EXAMEN

UNIDAD 3

Gomez Romero Enrique (1321124095)

Gonzalez Luna Freddy Daniel (1321124123)

Cristopher Emir García Costales (1321124802)

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.

¿Qué es la programación orientada a objetos?

La POO es un paradigma de programación que se basa en utilizar clases y objetos y de esta manera poder crear una estructura en el programa, dividiéndolo en pequeños trozos de código que son simples y reutilizables.

Para poder utilizar estos objetos que contienen un trozo de código es necesario instanciar individualmente los objetos, dentro de estos se pueden unir otros objetos creando de esta manera una forma de acortar el código y reutilizarlo las veces que sean necesarias.

Características y aplicaciones de eventos

Algunas de las características que existen en la programación orientada a eventos son:

- Eventos: Los eventos son condiciones las cuales se deben de cumplir para poder realizar un proceso especifico.
- No es ningún tipo de tecnología ni lenguaje de programación.
- Usualmente los eventos se manejan mandando a llamar una función o un método.

- Es un estilo de programación.
- Durante una aplicación que esta dirigida a eventos hay un bucle principal que “escucha” las acciones y así manda a llamar los eventos.

Algunas de las aplicaciones de los eventos son:

- Onclick: Se usa para realizar un evento al momento de hacer clic en algún elemento.
- getElementById: devuelve alguna referencia de acuerdo con el ID del elemento
- addEventListener: se utiliza para registrar un evento a un objeto en específico.
- setTimeout: se utiliza para dar una cantidad de tiempo en la cual se realizará el evento

Características de componentes y métodos visuales y no visuales.

Las características de los componentes visuales y la de los no visuales son:

1. Componentes visuales

- Interactúan con el usuario.
- Dan estilo a el programa.
- Componen toda la interfaz del programa.

2. Componentes no visuales

- Son intangibles
- Es la parte lógica del programa
- Se llevan procesos
- Se programan eventos

Procesos de desarrollo visual en proyectos distribuidos y de escritorio.

El proceso para desarrollar un proyecto de manera visual es tener en cuenta cuales son los requerimientos que esta pidiendo en cliente, entre los que se incluyen los colores a utilizar, tipografía, imágenes, videos, audios y en todo lo que se valla a utilizar.

Realizar un proyecto responsivo es lo principal para evitar que el programa final sufra de problemas, los proyectos visuales se destacan por la complejidad del diseño que tiene el proyecto.

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

Requerimientos visuales de proyectos distribuidos y de escritorio.

Los requerimientos de software son las necesidades de los clientes que requiere que el Sistema deba de cumplir de manera satisfactoria. Son los que definen las funciones que el sistema será capaz de realizar.

En este caso los requerimientos que se solicitarían son del campo visual del proyecto, el cual, se nos darán colores, formas, botones, distribución de elementos, inputs, tipografía, etc.

Para la obtención de estos elementos es necesario que el analista junto al cliente realice algunos Muckups de lo requerido cliente.

Herramientas y lenguajes de programación visual.

Bootstrap: Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales.

CSS3: las hojas de estilo en cascada te ayudan con el diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y de esta manera jugar con los tamaños colores y poder acomodar los elementos a gusto propio.

Diseño de programación visual

diseño, multimedia e interfaces de programación visual

Diseño: En la programación visual, sistémicas o geométricas también en donde ilustrar conceptos lógicos y procesos de diagramas de flujo.

Multimedia: conjunto de medios de comunicación que tienen vocación interactiva en un único espacio visual, como puede ser una pantalla o cualquier dispositivo digital.

Interfaces: La interfaz simple y fácil hace que sea más fácil de entender para un usuario principiante y no técnico que un lenguaje de programación regular

Interactividad entre componentes.

los componentes gráficos sencillos y la capacidad de interactuar con plataformas, 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 dieño-, etc).

Los componentes no visuales se pueden colocar en los formularios de la misma manera que los controles, aunque en este caso su posición es irrelevante.

E interactúan todos de forma que nos ayudara en recoger los datos o la acción de cada uno

Proceso de diseño e integración de contenidos y componentes

El diseño de software es un proceso iterativo por medio del cual se traducen los requerimientos en un “plano” para construir el software. Al principio, el plano ilustra una visión holística del software.

