

# DE LA **CREACIÓN**

---

# A LA **MATERIALIZACIÓN**

**ANÁLISIS DEL PIPELINE EN  
LA INDUSTRIA DE LOS VIDEOJUEGOS**

**JACQUELINE SOLER SANTOS**



**TRABAJO DE FIN DE GRADO**

**GRADO EN BELLAS ARTES**

**UNIVERSIDAD DE SEVILLA**

**2017/2018**



# **TRABAJO DE FIN DE GRADO**

GRADO EN BELLAS ARTES-UNIVERSIDAD DE SEVILLA

2017/2018

*DE LA CREACIÓN A LA MATERIALIZACIÓN  
ANÁLISIS DEL PIPELINE EN LA INDUSTRIA DE LOS VIDEOJUEGOS*

Jacqueline Soler Santos

Tutorizado por José Luis Molina

Vº. Bº. DEL TUTOR





**DECLARACIÓN DE RESPONSABILIDAD Y AUTORÍA DE TRABAJO ACADÉMICO**  
**TRABAJO DE FIN DE GRADO/TRABAJO FIN DE MÁSTER**

D. /D<sup>a</sup>. : \_\_\_\_\_, con DNI \_\_\_\_\_,  
Estudiante \_\_\_\_\_ del \_\_\_\_\_ Grado/  
Máster \_\_\_\_\_ de la  
Universidad de Sevilla.

**DECLARA QUE:**

El Trabajo Fin de Grado/Trabajo Fin de Máster denominado,

\_\_\_\_\_  
\_\_\_\_\_

Es de mi exclusiva autoría y ha sido elaborado respetando los derechos intelectuales de terceros, conforme las citas que constan en las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Por lo tanto, asumo la responsabilidad de sus contenidos y elaboración, sin incurrir en fraude científico o plagio.

Para que así conste, firmo la presente declaración en

Sevilla a \_\_\_\_\_ de \_\_\_\_\_ 201\_\_

Fdo. \_\_\_\_\_



# ÍNDICE

1. Introducción.....	8
1.1 Metodología y objetivos.....	10
1.2 El pipeline: etimología, características y aplicaciones en la industria.....	12
2. El pipeline creativo y su importancia.....	16
3. Organización y procesos artísticos en los videojuegos.....	22
3.1 Pre-producción.....	26
3.2 Producción.....	32
3.3 Finalling.....	38
3.4 Diferencias entre pymes y grandes empresas.....	42
3.5 Errores comunes y cómo evitarlos.....	46
4. Conclusión.....	52
5. Bibliografía.....	56

# INTRODUCCIÓN



En esta industria audiovisual tan extensa, compleja y heterogénea donde artistas y técnicos de sonido, imagen, managers, informáticos, programadores, ingenieros de sistema, productores y muchos perfiles profesionales más se ven obligados a trabajar juntos en el mismo proyecto bajo unas fechas de entrega y una presión muy intensa donde un solo fallo puede acarrear importantes pérdidas económicas existe un recurso que si bien no tiene por qué asegurar el éxito sabemos a ciencia cierta que determina el fracaso si no se hace correctamente. Dicho recurso es el pipeline .

“El pipeline es el pegamento que mantiene unido el trabajo de cada artista involucrado en la producción” (Dunlop, 2014:28).

Aunque posteriormente trataremos con mayor amplitud este concepto, un pipeline es todo el proceso que conlleva la creación de un proyecto, desde la idea más embrionaria al resultado más sofisticado, pasando por la creación, el diseño, la organización y el marketing. Todas estas variables dan lugar a una serie de problemas de tiempo, espacio, infraestructura y organización que el desarrollador y el productor deberán contemplar para evitar que todo el proyecto se convierta en una catástrofe.

---

1

*La palabra “pipeline” es originaria del inglés y significa, literalmente, “cañería”. Dado que la acepción concreta sobre la cual trabaja este TFG no tiene traducción en castellano, en ocasiones nos referiremos a ello como “el proceso de producción” o “el proceso organizativo”. No debe ser confundido con los procesos organizativos o de producción que tienen lugar dentro del propio pipeline.*

# **METODOLOGÍA Y OBJETIVOS**

En este Trabajo de Fin de Grado se tratará el tema del proceso de producción desde un punto de vista organizativo. Para ello recurriremos a múltiples fuentes, desde libros sobre el tema a entrevistas, conferencias, simposios y vídeos con las que elaboraremos un compendio de los diferentes procesos audiovisuales que existen actualmente en la industria de los videojuegos y las múltiples diferencias que existen entre ellos. Podemos reducir los objetivos a los siguientes:

- Establecer una definición clara de pipeline
- Analizar los procesos más comunes en la industria audiovisual
- Explicar cada proceso único del pipeline para videojuegos en profundidad
- Observar y catalogar las distintas diferencias entre dichos procesos según el volumen de trabajo y el presupuesto que manejan
- Mostrar de forma práctica la variación que sufren los proyectos según el pipeline al que están sujetos
- Subrayar los fallos más comunes que pueden darse en los distintos procesos
- Otorgar al lector un recurso que podrá usar de forma práctica si decide trabajar en la industria audiovisual

# **EL PIPELINE: ETIMOLOGÍA, CARACTERÍSTICAS Y APLICACIONES EN LA INDUSTRIA**

“En la actual industria de cine y videojuegos, la producción artística es una empresa común. Cada elemento de CG que la audiencia ve (o no se da cuenta que ve) en un cine, y cada personaje o escenario que un jugador encuentra en un videojuego, es el resultado del esfuerzo colectivo de muchos artistas y técnicos. Los desarrolladores de pipeline son la materia oscura que mantienen unida esa constelación de artistas” (Dunlop 2014:26)

Nuestro primer objetivo será establecer una definición de pipeline. Se trata de un concepto muy abstracto que varía mucho según sus aplicaciones, y tiene muchas. Cualquier empresa tiene un sistema de producción integrado, ya sea videojuegos, animación, efectos especiales, publicidad. Hasta las empresas de contabilidad tienen un sistema para evitar retrasos y poder controlar los documentos que se generan. Cualquier ente corporativo que requiera de una sistematización de procesos y de una jerarquía para funcionar es susceptible de tener un pipeline más o menos adecuado.

Pero, ¿qué es exactamente? Según Renée Dunlop, “es, realmente, la expresión técnica de una compleja red de actividades humanas”(Dunlop 2014:143). Aunque esta definición no es

muy detallada, es la más universal que podemos obtener. Un pipeline no es sino el conjunto de procesos y actividades realizadas entre el punto de inicio de un proyecto y su finalización.

Según Renée Dunlop, “es, realmente, la expresión técnica de una compleja red de actividades humanas”(Dunlop 2014:143)

Por tanto, es importante aclarar que el pipeline no es típico de los procesos audiovisuales, sino aplicable

a casi cualquier tipo de proyecto que requiera un mínimo de trabajo grupal. Su significado, “cañería”, hace referencia a la forma en que se estructura dicho proceso organizativo. Para mostrar esta referencia, procederé a explicar qué dos tipos de pipeline se usan actualmente en el mundo laboral.

“El método más tradicional y directo es el modelo de cascada [...] también conocido como de oportunidad única o de sentido único. En este modelo, el responsable del proyecto planea el proyecto de principio a fin y ese proceso es ejecutado tomando los mínimos pasos hacia atrás posibles” (Dongen 2007:14)

Según su planificación, este modelo apenas contempla la posibilidad de que se puedan cometer errores, y es muy pobre cuando es necesario incluir correcciones constantes de superiores u otras formas de feedback. En cierta manera el método de cascada es un pipeline más simple, en el sentido de que toda la información generada va en línea recta y no necesita reescribirse. “El desarrollo en cascada es mucho más rígido y lineal en el cual un producto es creado en fases secuenciales, normalmente definidas como análisis de requerimientos, diseño, implementación, probado, integración y mantenimiento. Cada fase debe ser completada antes de que el equipo pase a la siguiente” (Dunlop 2014:311) Es un modelo

muy estructurado y compartimentado que rara vez se usa en los videojuegos, en los que en la mayoría de los casos se opta por un pipeline iterativo:

“El nombre genérico para la mayor parte de procesos reiterativos es desarrollo ágil [...] La idea del desarrollo ágil no es evitar el cambio, sino gestionarlo tan bien como sea posible [...] El resultado es un bucle repetitivo de diseñar nuevas partes, desarrollarlas, testarlas, evaluar los resultados y re-diseñarlas para incorporar los resultados de la evaluación. Esta iteración continua de diseño, creación, evaluación y cambio es la base del desarrollo ágil” (Dongen 2007:16)

La principal diferencia entre estos dos métodos es la repetición. El método de cascada asume que un solo intento, si bien planificado, será suficiente. El método iterativo o ágil asume que a la primera no sale nunca nada y lo que hay que planificar es cómo hacemos posible el máximo número de correcciones sin ralentizar la producción. Ambos métodos son válidos; sin embargo, lo más normal será encontrar mezclados, en mayor o menor medida, ambos procesos.

