

ÍNDICE DE TODAS LAS TAREAS

TAREA 1.....	2
TAREA 2.....	8
TAREA 3.....	9
TAREA 4.....	10

Universidad Politécnica de Tecámac

INGENIERIA EN SOFTWARE

ASIGNATURA: Programación Visual

PROFESORA: Emmanuel Torres Servín

TAREA PROGRAMACIÓN VISUAL

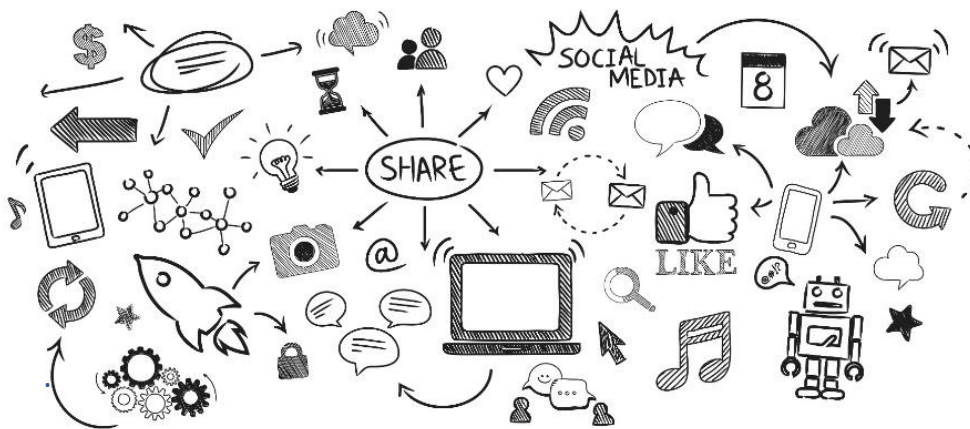
INTEGRANTES:

Ramírez George Samuel – 1321124307

Romero Hurtado Leslie Estefanía – 1321124300

GRUPO: 4322IS

ANÁLISIS DE LA PROGRAMACIÓN VISUAL



Conceptos de programación orientada a objetos.

La programación orientada a objetos es un paradigma de programación que se modela con objetos de hecho este paradigma se basa en el concepto de clases y objetos creando instancias individuales a los objetos. Algunos de los conceptos principales de POO son los siguientes:

- Objeto: Es un conjunto de variables y métodos que permite separar diferentes componentes de entidades de la vida real teniendo como resultado la instanciación de una clase en forma de un objeto
- Mensajes: Es el texto o petición que se le realiza para la ejecución de una tarea que requiera un objeto obteniendo valores de un atributo publico
- Método: Son todas las funciones que ayudan a que un objeto pueda efectuar un servicio respondiendo al mismo.
- Variable: es una palabra que representa algo que va a cambiar constantemente, que es inestables, inconsistente y mudable
- Atributos: características que diferencian a un objeto de otro, se puede decir que son las cualidades que se guardan en variables y estas pueden tener valores distintos,
- Instancia: es un modelo que procesa información, lee o especifica información, las instancias pueden ser todos los objetos de algún otro

Existen 4 pilares de la programación orientada a objetos que se describe en lo siguiente:

1. Encapsulación: es una característica de la programación orientada a objetos permite acceder a todo lo referente a un objeto a través de propiedades y métodos. En pocas palabras es cuando limitamos el acceso o damos un acceso restringido de una propiedad a los elementos que necesita a un solo miembro
2. Abstracción: La clase debe representar las características específicas de un objeto, en pocas palabras nos abstrae de la complejidad del mudo exterior dándonos una serie de atributos y comportamientos, es decir, propiedades y funciones, pudiendo crear objetos a partir de dicha clase
3. Herencia: Es cuando una clase obtiene rasgos (métodos y atributos) heredados desde otra clase existente. Teniendo una relación entre la clase general es decir una clase base y la clase más específica haciendo posible que podamos compartir métodos y datos entre clases, subclases y objetos
4. Polimorfismo: Es una cualidad de tener muchas formas, se refiere al hecho de tener varios objetos de diferentes clases base, pero se pueden utilizar de manera diferente sin tener que saber la clase exacta para poder hacerlo,

teniendo la posibilidad de definir varias clases indistintas denominados de forma idéntica con métodos y atributos iguales

Características y aplicaciones de eventos.