El diseño es importante porque permite que un equipo de software evalúe la calidad de éste antes de que se implemente, momento en el que es fácil y barato corregir errores, omisiones o inconsistencias.

Proceso de construcción de maquetado.

1.-·Arquitectura de la información definida: Tener clara la arquitectura de la información te ayudará a estructurar las páginas y los componentes de tu proyecto,

2.-·Flujos de navegación claros: Esto te ayudará a maquetar los componentes en el contexto de la aplicación y te guiará en la toma de decisiones de la estructura de páginas

·3.-¿Qué elementos son reusables en la interfaz?: los elementos que pueden identificarse en el escaneo visual inicial del diseño de la UI.

4.-·Busca constantemente buenas prácticas: os elementos que pueden identificarse en el escaneo visual inicial del diseño de la UI.

5.-·Atención a los detalles: Como maquetadores se vuelve crucial desarrollar el “ojo de píxel”, con el fin de ejecutar interfaces con detalles enfocados a la experiencia final de los usuarios.

DESARROLLO DE APLICACIONES CON PROGRAMACIÓN VISUAL

- Eventos visuales
  - representación gráfica en tiempo de diseño y ejecución
    - Las herramientas de programación visual son más voluminosas y pesadas en tamaño porque están basadas en gráficos
  - La acción podría deberse a la interacción del usuario
  - El objeto que provoca el evento se conoce como emisor del evento
- Componentes y herramientas visuales de entornos de desarrollo
  - Contamos con varios pero los que se llega a conocer son:
    1.-Microsoft Visual Studio
    2.-NetBeans
    3.-Eclipse
- Propiedades de los componentes visuales de proyectos de software
  - accesibilidad
    - programación de bajo código
  - Aperiencia
    - se basan en los gráficos y se componen de iconos, símbolos y bloques lógicos
  - Comportamiento
    - hace que sea más fácil de entender para un usuario principiante
  - datos
    - De normal llegan a ser numéricos , de texto o boléanos
  - diseño y estilo
- Reconocer la funcionalidad de los eventos visuales
  - sencilla disposición visual de las imágenes y los bloques facilita la comprensión, el diseño y la explicación para el usuario
  - cuando un usuario hace clic en un control en un formulario, el formulario puede provocar un evento
- Proceso de programación visual
  - programación visual ayuda a programa a escalar con el uso de su interfaz intuitiva y la fácil representación de los procesos automatizados
    - Los usuarios y desarrolladores pueden crear casi cualquier lógica con el mismo nivel de complejidad y flexibilidad que si lo hicieran con lenguajes de programación

CONCEPTOS DE VIDEOJUEGOS

Tipos de Game Designer:

El game designer se encarga de diseñar la historia, los personajes, los escenarios o las reglas de un videojuego

Son muchos los elementos que integran un videojuego: tecnología, narrativa y arte van de la mano para crear una historia que tiene sus personajes, sonidos, escenarios e interfaces. El game designer asume la labor de crear cada una de esas piezas y que estas encajen a la perfección.

Existen diferentes tipos de diseñadores de juegos, teniendo en cuenta su ámbito de acción:

Lead designer: Suele ser quien tiene la idea inicial o en su conjunto. Supervisa la actividad de todos los equipos, comprobando que trabajan en sintonía.

Game writer: Es el autor de la historia, la persona que escribe el guion.

Level designer: Se encarga del diseño de los mapas, fijando su duración, la dificultad o la ubicación y recorrido de los enemigos.

System designer: Diseña las reglas del juego.

Tipos de Storyboard:

Un Storyboard consiste en la realización de una secuencia de dibujos realizados en una plantilla, acompañados de textos breves que definen la estructura de la historia. Un proceso imprescindible en la fase de preproducción para el desarrollo de Videojuegos que permite previsualizar el resultado final de la producción.

Tipos de Storyboards y técnicas:

En el Diseño del juego: Aquí se establece una plantilla principal donde se refleja en las diferentes cuadriculas la estructura base del juego, la vista previa de todos los niveles que conformarán el juego.

