



UNIVERSIDAD POLITÉCNICA
DE TECÁMAC

TAREA PROGRAMACIÓN VISUAL

Materia: PROGRAMACION
VISUAL

Profesor: EMMANUEL
TORRES SERVIN

Alumnos:

-1321124244 Lourdes
Monserrat Guzmán Sánchez
-1321124275 Yareli Irais Garcia
Alvarado
-1321124234 Rafael Alcantar
Chavez

Grupo: 4322IS

Carrera: Ingeniería en
Software

INTRODUCCION

Los dispositivos móviles se están proliferando en la empresa a un ritmo exponencial.

Ganar visibilidad y control sobre los dispositivos corporativos y los personales que se conectan a la red de su empresa y tienen acceso a recursos corporativos. La solución de la administración de dispositivos móviles de AirWatch (MDM) le permite administrar las implementaciones a gran escala de los dispositivos móviles. Nuestra solución le brinda la habilidad de inscribir dispositivos rápidamente en su entorno empresarial, configurar y actualizar los ajustes del dispositivo over-the-air, imponer políticas de seguridad y conformidad, acceso seguro móvil a recursos corporativos y eliminar o bloquear dispositivos remotamente.

TAREA 1

Realiza un resumen del Análisis de la programación visual

programación orientada a objetos.

La programación orientada a objetos es un paradigma de programación que utiliza objetos y sus interacciones para diseñar aplicaciones y programas informáticos. Se basa en varias técnicas que incluyen herencia, abstracción, polimorfismo y encapsulamiento.

Características y aplicaciones de eventos.

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

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.

La programación orientada a eventos no es como tal un tipo de lenguaje de programación es mas 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.

Los programas orientados a eventos se convirtieron ya en algo muy común, algunos ejemplos notorios son: el procesamiento de textos, herramientas de dibujo, hojas de cálculo, etc.

componentes y métodos visuales y no visuales.

Los componentes visuales tienen una representación gráfica en diseño y ejecución tales como botones, barras de scroll y cuadros de edición y los componentes no visuales tienen como representación los temporizadores y cuadros de diálogo que no son visibles en la fase del diseño.

Los métodos son funciones asociadas al componente que pueden invocarse para que el componente realice distintas acciones.

Por ejemplo, todos los componentes visuales tienen un método llamado Show() para mostrarlos y otro llamado Hide() para ocultarlos.

Procesos de desarrollo visual en proyectos distribuidos y de escritorio.

El desarrollo de software moderno como herramienta de programación visual es catalogada como una solución fácil de usar para expertos que no están especializados en codificación.

El diseño visual simple de imágenes y bloques hace que sea fácil de entender, diseñar e interpretar para el usuario promedio y en lugar de ver líneas de código difíciles de descifrar, las personas pueden entender y explicar lógicamente conceptos complejos utilizando este lenguaje de programación visual.

Debido a su relativa simplicidad, la programación visual es una forma ideal de familiarizar a los usuarios con la programación. Este lenguaje de programación es más grande y por lo tanto ocupa más espacio en la computadora, lo que puede ralentizar el dispositivo debido a la cantidad de memoria que necesita en el disco.

Requerimientos visuales de proyectos distribuidos y de escritorio.

Definir los objetivos del proyecto.

Seleccione el resultado.

Definir riesgos y límites.

Usar ayudas visuales para mejorar la estrategia del proyecto.

Cálculo de presupuesto.

Hacer un plan de contingencia.

Documento de puntos de referencia.

Herramientas y lenguajes de programación visual.

Razor: La sintaxis Razor de ASP.NET le permite alinear C# directamente en sus vistas. Además, tiene acceso completo al modelo actual y a los tipos de .NET Framework.

TypeScript: Es un superconjunto de JavaScript que se compila en JavaScript sin formato y que le permite crear código más escalable.

JavaScript: JavaScript en Visual Studio es compatible con EcmaScript 6 y tiene el motor IntelliSense más avanzado. JavaScript es un lenguaje de primera clase en Visual Studio, puede usar la mayoría de las ayudas de edición estándar.

CSS: Este lenguaje de diseño gráfico define y crea la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML.

Python: Este es un lenguaje de programación dinámico y de alto nivel orientado a objetos, que a menudo se usa para el desarrollo rápido de aplicaciones.

C++: Un lenguaje de programación que se utiliza para crear aplicaciones para Windows, Linux, iOS y Android.

Visual Basic: Este lenguaje de programación es totalmente orientado a objetos, muy fácil de utilizar y está dirigido a la plataforma NET, y permite crear aplicaciones de Windows sin tantos problemas.