Alguna de las características que podemos encontrar en la programación orientada a objetos y programación orientada a eventos son las siguientes:

- Dependencia de eventos: Acciones que se le dan al usuario con el objetivo de separar la lógica de procesamiento de eventos.
- Orientada al servicio: Se ocupa para programar vaya la redundancia programas que son utilizados para el servicio ejecutando así los servicios dados por el sistema
- Eventos: Es una condición que surge al ir haciendo la ejecución del programa y que requiere alguna acción por parte del sistema. Estos incluyen la interfaz y las acciones que deben estar en el programa cuando se vaya ejecutando, interactuando con un objeto (un claro ejemplo es cuando realizamos usar el mouse para seleccionar un botón)
- Controlador de eventos: Podemos decir, que es una función o método que ejecuta una acción o tarea específica cuando se activa un evento determinado
- Funciones de activación: Se utilizar para seleccionar que controlador de eventos ocupar, es decir, también son funciones que deciden que ejecutar cuando se produce dicho evento
- Tiempo controlado: Es una tarea preestablecida porque el código es ejecutado en un momento determinado

Algunas de sus aplicaciones son las siguientes:

- Clic sobre un botón
- Doble clic sobre el nombre de un fichero para abrirlo
- Arrastrar un icono
- Pulsar una tecla o una combinación de teclas
- Elegir una opción de un menú
- Escribir en una caja de texto
- Mover el ratón

Características de componentes y métodos visuales y no visuales.

Los componentes en programación visual es una clase que implementa una interfaz, o implementa una clase que permiten solicitar y/o brindar un servicio o funcionalidad y estos a su vez se subdividen en dos partes; componentes visuales y componentes no visuales.

Una característica importante de los componentes es que se pueden diseñar utilizando una clase que es un componente en el entorno del desarrollador

Los métodos o componentes visuales son aquellos que cumplen con ser una representación gráfica de tiempo, diseño y ejecución (ejemplo: ventanas, botones, cuadros de texto, barras scroll, etc.) de las páginas web, que este a su vez se subdivide en dos tipos:

1. Componentes interactivos: esto quiere decir que el usuario va a manipular los datos implícitos en el programa y los eventos que contenga
2. Componentes gráficos: esto se hace mediante librerías, funciones básicas como por ejemplo TCanvas, no suelen recibir eventos del usuario, pero si del programador

Los métodos o componentes no visuales (cuadros de dialogo no visibles, formato de diseño de los paneles, temporizadores, formularios dentro del sitio web, controladores, etc.).

Procesos de desarrollo visual en proyectos distribuidos y de escritorio.

Un proceso de desarrollo de software es la descripción de secuencias de múltiples actividades que son asignadas por el jefe de equipo para generar un proyecto coherente, el objetivo básico de este es hacer predecible el trabajo requiriendo de predecir el costo, mantener calidad y predecir el tiempo de desarrollo, pero también definiendo las actividades que cada persona va a cumplir, en las fases del proceso (especificación, análisis, etc.) o tareas mas concretas (crear diagramas, escribir código, etc.) facilitando el modelado visual junto con la capacidad de apreciar diferentes elementos e interacciones del sistema en distintos niveles de abstracción de los datos.

Requerimientos visuales de proyectos distribuidos y de escritorio.

