

Aaron Rugama García  
aaronrugamagarcia@gmail.com

***RESOLUCIÓN DE PROCESOS PARA LA  
PRODUCCIÓN: LA FIGURA DEL TECHNICAL  
ARTIST COMO VARIABLE  
DIFERENCIADORA EN EL DESARROLLO DE  
UN VIDEOJUEGO***



Universidad  
Francisco de Vitoria  
**UFV** Madrid

Tutor:  
Gabriel Peñas Rodríguez  
gabriel.penas@ufv.es

UNIVERSIDAD FRANCISCO DE VITORIA  
Facultad de Comunicación  
4º Creación y narración de videojuegos, 2020

## **Resumen/Abstract**

El objetivo de este trabajo de investigación es estudiar la figura del Technical Artist desde el marco de la producción y desarrollo de videojuegos, para poder tener una definición clara del rol y comprender cuál es su influencia en el desarrollo de un videojuego. Investigar, contrastar y validar procesos propios de este rol para incluirlos en la administración y dirección de proyectos, que puedan incorporar dichas metodologías en el flujo de trabajo de la empresa, estructurando un sistema y agilizando procesos que resultan en una mejora en desarrollo y producto final.

The objective of this research work is to study the figure of the Technical Artist from the framework of the production and development of video games, in order to have a clear definition of the role and understand its influence in the development of a video game. Investigate, contrast and validate processes of this role to include them in the management of projects, which can quickly incorporate these methodologies in the workflow of the company, structuring a pipeline and streamlining processes that result in an improvement in development and final product.

## **Palabras Clave/Keywords**

Technical Artist, Pipeline, Video games, Project Management, Methodologies

## Índice

1. Introducción.....	3
1.1. Hipótesis.....	4
1.2. Objetivos y metodología.....	4
1.3. Estado de la cuestión.....	5
2. Desarrollo.....	15
2.1. Encuesta.....	15
2.2. Entrevista.....	16
2.3. Experimento.....	17
2.4. Aplicación de sistemas de cola al desarrollo: Little's Law.....	19
3. Análisis de resultados o implementación de contribuciones.....	26
3.1. Representación actual del Technical Artist.....	26
3.2. Experimentación en base a la figura del Technical Artist.....	27
3.3. Incorporación del Technical Artist al modelo de desarrollo basado en sistemas de cola.....	31
4. Conclusiones.....	32
5. Bibliografía.....	34
6. Ludografía.....	36
7. Anexos.....	37

## 1. Introducción

Si nos decidimos a buscar un puesto de trabajo como Technical Artist en la industria del videojuego, o buscamos información sobre que es o aquello que hace como desarrollador, nos encontraremos con que esta visto como un nuevo concepto dentro de la industria del videojuego (Career Explorer), sin embargo, podemos ver perfiles con una experiencia de más de diez años, siendo un rol asentado desde hace tiempo en estudios de desarrollo de todo el mundo.

A pesar de ello, se sigue tratando como algo novedoso tanto fuera como dentro de la industria del videojuego. Este hecho, junto con el amplio abanico de tareas que se atribuyen a esta posición, podría ocasionar en muchos casos una ambigüedad en la definición del trabajo que realiza, lo que podría provocar, en consecuencia, una limitación en el potencial de este.

Esto nos ha llevado a plantearnos un estudio en el que se defina la línea de participación de un Technical Artist en el desarrollo de un videojuego, así como su papel diferenciador en el plan de producción de un proyecto. Para ello, cimentamos nuestro estudio en las implementaciones del rol, en función a las posibilidades que proporciona, parametrizando sus contribuciones y aplicándolas a un sistema de producción basado en la teoría de colas propuesta inicialmente por Agner Krarup Erlang.

Buscaremos además una visión, definición y función del Technical Artist en base a la visión actual del rol en la industria, pudiendo introducir el resultado a través de la Ley de Little (Little, 1954) para calcular el rendimiento del plan de producción con y sin este rol dentro del equipo. Tomaremos por tanto un papel introductorio al *Tech Art*, así como un una ejemplificación e investigación en profundidad de aquellas habilidades y tareas que se buscan en este perfil en función al resto de departamentos y miembros de la compañía.

Una visión actualizada de las técnicas de producción de un videojuego junto con la simulación de un contexto de desarrollo específico será fundamental para analizar el rol desde una perspectiva real, por lo que este trabajo contará con implementaciones concretas que podrán ser replicadas. Esto dará la posibilidad de contrastar y ampliar la investigación en función de las necesidades que requiera cada proyecto, generando una guía con la que plantear la incorporación del Technical Artist (en adelante TA).

## 1.1. Hipótesis

Nuestro supuesto proviene de la multitud de tareas y habilidades completamente diferentes que se requieren (Jam City, 2009) debido a lo cual creemos que la figura del TA aún no está demasiado definida en el desarrollo de videojuegos, de tal forma que su valía termina por diluirse hacia campos asignados a otros roles. Por lo que, actualmente, el rol puede estar incorporado en la producción de una forma indebida, sin exprimir al máximo las posibilidades que brinda esta posición a un proyecto.

## 1.2. Objetivos y metodología

Los objetivos principales de este estudio son los siguientes:

- Definir la figura del TA y su rol dentro del desarrollo de un videojuego.
- Dar una visión concreta de sus responsabilidades y aptitudes, definidas en los diferentes papeles que desempeñan para afianzar el *pipeline* y la arquitectura del sistema/proyecto.
- Proporcionar y validar ejemplos en cuanto a la mejora en producción que realiza la figura dentro del equipo. Respaldar este rol como recurso indispensable para un desarrollo completamente optimizado en términos de tiempo, calidad, capital y cohesión de departamentos y diversidad de roles.

Para la consecución de estos objetivos seguiremos la siguiente metodología de estudio:

El trabajo partirá del análisis exhaustivo del rol para fundamentar las bases del problema en cuanto a lo que se entiende por TA. A través de documentos de desarrollo de videojuegos y experimentos prácticos podremos argumentar los objetivos del trabajo. Gracias a esto, podremos comparar datos empíricos contrastando procesos de desarrollo reales en ambientes cerrados (a través de experimentos individuales en los que usaremos diferentes herramientas y mediremos los resultados) o en desarrollos completos de principio a fin, a través de documentación real.

En primer lugar, se realizará un estudio de la documentación existente sobre el tema a tratar, entorno a la figura, metodologías y procesos ágiles en el desarrollo de videojuegos. Seguidamente, abordaremos el proyecto, con un enfoque cuantitativo y cualitativo, mediante técnicas de investigación directas, metódicas y planificadas, realizando entrevistas y formularios a involucrados en el desarrollo.

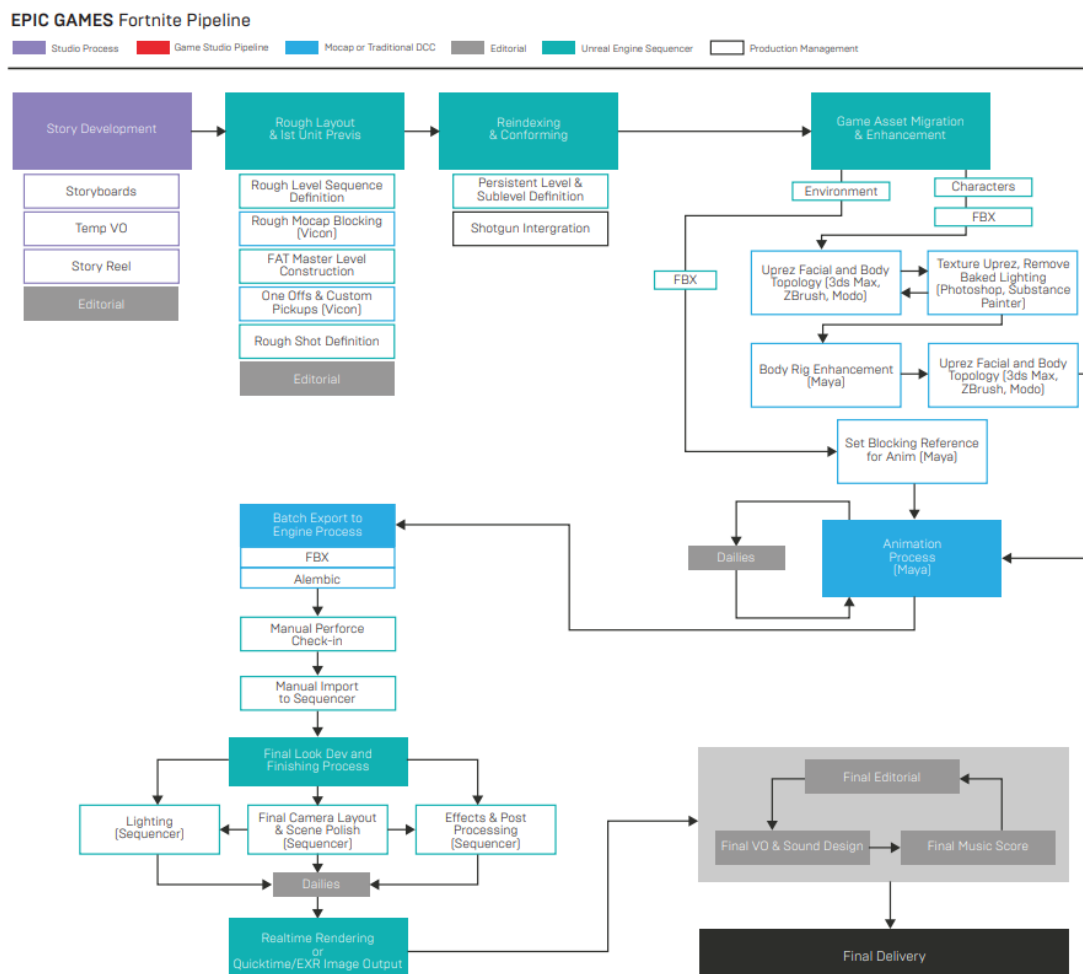
Se realizarán experimentos a diferentes perfiles, tanto artísticos como técnicos, con el fin de comparar distintos procesos de desarrollo que les

propondremos; analizando así, como influye la figura del TA en los mismos. Finalmente, con los resultados obtenidos se espera conseguir definir el rol y su importancia dentro del desarrollo de un videojuego.

La clave de la investigación será la experimentación con procesos reales individuales, la documentación proporcionada de desarrollos de producciones y la propia experiencia de profesionales del sector con perfiles basados en diferentes disciplinas del ámbito del desarrollo video lúdico.

### 1.3. Estado de la cuestión

Pese a la incorporación de nuestro rol (TA) en el desarrollo de un videojuego hace ya unos cuantos años, esta figura aún no está completamente definida a diferencia de otras como *level designers*, *physics programmers*, *concept artists* (Liming y Vilorio, 2011). En esta definición varía según el lugar donde se busque o la especialización de las personas que lo definan. Cayendo finalmente en una acumulación progresiva de responsabilidades que desorientan al profesional de su función principal, la elección y desarrollo de *pipelines* (figura 1), *workflow*, *streamlines*, implementación y cohesión departamental de un proyecto. (Hayes, 2008).



**Figura 1: Fortnite Trailer. Real-Time Art/Visual Pipeline. Fuente: Epic Games.**

La mayor parte de la industria coincide en que es uno de los roles más novedosos actualmente, que más campos de conocimiento requieren y que ayudan al desarrollo general de un videojuego. La responsabilidad que recae sobre él es de especial importancia debido a su papel transversal en el proyecto, de ahí la disolución que sufre el rol, repartiendo este trabajo en tareas individuales. (Thoerzen, 2015).

El rol se diluye, sobre todo, en los estadios iniciales de los desarrolladores, es decir, es realmente difícil saber qué cualidades, aptitudes y conocimientos debe aprender, al contrario que en otros roles donde encontramos un objetivo fijo entorno al que gira el aprendizaje (Liming y Vilorio, 2011). Las definiciones, sin embargo, comparten una cosa en común: la necesidad del TA de ver el proyecto de forma global, reconociendo rápidamente el punto de comienzo y final del desarrollo para, de ahí, trazar la compleja red que une ambos puntos.

Tradicionalmente, nuestro rol es aquel que actúa como puente entre programadores y artistas dentro del desarrollo de un proyecto. Las cualidades que les son atribuidas en redes como Career Explorer<sup>1</sup>, Get In Media<sup>2</sup> o Screen Skills<sup>3</sup>, comparten en esta descripción el carácter creativo, original e intuitivo, abiertos por completo a la investigación e innovación de nuevas técnicas con las que ampliar los horizontes del proyecto en el que participan.

Con este conjunto de cualidades, necesita desarrollar una serie de habilidades para poder llevar a cabo este trabajo. Para ello, volviendo a las definiciones colectivas que podemos encontrar en cualquier espacio especializado de trabajo, el TA debe contar con amplios conocimientos técnicos. Por supuesto, debe contar con una visión global del proceso de creación de arte en videojuegos, conocer las herramientas y software existente, con las posibilidades que estos programas brindan; de tal forma que pueda canalizar el arte desde sus estadios iniciales en un *concept art* hasta el resultado final en la plataforma correspondiente.

Para ello, se requieren tanto habilidades artísticas como de programación, requiriendo un conocimiento de diferentes lenguajes de programación, además de un amplio conocimiento de las API<sup>4</sup>s o librerías gráficas que tenemos a nuestro alcance, con sus respectivos lenguajes de *shader*. No obstante, el conocimiento de estos lenguajes, finalmente radica en la comprensión de cómo

---

<sup>1</sup> Career Explorer, Technical Artist: <https://www.careerexplorer.com/careers/technical-artist/>

<sup>2</sup> Get in Media, Technical Artist: <http://getinmedia.com/careers/technical-artist>

<sup>3</sup> Screen Skills, Technical Artist: <https://www.screenskills.com/careers/job-profiles/games/technical-art/technical-artist/>

<sup>4</sup> API: (*Application Program Interface*). Set de rutinas, protocolos y herramientas para desarrollar aplicaciones de *software*.

funciona un sistema micro-programable y cómo, a través de la computación gráfica, se ha llegado al entorno gráfico de trabajo habitual en los motores de videojuego o cualquier *DCC*<sup>5</sup> *software*.

Estos conocimientos técnicos que, a priori, podrían parecer excesivos para un perfil que incluye la palabra artista en su nombre, radican en la comprensión que el TA hace a posteriori para las tareas de optimización, canalización de datos o tratamiento de rutinas y bases de datos. Siendo una figura clave en la resolución de problemas y creación de herramientas tanto para artistas como programadores.

Además de esto, claro está, tenemos el rol artístico; que no solo consiste en la creación de *shaders* o efectos visuales, sino en la creación de los propios modelos, texturas, materiales, UVs, iluminación, post procesado, renderizado, *rigging*, animación o incluso *look dev*. Para ser muchas veces responsables además de la simulación de físicas de tejidos, fibras, músculos, destrucción e infinidad de procesos procedimentales.

Tras todo esto, cuando repasamos la lista de conocimientos, tareas y habilidades, podemos observar rápidamente que esta se extiende abordando una gran cantidad de apartados del desarrollo de un videojuego, cuyos desempeños tienen asociados roles concretos. Entonces, ¿cómo lidia el TA con todo esto? La respuesta es sencilla, es el puente entre todas estas prácticas a la hora de desarrollar el proyecto. No tiene por qué ser el que realice estas tareas, pero sí que las supervise y las dote de cohesión.

Aquí es donde se encuentra el papel de administración y producción del rol en un proyecto, su conexión directa con los departamentos de desarrollo. Así pues, para tener una visión clara y esquematizada de la participación de este rol en las diferentes secciones y prácticas de desarrollo, podemos elaborar un esquema que nos ayude a sintetizar la labor de esta posición.

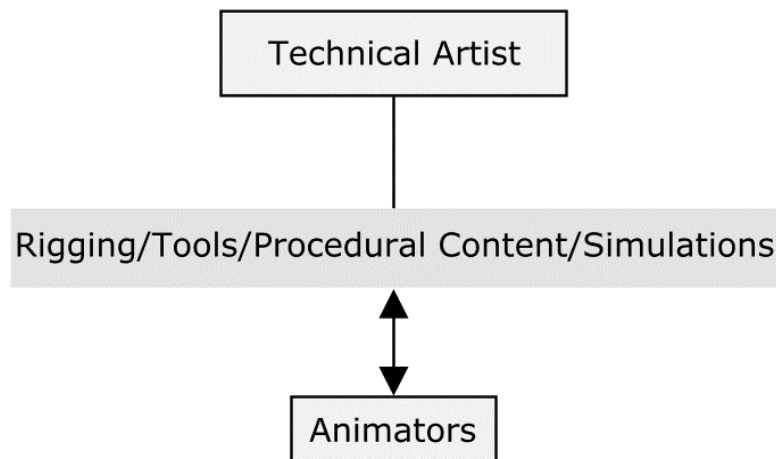
En base a diferentes estudios, esquemas y metodologías de trabajo de diferentes desarrolladoras de videojuegos, así como TAs individuales, podemos elaborar este esquema, basándonos principalmente en la visión de esta cuestión de algunos referentes como Jason Hayes (Lead Technical Artist, Rockstar) en "*The Code/Art Divide: How Technical Artist Bridge the Gap*", *Tech Art Aid*, Riot Games "*So you wanna make games??*", "*Understanding Tech Art*" de Ryan Brucks o "*Video Games: An introduction to the industry*" de Andy Bossom y Ben Dunning, donde hacen una recopilación exhaustiva de los roles existentes en la producción y desarrollo de videojuegos, además de recrear el recorrido y futuro desarrollo de esta industria.

