



UNIVERSIDAD POLITÉCNICA
DE TECÁMAC

TAREA COMPLETA



Alumnos:

1. González
Clímaco Bruno
2. Joaquín Espejel
Jesús Elías
3. Juárez Galicia
Karim Izahi
4. Gámez Ledo
Irving Jonathan

Matriculas:

1. 1321124272
2. 1321124258
3. 1321124269
4. 1321124302

Grupo:

4322IS

Carrera:

Ingeniería en
Software

ANALISIS DE LA PROGRAMACIÓN VISUAL

- **Programación Orientada a Objetos:**

La Programación Orientada a Objetos (POO) es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos. Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.

- **Características y aplicaciones de eventos.**

- Características

- Dependencia de eventos
- Orientada al servicio
- Eventos
- Controlador de eventos
- Funciones de activación
- Tiempo controlado

- Aplicaciones

Ya que la programación orientada a eventos resulta ser más un enfoque que un tipo de lenguaje, las aplicaciones que se basan en eventos pueden crearse en cualquier lenguaje de programación.

La programación orientada a eventos está transformando el mundo de los servicios de desarrollo de software y se está convirtiendo en parte integral del desarrollo actual. Su difusión fue estimulada por Windows y la difusión de entornos visuales.

- **Características de componentes y métodos visuales y no visuales.**

- Aplicaciones

Un componente de software es un elemento de un sistema de prestación de servicios predefinido que puede comunicarse con otros componentes y es un objeto escrito de acuerdo con las especificaciones. No importa cuál sea esa especificación, siempre que el objeto cumpla con la especificación.

- Visuales y No Visuales

Un componente es visual cuando tiene una representación gráfica en el diseño y el tiempo de ejecución (botones, controles deslizantes, cuadros de edición, etc.) y se dice que es no visual en caso contrario (temporizadores, diálogo que no aparece durante el diseño, etc.). Aparte de eso, no hay otras diferencias entre ellos, excepto, por supuesto, la que se deriva de la visualización del componente.

- **Procesos de desarrollo visual en proyectos distribuidos y de escritorio.**

El desarrollo de software moderno como herramienta de programación visual es 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. Entonces, 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 y la programación. Sin embargo, a pesar de la simplicidad del lenguaje, puede resultar engorroso porque tiene muchos gráficos. 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. Los VPL son muy limitados en la gama de características expresivas que utilizan. Esto dificulta la realización de operaciones más complejas, por lo que los gigantes tecnológicos del mundo de la programación rara vez utilizan este tipo de lenguaje.

- **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
- Calculo de presupuesto
- Hacer un plan de contingencia
- Documento de puntos de referencia

- **Herramientas y lenguajes de programación visual.**

- *Python:*

Es un lenguaje de programación orientado a objetos dinámicos y de alto nivel que se usa comúnmente para desarrollar aplicaciones rápidamente.

- *C++:*

Un lenguaje de programación y un entorno de desarrollo potente y flexible para crear aplicaciones para Windows, Linux, iOS y Android.

- *JavaScript:*

El editor de JavaScript en Visual Studio es compatible con EcmaScript 6 y cuenta con el motor IntelliSense más avanzado del mercado.

JavaScript es un lenguaje de primera clase en Visual Studio. Puede usar la mayoría de las herramientas de edición estándar (extractos, IntelliSense, etc.) al escribir código JavaScript en el IDE de Visual Studio.

- *TypeScript:*

Es un conjunto completo de JavaScript que se compila en JavaScript sin procesar y le permite crear un código más extensible.

- *C#:*

Es un marco multiplataforma gratuito, de código abierto y de alto rendimiento para crear aplicaciones modernas, basadas en la nube y conectadas a Internet. Con ASP.NET Core, puede crear aplicaciones y servicios web, aplicaciones IoT y servicios back-end móviles.

- *Razor:*

La arquitectura Razor de ASP.NET permite incluir C# directamente en sus vistas. Además, tiene acceso completo a los patrones y plantillas existentes de .NET Framework.