BIBLIOGRAFIA

<https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>

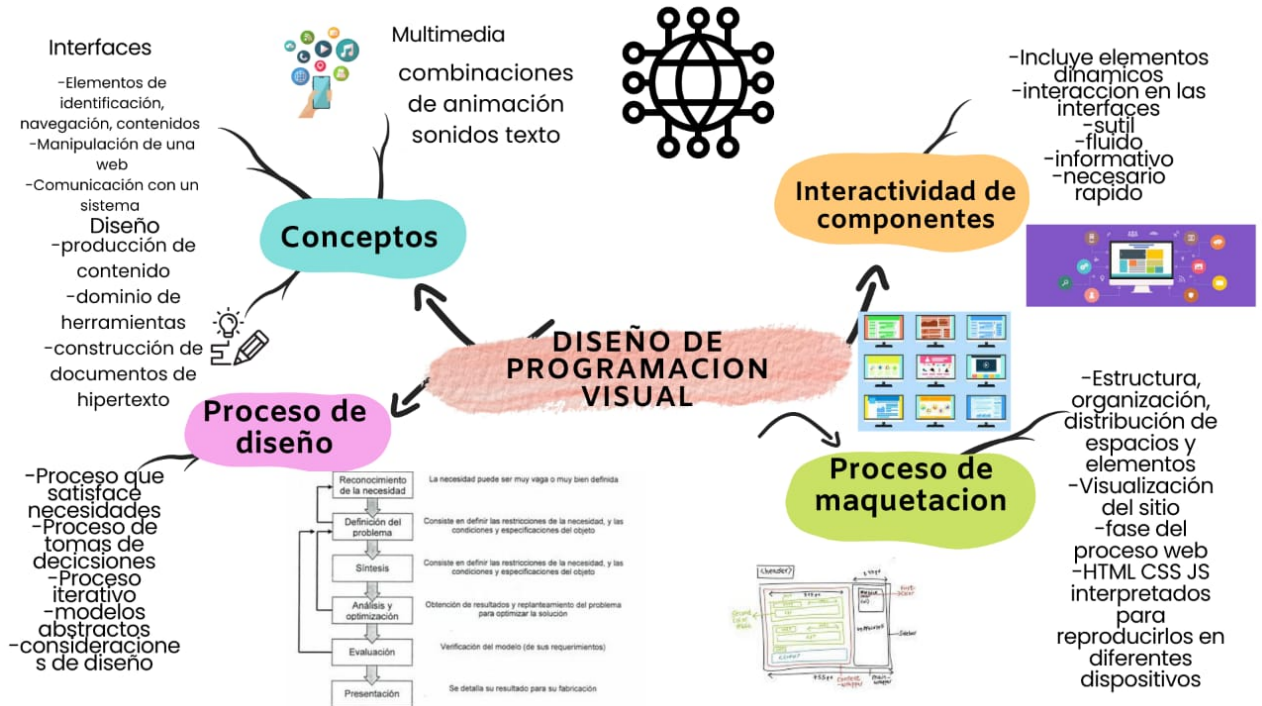
<https://www.lifeder.com/programacion-orientada-a-eventos/>

[https://elvex.ugr.es/decsai/builder/intro/3.html#:~:text=Un%20componente%20es%20visual%20cuando,de%20die%C3%B1o%2D%2C%20etc\).](https://elvex.ugr.es/decsai/builder/intro/3.html#:~:text=Un%20componente%20es%20visual%20cuando,de%20die%C3%B1o%2D%2C%20etc).)