Habiendo esclarecido los diferentes tipos de pipeline que existen, podemos establecer que el pipeline más correcto es aquel que mejor se adecúa a las características concretas de cada proyecto; de ahí que cada pipeline varíe tanto de un proyecto a otro como de un género a otro. “El pipeline puede consistir

en un gran número de componentes diferentes que están interconectados de varias formas. Algunas son muy independientes, mientras que otras deben trabajar juntas con, virtualmente, todo lo demás. Qué partes del pipeline son usadas en un proyecto cualquiera varía con los requerimientos de dicho proyecto” (Dongen 2007:7)

“Esta iteración continua de diseño, creación, evaluación y cambio es la base del desarrollo ágil” (Dongen 2007:16)

Como vemos, se trata de un proceso muy importante y, cuanto mayor es la inversión, más importante es la organización del pipeline. Por motivos de complejidad, los únicos sistemas que trataremos en este TFG serán los referentes al mundo de los videojuegos. Sin embargo, habría que dejar claro que cada proyecto tiene un proceso único y que la creación de pipelines tiene sus directrices, pero en la mayor parte de proyectos su labor consiste en buscar soluciones diplomáticas a problemas que podrían potencialmente significar la bancarrota del estudio. Hablaremos de este tema en mayor profundidad más adelante.



# **EL PIPELINE CREATIVO Y SU IMPORTANCIA**



“Para nuestros propósitos, (el pipeline) es la secuencia de operaciones requeridas para mover los activos artísticos<sup>2</sup> desde el proceso conceptual al producto terminado” (Lindeman, n.d.)

El primer paso de todo pipeline, ya sea para un videojuego, una película o una serie es la narración. Este viaje comienza con una idea escondida en la mente de un escritor, guionista o productor que se desarrolló hasta dar lugar a una historia. Afortunadamente para las mentes detrás de este proceso creativo, “la estructura de la historia ha sido el tema principal de clases impartidas de escritura de guiones durante décadas. Es conocida como una fórmula que, aplicada correctamente, puede asegurar la implicación emocional de la audiencia en una película” (Saunders and Novak, 2006:133). La más común es el argumento en tres actos (Planteamiento, nudo y desenlace) pero existen otras alternativas más universales y filosóficas, como podrían ser El viaje del guerrero, escrita por Joseph Campbell, o su contrapartida orientada a la escritura de guiones de Christopher Vogler El viaje del escritor.

Tres fases ordenadas en orden cronológico: la preproducción, la producción y la postproducción.

La historia será indispensable para el pipeline de una película de animación, podrá ser más o menos importante para un videojuego o totalmente superflua en un pipeline de efectos especiales, en la medida en que la historia vendrá dada según el argumento de la película y tendrá

escasa relevancia en las imágenes que se creen en el estudio de VFX. El funcionamiento de la historia es muy exclusivo de cada disciplina concreta de la industria audiovisual.

Una vez que se tiene la historia, existen unos procesos estándares compuestos por tres fases ordenadas en orden cronológico: la preproducción, la producción y la postproducción. Aunque estos protocolos son en teoría universales, las tareas que se realizan en ellos varían mucho según el ámbito hacia el que se orientan por lo que ahora mismo nos limitaremos a explicar grosso modo en qué consisten.

“La pre-producción [...] es la fase de planificación de un proyecto, [...] es donde el alcance completo del trabajo es definido, y los posibles puntos de fracaso identificados” (Dunlop 2014:57)

El objetivo de esta fase para la plantilla artística será establecer un estilo sólido para el videojuego. A partir del guión se evaluarán distintas propuestas de estilos que sean coherentes con la temática y el ambiente de la historia hasta llegar a un consenso. Se genera un documento que servirá para orientar a los artistas de la producción en el que se recogen las especificaciones visuales de todos los elementos que entran en el proyecto: desde la

2

*Debido a la inexistencia de una traducción apropiada, en este TFG nos referiremos a “art assets” como activos artísticos. Podemos definir “art assets” como todo el conjunto de contenido artístico creado en el contexto de la creación de un videojuego o película. Esto comprende desde bocetos hasta animaciones renderizadas definitivas.*

atmósfera general del trabajo hasta los detalles de los accesorios de los personajes secundarios. Durante este periodo se van atesorando documentos que sirvan como referencia para que los artistas sepan a qué atenerse a la hora de diseñar, modelar o animar las historias, especialmente si los elementos que intervienen son arquetipos que pueden tener equivalentes en otras producciones. Además, no es extraño durante esta fase subcontratar a artistas e ilustradores para que den su visión de un personaje o una escena concreta; el producto de su trabajo se añade a esas imágenes de referencia con comentarios sobre los aspectos que puedan ser relevantes para la producción.

Dichos documentos son creados y organizados por los jefes de diseño del departamento de arte, pero no son los únicos. Es el momento de calibrar cómo funcionará todo el proyecto teniendo en cuenta todas las variables. "Si el juego tiene un ciclo de producción corto, el productor debe asegurarse de que la calidad del producto no baja debido a la falta de tiempo disponible. Esto puede hacerse ajustando el presupuesto, aplazando la fecha de presentación, o restringiendo o reestructurando la extensión de las características del juego. El productor tiene que presentar un plan estructurado para todo el proyecto" (Saunders & Novak, 2006:366)

En términos de pipeline esta fase es todo un reto; nuevas ideas se están generando constantemente y deben ser supervisadas y aprobadas o descartadas por un equipo que debe decidir si el proyecto es realizable o si es artísticamente aburrido en un espacio de tiempo

bastante corto. "La única forma de evitar gastar un tiempo y dinero irremplazables en una idea imposible es empezar a validarla desde muy temprano- lo que significa que pipelines prototípicos deben ser organizados en el momento[...]. Los desarrolladores de pipeline necesitan improvisar soluciones a nuevos problemas rápidamente, lo que suele significar probar cosas que el sentido común y las buenas prácticas de escritura de código desaconsejarían" (Dunlop 2014:58) Efectivamente, a veces, dichas soluciones que prometían ser temporales acaban siendo permanentes por falta de tiempo y medios, por absurdo que pueda parecer el resultado.

En resumen, el objetivo de esta fase es dejar muy atado y compartimentado el estilo visual y conceptual del proyecto, allanar el camino para la siguiente fase del pipeline intentando evitar o contener problemas potenciales, crear planes de contingencia para los problemas que puedan surgir y organizar el desarrollo de producción del proyecto.

Es en la fase de producción en la que son contratados la mayor parte del personal. Si la fase anterior era planificación, esta fase se centra en la creación. Con tal cantidad de activos que generar, lo más normal es compartimentar mucho el trabajo y dar a cada grupo tareas muy específicas con protocolos muy concretos: esta fase es la prueba de fuego para comprobar

**Si el juego tiene un ciclo de producción corto, el productor debe asegurarse de que la calidad del producto no baja**

si el pipeline funciona.

En teoría, y si se ha pensado bien durante la pre-producción, en esta etapa encontraremos una cantidad de trabajo que ha aumentado drásticamente y fechas de entrega constantes. Para capear el temporal y evitar que las producciones se demoren más de lo previsto, lo más común es contratar mucho más personal, que deberá ser instruido en las prácticas propias de la empresa, necesitará una infraestructura propia, así como familiarizarse con el resto del equipo, y todas estas tareas llevan tiempo. Por tanto, en esta fase, la principal tarea del pipeline es arreglar problemas sobre la marcha, y la mayor parte de esas dificultades están relacionadas con la automatización. Es muy importante que el protocolo sea evitar el gasto inútil del tiempo de los artistas, especialmente si esos problemas puede resolverlos la propia máquina. Por tanto, la incorporación de un sistema de automatización que derive las tareas repetitivas y mecánicas es primordial, especialmente en grandes producciones. Además, es muy importante que sea contemplada la actualización constante de los documentos que sirven de guías para la producción; si uno está desfasado se podría dar el caso de que un grupo esté trabajando con información desactualizada, resultando ese error en

horas perdidas de esfuerzo. En importante tener presentes esas dos consideraciones durante dicho periodo.

Es muy importante que el protocolo sea evitar el gasto inútil del tiempo de los artistas

Una vez tenido eso en cuenta, el trabajo en este ciclo es mucho más mecánico que en la fase anterior. Básicamente se trata de crear todos los elementos que componen

el proyecto e irlos componiendo juntos progresivamente para asegurarse de que funcionan. Evidentemente, este proceso tiene sus pormenores y varía mucho según el tipo de trabajo que estamos realizando, así que lo veremos en mayor profundidad en los puntos siguientes. Finalmente, llega el proceso de post-producción. Es en esta etapa donde más diferencias se encuentran entre cine, videojuegos y VFX. "Durante esta fase, divergen radicalmente. En una película animada, el proyecto estará casi completado [...] pero en videojuegos, parte del trabajo más técnico aún está por llegar" (Dunlop 2014:66)

Por esta razón esta fase será explicada casi completamente en los puntos siguientes. Sin embargo, debe ser mencionado el único punto en común que tiene este proceso para toda la industria audiovisual: es la última oportunidad de arreglar y/o mejorar el proyecto. Incluso cuando el trabajo está ya terminado y no se van a hacer más cambios, todas las empresas tienen un protocolo de evaluación que sirve para reforzar los puntos que hayan podido estar descuidados durante la producción y la postproducción.

Este es el momento en que probar distintas posibilidades para el pipeline de futuras producciones, especialmente si se plantea la utilización de nuevos hardware o softwares. "La introducción de nuevas herramientas y métodos de trabajo es bienvenida pues esta es

una industria caracterizada por el alto grado de innovación tecnológica” (Fidas, n.d.) Dichas posibilidades son testadas y evaluadas en pequeños grupos de artistas, que son los que finalmente deciden si esa tecnología se implanta o no. Esta fase es muy importante, dado que “un gran porcentaje del presupuesto es gastado en el diseño artístico y la ingeniería dedicados a la creación de assets” (Fidas, n.d.) Dado que los tiempos en preproducción y producción son siempre insuficientes, el momento de modificar el pipeline para el próximo proyecto es ahora.