DISEÑO DE PROGRAMACIÓN VISUAL

CONCEPTO



PROCESO



DISEÑO DE PROGRAMACION VISUAL



CONSTRUCCION



INTERACTIVIDAD

Desarrollo de aplicaciones con programación visual

Programación Visual

La programación visual es un lenguaje donde se utilizan componentes gráficos como botones, símbolos e iconos, este lenguaje de programación le permite al usuario ilustrar visualmente la codificación generada por el programa u ordenador.
La programación visual permite el desarrollo de aplicaciones en sus plataformas con poco código.

Eventos visuales.

Un evento es una orden para la aplicación de que algo ha sucedido. Un ejemplo es cuando un usuario al hace click en un control, el formulario puede generar el evento de click y llamar a un procedimiento de controlador de eventos.

Componentes y herramientas visuales de entornos de desarrollo.

Aplicaciones o entornos de desarrollo que sirven para facilitar la estructura de un código mediante componentes visuales.

NETBEANS
ECLIPSE
BUILDER C++
DELPHI

Reconocer la funcionalidad de los eventos visuales.

La funcionalidad de un evento visual es que se envía una señal a la aplicación a la hora de hacer click, para que inicie un proceso, y este se muestre en pantalla.

Proceso de programación visual.

El proceso de un programa visual puede empezar por la creación mediante el uso de elementos gráficos y textuales, seguidos de un código de proceso.

**Propiedades de los
componentes visuales de
proyectos de software**

Accesibilidad

La accesibilidad en un proyecto de software es necesaria para garantiza que el proyecto este disponible para los usuarios.

Apariencia

La apariencia en un proyecto de software se necesita para que la interacción con el usuario sea lo más intuitiva posible.

Comportamiento

Los comportamientos de los componentes visuales en proyectos de software son controlados por instrucciones generadas por el usuario a la aplicación. Estos cambian dependiendo de las instrucciones generadas por la aplicación o usuario.

Datos

Los datos son indispensables para darle un valor al a las variables de un proyecto.

Estos son representaciones simbólicas (Numéricas alfabéticas o algorítmicas.) de un atributo.

Diseño y estilos

El diseño y estilo se usa para definir o darle un formato a la presentación de un aplicación proyecto o pagina web.

INVESTIGACIÓN DE PROGRAMACIÓN VISUAL

- **CONCEPTOS DE VIDEOJUEGOS:**

Los videojuegos son aplicaciones de entretenimiento interactivo que crean una experiencia en un televisor, computadora u otro dispositivo electrónico mediante comandos o controles específicos.

- **CONCEPTOS Y TIPOS DE GAME DESIGNER, STORYBOARD:**

1. **Game designer:**

El concepto del juego, en su forma más simple, es una visión del juego fácil de entender. También es una forma de vender tu idea de juego. Su idea de juego debe incluir exactamente qué es el juego y de qué se trata la creación. Esto incluye la historia, las ilustraciones y cómo ganas dinero en el juego.

TIPOS:

- **GENERALISTAS:**

Los recién mencionados; diseñadores de juegos junior y senior, así como líderes creativos y directores creativos. Estos perfiles son generalmente responsables de todas las áreas del diseño y la consistencia general.

- **ESPECIALISTAS EN MONETIZACIÓN:**

Un personaje muy reconocible, especialmente en los juegos móviles, ya que casi todos sus juegos requieren de alguien con experiencia en diseñar e implementar una estrategia de monetización adecuada (generando ingresos reales); además de saber equilibrar la economía en el juego.

