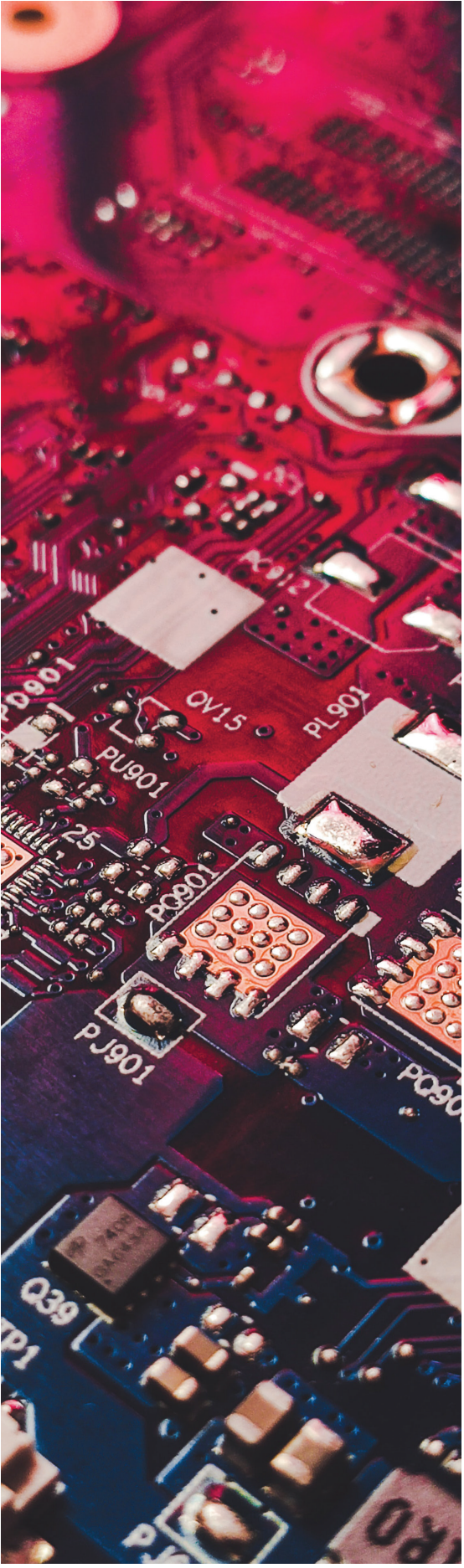
バグに侵されないゲームを作る：より良いゲームを、より速く制作するために

最初からバグを最小限に抑え、時間とコストを節約し、信用を確保。

unity

コンテンツ



クラッシュに（今以上に）気を配るべき理由

ゲームというものは、ほとんどがクラッシュしたりハングしたりするものです。QA 部門やテストシステムがいくら優れていても、ゲームプレイ時に起こり得るシナリオをすべて予測することなど不可能です。しかし、一度ゲームがリリースされてしまえば糾弾は避けられません。

PC ゲームは、言い尽くせないほど多様な環境でプレイされます。グラフィックスカードの種類や RAM の容量が違ったり、マシンが必要最低限のスペックを満たしていなかったりします。コンソールや固定プラットフォーム用のゲームであっても、製品がどのように使用されるかを予測しきるのは不可能です。プレイヤー数が増えるほど環境は多様になり、必然的にクラッシュも増えます。ただし、長期的に見れば、これらのクラッシュの修正コストはきちんとかけるべきです。

クラッシュによる悪影響は、指数関数的に膨れ上がります。ゲームがクラッシュすると、プレイヤーは怒り、離脱が発生し、収益が減少、ソーシャルメディアのフィードバックもガタ落ちとなって、そのゲームは成功のチャンスを逃してしまいます。

このような評判の低下を回復させるのは困難です。完璧なリリースなど存在しませんが、それでも、こういったクラッシュを引き起こす問題を見つけて修正する技術がなければ、エンジニアが作品に注いだ努力が水の泡となってしまいます。

過去のリリースから学ぶというのが一般的なアプローチですが、それと並行して次世代のベストプラクティスを導入することも、リリースを最大限にスムーズにする手段として一考の価値があります。クラッシュからデータをキャプチャし、重複排除とクラッシュのグループ化を行って、クラッシュ分析をさらに掘り下げ、エラー対応のためのワークフローを構築するという方法です。

