

TAREAS DE PROGRAMACION



UNIVERSIDAD POLITECNICA DE TECAMAC

MATERIA: PROGRAMACION VISUAL

INTEGRANTES DE EQUIPO

AGUILAR PALMA SAUL NICOLAS

FLORES CENTENO LUIS ANGEL

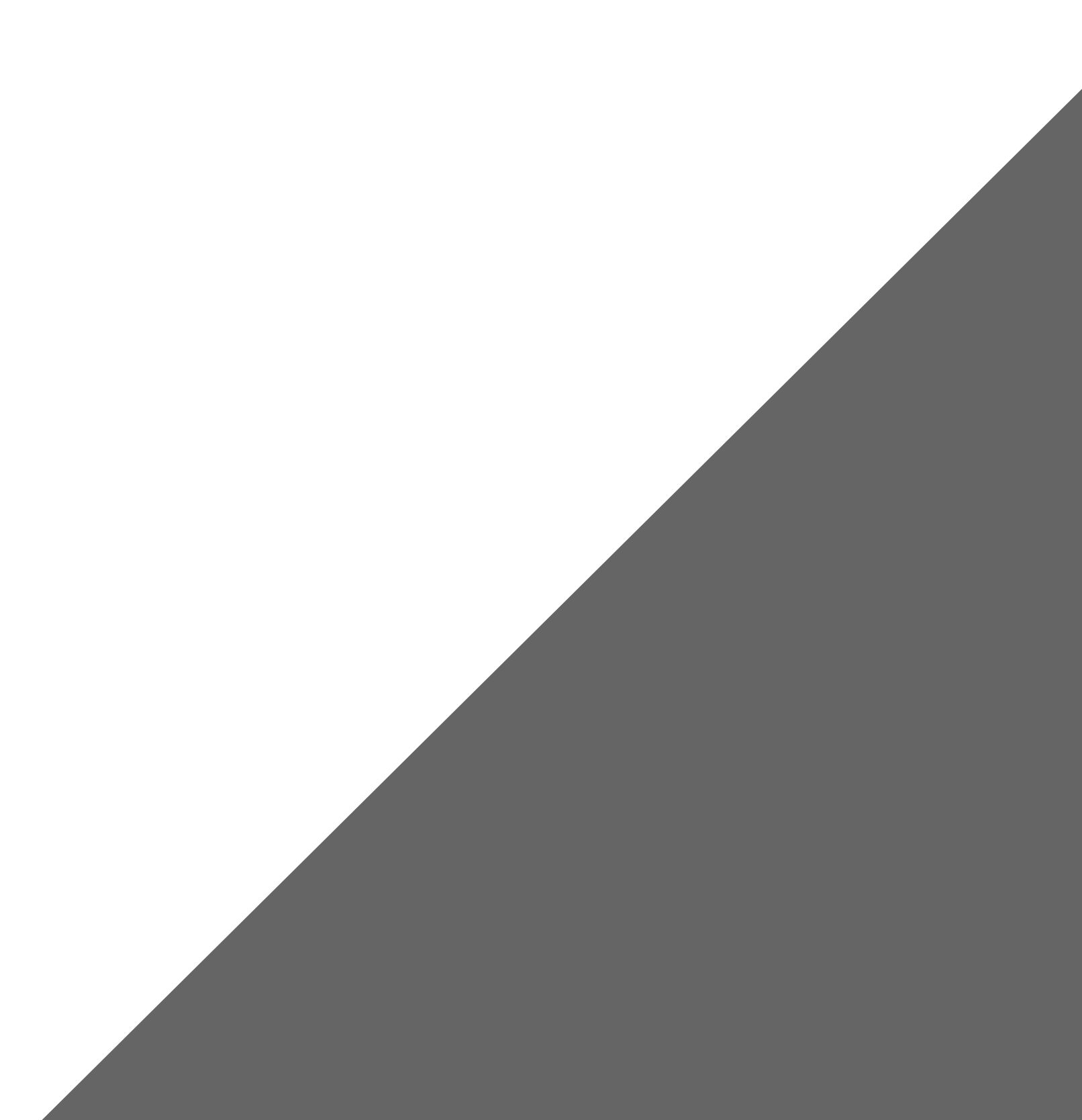
GONZALEZ CARAPIA RAUL

VAZQUEZ LARIOS DANIELA

GRUPO: 5322IS

Logotipo, nombre de la empresa

Descripción generada automáticamente



# INTRODUCCIÓN

En la programación visual, los elementos del lenguaje de programación están disponibles en forma de bloques diseñados de manera gráfica, por lo que también se la llama programación gráfica. La apariencia y el etiquetado de los módulos permite identificar qué tarea en el flujo del programa pueden resolver.