Es importante que la gestión visual en los proyectos nos brinde la posibilidad de tomar un punto de vista beneficiando mas que nada a hacer un software de calidad y con una buena interfaz, para esto podemos utilizar lo siguiente:

- Cronogramas y gráficos de proyectos: por ejemplo, un diagrama de Gantt que ayuda a crear un cronograma de actividades, asignando mejor los recursos y tiempos a los programadores
- Tableros Kanban: trabajan con sprints que permiten ver el progreso del proyecto ayudando en el seguimiento de errores, planificación de sprints, análisis de sprints, reuniones diarias de actualización ágiles y gestión de producción web
- Se considera que debemos definir los objetivos tanto específicos como generales del proyecto, determinar resultados iniciales, identificar riesgos y limitaciones, crear una ruta visual, evaluar recursos, trazar un plan y documentar

Herramientas y lenguajes de programación visual.

- ✚ C# / ASP.NET Core: compila aplicaciones web y servicios conectados a internet y basadas a la nube, aplicaciones de IoT y servicios de back-end móviles utilizando los lenguajes HTML y JavaScript
- ✚ 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.
- ✚ Javascript: nos permite crear aplicaciones web interactivas, con cambios en los contenidos en tiempo real, efectos personalizados y muchas características más.
- ✚ CSS, Less, Sass y Font Awesome en ASP.NET Core: posibilidad de definir y mantener de manera eficaz los estilos y los archivos de hojas de estilo (CSS), además de tener un fácil acceso a los iconos sin imagen que ayudan a que la interfaz del sitio sea más intuitiva. Ahí es donde entran los lenguajes y herramientas que admiten Less y Sass, y bibliotecas como Font Awesome.
- ✚ Python: Lenguaje de programación dinámico y de alto nivel orientado a objetos, se usa para el desarrollo rápido de aplicaciones.
- ✚ Visual Basic: orientado a objetos, está dirigido a la plataforma .NET, lo cual le permitirá crear aplicaciones de Windows sin dificultad.
- ✚ C++: Un lenguaje de programación y entorno de desarrollo eficaz y flexible para crear aplicaciones para Windows, Linux, iOS y Android.
- ✚ F#: con funcionalidades orientadas a objetos para la plataforma .NET, y que se centra en facilitar la programación funcional de cualquier tarea.

REFERENCIAS

1. José M. Drake. (-). Microsoft PowerPoint - OO_08_I2_Proceso.ppt. 04/08/2022, de Master de Computación Sitio web: https://www.ctr.unican.es/asignaturas/MC_OO/Doc/OO_08_I2_Proceso.pdf
2. José Manuel Alarcón. (26 de julio de 2021). Los conceptos fundamentales sobre Programación Orientada Objetos explicados de manera simple. 05/08/2022, de Campus MVP Sitio web: <https://www.campusmvp.es/recursos/post/los-conceptos-fundamentales-sobre-programacion-orientada-objetos-explicados-de-manera-simple.aspx>
3. Helmut Sy Corvo. (30 de marzo de 2020). Programación orientada a eventos: características, ejemplos, ventajas, aplicaciones. 05/08/2022, de Lifeder Sitio web: <https://www.lifeder.com/programacion-orientada-a-eventos/>
4. Lizbeth Ramos. (6 de abril del 2016). COMPONENTES VISUALES Y NO VISUALES. 05/08/2022, de Blogspot Sitio web: http://tematicosliz.blogspot.com/2016/04/componentes-visuales-y-no-visuales_6.html
5. Alejandro Bedini González. (-). Gestión de Proyectos de Software.pdf. 05/08/2022, de - Sitio web: <https://www.inf.utfsm.cl/~guerra/publicaciones/Gestion%20de%20Proyectos%20de%20Software.pdf>
6. Jesús Moreno León. (octubre 26, 2016). ¿LENGUAJES VISUALES?. 05/08/2022, de Programamos videojuegos y "apps" Sitio web: <https://programamos.es/lenguajes-de-programacion-visuales-o-lenguajes-basados-en-texto/>

MAPA MENTAL

CONCEPTO DISEÑO

En programación visual se dice que es una planificación de una solución de un software estableciendo relaciones visuales, sistémicas o geométricas entre las partes de dicho diseño

PROCESO DE CONSTRUCCIÓN DE MAQUETADO.