- **DISEÑADORES DE NIVELES (LEVEL DESIGNERS):**
suelen ser un perfil mixto entre diseñador y artista 3D.
- **DISEÑADORES DE CONTENIDO Y DE MISIONES:**
Estos diseñadores se enfocan en agregar y equilibrar el contenido del juego, desde assets como edificios, vehículos, personajes y armas, hasta misiones, rompecabezas y más.
- **DISEÑADORES DE NARRATIVA:**
Son responsables de escribir, diseñar e implementar todos los elementos narrativos del juego. A menudo trabajan con diseñadores de niveles y diseñadores de contenido para crear una experiencia de juego satisfactoria.
- **DISEÑADORES DE SISTEMAS:**
Se centran en el diseño de mecánicas de juego de nivel superior. En estudios más grandes, pueden ser más especializados (diseñadores multijugador, diseñadores de combate, diseñadores de progresión de jugadores, etc.).

2. STORYBOARD:

Un storyboard, también conocido como guión gráfico, es una combinación de dibujos secuenciales que ilustran la planificación de una obra audiovisual. Este es un paso esencial que debe seguir toda buena preproducción, ya que ayuda a guiar el rodaje. Se utilizan comúnmente en cine, televisión y comerciales, así como en muchas otras áreas de creación de videos.

La función principal del guión gráfico es permitir que todos los miembros imaginen y obtengan una vista previa del resultado final de la obra. De esta forma, podrán trabajar de forma coordinada para que el proyecto sea todo un éxito. Los guiones gráficos también incluyen una serie de

instrucciones en la parte inferior de cada foto. Resaltan los detalles más intrincados o los más difíciles de explicar.

TIPOS:

- Si observa la naturaleza de los guiones gráficos y cómo se hace, se puede dividir en varios tipos. Por ejemplo, puede encontrar guiones gráficos en color y guiones gráficos en blanco y negro. Algunos están perfectamente dibujados y explicados, mientras que otros son solo líneas.
- **STORYBOARDS AL USO:**
El clásico guión gráfico hecho a mano. Puede estar mejor o peor dibujado, pero su esencia sigue siendo la misma.
- **MINIATURAS**
Es un concepto bastante parecido al anterior, aunque como su propio nombre indica, en miniatura. Se representa todo en un tamaño mucho más pequeño, normalmente en un único folio.
- **GUIONES GRAFICOS ANIMADOS:**
Estos pueden ser animados de manera estándar mediante actores o representaciones, o incluso hacerse de manera digital con programas tecnológicos.

• TIPOS Y CARACTERÍSTICAS DE MOTORES DE VIDEOJUEGOS Y LENGUAJES DE VIDEOJUEGOS:

1. MOTORES DE VIDEOJUEGOS:

Un motor de videojuegos es un término que hace referencia a una familia de bibliotecas de programación que permiten el diseño, la creación y la representación de videojuegos. A la hora de elegir todos los motores de

videojuegos disponibles en el mercado, el aspecto más destacado son las capacidades gráficas, ya que se encargan de mostrar imágenes en 2D y 3D en pantalla, además de calcular polígonos, iluminación, texturas y más. .. Otras características a tener en cuenta a la hora de elegir son lo fácil que es aprender a utilizar un motor de videojuegos y lo fácil que es exportar el juego a diferentes plataformas. Algunas de las características más importantes son:

- **EL MOTOR DE FÍSICAS:**

Los motores de física permiten aplicar métodos físicos a los videojuegos, dándoles una sensación más realista de cómo los objetos interactúan con el entorno. Es decir, se encarga de realizar los cálculos necesarios para simular objetos con propiedades físicas como peso, volumen, aceleración, gravedad, etc.

- **EL MOTOR DE SONIDO:**

Los sonidos y la banda sonora de un juego de ordenador son igualmente una parte vital. El motor de sonido se encarga de apilar las pistas, cambiar su ritmo, eliminarlas de la reproducción y sincronizarlas, entre otras cosas.

- **EL SCRIPTING:**

Todos los motores de videojuegos tienen un lenguaje de programación que permite llevar a cabo la actividad de los personajes y elementos importantes para el juego de ordenador.

Dentro de las diversas opciones de motores de videojuegos podemos reconocer los motores famosos y los restrictivos o confidenciales, que son los que realizan las organizaciones de videojuegos importantes para planificar sus títulos más famosos.