En este proyecto podemos ver lo que es la programación orientada a objetos y sus características y ejemplos, además de eso se explica de una forma fácil de comprender.

La Programación orientada a eventos es un paradigma Término común en el vocabulario científico y en expresiones epistemológicas cuando se hacía necesario hablar de modelos o patrones.

Además, podemos ver lo que son los motores de videojuegos y sus características.

Los requerimientos necesarios para poder usar de una forma óptima y eficiente para sacar el máximo rendimiento de este.

Tarea 1:

Realiza un resumen del Análisis de la programación visual detallando los siguientes puntos:

Conceptos de programación orientada a objetos.

• Características y aplicaciones de eventos.

• Características de componentes y métodos visuales y no visuales.

• Procesos de desarrollo visual en proyectos distribuidos y de escritorio.

• Requerimientos visuales de proyectos distribuidos y de escritorio.

• Herramientas y lenguajes de programación visual.

# Conceptos de programación orientada a objetos

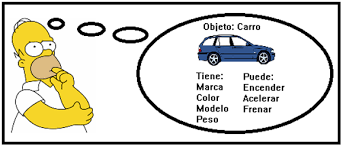
Es un paradigma de programación, es decir, Se basa en el concepto de clases y objetos.

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.

# Objetos

Es un conjunto de variables (o datos) y métodos (o funciones) relacionados entre sí. ​

​Los objetos en programación se usan para modelar objetos o entidades del mundo real (el objeto hijo, madre, o farmacéutica, por ejemplo).

Un objeto es, por tanto, la representación en un programa de un concepto, y contiene toda la información necesaria para abstraerlo: datos que describen sus atributos y operaciones que pueden realizarse sobre los mismos.

# Mensajes

Es la petición de un objeto a otro para solicitar la ejecución de alguno de sus métodos o para obtener el valor de un atributo público. ​

​En un mensaje siempre hay un receptor, lo cual no ocurre en una llamada a procedimiento.​

​Un mensaje consta de 3 partes: ​

​1. Identidad del receptor: Nombre del objeto que contiene el método a ejecutar. ​

​2. Nombre del método a ejecutar: Solo los métodos declarados públicos. ​

​3. Lista de Parámetros que recibe el método (cero o más parámetros)

# Métodos

Son aquellas funciones que permite efectuar el objeto y que nos rinden algún tipo de servicio durante el transcurso del programa. Determinan a su vez como va a responder el objeto cuando recibe un mensaje.

# Clases

Es un molde del que luego se pueden crear múltiples objetos, con similares características. ​

​Es una plantilla (molde), que define atributos (variables) y métodos (funciones) Define los atributos y métodos comunes a los objetos de ese tipo, pero luego, cada objeto tendrá sus propios valores y compartirán las mismas funciones.

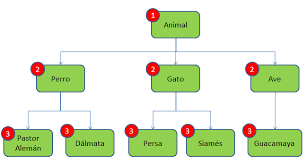
# Abstracción

Las características específicas de un objeto, aquellas que lo distinguen de los demás tipos de objetos y que logran definir límites conceptuales respecto a quien está haciendo dicha abstracción del objeto.

# Encapsulamiento

Es cuando limitamos el acceso o damos un acceso restringido de una propiedad a los elementos que necesita un miembro y no a ninguno más.

# Herencia

Es una propiedad que permite que los objetos sean creados a partir de otros ya existentes, obteniendo características (métodos y atributos) similares a los ya existentes. Es la relación entre una clase general y otra clase más específica. Es un mecanismo que nos permite crear clases derivadas a partir de clase base, nos permite compartir automáticamente métodos y datos entre clases subclases y objetos.

# Polimorfismo

Diagrama

Descripción generada automáticamenteSe refiere a la posibilidad de definir clases diferentes que tienen métodos o atributos denominados de forma idéntica, pero que se comportan de manera distinta.

# Características y aplicaciones de eventos

La Programación orientada a eventos es un paradigma Término común en el vocabulario científico y en expresiones epistemológicas cuando se hacía necesario hablar de modelos o patrones. de programación en el que la estructura y la ejecución de los programas van determinados por los sucesos o acciones que ocurren en el sistema, definidos por el usuario o por el propio sistema.

# Características

# Eventos

Cuando se produce o dispara un evento en programación permite al usuario realizar una serie de acciones lógicas para un determinado programa. sobre un determinado componente que presta un servicio de comunicación cuando se diseñan interfaces., se inicia un conjunto de acciones programadas por el usuario para ese evento concreto.