クラッシュの発生しないゲームなど、ほとんど伝説と言っていいものですが、匙を投げて運任せでゲームを制作するべきではありません。この e ブックでは、ゲームがバグに侵されることを防ぎ、より良いゲームをより速く制作できるようにする、次世代クラッシュ検出テクノロジーの実装についてご説明します。

次世代の分析方法

次世代のクラッシュ分析は、連動する 4 つのステップから成り立っています。関連するデータのキャプチャ、そのデータの重複排除とグループ化、分析による根本原因の特定、時間を賢く使うワークフローの実装です。

おそらく、すでに自社のクラッシュ検出テクノロジーや、無料のクラッシュ分析機能を使用した経験のある方も多いでしょう。もうお気づきでしょうが、これらのシステムには固有の問題が多く見られます。たとえば、メンテナンスサイクルの長さが原因で、新しい機能をすぐに利用できなかったりします。クロスプラットフォームのサポートも欠如しています。複数のプラットフォームを使用していると、それぞれがサイロ化することもよくあります。このためゲームデータをバージョン間で比較することがますます困難になり、作業が増えてしまいます。

これらのシステムは、元はソリューションであったはずなのに、往々にして重荷となります。それは個々のクラッシュに対して、何時間も原因究明に取り組む羽目になってしまうためです。次世代のクラッシュ分析テクノロジーは、アルゴリズムとルールベースのエンジンで複数のクラッシュをグループ化および診断し、小さなサブセットを作ることで、勘に頼らない作業を実現します。

次の章からは、次世代のクラッシュレポートのさまざまな切り口を詳しくご紹介し、エラーと戦う際にそれぞれが連動して強力な時間節約ツールとなる仕組みをご説明します。

重要なデータをキャプチャする

次世代のクラッシュ対応の第一歩は、適切なデータをキャプチャすることです。これはクラッシュレポートの基盤であり、プロセス内のすべてのステップに影響します。クラッシュダンプやログファイルを眺めるのは最初の一歩にすぎません。他にも考慮すべき情報源があります。

まずはクラッシュを報告したプレイヤーの OS 、ドライバー、システム情報を知るのが重要です。メモリの使用量と、使用しているアプリケーションのバージョン情報も欠かせません。また、ゲームの内部データも忘れてはいけません。ゲーム ID、セッション ID、プレイされていたステージなどの情報があります。