<https://visualstudio.microsoft.com/es/vs/features/web/languages/>

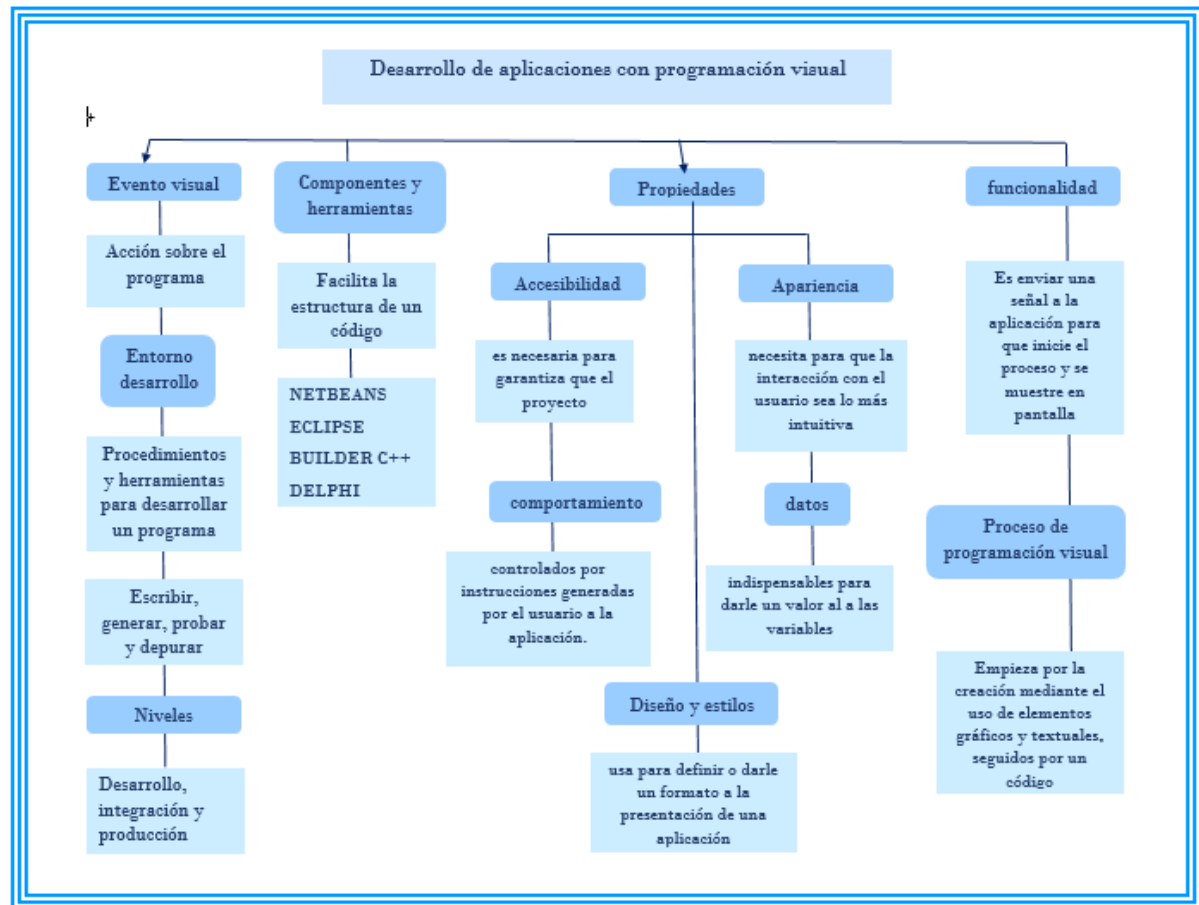
TAREA 2

Realiza un mapa mental del tema Diseño de programación visual



TAREA 3

Realiza un mapa conceptual del tema Desarrollo de aplicaciones con programación visual



TAREA 4

Realiza una investigación

Introducción

Esta investigación tiene como principal tema los videojuegos explicando a detalle todo el desarrollo de un videojuego, de como llega a nosotros ya desarrollado y principalmente solo viendo lo que viene siendo la parte visual por que no logramos ver todo el desarrollo, el análisis y las dificultades que se tuvieron que enfrentar para que llegue a nosotros un videojuego desarrollado y muy completo.

Todo esto con ayuda de los motores de videojuegos que también son explicados y redactados para que puedan conocer lo que son y para qué son, claramente también les informamos cuales son y de que sirven a la hora del desarrollo de un videojuego, ya que son la base fundamental para poder lograr los diseños gráficos del videojuego, algo bastante interesante.

También les explicamos que lenguajes de programación pueden servir para el desarrollo de un videojuego, en que les pueden ayudar con una breve investigación que se realizó para conocer sus ventajas y obviamente la facilidad, todo esto para conocer un poco del tema

Concepto y tipos de game designer y storyboard

Designer: es el profesional encargado de diseñar todos los elementos que componen el juego, desde el concepto, las mecánicas o los niveles, entre otros. No es estrictamente necesario que sea programador, ni ser un diseñador gráfico o artista conceptual. Su labor va mucho más allá.

se ha incrementado en los últimos años gracias al crecimiento de la industria de los videojuegos. Esta industria es una de las que más ingresos genera a nivel mundial. Desde el punto de vista del consumo y la generación de empleo, las cifras del mercado indican que, a corto plazo, podría superar a la industria cinematográfica.