Diagrama

Descripción generada automáticamenteLos Eventos son las acciones sobre el programa, como, por ejemplo:

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

# Propiedad

Una propiedad Acciones que definen un objeto. es una asignación que describe algo sobre un componente elemento que presta un servicio de comunicación cuando se diseñan interfaces., como, por ejemplo:​

Un formulario,​ Un botón de comando, ​Una caja de texto​, Una etiqueta

# Una imagen​.

Dependiendo de la propiedad Acciones que definen un objeto., esta se puede asignar en tiempo de diseño usando la ventana Propiedades y/o en tiempo de ejecución en el momento de programar el control objetos que poseen propiedades y métodos que facilitan el desarrollo de aplicaciones informáticas.

# Método

Un método es una función que es llamada desde el programa, a diferencia de los procedimientos, estos no son programados por el usuario, ya que vienen preprogramados en el lenguaje de programación, los métodos realizan tareas típicas y comunes para todas las aplicaciones; por eso están incluidos dentro del lenguaje de programación, esto con el fin de liberar al usuario de la tarea de programarlos. Cada tipo de componente elemento que presta un servicio de comunicación cuando se diseñan interfaces. tiene sus propios métodos; debemos tener en cuenta que los métodos solo pueden ser ejecutados en tiempo de ejecución no en tiempo de diseño. Algunos ejemplos de métodos son:​

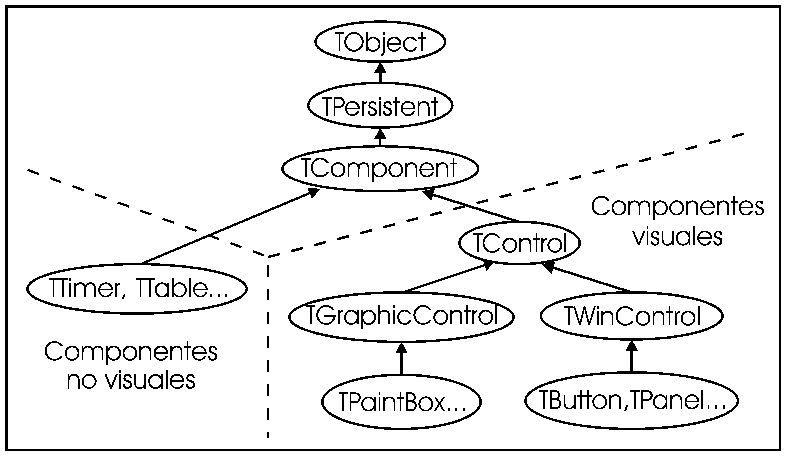
Diagrama

Descripción generada automáticamenteEn una ventana, el método MOVER, que mueve un formulario en un espacio de dos dimensiones en la pantalla.

# Características de componentes y métodos visuales y no visuales

Este estilo de programación presenta las siguientes características: Visual: los elementos se arrastran y sueltan en el flujo del programa para integrarlos. Orientado a acontecimientos: cada paso del programa comienza cuando ocurre un acontecimiento previamente definido.

Un componente es visual cuando tiene una representación gráfica en tiempo de diseño y ejecución (botones, barras de scroll, cuadros de edición, etc.), y se dice no visual en caso contrario (temporizadores, cuadros de diálogo -no visibles en la fase de diseño-, etc.)



# Procesos de desarrollo visual en proyectos distribuidos y de escritorio

Los sistemas distribuidos pueden concebirse como aquellos cuya funcionalidad se encuentra fraccionada en componentes que al trabajar sincronizada y coordinadamente otorgan la visión de un sistema único, siendo la distribución, transparente para quien hace uso del sistema.

En términos computacionales, se dice que un sistema distribuido es aquel cuyos componentes, de hardware o de software, se alojan en nodos de una red comunicando y coordinando sus acciones a través del envío de mensajes.

Con la programación distribuida, una ventaja es que las diferentes unidades informáticas pueden ejecutar diferentes algoritmos para resolver el mismo problema, lo que conduce a una solución significativamente mejor.

El objetivo de un proceso de desarrollo de programas es la formalización de las actividades relacionadas con el desarrollo del software de un sistema informático.

La mayoría de los proyectos que se desarrollan, finalizan tarde, cuesta mucho más de lo estimado. El software se encuadra entre los artefactos más complejos

que es capaz de desarrollar el hombre, y además dado que no tiene límites físicos por su carácter inmaterial, su dimensión se puede imaginar ilimitada.

# ¿Qué es un Software de escritorio?

Son los que se ejecutan en un ordenador/computadora de escritorio Mac, Windows o Linux. Su configuración está diseñada para usarse en un solo lugar, como un escritorio, de ahí proviene el nombre.

Originalmente así se dio por nombre, ya que antes las computadoras no eran portátiles. Ahora bien, la computadora portátil también puede usar software de escritorio ya que funcionan con los mismos sistemas operativos que los equipos de escritorio menos portátiles.

Pero para ello, se debe instalar en su computadora. La instalación se realiza desde un CD de software, DVD u otro medio.

