

Programación visual

Programación orientada a objetos:

La programación orientada a objetos se basa en el concepto de crear un modelo del problema de destino en sus programas. La programación orientada a objetos disminuye los errores y promueve la reutilización del código. Python es un lenguaje orientado a objetos. Los objetos definidos en Python tienen las características siguientes:

Identidad. Cada objeto debe ser distinguido y ello debe poder demostrarse mediante pruebas. Las pruebas `is` y `is not` existen para este fin.

Estado Cada objeto debe ser capaz de almacenar el estado. Para este fin, existen atributos, tales como variables de instancias y campos.

Comportamiento. Cada objeto debe ser capaz de manipular su estado. Para este fin existen métodos.

Python incluye las características siguientes para dar soporte a la programación orientada a objetos:

Creación de objetos basada en clases. Las clases son plantillas para la creación de objetos. Los objetos son estructuras de datos con el comportamiento asociado.

Herencia con polimorfismo. Python da soporte a la herencia individual y múltiple. Todos los métodos de instancias de Python son polimórficos y se pueden alterar temporalmente mediante subclases.

Encapsulación con ocultación de datos. Python permite ocultar los atributos. Cuando se ocultan los atributos, se puede acceder a los mismos desde fuera de la clase únicamente mediante los métodos de la clase. Las clases implementan métodos para modificar los datos

Programación orientada a eventos

La programación orientada a eventos se refiere a un modelo de la programación de computadoras, donde se utilizan los eventos que suceden para la determinación del flujo de control de un programa.

No es un tipo de tecnología o lenguaje de programación, sino un enfoque que se implementa durante la etapa de desarrollo del producto. Básicamente, separa la lógica de procesamiento de eventos del resto del código de un programa.

Este tipo de programación está diseñado para descubrir eventos mientras van sucediendo, usando un procedimiento apropiado de manejo de eventos para tratarlos, normalmente mediante una llamada a una función o método.

Teóricamente, el estilo de esta programación es compatible con todos los lenguajes de programación, aunque puede ser diferente en la forma de implementarse.

Dependencia de eventos

El flujo del programa viene dado por eventos que pueden ser acciones del usuario, mensajes de otros programas, etc., separando la lógica de procesamiento de los eventos del resto del código de un programa, contrastando así con el procesamiento por lotes.

Los eventos en sí pueden ser desde aceptar o rechazar una solicitud de préstamo, denominado evento de alto nivel, hasta que un usuario presione una tecla, que es un evento de bajo nivel.

Orientada al servicio

Se utiliza para escribir programas diseñados para el servicio sin ralentizar la computadora, ya que la orientación al servicio solo consume poco poder de procesamiento. Además, los servicios se ejecutan por lo general en el trasfondo del sistema operativo.

Eventos

Es una condición que surge durante la ejecución de un programa y que requiere alguna acción por parte del sistema. Cada evento es diferente por naturaleza, algunos requieren que el programa recobre y muestre cierta información, y otros que se inicien algunos cálculos y cambios de estado.

Procesos de desarrollo visual en proyectos distribuidos y de escritorio

Caracterización de los Sistemas Distribuidos. Un sistema distribuido es aquel en el cual los componentes, localizados en equipos en red, se comunican y coordinan sus acciones mediante el envío de mensajes. Esta definición presenta las siguientes características significativas de los sistemas distribuidos: concurrencia de los componentes falta de un reloj global fallos independientes de los componentes. Los sistemas distribuidos presentan una serie de retos: Heterogeneidad, si hablar de heterogeneidad nos referimos a la variedad y diferencia que podemos encontrar en los elementos que componen una red de computadoras sobre la que se ejecuta un sistema distribuido. Dicha heterogeneidad no sólo se aplica a las redes y al hardware de las computadoras, sino también a los sistemas operativos, los lenguajes de programación y las implementaciones en las que trabajan los diferentes desarrolladores.

Herramientas y lenguajes de programación visual

Algunos ejemplos de estas herramientas son: Scratch, adecuada a partir de los 8 años, ScratchJr, que es la versión de Scratch adaptada para niñas y niños de entre 5 y 7 años, Lightbot, etc. Incluso la empresa Google ha creado un precioso doodle llamado Coding for Carrots que ejemplifica este tipo de herramientas.

