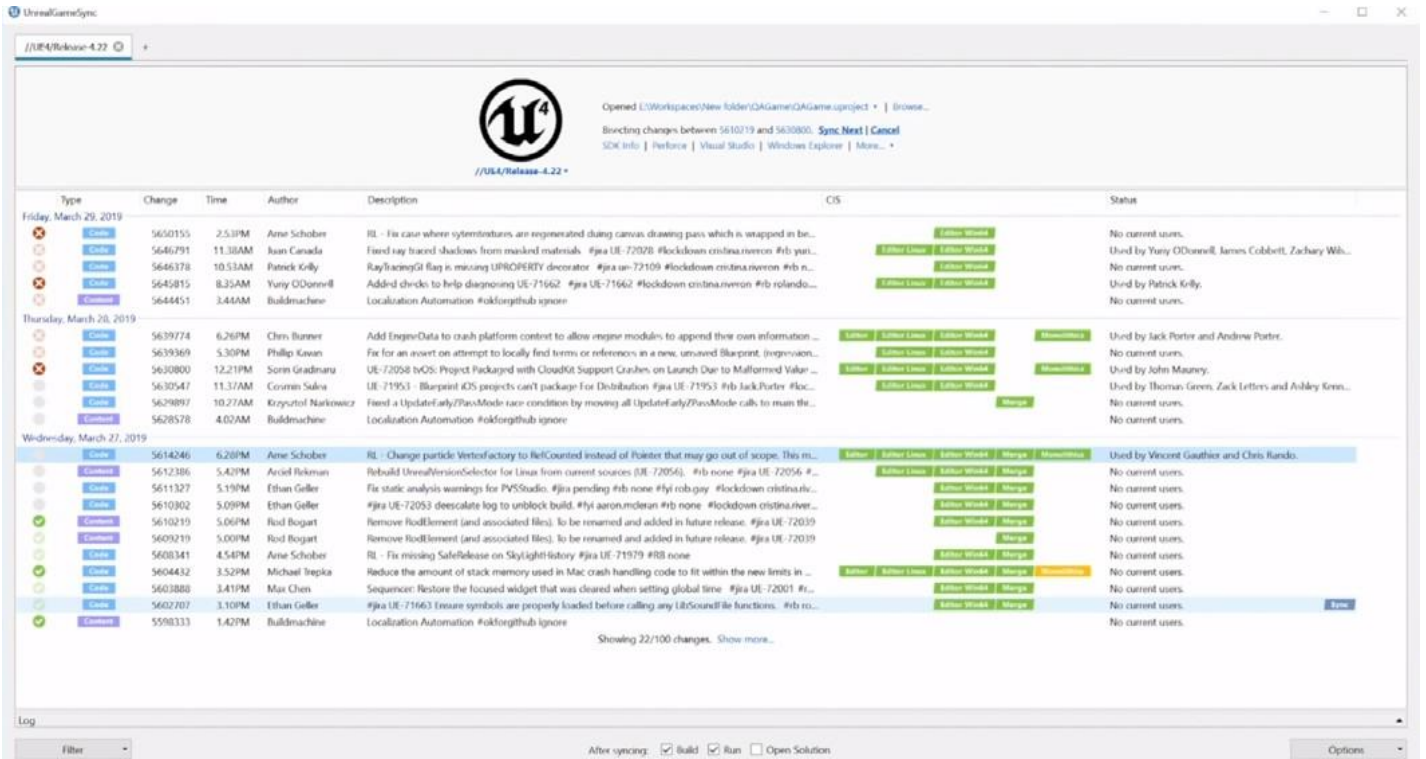
与 Perforce 的清理指令不同，工作空间清理工具只会检查预期位置是否有文件，然后再检查文件的长度和只读状态。根据你电脑的性能，整个过程只需要 10 到 20 秒，甚至都不够你去倒一杯咖啡的时间。

虚幻引擎

UGS 工作流程的优势不仅体现在游戏项目上。虚幻引擎的质量测试部门在做回归测试的时候也会使用 UGS。

平分模式

通常质量测试人员知道可行的改动列表和错误的改动列表之间会有功能上的回归错误，他们需要尽可能高效地将范围缩小到有问题的改动列表上。通过 UGS 面板，测试人员可以选择一系列的改动，然后运行平分模式。这个模式会过滤视图，使其只包含选定的改动。用户可以在这个视图将改动标记为可行或错误。



图片 7: UGS 平分模式

如果测试人员同步了更改和测试，并且没有立即找到错误，就可以选择“同步下一个”（Sync Next）指令，UGS 将会同步至上一个可行和上一个错误改动中间的位置，帮助测试人员执行二进制搜索，找到引入了错误的改动。

配置文件

最后，我们添加了对项目非常有用的自定义功能。

在状态面板下面有一个每日消息功能，它通常可以提供相关信息，告诉我们使用分支的日期。每日消息功能还会提供一个链接，让你可以查看我们错误数据库中的错误列表。

我们还有一个 Perforce 触发器，它要求提交的改动描述中必须包含错误编号。我们设置 UGS 去解析信息并创建一个链接，它会返回到错误追踪软件中相应的错误，这样用户只要在 UGS 中点击改动旁边的标记就可以把它打开。

所有这些功能均由配置文件驱动，并且基于检入到分支中的配置文件。这样即可根据不同项目的具体需求来调整 UGS。

在你的项目中使用 UnrealGameSync

《堡垒之夜》的成功依靠的是我们团队成员高效合作、彼此沟通的能力。我们在《Battle Breakers》和《虚幻争霸》遇到的工作流程方面的挑战让我们开发出了 UGS，它有助于我们将在同一空间中办公的优势延展到全球分工的开发环境中。

我们发布了虚幻引擎最新版本的 UGS，你可以通过任意版本的虚幻引擎来使用它。它的源代码可以在 Perforce 和 GitHub 上找到，相关的使用详情请查看我们的虚幻引擎[最终用户许可协议](#)。

要想了解更多关于设置 UGS 的信息，你也可以前往虚幻引擎官方文档，查看 [UGS 参考](#)。

作者

Ben Marsh

Robert Gervais

编辑

Michele Bousquet

Su Falcon

布局和媒体设计

Jung Kwak