Así mismo este sector contribuye a la empleabilidad: en 2019 hubo 7.935 trabajadores en esta industria y se prevé que en el año 2022 se alcancen los casi 11.000 empleados. Cataluña lidera la industria, el 28% de las empresas censadas en España tiene su sede en territorio catalán. Algunas de las más conocidas situadas en Barcelona son: Ubisoft, Gamelot, King, Scopely o Riot Games. La Comunidad de Madrid es el otro epicentro que reúne más empresas de videojuegos en territorio nacional. Mercury Steam es una de las que genera más proyección internacional y empleo. Igualmente, la capital española es la mayor sede de PlayStation Games Camp, que apoya el desarrollo de videojuegos independientes de Sony PlayStation.

Si hablamos de los perfiles profesionales que ocupan los empleados en las empresas españolas, nos encontramos que el pódium lo conforman: los programadores, los profesionales relacionados con el arte y los Game Designers, consecutivamente. Estos últimos, tienen una gran responsabilidad vinculada al éxito de un videojuego. **¿Sabes en qué consiste su trabajo y qué funciones tiene? ¡Vamos a contarte todo lo que necesitas saber sobre el rol profesional del Game Designer!**

- 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: Se trata de un conjunto de imágenes que forman una guía visual mediante la cual se pueden previsualizar las escenas que van a grabarse. Muchas veces usamos el término guión gráfico para referirnos a los storyboards. Y es que gracias a él podemos hacernos una ligera idea de lo que veremos luego en la pantalla.

En el cine, el storyboard nos cuenta las ideas principales de cada escena. Por lo que al tratarse de una guía visual meramente orientativa el equipo de rodaje puede hacer diferentes cambios a la hora de grabar o montar la escena, siempre y cuando el director lo vea conveniente.

Para hacer un storyboard hay que contar previamente con un guión. Y es que a la hora de elaborarlo partiremos siempre de un texto que funcione como una guía que debemos seguir. Una vez que tengamos claro lo que queremos plasmar en el Storyboard podemos hacerlo con algo tan simple como un lápiz y una hoja en blanco.

A la hora de crear el storyboard no es necesario ser muy detallistas. Y es que el storyboard se compone de bocetos rápidos y simples de forma que el resto del equipo de rodaje pueda entenderlo con facilidad.

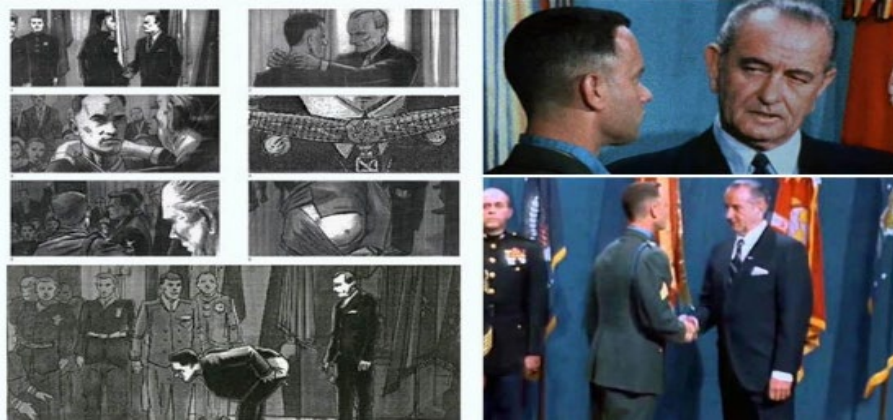
Además, podemos añadir una breve descripción debajo de cada plano en la que incluir diferentes indicaciones, frases de los personajes, movimientos de cámara, etc.

Un Storyboard se compone de diferentes elementos:

- La información del guión, la cual define en qué punto nos encontramos (escena, secuencia, lugar en el que transcurre...)
- Los cuadros, los cuales se dividen en grupos de tres, seis, o nueve fotogramas. En el interior de estos cuadros tenemos los dibujos que nos muestran lo que va sucediendo. En este sentido, el storyboard y el comic pueden llegar a parecerse bastante.
- La descripción, que suele ir debajo de los cuadros y nos describe todo lo referente a ese plano (qué está ocurriendo, movimientos y ángulos de cámara, iluminación, etc.)
- Por último tenemos un apartado en la parte inferior destinado al campo de observaciones, por si hay que hacer cualquier tipo de comentario para que el resto del equipo lo tenga en cuenta.

Da igual si quieres hacer un Storyboard para Stop Motion, un Storyboard de animación o uno para películas con actores de carne y hueso. El proceso suele ser el mismo en casi todos los casos.