Estas herramientas se usan con éxito en todos los niveles educativos formales (Infantil, Primaria, Secundaria, Bachillerato y Universidad), y también en escenarios educativos no formales. Se pueden usar en una gran diversidad de materias como lengua española, lengua extranjera, matemáticas, ciencias, historia, artes, etc

Scratch

El lenguaje de programación visual más popular y utilizado es Scratch. Fue lanzado por primera vez en 2007 y tiene como propósito enseñar a los niños a programar de manera sencilla. Su sucesor para niños de entre 5 y 7 años, ScratchJr, está disponible como una aplicación gratuita. También hay otros lenguajes de programación gráfica basados en Scratch, como BYOB 4.0/Snap!.

Blockly

Blockly es una llamada biblioteca. Proporciona un editor de programación visual al que se añaden aplicaciones Android, iOS y web. Blockly también utiliza bloques gráficos que encajan entre ellos. Los algoritmos programados se pueden exportar como código JavaScript o Python, por ejemplo.

NEPO

NEPO es gratuito, está basado en Scratch y utiliza la biblioteca Blockly. Esta biblioteca se ha ampliado con funcionalidades propias. Una ventaja de NEPO son sus interfaces abiertas, que permiten controlar otros sistemas de hardware o robots.

Grape

Grape es un entorno de desarrollo gráfico. Permite incluso a los principiantes en programación programar con microcontroladores en pasos simples.

App Inventor

App Inventor proviene originalmente de Google. Esta interfaz gráfica permite programar aplicaciones para teléfonos móviles Android con bloques gráficos.

Ardublock

Este lenguaje de programación gráfica está especialmente diseñado para programar el microcontrolador Arduino sin introducir texto.

Pure Data

Este lenguaje de programación visual está orientado tanto a flujos de datos como a los entornos de desarrollo. Pure Data permite producir software multimedia interactivo, por ejemplo, para sintetizadores.

Lego Mindstorms

Una serie de productos del fabricante de juguetes Lego, cuyo núcleo es la pieza de Lego programable: los motores eléctricos, sensores y piezas de tecnología propios de Lego permiten construir y programar robots y otros sistemas interactivos.

Diseño

Proceso de planificación creativa, en el que se persigue la solución para algún problema en concreto.

Proceso

En el diseño se elaboran un concepto y se crea una especificación de un artefacto de software, pensado para cumplir unos objetivos, utilizando un conjunto de componentes primitivos y sujeto a restricciones.

Multimedia

Es una combinación de texto, arte gráfico, sonido animación y video que que permiten a las personas comunicar ideas e información.

Diseño

de programación visual.

Integración

Integración de contenidos y componentes es una actividad de desarrollo de software que combina varios componentes de software diferentes en un conjunto.

Interfaces de programación visual

Conjunto de métodos y propiedades que no tiene ninguna implementación.

Interactividad de componentes

El usuario genera un interactividad con los componentes que se pueden presentar en la interfaz como lo son, botones, listas, etiquetas, etc.

DESARROLLO DE APLICACIONES CON PROGRAMACIÓN VISUAL

EVENTO VISUAL

Es una señal que comunica a una aplicación una serie de procedimientos y funciones que se ejecutan cuando alguno de sus componentes lo necesita.

FUNCIONALIDAD

La funcionalidad es importante porque permite realizar una tarea y, a continuación, devuelve el control al código de llamada, al ejecutarse.

COMPONENTES Y HERRAMIENTAS VISUALES DE ENTORNOS DE DESARROLLO.

es un programa informático compuesto por un conjunto de herramientas de programación.

Editor de texto.

Depurador.

Compilador

COMPONENTES VISUALES

proporciona un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

ACCESIBILIDAD

Tiene como objetivo lograr que las páginas web sean utilizables por el máximo número de personas, independientemente de sus conocimientos e independientemente de las características técnicas del equipo utilizado para acceder a la Web.

APARIENCIA

La apariencia es de suma importancia ya que será lo que el usuario verá, por lo tanto tiene que ser atractivo a la vista.

COMPORTAMIENTO

Son los actos o acciones manifiestas de los consumidores dentro de una web o App que se pueden observar a través de herramientas de medición o análisis de comportamiento.

DATOS

Sirven para organizar información y así facilitar el acceso rápido.

DISEÑO

Área enfocada en el desarrollo de interfaces digitales, como el diseño de sitios y aplicaciones para web.