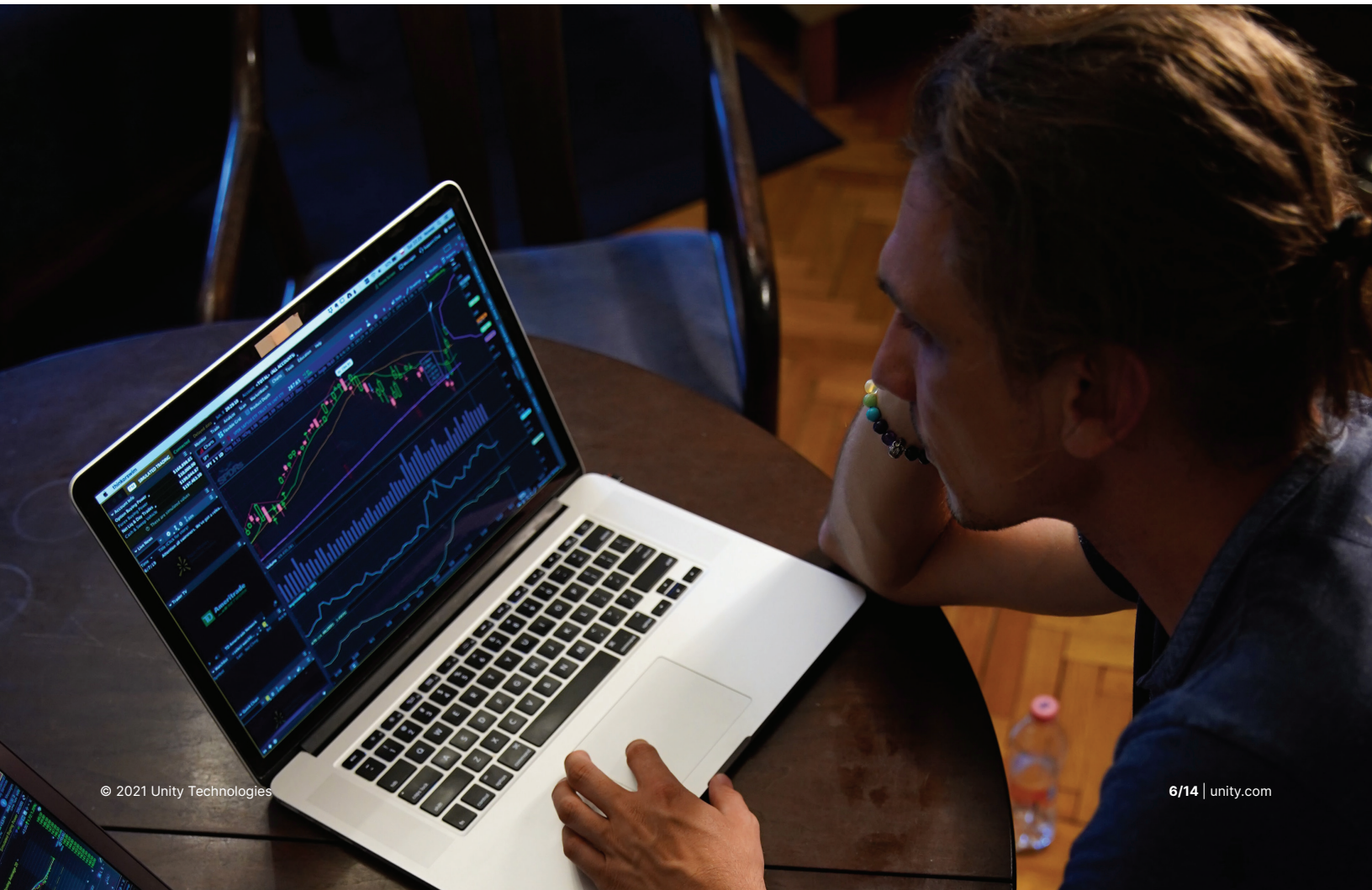
最新のプラットフォームであれば、クラッシュそのものを再現したスクリーンショットやビデオなど、もっと有用な情報を入手することもできます。クラッシュの根本原因に到達するための手がかりは、どれも非常に役立つものです。

これらのデータを活用すると、クラッシュに優先順位をつけやすくなり、調査も詳細にできるようになります。たとえば、あなたが PC ゲームをリリースしたばかりで、他のプラットフォームで問題が発生する前に、クラッシュを確実に解決しておきたいとします。このようなとき、データのキャプチャが役立つことがあります。

このようなデータをすべて収集して照合しておくと、エンジニアがバグを修正する際にも非常に便利です。仮説を立ててテストを実行しようとしたのにデータが見つからないという状況は、非常にストレスがたまります。

複数のクラッシュを引き起こしている可能性のあるユーザーに注目できるというのも、データ収集のメリットです。DRM システムのバイパスや、チート行為のためのアクセスを試みているユーザーは、平均的なユーザーよりも多くのクラッシュに遭遇する可能性が高いと言えます。最低限の要件を満たしていない環境でゲームを実行しているユーザーについても、同じことが言えます。

適切なデータを収集すれば、余分な情報を削ぎ落し、最も重要なクラッシュを発見して、優先的に対応できるようになります。しかし、次世代のクラッシュ分析ではさらにもう一歩踏み込んで、クラッシュをグループ化し、より効率的かつ効果的なワークフローを作り出すことができます。



データの重複排除でより良い結果を入手

ゲームから生成される数百、数千ものクラッシュレポートを確認して、特異なものを識別するというのは、人間にとってはあまりに重労働です。そのうえ、膨大な時間とリソースが吸い取られます。だからこそ、次世代のクラッシュ分析には、重複排除とバケット化のテクノロジーが備わっています。