# ¿Qué es un Sistema web?

Son sistemas informáticos complejos, como los programas que antes teníamos en la computadora, pero para internet, es decir, que se codifican en lenguajes soportados por los navegadores web y se alojan en un servidor en Internet.

# Naturaleza de las aplicaciones software

No existe un proceso de desarrollo universal. Debe configurarse de acuerdo con la naturaleza del producto y de la experiencia de la empresa.

# Tipos de aplicaciones:

Aplicaciones Monoprocesadoras: Se ejecutan en un solo computador. No se comunica con otras aplicaciones. Ej. Procesador de texto.

Aplicaciones Embebidas: Se ejecuta en un entorno computarizado

especial. Requiere codiseño hardware/software. El: Teléfono móvil.

Aplicaciones de Tiempo Real: Tiene entre sus especificaciones

requerimientos temporales. Naturaleza reactiva. El: Software de radar.

Aplicaciones Distribuidas: Se ejecuta en múltiples procesadores.

Requiere intercomunicación a través de la red. Ej: Aplicaciones de red.

Requerimientos visuales de proyectos distribuidos y de escritorio

Se lo pone fácil al lector

El sitio está optimizado para que sea legible en todos los dispositivos y el diseño es sencillo. Debe ser algo natural para el lector, pues si ve que el acceso a la información se complica, abandonará la página. Hay que buscar la estética, pero no perder de vista la funcionalidad y simplicidad. Claro, preciso y eficaz, sin saturar.

No hacen falta intermediarios

No existen estructuras complejas o largas explicaciones técnicas. Podemos incluir, por ejemplo, un desplegable en el que el usuario pueda ver más información si así lo desea. El lector debe saber a simple vista o en muy pocos pasos lo que está viendo y si le interesa o no.

# Trata de captar la atención

Utilizando, por ejemplo, la jerarquía visual. La jerarquía visual nos dice que los ojos miran de arriba abajo y de izquierda a derecha. Intenta utilizarlo a tu favor, posicionando en la parte de arriba y en la izquierda lo que quieres que el usuario vea primero. También puedes usar el conocido neuromarketing.

# Presenta contenidos interesantes y organizados

El contenido es muy importante. Los textos deben ser correctos gramaticalmente, sin faltas de ortografía (recuerda también que lo que los buscadores leen son los textos). Las imágenes y vídeos deben ser de calidad, que susciten interés. Además, es conveniente que todos los contenidos estén ordenados en categorías o secciones para hacer la página más clara y navegable.

# Utiliza vocabulario familiar (que no vulgar)

Siguiendo la línea de la simpleza, es más efectivo el uso de palabras comunes y familiares que el de tecnicismos, extranjerismos o palabras reducidas únicamente al uso profesional. Por otro lado, los textos deben ser claros y el lenguaje informal.

Incluye información de contacto

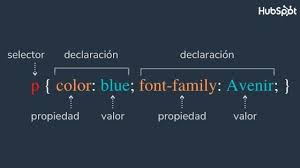
El teléfono, las redes sociales, la dirección para acceder a la navegación web e incluso información sobre los trabajadores de la empresa. De esta manera el usuario estará más predispuesto a dar su voto de confianza. Ya sabe quiénes somos, dónde estamos y cómo puede conocernos.

# Herramientas y lenguajes de programación visual

Las herramientas que utilizamos mucho para programación visual son HTML y CSS.

El lenguaje html (hypertext markup language) se utiliza para el desarrollo y creación de páginas web. Se compone de una serie de etiquetas que el navegador interpreta. Entre las etiquetas que se incluyen dentro del lenguaje HTML se encuentran: hipervínculos, etiquetas para imágenes, saltos de página, entre otras.



CSS son las siglas en inglés para «hojas de estilo en cascada» (Cascading Style Sheets). Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un usuario las visita. Funciona junto con el lenguaje HTML que se encarga del contenido básico de las páginas.

# Estos son algunos de los lenguajes de programación más usados para programación visual son: javascript, c, c++ y php.

JavaScript es el lenguaje de programación que debes usar para añadir características interactivas a tu sitio web, (por ejemplo, juegos, eventos que ocurren cuando los botones son presionados o los datos son introducidos en los formularios, efectos de estilo dinámicos, animación, y mucho más).

C es un lenguaje de nivel medio. Es el lenguaje de los compiladores, intérpretes, editores, sistemas operativos y programación embebida. Cuando se aprende programación en C casi tienes que aprender cómo se ejecutan los programas. Aprendes lo que significan cosas como registro, pila, mapeado de memoria, etc.

