

Soporte para el cuestionario de admisión de Arte Digital 2024

Perfil de ingreso

El aspirante a ingresar a esta carrera debe, además de haber cumplido satisfactoriamente con sus estudios de bachillerato, poseer espíritu lúdico, haber tenido experiencias con el arte, tener interés por el mundo digital, además de poseer los siguientes atributos:

Conocimientos:

- De las ciencias exactas, especialmente física y matemáticas.
- De la historia de México y del mundo, para permanecer consciente de su situación sociohistórica
- De la multiculturalidad planetaria y nacional, para reconocerla y apreciarla más allá de los prejuicios etnocéntricos

Habilidades:

- De lectura y comprensión de textos
- Del idioma inglés
- De comunicación interpersonal
- Del uso de las computadoras
- De autoaprendizaje

Actitudes y Valores:

- Disposición para integrarse a equipos de trabajo interdisciplinarios
- Capacidad autocrítica
- Respeto y tolerancia a la diversidad
- Conciencia de que el conocimiento, la imaginación y la creatividad son productos (y procesos) de creación colectiva

Perfil de egreso

El Licenciado en Arte Digital (LAD) de la BUAP busca un cambio positivo del mundo mediante experiencias sensibles que se realizan a través del arte y la tecnología, transformando al ser humano y sus formas de relacionarse.

Es un profesional que posee los conocimientos y habilidades artísticas, científicas y tecnológicas necesarias para realizar proyectos audiovisuales e interactivos, trabajando en equipos multidisciplinarios e impulsando su disciplina a través de la investigación y la innovación a partir de lo digital.

Sus principales formas de expresión son el desarrollo de videojuegos y animaciones, así como los procesos creativos que éstas involucran, lo que le permite desempeñarse en áreas relacionadas como la creación de efectos visuales, arte secuencial, diseño de personajes y escenarios, aplicaciones de realidad mixta y diversos tipos de arte experimental e interactivo.

Competencias Específicas:

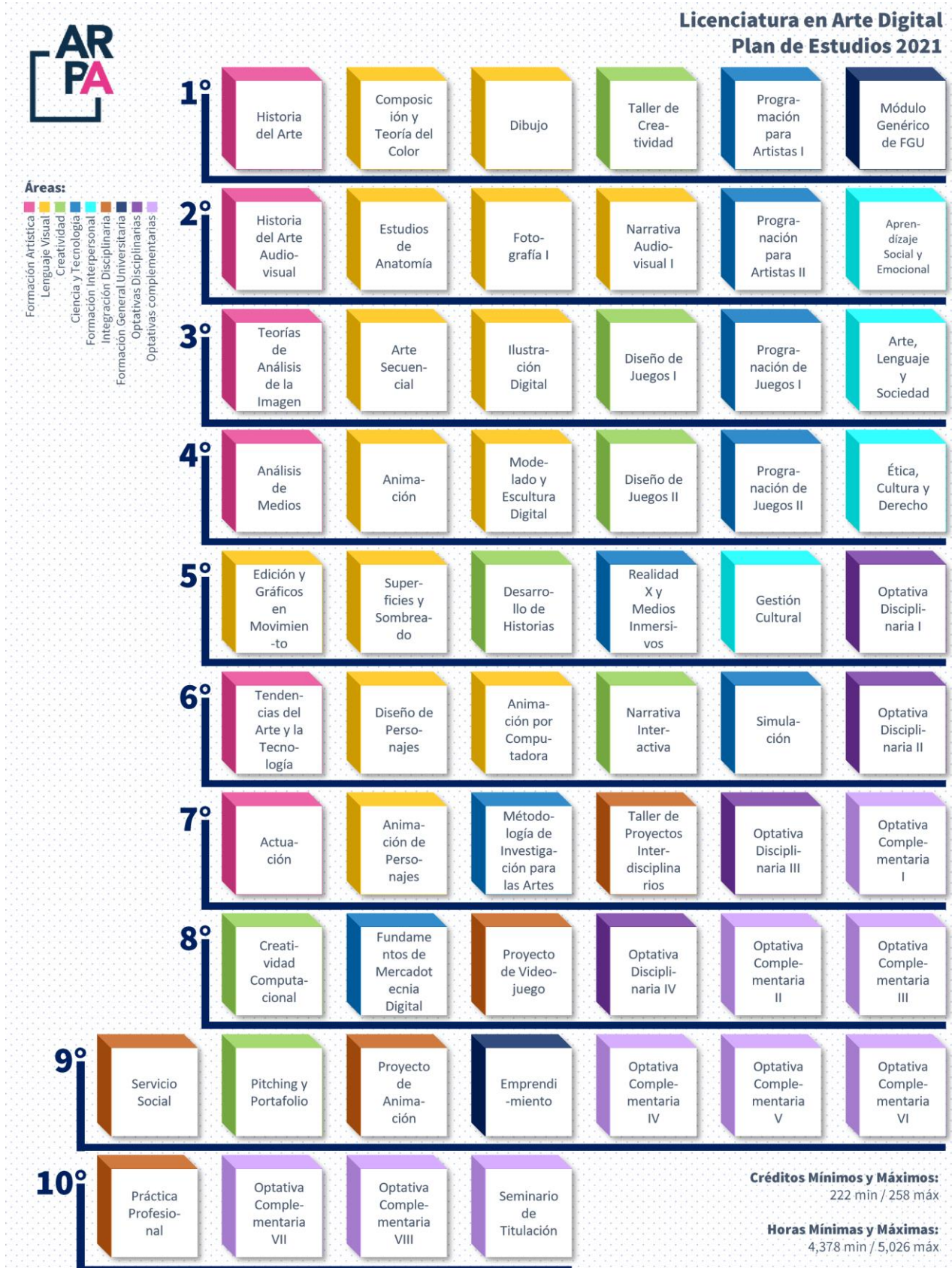
- Analiza la naturaleza del movimiento y la imagen para comunicar historias, ideas y emociones a través de representaciones gráficas y reflexiones estéticas.
- Sistematiza los elementos de experiencias audiovisuales y/o interactivas para crear videojuegos, simulaciones por computadora, expresiones artísticas y herramientas digitales, desarrollando un pensamiento lógico y matemático.
- Crea representaciones digitales de fenómenos reales o imaginarios aplicando un pensamiento computacional, para lograr una interacción entre estas representaciones y el ser humano, y así, transformar su sensibilidad e inteligencia.
- Maneja de manera profesional tecnología especializada (software y hardware) para la creación de arte digital, como aplicaciones de creación de contenido digital, motores de juegos, entre otros.
- Utiliza sus conocimientos teóricos, técnicos y tecnológicos para la resolución de problemas de forma creativa e innovadora, ejecutando proyectos artísticos de impacto social y cultural.
- Desarrolla habilidades sociales y emocionales, a través de la sensibilidad y la apreciación artística, para lograr metas personales y colectivas, establecer y mantener relaciones de apoyo, y tomar decisiones responsables.
- Desarrolla competencias profesionales de forma autodidacta para mantenerse a la vanguardia de las técnicas, conocimientos y habilidades propias de su disciplina.

Campo de trabajo

- En empresas especializadas en la producción de videojuegos, animaciones, cine y audiovisuales, ya sea de contenido artístico o publicitario.
- En el sector educativo, en la realización de materiales audiovisuales interactivos para fomentar el aprendizaje, laboratorios virtuales y similares, o bien, como docente o investigador del área.
- En empresas de base tecnológica realizando propuestas de simulación de procesos
- En organizaciones públicas o privadas, ofreciendo soluciones lúdicas a problemáticas específicas para incrementar la productividad o las ventas (gamification).
- Como profesional independiente prestando servicios de creación audiovisual, o bien, formando su propio estudio o empresa dentro de las industrias creativas.

