Fecha de recepción: marzo 2023
Fecha de aceptación: abril 2023
Versión final: mayo 2023

El metajuego del desarrollo de juegos

Rubén Darío Hernández Mendo(*)

Resumen: Ciertos videojuegos poseen cualidades particulares cuando permiten a sus jugadores el poder crear niveles o juegos propios, pueden posibilitar una mayor comprensión del proceso de desarrollo de videojuegos. Este tipo de juegos son llamados en el presente texto metajuegos y el lenguaje del juego es metonímico al jugar a crear juegos. Por medio del análisis de distintos juegos, con enfoques aplicados a las dimensiones de representación propuestas Se busca responder a los siguientes cuestionamientos: ¿Cuáles aspectos del desarrollo de juegos emulan tales metajuegos? y ¿Cómo los metajuegos contribuyen a estimular la comprensión de los procesos reales de diseño y desarrollo?

Palabras clave: desarrollo de juegos - metaconstrucción de un medio - juegos de construcción - juegos de construcción de juegos

[Resúmenes en inglés y portugués en la página 202]

(*) Maestro en Educación Superior por la Escuela Normal Superior del Estado de México, Ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Toluca. Actualmente adscrito a Universidad Tecnológica del Valle de Toluca, SAE Institute México y Universidad Anáhuac, con veintidós años de experiencia como docente en niveles Medio Superior, Superior y Posgrado. Autor del decálogo del videojugador para la revista Club Nintendo en 1995. Miembro de grupos de investigación en la Universidad Nacional Autónoma de México, Universidad Autónoma Metropolitana Unidad Azcapotzalco y Red de Investigadores de Juegos de Rol, todos en México.

Introducción: lo que el juego puede y el cine no

Los videojuegos, en una búsqueda continua de nuevas experiencias, de llevar a sus jugadore a lugares inimaginables y a situaciones más allá de lo cotidiano, han tenido que apelar a lo ordinario o las actividades serias de la vida. Los desarrolladores han encontrado en

no pocas ocasiones que el proceso de desarrollo de un videojuego en sí mismo puede ser divertido, desafiante y, en un sentido formal, cautivante, es decir, que pueden crearse videojuegos relacionados con la creación de videojuegos.
Desde la experiencia de Will Wright, abandonando el desarrollo del juego *Raid of Bungeling Bay* para convertir su editor de niveles en el icónico *Sim City* (Maxis 1989), y con ello producir todo un subgénero de simuladores; hasta *Game Builder Garage* (Nintendo 2021), los videojuegos sobre desarrollo de videojuegos permiten que sus jugadores puedan comprender, e incluso participar, del proceso creativo característico del medio. Un cinéfilo puede ver una película que aborde el mundo del cine, pero eso no le permite experimentar con mayor cercanía o fidelidad la producción fílmica. El lenguaje del juego se vuelve metonímico cuando se puede jugar a crear juegos. Juegos como *Game Dev Tycoon* (Greenheart Games 2012) posibilitan la mimesis del proceso, limitada sin duda, pero que de todas formas se propone simular los procesos reales de desarrollo.
El presente trabajo realiza una clasificación de este tipo de juegos, a los cuales se denominará metajuegos, desde un proceso deconstructivo con los casos ya mencionados, incluyendo además *Little Big Planet* (Media Molecule 2008), *Game Builder Garage* (Nintendo 2021), *While True:Learn()* (Luden.io 2018) y *Dreams* (Media Molecule 2020), estableciendo una comparativa de cuáles aspectos logra representar en mayor o menor grado de fidelidad de sus contrapartes de la vida real. El trabajo concluye con una perspectiva acerca del futuro de esta categoría de juegos, buscando responder a las siguientes preguntas: ¿cuáles aspectos del desarrollo de juegos emulan los metajuegos? ¿Cómo los metajuegos pueden contribuir a profundizar la comprensión del proceso real de desarrollo de juegos?

Will Wright metajugando crea *Sim City*

Los metajuegos pueden tener su origen de forma indirecta con la aparición de las computadoras personales, particularmente en Europa durante los años ochenta cuando, de acuerdo con Donovan, "para el tiempo en que Sinclair lanzaba en marzo de 1981 la ZX81, más barata, más poderosa e incluso más exitosa. La mayoría de quienes adquirían la ZX80 llegaban a la misma conclusión: debían crear y vender juegos" (2010, p. 114). Unos pocos años antes, el juego matemático *Game of Life* (Conway, 1970) ya fascinaba a Will Wright con la combinación de sistemas de juego basados en pocas reglas y la complejidad que estos desarrollaban: "es como el *Go*, donde mucha gente que conozco ha perdido grandes trozos de sus vidas en tales empresas." (Donovan, 2010, p. 187). La fascinación de Wright por uno de los juegos de mesa más longevos, y el haber encontrado eso mismo en los modelos matemáticos de Conway, contribuyeron a una búsqueda personal que lo condujeron a la creación de sistemas de juego diferentes a lo que existía en ese momento.