2. LENGUAJES DE VIDEOJUEGOS:

El lenguaje con el que se modifican los juegos de ordenador es un

componente principal para garantizar una experiencia de juego fluida. Parece ser que cuando jugamos a un juego de ordenador sólo nos fijamos en la parte visual, en su tacto, ya que es lo más terrorífico, sin embargo detrás de ese estilo hay un pesado trabajo de programación que permite su actividad. Hoy te presentamos los mejores dialectos de programación para que hagas juegos de ordenador fascinantes y de la mayor calidad.

Lenguajes:

- C++:

Este lenguaje de programación es uno de los más utilizados en el negocio por los expertos. Es un lenguaje famoso en los títulos AAA, se utiliza en juegos de ordenador para PlayStation y Xbox, y en juegos gratuitos. Es el lenguaje generalmente viable con la mayoría de los motores de juegos y tiene un tiempo de ejecución realmente rápido. Además, permite a los diseñadores tener un amplio dominio sobre los equipos, la memoria de los ejecutivos y las ilustraciones, y a pesar del hecho de que tiende a ser desconcertante para utilizar desde el principio, cuando usted consigue su caída, que realmente va a querer tratar con algún otro idioma.

- C#:

C# es un lenguaje de programación extremadamente famoso, particularmente en condiciones de Windows. Es algo menos adaptable y viable que C++, sin embargo, algunos motores como Solidarity le permiten programar con él y no está restringido a un marco de trabajo específico o etapa; usted puede hacer juegos para iOS, Android, Windows Play Station y Xbox. Es un lenguaje más sencillo de aprender que C++ y será una decisión decente en el caso de que estés empezando o, por otro lado, en el caso de que mantengas que más

personas deberían apreciarlo, prestando poca atención al gadget.

- **Java:**

Es un lenguaje habitualmente utilizado y tiene numerosas similitudes con C++. Su principal seña de identidad es su flexibilidad, ya que suele utilizarse en todos los escenarios, dispone de un enorme número de sistemas para el giro 3D, ofrece módulos de código abierto y su modelo puede actualizarse continuamente.

- **JavaScript:**

Este es quizás de los lenguajes más generalmente involucrados en la web y en el giro de los programas de juegos. La mayoría de los motores de juegos son viables con JavaScript, y tiene numerosas estructuras para 3D y un amplio surtido de bibliotecas. Asimismo, algunos motores de juegos, por ejemplo, Solidaridad lo utilizan, por lo que podemos utilizarlo para hacer una amplia gama de contenidos en el juego.

- **Python:**

A pesar de no ser un lenguaje de programación exclusivamente para la producción de juegos de ordenador, Python es un lenguaje totalmente adaptable y fuerte por esta razón. Su ejecución es mucho menos difícil que la de otros dialectos (permite llevar a cabo planes complejos con no muchas líneas de código), y su estructura Pygame permite a los diseñadores hacer modelos de sus juegos de ordenador de forma rápida y eficaz, y se ocupa esencialmente de todas las etapas y marcos de trabajo.

- **Lua:**

Por último, Lua es un lenguaje de programación sencillo, rápido y fácil de dominar. Viable con dialectos adicionales intrincados y de rápida ejecución, se utiliza igualmente para aplicaciones web y manejo de

imágenes. Este lenguaje es particularmente valioso para las actividades libres y los desarrolladores que simplemente están empezando en la llamada.

- **METODOLOGÍAS DE DESARROLLO DE VIDEOJUEGOS:**

- **Fase de Concepción:**

Todo comienza con un pensamiento a partir del cual se formarán los puntos de vista centrales. La clase o los tipos de la no totalmente fijados en piedra, cómo será el ciclo de juego, y además se hace un tablero de la historia en el que se maneja una amplia gama de supuestos que se pueden ajustar, como el estilo de los personajes, el clima, la música, etc. Cuando se entiende lo que hay que hacer, ha llegado el momento de planificar.