Lo primero que se debe hacer es transformar el diseño gráfico en una interfaz en un sitio web. Para este maquetado se necesita presentar una navegación simple, siguiendo el orden jerárquico respectivo

PROCESO DE DISEÑO E INTEGRACIÓN DE CONTENIDOS Y COMPONENTES.

- Full responsive: Crear un diseño adaptado a distintas pantallas.
- Accesibilidad web: Seguir las normas de accesibilidad para los potenciales usuarios.
- SEO: Incluir las etiquetas, plugins y código para los motores de búsqueda.
- Validación: Cumplir con los estándares HTML/CSS para una correcta visualización y buen funcionamiento del sitio.
- Compatibilidad cross browser: Asegurar la visualización y funcionamiento adecuado del sitio en los distintos navegadores disponibles.

DISEÑO DE PROGRAMACION VISUAL

Es un conjunto de acciones paso a paso que sigue una lógica básica de entrada, procesamiento y salida dándole formalidad a nuestros algoritmos

CONCEPTO MULTIMEDIA

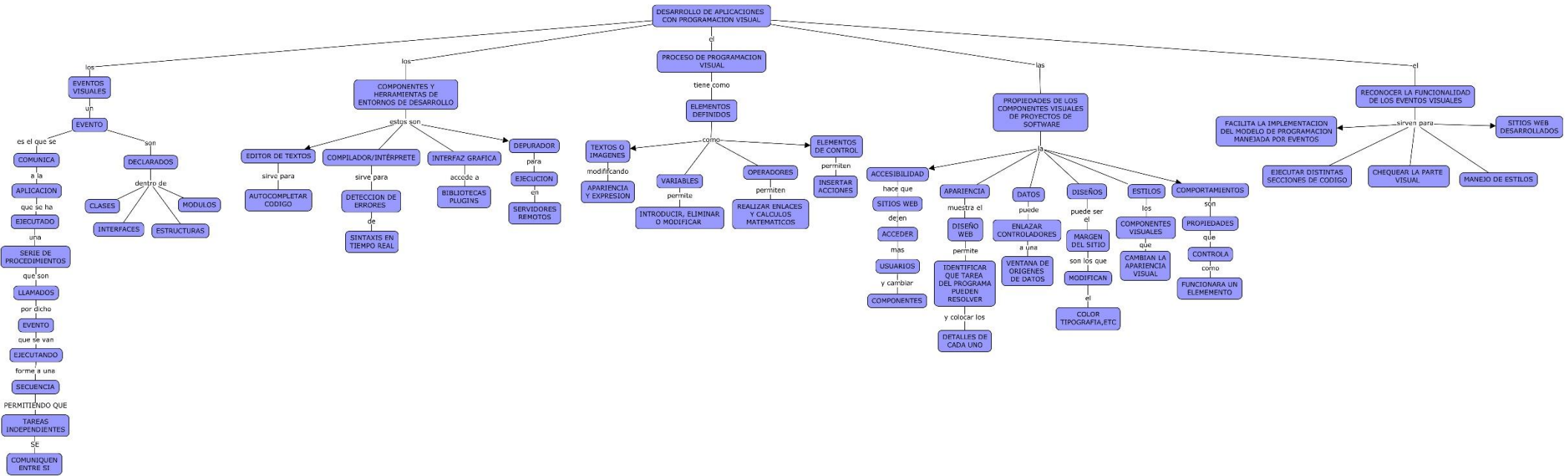
Esta orientado al desarrollo del entorno web y creación de aplicaciones interactivas que cuentan con lenguaje visual, texto, sonoro, colores, animaciones, videos, etc.

CONCEPTO INTERFACES

Permite programar aplicaciones con sus respectivos métodos y propiedades que, a su vez, estos son codificados en clases y se pueden crear objetos en POO. En otras palabras la interacción que se tiene con el programa

INTERACTIVIDAD ENTRE COMPONENTES.