---

<sup>5</sup> DCC: (Digital Command Control) estándar para operar en sistemas digitalmente.



Podemos distinguir entre los roles, secciones o departamentos con los que nuestro rol tiene conexión, y los procesos y prácticas con los que cohesiona dichos departamentos. Así, por ejemplo, se representa el esquema con Animación, propia del rol de animador, y el proceso o técnica por el cual el TA participa de ello serían las *tools*, el *rigging*, la simulación de físicas en determinados huesos del esqueleto, sistemas procedimentales o de *blend shapes*, etc. (Figura 2).



**Figura 2:** Proceso de relación entre roles y Technical Artist. **Fuente:** Propia.

Siguiendo este esquema, podríamos acotar al TA a nueve roles o departamentos con los que tiene una conexión directa:

- **Texture Artists:** viene dada por la implementación de texturas en el motor gráfico, el tratado de ellas, la aplicación de técnicas con las que conseguir mayor eficiencia a través de diferentes canales, la preparación de estas para crear efectos más tarde o incluso la creación de máscaras y mapas especiales para efectos de postprocesado y shaders
- **VFX:** se requieren tanto capacidades técnicas en creación de texturas, como conocimiento de sistemas de partículas y efectos geométricos, siempre teniendo en cuenta las restricciones del hardware (Bossom y Dunning, 2015). Debido a su papel transversal requiere que se mantengan en contacto con diferentes departamentos, tanto diseño como arte, para conseguir el efecto adecuado, manteniéndose al tanto de los requerimientos técnicos del proyecto.

El papel del TA en este departamento, además de la estandarización del *pipeline* para llevar a cabo en la creación de estos efectos, cobra

importancia en la incorporación de sistemas procedimentales y simulaciones.

- **Animators:** sienta las bases de este proceso en cuanto al trabajo de animación mediante *rigging*, *blend shapes* o *mocap*. Para que un animador pueda sentirse cómodo realizando su rol, debe contar con todas las herramientas posibles a su alcance, además de unos controles específicos para cada proyecto: trabajar en personajes *in game*, secuencias cinemáticas, *keyframes*, sistemas de locomoción o cualquier otra acción que conlleve movimiento (Bossom y Dunning, 2015).

Como ya hemos introducido antes, la variable diferenciadora en este proceso es la introducción de sistemas procedimentales en estas animaciones, la posibilidad de crear canales de compartición de animaciones entre modelos, la creación incluso de herramientas que automaticen procesos que pueden alargarse en el tiempo con los métodos tradicionales o investigar y desarrollar soluciones para problemas específicos.

- **Game Designers:** de acuerdo con Bossom y Dunning, el game designer tiene que ser capaz de dirigir a la audiencia a través de las estructuras narrativas, balanceando las mecánicas de juego en pos de las expectativas del proyecto que se está realizando, teniendo en cuenta el cronograma al que se ajusta la producción. Así que como no, cualquier ayuda para incrementar los procesos de *testing* de nuevas mecánicas e interacciones será beneficioso para el desarrollo del videojuego, muy en consonancia con los Tools programmers y UX designers.
- **Lighting y Character/Environment/Level/Props artists:** en cuanto a la iluminación y el arte 3D en un videojuego, tenemos una infinidad de variables que pueden aparecer durante el desarrollo. Cada proyecto cuenta con un plan de producción específico, un estilo artístico determinado y un estándar de plataformas en las que va a ser lanzado.

Este y muchos otros factores alteran el flujo de trabajo de los artistas a la hora de esculpir, modelar, texturizar, renderizar e iluminar; por ello necesitan fijar unos estándares a la hora de crear un flujo de trabajo que todo el proyecto debe seguir. El artista debe tener en cuenta, además de los requerimientos visuales, las exigencias técnicas del motor gráfico en cuanto a optimización y procesamiento.

Estas limitaciones, o tratamientos específicos las creará el TA en base a las variables que imponga dicho proyecto, como estilo artístico, motor gráfico, plan de producción o tamaño del equipo; estableciendo así un

canal de trabajo apropiado al proyecto, según sus cualidades y necesidades.

- **Graphic Engineers y QA:** la labor conjunta de nuestro rol junto con Graphic Engineers y Quality Assurance radica en la optimización visual del proyecto completo. Tras o durante la implementación de modelos, efectos, animaciones, entornos, texturas y demás *assets*<sup>6</sup> del proyecto, este equipo busca formas de incrementar el rendimiento gráfico del proyecto, estableciendo y elaborando perfiles según qué plataformas, niveles de renderizado y comportamiento del motor en el flujo de datos gráficos.
- **Pipeline y Tool Programmers:** en cuanto a la creación de herramientas tanto como para el equipo de desarrollo como para el propio motor gráfico, sea propio o no, en la gran mayoría de casos se necesitará hacer algunos cambios o modificaciones, además de agregar nuevas funcionalidades.

Muchos de estos requerimientos van siendo añadidos por los ingenieros del sistema pero muchas veces son necesarias una serie de características que, o no nos las proporciona el motor, o nos lo proporciona de una forma diferente. Estas características pueden formar parte de cualquiera de los ámbitos del juego, desde el comportamiento de las físicas hasta herramientas que proporcionen un canal de distribución de textos para los diálogos del videojuego.

- **Learning:** pese a no ser una sección o departamento como tal, una gran parte del tiempo del TA trata de crear documentación así como resolver dudas y problemas dentro del proyecto. Toda esta serie de herramientas propias, investigación y desarrollo de técnicas nuevas o específicas pueden ser desconocidas para el resto del equipo.

Esto quiere decir que, al igual que hablábamos antes del papel cohesionador de nuestro rol, este también debe comunicarse con los ingenieros y transmitir los mensajes al departamento de arte, muchas veces haciendo de traductor entre programación y arte, y viceversa. Enseñar e introducir además nuevas técnicas al entorno de trabajo, así como buenas prácticas a la hora de desarrollar cualquier modelo, efecto, animación o textura. Esto es sin duda fundamental para evitar posibles problemas y contratiempos en el futuro del proyecto.

- **Producers:** por último los productores, uno de los roles que más nos conciernen en este trabajo, son y deberían ser una conexión crucial entre equipo de desarrollo y proyecto de videojuego. Según la definición que

---

<sup>6</sup> Asset: aquellos recursos que pueden ser incluidos en el motor gráfico.

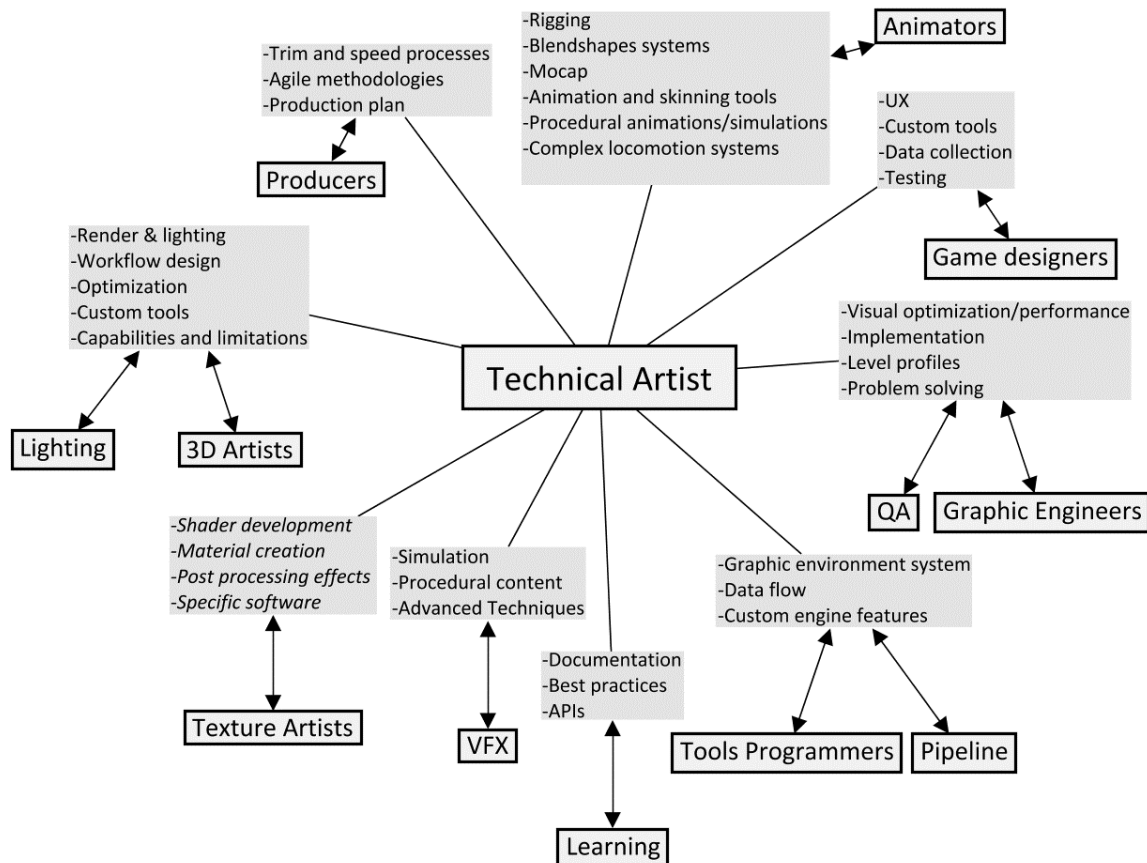
nos dan Bossom y Dunning: “el productor es el individuo en cuyos hombros se aguanta el proyecto, con la principal preocupación del proceso logístico del desarrollo, asegurar, ajustar y cumplir los tiempos de producción, además de fijar y no sobrepasar el presupuesto gestionando el equipo”.

“El trabajo del productor es hacer que todo funcione sin problemas, todo el mundo esté feliz y nadie se sienta oprimido.” Dylan Bale, Chief Production Officer en Born Ready Games y Edge Case Games, entrevistado en *Video Games: An introduction to the industry*.

TA y Productor tienen la función compartida de unificar departamentos. En muchos casos ejercen como negociadores velando por ambos. (Oscar Świerad, 2018). Así como el Productor introduce metodologías de trabajo ágiles para conseguir marcar los tiempos de producción, el TA ayuda a conseguir esos tiempos mediante su papel en el desarrollo. Toda la influencia que ejerce sobre cada uno de los departamentos y aspectos que hemos visto en los apartados anteriores, habilitan una serie de posibilidades en el plan de producción.

El Productor marca el paso del equipo de desarrollo, y el TA puede y debe acelerarlo mediante sus capacidades técnicas y de producción. Gracias a esta sinergia, se podría producir un cambio sustancial en la producción de un videojuego, acortando procesos que el Productor puede tener en cuenta al elaborar su plan.

No obstante, este impulso o estimulación en el desarrollo natural del proyecto con la variable de la figura del TA, debe asumir también la variable R&D (*Research and Development*), por la que el rol adquiere una dimensión de I+D+i, de ahí su carácter investigador y creativo. Esto se traduce en cargas de tiempo y trabajo en la búsqueda de soluciones, nuevas tecnologías e implementación de dichas herramientas y características en el flujo de trabajo del equipo.



**Figura 3:** Diagrama de participación interdepartamental del Technical Artist. **Fuente:** Propia.

En cuanto a la definición que nos dan Bossom y Dunning sobre el rol de TA, suscitan y hacen hincapié de nuevo en su papel de puente entre los equipos de arte y programación. También afirman que el individuo que llega a esta posición puede venir tanto de la disciplina del arte como de la programación ya que, como hemos visto (Figura 3), trabajan tanto en los visuales del videojuego como en la tecnología del motor de videojuego, pasando por la serie de prácticas que hemos ido viendo a lo largo del trabajo: *tools*, *VFX*, implementación, creación de *shaders*, *pipeline*...

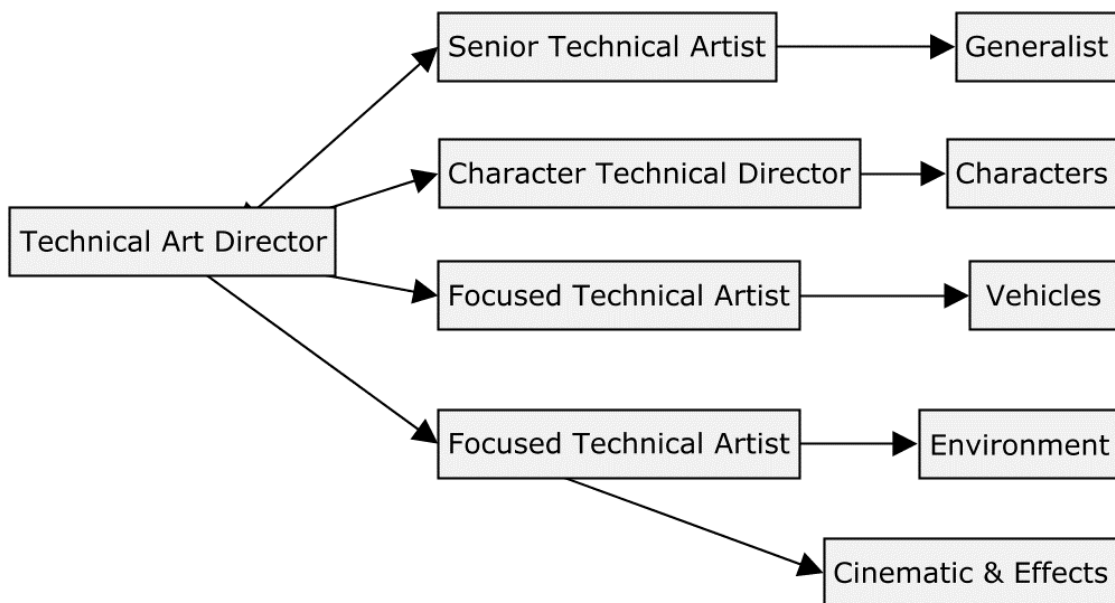
Sin embargo, para enriquecer esta definición, podemos ver como Jason Hayes ahonda más en cuál debería ser el papel del TA. A grandes rasgos resuelve el dilema concretando que esta figura debería ser capaz de diseñar y desarrollar todo el canal de trabajo de arte necesario para el proyecto. “Parte del rol es ser arquitectos de *pipelines* y *sistemas*” (Hayes, 2008). Aquí vemos una dimensión fundamental en la que planificar metodologías y flujos de trabajo se convierte en una de las cualidades primordiales. Estas se basan en un conocimiento muy extenso de hardware y software, así como en muchos casos matemático, físico y artístico. El apoyo del equipo en el puesto es casi siempre recíproco, retroalimentando el conocimiento de cada departamento, finalmente evolucionando el proyecto a un estadio superior.

Nuestro rol no solo diseña y especifica estos sistemas de trabajo (en coordinación con otras disciplinas), sino que es el que impulsa y defiende los cambios. Debido a su familiaridad con estos, los convierte en la vía principal de acción que debe seguir el equipo, y corrigen errores en las herramientas desarrolladas para apoyar la canalización de datos y trabajo (Hayes, 2008).

Viendo y asentando estos conocimientos y habilidades cabe esperar que, entre tanto trabajo, haya tipos de especialización. El rol, entonces, se puede ultra especializar en diferentes ámbitos o secciones del *Tech Art*. Con el paso del tiempo y la mejora del software y sistemas de desarrollo de videojuegos, hemos empezado a dotar a ciertos roles con el adjetivo “Technical”, al igual que pasa con el TA. Es normal dada la cantidad de técnicas que un desarrollador tiene a su alcance a la hora de crear un videojuego y como una misma tarea puede realizarse según aquello que queramos conseguir.

Para ilustrar esto último podemos ver el caso del Technical Animator y el Animator. Según Anthony Castoro (CEO, Heatwave Interactive) la diferencia entre estos yace en que el primero es quien hace la estructura que posibilita la animación del personaje, mientras que el Animator es quien da el movimiento al personaje. Observamos esto cuando pensamos en tecnologías de *motion capture*, complejos sistemas de locomoción, *rigging* facial basado en *blendshapes*, animación procedimental basada en físicas, etc.; y cómo desarrollar la arquitectura de estos sistemas, para que fluyan a tiempo real tanto en el motor gráfico como en la *build* final. Así se ve claro que esta distinción técnica de creación de sistemas de animación, y la propia animación en *keyframes* se divide en dos roles complementarios, Animator y Technical Animator.

Por lo tanto, podemos especializar al TA en disciplinas, en función de las propias necesidades y envergadura del proyecto. Dividiendo el rol en generalista, de personajes, vehículos, entornos y cinemáticas o efectos, así como un director dentro del departamento de *Tech Art* (Figura 4).



**Figura 4:** Estructura típica del equipo de arte de Volition. **Fuente:** Jason Hayes.

Por último, deberíamos hablar de la implementación de este rol en el estudio y cómo estructurar el equipo a la hora de contar con esta variable en el desarrollo. En este punto de la cuestión, actualmente, esta posición puede ser una variable diferenciadora en la producción de un proyecto, pero cómo y hasta qué punto podremos verlo a través de ejemplos concretos, según el estado del proyecto, sus necesidades, el tamaño del equipo y otras variables que hay que estudiar desde el plan de producción, incluyendo por ejemplo, un total de tres o cuatro TAs en un equipo de ochenta o noventa personas según afirma Jason Hayes.