C++ es un lenguaje Orientado a Objetos: El foco de la programación está en los objetos y la manipulación y configuración de sus distintos parámetros o propiedades. Rapidez: La compilación y ejecución de un programa en C++ es mucho más rápida que en la mayoría de los lenguajes de programación.

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser insertado en HTML. Muchas aplicaciones web están construidas con PHP.

Tarea 2:

Realiza un mapa mental del tema Diseño de programación visual abarcando los subtemas:

• Concepto de diseño, multimedia e interfaces de programación visual.

• Proceso de diseño e integración de contenidos y componentes.

• Interactividad entre componentes.

• Proceso de construcción de maquetado.

Diagrama

Descripción generada automáticamente

Tarea 3:

Realiza un mapa conceptual del tema Desarrollo de aplicaciones con programación visual desglosando los subtemas:

• Eventos visuales.

•Componentes y herramientas visuales de entornos de desarrollo.

•Propiedades de los componentes visuales de proyectos de software: accesibilidad, apariencia, comportamiento, datos, diseño y estilos.

• Reconocer la funcionalidad de los eventos visuales.

• Proceso de programación visual.

Imagen que contiene Texto

Descripción generada automáticamente

Tarea 4:

Realiza una investigación tomando en cuenta los siguientes temas y puntos.

Conceptos de videojuegos

• Conceptos y tipos de game designer, storyboard.

• Tipos y características de motores de videojuegos y lenguajes de videojuegos

• Metodologías de desarrollo de videojuegos.

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

# CONCEPTOS DE VIDEOJUEGOS

# Conceptos y tipos de game designer, storyboard

Un videojuego es una aplicación interactiva que permite simular experiencia en la pantalla de algún dispositivo electrónico.

Game designer: Con su traducción al español se refiere al diseñador del videojuego, que es el que se encarga de diseñar los elementos que componen el juego, cabe aclarar que en su caso no es necesario que sea un programador, se basa en interfaces del concepto, jugabilidad, mecánica, personajes, escenarios, reglas, niveles, logros, recompensas, habilidades, formas de juego, interacción, entre otros.  
Así mismo debe ser conocedor de las tendencias en videojuegos para crear algo atractivo, llamativo para los consumidores.

# Tipos de Game designer:

* Producer
* Content designe – Encargado del contenido
* Game writer – Escritor de diálogos, creador del guion, historia y concepto
* System designer – Reglas del juego, creador del gameplay
* Level designer – Crea espacio y cuerpo del juego
* Usability designer – Encargado de que el sistema y todo tenga una lógica, que todo funcione correctamente.

Storyboard: Se define como realización de dibujos con secuencia sobre una plantilla, como en las historietas, que contienen textos breves los cuales definen la estructura de la historia.

Es importante gracias a que tiene origen en el cine animado cuando Disney comenzó a usarlo en sus películas en el año 1930, se volvió popular que ahora se usa en videojuegos; es similar a cuando se emplea publicidad, cine o televisión, la diferencia tan importante es que en el caso de los videojuegos se requieren diferentes líneas de acción en función de los movimientos del jugador.

# Tipos de storyboards:

* Diseño del juego: Se establece una plantilla principal donde se tiene que reflejar las diferentes cuadriculas de la base del juego, funciona como estructura, sirve para tener una vista sobre todos los niveles que conformaran el juego.
* Diseño de niveles: Separar y distinguir cada nivel. A partir del storyboard base se crean otros storyboards, uno para cada nivel del juego, detallando acciones, escenas y objetivos.
* Diseño de escenas, objetivos y personajes: Se realizan storyboard de personajes, objetos y entornos del juego, importante ya que los jugadores van a interactuar con todo.

# Tipos y características de motores de videojuegos y lenguajes de video juegos

Motores de videojuegos: Programas especializados de computadores que son utilizados para crear un videojuego, se considera como núcleo que junta todos los elementos necesarios como: las reglas, objetivos, animaciones, diálogos, textos, interfaces, etc. Toda esa información es tomada e interpretada por el motor para crear el ambiente donde el desarrollador pueda interactuar con esos elementos.

# Tipos de motores:

Motor de físicas: Se encarga de realizar cálculos para que un objeto simule tener atributos físicos como peso, volumen, gravedad, aceleración. Hace posible aplicar aproximaciones físicas en la que se obtiene una sensación mas realista al momento de interacción con objetos en el entorno del videojuego.

Motor de sonido: Se encarga de las pistas (sonidos), modifica tasa de bits, la quita de reproducción las sincroniza con otras.

Lenguajes de videojuegos: Elemento fundamental para garantizar una experiencia de juego fluida. Dependiendo del videojuego a crear con grado de interactividad y teniendo en cuenta el dispositivo en el que se jugara determinara el lenguaje de programación a utilizar.