重複排除では、関連するセットや基になっているグループに、クラッシュがマッピングされます。2 つのエラーを個々の現象として扱うのではなく、1 つのフィンガープリントに結び付けるのです。そしてこのフィンガープリントは、問題の根本原因に直接マッピングされています。

これで、数千もの個々のエラーではなく、数百のフィンガープリントが手に入ることになります。重複排除とは、データを手動でより分けなくとも、クラッシュレポートの焦点を絞って優先すべき問題を見つけられるようにする手段のことです。バケット化によってクラッシュをグループ分けすれば、根本原因の発見とエラーの修正が簡単になります。

さて、クリアしなければならないことがまだ残っています。どうすれば適切にグループ化できるかを知ることは重要です。クラッシュのコールスタックをすべて使用することは困難です。根本的には同じクラッシュであっても、プラットフォームや OS、場合によっては OS のバージョンが違うだけで、開始フレームが異なる場合があるからです。

クラッシュログの最深部のフレームを使用しても上手くはいきません。なぜなら、そのフレームに達するまでのすべてのパスを表示しきれないからです。イベントベースの同時実行システムでは、タスクが何通りもの方法で発生するため、複雑さはさらに増します。

次世代の重複排除では、ルールベースのエンジンにより、クラッシュに最大限迫ることができます。問題の箇所に辿り着いたら、革新的な方法でクラッシュデータを視覚化できます。これにより、さまざまな情報セットの類似点と相違点を理解し、分析できるようになります。データをグループ化するアルゴリズムの存在によって、プラットフォーム、問題のないフレームの存在、さらには言語などに基づいた調節が実現します。

次世代のクラッシュレポートによって、エンジニアが手がけていたグルーピングの手間が解消され、細かいデータを延々と確認する手動作業も不要になることがわかりました。これで、ゲームにとって最も大きな問題に集中的に取り組めます。3 つ目のステップでは、新しくグループ化されたデータをさらに詳しく分析し、その結果に基づいて行動を起こします。

分析：
本当の力

次世代のクラッシュレポートの本当の力は、フィンガープリントを分析したときと、重複排除によって作成されたグループを分析したときに発揮されます。これは、入手したクラッシュレポートにどのように優先順位を付け、調査するのかに関わります。

中には優先的に解決すべきクラッシュもあります。分析機能を使用すれば、修正不要な問題を除外できます。それが悪用やチートをしているプレイヤーであっても、最低限の要件を満たしていない環境でゲームを実行しているプレイヤーであっても、分析によって余分な情報を切り分け、プレイヤーに影響しているバグやエラーに優先順位を付けられるようになります。

分析を使用することにより、視野を広く保った「鳥の目」で仮説を検証することができます。大切なのは優先順位付けです。つまり、エラーが発生した頻度や時期、クラッシュした時点でのサーバーのロードに注目するということです。

このアプローチは、問題を調査するときにも便利です。収集したデータは、非常に細かいものになる場合があります。プレイヤーが使用していたゲームマップ、同じクラッシュが発生したプレイヤーのメモリの平均使用率、クラッシュが起動時にのみ発生するかどうかなどです。

重複排除が十分であれば、上位 5 つのクラッシュを選び、優先的に診断や修正を開始できます。最も悪影響を及ぼしているクラッシュを把握できれば、プレイヤーの満足度が向上するだけでなく、エンジニアのストレスも最小限に抑えられます。次世代のクラッシュ分析は、時間を節約できるだけでなく、チームの健康を守るのにも役立ちます。

ここで、優先度の高いクラッシュが必ずしも頻繁に発生するとはかぎらないということを思い出しておきましょう。これまでに述べた要因の多くは、原因の一部ではあるかもしれません。しかしそれは、ドライバーの更新や、ウイルス対策ソフト、その他の制御できない物事でも同様です。重複排除と分析を組み合わせれば、新しく発生したクラッシュが、すでに取り組みを進めている別のエラーとよく似たものなのかどうかを見分けることができます。

クラッシュがプレイヤーに及ぼしている影響も考慮する必要があります。ゲームのシャットダウン中にのみ発生するバグは、セッションの最中に発生するものよりも優先順位は低くなります。また、サーバーのクラッシュは多くのプレイヤーに影響がある一方で、発生は 1 回きりであることが大半です。