Conceptos de Videojuegos

Introducción

Un videojuego es una aplicación interactiva orientada al entretenimiento que, a través de ciertos mandos o controles, permite simular experiencias en la pantalla de un televisor, una computadora u otro dispositivo electrónico. Los videojuegos se diferencian de otras formas de entretenimiento, en que deben ser interactivos; es decir, los usuarios deben involucrarse activamente con el contenido. El concepto de videojuego se utiliza para referirse a cualquier juego digital interactivo, independientemente de su soporte físico. Pueden ser muy distintos entre sí, tanto en complejidad como en calidad gráfica y en temática.

GameDesigner

Los diseñadores de videojuegos o Game Designers tienen un rol muy importante en el proceso de creación y deben ser conocedores de las tendencias en videojuegos para poder construir un producto atractivo para los consumidores. A continuación, vamos a exponer algunas de las funciones que tiene este rol.

Además de la figura de Game Designer, existen otro tipo de perfiles que hacen posible que podamos disfrutar de ellos como: programadores, animadores 3D, modeladores 3D, entre otras.

Esta industria es muy competitiva y las empresas cada vez demandan profesionales mejor preparados con formación en el sector del videojuego. Por este motivo, estudiar un Grado Superior en Videojuegos y Ocio Digital te ayudará a conseguir trabajo como desarrollador en el mundo de los videojuegos. Asimismo, si lo que te gustaría es modelar distintos personajes y escenarios también existe la opción de especializarte cursando un Grado Superior en Animaciones 3D, Juegos y Entornos Interactivos.

- **Define la mecánica del videojuego**

La figura del Game Designer será la encargada de definir el argumento del videojuego. Deberá pensar qué historia quiere contar y plantear la mecánica que va a seguir el juego para poder establecer, más tarde, todos los otros aspectos. Esbozará un primer esquema para trasladar al resto de departamentos y que puedan comenzar a gestionar el videojuego por proyectos.

- **Gestiona proyectos con otros departamentos**

Esta figura, creadora de videojuegos, tendrá la responsabilidad de trasladar las ideas a los distintos departamentos y que el concepto sea entendido por todos los trabajadores inmersos en la creación del mismo. Dividir las tareas por proyectos ayudará a agilizar el lanzamiento del juego y que sea todo un éxito.

- **Delimita las reglas y niveles**

El Game Designer delimitará las normas del videojuego, es decir, diferenciará lo que sí que está permitido, de lo que no. Además, establecerá los distintos niveles de los que consta el videojuego. Este profesional, deberá ponerse en todas las tesituras posibles que el jugador pueda experimentar durante el juego para definir estos aspectos.

- **Crea los personajes**

Tendrá el poder de definir el número de jugadores, así como qué rol tiene cada uno, el guion (si lo hay) y sus motivaciones. Depende de la temática del videojuego, tendrá que pensar en cómo van a ser los personajes: humanos, animales, figuras mitológicas, etc. Participará en la elección del aspecto físico y su objetivo será que los personajes estén alineados con el argumento del videojuego.

- **Define el escenario**

Los diseñadores de videojuegos también son responsables de decidir cómo va a ser el entorno en el que se va a desarrollar la acción. Siempre ligado al argumento, puede ser un escenario natural, futurista, realista e incluso ser tan original que no

tenga un carácter definido. Su función también consistirá en pensar cuántas fases se divide el juego para cuantificar el número de escenarios que se necesitan.

- **Fija las recompensas**

Cuando un jugador hace bien una acción dentro del videojuego, suele obtener una recompensa. El Game Designer tendrá que pensar qué tipo de recompensas recibirán los usuarios: cuantificar la recompensa y clasificarlas por grupos, definir en qué momento las recibirá el usuario, cómo se cobrarán, para qué servirán, etc.

- **Elige las habilidades**

Este rol profesional, deberá pensar qué habilidades tiene cada personaje. Clasificar los diferentes personajes en distintas categorías según las habilidades que presentan suele ayudar en el proceso de creación de un videojuego.

- **Marca los patrones de interacción**

El Game Designer marcará los criterios sobre cómo interactúan los personajes entre sí y cuál es el nivel de interacción que el usuario tiene con el juego. Asimismo, se tendrán en cuenta aspectos como, por ejemplo, la sociabilización, es decir, si el usuario puede interactuar con otros usuarios reales y cómo debe efectuarse esta acción.