C++: Lenguaje más popular en los títulos AAA, se utiliza para videojuegos de PlayStation y Xbox, es el lenguaje mas compatible con la mayoría de motores de juego y cuenta con un tiempo de ejecución bastante rápido.

Permite tener un control más amplio sobre el hardware, gestión de memoria y gráficos.

C#: Popular en entornos de Windows, menos flexible y compatible, pero uno de los motores como unity permiten programar con el y no esta determinado a un sistema operativo o plataforma. Se puede crear videojuegos para iOS, Android, Windows, PlayStation y Xbox.

Java: Frecuentemente utilizado teniendo bastantes similitudes con C++, es versátil ya que se puede utilizar en todas las plataformas, gran cantidad de frameworks para el desarrollo en 3D, ofrece módulos de código abierto y se puede actualizar constantemente.

JavaScript: Lenguaje mas utilizado en el desarrollo de videojuegos web y de navegador, la mayoría de motores son compatibles, múltiples frameworks para 3D, variedad de bibliotecas y se puede crear toto tipo de scripts dentro del juego gracias a unity.

Python: No es exclusivo para creación de videojuegos, pero es un lenguaje muy flexible y potente. Su ejecución es mas simple, permite plasmar ideas complejas con pocas líneas de código, su framework Pygame permite crear prototipos de los videojuegos de manera rápida y sencilla, funciona en todas las plataformas y sistemas operativos.

Lua: Lenguaje mas sencillo, rápido y fácil de aprender, compatible con lenguajes complejos y de rápida ejecución. Especialmente útil para proyectos independientes y programadores que están empezando.

Metodologías de desarrollo de videojuegos

# Scrum:

Metodología ágil para gerenciar y controlar el desarrollo de software de un producto en forma iterativa e incremental. Una de sus características es que no indica practicas especificas a seguir durante el desarrollo lo cual brinda flexibilidad y permite ajustar el proceso a la realidad y forma de trabajo en cada proyecto, asi como los diferentes requerimientos de los clientes.

Scrum se estructura en tres fases denominadas pre-game, game y post-game.  
Durante el pre-game se define el producto basado en las características conocidas, estimando su tiempo y costo. También se analiza el sistema a construir, se define arquitectura y se realiza un diseño de alto nivel de la solución.

La fase game consta de iteraciones, que duran de dos a seis semanas, donde se desarrollan las características del producto. Al comienzo de cada una se realiza su planificación, donde se describen, priorizan y estiman las características que se van a desarrollar y al concluir se evalúan su resultado.

El post-game es el cierre del proyecto, donde se prepara la liberación del producto, se verifican las versiones a entregar y se realiza la documentación final.

La metodología define tres roles entre los cuales se dividen todas las responsabilidades de un proyecto:

Product owner: esta a cargo del proyecto y es quien maneja y prioriza las características a desarrollar.

Scrum master: es el responsable de que los miembros del equipo sigan el proceso como es debido y de remover los impedimentos que surjan en el transcurso de este.

Scrum team: es un equipo multidisciplinario y auto organizado, su cometido principal es construir el producto que el producto owner especifica.

SUM:

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

SUM fue concebida para que se adapte a equipos pequeños y para proyectos cortos, menores a un año de duración, con alto grado de participación del cliente.

La metodología define cuatro roles: equipo de desarrollo, productor interno, cliente y verificador beta. El productor interno y el cliente se corresponden en forma directa con los roles de Scrum Master y Product Owner de Scrum respectivamente.

El equipo de desarrollo tiene las características del Scrum team, pero a diferencia de Scrum se definen subroles dentro del equipo. Estos se corresponden con los que se utilizan habitualmente en la industria local y son los de programador, artista grafico, artista sonoro y diseñador de juego.

El programador define la arquitectura, realiza el diseño, implementación y verificación de los componentes de software e integra el contenido audiovisual del videojuego. Los subroles de artista grafico y artista sonoro se encargan de la creación del contenido audiovisual del videojuego.

El artista grafico realiza el arte de concepto, el arte 2D, el modelado 3D y la creación de animaciones y texturas.

El artista sonoro se encarga de la creación, grabación, mezcla y edición de los efectos de sonido y música del juego. Diseñador de juego es el encargado de diseñar el gameplay, la historia, el ambiente, los personajes y todos los elementos que hacen a la experiencia del jugador. Además, diseña los niveles, misiones y los desafios que enfrenta el jugador.

El rol de verificador beta no está presente en Scrum, pero si se detecta su existencia en el relevamiento de la realidad local y en la industria del videojuego en general. Su responsabilidad es la de realizar la verificación funcional del videojuego y comunicar su resultado, puede no poseer experiencia ni ser jugador frecuente y participar igualmente de la verificación.

Desarrollo de prototipos de videojuegos

• Concepto, tipos y características de los motores de videojuego. • Integración de motores de videojuegos con programación visual de acuerdo con los requerimientos del videojuego.