Son muchas las tareas, habilidades y competencias que se le atribuyen a nuestra figura, por lo que acotar y centrar los esfuerzos de este rol es vital para el funcionamiento de los engranajes del proyecto. Como ya hemos visto, la cohesión departamental y las nuevas posibilidades que incluye al desarrollo son cuantiosas y variadas, siempre y cuando se ponga el foco de atención en aquellas tareas primordiales de diseño y desarrollo de flujos de trabajo, así como investigación y aplicación de nuevas tecnologías.

## 2. Desarrollo

Como hemos visto anteriormente, la implicación de nuestro rol en el proyecto incrementa las posibilidades que tenemos a la hora de crear flujos de trabajo. Esta influencia interdepartamental pivota sobre el departamento de *Tech Art*, por ello puede funcionar como papel vital en el plan de producción de videojuegos.

En esta investigación vamos a abordar estas dos situaciones, tanto a nivel de estudio como individual, así como observar las diferencias entre su influencia en distintos perfiles, equipos y situaciones. Para ello basaremos el estudio en pruebas experimentales con herramientas de desarrollo concretas, comparando los procesos con y sin nuestra variable diferenciadora. A su vez, correlacionaremos datos en base a encuestas y entrevista, variando entre diferentes roles, así como TA profesionales de pequeños estudios, compañías medianas y grandes empresas.

Con este acercamiento podremos contrastar la visión del perfil respecto al equipo y proyecto, viendo cómo se comporta nuestro rol en diversas situaciones y si lo definido anteriormente se cumple. No obstante, el acercamiento previsto debería variar según “Work for Play” o “Video Games: An Introduction to the Industry”, por lo que las respuestas se diferenciarán en función de variables que pretendemos reconocer a través de este estudio del rol. Así pues, desde este punto de partida la investigación estará comprometida a estos factores situacionales y a como percibirlos en una situación real.

El siguiente punto en el trabajo es el acercamiento del TA a su rol de producción: planes de trabajo como *pipelines* o *workflows* y su labor social del equipo para trabajar de primera mano con los Productores del proyecto. Una de las principales preguntas de este estudio se refiere a cómo participa en esta cohesión del proyecto y si su valor aumenta cuando ejerce un papel de producción a través de la inclusión de metodologías ágiles, por lo que debería ultimar un ahorro en costes y tiempo de producción.

En base a esto, la investigación se dividirá en tres fases, donde analizaremos pormenorizadamente aquellos datos que resulten de la encuesta, los experimentos y las entrevistas, así como el estudio en cuestión de la aplicación de los sistemas de cola al desarrollo de un videojuego, más concretamente a través de la Ley de Little (Little’s Law, 1954).

### 2.1. Encuesta

Para realizar la investigación es necesario recoger y contrastar los datos y experiencias reales que involucran a la figura del TA. Por esto la realización de una encuesta es indispensable para la correcta interpretación del rol en un ambiente de trabajo. No obstante, la recopilación de datos no se limitará a



profesionales del sector, sino que ampliaremos la muestra a desarrolladores y diseñadores de contenido audiovisual de otras industrias como el cine o la televisión, así como estudiantes, jugadores o seguidores del videojuego.

Uno de nuestros principales objetivos consiste en concretar el rol por lo que necesitaremos una visión global de este trabajo, así como de los respectivos perfiles de una compañía. Por el otro lado, la incorporación de estudiantes y jugadores nos permitirá reconocer la visibilidad del rol en el público y los perfiles no profesionales, pudiendo contrastar y cruzar estos resultados con los obtenidos según los encuestados, en base a su rol y situación.

Con la utilización de una muestra tan abierta pretendemos aportar profundidad y nuevas incógnitas a la investigación, por lo que podremos dar paso a futuros estudios en base a los resultados.

La realización de la encuesta se ha dividido según la situación y entorno del encuestado respecto a nuestro rol, por lo que podremos diferenciar los resultados en diferentes bloques. Estos bloques se dividen principalmente en “trabajo profesionalmente en videojuegos”, “estudio videojuegos”, “soy jugador o seguidor del sector” y por último “trabajo en otro sector del diseño y desarrollo audiovisual”. Esta selección aborda el puesto desde diferentes ámbitos, pensando en variables como “experiencia”, “rol” o “sector” de la muestra.

Una vez realizada la división en perfiles, subdividimos de nuevo algunos de ellos según especializaciones. Así aquellos que trabajen de forma profesional en videojuegos podrán diferenciarse entre TA u otro, especificando en tal caso cuál es su rol. Durante este aterrizaje hemos sentado las bases de sus respuestas y podremos correlacionar perfiles con seguridad, para que las preguntas siguientes tengan una posible explicación, ampliando a un nivel más la recogida de datos.

En cuanto a la batería de preguntas principal, analizaremos cuales son las expectativas de aquellos perfiles sobre nuestro rol y como ha sido la participación y percepción del mismo en los diferentes proyectos en los que el encuestado ha participado. Estas preguntas están orientadas tanto a procesos técnicos (*Shader Development, Scripting, Rigging...*) como a procesos de producción (planes de producción y trabajo, *pipeline, workflow*) para poder estudiar sinergias entre los diferentes perfiles y departamentos con la figura del TA como hemos analizado previamente.

## **2.2. Entrevista**

Estas entrevistas se dirigirán exclusivamente a TAs. El objetivo de estas es recopilar ejemplos, definiciones (Anexo 1) o situaciones específicas en las que el rol se ve involucrado. Así tendremos información obtenida de primera mano por especialistas en este ámbito, pudiendo apoyarse el estudio en

diferentes profesionales, ilustrando los resultados a través de la experiencia de estos.

### 2.3. Experimento

En esta fase de la investigación realizaremos dos experimentos basados en la realización de tareas individuales, según las necesidades de diferentes proyectos. Las pruebas se llevarán a cabo de forma individual bajo supervisión, indicando qué prácticas se deben realizar. Estos experimentos han sido desarrollados específicamente para esta investigación en base a aquellos datos que se quieren recabar, según las tareas que se llevan a cabo en un estudio de videojuegos.

Ambos experimentos se centran en la agilización de procesos mediante técnicas y herramientas desarrolladas por el TA. En estos casos hemos acotado el objetivo para escalar la prueba a un entorno simplificado en el que poder llevar a cabo el experimento, para obtener finalmente estas dos pruebas basadas en incrementar la velocidad de desarrollo de una tarea específica, con un contexto plausible:

- **Exportación de objetos:** en este caso, partimos de la situación en la que nuestro desarrollo necesita un diseño modular. Para ello, el equipo de *Environment Art* ha creado una serie de *assets* en los que divide el escenario de forma que, a continuación, puedan elaborar un entorno adecuado a las especificaciones del departamento de Diseño de niveles.

La creación de estos *assets* ha sido realizada en 3DS Max en una misma escena y el objetivo consiste en exportar estos *assets* a una carpeta de forma individual, correcta y ordenada, una tarea que podríamos calificar como tediosa cuando se trata de muchos objetos.

El experimento consistirá entonces en conseguir este objetivo con y sin la figura del TA en el proceso. La realización de la tarea sin este será libre, tan solo deberá exportar dichos objetos como se indica. A continuación, se repetirá el proceso pero esta vez con una pequeña herramienta desarrollada para acelerarlo.

Esta herramienta tan solo será un *script* (Anexo 2) con el que exportar varios objetos al mismo tiempo, conservando su nomenclatura y modificaciones al igual que el método por defecto de 3DS Max. Por lo tanto, el experimento se realizará con y sin la implementación de dicho código en el proceso de exportación.

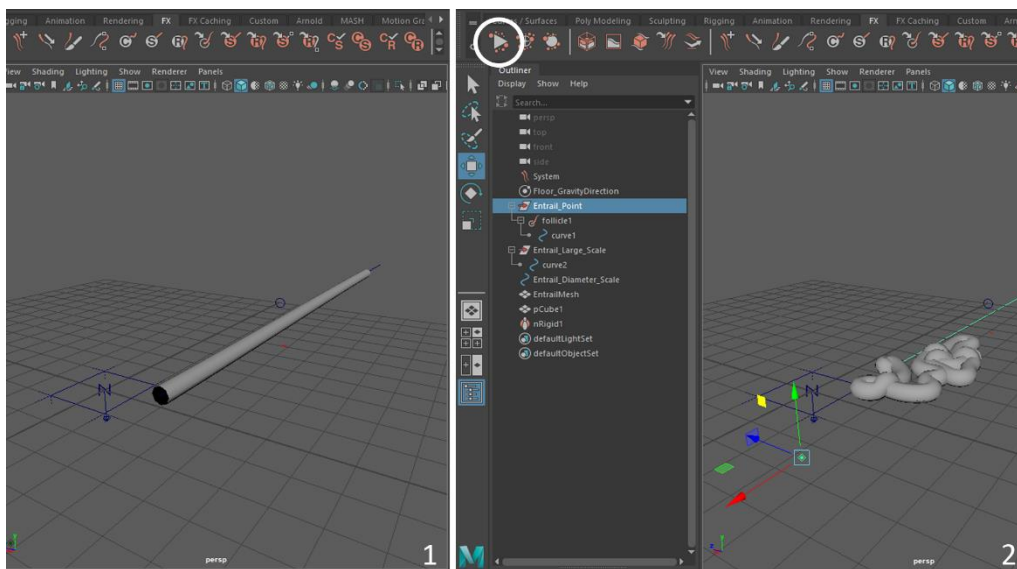
- **Creación de Assets:** en esta práctica partiremos de la situación en la que un videojuego de terror necesita agilizar el proceso de creación de ciertos *assets* en Maya, que tendrán mucho peso en el escenario para crear una

atmósfera propia del género. Desde la dirección de arte, se ha decidido crear un escenario de temática gore, así que los artistas 3D deberán crear una gran cantidad de entrañas, vísceras, sangre, etc.

Los modeladores, han recalcado que la creación de entrañas realistas y variadas es un proceso tedioso y complejo si además el departamento de VFX quiere realizar todo tipo de efectos y simulaciones con ello. El objetivo de este experimento será entonces, crear la malla de esas entrañas, así como preparar el mapeado de UVs de forma vertical para que se pueda aplicar fácilmente un *shader* con una función que permita el desplazamiento de los vértices.

En este caso, estudiaremos al igual que antes la realización de esta tarea con y sin la participación de nuestro rol. La práctica sin participación será libre, sin ningún tipo de método o técnica a seguir, tan solo el planteamiento que elija seguir el sujeto. Por el otro lado, tendremos el mismo proceso a través de una técnica ideada por el TA.

Esta técnica consiste en la creación de entrañas de una forma dinámica utilizando las herramientas de simulación de Maya (Figura 5). En este caso se crea un sistema con el que poder crear los assets con un cilindro, para después subdividirse y recrear la animación de unas entrañas cayendo al suelo a través de un *nHair*<sup>7</sup>. Esta simulación es completamente procedimental por lo que nunca habrá un *asset* igual que otro.



**Figura 5:** Técnica de simulación de físicas y gravedad basado en *nHair* de Maya. **Fuente:** Propia.

<sup>7</sup> *nHair*: <https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/Maya-CharEffEnvBuild/files/GUID-AB32EA9E-90CE-4272-B523-D326864E9638-htm.html>

## 2.4. Aplicación de sistemas de cola al desarrollo: Little's Law

Por último, tras desarrollar este estudio, incorporaremos y contrastaremos resultados obtenidos con el teorema *Little's Law*, viendo cómo afecta la figura del TA en flujo de trabajo de los diferentes departamentos de un proyecto. Esta ley se aplica en sistemas de colas de espera (*queuing systems*) que contienen unidades (*items*) que llegan al sistema, pasando a ser unidades en espera que se convertirán en unidades procesadas cuando acabe su tiempo de espera en la cola, una vez completan el proceso salen del sistema y dan paso a nuevas unidades.

Little's Law establece que el número promedio de *unidades en el sistema*( $L$ ) es igual a la *tasa de unidades que llegan al sistema por unidad de tiempo*( $\lambda$ ) multiplicado por el tiempo estimado de espera que una unidad permanece en el sistema( $W$ ).

$$L = \lambda W$$

**Figura 6:** Teorema de Little's Law. **Fuente:** Jhon Little.

Esta aplicación se puede dar en cualquier sistema de colas de espera, desde una cola de clientes en un supermercado hasta una cadena de montaje de coches. En esta investigación aplicaremos el teorema a un sistema de desarrollo de un videojuego, basándonos en el estudio realizado por Justin Fischer en *Game Planning with Science*.

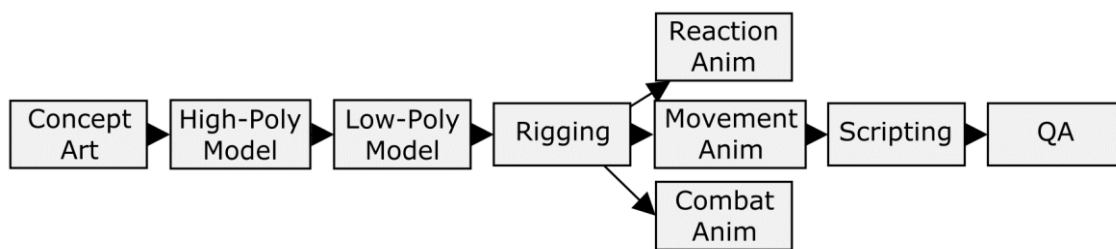
La elección de esta fórmula, pese a necesitar un sistema estable y no preventivo, viene en base al sistema de flujo de trabajo típico en una empresa de videojuegos con metodologías ágiles como Scrum o Kanban. Si bien es cierto que en la producción de un videojuego tenemos multitud de variables que pueden alterar los resultados de este teorema, podemos, sin embargo, parametrizar el tiempo de desarrollo y optimizarlo en función del contexto al que nos enfrentemos, preparando un escenario ideal donde introducir la variable del TA.

Para poder ver la utilización práctica de este teorema, partiremos de la situación planteada por Justin Fischer sobre el proceso de desarrollo de una tarea en un videojuego. Su planteamiento imagina cualquier proceso de creación como una tarea subdividida en tareas más pequeñas. Dichas tareas son las unidades que Little nos presenta en su teorema, unidades de trabajo que deben fluir a través del sistema para poder finalizar esa tarea mayor. Para Fischer dicha tarea es, por ejemplo, la creación de un personaje.

Como en cualquier diseño de un flujo de trabajo, debemos determinar cuáles son aquellos pasos que debemos dar para conseguir el trabajo que proponemos hacer. Un *pipeline* que todos los trabajadores sigan, para poder

determinar la dirección y avance de la producción de una forma conjunta. Siempre surgirán cambios o situaciones impredecibles a lo largo del proyecto, pero desde un primer momento debemos tener claro cuáles son los pasos a seguir para completar satisfactoriamente y en plazo las tareas asignadas a cada departamento. Si tenemos esto claro desde un principio, podremos pivotar en otras direcciones cuando los problemas surjan.

El *pipeline* diseñado para el videojuego, tendrá sus particularidades según la naturaleza del proyecto, en este caso, este es el diagrama de flujo sugerido por Fischer a la hora de crear un personaje.



**Figura 7:** Diagrama de flujo en un caso hipotético para desarrollar un personaje.

**Fuente:** Game Planning with Science.

Cada una de estas cajas serán nuestras unidades a lo largo de nuestro flujo de trabajo en cualquier desarrollo. Como ya hemos señalado antes, determinar esto será crucial a la hora de elaborar nuestro plan y poder ver el teorema de Little en práctica. Identificar cuáles son las tareas a realizar para conseguir el resultado esperado es, además, plantear el escenario al que el TA se ve sujeto a la hora de idear fórmulas con las que facilitar el progreso de estas unidades a lo largo del sistema.

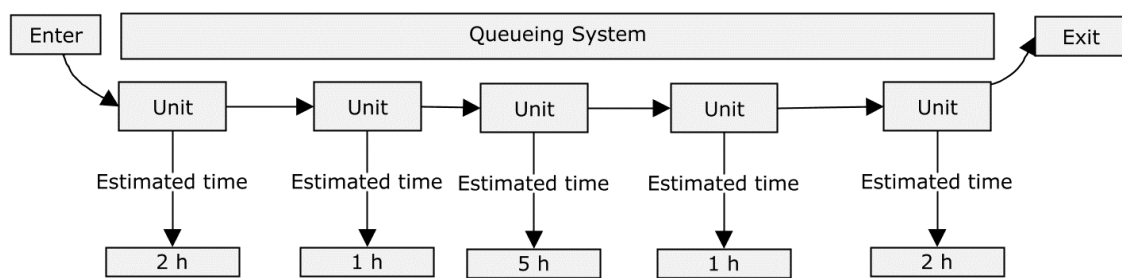
El siguiente concepto que Fischer introduce son los cuellos de botella. Es aquí donde creemos que el rol podrá tener un papel fundamental en la resolución de procesos con los que disminuir los tiempos de espera entre unidades. Y es que así es como se forman los cuellos de botella, creando acumulaciones de unidades en determinados lugares del sistema. Estas unidades, pasan a ser unidades en espera, por lo que retrasan la producción en dicho *pipeline*.