En las paginas web se permiten representar documentos pero también se pueden proporcionar elementos para interactuar con el.





PROFESOR:

ESTUDIANTE LESLIE ESTEFANIA
ROMERO HURTADO

Leslie Estefanía Romero Hurtado.
1321124300

Samuel Ramírez George.
1321124307

INTRODUCCION:

En este documento desarrollaremos el tema *de Videojuegos* pasando por diferentes subtemas, para de esta manera abarcar la mayor cantidad de información que sea posible.

Durante el mismo desarrollaremos temas como: Conceptos de videojuegos, Tipos y características de los motores de videojuegos, transición narrativa, por mencionar algunos, cada uno con su respectivo desarrollo y ejemplos que puedan ser citados.

Esta investigación se realizará con el fin de que los alumnos de la Universidad Politécnica de Tecámac y en específico los que cursan la carrera de Ingeniería en Software puedan adquirir si bien no todo la información sobre el tema, si la suficiente información que genere en ellos la curiosidad por saber mas del tema.

Al final de esta terminaremos con las conclusiones que se arrojen y con la bibliografía de donde fue consultada la información que aquí se presenta, esperando que este material pueda ser de apoyo para cualquier persona que lo lea y con la finalidad de poder hacer crecer el conocimiento de la comunidad estudiantil.

VIDEO JUEGOS.

Concepto de Videojuegos.

¿Qué es un Videojuego?

Los Videojuegos se definen como aplicaciones interactivas orientadas a el entretenimiento que a través de mandos y controles nos permiten simular diferentes experiencias en la pantalla de un televisor, computadora u otro dispositivo electrónico.

Los videojuegos son diferentes a otras formas de entretenimiento ya que los videojuegos forzosamente deben de ser interactivos esto quiere decir que los usuarios deben de involucrarse con el contenido.

Como la palabra lo dice los Videojuegos son un tipo de juegos que hacen uso de la tecnología para permitir una interface la cual ofrezca una experiencia interactiva tanto de manera individual como de multijugador si es que el juego da esa opción y esto puede ser por medio de dispositivos como ordenadores, dispositivos móviles, consolas etc.

Los videojuegos hoy en día se han hecho muy populares a nivel mundial esto ha permitido que se convierta en un negocio muy rentable a través de la comercialización de los videojuegos, los patrocinios de las empresas para el desarrollo.

La palabra videojuego esta compuesta por video y juego, la primera proviene del verbo en latín *videre* que significa ver y la segunda tambien procede del latín *iocum* que se refiere a diversión.

Dentro de los géneros que se distinguen en los videojuegos se encuentran los siguientes:

- Acción.
- Estrategia.
- Simulación.
- Deportes.
- Carreras
- Aventuras.

Game Desinger.

Se refiere a profesional encargado de la creación de los videojuegos, este se encarga de diseñar todos los elementos que componen el juego, el concepto, los niveles, la complejidad entre otras cosas, se dice que no necesariamente debe de ser un programador o diseñador gráfico.

La labor principal del game desinger es dar un equilibrio a todos los aspectos del juego, desarrolla toda la idea de un videojuego como los personajes, la historia y el mundo donde se desarrolla, también se encarga de revisar el proceso para el desarrollo del juego fungiendo como un Líder.

Storyboard:

Es un conjunto de imágenes que forman una guía visual, con esta se pueden pre-visualisar las escenas del videojuego, también se usa el termino Guion Grafico para referirnos a los storyboards, este nos ayuda a hacernos una idea de lo que nos espera durante el videojuego.

Elementos:

- Información del guion.
- Los cuadros.
- La descripción.
- Observaciones.

Algunos ejemplos:



El storyboard se origina en el cine de Animación, los pioneros de esto fue el estudio de Walt Disney comenzó a utilizarlo en sus películas en el año de 1930 aunque su uso se popularizo hasta diez años después y se extendió a otras compañías del entretenimiento y aquí es donde llega a el sector de los Videojuegos.