[Ejemplo storyboard](#)



Tipos y características de motores de videojuegos y lenguajes de videojuegos

Tipos

Unity, Cryengine y Unreal Development Kit (UDK). no gratuitas: Unreal Engine, MT Framework, Samaritan, Titanium, Rockstar Advanced Game Engine (RAGE), Frostbite, Havok, Anvil Engine y 4A Engine.

Características

Unity es un motor gráfico gratuito, flexible, con una gran gama de recursos. Es una opción muy interesante para los desarrolladores jóvenes porque permite hacer buenos desarrollos sin coste, es sencillo de aprender y de usar y, además, al ser multiplataforma, cualquier proyecto se puede exportar a sistemas operativos móviles como Android, iOS o Windows Phone sin problemas. Dispone de todo lo necesario para diseñar proyectos en 2D y 3D.

Cryengine es uno de los motores gráficos más usados para el desarrollo de videojuegos para consolas, PC y dispositivos móviles, ya sea PlayStation 4, Xbox One, Wii U, sistemas operativos Windows, Mac OS X, Linux, iOS o Android. Cryengine aporta una gran gráfica, basado en una óptima solución de renderizado en tiempo real y una tecnología en 3D muy eficiente.

Unreal Development Kit es la versión gratuita de Unreal Engine. Este motor gráfico para el desarrollo de videojuegos utiliza UnrealScript como lenguaje de programación y es compatible con Windows, sistema operativo de Microsoft. Al igual que los otros dos motores gráficos permite el diseño de imágenes en 2D y 3D. En general es una solución usada por pequeños desarrolladores.

Lenguajes de Videojuegos

Tipos

C++

C#

Java

Características

C++

Es uno de los lenguajes de programación más utilizados por los profesionales del sector. Se utiliza para algunas de las principales plataformas de hardware como PlayStation y Xbox, siendo uno de los más compatibles con la mayor parte de motores de juego. Unido a un rápido tiempo de ejecución, C++ permite a los desarrolladores tener un amplio control sobre el hardware. Dominar este tipo de programación nos facilitará en gran parte la enseñanza de otros lenguajes.

C#

Uno de los más extendidos en el entorno de Windows y de los favoritos por programadores de videojuegos. Siendo más fácil de aprender que el C++, también se caracteriza por ser algo más limitado que el anterior. No obstante, se pueden crear juegos con este lenguaje para múltiples plataformas (PlayStation, Xbox, Android, iOS, Windows). Magnífica opción para principiantes.

Java

Con cierta similitud a C++, y siendo uno de los más utilizados mundialmente, con este lenguaje se puede programar en la mayor parte de plataformas. Entre sus 'pros' se encuentra la gran cantidad de frameworks para el desarrollo 3D, haciendo que podamos hacer prácticamente de todo con este lenguaje. Uno de sus 'cons': se ejecuta dentro de su máquina virtual, llevando esto a una pérdida de rendimiento.

Metodologías de desarrollo de videojuegos

La metodología 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 (de tres a siete integrantes que trabajan en un mismo lugar físico o estén distribuidos), y para proyectos cortos con alto grado de participación del cliente.

El ciclo de vida se divide en fases iterativas e incrementales que se ejecutan en forma secuencial con excepción de la fase de gestión de riesgos que se realiza durante todo el proyecto. Las cinco fases secuenciales son: concepto, planificación, elaboración, beta y cierre.

Proceso de diseño de interfaces de videojuegos en 2d y 3d

La interfaz de un videojuego, la UI, es lo primero que nos atrae como

jugadores en una consola. Es nuestra puerta de entrada, la invitación que nos hacen los diseñadores para entrar al mundo que han construido.

Diseñar una buena UI es todo un desafío. Muchas veces, los estudios descuidan esta faceta porque no tienen tiempo o recursos suficientes para dedicarle, pero un mal diseño de interfaz de un videojuego es lastrar la experiencia del usuario. Un buen diseño nos permitirá empezar en el juego de una manera sencilla e intuitiva.

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.

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. A continuación vamos a hablarte un poco de cada una de ellas, usando algunos ejemplos de juegos que seguro que conoces y que emplean alguna de estas tipologías para su diseño de interfaz. ¡Vamos a ello!

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. Aunque las cosas van cambiando en este sentido, se trata del diseño de interfaz más ampliamente usado en los videojuegos.

Espaciales

Cuando hablamos de un diseño de interfaz espacial estamos ante un punto medio en los anteriores conceptos. Son elementos que están integrados en la acción del juego, pero los personajes no tienen constancia de ellos. A veces puede cortar la narrativa del juego, como sucede, por ejemplo, con los **elementos de un tutorial**.