Storyboard

El Storyboard no es otra cosa que la realización de una secuencia de dibujos realizados en una plantilla, acompañados de textos breves que definen la estructura de la historia. Un proceso imprescindible en la fase de preproducción para el desarrollo de Videojuegos que permite previsualizar el resultado final de la producción.

La creación de un Storyboard para videojuegos es muy similar al que se emplea en otros sectores como publicidad, cine o televisión. Sin embargo, difiere en un elemento principal: A diferencia de otros sectores donde solo hay una línea de acción, normalmente establecida por el director del guión, el diseño de un videojuego requiere que el Storyboard cuente con diferentes líneas de acción, en función de los movimientos del jugador.

Su objetivo principal es hacer el videojuego inmersivo, estableciendo la interacción que el jugador tendrá con los diferentes niveles, objetivos y fases del juego.

En definitiva, el resultado final en la producción de un videojuego dependerá, en gran medida, de la planificación previa realizada a partir del Storyboard en la fase de diseño del juego. Además, su uso permite prevenir errores antes de que sean llevados a la pantalla y, a su vez, ahorrar costes de producción.

Motores Gráficos

No es posible desarrollar un videojuego sin un motor gráfico. Es por tanto una elección que cualquier programador de videojuegos ha tenido que hacer cada vez que se haya planteado un proyecto así. Dentro del mercado existen varias opciones, algunas más potentes y otras más flexibles. La elección depende en gran medida del tipo de desarrollo y también de las necesidades de los profesionales que se vayan a encargar de su conceptualización y diseño.

Dentro de todas estas opciones, hay varias que son totalmente gratuitas, con un gran rendimiento para desarrolladores independientes que quieren hacerse un hueco dentro de este sector. Estos son algunos ejemplos:

- Unity
- CryEngine
- Unreal Engine
- Game Maker Studio
- Godot engine, etc.

Además, existen otras alternativas no gratuitas:

- MT Framework
- Samaritan
- Titanium
- Rockstar Advanced Game Engine (RAGE)
- Frostbite, etc.

Lenguajes de Videojuegos

El lenguaje con el que están programados los videojuegos es un elemento fundamental para garantizar una experiencia de juego fluida. Parece que cuando jugamos a un videojuego solo nos fijamos en la parte visual, en su estética, porque es lo más espectacular, pero detrás de esa estética hay un arduo trabajo de programación que permite su funcionamiento.

- **C++**

Este lenguaje de programación es uno de los más utilizados en el sector por profesionales. Es un lenguaje popular en los títulos AAA, se utiliza en videojuegos para PlayStation y Xbox, y en juegos independientes. Se trata del lenguaje más compatible con la mayoría de los motores de juego y tiene un tiempo de ejecución bastante rápido. Por otro lado, permite a los desarrolladores tener un control amplio sobre el hardware, la gestión de la memoria y los gráficos, y, aunque al principio puede resultar complejo de utilizar, una vez te haces a él, podrás manejar cualquier otro lenguaje.

- **C Sharp (C#)**

C# es un lenguaje de programación muy popular, sobre todo en entornos Windows. Es un poco menos flexible y compatible que C++, pero algunos motores como Unity permiten programar con él y no está limitado a un determinado sistema operativo o plataforma; se pueden crear juegos para iOS, Android, Windows Play Station y Xbox. Es un lenguaje más fácil de aprender que el C++ y será una buena opción si estás empezando o si quieres que lo disfruten más personas, con independencia del dispositivo.

- **Java**

Se trata de un lenguaje frecuentemente utilizado y presenta muchas similitudes con C++. Su principal característica es la versatilidad, ya que se puede utilizar en todas las plataformas, dispone de gran cantidad de frameworks para el desarrollo 3D, ofrece módulos de código abierto y su modelo se puede actualizar constantemente. ¿El problema? Que se ejecuta dentro de su máquina virtual, y esto supone una pérdida de rendimiento.

- **JavaScript**

Este es uno de los lenguajes más utilizados en el desarrollo de videojuegos web y de navegador. La mayoría de motores de videojuegos son compatibles con JavaScript, y cuenta con múltiples frameworks para 3D y una gran variedad de bibliotecas. Además, algunos motores de videojuegos como Unity lo utilizan, por lo que podremos usarlo para crear todo tipo de scripts dentro del juego.