En el Diseño de niveles: A partir del storyboard creado anteriormente, creamos un Storyboard para cada uno de niveles del juego por separado detallando las diferentes acciones, escenas y objetivos del juego. Su función es separar y distinguir cada nivel.

En el Diseño de escenas, objetos y personajes: Realizamos el Storyboard de los personajes, objetos y entornos del juego para describir como interactúan entre sí. El Storyboard de un personaje es muy importante, ya que los jugadores serán los que interactuarán con estos personajes durante el juego.

Tradicional: Pueden ser en blanco y negro, a color, bocetos e incluso ilustraciones más elaboradas, todo en función del artista y las necesidades de la producción.

Digital: Mediante softwares que disponen de herramientas para la creación de Storyboards como es el caso de Toon Boon Storyboard Pro, software estándar en la industria, que permite planificar la animación 2D y la creación de recursos para juegos.

Tipos y características de motores de videojuegos y lenguajes de videojuegos:

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

El aspecto más destacado a la hora de elegir un motor de videojuegos entre todos los disponibles que hay en el mercado son las capacidades gráficas, ya que son las encargadas de mostrar las imágenes 2D y 3D en pantalla, así como calcular algunos aspectos como los polígonos, la iluminación, las texturas

Otras características para tener en cuenta a la hora de la elección son la facilidad de aprender a usar el motor de videojuegos y la facilidad para exportar el juego a diferentes plataformas.

Algunas de las funcionalidades más importantes son:

El motor de físicas

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

El motor de sonido

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

El scripting

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

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

Los motores populares más utilizados y que más posibilidades dan al desarrollador son:

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

Unity 3D: Se trata de una de las innovaciones más importantes creadas por la comunidad científica y de videojuegos y permite jugar a complejos videojuegos en 3D sin necesidad de instalarlos en el ordenador. Los videojuegos creados con el

motor Unity 3D se pueden jugar en un navegador con el reproductor Unity Web Player, eliminando la necesidad de instalar el videojuego.

Frostbite Engine: Este motor para videojuegos creado por Digital Illusions CE se utiliza para crear videojuegos de acción en primera persona. Se presentó principalmente para la serie de videojuegos Battlefield.

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

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

¿Qué lenguaje de programación usar?

Dependiendo del tipo de videojuego que quieras desarrollar, existen diferentes lenguajes de programación. En función del grado de interactividad y el dispositivo que se vaya a utilizar para jugar y la rapidez de respuesta que queramos conseguir, deberemos elegir uno u otro. Además, podemos emplear varios lenguajes si el grado de complejidad del diseño es alto, para que todas las capas del juego funcionen correctamente.

C++

Este lenguaje de programación es uno de los más utilizados en el sector por profesionales. Es un lenguaje popular en los títulos AAA, se utiliza en videojuegos para PlayStation y Xbox, y en juegos independientes. Se trata del lenguaje más compatible con la mayoría de los motores de juego y tiene un tiempo de ejecución bastante rápido

C Sharp

C# es un lenguaje de programación muy popular, sobre todo en entornos Windows. Es un poco menos flexible y compatible que C++, pero algunos motores como Unity permiten programar con él y no está limitado a un determinado sistema operativo o

plataforma; se pueden crear juegos para iOS, Android, Windows Play Station y Xbox. Es un lenguaje más fácil de aprender que el C++.

Java

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

JavaScript

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

Python

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

Lua

Es un lenguaje de programación sencillo, rápido y fácil de aprender. Compatible con lenguajes más complejos y de rápida ejecución, también se usa para aplicaciones web y procesamiento de imágenes. Este lenguaje es especialmente útil para proyectos independientes.

Metodologías de desarrollo de videojuegos.

Las metodologías para el desarrollo de videojuegos serios es un tema sumamente variable, como consecuencia de esto se ha dado pie a la improvisación y a una falta de estandarización a la hora en que estos son desarrollados, por esta razón se decidió elaborar una revisión de las metodologías para el desarrollo de videojuegos serios y así poder determinar su estado.