• Transición narrativa y lenguaje visual de videojuegos.

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

# Concepto, tipos y características de los motores de videojuego

El término motor de videojuego (en inglés game engine), o simplemente motor de juego, hace referencia a una serie de rutinas de programación que permiten el diseño, la creación y el funcionamiento de un videojuego.

La funcionalidad típica que provee un motor de videojuego incluye: un motor gráfico para renderizar gráficos 2D y 3D, un motor físico que simule las leyes de la física (o simplemente para generar detección de colisiones), animación, scripting, sonidos, inteligencia artificial, redes, retransmisión, gestión de memoria, escenarios gráficos y soporte para lenguaje por secuencia de comandos.

Los motores de videojuegos suelen proporcionar un conjunto de herramientas de desarrollo visual y componentes de software que puedan ser reutilizables.​ Estas herramientas generalmente se proporcionan en un entorno de desarrollo integrado que permiten crear videojuegos de forma rápida y simple a través de una base de datos. En otros casos, los motores se distribuyen con una interfaz de programación de aplicaciones (API) incorporada​ y otros motores, sin embargo, se distribuyen como un conjunto de herramientas que agilizan y simplifican aún más el desarrollo de un videojuego, como por ejemplo: los entornos de desarrollo integrado, scripts preprogramados, y los middlewares (capaces de interconectar varios programas). Esto resulta útil a la hora de conseguir una plataforma de software flexible y reutilizable que evite la compra de otros recursos ajenos, lo cual ayuda a tener todo lo necesario para hacerlo funcional de manera inmediata, reducir los costos, complejidades y tiempos de comercialización.

# 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

Mano sosteniendo un control de televisión

Descripción generada automáticamenteTodos 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 importante de videojuegos para diseñar sus títulos más populares.

Integración de motores de videojuegos con programación visual de acuerdo con los

requerimientos del videojuego

# ¿Qué es Unreal Engine?

Unreal Engine es un motor de juego creado por la compañía Epic Games que fue desarrollado principalmente para los shooters en primera persona, y a partir de ahí ha sido utilizado con éxito en gran variedad de géneros.

Si nos referimos a su última versión, Unreal Engine 4, está diseñada para una gran lista de plataformas: Microsoft Windows, macOS, Linux, SteamOS, HTML5, iOS, Android, PlayStation 4, Nintendo Switch, Xbox One SteamVR/HTC Vive, Oculus Rift, PlayStation VR, Google Daydream, OSVR y Samsung Gear VR.

Antes de referirnos a los requisitos de Unreal Engine vamos a dejar bien claro para qué sirve exactamente este motor gráfico.

Requisitos mínimos para Unreal Engine

OS Windows 7/8 64bit o mayor

Procesador 2.0+ GHz

RAM 2 GB de RAM en sistema

Video compatible con DirectX10 Junio 2010

Disco Duro 15GB de espacio libre

Requisitos para la creación de contenido

OS Windows 7/8 64-bit

Procesador Quad-core Intel o AMD, 2.5 GHz o mayor

RAM 8 GB de RAM en sistema

Video NVIDIA GeForce 470 GTX o AMD Radeon 6850 HD o mayor

Requisitos para desarrollar con Unreal Engine

Windows – Visual Studio 2013 Professional o Visual Studio 2013 Express

iOS – iTunes 11 o mayor / Mac OS X 10.9.2 Xcode 5.0 o mayor

¿Qué es unity?

Unity es una plataforma en la que podemos crear videojuegos, aplicaciones y herramientas para el desarrollo de videojuegos, es uno de los motores para realizar videojuegos.

Con unity puedes desarrollar juegos en 2D y 3D con mucha facilidad, su lenguaje de programación C# es fácil de aprender y además en el mismo sitio web tienes acceso a cursos básicos, inclusive en este sitio web puedes encontrar cursos tanto básicos como algunos dirigidos a hacer algo específico, como mover un fondo de pantalla en unity, además de todo si eres una sola persona y tienes algo de presupuesto, la asset store está llena de recursos que puedes comprar muchas veces a precios Logotipo, nombre de la empresa

Descripción generada automáticamentemuy rebajados y algunos gratis.

Requisitos de Unity Hub

Sistema operativo: Windows 7 sp+1, 8, 10 pero solo en sus versiones de 64 bit, Max OS X 10.2 o superior, Ubuntu 16.04 ó 18.04, CentOS 7

Tarjeta grafica que sea capaz de correr Dx 10.

Requisitos recomendados de unity para desarrolladores

Tabla

Descripción generada automáticamente

# Transición narrativa y lenguaje visual de videojuegos.

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

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.