Estos casos se ven inicialmente en la estimación de trabajo que le atribuimos a las unidades, es decir, el tiempo físico, sin interrupciones, que una unidad necesita para ser realizada. Si en el sistema propuesto, el Concept Artist realiza cinco conceptos de personaje en el tiempo que el artista 3D esculpe un modelo *high-poly*, diríamos que no ocurriría nada grave en el desarrollo normal de la creación del personaje, ya que son necesarios diferentes diseños hasta dar con el adecuado.

Sin embargo, si el proyecto conlleva desarrollar multitud de personajes y el siguiente artista 3D termina la retopología del personaje, en un tiempo inferior

al que Concept Artist y el encargado de *high-poly* realizan su trabajo con nuevos personajes, nuestro cuello de botella empezará a formarse. Estos tiempos de espera hasta que la unidad avance hasta la siguiente tarea, van incrementando a medida que las unidades avanzan en el sistema haciendo que en determinados ciclos se formen colas en las que las unidades esperan demasiado tiempo, provocando atascos en el desarrollo.

Estos cuellos de botella pueden formarse también de forma inversa. En este caso el artista 3D encargado de realizar la retopología no tiene unidades en las que trabajar, por lo que el trabajo deja de fluir por una cuestión de falta de tareas. En el otro caso, tenemos una unidad que para realizarse necesita una cantidad superior de tiempo con respecto a las demás. Esta unidad, si cuenta con la misma cantidad de recursos para ser desarrollada, que el resto de las tareas, será claramente un nuevo cuello de botella, ya que las unidades anteriores se acumularán en espera. Ambos casos son complementarios ya que de una manera u otra se forman cíclicamente a lo largo del desarrollo, digamos que se retroalimentan en base al tiempo que las unidades permanecen en el sistema.



**Figura 8:** diagrama propio de un sistema de cola. **Fuente:** propia.

Una vez planteados los elementos de este sistema, podemos traducir numéricamente nuestro flujo de trabajo y gestionar el sistema de cola para identificar y resolver estos cuellos de botella, a través de una tabla de capacidad. En esta tabla calculamos la capacidad de proceso que tiene nuestro sistema, es decir, podemos aproximar el valor temporal que tendrán las unidades, el porcentaje de progreso por ratio de tiempo o la utilización de recursos activos para procesar las unidades que entran y salen del sistema. Para mostrar esto, Fischer elabora la siguiente tabla según el *pipeline* sugerido anteriormente.

Resource Capacity						
Resource	Unit Load (Days to make that field for one character)	Unit capacity (Characters/day)	Number of team members	Total characters per day	Process Capacity	Resource Utilization
Concept Artist	2.00	0.50	1.00	0.50	0.17	33.33%
HighPoly	5.00	0.20	1.00	0.20		83.33%
LowPoly	3	0.50	2.00	0.67		25.00%
Rigger	1	0.50	1.00	0.50		16.67%
Reaction Animator	3.00	0.33	1.00	0.33		50.00%
Movement Animator	5.00	0.20	1.00	0.20		83.33%
Combat Animator	7.00	0.14	2.00	0.43		58.33%
Scripter	6.00	0.17	1.00	0.33		100.00%
QA	3.00	0.33	1.00	0.33		50.00%

**Tabla 1:** Caso hipotético de tabla de capacidad de recursos para el proceso de desarrollo de un personaje de videojuegos. **Fuente:** Game Planning with Science.

En la Tabla 1 podemos observar el tiempo promedio en el que una unidad se realiza (*Unit Load*), el progreso de esa tarea en una unidad de tiempo, en este caso, un día (*Unit Capacity*), el número de miembros que realizan dicha tarea (*Number of team members*) o el progreso de la tarea en función de los miembros que participan en ella (*Total characters per-Day*). Pero donde realmente podemos observar la practicidad de esta fórmula es en la capacidad de proceso (*Process Capacity*) y en el porcentaje de recursos utilizados (*Resource Utilization*).

La capacidad de proceso nos permitirá identificar el porcentaje en el que las unidades están siendo desarrolladas, es decir, si utilizamos un 60%, dicho recurso se mantendrá un 40% de su tiempo en espera. Inicialmente, bajo esta situación, lo esperado sería intentar conseguir que los recursos mantuvieran un porcentaje muy elevado para disminuir al mínimo el tiempo de espera, pero en última instancia, esto produce un error mayor, ya que si una unidad llega al límite de su capacidad de recursos (100%) formará el cuello de botella que indicamos antes.

En el contexto sugerido, el flujo de trabajo se comienza a ralentizar en la unidad que pertenece al *Scripter*. Para solventar este problema, lo lógico, sería incrementar el número de miembros en la realización de dicha tarea, dividiendo el trabajo en diferentes trabajadores. Otra solución, sería acortar el proceso de producción de dicha unidad incrementando el número de horas que dicho

miembro dedica a esta tarea. No obstante, ambas soluciones son incorrectas si tratamos de eliminar este cuello de botella. Para explicar esto, hemos replicado el caso incorporando otro *Scripter* a nuestra tabla.

Resource	Resource Capacity					Process Capacity	Resource Utilization
	Unit Load (Days to make that field for one character)	Unit capacity (Characters/day)	Number of team members	Total characters per day			
Concept Artist	2.00	0.50	1.00	0.50	0.20	40.00%	
HighPoly	5.00	0.20	1.00	0.20		100.00%	
LowPoly	2.00	0.50	2.00	1.00		20.00%	
Rigger	2.00	0.50	1.00	0.50		40.00%	
Reaction Animator	3.00	0.33	1.00	0.33		60.00%	
Movement Animator	5.00	0.20	1.00	0.20		100.00%	
Combat Animator	7.00	0.14	3.00	0.43		46.67%	
Scripter	6.00	0.17	2.00	0.33		60.00%	
QA	3.00	0.33	1.00	0.33		60.00%	

**Tabla 2<sup>8</sup>:** Variación del caso hipotético de tabla de capacidad de recursos para el proceso de desarrollo de un personaje de videojuegos de Fischer. **Fuente:** Propia.

Podemos comprobar entonces que, en cierto modo, los cuellos de botella permanecerán siempre en el proyecto. Esto reside en la capacidad de proceso del proyecto, que recae en los miembros del equipo, no en su capacidad de trabajo, experiencia o tiempo que dedican al desarrollo, sino en la situación y propiedades de la unidad que procesan. Viendo la fórmula de esta manera, tan solo, como bien termina calculando Fischer, podremos optimizar el sistema de colas a medida que se forman los cuellos de botella. Little nos permite predecir cómo y en qué momentos esta utilización de recursos se verá afectada por el desarrollo natural de las unidades a través del sistema, así que la planificación del proceso en función de estos cuellos de botella es vital para diseñar un plan con el menor número de unidades en espera.

Así pues, donde esta investigación pone su punto de mira, es en la optimización de ese proceso de las unidades a través del sistema. Bien es cierto que hay multitud de posibilidades que afectan al modelo, todas estas pueden incluirse en el mismo, así que cuanto más previsión introduzcamos al sistema, mayor será la probabilidad de desarrollar un plan estable. En un sistema real

<sup>8</sup> Tabla de capacidad: esta tabla podrá ser replicada libremente a través del Anexo 4.



deberíamos introducir aquellas variables que alteran directamente el plan de producción como distinción de roles y experiencia profesional, inclusión de actividades y periodos festivos, perfiles multidisciplinares que realizan más de una unidad al mismo tiempo, etc.

La variable que nosotros introduciremos será la figura del TA en este sistema. Para ello, debe solventar aquellas necesidades que las unidades necesitan para acortar los procesos que generan los cuellos de botella. De esta forma podríamos solucionar el problema de las unidades en espera y tener una herramienta más a la hora de plantear nuestro *pipeline*, variando la capacidad de proceso de un sistema sin variar su cantidad de miembros o total de unidades de trabajo por unidad de tiempo.

Como ya hemos visto antes, los cuellos de botella seguirán formándose, pero una vez identificados, nuestro rol podrá resolverlos antes de que se formen sin tener que incrementar o disminuir la plantilla de miembros o incrementar las jornadas laborales. El planteamiento formado en esta investigación, pretende confirmar de forma acotada esta optimización del plan de trabajo en base a los experimentos planteados anteriormente y así, creando un entorno simulado en el que estas unidades “problemáticas” puedan ser solventadas de la forma más adecuada para el proyecto.

El diseño planteado para la investigación nos permitirá analizar el rol como una variable diferenciadora en el proceso de producción y desarrollo de un videojuego. Para ello, deberemos entonces analizar las fases en el orden que hemos propuesto anteriormente. Este análisis científico en etapas, se apoyará en los resultados obtenidos para finalmente poder afirmar si la incorporación del TA en la planificación y producción del videojuego, afectará sustancialmente en el desarrollo. Pudiendo comprobar nuestra hipótesis a través de la aplicación práctica de Little desarrollada por Fischer, introduciendo en el modelo nuestra variable diferenciadora.

Nuestro análisis contará con tres etapas. Las dos primeras, se estudiarán individualmente de forma aislada, para después conformar una visión conjunta de ambas que pueda aplicarse a la tercera.

### **1. Primera etapa.**

Representación actual del TA: estudiaremos su figura en base a las encuesta y entrevistas realizadas. Obtendremos la visión y definición global e individual del rol según la información recogida de diferentes perfiles basados en la industria del videojuego, desde profesionales a estudiantes y jugadores.

## **2. Segunda Etapa.**

Experimentación en base a la figura del TA en procesos de desarrollo: incluiremos los dos experimentos desarrollados anteriormente, recabando datos durante el proceso a la vez que parametrizamos las técnicas utilizadas por los sujetos. Tras la finalización del experimento, incluiremos una pequeña batería de preguntas dónde estudiar de forma cualitativa la respuesta hacia la incorporación de las nuevas técnicas que han utilizado en las tareas que han realizado.

## **3. Tercera Etapa.**

Incorporación del TA al modelo de desarrollo basado en sistemas de cola: finalmente volcaremos todo el análisis previo en el estudio de las posibilidades que permite la incorporación del rol en el equipo. Apoyaremos nuestras contribuciones en las anteriores etapas para formular una implementación sólida, pero abierta a la inclusión de nuevas variables en el sistema.

El objetivo de este análisis por etapas es, confrontar el estudio de la situación de aquella forma que la situación se da en un contexto real. Realizar una observación contextual del caso a través de las diferentes etapas para finalmente, habiendo aplicado un método inductivo, progresar o afianzar unas bases para el puesto en el marco de un proyecto.

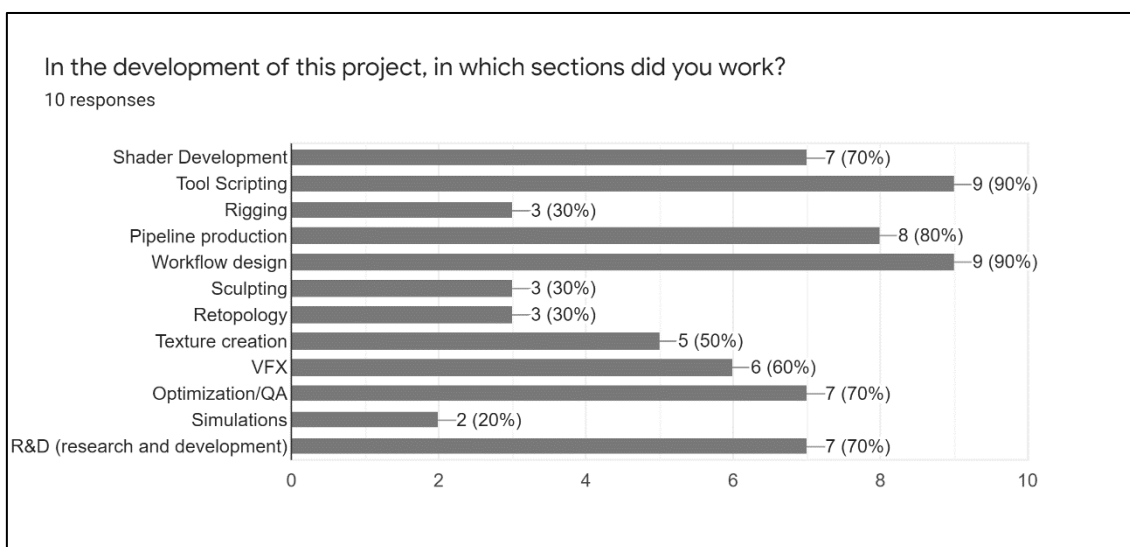
### 3. Análisis de resultados o implementación de contribuciones

#### 3.1. Representación actual del Technical Artist.

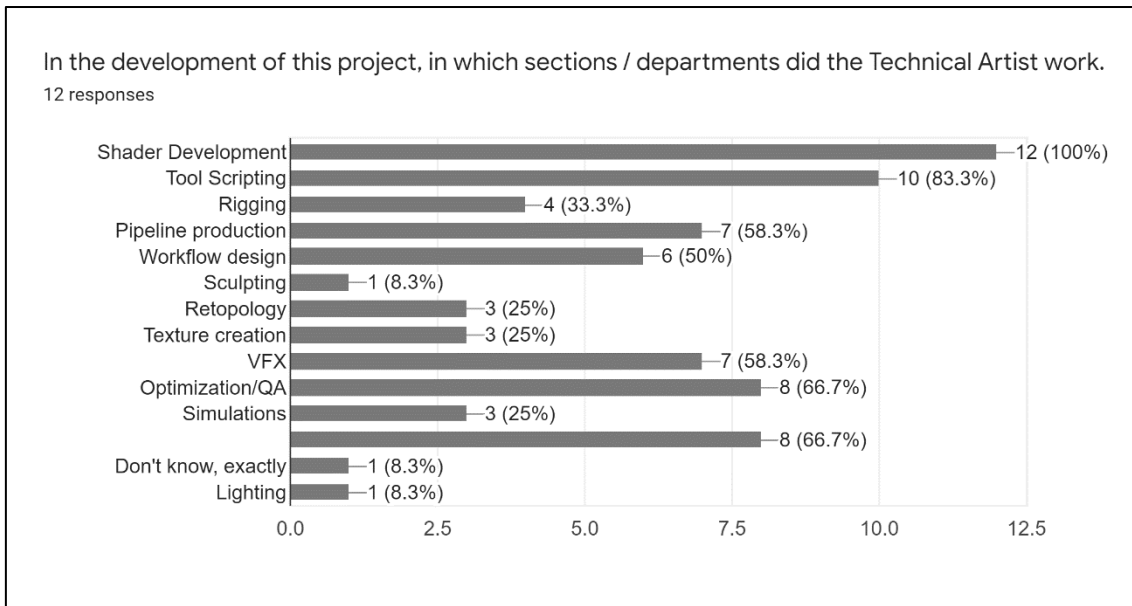
Con 54 participantes en nuestro estudio, hemos podido evaluar la percepción que actualmente se tiene sobre este rol. La inclusión de diferentes perfiles y roles profesionales ha permitido establecer diferentes posiciones pese a finalmente observar algunos rasgos comunes entorno a la disposición del cargo en el proyecto (Anexo 6).

En cuanto al proceso de desarrollo en el estudio, identificamos aquellas tareas en las que se ve más usualmente envuelto y preguntamos a los diferentes perfiles en qué medida participó el departamento de *Tech Art* en dicho proyecto. A su vez, dividimos las respuestas según su rol, es decir, TA u otro. Así podemos ver el resultado en dos tablas diferenciadas, unas desde la perspectiva del rol y la otra según la percepción del resto de miembros del equipo.

Inicialmente, se buscó objetivamente aquellas tareas que les habían sido asignadas al rol, pudiendo ver como estas tareas varían en pos de la procedencia del individuo, viéndose así la polivalencia del puesto respecto a la compañía en la que trabaje.



**Gráfico 1:** Secciones del desarrollo en las que el TA se vio involucrado. Según el *Tech Art*.  
**Fuente:** Propia.



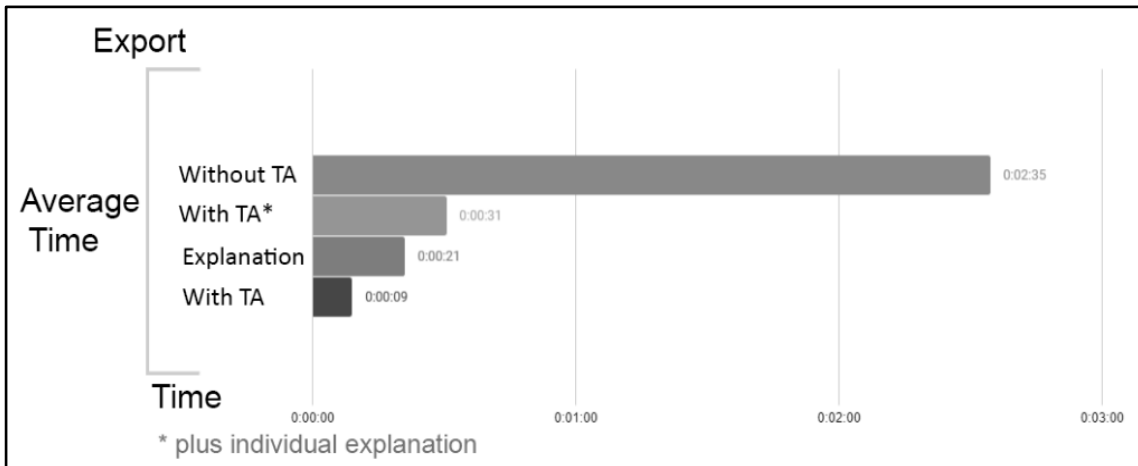
**Gráfico 2:** Secciones del desarrollo en las que el TA se vio involucrado. Según el resto de los miembros del equipo. **Fuente:** Propia.