- **Fase de diseño:**

Se empieza por caracterizar los componentes que conforman el juego. Se crea la historia, se hacen retratos de guion para decidir los objetivos, se eligen los personajes principales, la circunstancia única, etc. A partir de estos retratos de contenido, los artesanos intentan hacerse una idea del aspecto del juego, de la forma en que se imaginarán los personajes, las situaciones, los objetos, etc. Su responsabilidad es introducir la propuesta visual para dar forma a la primera idea. Los componentes sonoros del juego son igualmente retratados: efectos de audio, vibración, música, voces, etc. Aunque en este punto no se crea ni se graba nada. Al mismo tiempo, se determina el funcionamiento general del videojuego, algo que depende de la clase, ya que aportan la forma en que se conectan los elementos virtuales dentro del juego. Por último, con un pensamiento más claro de hacia dónde se dirigirá el juego, se realiza la configuración de la programación, que retrata la manera en que se ejecutará el juego de ordenador, el lenguaje o

dialectos de programación que se utilizarán, las técnicas que se seguirán, etc.

Todo lo anterior tendrá el objetivo de crear el Archivo del Plan que indicará el avance de la mano de obra, la mecánica y la programación del juego de ordenador.

- **Fase de planificación:**

El objetivo de esta etapa es reconocer las distintas tareas para fomentar el juego de ordenador. El trabajo se difunde entre las distintas partes del grupo de avance, se fijan los tiempos de corte, se organizan reuniones de seguimiento, etc.

- **Fase de producción:**

Cuando se tiene claro que hay que terminar, cómo hacerlo y se dispone de una oportunidad para hacerlo, entonces la creación comienza con el objetivo de hacer el juego, en una representación o modelo subyacente para ir avanzando poco a poco.

Por lo tanto, cada una de las tareas de la fase de organización se realiza con el registro del plan como ayuda: programación, delineación, mejora de la interfaz, movimiento, visualización, sonido, etc.

En el caso de que cada una de las piezas esté por fin recogida con precisión, esta etapa está terminada (hasta nuevo aviso). Sea como fuere, al igual que en la mejora de la programación convencional, es innegable que es un reto acertar en la primera ocasión en la que, entramos en una etapa para probar completamente el juego de ordenador.

- **Fase de pruebas:**

En esta fase se corrigen los errores del proceso de programación y se mejora la jugabilidad a medida que se prueba el juego.

Generalmente encontraremos dos tipos: las pruebas alfa, realizadas por

un pequeño grupo de personas generalmente involucradas en el desarrollo, y las pruebas beta, realizadas por un equipo externo de jugadores. Las primeras tienen como objetivo corregir defectos graves y mejorar características fundamentales no contempladas en el documento de diseño, mientras que las segundas se centran en detectar errores menores y perfeccionar la experiencia del usuario.

- **Fase de distribución:**

El transporte es el método que consiste en hacer duplicados del juego terminado y llevarlo a las tiendas (ya sean físicas o informáticas) para que los jugadores puedan conseguirlo o hacerse con él.

Por otra parte, la exhibición es también fundamental para lanzar el juego y conseguir tantos jugadores como sea razonable. No incluye una petición concreta dentro del ciclo de mejora, ya que algunas organizaciones comienzan a batallar con sus videojuegos meses o incluso mucho tiempo antes de distribuirlos. En realidad, depende de los activos que los ingenieros necesitan para avanzar en el trabajo y no incluye ser una división dentro de la organización real, sin embargo, tanto el transporte y la exhibición puede ser designado a las organizaciones externas tenían experiencia práctica aquí.

- **Fase de mantenimiento:**

Aunque el juego esté terminado y en posesión de los jugadores, su ciclo de vida no está ni mucho menos terminado. La etapa de mantenimiento es una oportunidad ideal para corregir nuevos errores, seguir desarrollándolo, etc. Esto se hace mediante la entrega de parches o actualizaciones en el mercado.