Agile:

Agile está bien preparado para manejar la inconsistencia y la dificultad durante la participación en proyectos de desarrollo de juegos. Esta metodología fue desarrollada para resolver bloqueos y obstáculos crecientes con Waterfall y otros métodos altamente estructurales e inflexibles. Agile valora las herramientas de comunicación y relación y sus individuos. Es decir, Agile cuenta con la colaboración de los clientes durante todo el proceso de desarrollo. Escucha las respuestas cambiantes en lugar de seguir un único plan constante. Y el motivo de Agile es concentrarse en presentar software funcional en lugar de centrarse en la documentación.

Construye un equipo en iteraciones o sprints cortos, sin un orden específico, cada uno de los cuales contiene una duración definida y una lista de entregables. A través de sprints, el equipo trabaja hacia los objetivos de entregar software que está funcionando. Y, a veces, proporciona una salida beta o alguna otra cosa tangible.

El enfoque Agile prioriza en gran medida la satisfacción de los clientes. Esto puede ser posible cuando el equipo le entrega continuamente resultados de prueba, trabajo y características priorizadas.

Cascada

Si hablas de desarrollo de software de juegos. La Cascada es la opción más antigua y consecutiva. Sin embargo, normalmente se considera un método obsoleto. Sin embargo, es útil comprender la estructura y la historia reales para mejorar la flexibilidad de las metodologías más modernas. Pero no tiene el potencial como Agile para manejar las complejidades.

La metodología de Waterfall consiste en desarrollar el enfoque en una serie de etapas en las que se desarrollan los proyectos del juego. Cada paso conduce a una etapa sucesiva, y cada paso es más complejo que el anterior, lo que lo hace costoso. El concepto se crea a partir de una idea, y esto inicia el proceso de un proyecto. Una vez que el concepto coincida con los requisitos definidos para la etapa de la idea, el proyecto pasará a la siguiente etapa.

Programación extrema (XP)

La programación extrema (XP) es una metodología de desarrollo de juegos y software del marco ágil. El motivo de la programación extrema (XP) es desarrollar software de alta calidad y juegos con el mejor rendimiento. La programación robusta (XP) sigue el conjunto de valores: comunicación, sencillez, valentía, retroalimentación y respeto.

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

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

Aspectos a tomar en cuenta:

Entorno/Plataforma: Se debe entender dónde se va a jugar el juego que estás diseñando. Debes tener en cuenta las posibilidades y limitaciones que te ofrece la plataforma. No es lo mismo hacer juegos para smartphones que para una consola o que para un PC.

Contenido: Un buen diseño de UI proporciona al jugador toda la información necesaria para que pueda interactuar con el juego y que todo sea fluido.

Diseño Visual: Los videojuegos, casi siempre, entran por los ojos. Un apartado visual feo o denso en la interfaz del juego puede resultar contraproducente y sacar

al jugador de la experiencia inmersiva que quieres proporcionarle. Debes definir el estilo de arte.

Arquitectura de la información: Definir qué elementos son de mayor o menor importancia para el usuario y organizarlos de tal forma que todo resulte en un diseño de interfaz coherente y relevante.

Estilos de diseño de la interfaz de un videojuego

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

Diegéticas

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

Ejemplos:

Dead Space. En el juego de Visceral podemos ver, por ejemplo, la barra de vida del personaje integrada dentro del traje que lleva puesta. O como su inventario se despliega a través de un dispositivo que el protagonista lleva integrado y se muestra ante él sin detener la acción del juego.

Fallout. Para este caso nos podemos centrar en el uso que se hace del PipBoy, un dispositivo que nuestro protagonista lleva integrado en el brazo y que permite que accedamos al inventario, diario de misiones y radio, entre otras opciones.

Assassins Creed. En los populares juegos de Ubisoft, los protagonistas usan un águila a través de la que, como jugadores, podemos marcar enemigos en el mapa.

No diegéticas

Cuando hablamos de diseño de interfaces no diegéticas o extra diegéticas, nos estamos refiriendo a los elementos que se muestran fuera del mundo del juego y

que solo son visibles y audibles para el jugador. Aunque las cosas van cambiando en este sentido, se trata del diseño de interfaz más ampliamente usado en los videojuegos.