Como vemos, un pipeline no es una estructura rígida como una cañería, aunque ciertos tipos se distribuyan como una. Debe ser un sistema flexible y en constante movimiento que se integre totalmente con el tipo de proyecto que vamos a realizar. Habiendo descrito por encima en qué consiste un pipeline destinado a las empresas audiovisuales, pasaré a tratarlo en profundidad enfocándome en el tipo de producción en el que se centra.



# **ORGANIZACIÓN Y PROCESOS ARTÍSTICOS EN LOS VIDEOJUEGOS**

“Desarrollar un juego es como intentar construir un avión mientras se está aprendiendo a volar” (Saunders & Novak, 2006:354)

El pipeline en videojuegos no siempre fue necesario. Al contrario que en la animación, que siempre hizo falta al menos un dibujante, en esta industria los primeros diseñadores eran ingenieros de programación. Aunque los primeros videojuegos datan de los años 50, no fue hasta los años 70- 80 que empezaron a desarrollarse juegos cuyo diseño, programación y animación fueron lo suficientemente complejos como para necesitar un pipeline específico: antes de eso, prácticamente ni siquiera era necesario un artista; con un par de buenos programadores bastaba. A medida que fue pasando el tiempo, el nombre del artista fue asentándose y convirtiéndose en un rol principal, generando un modelo de empleo típico de esta industria. “No es extraño para los empleados de la industria de los videojuegos dejar su compañía y comenzar una nueva para poder mantener su liderazgo creativo” (Saunders & Novak, 2006:40). Este modelo sigue perpetuándose hasta nuestros días; por poner un ejemplo, ArenaNet, creadores del videojuego GuildWars, está formada casi íntegramente por antiguo personal de Blizzard, creadores del mítico juego World of Warcraft.

“Desarrollar un juego es como intentar construir un avión mientras se está aprendiendo a volar”  
(Saunders & Novak, 2006:354)

Conforme el juego va perfeccionándose, comienza a desarrollar un pipeline paralelo al de las películas de animación. “Sacado a la venta en 1982 por Rally Midway, Tron fue el primer videojuego de arcade en ser creado por un estudio de cine y animación” (Saunders & Novak, 2006:43). Como veremos a conti-

nuación, muchos procesos de creación de activos artísticos son similares para los videojuegos y la animación 3D: ambas van de la mano con los avances en la industria tecnológica y los programas de modelado son similares, por lo que un avance en un sistema para videojuegos es adaptable a una película y viceversa. Y esa no es la única característica común: “Los factores que ayudaron a estas industrias a crecer en la misma dirección fueron las similitudes, los problemas en la producción de arte, el uso de herramientas de producción parecidas y el intercambio de talento” (Harmon, 2006:1)

El pipeline más óptimo para los videojuegos es el pipeline iterativo. “Dejemos de lado los desarrollos en cascada e implantemos los ágiles. Los juegos consisten en la reiteración- no escribir un enorme tomo de diseño y pasárselo al equipo de producción. Hacer que tu equipo sea ágil y mantener una versión jugable del juego lo más pronto posible es una gran forma de tener éxito” (John Comes para Game Development Essentials, Saunders and Novak, 2006:366). Debido a que los videojuegos son tan dependientes del equipo técnico, trabajando en consonancia con el departamento de programación, generalmente requieren de múltiples subfases: esta es la principal particularidad del pipeline orientado a los videojuegos. En algunas de estas etapas no interviene el departamento de arte en absoluto; en otras, es casi el único integrante. El principal problema que se presenta a la hora de elaborar

un estudio es que no hay un consenso: todas estos procesos hacen del pipeline para videojuegos algo muy orgánico; por tanto, las iremos estudiando en orden de aparición, pero debe quedar claro que muchas de estas subfases tiene características que pueden entrar en varias fases a la vez.

Asimismo, es importante recordarle al lector que, si bien aquí daremos una sensación de linealidad, la mayor parte de estudios suelen estar trabajando de forma paralela en varias partes y fases del desarrollo del juego de forma simultánea.





# **PRE-PRODUCCIÓN**

“La primera tarea suele ser la definición del concepto del juego. Esto tiene lugar antes de que la pre-producción comience formalmente y continúa a lo largo de esta etapa. El concepto y género del juego jugarán un factor en decidir qué plataforma y pipeline son mejores para la tarea” (Dunlop, 2014:60)

Podría decirse que el Concept es el momento de la evaluación y toma de decisiones. Es muy importante la creación del Pitch Document o Documento de presentación: la función de este archivo es “vender la idea a un productor u otra fuente de ingresos con la esperanza de ser contratado para crear el juego” (Daiger, 2009). En este documento trabajan los jefes de los departamentos de arte y diseño para que sea visualmente atractivo y fácilmente vendible por el departamento de arte. Suele ir acompañado de una demo, normalmente jugable, en la que ya pueden apreciarse el tipo de jugabilidad, el aspecto general del juego y alguna parte de la historia en caso de que exista una narrativa. Además, se deben presentar una serie de presupuestos desglosados y una línea de tiempo en la que se especifique el flujo de trabajo en relación al tiempo que llevará cada departamento. Una vez que estos tiempos están establecidos y el trabajo más o menos organizado, se acuerdan una serie de fechas

límite o milestones en las que debe tenerse terminada y enviada al productor una parte concreta del trabajo. Para resumir, esta es la “fase para crear planes de trabajo para el juego, probar nuevas ideas que vayan a ir incluidas, crear y probar pipelines de producción, aprender a usar una nueva tecnología (si es necesario) y resolver problemas. La

“El concepto y género del juego jugarán un factor en decidir qué plataforma y pipeline son mejores para la tarea” (Dunlop, 2014:60)

mayor parte del trabajo generado durante esta fase será usado en las otras fases, haciendo de ella la parte más importante del desarrollo. Un juego pasará en esta fase entre uno y tres meses” (Harmon, 2006:14).

Además de todo eso, esta etapa es la de recopilar información de referencia. Este trabajo se enfoca de la misma forma que el periodismo de investigación, y depende mucho del presupuesto. Para el departamento artístico consiste, básicamente, en reunir datos que puedan ser de interés o ayuda para los creadores en pre-producción. Dichos datos (dibujos y fotografías, canciones, relatos, básicamente cualquier cosa que ayude al equipo de artistas a inspirarse) son almacenados como ayuda para el equipo de pre-producción que decida el diseño final.

En términos artísticos, esta fase (o, según algunos autores, pre-fase) es de poco interés creativo. Si bien es cierto que se deja atada de manera general la idea y el estilo visual del juego y ya se define el argumento de la narrativa, estos elementos son muy provisionales. El principal objetivo es organizar logísticamente al equipo, distribuir tareas, buscar patrocinadores y productores, redactar contratos e implantar pipelines provisionales. Todos los elementos creados para el documento de presentación y la demo (personajes, activos,

ambientes, iluminaciones, narrativas, sonido, etc.) serán sometidos a un fuerte escrutinio y, probablemente, modificados en las siguientes fases.

“La fase de Concept comienza cuando una idea para un juego es planteada- y termina cuando se toma la decisión de comenzar a planear el proyecto (conocida como fase de pre-producción)” (Saunders & Novak, 2006:352). Aunque ya se hicieron planificaciones prototípicas y provisionales en la fase de Concept, todas esas ideas pasan por una fase de corrección constante. Se replantean las ideas, narrativas, se hacen todo tipo de dibujos preparatorios, hojas de personaje, estudios de iluminación y demás. Se replantea la interfaz y la jugabilidad, y se revisa el pipeline en pos de implantar uno más acorde con el presupuesto del proyecto.

Para el mundo de los videojuegos, la fase de pre-producción suele centrarse en terminar el Game Design Document o Documento de Diseño del Juego: “Uno de los documentos más importantes en el proceso de producción del videojuego, el documento de diseño del juego expone el estilo general, la historia y las características de jugabilidad. Es escrito por los diseñadores principales, y es uno de los primeros documentos escritos una vez que se le da luz verde al juego” (Daiger, 2009)

**En el Documento de Diseño van contenidas todas las directrices necesarias para los equipos de Diseño, Arte y Programación.**

Este documento será la referencia global de los artistas durante la etapa de producción. Por tanto, es esencial que sea accesible para todo el personal, muy claro y esté constantemente actualizado. Toda la etapa de pre-producción del departamento artístico gira en torno a este documento, donde se suelen incluir las referencias cosechadas durante la fase de Concept para mayor esclarecimiento. A través de este documento recibirán las tareas a completar el resto de departamentos, por lo que suele ser imperativo que esté terminado en el primer o segundo milestone. Aunque este no es el único documento; para tareas, interfaces, personajes, ambientes y demás elementos específicos se generan otros documentos que tratan los pormenores y detalles de forma más concreta y, al igual que el Documento de Diseño del Juego, deben estar constantemente actualizados. Para asegurarse de esto, la política de la mayor parte de empresas es tener una sola versión del documento donde van sobrescribiéndose las modificaciones. El sistema es parecido al de las bibliotecas: cada vez que un artista (generalmente el lead designer o diseñador jefe) hace una modificación en el documento, queda anotado quién y cuando la hizo. La versión anterior a la modificación queda almacenada en el servidor, pero no es accesible para el resto del personal.