> Todo comenzó a mediados de los 80s con un juego llamado *Raid of Bungeling Bay* (Brøderbund 1984), en el cual Will Wright diseñaba islas que servían de objetivos para un helicóptero de ataque (...) Descubrí que me divertía más

> construyendo las islas de lo que lo hacía volando con el helicóptero (...) (Gamespot.com, 2009)

Entonces el jugar con algo serio experimentó un cambio de perspectiva cuando, por un acto de serendipia[1], Wright piensa en combinar su trabajo con un área de interés, como señala en la entrevista:

> (...) Casi al mismo tiempo, realmente me interesé en la planificación urbana (...) decía (...) especialmente la obra de Jay Forrester. (...) Así que Wright tomó su editor de islas, agregó elementos tales como autos y personas, y utilizó las teorías de Forrester para simular la evolución de los entornos con el paso del tiempo. Tras trabajar él solo en el proyecto por un año, Wright confiaba en que había creado una experiencia única de entretenimiento, aun cuando su filosofía de diseño abierto iba en contra de los típicos juegos computarizados (Gamespot, 2009)

En los términos tipológicos de Caillois (2001), los juegos sobre desarrollo de videojuegos quedan incluidos en la categoría de juegos *mimicry, debido a que se trata de juegos mayormente mim*éticos que imitan cosas, acciones y procesos existentes. Si bien la obra de Wright no fue la primer aproximación al desarrollo de los futuros management sims, puesto que existió *Utopia* (Mattel, 1982) para la consola Mattel Intellivision (Donovan, 2010), sí dio un impulso de ventas y visibilidad a las posibilidades del diseño de gameplay basado en procesos reales:

> Típicamente en el mercado de juegos lanzas un juego y el 80% de las ventas se alcanza en los primeros seis meses (…) *SimCity* tuvo un perfil completamente distinto. El primer año vendió bien,; el segundo año vendió mucho más y el tercero aun más (Donovan, 2010 p. 194)

SimCity además permitió mostrar el potencial de los juegos para abordar cuestiones serias, gracias a la implementación de modelos matemáticos y procedimentales de las áreas vistas como serias dentro del juego. "El juego creativo cambia el propósito de cosas serias en no serias; convierte una escalera o un pasamanos en algo jugable" (Juul, 2019: 193). Esto fue lo que ocurrió cuando Wright propuso que construir ciudades es algo divertido ofreció el juego al estudio Brøderbund: "es una caricatura de cómo funciona una ciudad. Realmente enfatizamos cosas como la gentrificación. (…) ¿Cómo ganas o pierdes? (…) No lo haces (…), solo construyes y administras tu ciudad y observas qué sucede" (Donovan, 2010, p. 190). Para poder comprender con mayor detalle este tipo de juegos será necesario hacer también una revisión de las características del proceso mismo de desarrollo de videojuegos.

Dimensiones del proceso de desarrollo

Cuando se desarrolla un videojuego, es necesario definir diversos componentes, no solo en términos de los "objetos" con los que el jugador realizará el acto lúdico, sino también de las estructuras lógicas que lo gobernarán. Los juegos sobre desarrollo de videojuegos requieren de factores técnicos, ya que deben generar un historial de los juegos desarrollados por un estudio, la implementación de mecánicas nuevas, las curvas de ventas acumuladas por días, semanas o meses e incluso años. Estos volúmenes de datos superarían con facilidad el almacenamiento necesario para guardar todos los juegos existentes de una plataforma en los años ochenta, apenas alcanzaría para una pista de audio o un video tutorial. Además de la inclusión de una estructura narrativa y de formas visuales para comunicar detalles del proceso de desarrollo, como la aparición de errores, de ventas, de recepción del público, estos juegos necesitan orientar sus esfuerzos hacia una parte del proceso mismo de desarrollo, y sacrificar la fidelidad en relación con características como el diseño de arte para el juego o el diseño de gameplay, escenarios, balanceo de la dificultad o sistemas de progresión, solo por mencionar algunos. Por ello es relevante plantear, dentro del presente texto, estos elementos en la forma de dimensiones del proceso creativo, las cuales permiten a los desarrolladores dar énfasis a unos en lugar de otros. Estas dimensiones se aproximan a las líneas rizomáticas: "rn un rizoma no hay puntos o posiciones, como ocurre en una estructura, un árbol, una raíz. En un rizoma solo hay líneas" (Deleuze, Guattari, 2004, p. 14). Es decir, que un metajuego no incluye solamente estados puntuales que describan un juego de manera categórica, sino que forman parte de un continuo donde existe una mezcla de características de enfoques, si bien puede percibirse una dominancia por alguno de ellos o la relación entre dos pudiera ser más fuerte, más acoplada.
Es necesario reconocer que si con el juego se desarrolla algo que sea capaz de crear videojuegos completamente funcionales, e incluso comercializables, entonces conceptualmente se estaría creando una herramienta de desarrollo en lugar de un juego, lo cual se aleja del propósito del presente trabajo. Por lo tanto, los enfoques propuestos aquí, admitiendo que es posible la existencia de más, son:

> . *El enfoque en el proceso de desarrollo como un todo* es aquel centrado y más compatible con los juegos del tipo *Tycoon*. En ellos la estructura se centra en la creación de un estudio de desarrollo y en el "desarrollo" del mayor número posible de juegos, de las tecnologías y tendencias de desarrollo, así como el proceso de "comercializar" el producto "desarrollado". En este enfoque, el "juego producido" en realidad es solamente un constructo abstracto, por lo general solo importarán variables como el título del juego, una descripción de su género, las ventas generadas y la recepción que el juego logró entre el público simulado.
>
> . *El enfoque de construcción de niveles o escenarios* es donde el juego principal permite la creación y la ejecución de escenarios jugables, pero donde también el juego mismo funge como una "plataforma" para diseñar y compartir los

> contenidos ideados por el jugador (en algunos casos se permiten esfuerzos colaborativos entre múltiples usuarios-jugadores). La restricción principal para considerar a un juego dentro de este enfoque es la dependencia total del nivel o escenario creado, teniendo que ser jugado desde el propio juego principal.

Para el e*nfoque de lógica/programación* se da mayor relevancia a los procesos de razonamiento necesarios para la creación de características presentes en muchos videojuegos, así como su implementación por medio de interfaces aproximadas o semejantes a las de herramientas reales de desarrollo. Este enfoque puede parecer el más sencillo, puesto que se relaciona más con los juegos de tipo puzzle, pero tales puzzles pueden ser de una complejidad importante y, en diversos casos, pueden resolverse de múltiples formas.
Como puede notarse, los enfoques propuestos van desde una perspectiva macro hacia una micro en relación con el proceso de desarrollo: desde una visión más global, pero restringida, sobre la construcción real del juego, hasta una orientada al razonamiento necesario para construir elementos lógicos, pero sin concretarlos como un producto independiente, pasando por el punto medio, donde se crean partes importantes, y jugables directamente, pero dependientes por completo del propio juego principal.
A continuación se analizarán casos representativos Cada uno de ellos muestra de forma clara el enfoque correspondiente y el orden en el cual han sido propuestos estos enfoques, si bien todos los juegos suelen contar con algún elemento relacionado con los otros. Es probable que los enfoques señalados funcionen como un punto de partida y abran a nuevas posibilidades de diseño.

Game Dev Tycoon y el proceso macro de desarrollo

El desarrollo de videojuegos sigue, en líneas generales, un proceso similar al de crear una película o una serie de televisión o de streaming, en tanto poseen fase de preproducción, una de producción y una de postproducción. Bajo esa lógica, *Game Dev Tycoon* procura colocar al jugador en el rol del director de un estudio independiente de desarrollo de videojuegos, comenzando en la cochera de la casa de sus padres en la década de los ochenta. Por lo tanto, las primeras acciones que tendrá el jugador será elegir su nombre y el del estudio.
El juego establece entonces un core loop ligado al desarrollo, ya sea de un nuevo videojuego o de un nuevo motor de videojuegos. Para ello se deberá elegir una serie de características del software a desarrollar en la fase de preproducción y comenzar con el trabajo de "programación". Durante este trabajo, se generarán puntos de experiencia en tres indicadores: diseño, tecnología e investigación. Existe además de un cuarto tipo de puntaje: bugs.
Este último representa el nivel de errores que el juego presentará si es lanzado sin modificaciones, afectando la percepción del público acerca de la calidad del juego. Por lo general, después sigue una fase de depuración donde se corrigen dichos errores y se obtiene experiencia adicional al hacerlo. También se producen nuevos bugs, que pueden corregirse. Este loop base se repetirá, con elecciones diferentes en términos del juego/motor en las

fases de producción y postproducción, hasta el momento del lanzamiento. Es importante destacar esta característica, puesto que el juego busca simular algo que ocurre en verdad en el desarrollo de videojuegos. La complejidad de esos sistemas es tal que aun los programadores más experimentados suelen pasar por alto las consecuencias de ciertas acciones de los jugadores; la interacción entre los elementos del juego pueden ser tan diversas e imprevistas que es común se generen combinaciones "ilógicas" pero válidas por la forma en que se ha programado. Parte del trabajo de depuración en el desarrollo de un videojuego no solo se orienta a la corrección de errores, sino también a analizar si se pueden permitir esas interacciones no vistas previamente por el equipo de desarrollo. Esto concuerda con lo observado por Edgar Morin en relación con el valor del error.