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

Concepto, tipos y características de los motores de videojuego.

Concepto

Un motor de videojuego es un término que hace referencia a una serie de librerías de programación que permiten el diseño, la creación y la representación de un videojuego.

El aspecto más destacado a la hora de elegir un motor de videojuegos entre todos los disponibles que hay en el mercado son las capacidades gráficas, ya que son las encargadas de mostrar las imágenes 2D y 3D en pantalla, así como calcular algunos aspectos como los polígonos, la iluminación, las texturas ... Otras características para tener en cuenta a la hora de la elección son la facilidad de aprender a usar el motor de videojuegos y la facilidad para exportar el juego a diferentes plataformas.

El motor de físicas

El motor de físicas es el que hace posible aplicar aproximaciones físicas a los videojuegos para que tengan una sensación más realista en la interacción de los objetos con el entorno. En otras palabras, es el encargado de realizar los cálculos necesarios para que un objeto simule tener atributos físicos como peso, volumen, aceleración, gravedad ...

El motor de sonido

Los sonidos y la banda sonora de un videojuego es también una parte muy importante. El motor de sonidos es el encargado de cargar pistas, modificar su tasa de bits, quitarlas de reproducción, sincronizarlas entre otras cosas.

El scripting

Todos los motores de videojuegos tienen un lenguaje de programación que permite implementar el funcionamiento de los personajes y objetos que forman parte del videojuego.

Dentro de las diferentes opciones de motores de videojuegos podemos distinguirlos en populares y motores propietarios o privados que son los creados por empresas importantes de videojuegos para diseñar sus títulos más populares.

Más Utilizados.

Unreal Engine: Fue creado por Epic Games en 1998. En 2012 se presentó Unreal Engine 4, una nueva versión del motor. Entre las empresas que lo utilizan se encuentran Electronic Arts. y Ubisoft. Utiliza el lenguaje de programación C++.

Unity 3D: Se trata de una de las innovaciones más importantes creadas por la comunidad científica y de videojuegos y permite jugar a complejos videojuegos en 3D sin necesidad de instalarlos en el ordenador. Los videojuegos creados con el motor Unity 3D se pueden jugar en un navegador con el reproductor Unity Web Player, eliminando la necesidad de instalar el videojuego.

Frostbite Engine: Este motor para videojuegos creado por Digital Ilusionas CE se utiliza para crear videojuegos de acción en primera persona. Se presentó principalmente para la serie de videojuegos Battlefield. Ha jugado un papel fundamental en prácticamente todos los videojuegos de EA. La nueva versión del motor Frostbite Engine es Frostbite 3.

Decima Engine: Alberga herramientas y características para crear inteligencia artificial, física, lógica y mundos en el desarrollo, así como compatibilidad con 4K y HDR.

Luminous Studio: Es un motor de videojuegos multiplataforma desarrollado y usado internamente por Square Enix. Con este motor se desarrolla el juego Final Fantasy.

Tipos de Motores.

Unity, Cryengine y Unreal Development Kit (UDK). no gratuitas: Unreal Engine, MT Framework, Samaritan, Titanium, Rockstar Advanced Game Engine (RAGE), Frostbite, Havok, Anvil Engine y 4A Engine.

Características

Unity es un motor gráfico gratuito, flexible, con una gran gama de recursos. Es una opción muy interesante para los desarrolladores jóvenes porque permite hacer buenos desarrollos sin coste, es sencillo de aprender y de usar y, además, al ser multiplataforma, cualquier proyecto **se puede exportar a sistemas operativos móviles** como Android, iOS o Windows Phone sin problemas. Dispone de todo lo necesario para diseñar **proyectos en 2D y 3D**.

Cryengine es uno de los motores gráficos más usados para el desarrollo de videojuegos para consolas, PC y dispositivos móviles, ya sea PlayStation 4, Xbox One, Wii U, sistemas operativos Windows, Mac OS X, Linux, iOS o Android. Cryengine aporta una gran gráfica, basado en **una óptima solución de renderizado en tiempo real y una tecnología en 3D muy eficiente**.

Unreal Development Kit es la versión gratuita de Unreal Engine. Este motor gráfico para el desarrollo de videojuegos [utiliza UnrealScript como lenguaje de programación](#) y es compatible con Windows, sistema operativo de Microsoft. Al igual que los otros dos motores gráficos permite el diseño de imágenes en 2D y 3D. En general es una solución usada por pequeños desarrolladores.