Como se puede observar en los gráficos 1 y 2, tanto TAs como el resto de los perfiles, señalan diferentes tareas a la hora de atribuir un rol al puesto. En muchos casos, las actividades que se le atribuye, pese a formar parte de una disciplina como arte o programación, se alejan mucho entre sí. No obstante, la muestra tiende hacia *Shader Development*, *Tool Scripting* y optimización. Esta tendencia sin embargo, coincide y difiere en otras tareas en función de la compañía en la que trabajan o el conocimiento que tienen sobre la figura que estamos investigando, así como su relación con ella.

### 3.2. Experimentación en base a la figura del Technical Artist.

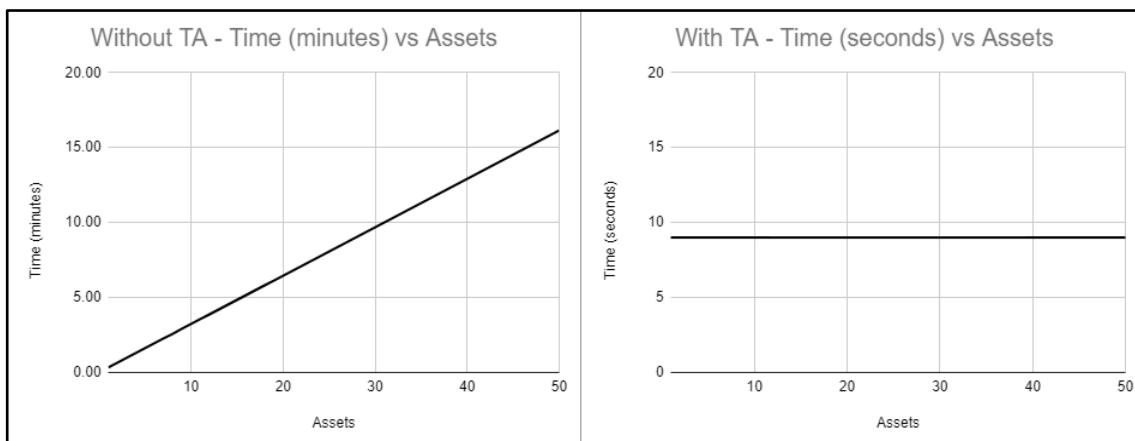
En cuanto a los experimentos desarrollados anteriormente, un total de 12 participantes realizaron individualmente dos pruebas en las que realizar dos tareas propias del desarrollo de un videojuego. Cada individuo realizó ambas pruebas seguidamente, para después compartir sus impresiones tanto de las pruebas como de su visión del rol.

En cuanto a la prueba “Exportación de objetos”, hemos podido observar que toda su fundamentación tiene un carácter exponencial, por lo que el nivel de optimización que obtendremos en el desarrollo, basándonos en los resultados obtenidos, irá en consecuencia al número de objetos que queramos exportar. En este caso, para agilizar el proceso, realizamos la prueba con un total de ocho objetos, por lo que si quisiéramos calcular el experimento en base a otro número de objetos, el coeficiente resultante con la incorporación del TA no variará, mientras que sin él, el tiempo de desarrollo aumentaría exponencialmente.



**Gráfico 3:** Tiempos promedios en el experimento de exportación de *assets*. **Fuente:** Propia.

El gráfico 3 nos muestra el tiempo promedio que un desarrollador consume en la exportación de tan solo ocho objetos. Por lo tanto, tardará 19.37 segundos en exportar un objeto, cifra que aumentará exponencialmente en función de la batería de objetos que vayamos exportar, a diferencia de la técnica presentada por el TA, que mantendrá el tiempo promedio de exportación sea cual sea la cantidad de *assets* que tengamos. Obtendremos así una herramienta que acelere el proceso de exportación en función de la batería de objetos que necesite el proyecto, acortando el proceso en función de la tediosidad de la tarea. Así pues, cuantos más *assets* tengamos, mayor será el impacto que la herramienta aporte al sistema.



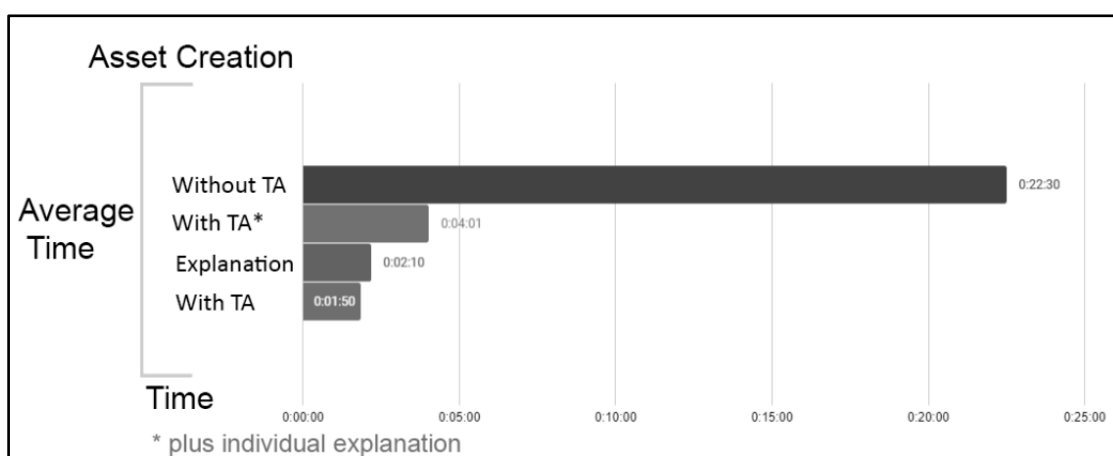
**Gráfico 4:** Variación en la exportación de objetos en función de la incorporación del TA. **Fuente:** Propia.

Con una técnica así, podremos asignar valores reales a procesos que serían difíciles de calcular en función de la variabilidad del proyecto, atribuyendo un tiempo estable al desarrollo de determinados *assets*. En este caso, representamos la varianza con una herramienta de exportación, pero a través del TA, o un *Tools Programmer* si se diera el caso, podríamos estabilizar procesos a periodos de tiempo fácilmente reconocibles para la producción de proyecto y asegurar el desarrollo de las tareas.

Esto nos lleva al siguiente experimento, la creación de *assets* específicos cuyo desarrollo puede suponer una importante carga de recursos en cuanto a tiempo y equipo. Estas tareas en muchos casos repetitivas, como la creación de objetos muy parecidos entre sí con características semejantes pero formas diferentes, pueden requerir una gran cantidad de horas en un proceso que podríamos solventar procedimentalmente.

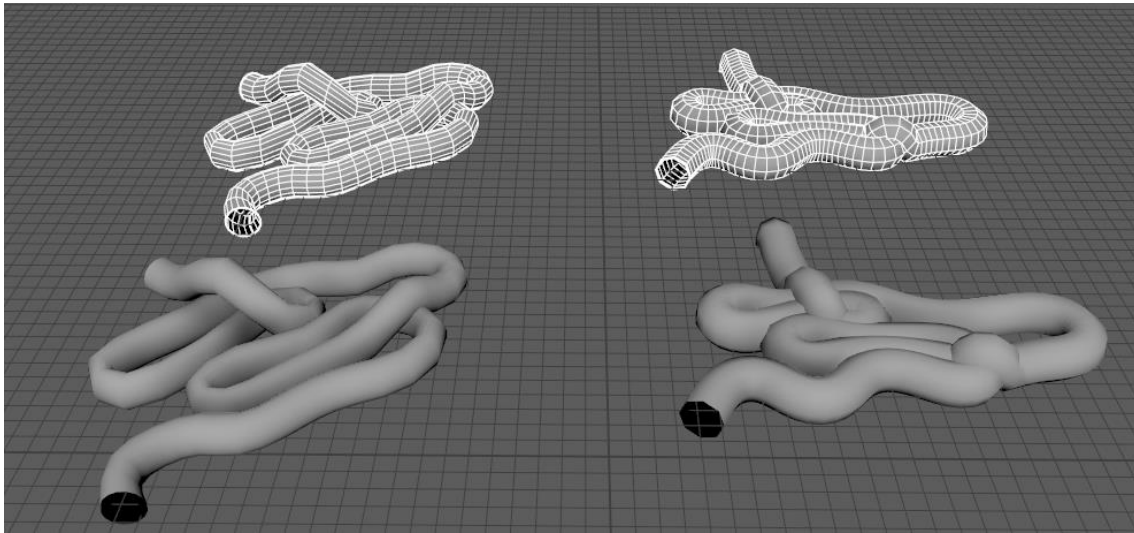
La extensión progresiva, por ejemplo, de los videojuegos de mundo abierto, impulsa a los desarrolladores a crear técnicas más eficaces a la hora de cubrir extensiones de terreno mayores, así como detallar espacios acordes a las expectativas técnicas y gráficas de los videojuegos actuales (Gordon, 2020). Esto desemboca en una mayor importancia en I+D+i, donde el equipo de *Tech Art* podrá proponer soluciones específicas a requerimientos propios del proyecto.

Hemos simulado esta situación con los participantes que se unieron al experimento, recreando un contexto en el que poder desarrollar determinados *assets*, al igual que en el experimento anterior, con y sin la incorporación del TA al sistema. Así podremos reflejar otra de sus aplicaciones en la producción del proyecto, incluyendo aquellas técnicas y herramientas que agilicen el proceso, al igual que en esta prueba.



**Gráfico 5:** Tiempos promedio en el experimento de creación de *assets*. **Fuente:** Propia.

El tiempo que un desarrollador utiliza en esta tarea, asciende a una media de 22 minutos 30 segundos. En este modelo, los participantes, contemplaron diferentes técnicas a la hora de crear el *asset*, por lo que el proceso varió en la técnica utilizada, pero como vemos la media de tiempo empleado sigue siendo muy superior pese a la elección de diferentes procesos para obtener el mismo resultado. Desde un proceso tradicional a base de extrusiones hasta NURBS<sup>9</sup> y creación de curvas, el proceso que se plantea desde el enfoque propuesto por el TA, sigue siendo más eficiente tanto en tiempo como en calidad del resultado final.



**Figura 9:** De izquierda a derecha. *Asset* realizado libremente según el participante y *asset* realizado a través de la técnica desarrollada por el TA. **Fuente:** Propia.

El tiempo empleado por el desarrollador mediante la nueva herramienta, gira en torno a una quinta parte del tiempo utilizado mediante las demás técnicas utilizadas. Además, contamos con una explicación de la herramienta que consume la mitad del proceso completo, por lo que en tal caso reduciría aún más el tiempo de desarrollo. No obstante, este estudio intenta representar lo más objetivamente el proceso, por lo que, pese a que en un contexto real, el TA podría elaborar una API interna para los desarrolladores y suprimir este tiempo adicional, queremos estudiar y matizar el carácter lectivo al que se somete el rol.

Esto último es especialmente importante tras organizar las impresiones de los participantes tras el experimento, ya que la muestra completa aseguraba haber aprendido durante la duración de la prueba. Tras la utilización de la técnica o herramienta, muchos de ellos destacaban la ayuda que nuestro rol supondría en un desarrollo a la hora de crecer profesionalmente, ya que, al igual que en el experimento, se plantean procesos que un perfil determinado no tendría en mente a la hora de comenzar cualquier desarrollo. Por lo tanto, volvemos a subrayar el papel que tiene el I+D+i que realiza para trasladar esas nuevas

<sup>9</sup> NURBS: (Non-uniform rational basis spline) modelo matemático para representar curvas y superficies.

técnicas y conocimientos a las respectivas unidades del sistema, en definitiva, el resto del equipo.

Los resultados obtenidos reflejan, en ambos experimentos, un mayor control en el desarrollo natural del videojuego. No solo agilizar sino aliviar la producción a nivel de cada uno de los perfiles, teniendo en cuenta los diferentes apartados y adversidades de la tarea que realiza cada desarrollador, proporciona en muchos casos un ambiente de calidad, mejora y aprendizaje, al igual que hemos podido contemplar de forma simulada en los experimentos.

### **3.3. Incorporación del Technical Artist al modelo de desarrollo basado en sistemas de cola.**

Tras el planteamiento y estudio de la figura del TA en la producción y desarrollo de un videojuego, podemos abordar el impacto que tiene el rol sobre el sistema a partir del planteamiento que hemos propuesto antes según la *Ley de Little*. Para ello tan solo tenemos que analizar el proyecto de la misma forma que analizamos un sistema de cola, aplicando el *pipeline* que mejor se adapte al proceso que debemos realizar en base la posición y contexto del producto que tenemos en mente.

La forma de integrar al rol, puede ser tanto a través de una unidad más como de una variable del sistema (videojuego) que engloba a los diferentes subsistemas (departamentos). De una u otra forma podremos añadir previamente el trabajo del puesto, como figura que estructura las unidades posteriores al mismo de forma secuencial o global. La elección de cómo integrar la variable, puede venir dada en función del tamaño del equipo así como la existencia o no de un departamento de *Tech Art* que se divida en disciplinas que abarquen diferentes procesos (Hayes, 2008).

Por tanto, el rol que asume en la producción a largo plazo de un proyecto, es equivalente al de una herramienta de optimización del propio sistema. Como veíamos anteriormente, para liberar un cuello de botella, podemos recurrir a la variación de los recursos que atribuimos a las diferentes unidades, es decir, variar la cantidad de trabajadores o tiempo que dedican a dicha tarea. Así, disolvemos el cuello de botella con antelación a que se forme o retenga demasiadas unidades en una sección del sistema, en definitiva, optimizando el sistema.

Esta posición, por tanto, podrá desde la producción y planteamiento del proyecto, incorporar aquellas soluciones que mitiguen la retención de unidades en el sistema, así como corregir las desviaciones que sufra el *pipeline* debido a problemas o imprevistos a lo largo del proceso. Por lo que, uniendo esto a la corrección de cuellos de botella en base a la moderación de recursos que acabamos de exponer, tenemos un plan de producción mucho más sólido gracias, irónicamente, a la flexibilidad que aporta la variable del TA.



## 4. Conclusiones

Como hemos observado a lo largo del estudio, la figura del Technical Artist abarca un amplio abanico de tareas y procesos en el desarrollo de un videojuego, provocando en efecto, que no haya una definición propia del rol. Podemos ver en los anexos de este trabajo y junto a los resultados obtenidos, que muchas de las definiciones y características aportadas señalan la multidisciplinariedad del rol como propiedad fundamental de esta posición.

Esta naturaleza por tanto, impone como ya hemos señalado un papel transversal al proyecto, siendo la producción una de las secciones donde su papel exprime por completo su potencial. La eficiencia y agilidad en un rol cuya participación en el videojuego debería hibridar desarrollo y producción para afinar al máximo las necesidades del equipo y el programa tanto técnica como administrativamente.

Y es que, además de la recopilación de datos que nos ha proporcionado la fase de encuesta, los experimentos dejan detalladamente claro el impulso que supone la incorporación del rol tanto para el estudio como para el videojuego, haciendo que los procesos de desarrollo que antes llevarían horas, ahora podrían ser cuestión de minutos o incluso segundos.

Será entonces, el encargado de desarrollar e investigar esas técnicas y herramientas que ya hemos visto, e introducirlas en el plan de producción del proyecto, sabiendo de ante mano como facilitar al equipo de desarrollo aquellas tareas que implican mayor frustración, tediosidad o franjas de tiempo muy superiores al resto de procesos. Seguirá por tanto siendo un rol que suponga tener una visión global del proceso real que implica desarrollar y producir un videojuego, para poner esos conocimientos en práctica a la hora de elaborar un plan con el que el proyecto consiga una ventaja sustancial en el desarrollo de este.

De esta forma podríamos conseguir tratar el rol desde un nuevo punto de vista probado, dando paso ahora a incluir este modelo en diferentes equipos para ver el comportamiento de este método en diferentes producciones. Introducir nuevas variables y conceptos así como nuevas técnicas orientadas a plantear la producción, serán fundamentales para optimizar la inclusión del puesto en el proyecto, bajo la comprensión de este como un sistema de subsistemas de colas.

Hemos realizado una concreción a lo largo de este trabajo de investigación donde concretamos aquello donde el Technical Artist sería más eficaz, no obstante, esto mismo sirve como punto de partida a la hora de elaborar futuras investigaciones con los diferentes roles que componen el equipo de una compañía. Esta misma hipótesis podría ser planteada en departamentos y trabajos concretos, aproximándonos a una mayor optimización en la resolución

de procesos para la producción de un videojuego, mediante el estudio sistemático de las diferentes labores que componen el desarrollo.

Estudiar pormenorizadamente aquellas tareas de arte, programación o diseño, será el siguiente paso a la hora de poder optimizar cada tarea del sistema y hacer que los demás miembros que componen el equipo puedan desarrollar al máximo su potencial, introduciendo nuevas variables optimizadoras al videojuego y afinando cada vez más el flujo de trabajo. Ya que, así como hemos demostrado el enorme impacto que genera este rol sobre la producción, podríamos comenzar nuevas investigaciones caracterizadas por otros nuevos roles que podrían dar paso a mejores técnicas y herramientas que impulsen la calidad del producto final, así como las condiciones laborales y el desarrollo profesional y personal del equipo.