> Es interesante ver que el problema del error transforma el problema de la verdad, pero no lo destruye; no se niega la verdad, pero el camino de la verdad es una búsqueda sin fin. Los caminos de la verdad pasan por el ensayo y el error; la búsqueda de la verdad solo se pude (sic) hacer a través del vagabundeo y de la itinerancia (Morin et al 2003, p. 30)

Se entra entonces en la etapa del ciclo de vida del juego ya lanzado, generando ventas y una evaluación por parte del público consumidor que se refleja en la calificación otorgada por las revistas especializadas en el tema. Entre mejor sea la valoración, mayores serán las ventas y el tiempo en que el juego siga vendiendo copias. El resultado aportará entonces el contexto desde el cual el loop principal se reinicia, contando con más o menos recursos económicos para el siguiente proyecto. También los miembros del equipo de desarrollo habrán cambiado, ya que tendrán mayor experiencia como desarrolladores y, además, pueden estar experimentando mayor estrés y cansancio. Si el resultado en ventas del juego es catastrófico el estudio se declarará en bancarrota, terminando así con la partida. Debido a esto se pueden tener juegos con baja recepción, pero que no alcanzan a provocar el colapso económico del estudio, aspecto propio del desarrollo de videojuegos que el juego analizado imita de la realidad. A lo largo de la partida, y conforme el estudio va adquiriendo una mejor reputación, la cantidad de trabajo aumenta, obligando al jugador a contratar personal adicional y teniendo que vigilar aspectos como la carga de trabajo, la capacitación y el descanso de los empleados. Estos pueden renunciar si sufren diversas formas de explotación laboral, pero también pueden experimentar una mayor lealtad al estudio en la medida que se sienten reconocidos por el dueño.

Habrá momentos en los que el estudio no alcance por sí mismo a distribuir suficientes copias y desee llegar a un mayor número de consumidores, por lo cual aparecerán distribuidores (o *publishers*), agentes con quienes se pueden establecer contratos para poder alcanzar ventas mayores, pero también se incrementa el riesgo de pérdida, derivado del porcentaje de ganancia que la empresa distribuidora establezca. De igual manera afectan las penalizaciones por no lograr las ventas y/o recepción crítica especificada al inicio del contrato por el nuevo juego. Otra forma posible para incrementar ingresos o popularidad del estudio es vendiendo o poniendo a disponibilidad del público los motores de juego

que el estudio haya desarrollado. En este último componente por lo general el indicador beneficiado excluye al otro.
El juego se gana si el personaje obtiene el reconocimiento de trayectoria profesional, el lifetime achievement, si bien se puede seguir desarrollando juegos posterior a este logro. El juego además emula el proceso de cambio que ha sufrido la tecnología de videojuegos con la llegada de nuevas consolas, las cuales son parodias, ¿o quizá homenajes?, de las que realmente han existido a lo largo de los años, aunque sus ciclos de vida no cubren con total precisión los años en los que fueron puestas a la venta y/o retiradas de producción.
Si bien el juego busca presentar al jugador el proceso de desarrollo en términos generales, el nivel de detalle se ve afectado al tener que simplificar las decisiones del jugador. La asignación de tareas, sean para el director del estudio o el personal contratado, se reduce al porcentaje de tiempo establecido para los empleados. El personal que se encuentre en capacitación estará enfocado completamente a esta actividad, y en cuanto al trabajo de programación este se representa con solo un contador de bugs, en ningún momento el jugador deberá aplicar conocimientos reales de programación de juegos.
Game Dev Tycoon, publicado en 2012, se ha mantenido con ventas suficientes para lanzarse en diversas plataformas de dispositivos digitales, pero poco en consolas, con excepción del lanzamiento en 2020 para la *Nintendo Switch*. Una característica interesante fue la implementación de una versión especialmente diseñada para los canales de piratería, la cual fue distribuida directamente por los propios desarrolladores en sitios de torrents. Dicha versión castigaba a los jugadores gradualmente con afectaciones dentro de la economía del juego, llevándolos inevitablemente a la bancarrota. Algunos jugadores se quejaron de esto en foros, exhibiéndose a sí mismos como consumidores ilegales del juego.
Con esto queda expuesto cómo el enfoque macro del proceso, con un *enfoque en el proceso de desarrollo como un todo*, se ve obligado a simplificar el detalle fino del desarrollo, pero permite una visión de conjunto sobre el desarrollo de juegos que para muchos jugadores podrá resultar desconocida, contribuyendo a comprender un poco más sobre el proceso real de creación de un videojuego. A continuación se analiza el caso para el e*nfoque de construcción de niveles o escenarios*.

Media Molecule y sus metaconstructores

Cuatro años antes del lanzamiento de *Game Dev Tycoon*, el estudio exclusivo de Sony, Media Molecule, causaba sensación con el lanzamiento del primer juego de la serie *Little Big Planet* (2008), el cual presentaba a una de las mascotas no oficiales de la marca PlayStation: Sackboy/Sackgirl, un pequeño títere tejido que debía explorar y salvar el mundo de los sueños y que servía como justificación narrativa para cumplir con tres principios: jugar, crear, compartir.
Los principios del juego se organizan en una especie de pirámide, donde el de jugar abarca la totalidad de la experiencia y se entreteje con los otros dos, es decir, que siempre se está jugando, pero la naturaleza de cada acto lúdico es distinto en función de los otros dos principios.