Ejemplos:

GTA. Igual que pasa con Fallout, los GTA tienen una mezcla de tipos de interfaces. En lo que nos ocupa, podemos tomar como ejemplo la presencia del minimapa o el despliegue de un menú cuando tenemos que seleccionar un arma. Elementos que facilitan la interacción del jugador, pero de los que el protagonista no es consciente.

Espaciales

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

Ejemplos de este tipo de diseño de interfaz en un videojuego serían los marcadores sobre la cabeza de los enemigos o las señalizaciones sobre el mapa

Meta

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

Por poner algunos ejemplos concretos, en The Last of Us puedes sentir la presencia de los enemigos a través de las paredes. El cómo eso se muestra en pantalla es un ejemplo de diseño de interfaz de tipo meta. Forma parte de la narrativa, favorece la inmersión del jugador, y no está dentro del espacio del juego, sino que forma parte de cómo percibe el mundo el protagonista.

Desarrollo de prototipos de videojuegos

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

Tipos:

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

Unity

Es uno de los mejores motores 2D y 3D, ofreciendo respuesta a las necesidades de los desarrolladores independientes, con licencia gratuita y precios asequibles. Permite exportar a un montón de sistemas y tiene soporte para DirectX 11, mecanim animation

Godot

Es un motor gráfico gratuito de código abierto que, según su propia descripción, "proporciona una gran variedad de herramientas comunes" para que "puedas simplemente enfocarte en hacer tu juego sin tener que reinventar la rueda". Posee una interfaz sencilla para que no te pierdas por sus menús y que además permite diseñar videojuegos sin que tengas que adquirir conocimientos de programación.

Unreal Engine

Uno de los motores más potentes del panorama y por ello también uno de los que quizás pueda parecer menos adecuado para las personas con menos experiencia. Unreal Engine en realidad una herramienta muy útil si queremos trabajar con elementos predefinidos y conseguir resultados impresionantes. Cuenta con una sección dedicada exclusivamente a enseñar a los usuarios a moverse por sus interfaces.

Características:

Programa de juego principal

La lógica del videojuego debe ser implementada a través de diversos algoritmos. Es una estructura distinta de cualquier trabajo de renderizado o de sonido.

Renderización

La renderización es el proceso en el cual se generan los gráficos 3D por computadora a fin de mostrar en pantalla el aspecto visual del videojuego. Genera gráficos en 3D por varios métodos (como la rasterización gráfica, el trazado de rayos, la partición binaria del espacio, entre otros) y se ocupa de mostrar escenarios, modelos, animaciones, texturas, sombras, iluminaciones y materiales.

Polígonos

Todo elemento que tenga una composición tridimensional es clasificado por polígonos. Es el procedimiento más primitivo en la creación de un mundo tridimensional y su complejidad geométrica varía de acuerdo con las capacidades técnicas del hardware. Dicha complejidad se puede clasificar en una composición poligonal baja, media y alta. Esto dará como resultado un determinado nivel de detalle

Audio: El audio de un videojuego se llega a manejar de muchas maneras y esto depende de las capacidades que tenga el motor. Hoy en día los motores de última generación soportan muchos formatos de sonido, siendo los más populares el WAV y el OGG.

Motor físico

El motor físico es responsable de emular las leyes de física en forma realista dentro del motor de videojuego. Específicamente, proporciona un conjunto de funciones para simular acciones reales a través de variables como la gravedad, la masa, la fricción, la fuerza, la flexibilidad y las colisiones, actuando sobre los diversos objetos dentro del juego al momento de su ejecución.

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

Dependiendo de los requerimientos que tenga el videojuego, la estructura del contenido y la arquitectura de la información y los objetivos que quiera cumplir el trabajo final se deberá escoger un motor que cumpla con estas expectativas y que permitan al equipo de desarrollo lograr producir el videojuego basándose en los objetivos que tiene la programación visual, tales como que las interfaces de usuario sean funcionales y que sean intuitivas para el jugador así como también analizar la renderización y carga de los contenidos para que se logre ejecutar según los requerimientos físicos del dispositivo en el que se ejecutara el videojuego.