- **Python**

A pesar de no ser un lenguaje de programación exclusivo para la creación de videojuegos, Python es un lenguaje muy flexible y potente para esto. Su ejecución es mucho más simple que la de otros lenguajes (permite plasmar ideas complejas con pocas líneas de código), y su framework Pygame permite a los desarrolladores crear prototipos de sus videojuegos de manera rápida y sencilla, y funciona prácticamente en todas las plataformas y sistemas operativos.

Metodologías de desarrollo de videojuegos.

Las metodologías utilizadas para el desarrollo de videojuegos siguen principios ágiles por ser iterativas e incrementales, tener interacción frecuente con el cliente y ser flexibles ante los requerimientos cambiantes. Otra característica es que las decisiones se toman en base a la experiencia, sin existir un proceso definido ni técnicas específicas a seguir.

En promedio, cada proyecto lo realizan de tres a cuatro personas que cubren los roles de productor, programador, artista gráfico, diseñador de juego y artista sonoro. Las tareas de este último habitualmente son realizadas por empresas externas especializadas. La tendencia a utilizar metodologías ágiles para videojuegos tomó fuerza en los últimos años por existir varios casos de empresas en la industria que logran adaptar estas metodologías y además ser un tema actual en uno de los eventos principales como es la Game Developer Conference (GDC).

- **Metodología Sum**

La metodología SUM para videojuegos tiene como objetivo desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso para incrementar su eficacia y eficiencia. Pretende obtener resultados predecibles, administrar eficientemente los recursos y riesgos del proyecto, y lograr una alta productividad del equipo de desarrollo.

SUM fue concebida para que se adapte a equipos multidisciplinarios pequeños (de tres a siete integrantes que trabajan en un mismo lugar físico o estén distribuidos), y para proyectos cortos (menores a un año de duración) con alto grado de participación del cliente. La definición de la metodología se basa en el Software and Systems Process Engineering Metamodel Specification (SPEM), un meta-modelo para describir procesos y metodología

Proceso de diseño de interfaces de videojuegos en 2D y 3D.

La interfaz de usuario en un videojuego es el punto de interacción entre el jugador y el juego. Su objetivo fundamental es el de brindar la información necesaria para que el usuario pueda hacer todo lo que el juego le propone de manera totalmente fluida. Un buen diseño de UI guía de manera directa o intuitiva para que el jugador pueda recorrer el mundo de tu videojuego de forma correcta.

Cosas a tener en cuenta a la hora de diseñar la interfaz de un videojuego:

- **Entorno/Plataforma.** Lo primero a tener en cuenta es dónde se va a jugar el juego que estás diseñando. Debes tener en cuenta las posibilidades y limitaciones que te ofrece la plataforma. No es lo mismo hacer juegos para smartphones que para una consola o que para un PC.
- **Contenido.** Un buen diseño de UI proporciona al jugador toda la información necesaria para que pueda interactuar con el juego y que todo sea fluido.
- **Diseño Visual.** Los videojuegos, casi siempre, entran por los ojos. Un apartado visual feo o denso en la interfaz del juego puede resultar contraproducente y sacar al jugador de la experiencia inmersiva que quieres proporcionarle. Debes definir el estilo de arte.
- **Arquitectura de la información.** Definir qué elementos son de mayor o menor importancia para el usuario y organizarlos de tal forma que todo resulte en un diseño de interfaz coherente y relevante.

Estilos de diseño de la interfaz de un videojuego

Diegéticas, No Diegéticas, Espaciales y Meta. Estas son algunas de las clasificaciones que te vas a encontrar para las diferentes interfaces que se diseñan para un videojuego.

- **Diegéticas**

Igual que sucede con la música en el cine, cuando hablamos de que una interfaz de un videojuego es diegética, quiere decir que está incluida dentro del mundo del juego. Es decir, que se trata de un tipo de interfaz que puede ser vista, escuchada y tocada por los personajes del juego. Forman parte de la propia narrativa del juego.

- **No diegéticas**

Cuando hablamos de diseño de interfaces no diegéticas o extra diegéticas, nos estamos refiriendo a los elementos que se muestran fuera del mundo del juego y que solo son visibles y audibles para el jugador.

- **Meta**

Se trata de una parte del diseño de interfaz muy curiosa ya que están dentro de la historia del juego, pero no lo están en el espacio de juego. Se trata de un diseño que pretende hacer que la experiencia sea aún más inmersiva, por ejemplo, las gotas de sangre en la pantalla o los vidrios rotos.