Tipos de Storyboard:

- Storyboard Animado: Son bocetos individuales estos tratando de crear una sensación de tiempo y movimiento.
- Storyboard Digimatics: Son imágenes digitales unidas con el fin de crear una sensación de tiempo y movimiento
- Storyboard de Miniaturas: Este suele ser mas pequeño que los dos anteriores y se puede realizar en una sola hoja de papel, se dibuja mucho más rápido y utiliza garabatos en el panel de acciones.
- Storyboard Tradicional: son dibujos creados a lápiz creados por un artista bajo la supervisión del productor o de el director aunque es habitual que los productores creen su propio storyboard al gusto.

Motores de Videojuegos.

Los motores de videojuegos es un término que nos refiere a una serie de librerías de programación que permiten el diseño, la creación y la representación de un videojuego, es decir, se refiere a el lugar optimo donde se desarrolla un videojuego.

Tipos de motores de videojuegos y sus características:

Motor de física: el motor de física se encarga de que el juego tenga leyes de la naturaleza semejantes a las nuestras, por ejemplo la gravedad, las colisiones elásticas e inelásticas, la fricción, el movimiento de objetos debido al viento (incluyendo al mar), etc.; pero debido a que el videojuego está enfocado a audiencias que esperan ver reacciones directas de sus acciones, estas leyes deben físicas ser modificadas para generar un sentimiento de “hiperrealismo”, y de esta forma permitir en el usuario mayor interés en el mundo virtual.

Motor de renderización: Se encarga de la parte grafica del videojuego. Es código mediante el cual se visualizan los elementos en la pantalla. Para realizar esta función existen dos apis estándares en el desarrollo de aplicaciones graficas: OpenGL (open graphic library) y Direct3D.

Motor de Sonido: Los sonidos en un videojuego es una parte muy importante, el motor de sonido es el encargado de cargar las pistas, modificar su tasa de BTS, quitarlas de reproducción, sincronizarlas entre otras cosas.

Scripting: Los motores de videojuegos tiene un lenguaje de programación que le permite implementar el funcionamiento de los personajes y los objetos que forman parte de del mismo.

Motor de Entrada/Salida: Existen pocos APIs de amplia utilización que faciliten la creación de motores de entrada/salida de libre utilización, siendo DirectInput uno de los estándares más usados profesionalmente, debido a su capacidad de adquirir datos de entrada desde los principales puertos del computador

Los motores populares más utilizados y que dan mas posibilidades a los desarrolladores son:

- Unreal Engine.
- Unity 3D.
- Frostbite Engine.
- Decima Engine.
- Luminus Studio.

Lenguajes de Videojuegos.

El lenguaje de programación va a depender del tipo de videojuego que quieras programar o desarrollar, además si el grado de complejidad del diseño es alto se pueden utilizar más de un solo lenguaje de programación para que todas las capas del juego funcionen correctamente.

Algunos de los lenguajes que se pueden utilizar son:

C++ Es uno de los lenguajes más utilizados en el sector, es popular en títulos AAA, se utiliza para videojuegos en PlayStation y Xbox, se trata del lenguaje más compatible con la mayoría de los motores de juego y tiene un tiempo de ejecución muy rápido.

C# Es muy popular este lenguaje, sobre todo en Windows, es menos flexible y compatible que C++ pero la diferencia es que motores como Unity permiten programar con él y no se limita a una sola plataforma o sistema operativo.

Java es frecuentemente utilizado ya que presenta muchas similitudes con el lenguaje de C++ entre sus principales características está que es un lenguaje versátil ya que se puede utilizar en todas las plataformas y cuenta con una gran cantidad de frameworks para desarrollar en 3D.

Lua: Este es un lenguaje de programación sencillo y fácil de aprender, es compatible con lenguajes más complejos y también es de rápida ejecución, tiene utilidades en aplicaciones web y el procesamiento de imágenes.