Transición narrativa y lenguaje visual de videojuegos

El estilo artístico de un videojuego puede suponer que un jugador quiera saber más de él o no. El arte debe reflejar el espíritu del juego y responder a su jugabilidad, pero también hay que tener en cuenta los medios y el tiempo del que se dispone.

Los videojuegos y la fotografía son campos que están totalmente unidos, ya sea por lo visual y artístico o por lo técnico

Para tratar estos temas se han desglosado los videojuegos según su objetivo como obra. Primero, en los juegos que no tienen un lenguaje visual con una intención clara pero sí que tienen un uso del color inteligente, normalmente los propios estudios son los que toman estas elecciones sin necesitar la ayuda de profesionales de la fotografía.

Minecraft es un juego que destaca por su sencillez, por eso le acompaña una paleta de colores tan reducida, los verdes del césped son un conjunto de píxeles de tres o cuatro tipos verdes diferentes, lo suficiente para identificarlo junto al verde de los árboles.

League of legends es un juego competitivo y tiene unos colores suaves, posiblemente para que el jugador pueda estar más tiempo jugando sin saturarse.

Por otra parte, tenemos Hollow Knight, un juego desarrollado por solo 4 personas que no cuenta su historia de manera directa, son los jugadores quienes tienen que ir investigando y relacionando las cosas que se narran y se ven dentro del juego, aquí es donde cobra mayor importancia la fotografía. Las imágenes se usan para expresar algo y el juego consigue un estilo visual interesante. Los colores nos ayudan de manera inconsciente a saber en qué zona del mapa estamos y si esta tiene elementos peligrosos, pero también tiene una intención narrativa.

La ciudad principal, por ejemplo, destaca por su azul oscuro y desaturado, la infección que acabó con esta es de color naranja, su color complementario.

Los videojuegos forman parte de la cultura audiovisual, la fotografía de estos representa una idea y cada jugador la interpreta de manera individual. Es innegable la importancia de la fotografía a la hora de crear un videojuego, sobre todo si tienen un propósito artístico, en las grandes producciones el apartado visual es casi un concepto diferente, cuentan con más elementos porque se pretende hacer más realista y esto se consigue equipos enormes de gente trabajando desde la creación de una hamburguesa hasta la representación del viejo oeste, pasando por un montón de carteles que encontramos en los mapas y que dan vida al juego.

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

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

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

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

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

Concepción:

Se deberán plantear los aspectos fundamentales que conformarán el videojuego, entre los que se encuentran:

Género: dentro de qué géneros se va a desarrollar el juego. De no corresponder a un género muy conocido, se deben especificar las características.

Gameplay: lo que generará diversión a la hora de jugarlo.

Conceptos: algunas ideas sueltas acerca de cómo debe lucir el juego en cuanto a personajes, ambientación, música, etc.

Diseño

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

Historia: forma en que se desenvolverán los personajes del juego y la historia del mundo representado. Casi todos los juegos tienen historia.

Guion: el proceso comienza con una reunión de todo el equipo de desarrollo, para que todo el equipo tenga la oportunidad de aportar sus ideas o sugerencias al proyecto. A partir de aquí el equipo de guion trabaja por conseguir un borrador en el que queden plasmados cuáles serán los objetivos en el juego, las partes en las que se dividirá, el contexto en el que se desarrollará la acción, cuales, y cómo serán los personajes principales del juego, etc.

Arte conceptual: se establece el aspecto general del juego. En esta etapa un grupo de artistas se encargan de visualizar o conceptualizar los personajes, escenarios, criaturas, objetos, etc. Estos artistas se basan en las ideas originales de los creadores y luego entregan una serie de propuestas impresas o digitales de cómo lucirá el juego. Posteriormente, el director de arte se encargará de escoger de entre las opciones aquellas que se apeguen más a la idea original.

Sonido: detallada descripción de todos los elementos sonoros que el juego necesita para su realización. Voces, sonidos ambientales, efectos sonoros y música.

Mecánica de juego: es la especificación del funcionamiento general del juego. Es dependiente del género y señala la forma en que los diferentes entes virtuales interactuarán dentro del juego, es decir, las reglas que rigen este.