En esa combinación de narrativa y jugabilidad está la clave y uno de los equipos de desarrollo con más reconocimiento en este campo es Irrational Games está de estreno. Aprovechando el paso de sus miembros por Madrid hemos querido saber más de cómo se construye el guión de un videojuego.

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

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

etapas-para-desarrollar-un-videojuego-factores-de-impacto

Entre los factores de impacto mas comunes tenemos:

Género. Sea un Role Playing Game (RPG o juego de rol por sus siglas en inglés), First Person Shooter (FPS o juego de acción en primera persona), juego de plataformas, puzzles (acertijos) o educativo, etc. El tipo de juego a desarrollar marca de forma importante los fundamentos de producción requeridos.

Portabilidad. Se refiere a las plataformas servicios y sistemas destino donde el videojuego estará disponible: PC, Mac, PlayStation, Xbox, Nintendo, dispositivos móviles, navegadores de Internet, dispositivos de VR etc. Es importante conocer las limitaciones técnicas y las restricciones para el diseño de juego de cada una de ellas; videojuegos que no respeten las respectivas guías de desarrollo corren el riesgo de no ser publicados o incluso prohibidos en ciertas plataformas.

Diseño de Juego (Game design). Se refiere al diseño del videojuego y sus reglas el cual responde preguntas como: ¿De qué se trata el juego? ¿cuál es su lógica? ¿cómo interactúa el usuario? ¿cuáles son los objetivos? ¿cuál es la progresión del personaje? Son el tipo de cuestionamientos que definen la lógica y el funcionamiento de un videojuego y que posiblemente requieran un desarrollo particular.

Presupuesto. El recurso financiero disponible para trabajar el videojuego. Es común que los desarrolladores obtienen inversión parcial o completa de publishers, inversionistas privados o gobierno.

Guía artística visual. Es la línea que dicta como se verá el videojuego y todos sus elementos gráficos y bajo la cual se regirá toda la producción. Ya sea para el desarrollo de un juego hiperrealista AAA o el desarrollo de un juego casual estilizado para niños, la guía de arte debe acomodar las necesidades y expectativas para cada mercado meta.

Equipo de desarrollo. Tanto el presupuesto como el equipo de desarrollo son cruciales elementos para una producción, la falta de alguno de ellos bloquea prácticamente el buen funcionamiento del proyecto.

Milestones

Conocidos en español como hitos, los milestones son puntos claves en el estado de construcción del videojuego. Dichos milestones usualmente son establecidos por los inversionistas o publishers a forma de revisiones que marcan el progreso de la construcción.

Para cada milestone se acuerdan funcionalidades que deben estar presentes en el build del juego, por ejemplo, físicas funcionales, sistema de navegación final, sistema de colisiones activo, etc. En base a esas funcionalidades se establece si el juego va o no en tiempo. Un dato importante: muchos inversionistas o publishers pagan a contra-entrega de milestone completado satisfactoriamente.

Toda producción pasará a lo largo de varias etapas. Cada una de ellas marca un estado específico en su construcción y tiene características particulares.

Las Etapas Del Desarrollo

Como se mencionó, no hay una receta milagrosa que diga al pie de la letra como hacer un videojuego. Sin embargo existen ciertas etapas recurrentes que dan una idea del proceso que conlleva dicha producción.

etapas-para-desarrollar-un-videojuego-fases

Una buena organización Es importante tener una buena organización durante el desarrollo ya que de lo contrario la producción corre el riesgo de salirse de tiempo y presupuesto así como generar mas “bugs” de los necesarios.

Debido a que una gran parte en la producción de videojuegos conlleva un desarrollo de programación, metodologías utilizadas en desarrollo de software es especialmente popular la administración del desarrollo, progreso, debug (depuración) y QA (control de calidad) a través de metodologías de Desarrollo Agile.

# CONCLUSIÓN

Después de terminar nuestras investigaciones se logró comprender la mayoría del o los temas así ayudándonosla a aprender un poco las de la programación visual y de los motores de videojuegos, fue muy interesante ya que se aprendieron cosas nuevas y se reforzó lo que ya sabíamos, gracias a la investigación por fin pude entender como se usa la programación visual y sus componentes, también sirvió para aprender un poco más de los videojuegos.

Es muy interesante lo que se puede hacer en el mundo de la programación y sus infinitas posibilidades sobre todo porque es un tema donde se puede hacer cualquier cosa que te puedas imaginar.

Además de eso aprender todo sobre lo que es la programación orientada a eventos y orientada a objetos es la base para poder realizar los proyectos que tengas en mente.

Esto de la creación de juegos es muy interesante ya que puedes hacer cosas sorprendes gracias a los motores de videojuegos y no solo como un entretenimiento ya que se puede extender a muchas áreas diferentes de nuestro día a día.