Desarrollo de prototipos de videojuegos

- **Alpha**

Se podría decir que la versión Alfa de un videojuego es la versión más primitiva del mismo, pues es la primera versión del juego que es funcional. Sin embargo, esta versión de desarrollo suele ser muy inestable, por lo que suele pasarse la versión Alfa a varios probadores de videojuegos, los conocidos como testers, para que detecten los errores que tiene esta primera versión del juego y reportarlos a los desarrolladores. Una vez estos reciban el feedback de los testers acerca de la versión Alpha podrán trabajar en solucionarlos y mejorar el videojuego.

- **Beta**

Una vez el videojuego es probado y corregido en su versión Alfa, surge la primera versión completa del videojuego: la Beta. Esta es a su vez la última versión de prueba que los desarrolladores pasar a los testers para que detecten los últimos errores antes de sacar la versión definitiva, que puede ser una Beta cerrada o abierta, en cuyo caso es accesible al público y este también puede reportar posibles mejoras.

Integración de motores de videojuegos con programación visual

Se integran de diferentes maneras, como podrían serlo por medio de API's, con el Buffer, OpenGL, etc

- **API:** Es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.
- **API GRÁFICO:** Es un API que ofrece prestaciones para desarrollos gráficos digitales a programadores y diseñadores. En el ámbito de los videojuegos, los API gráficos más conocidos son Direct3D y OpenGL.
- **BUFFER:** Es un espacio de memoria, en el que se almacenan datos para evitar que el recurso que los requiere, ya sea hardware o software, se quede en algún momento sin datos.
- **DirectX:** Es una colección de APIs creadas para facilitar tareas relacionadas con la programación de juegos en la plataforma Microsoft Windows. Las APIs son: Direct3D, DirectInput, DirectSound, DirectGraphics, DirectMusic, DirectPlay.
- **DIRECT3D:** El objetivo de esta API es facilitar el manejo y trazado de entidades gráficas elementales, como líneas, polígonos y texturas, en cualquier aplicación que despliegue gráficos en 3D, así como efectuar de forma transparente transformaciones geométricas sobre dichas entidades. Direct3D provee también una interfaz transparente con el hardware de aceleración gráfica
- **OPENGL:** Es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.
- **PÍXELES:** Es la menor unidad en la que se descompone una imagen digital, ya sea una fotografía, un fotograma de vídeo o un gráfico.

Transición narrativa y lenguaje visual de videojuegos.

Los videojuegos son otra forma de contar historias con la peculiaridad de que el receptor tiene la posibilidad de tomar las decisiones que determinan el camino del protagonista. La interactividad es muy grande pero el jugador no tiene libertad total para determinar el futuro de su personaje ya que las acciones están marcadas por el objetivo final de la partida.

Tipos de Narrativa

- **Narrativa lineal:** se trata de una línea recta que no se separa de su objetivo. Se caracteriza también por no ofrecer capacidad alguna de toma de decisiones sobre ningún aspecto de la narración de la historia. Dentro de esta línea hay un segundo tipo llamado string of pearls (collar de perlas en español) que es exclusivo de videojuegos. Son narraciones lineales interrumpidas por pequeños momentos de decisión del jugador.
- **Narrativa ramificada:** siendo una de las más importantes en los videojuegos, esta narrativa (ramificada o de branching) es conocida por ofrecer al jugador caminos alternativos. Su denominación (branching) viene por su parecido a un árbol, siendo el tronco el argumento principal y las ramas sus posibles caminos alternativos. El contenido y su gameplay se verán afectados por las decisiones del jugador. Igual que con la narrativa lineal, en esta también podemos encontrar otros tipos exclusivos de narraciones, como es el caso de la narrativa binaria o limitada, ofreciendo al jugador opciones de tipo A/B.
- **Narrativa en embudo:** en el funneling la historia se presenta separada en varias opciones. Sin embargo, todas esas escisiones terminan siempre en cuellos de botella, dando la posibilidad al jugador de explorar en la historia, pero avanzando y progresando por un único arco narrativo.
- **Narrativa no lineal:** con este tipo de narrativa, el orden cronológico no está presente, a su manera. En la narración multinivel, que es característica de los juegos tipo MMO's, los bloques narrativos no guardan relación entre sí

Lenguaje visual de videojuegos.