Diseño de programación: describe la manera en que el videojuego será implementado en una máquina real (PC, consola, móviles, etc) mediante un cierto lenguaje de programación y siguiendo una determinada metodología. Generalmente en esta fase se generan diagramas de UML que describen el funcionamiento estático y dinámico, la interacción con los usuarios y los diferentes estados que atravesará el videojuego como software.

Pruebas

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

Pruebas físicas: se llevan a cabo por los diseñadores y programadores del juego. Se crean prototipos que simulan los eventos que pueden suceder en el juego. Un prototipo físico puede utilizar papel y lápiz, tarjetas de índice, o incluso ser actuado fuera. Sobre la base de los resultados de estas pruebas se puede hacer una mejor aproximación al balance del videojuego, pueden prevenir problemas de programación. El objetivo es jugar y perfeccionar este simplista modelo antes de que un solo programador, productor o artista gráfico estén cada vez más introducidos en el proyecto. De esta manera, el diseñador del juego recibe retroalimentación instantánea en lo que piensan los jugadores del juego y pueden ver inmediatamente si están logrando sus metas.

Pruebas alpha: se llevan a cabo por un pequeño grupo de personas, que con anterioridad estén involucradas en el desarrollo, lo que puede incluir artistas, programadores, coordinadores, etc. El propósito es corregir los defectos más graves y mejorar características de jugabilidad no contempladas en el documento de diseño.

Pruebas beta: estas pruebas se llevan a cabo por un equipo externo de jugadores, bien sea que sean contratados para la ocasión o que sean un grupo componente del proyecto (grupo QA). De estas pruebas, el videojuego debe salir con la menor cantidad posible de defectos menores y ningún defecto medio o crítico.

Mantenimiento

Una vez que el juego alcanza su versión final (RTM) y se publica, aparecerán nuevos errores o se detectarán posibles mejoras. Es necesario recopilar toda la

información posible de los jugadores y a partir de ahí realizar los cambios oportunos para mejorar el juego en todos sus aspectos, ya sea de diseño, jugabilidad, etc. Estas correcciones o mejoras se hacen llegar a los usuarios en forma de parches o actualizaciones, que en ocasiones pueden incluir algunas características nuevas para el juego.

Funcionalidad

La funcionalidad es un factor clave para el desarrollo, la producción y el lanzamiento del juego. La funcionalidad se logra cuando, a través de la corrección de las mecánicas y dinámicas del juego, un jugador, sin ayuda de los desarrolladores, puede jugar sin ningún problema. Este punto se lleva a cabo desde la concepción y el desarrollo de las pruebas del juego, llevando las pruebas a cabo es la única manera de refinar el proceso de funcionalidad. Un ejemplo claro de que el juego es funcional es cuando en las pruebas los jugadores pueden completar las metas o logros esperados en el tiempo que se planeó con las instrucciones que el mismo juego les brinda.

Referencias

El Storyboard en Diseño de Videojuegos | Arteneo. (2022). Retrieved 30 July 2022, from https://www.arteneo.com/blog/storyboard-videojuegos-escuela-madrid/

Garcia, A. (2022). Los 6 mejores lenguajes de programación para videojuegos. Retrieved 29 July 2022, from https://profile.es/blog/lenguajes-programacion-videojuegos/

Garcia, T. (2022). Top 10: Los motores gráficos más importantes – NeoTeo. Retrieved 30 July 2022, from https://www.neoteo.com/top-10-los-motores-graficos-mas-importantes/

Garit, F. (2022). Retrieved 30 July 2022, from https://repositorio.una.ac.cr/bitstream/handle/11056/22597/55-Texto%20del%20art%C3%ADculo-108-1-10-20210102.pdf?sequence=1&isAllowed=y

School, T. (2022). El diseño de interfaz de usuario en un videojuego | Tokio School. Retrieved 29 July 2022, from https://www.tokioschool.com/noticias/diseno-interfaz-videojuego/

Wajid, H. (2022). Research Methodology During Game Development. Retrieved 28 July 2022, from https://itchronicles.com/agile/research-methodology-during-game-development/