En el Documento de Diseño del Juego van contenidas todas las directrices necesarias para los equipos de Diseño, Arte y Programación. Son el resultado de muchos prototipos desechados y variaciones, por lo que se espera que el equipo que trabaje en la redacción de

dichos documentos, que en grandes producciones pueden acumular hasta miles de páginas (en general con muchas imágenes), lo haga de forma rápida y eficaz. En este tipo de producciones se valora mucho la multidisciplinariedad: es importante remarcar la figura del Technical Artist o Technical Director, “un artista especializado en aspectos de la producción que requieren programación, automatización, o conocimientos gráficos avanzados” (Dunlop, 2014:436). Este tipo de personas son escasas y especialmente importantes en esta fase, dado que saben qué ideas son interesantes artísticamente, pero irrealizables o muy costosas en términos técnicos y a la viceversa, encontrando el sano punto medio entre estas dos variantes. Evidentemente, están apoyados por un equipo de los principales responsables de cada área.

¿Qué supone esta fase exactamente para el departamento de arte? En general, significa la materialización definitiva en dos dimensiones de todos los personajes, escenarios, ambientes y activos pensados por los guionistas. Procesos de siluetas negras, pruebas de color, escalas de grises, vistas ortogonales y demás información es creada para generar la hoja de personaje, o en su defecto el equivalente para el resto de elementos visuales del videojuego

“Estas hojas de personaje, junto a las otras piezas del concept art, son usadas por los modeladores 3D como guías para crear la maqueta del personaje, de manera similar a cómo los trabajadores de la construcción usan planos hechos por un arquitecto como guías para construir edificios [...] guías que el modelador trata de seguir tan precisamente como le es posible.” (Tapio Terävä, 2017:12).

La función del jefe de departamento es asegurar la claridad de la información producida. Algunos personajes no cambiarán apenas la forma que tenían durante la fase de Concept; otros darán un giro de 180 grados. En definitiva, el objetivo del departamento de arte durante todo este proceso es ser claros, exhaustivos y precisos a la hora de crear contenido para ponérselo fácil a los modeladores 3D que irán tras sus pasos en la siguiente etapa de producción.

Una vez que toda esta plantilla llega a consenso sobre todos los aspectos del juego que deben quedar recogidos en el Documento de Diseño y el productor o supervisor da luz verde, entraríamos en la siguiente fase: la producción. Pero para poder hacer esto, primero debemos comprobar que todos los elementos planteados por los diferentes departamentos funcionan bien juntos; que el estilo es atractivo y la interfaz de usuario y demás mecánicas de juego son operativas. Para esto entramos en una nueva subfase: la creación de la demo o prototipo. Este prototipo es más un reclamo promocional que un documento útil para la finalización del proyecto: “La definición común de un prototipo en la industria de los videojuegos es [...] un software operativo que captura en la pantalla la esencia de lo que hace al juego especial, lo que lo diferencia del resto y lo que lo transformará en un éxito” (Saunders & Novak, 2006:354).

En este tipo de producciones  
se valora mucho la  
multidisciplinariedad

Como veremos más específicamente en uno de los puntos subsiguientes, esta fase puede aparecer tanto al principio como al final del proceso de pre-producción, y a veces en ambas ocasiones. Lo que sí es cierto es que la producción de esa demo en la que intervienen todos los departamentos será a la vez una versión embrionaria del juego que presentar a los productores, un milestone muy grande y decisivo para llevar a cabo la compleción del proyecto y, para los desarrolladores, representará el testeo de todas las decisiones hechas hasta ahora. “Un objetivo común al final de la pre-producción es producir una demo jugable que pueda ser presentada públicamente para aumentar las expectativas del juego” (Dunlop, 2014:62)

Antes de hacer el prototipo, algunas empresas crean una versión física del juego muy similar a un juego de mesa tradicional para asegurarse de que la mecánica del juego funcione bien. Tras esto, se procede a crear una versión digital, pero intentar crear una recreación de un juego terminado antes siquiera de haber entrado en producción presenta una multiplicidad de complejidades técnicas. Normalmente, si la tecnología no está aún desarrollada y su desarrollo está planeado durante el proceso de producción, el efecto es imitado con motores preestablecidos. Para este tipo de supuestos, “debes prepararte para hacer demostraciones exclusivas que prueben que las diferentes piezas planificadas de tecnología son posibles de hacer [...] Los motores de creación de videojuegos suelen ser usados para el prototipado, incluso si el juego final será desarrollado en una tecnología completamente nueva” (Saunders & Novak, 2006:354)

Un objetivo común al final de la pre-producción es producir una demo jugable

Esta demo es una versión en miniatura de lo que el juego pretende ser y, como tal, dentro del pipeline general se implementa otro muy parecido a este, pero mucho más reducido. “Un proyecto de diseño es creado e implementado por los artistas, animadores y programadores, y después testado por el departamento de control de calidad. La producción no suele llevar mucho, y sólo un puñado de personas de cada departamento son necesarias” (Daiger, 2009)

Si la demo es aprobada, entraríamos en lo que se conoce propiamente como la etapa de producción. Esta fase es una fase intermedia porque asimila los conceptos realizados en pre-producción, pero también desarrolla nuevas herramientas y versiones que son características de la etapa de producción. Por ello se le da tanta importancia, y es probablemente el mayor punto de inflexión en la creación de un juego porque representa el último momento de posibilidad de corrección: “El milestone del final de la pre-producción es uno de los puntos decisivos para seguir o parar el proyecto, y hay una gran probabilidad de que un juego con una demo poco inspirada sea aparcada en vez de dada luz verde, o al menos sea mandada a otra ronda de pre-producción” (Dunlop, 2014:61). Este tipo de correcciones y contenciones son características del pipeline en videojuegos.



# PRODUCCIÓN



En términos temporales, esta es la etapa más larga de un videojuego cuyo resultado es un proyecto completo. También es la fase donde se hace patente lo adecuadas de las predicciones del pipeline realizado durante la fase de pre-producción.

Ya hemos clarificado que esta es la fase donde todos los contenidos del juego son producidos siguiendo las directrices de lo establecido en la etapa anterior. "Todo el trabajo durante la fase de producción lleva al juego cerca a estar terminado. Implementar nuevas ideas (conocido como "feature creep"<sup>3</sup>) es evitado porque pueden requerir una mayor preparación o, en el peor caso, comenzar la producción desde cero. Cualquier cosa que distraiga de la producción de los activos artísticos del juego se pospone al periodo de alpha" (Harmon, 2006:16)

Como vemos, la parte "creativa" del juego ya ha quedado atrás: ahora se trata de que los artistas en plantilla simplemente sigan las directrices que los artistas de pre-producción han establecido. El número de trabajadores se dispara: dado que la labor a realizar ya no es exactamente creativa, sino bastante más mecánica, los artistas son elegidos en función más

de su capacidad de trabajo que de las ideas originales que puedan aportar al proyecto. Vamos a revisar en profundidad en qué consiste el trabajo del departamento de arte durante esta etapa.

En términos temporales, esta es la etapa más larga de un videojuego

En general, esta fase se inaugura realizando los modelos 3D y/o, en su defecto, las animaciones que

el juego pueda necesitar tomando como base el Documento de Diseño del juego. Normalmente los motores de desarrollo de 3D son externos al motor del juego y están basados en polígonos, aunque esto puede variar según el proyecto.

"Cuando se crean modelos 3D, el modelador tiene diferentes técnicas a su disposición. Dos aproximaciones comunes son el modelado de caja y el modelado de arista. En el modelado de caja el modelador empieza con una forma primitiva pre-hecha, como un cubo, un cilindro [...] y empieza a añadirle detalles y características. En el modelado de arista el modelador comienza a construir el modelo a partir de polígonos individuales, [...] que crean la forma del modelo" (Tapio Terävä, 2017:14)

Casi todos los programas que trabajan el 3D están enfocados de manera que simulen el comportamiento y la textura de la arcilla, con la ventaja de que se les puede añadir modificadores, haciendo posibles cambios que en la vida real serían imposibles o muchísimo más costosos de hacer.

Confiar en motores externos para la creación de los activos 3D del videojuego tiene sus ventajas y sus inconvenientes. La principal ventaja es que son programas de fácil acceso

3

*Feature creep: fenómeno típico del proceso creativo en videojuegos en el que el artista se centra tanto en una característica concreta que resulte atractiva artísticamente que obvia el resto de tareas a realizar o hace previsiones irreales sobre estas. Es, en resumen, la necesidad de hacer el juego más atractivo constantemente y, por tanto, no llegar nunca a terminar nada.*

(algunos incluso de software libre) y están específicamente desarrollados para eso, por lo que siempre van a ofrecer mayor facilidad y versatilidad a la hora de trabajar. Pero tienen una gran contra: "la mayor parte de herramientas para la creación de activos artísticos no los muestran de la misma manera en que se verían en el videojuego" (Dongen, 2007).

Es decir, que a lo mejor el renderizado se ve en la pantalla del software de forma diferente a como se vería en la pantalla del videojuego. Esto significa que el artista está trabajando relativamente a ciegas: el modelo puede dar problemas en la interfaz (se pueden abrir polígonos, invertir las normales, que el número de polígonos sea demasiado para lo que el procesador del juego pueda soportar, etc) y el modelador puede no saberlo y seguir trabajando. Por esto es muy importante que una versión del juego operativa sea desarrollada lo antes posible: para cargar en esa matriz los modelos en 3D y comprobar que funcionan. Este proceso debe ser hecho con frecuencia, por lo que lo normal es crear una interfaz igual a la del juego pero que no tenga que cargar el resto de elementos, haciendo así que el testeo sea mucho más ágil.