Integración de motores de videojuegos con programación visual de acuerdo con los requerimientos del videojuego.

Los motores de videojuegos en programación visual es decir los que son mas de diseño son los más populares y conocidos. Mientras que los videojuegos de estrategia en tiempo real y los simuladores de vuelo apuestan en la calidad gráfica, los motores de videojuegos en primera persona apuestan más a la acción, siendo DOOM el más conocido; y son importantes porque marcan un inicio histórico en la creación y el uso de motores de videojuegos.

Las mejoras de GPU como Shader Model 3 y Shader Model 4, posibilitó mejoras a nivel gráfico. A partir del 2009, se anunciaron dos importantes avances en los principales motores de videojuegos existentes hasta la fecha: id Tech 5 (que fue usado por primera vez en Rage, y cuenta con una nueva tecnología llamada texturización virtual) y CryEngine 3, que fue utilizado para desarrollar el videojuego Crysis 2.

Pocas empresas han discutido qué planes tienen programados para el futuro en cuanto al surgimiento de nuevos motores. Sin embargo, el id Tech 6, eventual sucesor del id Tech 5, es una excepción. La información preliminar sobre este motor (que todavía está en fase inicial de desarrollo) sugiere que id Software está investigando cómo poder integrar la tecnología trazado de rayos y la rasterización gráfica.⁴⁹ Se ejecutará a través de una renderización geométrica representada por vóxeles (en lugar de triángulos) almacenados en un árbol octal. John Carmack afirma que este procedimiento será una manera más eficiente para almacenar los datos en 2D, así como los datos geométricos en 3D, al no tener problemas de almacenamiento

Transición narrativa y lenguaje visual de videojuego

Transacción Narrativa

la narrativa se refiere a la manera de contar esa historia, a la experiencia que lleva al jugador a sumergirse en la narración de los hechos.

Principalmente existen dos tipos de narrativas: **lineales y no lineales**. Mientras que las primeras se caracterizan por seguir un orden cronológico de los acontecimientos, las segundas no.

- 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. Aquí entrarían juegos tan

famosos como la saga Uncharted, The Legend of Zelda: Ocarina of Time, o The Last of Us. 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 (Quake, Call of Duty).

- 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 (Beyond: Two Souls, The Secret of Monkey Island). Este poder de decisión otorga al jugador momentos que modifican la historia, proporcionando la sensación de que esa toma de decisiones supone un impacto en el progreso de esta.
- 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 (The Witcher 2, Far Cry 2). Una variación de esta la comprendería la narrativa de caminos críticos, en la que el videojuego contaría con un solo camino, pero en la que se van encontrando ramificaciones a través de misiones secundarias, que deben ser simples para que el jugador no dé muchas vueltas. En esta categoría entran juegos como Grand Theft Auto V o Fallout 4.
- 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, pero sí vale

para que el jugador aúne el universo y el tema del juego. Juegos como League of Legends o World of Warcraft son ejemplo de ello.

lenguaje visual de videojuego

Programar un videojuego es algo muy atractivo para los entusiastas de la informática y los juegos. Sin embargo, para empezar a crear un videojuego, es necesario empezar por aprender todas las reglas que regulan un lenguaje informático.

Para comenzar a desarrollar un videojuego, puedes elegir los lenguajes C, Python y Java. Estos dos últimos son los más fáciles si acabas de empezar a programar. Por otro lado, estos lenguajes tienen ciertas limitaciones en lo que se refiere a las posibilidades de desarrollo y, por lo tanto, no podrás salir de los juegos básicos. Si tu deseo es superarte a ti mismo, te recomendamos que comiences a aprender el lenguaje C y luego el C++.

las aplicaciones de diseño gráfico han progresado notoriamente, igual que la "expertos" de los diseñadores. Capaces de crear impresionantes copias de la realidad, lo son también de "realidades alternas", como ha sido extraordinariamente exhibido en la película Avatar. Fuera del cine de fantasía, los videojuegos son campo de trabajo importante para los grafistas expertos.

Explicar el proceso de desarrollo de videojuego acorde a los elementos de programación visual.

El desarrollo se lleva a cabo por un desarrollador de videojuegos, que puede variar de una persona a un gran negocio. Los videojuegos comerciales para PC y videoconsolas son normalmente financiados por una distribuidora y tardan varios años en desarrollarse. Los videojuegos independientes pueden tomar menos tiempo y pueden ser producidos a bajo precio por individuos y pequeños desarrolladores. La industria de videojuegos independientes ha visto un aumento en los

últimos años con el crecimiento de los nuevos sistemas de distribución en línea y el mercado de juegos para móviles.