クラッシュの数を減らすために重要なのは、すべての情報を把握し、それらの意味を理解することです。これらは両方とも、次世代のクラッシュ分析テクノロジーによって大幅に容易になります。そうすれば、問題の修正に集中して、プレイヤーに優れたゲーム体験を提供できるようになるでしょう。

大切なのは ワークフロー

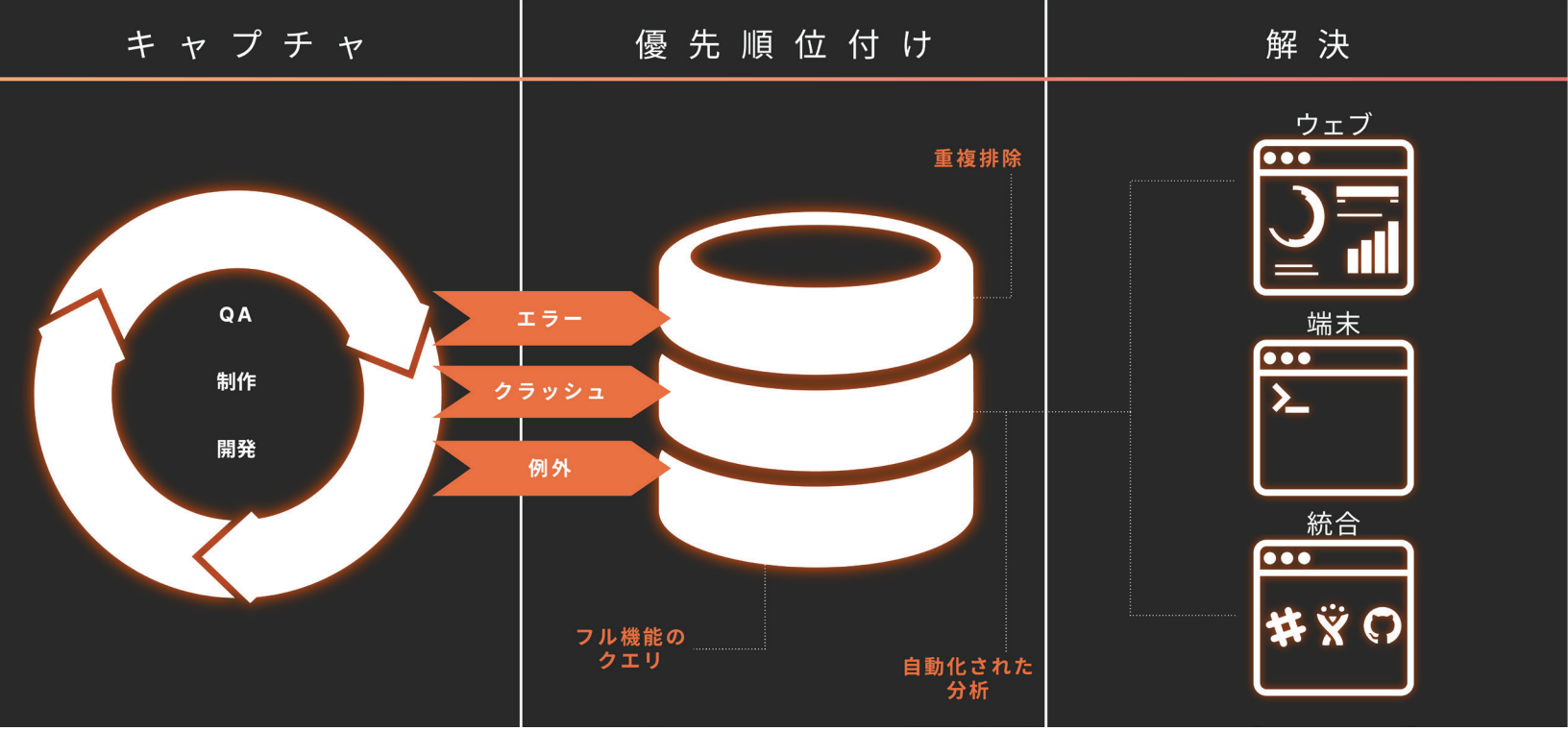
最後に行うのは、今までにご説明したステップを組み合わせ、わかりやすくて管理も簡単なワークフローを作成することです。以前なら、チームの誰かが全体像を見渡したり、クラッシュ率のグラフを毎週見比べたりしていましたが、次世代のクラッシュ分析ではそれらが大幅に効率化されるため、クラッシュの修正が全社的に高速化されます。