## 5. Bibliografía

- Ambrosiussen, P. (2018) How to develop effective Software Tools. Recuperado de: <https://80.lv/articles/how-to-develop-effective-software-tools/>
- Dunning, B. y Bossom, A. (2015). Video Games: An Introduction to the Industry. University for the Creative Arts, Farnham, UK. Fairchild Books.
- Calle, D., Neufeld, E. y Schneider, K. (2005). Requirements Engineering and the Creative Process in the Video Game Industry. Department of Computer Science, University of Saskatchewan. Recuperado de: <http://experiencefirstdesign.com/wp/wp-content/uploads/2012/12/01531045.pdf>
- De Boer, R. (2017) Real-time Photorealism. Recuperado de: <https://origin.80.lv/articles/real-time-photorealism/>
- Fischer, J. (2016) Game Planning with Science. Breaking the Wheel. Recuperado de: <https://www.breakingthewheel.com/video-game-art-pipelines/>
- García-Obledo, F. (2017) VFX for Games Explained. Recuperado de: <https://80.lv/articles/vfx-for-games-explained/>
- Godoy, A. y Barbosa, E. (2010) Game-Scrum: An Approach to Agile Game Development. Instituto de Ciências Matemáticas e de Computação. Recuperado de: [http://www.sbgames.org/sbgames2010/proceedings/computing/short/Computing\\_short19.pdf](http://www.sbgames.org/sbgames2010/proceedings/computing/short/Computing_short19.pdf)
- Gordon, L. (2020) Beyond GTA: How Tech and Communities Could Change the Future of Open-World Games. From user-generated content to indie stories about loneliness. Recuperado de: <https://www.theverge.com/2020/2/25/21142766/open-world-games-future-tech-gta-zelda-red-dead-cyberpunk-sable>
- Hayes, J. (2008) The Code/Art Divide: How Technical Artists Bridge the Gap. Recuperado de: [https://www.gamasutra.com/view/feature/1651/the\\_codeart\\_divide\\_how\\_technical\\_.php?print=1](https://www.gamasutra.com/view/feature/1651/the_codeart_divide_how_technical_.php?print=1)
- Jam City. (2019) Beyond game design - 5 absolutely vital jobs in game development. Recuperado de: [https://www.gamasutra.com/view/news/354994/Sponsored\\_Beyond\\_game\\_design\\_5\\_absolutely\\_vital\\_jobs\\_in\\_game\\_development.php](https://www.gamasutra.com/view/news/354994/Sponsored_Beyond_game_design_5_absolutely_vital_jobs_in_game_development.php)

- Kanode, C. y Haddad, H. (2009) Software Engineering Challenges in Game Development. Computer Science Department, Kennesaw State University. Recuperado de: [https://shop.tarjomeplus.com/UploadFileEn/TPLUS\\_EN\\_4801.pdf](https://shop.tarjomeplus.com/UploadFileEn/TPLUS_EN_4801.pdf)
- Keith, C. (2010) Agile Game Development with Scrum.
- Leppänen, M. y Koutonen, J. (2013) How are Agile Methods and Practices Deployed in Video Game Development? A Survey into Finnish Game Studios. Department of Computer Science and Information Systems, University of Jyväskylä, Finland. Recuperado de: [https://www.researchgate.net/profile/Mauri\\_Leppanen2/publication/255823539\\_How\\_Are\\_Agile\\_Methods\\_and\\_Practices\\_Deployed\\_in\\_Video\\_Game\\_Development\\_A\\_Survey\\_into\\_Finnish\\_Game\\_Studios/links/55c866c008aebc967df89ebb.pdf](https://www.researchgate.net/profile/Mauri_Leppanen2/publication/255823539_How_Are_Agile_Methods_and_Practices_Deployed_in_Video_Game_Development_A_Survey_into_Finnish_Game_Studios/links/55c866c008aebc967df89ebb.pdf)
- Liming, D. y Vilorio, D. (2011) Work for play: Careers in video game development. Game Developer Magazine, San Francisco. Recuperado de: <https://files.eric.ed.gov/fulltext/EJ945968.pdf>
- Maximov, A. (2017) 4 Technologies to Change Art Production. Recuperado de: <https://origin.80.lv/articles/4-technologies-to-change-art-production/>
- Murphy-Hill, E., Zimmermann, T. y Nagappan, N. (2014). Cowboys, Ankle Sprains, and Keepers of Quality: How Is Video Game Development Different from Software Development? North Carolina State University. Recuperado de: <http://cdn.cocimq.com/cms/uploads/soft/140508/8370-14050Q00254.pdf>
- Musil, A., Musil, J., Winkler, D. y Biffi, S. (2010) Improving Video Game Development: Facilitating Heterogeneous Team Collaboration through Flexible Software Processes. Software Engineering Integration for Flexible Automation Systems, Institute of Software Technology and Interactive Systems. Recuperado de: [https://s3.amazonaws.com/academia.edu.documents/51379992/Improving\\_Video\\_Game\\_Development\\_Facilit20170116-27160-v01i9n.pdf?response-content-disposition=inline%3B%20filename%3DImproving\\_Video\\_Game\\_Development\\_Facilit.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20200122%2Fus-east-1%2Fs3%2Faws4\\_request&X-Amz-Date=20200122T193024Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-](https://s3.amazonaws.com/academia.edu.documents/51379992/Improving_Video_Game_Development_Facilit20170116-27160-v01i9n.pdf?response-content-disposition=inline%3B%20filename%3DImproving_Video_Game_Development_Facilit.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20200122%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200122T193024Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-)

Signature=c0066731409faaf5a0400c6adbfc731b6bc7eb0c9c69a8f27be1b3be03780eb7

- Petrillo, F. y Pimenta, M. (2010) Is Agility out there? Agile Practices in Game Development. Institute of Informatics - Federal University of Rio, Brazil. Recuperado de: [https://www.researchgate.net/publication/220961952\\_Is\\_agility\\_out\\_there\\_Agile\\_practices\\_in\\_game\\_development](https://www.researchgate.net/publication/220961952_Is_agility_out_there_Agile_practices_in_game_development)
- Stafford, L. (2018) The Art of Technical Production in Games. Recuperado de: <https://80.lv/articles/the-art-of-technical-production-in-games/>
- Stolecki, P. (2018) Being a Technical Artist in An Emerging Studio. Recuperado de: <https://80.lv/articles/being-a-technical-artist-in-an-emerging-studio/>
- Thoerzen, M. (2015) The Future of Game Development Technology. Recuperado de: <https://80.lv/articles/the-future-of-game-development-technology/>
- Zackariasson, P., Walfisz, M. y L. Wilson, T. (2006) Management of Creativity in Video Game Development. Umeå School of Business and Economics, Umeå University, Sweden. Recuperado de: [https://www.researchgate.net/publication/252442746\\_Management\\_of\\_Creativity\\_in\\_Video\\_Game\\_Development](https://www.researchgate.net/publication/252442746_Management_of_Creativity_in_Video_Game_Development)

## **6. Ludografía**

Epic Games (2017). Fortnite. Epic Games. PC.

## 7. Anexos

### Anexo 1: Recopilación de descripciones del rol del TA

#### Pregunta:

Para ayudar al proceso de investigación de este trabajo y la recopilación de datos y descripciones sobre la figura del Technical Artist, nos sería extremadamente útil que compartieses una descripción del rol según tu opinión y experiencia.

#### Respuestas:

"I think that the way that the T A works influent directly on the final expected work. Even the whole game will be personal for this T A, because is himself the one that put a "face" to all the lines of code behind that game. Is grateful see your art having live on that video game."

"Optimizes the workflow and shows how things should be done."

"Tech Art seems to change a lot per studio, and per project needs. I think it's very important to be adaptable, and to adopt a wide skill set."

"A bridge between both Programming and Art departments. It involves different tasks, depending on the project requirements, but mostly they can be summarized in Research & Development, Workflow/Pipeline design, Tools & Shader development and Optimization/QA"

"I have worked as a technical artist at a number of companies and on many different games. The role can vary widely from project to project and based on the particular focus of the individual technical artist. I suppose the essence of the technical artist is to facilitate the production of the game, to provide a service to the various departments to help them achieve their goals, whether that be through a great rig, a tool to automate a repetitive process, a unique visual element to satisfy a design requirement or an optimization so the game can hit its framerate target. We provide solutions to the technical challenges of game development."

"Inter-disciplinary problem solving for content creation."

"Optimisation of the work pipeline"

"The Technical Artist need to be on vanguard of the technology to get the more efficiency on the product results."

“It can mean several things, depending on pipeline, size of team, etc. But to me, it is the artist that is able to grab the concepts and ideas and see that they end up being implemented efficiently into the game while staying true to the original vision. Though often, they work more with the "technical" part of production, such as R&D, textures, simulations, etc. They are a critical role in the production and the designers need to be working with them constantly in order for a high quality and efficient product to be created.”

“A Technical artist bridges gap between art and code, they help to make the lives of both artists and programmers easier and they must therefore understand both roles. A tech artist must also be able to communicate effectively with both roles. A tech artist will help to solve any problems that occur within the development cycle and help to speed up repetitive tasks to bring out the best in the team. A tech artist will also help with R&D to discover new workflows and identify weaknesses in current pipelines. A tech artist is a very important member of the team and a great tech artist can accelerate the whole team production speed.”

“Technical Art is about streamlining the asset production pipeline, from asset creation, to import/export, evaluation and sharing.”

“A tech artist does research on technical aspects of the graphical side of game development and supports the artists in learning new tools, building new pipelines or doing technical work like writing shaders.”

“It's the one who makes sure that the pipeline goes as smooth as possible organizing and helping the art department.”

“A technical artist fills a very supportive function for different art & dev teams inside a company. Technical artists create tools to speed-up the workflow, but also assist in production itself by creating assets (often to add to a internal library of assets) and by helping with optimization. Technical artists in my opinion are the bridge between artists and programmers as they understand both worlds.”

“Bridge between art and programming, often automatization of art processes in games.”

“Assuming that the role of technical artist is, as its definition says, a hybrid between art and programming. But to be more concise, I see him as the one that contributes the spark to art, be it through textures that only with images are not achieved, be it through a code that perhaps a programmer does not think of but being someone who belongs to the field of the arts has a better vision. I see it as

a very important role since it can be the bridge of communication between programmers and artists.”

“It is a bridge between Artists and the rest of the team. It is a role that knows from both sides and defines what it is possible or not to improve the quality of your games. Artists will ask you if something they want is possible to reproduce it in your engine, and the programming team will help you providing resources and tools to make that task possible. You will have to know a little bit about every process that involves making a game in both sides, programming and art, to complete the desires of artists with the efficiency and optimizations that programmers require. Shaders, lighting, mesh optimization, tools, pipeline and workflow definitions, world building... you will be in the middle of two groups that create lot of content that needs to be cohesive by technical artists.”

“While a seemingly forgotten role, I believe technical artists have a huge responsibility when it comes to 3D details and the overall polish of environments and materials. It looks very tedious, but the results are amazing!”

“I do lots of different Things as a Tech Artist, I do research for different areas of the game, Like efficient Workflows and I build Pipelines for artists to work with, i also do sculpting and modelling from time to time, I’ve seen that Tech artists are pretty much Generalists in many cases, which also applies to me, I do a bit of everything.”

“In my opinion, a tech artist is one that has skills in art and programming and can implement shaders, make code/tools to change visuals for designers to use, and optimize the previously mentioned for better performance.”

“Se podría decir que son los encargados de realizar el apartado de los gráficos del juego supongo.”

“Es un rol entre el de un artista y un programador cuya función es la de crear técnicas que sirvan para facilitar o reducir el trabajo al resto de trabajadores y/o mejorar la calidad de estos.”

“Se encarga de programar el propio juego y crearlo desde cero.”

“Para mí el Technical Artist hace un poco de todo y es una forma de que todos se adapten a un tipo de trabajo determinado. Todos aprenden a desarrollar el juego de la misma manera y si hay algún problema es el que lo soluciona.”



## Anexo 2: Max Script desarrollado para la realización del experimento de exportación.

```
FbxExporterSetParam "lights" false
FbxExporterSetParam "cameras" false
FbxExporterSetParam "UpAxis" "Z"

if selection.count != 0 then
(
    mySelection = selection as array
    deselect mySelection
    for obj in MySelection do

        (
            select obj

            if obj.isanimated == true then
            (
                FbxExporterSetParam    "Animation" true
            )
            else
            (
                FbxExporterSetParam    "Animation" false
            )

            savePath = maxfilepath
            pathName = savePath + obj.name + ".fbx"
            exportFile pathName #noPrompt selectedOnly: True
using:FBXEXP

        )
    )
else
(
    messagebox "No Selection"
)
```

- Para reproducir el experimento (una vez que hayamos seleccionado los objetos que deseamos exportar) tan solo es necesario copiar este código en un archivo “.txt” y arrastrarlo a la ventana “*viewport*” de 3DS Max.

### Anexo 3: Tablas de resultados en experimentos.

#### Experimento 1: Exportación de assets en 3DS Max.

Export	Tiempo base (sin TA)	Tiempo con TA (+Explicación)	Explicación	Tiempo con TA (sin explicación)
	0:02:43	0:00:30	0:00:21	0:00:09
	0:02:21	0:00:29	0:00:21	0:00:08
	0:02:11	0:00:34	0:00:21	0:00:13
	0:02:12	0:00:30	0:00:23	0:00:07
	0:02:31	0:00:26	0:00:19	0:00:07
	0:02:16	0:00:23	0:00:20	0:00:03
	0:02:43	0:00:30	0:00:21	0:00:09
	0:03:02	0:00:36	0:00:25	0:00:11
	0:02:27	0:00:31	0:00:22	0:00:09
	0:02:38	0:00:33	0:00:18	0:00:15
	0:02:41	0:00:35	0:00:20	0:00:15
	0:02:51	0:00:36	0:00:28	0:00:08
	Tiempo normal	Tiempo con TA (+Explicación)	Explicación	Tiempo con TA (sin explicación)
Media	0:02:35	0:00:31	0:00:21	0:00:09

## Experimento 2: Creación de assets en Maya.

Asset Creation	Tiempo base (sin TA)	Tiempo con TA (+Explicación)	Explicación	Tiempo con TA (sin explicación)
	0:18:23	0:04:02	0:02:11	0:01:51
	0:34:58	0:04:42	0:02:28	0:02:14
	0:28:36	0:04:26	0:02:31	0:01:55
	0:21:14	0:04:15	0:02:19	0:01:56
	0:26:42	0:03:28	0:02:05	0:01:23
	0:22:56	0:02:58	0:02:01	0:00:57
	0:26:31	0:03:37	0:02:11	0:01:26
	0:15:56	0:03:25	0:02:05	0:01:20
	0:23:18	0:03:47	0:01:58	0:01:49
	0:22:03	0:04:32	0:02:09	0:02:23
	0:19:57	0:03:59	0:02:15	0:01:44
	0:19:09	0:04:03	0:02:08	0:01:55
	Tiempo normal	Tiempo con TA (+Explicación)	Explicación	Tiempo con TA (sin explicación)
Media	0:22:30	0:04:01	0:02:10	0:01:50

#### **Anexo 4: Tabla de capacidad de recursos en un caso hipotético pipeline de personales en el desarrollo de un videojuego.**

A través del siguiente enlace se podrá acceder a una copia de la tabla con la que realizar propuestas propias de desarrollo basado en sistemas de colas, descargando el archivo y aplicándolo a cualquier *pipeline*:

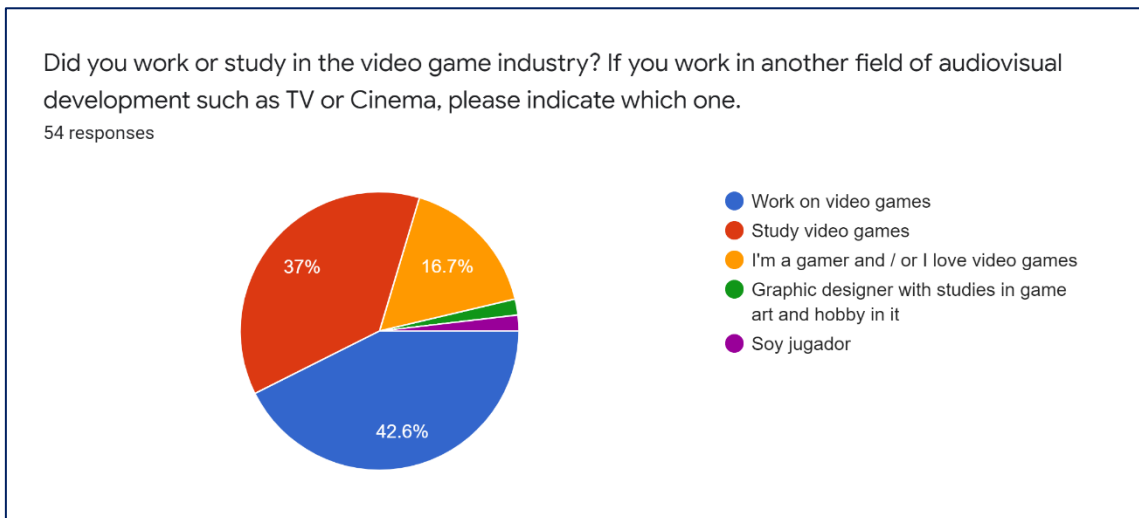
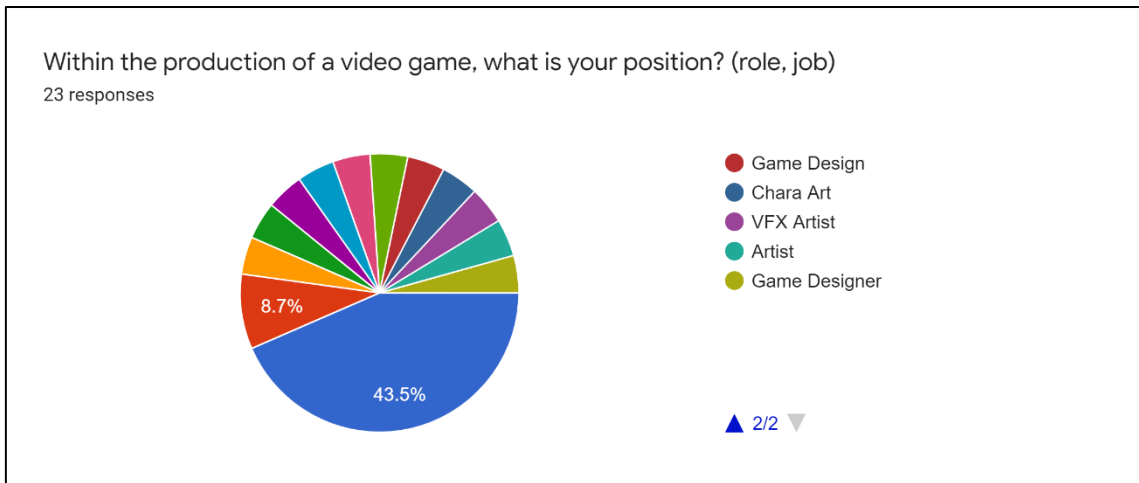
<https://docs.google.com/spreadsheets/d/1B87g1EI5OTEktLMJIAhHZHpUH60VjGVj8gX1XnUXyjs/edit?usp=sharing>

#### **Anexo 5: Técnica de creación de assets.**

A través del siguiente enlace se podrá acceder a una copia de la escena que se propuso a los participantes en el segundo experimento:

<https://drive.google.com/file/d/1tVFwpRz4fB-kP6ReeVTo8pGEN51v-fKn/view?usp=sharing>

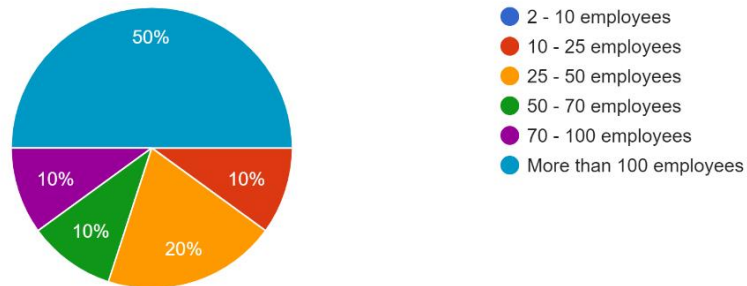
## Anexo 6: Gráficas resultado de la encuesta.



## Section: Im a Technical Artist

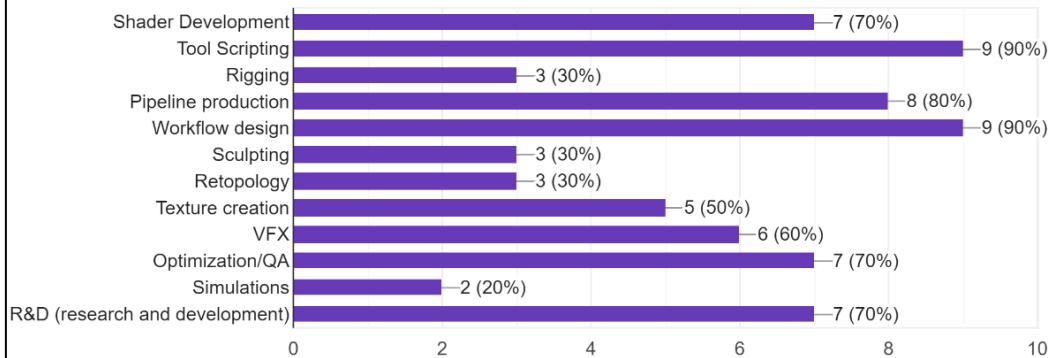
How many employees did (approximately) the company you worked for have?

10 responses

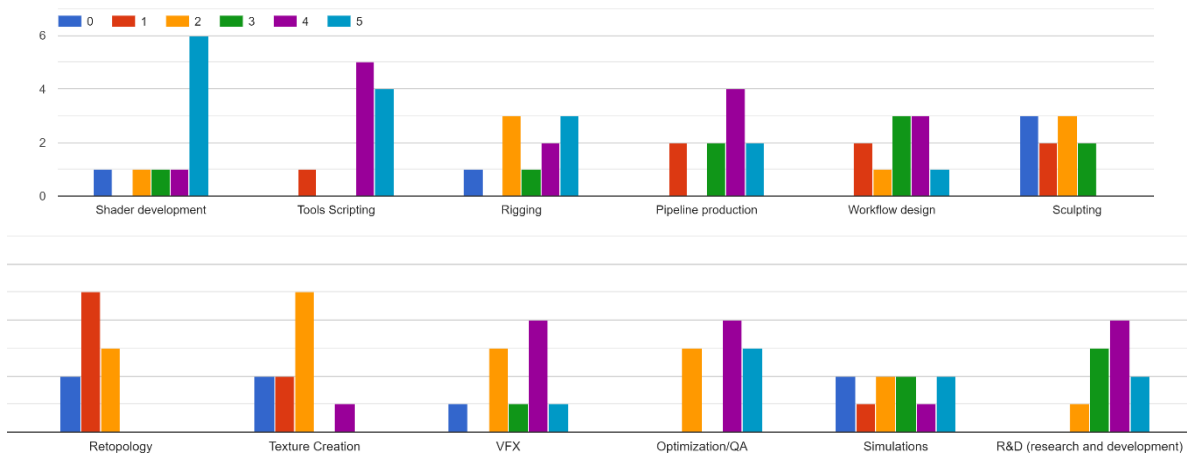


In the development of this project, in which sections did you work?

10 responses

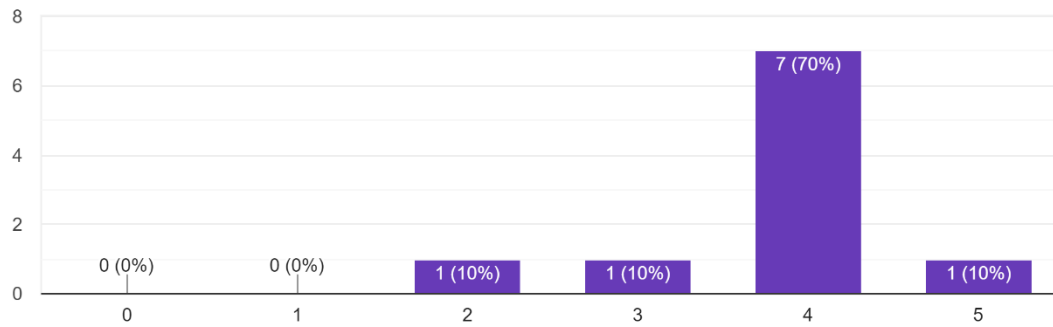


Evaluate from 0 to 5 how much the Technical Artist contributes to these sections.



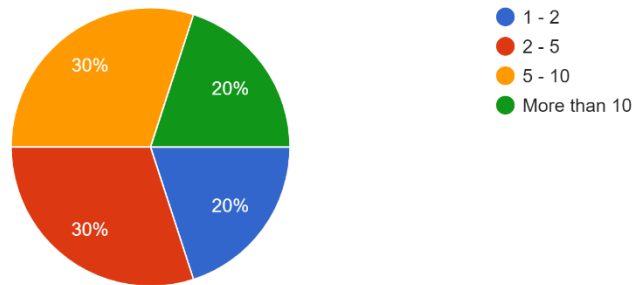
From 0 to 5 globally evaluate your involvement creating a work plan for the project (production, pipeline, workflow)

10 responses



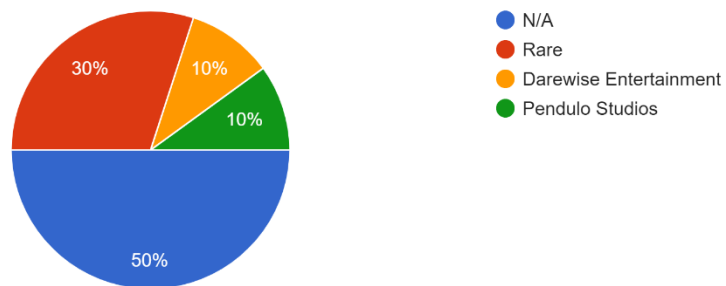
Indicate how many years you have been working on video games professionally.

10 responses

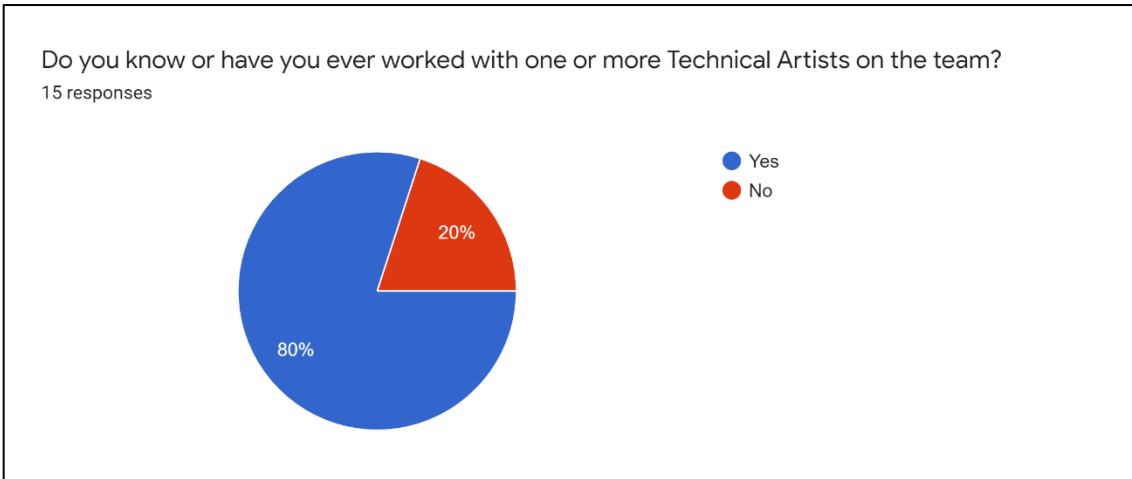


What is the most recent company you have worked for? If you do not agree to disclose this information, please indicate with N / A.

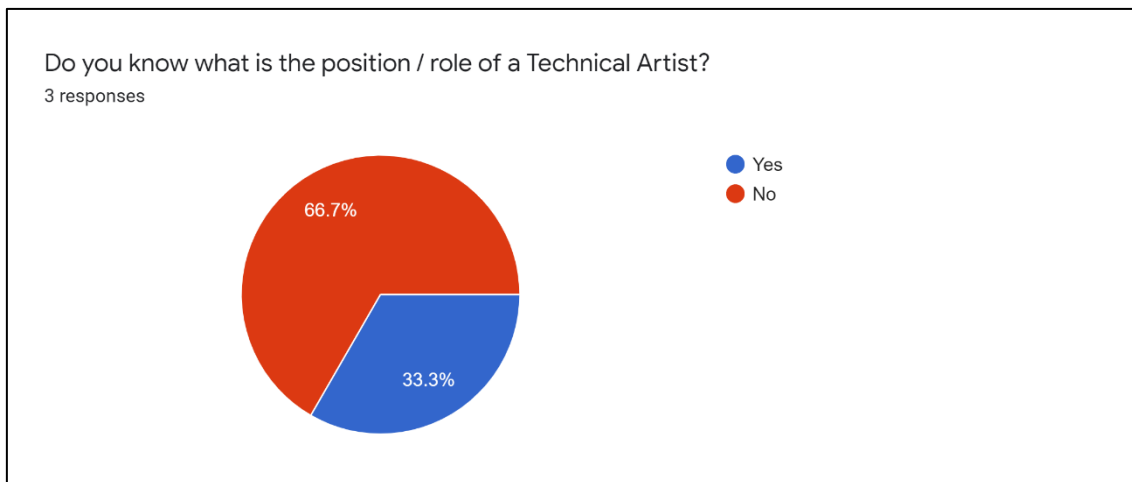
10 responses



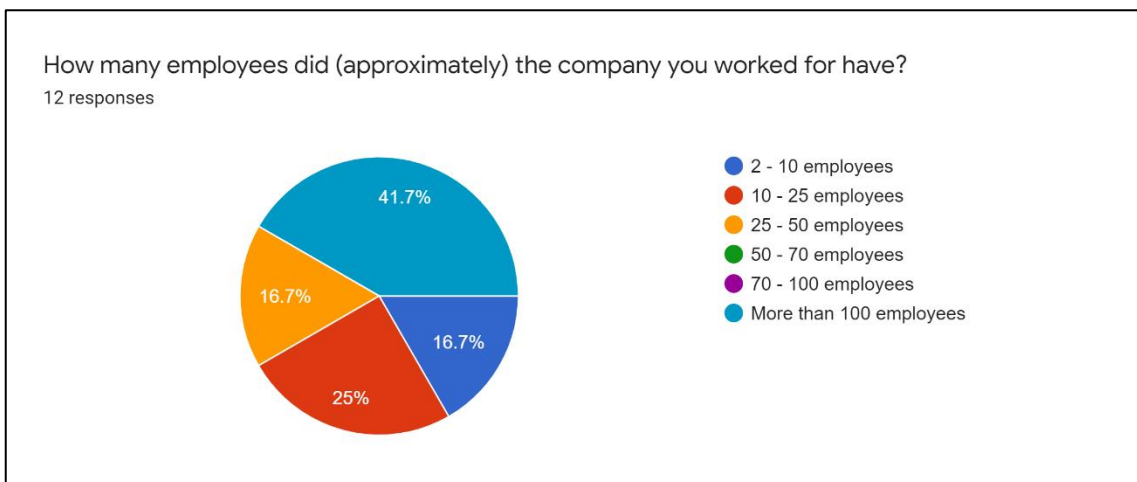
### Section: I work in video games or another audio-visual field



### Section: I have not worked with a TA



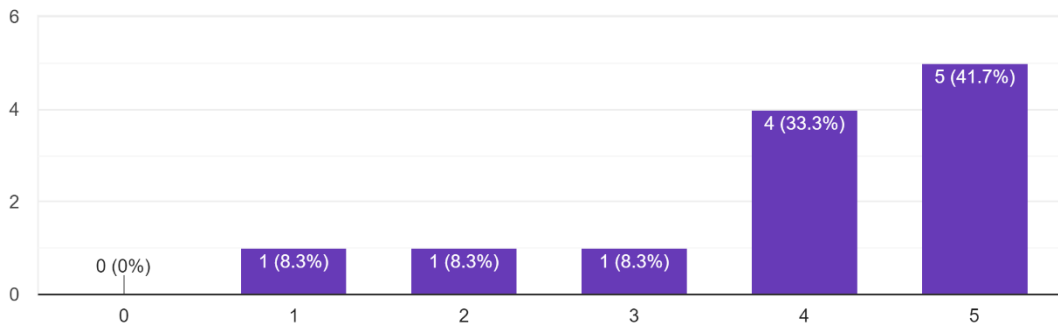
### Section: I have worked with a TA





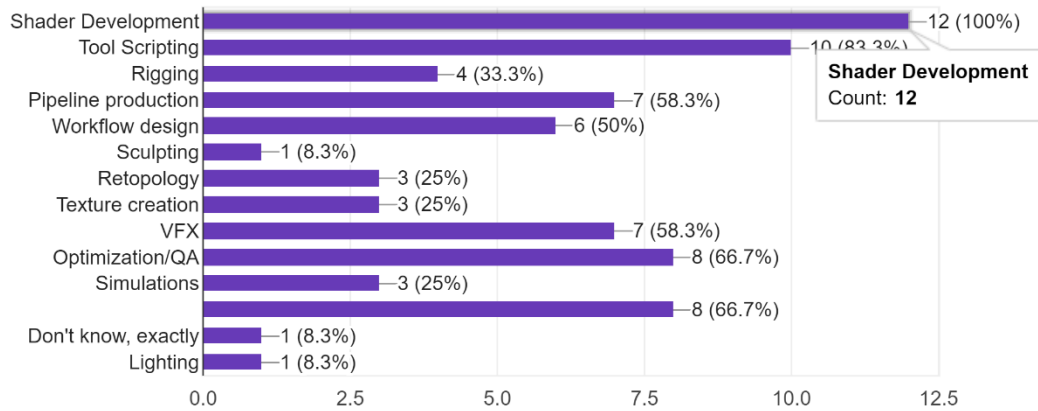
In what grade did you have relation with the Technical Artist?