Aún así, modelar toda esa cantidad de activos lleva una cantidad de tiempo ingente, y hay muchas características del juego que deben ser testadas desde el principio. Para estos casos, el pipeline tiene un plan de contingencia que Dongen llama "el atajo": dejar fuera o sustituir por versiones de muy pocos polígonos los activos artísticos para

generar desde muy temprano una versión jugable fiel a la realidad. Al principio el juego será una amalgama de bultos grises que pueden moverse; a medida que se vayan terminando de hacer los activos, se van incorporando al juego. De esta manera, podemos ver enseguida si el activo desarrollado da algún tipo de error dentro de la matriz del juego con tiempo suficiente para corregirlo. Si solo incorporásemos los activos al juego una vez que estuvieran todos hechos (como sería el caso del pipeline lineal) no tendríamos ningún tiempo de reacción en caso de que alguno de los activos diera error, y corregirlo supondría costes adicionales de tiempo y dinero. Por eso los videojuegos se acoplan mejor al pipeline reiterativo: durante la producción del videojuego, los activos entran en un bucle constante de creación-supervisión- testeo del que solo salen si se incorporan bien con el resto del juego, evitando sorpresas desagradables de última hora.

La mayor restricción para los artistas durante esta fase es de tipo técnico. La más común es la cantidad de polígonos y la topología de estos: "Además de recrear la forma de un objeto fielmente, el modelador es responsable de asegurarse que la topología es correcta. Esto incluye asegurarse de que los polígonos a partir de los cuales está construida la malla siguen la forma interna de la geometría y son de un tamaño y forma apropiados [...] Es también necesario asegurarse de que el modelo se mantiene bajo un conteo de polígonos máximo,

Los videojuegos se acoplan mejor al pipeline reiterativo

para que sea más fácil de manipular y renderizar” (Dunlop, 2014:85)

Es por esta razón que se pueden distinguir dos tipos de modelado 3D: el low poly (baja poligonización), o modelos con baja resolución de polígonos, y el high poly (alta poligonización), modelos con alta resolución de polígonos. “Los así llamados high poly, que son versiones suavizadas e intrincadamente detalladas de los modelos menos detallados usados en el juego, que son correspondientemente llamados modelos low poly. Los detalles de la superficie de los modelos high poly pueden ser transferidos a las texturas de los modelos de low poly en un proceso llamado bakeado” (Tapio Terävä, 2017:16)

La principal diferencia entre estos dos es que el primero puede pasar directamente a ser texturizado, mientras que para texturizar el segundo es necesaria hacer una retopología para reducir el número de polígonos. La retopología es importante por dos motivos principales: en primer lugar, para reducir el número de polígonos. Un mayor número de polígonos en el motor del juego requiere una mayor capacidad de procesador: exceder esa capacidad puede traducirse en glitches, paradas repentinas del juego, sobrecalentamiento, etc.

En segundo lugar, para asegurar que las animaciones de los personajes vayan de forma fluida y no haya polígonos que se deformen inadecuadamente o colapsen unos contra otros.

Un mayor número de polígonos en el motor del juego requiere una mayor capacidad de procesador

Por tanto, una retopología adecuada será mucho más importante en un objeto que vaya a ser animado

posteriormente que en uno estático. Actualmente hay programas que realizan la retopología de los modelos de forma instantánea, pero es aconsejable que el artista revise el modelo después de haber sido retopologizado y, si es necesario, añada polígonos en zonas que van a moverse mucho. Una mayor densidad de polígonos asegura una animación más fluida. “Esto permite crear animaciones más naturales, dado que los animadores tienen más control sobre la deformación del modelo 3D” (Tapio Terävä, 2017:18)

Una vez que la retopología es la adecuada, el modelo pasa a ser pintado y texturizado. “El equipo artístico se encargará de crear todos los mapas de texturas que son añadidos a los objetos 3D para darles más vida y carácter, y también se encargará de animar los personajes y objetos que se mueven en el juego” (Edwards, R., 2006). Como ya hemos mencionado antes, es el momento de hacer el bakeado de texturas: una forma de dar al espectador la ilusión de que el modelo low-poly que será incorporado al motor del juego es un modelo high-poly. Para esto, la mayor parte de softwares 3D llevan incorporado un motor específico en el que transforman todas esas pequeñas texturas del modelo en alta resolución en una especie de pegatina. El ordenador necesita coordenadas en las que pegar esta pegatina: para eso está la fase de UVWrap. Después de terminar y chequear la retopología, se marcan en las aristas de los polígonos una serie de “costuras”: líneas de corte que dividen la super-

ficie de la malla en varios trozos. Al realizar el bakeado de texturas, el programa copia la pegatina del bakeado en las coordenadas específicas de cada costura, asegurando que al moverse la pegatina esté en el lugar correcto. Este proceso es aplicable a los diferentes mapas de texturas que vayamos a incorporar al modelo, que en general suelen agruparse en tres categorías: "Debido a las limitaciones del hardware necesitaremos extraer todos los detalles del modelo high poly al modelo low poly del juego. Para esto necesitamos generar un mapa de normales (para extraer los detalles del 3D), un mapa de difusión (para los colores) y un mapa de oclusión (la forma en que la luz se refleja en la vida real)" (Tironeac, G., 2013) Después de realizar el bakeado, se crea un mapa de difusión en el que se pinta sobre el modelo y finalmente, se controlan las luces mediante shaders como el mapa de oclusión. Los shaders se incorporan a cada uno de los elementos en particular, pero hay shaders dentro del motor del juego que afectan a todos los elementos para simular ambientes muy concretos como la refracción del agua o la luz del sol.

Si el elemento en el que se ha trabajado es estático y no requiere de animación, está listo para ser incorporado al motor del juego y testado. Sin embargo, si se trata de un personaje o un NPC que requiere de movimiento, será animado conforme a los parámetros del juego. En contraposición a la animación para cine, donde se trabajan los movimientos de los personajes de forma lineal, en videojuegos se trabaja en ciclos. En esencia las mecánicas de cada proceso son similares: se trabaja con keyframes, normalmente siguiendo los doce principios de la animación, etc. pero en videojuegos se crean una multiplicidad de pequeñas animaciones, cada una de ellas asociada a un comando específico, que se desencadenan cuando el jugador ejecuta dicho comando. Si además de estas animaciones hay una línea argumental definida, es probable que sea necesaria una animación más tradicional en la que entran en juego no solo los parámetros de movimiento del personaje, sino también la posición de la cámara, gestos concretos que no hayan sido animados para los comandos de jugabilidad, etc.

En videojuegos se crean una multiplicidad de pequeñas animaciones, cada una de ellas asociada a un comando específico

Hay diferentes maneras de realizar una animación. La más común es el proceso de rigead: "Es el proceso de creación de un rig, el cual es un sistema de huesos digitales, articulaciones y otros controles enlazados al modelo 3D, y que los animadores pueden usar para mover y doblar el modelo para crear animaciones" (Tapio Terävä, 2017:30). Este esqueleto se crea debajo de la malla, dentro del personaje, y al moverse arrastra con él los polígonos que estén encima. Según la complejidad del movimiento que vaya a efectuar el personaje, se le incluirán más o menos articulaciones a ese esqueleto digital. Es importante que las articulaciones estén en el sitio correcto. "Si no, la rotación creará un resultado que hará que parezca que el brazo está roto" (Ward, A. 2013) A la hora de animar, lo que se anima es el esqueleto, dando sensación de movimiento al personaje. Una forma alternativa a este pro-

ceso es el llamado "Motion Capture", o la grabación de movimientos de personas reales con sensores que luego permitan recrearlo digitalmente mediante la superposición de la posición neutra del esqueleto a la posición neutra del modelo. Una vez hecho esto, todas las partes independientes creadas durante todo este proceso (el modelo low poly, los mapas de texturas, shaders, etc) son exportados al motor del juego y se comprueba que se ven y se mueven correctamente. Este proceso se conoce como renderizado.

Además de todo esto, en muchos juegos de RPG es necesario simular los efectos del agua, refracciones de luces, efectos y demás, procesos que normalmente trabajarían artistas de efectos especiales. La animación en videojuegos es un proceso largo y complicado: se trata de darle vida y movimiento al juego, y hacerlo inmersivo.

Una vez que todos los elementos se han incorporado al motor del juego, se han renderizado todos los activos artísticos, se han desarrollado las interfaces de usuario y se ha testado la jugabilidad a grosso modo comprobando que cada nivel es jugable y correcto, pasamos a la etapa de finalling o post-producción.

# **FINALLING**

Este momento también está comprendido por una serie de etapas. La primera de éstas es la fase alfa, en la que en teoría el juego es jugable de principio a fin. Aún puede tener pequeños fallos y algunos activos artísticos pueden necesitar revisión, "en lugar de crear y construir, la fase alfa es acerca de terminar y repasar el juego. Si algunas características deben ser omitidas para llegar a la fecha de entrega, este es el momento de hacerlo" (Saunders & Novak, 2006). Por tanto, si en la fase anterior la voz cantante la llevaban el departamento de arte y el de programación, ahora el protagonista será el departamento de control de calidad, que se asegurará de testar cada característica del juego al menos una vez.