El principio de crear consiste en un módulo de desarrollo de niveles, robusto pero al mismo tiempo sencillo. Lo primero obedece a que el jugador-creador dispone de un conjunto de componentes que ha ido recolectando dentro del componente principal de jugar, en su modo historia, siendo parte de los coleccionables que deberá obtener en retos de plataformeo. Estos componentes están representados por medio de burbujas, cuyo contenido es precisamente el componente indicado. Además se incluyen diferentes tipos de objetos que sirven como piso o plataformas, mecanismos básicos como interruptores, sensores y "mecanismos", y también se cuenta con conjuntos de apariencia estética para los componentes. Entre más progrese el jugador en la recolección de estos componentes en el modo historia, mayor será el abanico de recursos a usar en el editor de niveles.

El principio de jugar se deriva de la ausencia de programación por medio de lenguajes textuales, mediante la implementación de un sistema de nodos que permite configurar algunos comportamientos simples de movimiento o reacción de objetos y personajes. Esto distingue al juego si se le compara con *Game Dev Tycoon*.

El principio de compartir se logra por medio de una comunidad diseñada para que un jugador pueda acceder y jugar al contenido creado por otros jugadores, así como él mismo puede poner en disponibilidad de otros sus creaciones.

Little Big Planet perdió presencia de manera gradual debido a cierto nivel de repetitividad en sus contenidos, a pesar de que el editor de niveles fue cada vez más complejo. Esto permitió que el estudio lanzara *Dreams en 2020*, el cual mejoró la propuesta de su antecesor. Ahora el editor de niveles quedó liberado por completo de la estética original de *Little Big Planet*, la cual estaba orientada a una audiencia muy joven, y por lo tanto sus personajes y elementos tendían a una ternura muy marcada. Además de estimular un proceso de creación colaborativo con jugadores-creadores diversos, el juego permite implementar casi cualquier tipo de idea, además de permitir la utilización de recursos audiovisuales literalmente infinitos. La única restricción que al momento posee es la propia del *enfoque de construcción de niveles o escenarios*: todos los niveles, e incluso los proyectos de mayor alcance, capaces de calificar como juegos en forma, solo pueden jugarse dentro del propio juego.

El principal aporte de este subtipo de juegos, en el sentido de la comprensión del proceso de desarrollo de juegos, radica en poder mostrar al jugador el comportamiento del sistema de juego en sí mismo: el jugador puede identificar características asociables al gameplay, la dificultad, curvas de aprendizaje, balanceo de habilidades, tiempos o recursos del propio juego, etcétera. Como aspecto débil, en ese sentido, el juego solo permite una apreciación cualitativa del comportamiento del juego, puesto que su propio enfoque no considera la generación de métricas ni de mecanismos formales de testing. De cualquier forma, es incuestionable que este tipo de juegos puede estimular la intuición o expresar con mayor formalidad un aprendizaje tácito (Schön, 1983) sobre el diseño del nivel. Dichos aprendizajes son tácitos en términos de que el jugador, en su rol como creador, tiene la capacidad de modificar con una intención clara el aspecto que está manipulando, por ejemplo la dificultad, pero solo puede hacerlo si se le pide que explique con mayor detalle sus acciones o las razones por las cuales ha tomado una decisión específica; en este caso, es muy probable que no sepa explicarlo. En otras palabras, estos elementos del juego estimulan una habilidad práctica en el jugador como creador.

Tras analizar *Game Dev Tycoon* y los juegos de Media Molecule, es posible identificar con claridad dos niveles específicos dentro del proceso general de desarrollo de juegos: Un nivel de gestión del juego, es decir, la visión del mismo como un proyecto de desarrollo de software o un proyecto de desarrollo creativo, el cual es el propuesto con los juegos *con enfoque en el proceso de desarrollo como un Todo*, y también un nivel de características inherentes al medio, en este caso, el diseño de un sistema de juego, ligado a los juegos con e*nfoque de construcción de niveles o escenarios*. Falta un tercer nivel, el cual se relaciona con la implementación de los diversos elementos a crear mediante las plataformas o interfaces necesarias. Es decir que, para poder contar con una perspectiva integral del desarrollo de juegos, es necesario que también se exploren los juegos que posean el e*nfoque de lógica/programación*.

While True: Learn() y la meta-lógica del juego

Comprender los procesos lógicos que rigen un juego, que en su conjunto forman el sistema de juego, puede ser una tarea en verdad compleja, y su funcionamiento depende en gran medida de la manera en la que se lo implementa, lo que vuelve necesario cierto grado de comprensión de la programación o de procesos computables. Es el caso del juego seleccionado para ilustrar el e*nfoque de lógica/programación*. Este desarrollo de Luden.io en 2018 no se enfoca de forma exacta en la programación de videojuegos, aunque acerca a los jugadores al proceso de aprendizaje sobre el funcionamiento de un videojuego, y se establecen el aprendizaje máquina, la inteligencia artificial, las redes neuronales y el big data. El juego entonces es construido a partir de una premisa narrativa sencilla: un joven ingeniero quiere comprender el lenguaje de los gatos y para ello requiere del desarrollo de un software capaz de realizar semejante tarea. Esto lo logrará resolviendo problemas de dificultad incremental relacionados con el aprendizaje de los diversos conceptos y técnicas de programación antes referidas.
El core loop se construye a partir de la solución de un nuevo problema en el que el jugador tendrá que utilizar un número limitado de bloques de código, los cuales pueden configurarse con la creación de condiciones lógicas que filtren los datos (en realidad, se trata de figuras geométricas con propiedades distintas, como forma y color). El sistema de bloques se asemeja a los lenguajes visuales existentes en múltiples plataformas reales de desarrollo de software, incluidos aquellos orientados al desarrollo de videojuegos. Cada puzzle consiste de tres secciones base:

> . una "zona fuente" de datos de entrada, la cual genera una o más secuencias de datos, en los términos previamente descritos, que deberá ser "procesada" por el código
>
> . una "zona de procesamiento", donde los datos serán filtrados u organizados de acuerdo con sus características y velocidades de aparición como entrada de

> datos. Esta es la zona donde realmente el jugador tiene la posibilidad de elegir los distintos bloques de código que realizarán las operaciones.
>
> . una "zona de salida", la cual contendrá al menos dos puntos de salida. Uno para los datos válidos que requiere el puzzle y otra como basurero, el cual recibirá todos los datos que nosean pertinentes para el puzzle a resolver. Pueden requerirse más puntos de salida porque en diversos puzzles se deben separar al menos dos grupos de datos, cada uno con una característica particular, como sea el color o la forma e incluso en combinaciones específicas

Como se puede notar, la estructura del juego es sencilla, pero la complejidad de los ejercicios llega a incrementarse. Esto tiene correspondencia con la realidad de la programación, puesto que la combinación de pequeños bloques de código pueden llevar a sistemas muy elaborados, consistentes de cientos hasta millones de estas pequeñas estructuras. Lo cierto es que el juego no llega a contar con más de cuarenta nodos en sus ejercicios más difíciles, aunque esto no implica que sean fáciles de resolver.

Uno de los problemas que se observan en el juego, desde la perspectiva de la herramienta educativa, radica en su un sistema poco amigable con el jugador. Si bien permite un número ilimitado de intentos para resolver cada puzzle, la retroalimentación que proporciona al jugador es escasa, por lo tanto, irónicamente, mucho más cercana a la realidad del proceso genuino de desarrollo. En apariencia esto no debería ser criticado, pero la razón por la cual el juego falla es precisamente debido al e*nfoque de lógica/programación*: el juego solo se centra en el aspecto de lógica y deja de lado otros elementos que podrían relacionarse como los tiempos para encontrar la solución. Un problema que es tratado de forma superficial son los costos, que solo se relacionan con el uso del menor número posible de nodos permitidos para el ejercicio, mas no con el tiempo invertido en cada corrida de prueba. Además, el diseño como puzzle puede dar ideas inapropiadas sobre la calidad de la solución, así como un diseño de ejercicios más rígido. En otras palabras, puede afirmarse que la eficiencia en el juego no tiene tanta similitud con los criterios reales de eficiencia en un proyecto real.

Game Builder Garage o Unity for Dummies

Si bien el desarrollo de juegos estuvo ligado por décadas a los profesionales de las áreas de cómputo y diseño, la aparición de herramientas de desarrollo, mucho más accesibles tanto económica como técnicamente, ha permitido una apertura hacia el público en general para la creación de videojuegos. Uno de los casos más sonados se suscitó en 2015, cuando un joven de 24 años, Toby Fox, desarrolló prácticamente solo el afamado juego de rol *Undertale*. Gran parte del éxito de dicho juego radica en que el jugador puede resolver todos los enfrentamientos del juego desde dos enfoques opuestos, uno de combate tradicional y otro pacifista, donde las opciones de diálogo entre el personaje principal y el enemigo pueden evitar el choque entre ambos. Desde una perspectiva técnica, se ha comentado con frecuencia que

Fox no sabe programar. Este comentario en realidad es una verdad a medias, puesto que es cierto que no conoce lenguajes de programación como Java o C, pero eso no implica que no sepa razonar la lógica necesaria para implementar el juego. Lo cierto es que trabajó con la herramienta de desarrollo *Game Maker 2000*, la cual posee un sistema de construcción estilo LEGO, donde la combinación de bloques funcionales y la posibilidad de combinarlos en bloques de mayor complejidad (principios que se traducen con facilidad a conceptos de programación orientada a objetos y eventos) permiten a usuarios con poca o nula experiencia directa con la programación realizar tareas de programación. De hecho, el propio Fox lo señala, cuando ya se le preguntaba por el porteo del juego a otras plataformas de juego:

> Toby explica que el (sic) no se encargaría de hacer un port del juego (para WiiU), debido a que programo (sic) *Undertale* en *Game Maker* un programa que no permite exportarlo a un formato que trabaje en plataformas de Nintendo, aparte él no sabe programar con lenguajes de programación (Langaria.net, 2016).