Una vez que hemos definido el aspecto visual y ambientación de nuestro juego tendremos que ver qué características del diseño pueden tener un impacto en los gráficos. Y es que el arte, además de tener que ser interesante y atractivo también debe apoyar al diseño del juego, adaptándose y dando soluciones a las necesidades del gameplay, potenciando las mecánicas de juego y siendo un fin para comunicarse con el jugador. Por lo tanto el diseñador deberá prestarle especial atención y tener en cuenta también como va a ser el apartado gráfico desde el principio, trabajando junto con el equipo de arte para que ambas áreas se complementen.

El proceso de desarrollo de videojuego acorde a los elementos de programación visual.

Existen distintos factores que influyen en el proceso de creación de un videojuego por lo cual no existe una receta definitiva. Conocer dichos factores es vital para entender y medir el proyecto a desarrollar, comprender las limitaciones y asegurarnos de una producción exitosa.

- **Género.** Sea un Role Playing Game (RPG o juego de rol por sus siglas en inglés), First Person Shooter (FPS o juego de acción en primera persona), juego de plataformas, puzzles (acertijos) o educativo, etc. El tipo de juego a desarrollar marca de forma importante los fundamentos de producción requeridos.
- **Portabilidad.** Se refiere a las plataformas servicios y sistemas destino donde el videojuego estará disponible: PC, Mac, PlayStation, Xbox, Nintendo, dispositivos móviles, navegadores de Internet, dispositivos de VR etc. Es importante conocer las limitaciones técnicas y las restricciones para el diseño de juego de cada una de ellas; videojuegos que no respeten las respectivas guías de desarrollo corren el riesgo de no ser publicados o incluso prohibidos en ciertas plataformas.

- **Diseño de Juego (Game design).** Se refiere al diseño del videojuego y sus reglas el cual responde preguntas como: ¿De qué se trata el juego? ¿cuál es su lógica? ¿cómo interactúa el usuario? ¿cuáles son los objetivos? ¿cuál es la progresión del personaje? Son el tipo de cuestionamientos que definen la lógica y el funcionamiento de un videojuego y que posiblemente requieran un desarrollo particular.
- **Presupuesto.** El recurso financiero disponible para trabajar el videojuego. Es común que los desarrolladores obtienen inversión parcial o completa de publishers, inversionistas privados o gobierno.
- **Guía artística visual.** Es la línea que dicta como se verá el videojuego y todos sus elementos gráficos y bajo la cual se regirá toda la producción. Ya sea para el desarrollo de un juego hiperrealista AAA o el desarrollo de un juego casual estilizado para niños, la guía de arte debe acomodar las necesidades y expectativas para cada mercado meta.
- **Equipo de desarrollo.** Tanto el presupuesto como el equipo de desarrollo son cruciales elementos para una producción, la falta de alguno de ellos bloquea prácticamente el buen funcionamiento del proyecto



Referencias

- A. (2019, 18 junio). *El Storyboard en Diseño de Videojuegos*. Arteneo.
<https://www.arteneo.com/blog/storyboard-videojuegos-escuela-madrid/#:%7E:text=El%20Storyboard%20no%20es%20otra,la%20estructura%20de%20la%20historia.>
- BBVA API_Market. (1970, 1 enero). *Los mejores motores gráficos de videojuegos (I): soluciones de código abierto*. <https://www.bbvaapimarket.com/es/mundo-api/los-mejores-motores-graficos-de-videojuegos-i-soluciones-de-codigo-abierto/#:%7E:text=B%C3%A1sicamente%20son%20tres%3A%20Unity%2C%20Cryengine,Anvil%20Engine%20y%204A%20Engine.>
- Capdevila, A. G. (2022, 18 abril). *Los 6 mejores lenguajes de programación para videojuegos*. Profile Software Services. <https://profile.es/blog/lenguajes-programacion-videojuegos/>
- Marrero, L. (2022, 3 marzo). *Storyboard ¿qué es y cómo se hace?* SOYDECINE.COM.
<https://soydecine.com/storyboard-que-es-y-como-se-hace/>
- School, T. (2022, 29 julio). *El diseño de interfaz de un videojuego*. Tokio School.
<https://www.tokioschool.com/noticias/diseno-interfaz-videojuego/>
- Varela, R., & Varela, R. (2019, 10 abril). *Alfa / ¿Qué significa Alfa?* Geekno.
<https://www.geekno.com/glosario/alfa>