La fase alfa es más un reto para el departamento de programación que el de arte. A nivel de testeo, el principal objetivo es identificar y reparar posibles bugs o errores de código que pueda tener el juego. "Simultáneamente a los esfuerzos del departamento de control de calidad para encontrar bugs tiene lugar una búsqueda similar por parte del productor. El productor busca problemas e inconsistencias en el juego y las comunica al desarrollador" (Daiger, 2009:20)

La fase alfa es más un reto para el departamento de programación que el de arte

Sin embargo, también merece atención la labor del departamento de arte, que consistirá básicamente en mejorar los activos que fueron hechos en producción. "Durante el alfa, los activos artísticos suelen ser devueltos a producción para pequeñas modificaciones. [...] A veces, incluso vuelven a la fase de Concept [...] Los artistas comprue-

ban el rendimiento de los activos en los fotogramas por segundo del motor del juego. Luego, evalúan cambios a modelos o texturas que mejorarán la fluidez de los fotogramas por segundo" (Harmon, 2006:17)

En términos generales, el director artístico suele darse cuenta de estos pequeños errores antes de enviarlos al departamento de calidad, donde serán testados por personal externo al equipo de desarrollo. Para que el juego pase a la fase beta, la siguiente subfase del finaling, hay nueve reglas que debe cumplir, entre ellas: debe ser jugable de principio a fin, debe tener al menos una lengua integrada, una interfaz básica con documentación preliminar, debe ser compatibles con sus hardwares y softwares y, para el departamento que nos compete, todo el arte y el audio del juego deben estar correctamente colocados y ser funcionales. Hay un par de requisitos más que no vamos a tratar aquí porque son competencias específicas del departamento de programación. Una vez que el juego cumple todos esos requisitos, está listo para pasar a la fase beta, la última fase del juego en producción.

En beta, el juego está listo para ser publicado. Aún hay bugs que resolver, pero ya es perfectamente jugable. Esta etapa trata sobre todo de testeo y optimización: se cambian algunos aspectos que puedan optimizar la fluidez del juego basándose en las opiniones que puedan aportar los game testers. Algunos juegos hacen una versión abierta de la fase beta "donde

un grupo específico de consumidores prueban el juego, en parte para generar opiniones positivas, en parte para obtener feedback de última hora, en parte para localizar bugs que se les haya podido escapar a los testadores profesionales” (Dunlop, 2014:70)

El departamento de arte, a estas alturas, ya se ha ido en su gran mayoría a casa. Una vez que el trabajo está terminado y el juego listo para su distribución, las empresas aprovechan este momento para probar posibles pipelines, nuevos motores de creación y gestión de datos, etc. que puedan ser útiles para proyectos futuros.

Por supuesto, hay juegos que son constantes: la mayor parte de juegos online requieren un pipeline en miniatura porque necesitan estar generando nuevo contenido cada pocos meses. Además, es común para las empresas de videojuegos tener que relizar algún parcheo informático para arreglar algún bug; muchos tienen un soporte técnico. A esto se le suma la creación de expansiones, que en general se hacen a partir de ese contenido del videojuego que se quedó fuera para evitar retrasar la fecha de lanzamiento. Pero, exceptuando estos casos para los que se designan equipos específicos, el equipo de producción puede descansar. El juego está terminado.





# **DIFERENCIAS ENTRE PYMES Y GRANDES EMPRESAS**

El principal motor de las diferencias en el pipeline entre empresas AAA (grandes productoras) y juegos indie (pequeña y mediana empresa) es el presupuesto. Sin embargo, en los últimos años hemos experimentado un auge de los juegos indie frente a las producciones millonarias.

Podemos atribuir este crecimiento a la diferencia de libertad a la hora de elegir un argumento o una estética: las empresas de triple A elegirán normalmente ideas más accesibles y universales, en definitiva, más comerciales. Por el contrario, las empresas indie podrán permitirse más fácilmente juegos cuyo argumento sea más controvertido o tengan una estética más original. Esto tiene sus pros y sus contras: "La mayor parte del tiempo, la gente tendrá ideas geniales pero generalmente estas ideas son demasiado grandes para ser realizadas por un equipo indie. Hay que pensar cuidadosamente si se tienen los recursos para desarrollar esas grandes ideas; de lo contrario, es mejor desarrollar algo pequeño pero estar seguro de que puedes completarlo" (Gamedevelopertips, 2016)

Comenzaremos estableciendo qué es cada una. Las pymes están representadas en el mundo de los videojuegos como empresas indie. En general, son empresas pequeñas o medianas que trabajan en uno, dos juegos a la vez y cuyo personal está acostumbrado a la multidisciplinariedad.

En los últimos años hemos experimentado un auge de los juegos indie frente a las producciones millonarias

Las grandes empresas están representadas en el mundo de los videojuegos como empresas de triple A:

estas empresas tocan múltiples producciones de grandes presupuestos y por tanto tienen una mayor facilidad de recursos y mayor especialización y compartimentalización de los roles del personal. " Los estudios indie están ayudando a llevar la innovación por caminos interesantes, particularmente cuando se trata de pensar fuera de la caja con una dirección visual fresca y concepts de juegos inusuales [...] A la vez, el ultra detallado fotorealismo y el alcance masivo siguen teniendo la soberanía en el espacio AAA [...] Vemos proyectos mayores, más brillantes y más épicos viniendo de franquicias establecidas y estudios más grandes, donde los artistas y diseñadores empujan los últimos hardwares a más allá de sus límites" (Foundry, 2017)

Así, muchos estudios indie tenderán hacia estéticas más minimalistas, mientras que los estudios de triple A podrán permitirse cierto barroquismo a la hora de añadir contenido al juego.

La dinámica de trabajo de los artistas en las empresas grandes y las pymes también es muy diferente: "La especialización de los individuos del equipo en las producciones AAA versus el enfoque de llevar varios sombreros de los artistas indie es normalmente lo que diferencia la dinámica creativa a lo largo de las diferentes áreas de la industria" (Foundry 2017)

Dejando de lado a los trabajadores como individuos, encontramos que en general el pipeline para las empresas indie suele estar menos desarrollado. “Con un equipo pequeño, y con comunicación de uno a otro entre el programador [...] y el artista [...] hay mucha menos necesidad—o más correctamente, mucha menos percepción de necesidad para todas las herramientas y procesos que los grandes equipos esperarían” (Theodore, Steve. 2016) Como veremos en el siguiente punto, es muy raro que las empresas de triple A se vean obligadas a cancelar proyectos por errores en el pipeline dado que cada uno de los títulos que desarrollan está concienzudamente planeado, y los pipelines están siendo permanentemente reajustados para que el próximo proyecto vaya más rápido o sea más barato.

Por otro lado, las empresas de triple A son su propio productor. Las empresas indie dependen en general de un productor o patrocinador externo al que deben venderle la idea antes siquiera de que el proceso de pre-producción empiece, por lo que deben contar con una fase extra de ensamblar de prisa y corriendo una presentación, un pipeline, una demo, etc. con las que vender el videojuego.

Estas son solo algunas de las diferencias más características, pero cada proyecto es un mundo y cada empresa tiene su propio *modus operandi*. Pasaremos ahora a debatir los fallos más comunes en el pipeline: dado que las empresas indie tienden a tener un pipeline más débil, este punto servirá como otra forma de ilustrar las diferencias y vulnerabilidades del pipeline indie versus triple A.



# **ERRORES COMUNES Y CÓMO EVITARLOS**

“Los propios pipelines se asemejan a los juegos online. Algunos desarrolladores se consuelan en el hecho de que el error de hoy puede ser corregido en la actualización de la semana que viene, pero compañías más inteligentes se dan cuenta de que esos problemas heredados pueden ser una carga durante mucho tiempo, y testan el juego en consecuencia” (Dunlop, 2014)

A la hora de enfocar un problema en el pipeline, muchas veces hay que preguntarse si, a estas alturas, vale la pena solucionarlo. Es decir: qué ahorra más tiempo y dinero a la compañía, seguir trabajando en un sistema con fallos hasta que la tarea esté finalizada o parar motores, implementar un nuevo sistema más óptimo y tener que reajustar todo lo que se ha hecho.

Este tipo de preguntas son las que deben plantearse cuando el juego entra en producción y queda patente que las estimaciones que hemos hecho se han quedado cortas.

Y es que la creación de un proyecto similar es algo muy orgánico, y es probable que la idea inicial que se tenía no tenga mucho que ver con el producto terminado. Si un problema

**En producciones de triple A, es más probable que se tomen su tiempo a la hora de arreglar y optimizar este tipo de situaciones.**

sale a mitad del camino, muchas veces es más rápido elegir una vía alternativa que arreglarlo; en esta industria la constante es el ahorro, ya sea de dinero, de tiempo o de personal.

Muchas empresas indie elegirán esta opción porque el presupuesto es el que es; en producciones de

triple A, es más probable que se tomen su tiempo a la hora de arreglar y optimizar este tipo de situaciones.

Es importante entender que todo error en el pipeline se acaba manifestando como un error en el propio juego: algunos son muy obvios, como glitches. Otros no se encuentran hasta que es necesario hacer una actualización muy concreta. En nueve de cada diez casos, los errores que encontraremos serán errores de planificación. “Muchas veces un juego es planeado, producido y sacado a la venta como un producto totalmente distinto de lo que estaba concebido inicialmente. El alcance del juego podría no haber sido realista, la planificación puede no ser suficiente desde el comienzo, y la dirección de proyecto puede cambiar múltiples veces durante el proceso de creación” (Sundstr & Mayes, 2012)

Para evitar este tipo de errores, las empresas tienen una serie de evaluaciones que se realizan una vez que se ha entrado en finalling y hay poca gente trabajando en el juego en sí. De la misma manera que este momento se aprovecha para testar nuevos softwares, también se aprovecha para evaluar todo el proceso de trabajo, de principio a fin. A estas evaluaciones se las conoce como post-mortems: “Post-mortems es lo que el diseñador, los miembros del equipo o a veces el productor de un juego escriben después de que el juego ha

sido terminado y lanzado. Suele contener la descripción de cómo el periodo de producción del juego se ha desarrollado, qué fue bien en el juego y qué no fue tan bien" (Moore, Sward, 2007:626)

En estos post-mortems suelen estar las claves para evitar cometer los mismos errores en el próximo proyecto, por lo que las empresas suelen ser muy exhaustivas con ellos. Para presentar la lista de los errores más comunes, nos hemos basado en los post-mortems de una serie de juegos. Como veremos, la mayor parte son de estudios indie.