-
- No obstante, también es una valiosa oportunidad para seguir sacándole partido. Ya sea en forma de microtransacciones, membresías de pago o incluso extensiones totales que añaden nuevos elementos al juego sin

ajustar el motor del juego por dentro y por fuera, supongamos que sería más o menos como capitalizar la base subyacente.

- **PROCESO DE DISEÑO DE INTERFACES DE VIDEOJUEGOS EN 2D Y 3D:**

- La interfaz de usuario de un juego de ordenador es la marca de la colaboración entre el jugador y el juego. Su objetivo principal es dar los datos esenciales para que el cliente pueda hacer todo lo que el juego le propone de forma absolutamente líquida. Una configuración decente de la UI guía de manera inmediata o natural para que el jugador pueda recorrer el universo de su juego de ordenador con precisión.

Aquí hay una serie de cosas que hay que recordar mientras se planifica el punto de conexión de un juego de ordenador:

- **Entorno:**

Lo principal que hay que tener en cuenta es el lugar en el que se desarrollará el juego que se está planificando. Debes tener en cuenta los posibles resultados e impedimentos que presenta el escenario. No es algo similar hacer juegos para teléfonos móviles que para un centro de control o un PC.

- **Contenido:**

Una configuración decente de la interfaz de usuario proporciona al jugador todos los datos que se espera que se comuniquen con el juego para que todo funcione como se espera.

- **Diseño visual:**

Los juegos de ordenador, muy a menudo, entran por los ojos. Unos visuales repugnantes o gruesos en el punto de conexión del juego pueden ser contraproducentes y alejar al jugador de la experiencia vívida

que hay que darle. Hay que caracterizar el estilo de trabajo.

- **Arquitectura de la información:**

Caracterice qué componentes son de mayor o menor importancia para el cliente y coordínelos para que todo aporte un plan de puntos de conexión consciente e importante.

- **DESARROLLO DE PROTOTIPOS DE VIDEOJUEGOS:**

- La formación de un juego de ordenador es un ciclo largo y desorbitado, y es excepcionalmente normal que se cometan confusiones durante su desarrollo. Para percibir, aprender y arreglar estos errores, se elaboran modelos o maquetas para obtener la visión genuina del juego y los datos fundamentales para diseccionarlo. Así, un modelo es una variante acabada con todas las funcionalidades del juego. Además, el modelo se examina como una pieza del giro iterativo, viendo en cada ciclo los errores que surgen y proponiendo ideas para solucionarlos.

- **Cuantos más prototipos se realicen, mayor será la calidad del juego.**

Por otro lado, los prototipos siguen las siguientes bases:

- No se hacen sin razón, son para abordar cuestiones y, de esta manera, debe haber indagaciones detrás de ellos.
- Por lo general, hay que empezar sin ninguna preparación, a pesar de que se pueden reutilizar algunos pensamientos.
- Tienes que centrarte en ellas, intentando hacer primero la más básica.
- Puede completar algunos modelos al mismo tiempo para probar algunas regiones de forma independiente y ser sustancialmente más eficaz.
- No es necesario que sea en el PC, puede ser perfectamente en lápiz y papel.
- No debemos pensar a menudo en la naturaleza de nuestro modelo.
- Podemos depender de ciertos patrones, por ejemplo, "Lo que ves es lo que tocas".

- Deberíamos utilizar instrumentos que nos proporcionen una tonelada de trabajo hecho, por ejemplo, elementos 3D predefinidos o programación "en caliente" (sin reunir). Los aparatos absolutamente más conocidos, tanto en la creación de prototipos como en la mejora de juegos, son Solidarity (Zombeer), Unbelievable Motor (Batman Arkham Adventure) o CryEngine (Long ways).