たとえば、すでに対処済みのフィンガープリントでクラッシュが発生した場合、余分な警告を追加する必要はありません。代わりに、新しいクラッシュのグループを優先的に見て、発生時の対応や発生のタイミングを確認します。一般的なメッセージを送信するのではなく、クラッシュがグループ化されたコンテキストを使用して、サポートチームによる対応を支援できます。

クラッシュの調査に必要なすべてのデータの照合とアクセスは、1 つのシステムから行えます。クラッシュのグループはチケットにリンクされているため、複数のシステムを更新する必要はありません。クラッシュの情報に基づいて、インテリジェントな警告を設定しましょう。基にする情報は、コールスタックや、地理的地域だけでなく、その他の情報でも構いません。いずれにせよ、対応の高速化が自然に実現します。

次世代のクラッシュ分析では、ワークフローを逆向きに辿ることもできます。最初は、消費者向けの最終製品、つまりゲームの改善のために使い始めたかもしれません（もちろん、プロセス中で最も重要な部分です）。しかし、QA システムや開発システムと組み合わせることもできます。それどころか、サーバーの監視に使用してもよいのです。

ワークフローを改善すると、無駄なサイクルが減り、それによってストレスも軽減され、チームの幸福度とプレイヤーベースの満足度が向上します。これこそ、誰もが求めている状態です。



Backtrace とともに次世代へ

クラッシュはゲームを台無しにします。チームにとっても、プレイヤーのコミュニティにとっても有害です。対応を誤れば、制作スタジオやゲームの評判を落とすに留まらず、チームの士気に関わります。だからこそ、次世代のクラッシュ分析がきわめて重要です。1 人プレイ用オフラインゲームの制作であれ、サービスとしてのゲームの制作であれ、クラッシュの制御はワークフローに必ず組み込むべき要素です。

かつては手動でのクラッシュ検出と比較に途方もない時間が費やされていましたが、次世代のクラッシュ分析ではその時間が劇的に短縮されます。クラッシュログを隅から隅まで確認して、比較と対照を行うのではなく、ソフトウェアに作業をさせ、かつてない量の情報を基に優先順位付けと分析を行うことができます。

ゲームの機能をかきわけてクラッシュを探さなくても、アルゴリズムとルールベースのエンジンで余分な情報を切り分け、最も影響の大きなクラッシュに集中できます。時間のかかるデータのスパイラルにエンジニアを送り出さなくても、すでに無駄な部分の削ぎ落された重要な情報が手に入り、チームが検出ではなく修正に力を注げるようになります。

適切なデータのキャプチャ、重複排除とフィンガープリントによる共通原因のクラッシュのグループ化、新しい革新的な方法でのデータ分析、今までよりもはるかに効率的なワークフローの作成。次世代のクラッシュ分析は、エラーとの戦いになくてはならない重要なツールです。今こそ、クラッシュをより真剣に受け止めて、それらが引き起こす悪影響を認識し、スマートかつ効率的な最新手法で対処するべきときです。

トロルや悪質なレビューにあふれたこの世界では、魅力を発揮しきれないうちにゲームが台無しにされかねません。時間は貴重なリソースの 1 つです。次世代のクラッシュ分析を、ぜひエラー検出戦略の強固な基盤としてください。次世代に一歩踏み出せば、時間とコストを節約し、信用を守ることができます。どれもけっして、失ってはならないものです。

Unity の公認ソリューションパートナーである Backtrace の支援を得て上記を実現する方法の詳細については、こちらをご確認ください。 今すぐ無料で利用したい方は、こちらからお試しください。

Unity の DevOps ツールも確認し、より高い品質を効率的に実現する方法をお確かめください。


unity