Un error muy común es comenzar la producción antes de que la pre-producción haya terminado o, en su defecto, recortar tiempo y personal de la fase de pre-producción para añadirlos a la fase de producción. "La industria responde a la intensa competitividad comprimiendo la pre-producción al periodo de tiempo más corto posible" (Betke 2003:26)

Esto se traduce en una planificación desastrosa: el equipo de producción se basa en una documentación difusa o incompleta, y la planificación en términos de tiempo y personal han sido hechos aprisa y corriendo. En el afán por economizar recursos y meterse en producción cuanto antes, encontramos que muchas opciones no han sido tenidas en cuenta.

Un error muy común es comenzar la producción antes de que la pre-producción haya terminado

El recorte en esta fase es, con diferencia, el fallo más común en los pipelines porque, debido a que se hacen sin pensar demasiado, tienden a desembocar en tiempos no realistas, subestimándose el tiempo que lleva realmente desarrollar el juego. Este es el caso del juego "Black and White", de Lionhead Studios: "Subestimamos cuánto tiempo llevaría construir y escribir la historia de Black and White [...] Pensamos que no llevaría más de dos meses, pero después de un tiempo nos dimos cuenta que no teníamos las habilidades necesarias para ocuparnos de este aspecto vital del juego" (Molyneux, 2001)

Otro error muy común es la inexperiencia del equipo fundador. Este es el caso del videojuego "Magicka", de Arrowhead Game Studios. La idea inicial fue creada durante el último año de carrera de los desarrolladores, que pasaron a desarrollarla directamente. Por tanto, a la hora de evaluar nuevas ideas, implementar nuevos sistemas de trabajo, etc. nuestros desarrolladores no tenían mucha idea de lo que estaban haciendo: "Como ninguno de nosotros tenía experiencia real en el desarrollo de un videojuego, cada idea era evaluada no en base a su viabilidad, presupuesto o su posible valor de retorno, sino del autor de dicha idea contándonos al resto acerca de las hilarantes situaciones que podrían presentarse en las imaginadas mecánicas de jugabilidad" (Pilestedt, J. 2011)



Este es un caso típico de feature creep: el juego se volvió tan atractivo, tan interesante, había tantas posibilidades, que siguieron incluyendo contenido nuevo sin llegar a terminar nada, y obviando el hecho de que la creación de todos esos nuevos activos eran una carga enorme de presupuesto y retrasaba mucho las fechas de entrega.

Otro error común es no tener las herramientas adecuadas de optimización y adición de contenido en el videojuego: muchas veces, aspectos concretos de la producción se subcontratan a empresas especializadas que realizan el contenido y luego lo envían al equipo de producción. Es muy común encontrar pequeñas diferencias, por ejemplo nombrando los activos artísticos, y esas diferencias deben ser corregidas. Asumir que no hará falta un tiempo de corrección para el material externo es un fallo de pipeline que suele acompañar a no tener un sistema informático que minimice o automatice este tipo de correcciones. Es el caso del videojuego "Bioshock", de 2K Boston/ 2K Australia: " El problema principal fue que una gran cantidad de contenido llegó online después de la fase beta, y el equipo tuvo que apañárselas para instalar correctamente ese contenido a la vez que corregían bugs" (Finley, A. 2008)

Otro error común es no tener las herramientas adecuadas de optimización y adición de contenido en el videojuego

Un gran error es tener equipos divididos de personal. Aunque es bueno que los miembros se cuestionen, se reten y se marquen la excelencia como objetivo, la competitividad en exceso puede terminar convirtiéndose en algo insostenible. Es el caso del juego Deus Ex, de Ion Storm: "Estábamos haciendo un juego diseñado para romper

las fronteras entre géneros, y pensé que un poco de argumentación y competición podrían llevar a una síntesis de ideas interesantes. Pensé que podría controlar la tensión entre los dos grupos y que los dos grupos y el juego serían más fuertes por ello. Mi plan no funcionó" (Spector, W. 2000). Dado que cuando se acerca un milestone es normal trabajar turnos muy largos, es importante que el clima de trabajo sea el adecuado. Aunque también podemos encontrar el problema opuesto: que los integrantes de los grupos se lleven tan bien, que por temor a herir sensibilidades no se planteen ni se cuestionen las ideas del resto de integrantes del grupo. Ambos problemas, a la larga, son igual de malos.

Y es que todos estos problemas son complicados: subestimar los tiempos requeridos, recortar tiempo de la fase de pre-producción, fomentar tensiones negativas en el espacio de trabajo, la inexperiencia del equipo, feature creeps, etc. pero todos tienen una solución posible. El peor error de todos es el que vamos a tratar a continuación. Es el caso de la saga Trine, de Frozenbyte. La saga Trine consta de tres volúmenes: es un juego de plataformas que originalmente se concibió como un juego de triple A pero a la hora de encontrar patrocinadores no tuvo éxito, por lo que tuvo que reinventarse en un juego indie de bajo presupuesto. Y cometió todos los fallos que hemos mencionado antes y unos cuantos más: daban a los artistas completa libertad, dando el peor caso de feature creep posible, retrasaban

o no entregaban los hitos, incrementaron el presupuesto hasta el punto de recurrir a los fondos propios de la empresa (de manera que si el juego no tenía éxito, la empresa quebraría), tenían perspectivas irreales de diseño y producción, no pudieron solventar las diferencias de los distintos motores de juego (el juego para Xbox y ordenador funcionaba, pero para la PS3 era un cúmulo de bugs y glitches)... en fin, un desastre. Pero no acabó ahí: milagrosamente Trine salió al mercado y tuvo éxito. Hasta tal punto, que decidieron hacer un Trine 2. Y los fallos de pipeline de Trine 2 son exactamente iguales que los de Trine: "Trine como juego salió genial. Trine como proyecto fue un gran desastre - una serie de catastróficas desdichas que acabaron a un pelo del miserable fracaso. Pero hicimos el juego y al final del día es lo que cuenta. [...] Estábamos, y aún estamos, muy orgullosos de Trine y de lo que hemos conseguido" (Hyvärinen, L., Kinnunen, J. 2010)

El peor error que puede cometerse en el pipeline y, en general, en esta industria, es no aprender de tus errores.



# CONCLUSIÓN

“Usar y construir un pipeline es, ante todo, una forma de gestión. De todas formas, tener un pipeline no hace el trabajo y la gestión por ti, sólo lo hace más claro[...] El pipeline describe el proceso de producción, haciendo posible examinar cómo el trabajo real se hace antes y durante la producción” (Ciszek, 2012:12)

Los conocimientos adquiridos durante la elaboración de este TFG me han abierto una serie de puertas. No sólo he podido optimizar mi workflow a título personal, como explicaré en los siguientes párrafos: también me ha dado un conocimiento profundo de la mecánica subyacente de esta industria al que he podido dar aplicación práctica en una serie de casos.

Mi pipeline personal antes de realizar este TFG era, ante todo, desorganizado. Gran parte del trabajo que yo hago es en 2 dimensiones, relegando el 3D al terreno personal. Y si bien el workflow que yo aplico al 3D es un proceso muy convencional, mi objetivo al comenzar este TFG es que todo mi trabajo en 2D fuera aprovechable para, si se daba el caso, realizarlo en 3D sin mayor problema.

Antes que nada, explicar al lector que el terreno que me resulta de mayor interés es la

creación de personajes para videojuegos. Como se puede entender, esta disciplina está principalmente orientada a, eventualmente, realizar una maqueta en 3D del personaje y animarlo. Dentro de esta fase, el proceso que más me interesa, el que más trabajo, es la fase de concept. En los trabajos previos a esta investigación, estaba incurriendo

Tener un pipeline no hace el trabajo y la gestión por ti, sólo lo hace más claro

sin saberlo en errores muy comunes que pueden darse en el pipeline: los personajes que diseñaba apenas pasaban por una fase de concept y no había mecánica ninguna. Se me ocurría el personaje, lo dibujaba y a por el siguiente proyecto. Pero con los conocimientos adquiridos durante esta investigación, me vi obligada a modificar este modus operandi de la siguiente manera: para comenzar, me forcé a mi misma a realizar estudios previos. Ya fueran estudios de anatomía, sombras negras, thumbnails o cualquier otro tipo de dibujo previo a la ilustración. Esto no solo me daba una perspectiva más amplia a la hora de ser creativa, sino que me permitía comprobar antes de empezar hasta qué punto la anatomía era coherente y la figura legible. Es importante que el personaje pueda “leerse” bien desde una multiplicidad de puntos de vista: en mis peripecias en calidad de jugadora de videojuegos empedernida me ha ocurrido en múltiples ocasiones el tener que parar a buscarme en la pantalla. Es mejor evitar este supuesto creando un personaje cuya silueta sea muy característica y muy marcada, haciéndolo reconocible.