12 responses

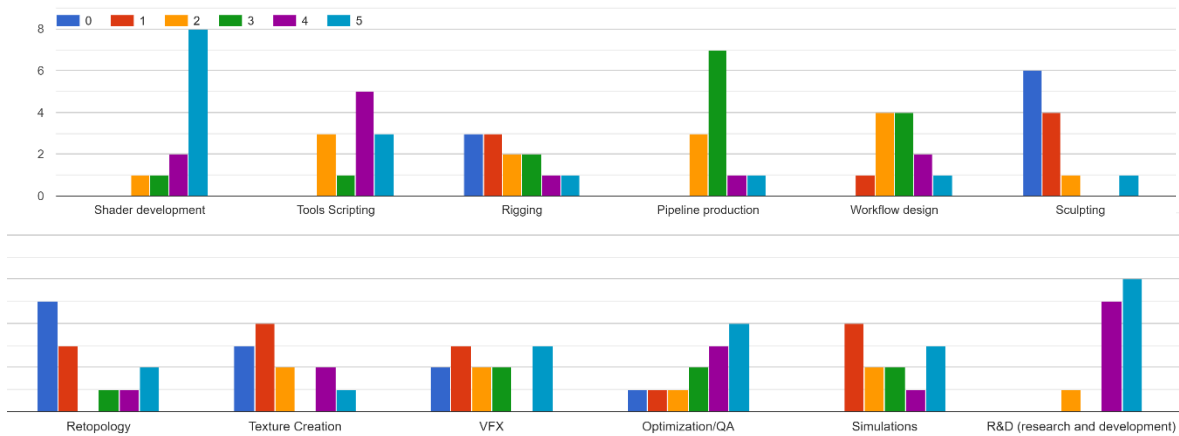


In the development of this project, in which sections / departments did the Technical Artist work.

12 responses

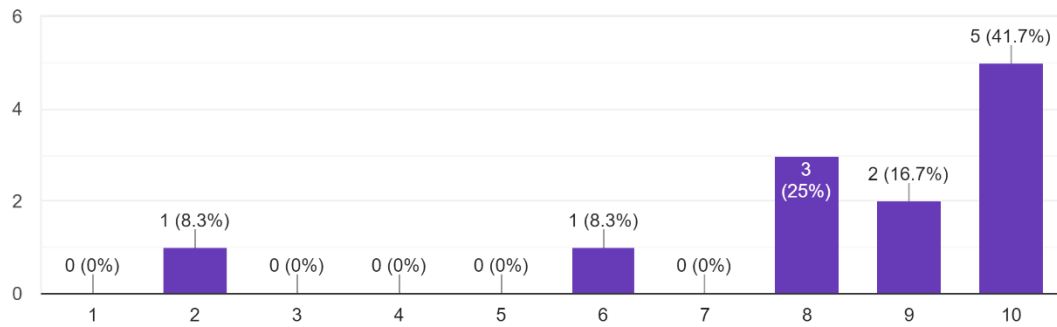


Evaluate from 0 to 5 what the work of the Technical Artist should be in these sections.



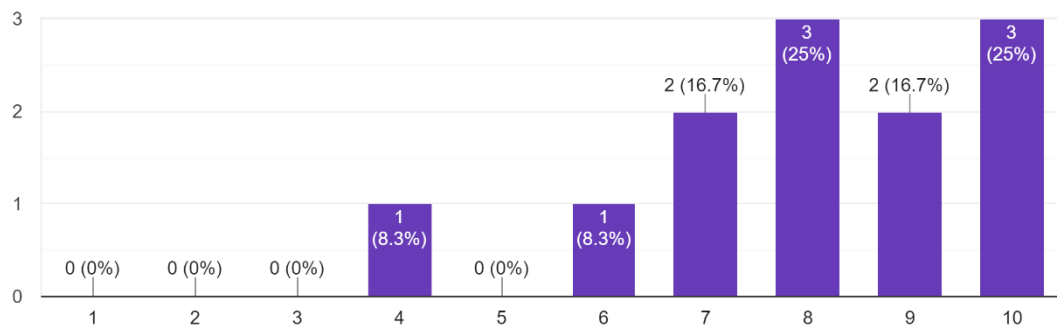
From 1 to 10 evaluate globally how involved the Technical Artist should be in the production plan of the project.

12 responses



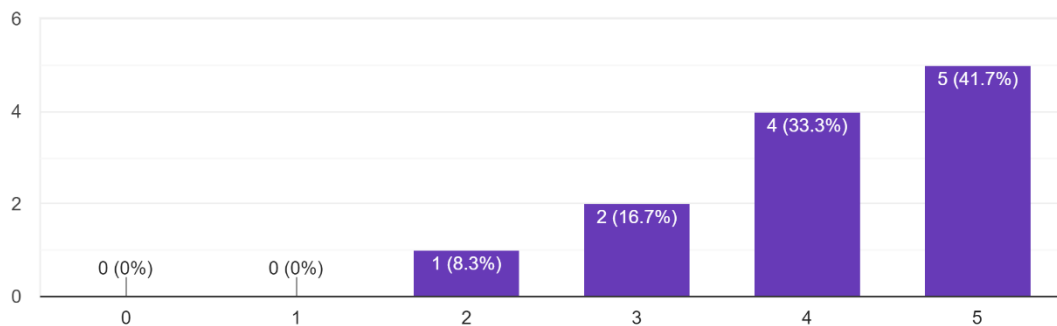
From 1 to 10, evaluate how the figure of the Technical Artist influenced the development of your individual work in the team.

12 responses



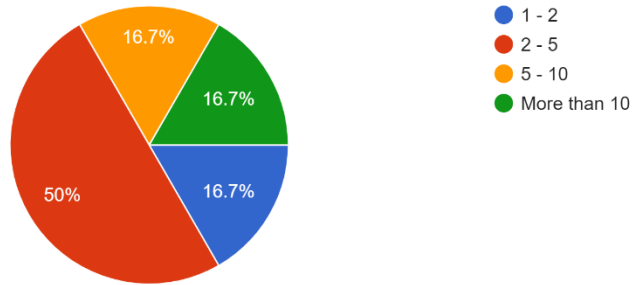
Did the Technical Artist influence your learning? (New techniques, optimization, new resources, new possibilities)

12 responses



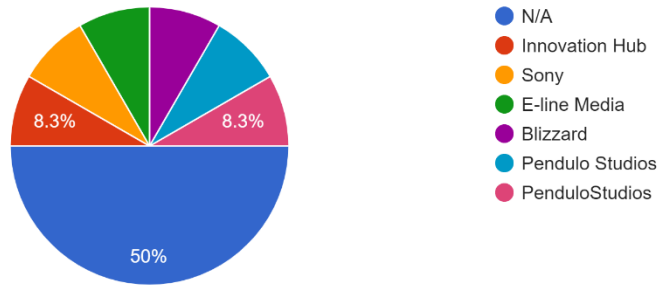
Indicate how many years you have been working on video games professionally.

12 responses



What is the most recent company you have worked for? If you do not agree to disclose this information, please indicate with N/A.

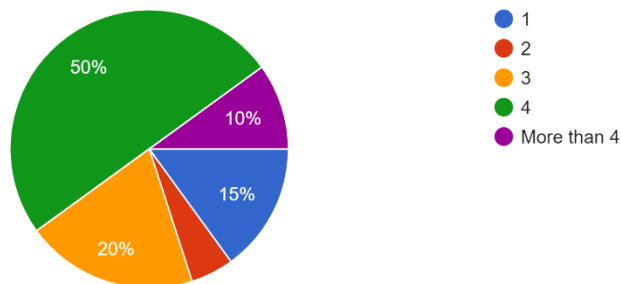
12 responses

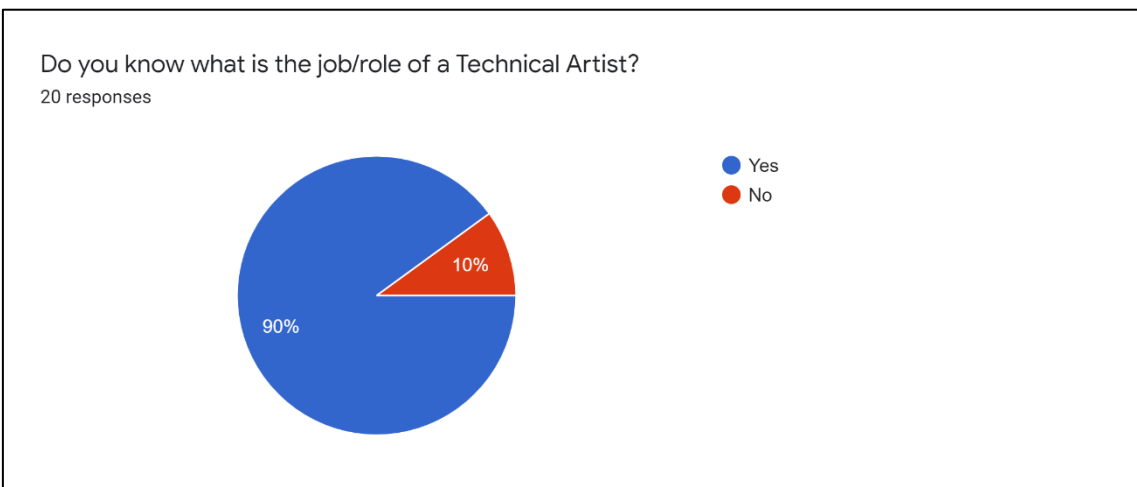
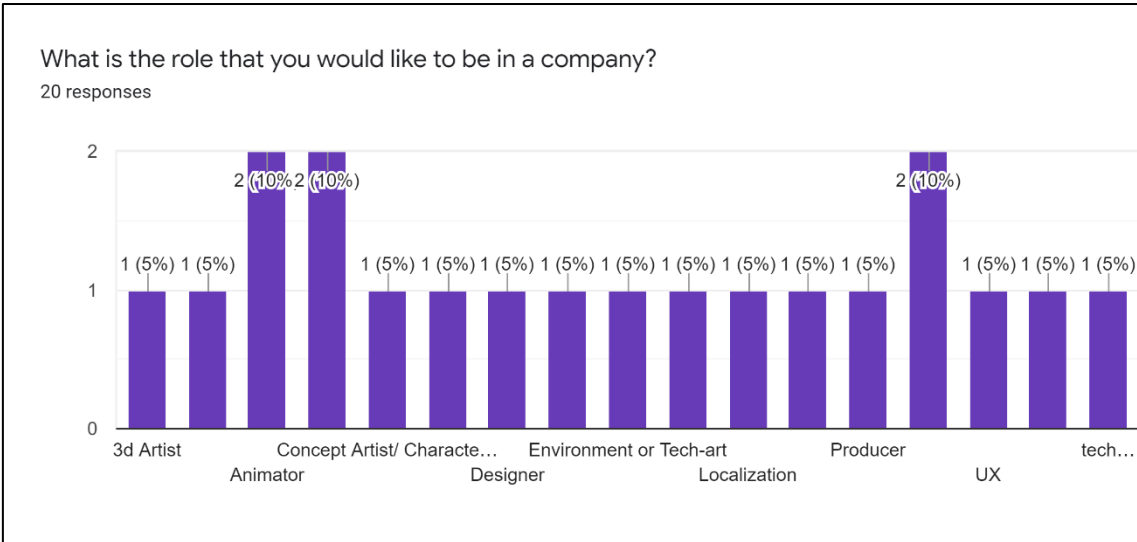


## Section: I study videogames

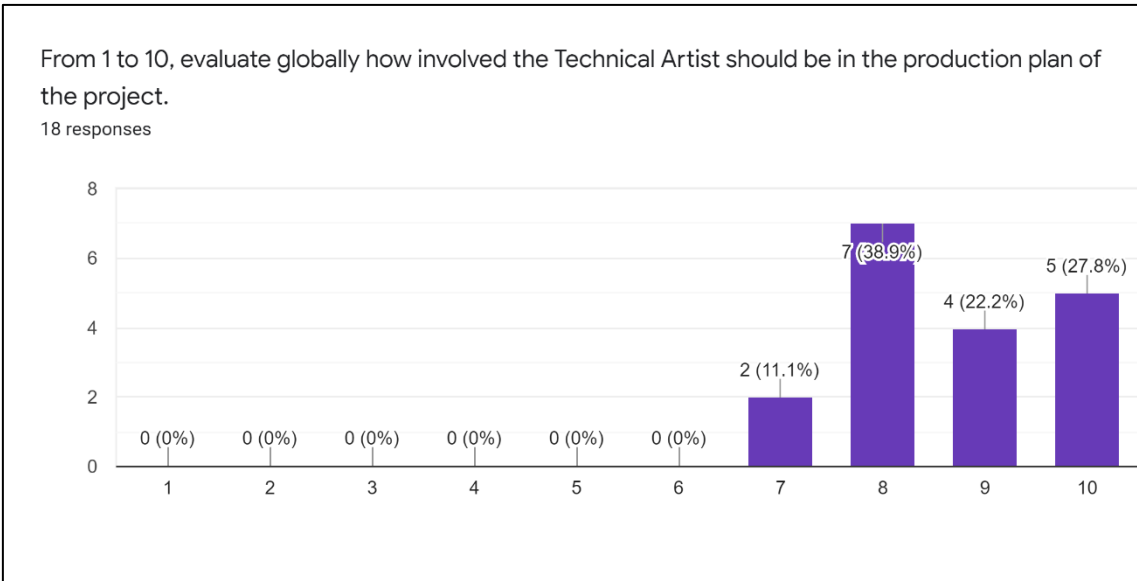
How many years have you been studying video games?

20 responses

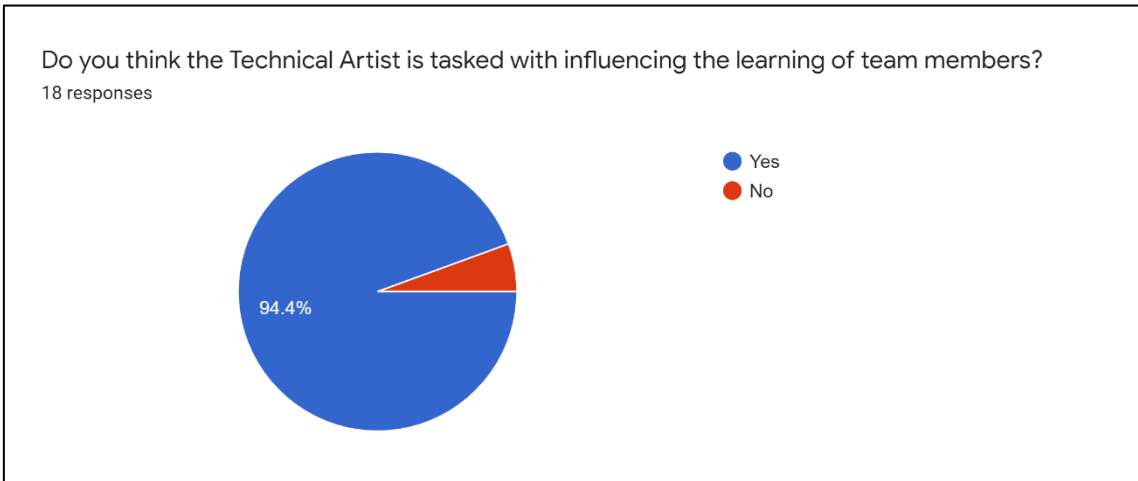
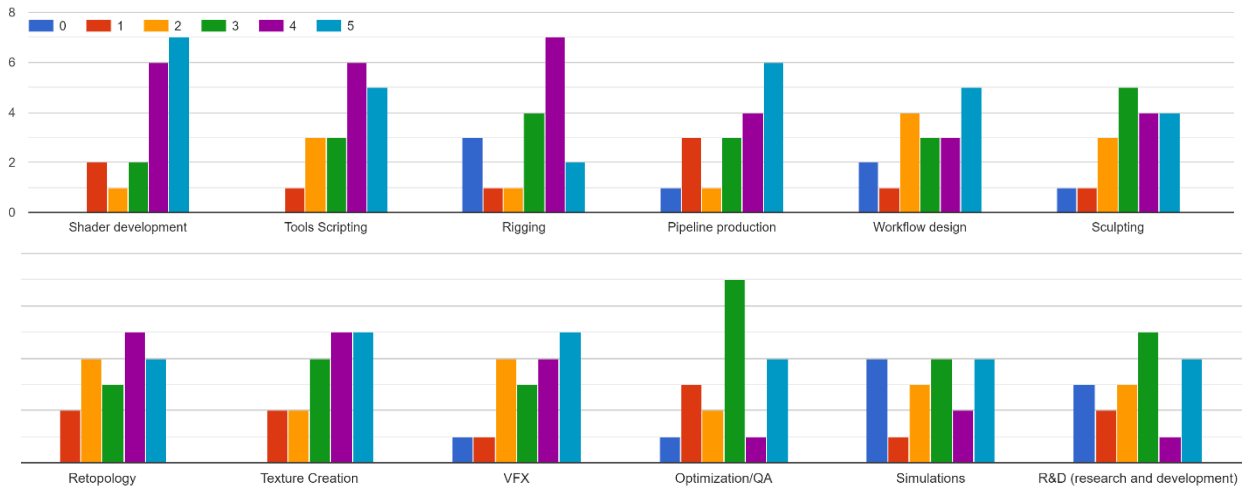




**Section: I study videogames and I know what a TA is**



Evaluate from 0 to 5 what you think should be the work of the Technical Artist in these sections.



### Section: I'm gamer/ I love videogames