Metodologías del desarrollo de videojuegos.

Metodología SUM:

Esta metodología tiene como objetivo desarrollar videojuegos de calidad en tiempo y costo así también busca la mejora continua del proceso para incrementar su eficacia y eficiencia, su estructura permite especificar el proceso de desarrollo sin mencionar las practicas especificas esto lo hace flexible y adaptable a cada realidad.

Esta metodología define roles

- Equipo de desarrollo.
- Productor interno.
- Cliente y verificador beta.

Ciclo de vida:

Este se divide en fases iterativas que se ejecutan de forma secuencial con la excepción de la gestión de riesgos ya que esta se realiza durante todo el proyecto. Las cinco fases secuenciales son:

- Concepto
- Planificación.
- Elaboración.
- Beta.
- Cierre.

Estas fases aparecen como adaptación al desarrollo de videojuegos de las fases de pregame, game y postgame que se presentan en Scrum donde las dos primeras de estas coinciden con las fases de planificación y elaboración de esta.

Proceso del diseño de interfases de videojuegos en 2D y 3D.

Definición: La interfaz de los videojuegos es lo primero que nos atrae como jugadores en una consola, de alguna manera se puede decir que es nuestra puerta de entrada, con lo que los diseñadores nos invitan a pasar para entrar a el mundo que ellos han construido. Debido a esto es muy importante diseñar una buena interfaz y generalmente esto se considera un desafío ya que los estudios descuidan esta parte y no tienen tiempo o recursos suficientes.

Estilos de interfaz en un videojuego:

- Diegéticas: Se refiere a que está incluida dentro del mundo del videojuego, puedes ser vista, escuchada o tocada por los personajes del juego,
Ejemplos: Dead Space, Fallout, Assassins Creed etc.
- No diegéticas: Se refiere a los elementos que se muestran fuera del mundo del juego y que solo son visibles y audibles para el jugador.
Ejemplos: Diablo III, GTA etc.
- Espaciales: Esta se refiere a un punto medio entre los dos anteriores, cuenta con elementos integrados a la acción del juego, pero los personajes no tienen conciencia de ellos.
- Meta: Es una parte del diseño de la interfaz que esta dentro de la historia del juego, pero no lo están en el espacio del juego, un ejemplo de esto podrían ser las gotas de sangre en las pantallas o los vidrios rotos.



Desarrollo de prototipo de videojuegos.

El desarrollo del prototipo de videojuegos se apoya en herramientas con las que es más fácil, rápido y barato probar distintas ideas para poder cambiar parámetros y explorar posibilidades, para de esta manera llegar lo antes posible a tener lo que se conoce como el molde con el que podamos entrar a una fase de producción de una manera más confiada una vez que se aprobaron los prototipos.

En este caso en cuanto mas prototipos se realicen, mayor será la calidad del juego. Por otro lado los prototipos siguen bases como son:

- Se hacen para responder cuestiones o preguntas serias.
- Se deben priorizar.
- No necesariamente se hacen en ordenadores pueden ser a lápiz y papel.
- Algunos se basan en corrientes como el "What You See Is What You Play"

Se cuenta con múltiples opciones para hacer un prototipo existe un gran abanico de posibilidades que nos permiten implementar ideas primarias, existen motores de videojuegos que se pueden obtener de manera gratuita que nos brindan ayuda para desarrollar los prototipos.

Transición narrativa y lenguaje visual de videojuegos.

La narrativa de los videojuegos se refiere a la interacción que existe entre el videojuego y los usuarios, esto lleva a que los usuarios puedan tomar decisiones que directamente afecten a los personajes del juego, de esta manera alterando su propósito y su destino en el juego.

Como tal los videojuegos son una forma más de contar historias estas historias tienen la particularidad de que el usuario puede tomar decisiones que determinan el camino de los personajes, de esta manera la libertad es muy grande pero no es total y absoluta.