- **CONCEPTO, TIPOS Y CARACTERÍSTICAS DE LOS MOTORES DE VIDEOJUEGO:**

- 1. **MOTORES DE VIDEOJUEGOS:**

Un motor de videojuegos es un término que hace referencia a una familia de bibliotecas de programación que permiten el diseño, la creación y la representación de videojuegos. A la hora de elegir todos los motores de videojuegos disponibles en el mercado, el aspecto más destacado son las capacidades gráficas, ya que se encargan de mostrar imágenes en 2D y 3D en pantalla, además de calcular polígonos, iluminación, texturas y más. .. Otras características a tener en cuenta a la hora de elegir son lo fácil que es aprender a utilizar un motor de videojuegos y lo fácil que es exportar el juego a diferentes plataformas. Algunas de las características más importantes son:

- **EL MOTOR DE FÍSICAS:**

Los motores de física permiten aplicar métodos físicos a los videojuegos, dándoles una sensación más realista de cómo los objetos interactúan con el entorno. Es decir, se encarga de realizar los cálculos necesarios para simular objetos con propiedades físicas como peso, volumen, aceleración, gravedad, etc.

- **EL MOTOR DE SONIDO:**

Los sonidos y la banda sonora de un juego de ordenador son igualmente una parte vital. El motor de sonido se encarga de apilar las pistas,

cambiar su ritmo, eliminarlas de la reproducción y sincronizarlas, entre otras cosas.

- **EL SCRIPTING:**

Todos los motores de videojuegos tienen un lenguaje de programación que permite llevar a cabo la actividad de los personajes y elementos importantes para el juego de ordenador.

Dentro de las diversas opciones de motores de videojuegos podemos reconocer los motores famosos y los restrictivos o confidenciales, que son los que realizan las organizaciones de videojuegos importantes para planificar sus títulos más famosos.

- **TRANSICIÓN NARRATIVA Y LENGUAJE VISUAL DE VIDEOJUEGOS:**

- **Narrativa de videojuegos:**

Es muy posiblemente del término más repetido en la discusión de los juegos de ordenador en curso, eso es lo que una especie de palabra de brujería, rápidamente, parece levantar la charla. Por lo tanto, una idea con la que cualquier persona es incluso marginalmente consciente de la discusión que abarca el medio es muy probable que actualmente reconocible. Obviamente, su promoción es una situación con dos vertientes que de igual manera que hace crecer la cosa, añade desorden respecto a su significado. En consecuencia, el texto que hoy nos ocupa no es más que un tímido empeño por combatir el olvido con el motivo de que, a través de creadores e investigadores del tema, se revele alguna visión del mismo al tiempo que se caracterizan términos como historia, trama o leyenda.

- **Lenguaje visual de videojuegos:**

El estilo de elaboración de un juego de ordenador puede implicar que un jugador tenga que descubrirlo o no. Basta con recordar que Borderlands tenía al principio un estilo sustancialmente más práctico y el estudio optó

por transformarlo en algo más cercano al cómic, lo que atrajo mucha más atención de los jugadores. La artesanía debe reflejar el alma del juego y responder a su interactividad, pero también hay que tener en cuenta los medios y el tiempo disponibles. Esta multitud de componentes son los que han tenido en cuenta los alumnos del Postgrado en Artesanía y Plano Visual de los Juegos de Ordenador del Foco Universitario de Innovación y Artesanía Avanzada (U-bit) para realizar tres juegos de ordenador con estilos visuales totalmente diferentes: Cry of Iron, Maskerade: The Vacant Cry y Nox.

- **EXPLICAR EL PROCESO DE DESARROLLO DE VIDEOJUEGO ACORDE A LOS ELEMENTOS DE PROGRAMACIÓN VISUAL:**

Antes de su lanzamiento, los juegos de ordenador deben pasar por algunas fases durante su desarrollo. Independientemente del tipo de juego que sea, o de la fase en la que se encuentre, suelen pasar por ciclos comparativos durante su perfeccionamiento.

Procesos:

- **Planeación**
- **Preproducción**
- **Producción**
- **Periodo de prueba**
- **Prelanzamiento**
- **Lanzamiento**
- **Post – lanzamiento**

Todo esto para la una buena ejecución y evitar fallas al momento del lanzamiento, pasando por diferentes fases para rectificar errores