Para que el creador haga esto en el desarrollo de un videojuego generalmente hace el siguiente proceso:

- Concepción de la idea del videojuego
- Diseño
- Planificación
- Preproducción
- Producción
- Pruebas
- Mantenimiento

El proceso es similar a la creación de software en general, aunque difiere en la gran cantidad de aportes creativos (música, historia, diseño de personajes, niveles, etc) necesarios. El desarrollo también varía en función de la plataforma objetivo (PC, móviles, consolas), el género (estrategia en tiempo real, RPG, aventura gráfica, plataformas, etc) y la forma de visualización (2D, 2.5D y 3D).

Cabe mencionar que el diseño de juegos es usualmente considerado un proceso de creación iterativo, esto quiere decir que los diseñadores tendrán que pasar por cada uno de estos pasos repetidas veces (cambiando y mejorando aspectos) hasta que consideren que el resultado sea el mejor.

Concepción

Al tener una idea inicial en mente comienza esta etapa en la cual deberán plantear los aspectos fundamentales que conformarán el videojuego, entre los que se encuentran:

Diseño

En esta fase se detallan todos los elementos que compondrán el juego, dando una idea clara a todos los miembros del grupo desarrollador acerca de cómo son.

Planificación

En esta fase se identifican las tareas necesarias para la ejecución del videojuego y se reparten entre los distintos componentes del equipo desarrollador. También se fijan plazos para la ejecución de dichas tareas y reuniones clave, con la ayuda de herramientas de diagramación de actividades como [GANTT](#) y [PERT](#).

Preproducción

Durante la etapa de preproducción se le asigna el proyecto a un pequeño equipo, con la finalidad de verificar la factibilidad de la idea.

Este equipo trabajará para crear un nivel o ambiente del juego, acercándose lo más que se pueda al producto final. La preproducción es una de las partes más críticas del proceso ya que determinará la viabilidad del juego.

Producción

Aquí se llevan a cabo todas las tareas especificadas en la fase de planificación, teniendo como guía fundamental el documento de diseño. Esto incluye, entre otras cosas, la codificación del programa, la creación de [sprites](#), tiles y modelos 3D, grabación de sonidos, voces y música, creación de herramientas para acelerar el proceso de desarrollo,

Pruebas

Al igual que en otros tipos de software, los videojuegos deben pasar en su desarrollo por una etapa donde se corrigen los errores inherentes al proceso de programación y se asegura su funcionalidad. Además, a diferencia de aquellos, los videojuegos requieren un refinamiento de su característica fundamental, la de producir diversión de manera interactiva (jugabilidad). Generalmente, esta etapa se lleva a cabo en tres fases:

- Pruebas físicas: se llevan a cabo por los diseñadores y programadores del juego. Se crean prototipos que simulan los eventos que pueden suceder en el juego. Un prototipo físico puede utilizar papel y lápiz, tarjetas de índice, o

incluso ser actuado fuera. Sobre la base de los resultados de estas pruebas se puede hacer una mejor aproximación al balance del videojuego, pueden prevenir problemas de programación. El objetivo es jugar y perfeccionar este simplista modelo antes de que un solo programador, productor o artista gráfico estén cada vez más introducidos en el proyecto. De esta manera, el diseñador del juego recibe retroalimentación instantánea en lo que piensan los jugadores del juego y pueden ver inmediatamente si están logrando su metas.

- Pruebas alpha: se llevan a cabo por un pequeño grupo de personas, que con anterioridad estén involucradas en el desarrollo, lo que puede incluir artistas, programadores, coordinadores, etc. El propósito es corregir los defectos más graves y mejorar características de jugabilidad no contempladas en el documento de diseño.
- Pruebas beta: estas pruebas se llevan a cabo por un equipo externo de jugadores, bien sea que sean contratados para la ocasión o que sean un grupo componente del proyecto (grupo QA). De estas pruebas, el videojuego debe salir con la menor cantidad posible de defectos menores y ningún defecto medio o crítico.

Mantenimiento

Una vez que el juego alcanza su versión final (RTM) y se publica, aparecerán nuevos errores o se detectarán posibles mejoras. Es necesario recopilar toda la información posible de los jugadores y a partir de ahí realizar los cambios oportunos para mejorar el juego en todos sus aspectos, ya sea de diseño, jugabilidad, etc. Estas correcciones o mejoras se hacen llegar a los usuarios en forma de [parches o actualizaciones](#), que en ocasiones pueden incluir algunas características nuevas para el juego.