La tecnología avanza para implicar cada vez más a la persona, algo que se nota especialmente en géneros como la aventura, la acción o el terror. La ambientación, los personajes y la trama son determinantes para enganchar al jugador, pero a diferencia de otros campos como en el cine o la literatura, aquí también cuenta la parte de juego y su forma de transmitir diversión

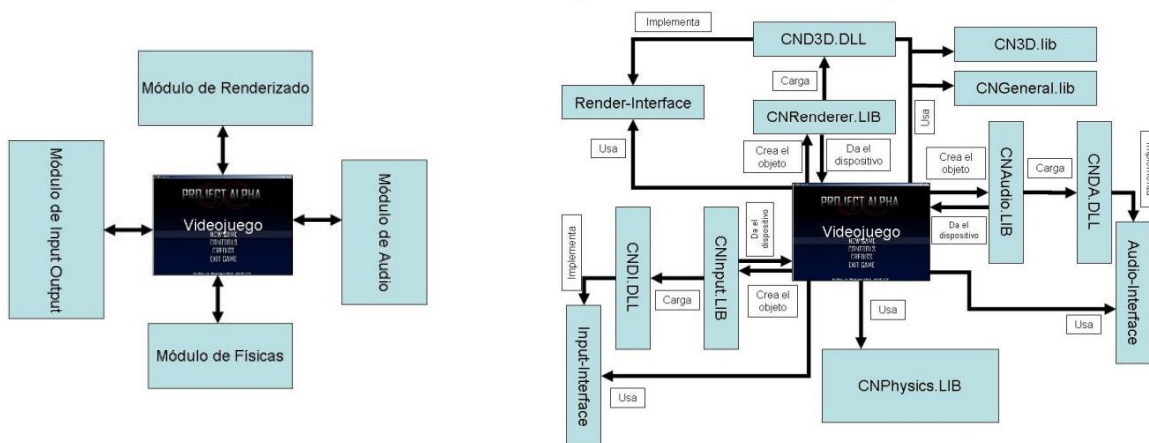
¿Cómo contar una historia sobre la que no tienes el control? Los vídeos y las animaciones programadas han sido la forma más común de contar las historias en las grandes producciones. De esa forma en realidad hay una ruptura entre lo que es el propio momento de juego y la interacción entre personajes que hace avanzar la trama.

A diferencia de la mayoría de los videojuegos, los títulos de Irrational Games funden estos dos conceptos y lo exponen todo en uno. “Siempre tratamos de asegurarnos de que no controlamos al jugador y le permitimos experimentar a su propio ritmo.

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

Los motores se desarrollan enfocándose en un concepto genérico modular, permitiendo la creación de videojuegos de distinta índole, priorizando los juegos de disparos en primera persona, juegos de aventura en tercera y primera persona y los juegos de deporte. El código del motor se encuentra generalmente siempre en inglés, permitiendo abarcar una mayor población de posibles usuarios alrededor del mundo.

Ejemplos:



Conclusiones:

Después de hacer el análisis de los elementos que debe llevar la realización de un videojuego podemos concluir que gracias a la tecnología, hoy en día existen múltiples herramientas que de alguna manera nos ayudan a la creación de los mismos.

También podemos concluir que a pesar de estas herramientas el desarrollo de los videojuegos es un tema que lleva bastante tiempo como inversión y esfuerzo. A Través de la investigación pudimos conocer un poco más sobre los antecedentes de los videojuegos y también sobre como han ido evolucionando continuamente.

Bibliografía:

https://es.wikiversity.org/wiki/Dise%C3%B1o_de_videojuegos/Creaci%C3%B3n_de_prototipos_de_videojuegos#:~:text=La%20creaci%C3%B3n%20de%20prototipos%20se,producci%C3%B3n%20de%20una%20manera%20confiada

<https://www.tokioschool.com/noticias/disenio-interfaz-videojuego/>

<https://profile.es/blog/lenguajes-programacion-videojuegos/>

<https://www.occamagenciadigital.com/blog/tipos-de-storyboard>