Otra de las modificaciones que incorporé a mi pipeline fue la organización dentro del motor utilizado. Muchas producciones AAA manejan cantidades increíbles de información, y uno de los problemas que esto plantea es la nomenclatura de los activos artísticos: asegurarse que el archivo está correctamente etiquetado y nombrado son requisitos necesarios para

que el motor del juego pueda cargar el activo en la plataforma. Un error ortográfico, por pequeño que sea, puede significar un desastre, especialmente si tenemos en cuenta que la mayor parte de lenguajes de programación son estrictos en la nomenclatura a la hora de escribir código. Por tanto, otra de las mejoras que añadí a mi proceso fue la manera de nombrar a los activos: ya fueran las capas de Photoshop, los pinceles, los mapas de normales... daba igual. A la larga me di cuenta de que ahorraba mucho más tiempo que el que había pasado renombrando todos esos elementos.

La siguiente incorporación la realicé emulando una hoja de diseño. Ésta, cada vez que sufre un cambio, se reescribe para asegurar que nadie del equipo está trabajando en una versión antigua. Sin embargo, el servidor almacena siempre una copia de las versiones anteriores para, en caso de que sea necesario volver a un estado anterior, el trabajo descartado sea el mínimo posible. Así pues, adquirí el hábito de hacer un guardado del proyecto cada poco tiempo cuyo nombre era un número, asegurando así no sobrecribir nada en ningún archivo. Muchos programas de pintura digital, como Krita, traen este mecanismo incorporado a la hora de guardar los archivos. Además, mantenía siempre una pequeña hoja de ruta en la que apuntaba las modificaciones futuras que debía realizar en mi proyecto. Como muchos de estos trabajos ocupan más de dos jornadas, encontré especialmente útil dejar escrito de un día para otro las modificaciones que debía realizar: "Añadir luces a la cara", "Bloquear las capas inferiores y ponerles fecha", "Realizar el boceto del fondo", "Descargar de internet ejemplos de retopología". Estas órdenes emulan el tipo de instrucciones que da el lead designer al resto del equipo, donde es importante que el lenguaje sea claro, conciso y concreto.

Es importante que el lenguaje sea claro, conciso y concreto.

Por último, la modificación final que apliqué a mi propio sistema de trabajo después de concluir esta investigación, y probablemente la imposición que me resulta más difícil de cumplir: no añadir contenido una vez que ha finalizado la fase de concept. Es curiosa la pulsión creativa que ocupa a los artistas cuando disfrutan con su labor, y si bien trabajando en solitario no supone ningún hándicap, cuando la tarea de varias personas dependa de la mía será importante que sepa restringirme y no incurra en el feature creeping. Por tanto, mi resolución fue no añadir contenido a partir del comienzo de la producción, cuando la figura pasa del plano bidimensional al tridimensional. En esta resolución aún estoy trabajando. Realizar esta investigación me beneficia, además, en otro sentido. Soy partícipe de la opinión de otros autores renombrados en esta industria, como Dunlop, de que el conocimiento del pipeline general de la producción es necesario para trabajar en videojuegos, especialmente porque quien lo conoce suele entender mejor que está trabajando en grupo y por tanto sabrá actuar en consecuencia. Como experiencia personal, puedo añadir que el conocimiento del pipeline me ayudó a la hora de realizar la entrevista para el Doble Máster en Concept Art

y Videojuegos de la UCM que cursaré el año que viene. Pude utilizar mis conocimientos en la materia para informar a la entrevistadora no solo de los procesos de 3D, sino de todos los procesos en general. Dado que la didáctica de esta escuela es generar un TFM que sea la demo de un videojuego poniendo en conjunción diferentes grupos de diferentes disciplinas del máster (programación, marketing, diseño y arte) considero que este conocimiento puso mi solicitud de matrícula en mayor valor frente a otras candidaturas.

Mi conclusión, en general, es que el conocimiento del pipeline es necesario para trabajar en esta industria. Pero si hay un requisito mayor, yo diría que es jugar. Jugar a muchos videojuegos, lo más heterogéneos posibles, supone una influencia muy positiva para todos aquellos que quieran dedicarse a esto. A través de los juegos podemos plantearnos toda una serie de cuestiones que nos ayudarán a la hora de realizar nuestro trabajo, sea cual sea, dentro de esta industria. Nos permitirán replantearnos nuestra forma de trabajar y nuestro pensamiento crítico. Recordemos que estamos desarrollando juegos: divertirse es parte del proceso.

# **BIBLIOGRAFÍA**



- Ciszek, P. (2012). 3D Production Pipeline in Game Development. University of Jyväskylä.
- Daiger, K. (2009). The Documents and Assets Created During the Video Game Production Process. University of Texas.
- Dongen, J. van. (2007). The Game Asset Pipeline. Master Thesis, 29(1), 11–38.
- Dunlop, R. (2014). Production Pipeline. Fundamentals for film and games.
- Fidas, C. (n.d.). Evaluation of a new asset creation pipeline for indie game developers.
- Harmon, J. L. (2006). The real time art production pipeline: An examination of modeling for surfacing as used by professionals from Electronic Arts Tiburon, 129.
- Lindeman, R. W. (n.d.). The Game Art Pipeline.
- Saunders, K., & Novak, J. (2006). Game development essentials.
- Sundstr, Y., & Mayes, A. (2012). Game design and production. Frequent problems in game development.
- Tapio Terävä. (2017). Workflows for Creating 3D Game Characters. Retrieved from [http://www.theseus.fi/bitstream/handle/10024/131241/Terava\\_Tapio.pdf?sequence=1&isAllowed=y](http://www.theseus.fi/bitstream/handle/10024/131241/Terava_Tapio.pdf?sequence=1&isAllowed=y)
- Moore E. Michael, Sward Jennifer, 2007, Game design and development - Introduction to the game industry, Pearson Education, New Jersey
- Betke Erik, 2003, Game design and production, Wordware Publishing, inc., Texas
- Edwards, R. (2006). "Phase 3: Production" en The Game Production Pipeline: Concept to Completion [en línea], [consulta:30 abril 2018]. Disponible en: <http://www.ign.com/articles/2006/03/16/the-game-production-pipeline-concept-to-completion?page=2>
- Tironeac, G. (2013). "Maps extraction from ZBrush mesh to game mesh" en The pipeline behind modeling and animating a game character in Zelgor [en línea], [consulta 30 abril 2018]. Disponible en: <https://assist-software.net/blog/pipeline-behind-modeling-and-animating-game-character-zelgor>
- Ward, A. (2013). "Kila chapter 5- Skeleton building and binding" en Game Character Creation Series [en línea], [consulta 30 abril 201]. Disponible en: <https://cgi.tutsplus.com/articles/game-character-creation-series-kila-chapter-5-skeleton-building-and-binding--cg-30066>
- Gamedevelopertips 2016. Gamedevelopertips.com [en línea] [consulta 06 mayo 2018]. Disponible en: <http://gamedevelopertips.com/game-development-pipeline/>

Foundry, 2017. "Crafting visual experiences in indie and AAA" en From artist to engine: revolutionizing the game art pipeline [en línea], [consulta 6 mayo 2018]. Disponible en: <https://www.foundry.com/industries/gaming/modern-game-production>

Theodore, Steve. (2016). How do top games studios setup their pipeline for development-Triple A vs. Indie vs. Mobile Games [en línea], [consulta 6 mayo 2018]. Disponible en: <https://www.quora.com/How-do-top-games-studios-setup-their-pipeline-for-development-%E2%80%94triple-A-vs-indie-vs-mobile-games>

Molyneaux, Peter. 2001. "What went wrong" en Postmortem: Lionhead Studios' Black and White [en línea], [consulta 7 mayo 2018]. Disponible en: [http://www.gamasutra.com/view/feature/131476/postmortem\\_lionhead\\_studios\\_.php?page=3](http://www.gamasutra.com/view/feature/131476/postmortem_lionhead_studios_.php?page=3)

Pilestedt, Johan. 2011. Postmortem: Arrowhead Game Studios' Magicka [en línea], [consulta 7 mayo 2018]. Disponible en: [http://www.gamasutra.com/view/feature/134840/postmortem\\_arrowhead\\_game\\_.php?page=1](http://www.gamasutra.com/view/feature/134840/postmortem_arrowhead_game_.php?page=1)

Finley, Alyssa. 2008. "Narrative content development happened late" en Postmortem: 2K Boston/2K Australia's BioShock [en línea], [consulta 7 mayo 2018]. Disponible en: [http://www.gamasutra.com/view/feature/132168/postmortem\\_2k\\_boston2k\\_.php?page=3](http://www.gamasutra.com/view/feature/132168/postmortem_2k_boston2k_.php?page=3)

Spector, Warren. 2000. "Our original team structure didn't work" en Postmortem: Ion Storm's Deus Ex [en línea], [consulta 7 mayo 2018]. Disponible en: [http://www.gamasutra.com/view/feature/131523/postmortem\\_ion\\_storms\\_deus\\_ex.php?page=4](http://www.gamasutra.com/view/feature/131523/postmortem_ion_storms_deus_ex.php?page=4)

Hyvärinen, Kinnunen. 2010. "Conclusion" en Postmortem: Frozenbyte's Trine [en línea], [consulta 12 mayo 2018]. Disponible en: [http://www.gamasutra.com/view/feature/134198/postmortem\\_frozenbytes\\_trine.php?page=6](http://www.gamasutra.com/view/feature/134198/postmortem_frozenbytes_trine.php?page=6)