Estrictamente hablando, lo afirmado por Fox es incorrecto. Sí sabe programar, solo que el lenguaje que utiliza tiene un enfoque más visual que textual. Él es capaz de razonar el proceso, las interacciones posibles que permiten el juego y expresarlas en el lenguaje (visual) de programación que posee *Game Maker*.

Con la cada vez mayor presencia de metajuegos de desarrollo de juegos, Nintendo crea en 2021 su propio producto: *Game Builder Garage*, el cual se convierte en una especie de sucesor espiritual de uno de su primer metajuego: la serie *Super Mario Maker* (Nintendo, 2015), que logró el éxito al implementar un editor de niveles temáticos de los juegos de la franquicia estandarte de la empresa. Como podrá inferirse de los conceptos previamente expuestos, los juegos mencionados pertenecen al e*nfoque de construcción de niveles o escenarios*, lo cual restringió de manera importante el potencial creativo de los jugadores, aún cuando diversos niveles destacaron por su dificultad e ingenio. Por ello *Game Builder Garage* se enfocó en crear un juego con e*nfoque de construcción de niveles o escenarios*, pero con mayor peso en el e*nfoque de lógica/programación*, esto es que siguió como una plataforma para creación de juegos, pero con mayor soporte para el segundo e*nfoque*. Esto fue logrado a partir de la creación de un lenguaje visual de desarrollo propio, basado en nodos, pero denominados como nodons, pequeños entes de código o construcción genométrica de escenarios. Resulta claro el estilo de marca de Nintendo para caracterizar la filosofía de diseño de juegos de la compañía, donde hasta los componentes de desarrollo de juegos son personajes.

El juego consta de dos grandes modos: por un lado hay un sistema de tutoriales, siete en total, partiendo en el primer tutorial desde los conceptos y los nodons básicos. Tras terminar con dicho tutorial, el jugador puede acceder al modo libre, pero el propio juego recomienda realizar los seis restantes para una mejor comprensión de sus mecánicas creativas. Esos tutoriales restantes van acercando al jugador a juegos más complejos, llegando a poder diseñar escenarios tridimensionales y controles de cámara, entre otras características. El segundo modo de juego es el ya referido modo libre, donde el jugador contará con la totalidad de nodons del juego para crear. Debido al diseño del juego con un modo tuto-

rial tan robusto, el modo libre ya no aporta retroalimentación y permite experimentar la creación de los juegos, su imaginación y comprensión.Las herramientas que son ofrecidas establecen un límite, pero hay otro: es inevitable que el juego manifieste restricciones técnicas por la plataforma en la cual se opera. La cantidad de memoria para construir un juego se ve limitada por la propia memoria del sistema, la cual además está ejecutando el juego base. Además, la consola tiene una capacidad de almacenamiento mucho más limitada, partiendo del hecho que el sistema cuenta de forma interna con un máximo de 32 GB, de los cuales su propio sistema operativo suele ocupar unos 8 GB y los medios externos de almacenamiento son tarjetas micro SD, la consola soporta hasta 1TB extra, pero los costos de las unidades de esta capacidad superan ampliamente discos duros rígidos en plataformas no dedicadas. Aun teniendo en cuenta este problema, puede afirmarse que este juego propone una relación más integrada entre los *enfoques de construcción de niveles o escenarios* y *de lógica/programación*.

Conclusiones: los metajuegos de desarrollo de videojuegos invitan a conocer más sobre el medio

Como se ha expuesto, el crecimiento exponencial del poder de cómputo y del almacenamiento de datos ha permitido que los juegos sobre desarrollo de videojuegos se vuelvan viables en los últimos años. Aún dentro de la corta historia del medio, la capacidad para almacenar partidas donde el jugador pudiera crear un nivel completo o, incluso, un juego de mediana duración, por ejemplo, de algunas horas, comienza a permitir en consolas el guardado de datos en el orden de kilobytes hacia el final de la década de los años 1990. En algunos casos, se alcanzaban ya los megabytes suficientes para una construcción con mayor cantidad de detalles. Asumiendo que la historia de los juegos comienza desde siglos previos a la era Cristiana, el grado de desarrollo lúdico se mantuvo casi estático por algunos milenios. Debe notarse que todos los metajuegos analizados en este artículo pertenecen al siglo XXI y que su número se incrementa a partir de la segunda década del mismo siglo. Bajo esa escala, los metajuegos resultan una invención nueva, si bien el interés por ello ha existido desde antes.

> No tengo idea qué conectará conmigo, o dónde, o qué tipo de conexión formará. Y entonces, en lugar de esperar de forma pasiva, deseo actuar con propósito y atesorar los encuentros resultado de mis decisiones. Siento lo mismo al conocer personas (Kojima, 2021, p. 9)

El jugar dentro de un juego que permite crear videojuegos es una forma en la que se puede cumplir con lo sugerido por Hideo Kojima: un proceso activo, con propósito específico y que además suma una dimensión que el jugador puede valorar, incluso en términos afectivos, volviéndose un potencial creador de videojuegos. Este posicionamiento también es compatible con principios rizomáticos, puesto que las conexiones que se plantean son cruces entre líneas rizomáticas que propician rupturas asignificantes: "Un rizoma puede

ser roto, interrumpido en cualquier parte, pero siempre recomienza según ésta o aquella de sus líneas y según otras" (Deleuze, Guattari, 2004 p. 15). Para ello es necesario concebir la presencia de rizomas en el metajuego, entendiéndolo como una construcción abierta a partir de la complejidad inducida, como señalaba Wright acerca del *Go* y de *Game of Life*. El segundo rizoma reside en la actividad del propio jugador, específicamente en aquello relacionado con la experiencia del juego. El diálogo entre ambos estimula una mayor comprensión en el jugador sobre los procesos constructivos de juegos.
Esta nueva posibilidad de metajugar se relaciona con lo señalado por Donald Winnicott, quien postula sobre la persona que juega que

> Si (…) no recibe esta oportunidad (de jugar con objetos) entonces no hay área donde (…) pueda (…) tener experiencia cultural; en consecuencia no tiene un vínculo con la herencia cultural, y por lo tanto no habrá contribución al fondo cultural." (Winnicott, 2005, p. 137)

El acto de jugar contribuye de forma significativa para el desarrollo de las personas, y con ello potencialmente también lo hará con la sociedad misma. La creación de juegos, entonces, permite acercar a creadores y consumidores en torno del alfabetismo lúdico propuesto por José Zagal, quien plantea como evidencias del alcance de este grado de aprendizaje las habilidades de "jugar juegos (…) comprender significados con respecto de los juegos (y) hacer juegos " (Zagal, 2011, p. 23). Desde su propuesta, los jugadores que se limitan solo a jugar, a operar los mandos y a divertirse, solo alcanzarían habilidades relativas a la comprensión. Por eso, los metajuegos de desarrollo de videojuegos ofrecen una oportunidad particular para formar a futuros diseñadores de juegos.

Nota

1. La serendipia no debe interpretarse solo como casualidad, sino como un proceso de descubrimiento derivado en efecto de un evento circunstancial, pero observado por alguien quien decide investigar qué es lo que causa o explica lo sucedido.

Referencias bibliográficas

Caillois, R. (2001) *Man, play and games*. The University of Illinois Press.
Deleuze, G. Guattari, F. (2004) *Mil mesetas. Capitalismo y esquizofrenia*. Pre-Textos..
Donovan, T. (2010) *Replay. The History of Videogames*. Yellow Ant.
Gamespot (2 de febrero de 2009) Gamespot Presents: Geoff Keighley behind the games. SIMply Divine. The story of Maxis Software. The Idea. *Archive.org.* https://web.archive.org/web/20090202040300/http://www.gamespot.com/features/maxis/page2.html

Juul, J. (2019) *Handmade pixels. Independent Video Games and the Quest for Authenticity*. MIT Press.
Kojima, H. (2021) *The Creative Gene. How Books, Movies and Music Inspired the Creator of Death Strainding and Metal Gear Solid*. Viz Media.
Langaria (4 de febrero de 2016) El creador de Undertale desmiente los rumores de la versión de Wii U. *Langaria*. https://langaria.net/2016/02/04/el-creador-de-undertale-desmiente-los-rumores-de-la-version-de-wii-u/
Morin, E.; Ciurana, E.; Motta, R. (2003) *Educar en la era planetaria*. Gedisa.
Schön, D. (1988) *El profesional reflexivo. Cómo piensan los profesionales cuando actúan. Temas de Educación Paidós*. Paidós.
Winnicott, D. (2005) *Playing and Reality*. Routledge.
Zagal, J. (2011) *Ludoliteracy. Defining, Understanding and Supporting Games Education*. ETC Press.

Abstract: Certain videogames have particular qualities when they allow players to create levels or games of their own. They can make possible a greater game development process understanding. This type of games will be defined as metagames, the game language is methonymical when the game is creating games. The questions seeking for an answer are: Which game developments aspects are emulated by metagames? And How metagames contribute to a better understanding for real design and development processes? This will be answered trough an analysis of different games, applying proposed representational dimensions approaches.
It aims to answer the following questions: What aspects of game development emulate such metagames, and how do metagames contribute to stimulate the understanding of real design and development processes?

Keywords: game development - media metaconstruction - building games - game building games.

Resumo: Certos videogames têm qualidades particulares quando permitem que seus jogadores criem seus próprios níveis ou jogos, podem possibilitar uma maior compreensão do processo de desenvolvimento de videogames. Esses tipos de jogos são chamados no presente texto de metajogos e a linguagem do jogo é metonímica ao jogar para criar jogos. Procura responder às seguintes questões: Que aspectos do desenvolvimento de jogos emulam tais metajogos? e Como os metajogos ajudam a estimular a compreensão dos processos reais de design e desenvolvimento?

Palavras chave: desenvolvimento de jogos - metaconstrução de um meio - jogos de construção - jogos de construção de jogos

[Las traducciones de los abstracts fueron supervisadas por su autor]