La información completa del plan de estudios la puedes acceder en <https://arpa.buap.mx>

Mapa curricular



Legislación universitaria

Puedes consultar el Reglamento de Requisitos y Procedimientos para la Admisión, Permanencia y Trayectoria Académica de los Alumnos de Modalidad Escolarizada de la BUAP en

https://repositorio.buap.mx/rdocencia/public/inf_public/2019/0/Reglamento-Admision-Alumnos-Modalidad-Escolarizada.pdf

El 11 de noviembre de 2019, luego de una serie de acciones institucionales se publicó el PROTOCOLO PARA LA PREVENCIÓN Y ATENCIÓN DE LA DISCRIMINACIÓN Y VIOLENCIA DE GÉNERO EN LA BUAP con el objetivo general de establecer la coordinación de instancias y autoridades universitarias necesarias para prevenir, sancionar, erradicar la discriminación y violencia de género y consolidar la cultura de perspectiva de género, estableciendo los mecanismos institucionales, a fin de garantizar a la comunidad universitaria una vida incluyente y libre de violencia. Puedes consultarlo en:

<https://diige.buap.mx/sites/default/files/protocolo%20completo.pdf>

Apéndice A. Referencia sobre el diseño y programación de videojuegos

Programación de computadoras

La programación es el proceso de crear un conjunto de instrucciones que le digan a una computadora cómo realizar una tarea. La programación se puede hacer al usar una variedad de lenguajes de programación como SQL, Java, Python y C++.

Tipos de Datos

Un tipo de datos es la propiedad de un valor que determina su dominio (qué valores puede tomar), qué operaciones se le pueden aplicar y cómo es representado internamente por el computador.

Todos los valores que aparecen en un programa tienen un tipo.

Para programar, es necesario identificar a qué tipo de datos pertenecen cada una de las variables o constantes, ya sean estos números, letras, palabras, frases, entre otros y cada uno corresponde a un tipo de dato, que requerirá de una manipulación específica para obtener los resultados deseados. Cada lenguaje de programación dispone de un conjunto de tipos de datos que puede manipular; sin

embargo los que aquí se presentan corresponden a aquellos que todos los lenguajes de programación disponen, y que se conocen como tipos de datos estándares. Los tipos de datos más usados son:

Numéricos

- **Enteros.** Este tipo dato corresponde a aquellas variables que exclusivamente pueden recibir VALORES SIN PARTE DECIMAL. Generalmente se utilizan en las variables que contienen cantidades de elementos que no pueden fraccionarse, como el número de personas, el número de edificios, entre otros.
- **Flotantes.** Este tipo de dato, en cambio corresponde a aquellas variables cuyos VALORES TIENEN PARTE DECIMAL, aunque al mismo tiempo puede contener valores enteros. Así el lado de un triángulo puede ser 3.5 (0.5 es la parte decimal), pero también podría haber tomado un valor de 3.0 (0.0 es la parte decimal).

Texto

- **Caracteres.** En este tipo de dato se encuentran todos los caracteres conocidos, una letra, un número, un símbolo especial. Por lo tanto, está conformado por los dígitos: '0', '1', '2', ... , '9'; letras: 'a', 'b', 'c', ... , 'z'; mayúsculas: 'A', 'B', 'C', ... , 'Z'; y caracteres especiales: '%', '*', '?', ... , '/'. En algunos lenguajes de programación como Java y C#, se utiliza la comilla simple (' ') para identificar un carácter, sin embargo esto puede cambiar dependiendo del lenguaje de programación.
- **Cadenas.** Constituyen conjuntos de caracteres, es decir la UNIÓN DE VARIOS CARACTERES, que pueden ser palabras o frases. El valor de este tipo de datos se encierra generalmente entre comillas (" ").

Lógicos

- **Booleanos.** Los booleanos o tipos de datos lógicos, únicamente reciben dos valores: verdadero o falso. Se utilizan generalmente como banderas, para identificar si se realizó o no un proceso.

Instrucciones de decisión

Las sentencias de decisión o también llamadas de CONTROL DE FLUJO son estructuras de control que realizan una pregunta cuya respuesta es verdadero o falso (evalúa una condición) y selecciona la siguiente instrucción a ejecutar dependiendo del resultado.

Con frecuencia, en un programa de computadora es preciso cambiar el flujo de ejecución de las instrucciones, es decir, el orden en que las instrucciones son ejecutadas. Muchas veces tenemos que tomar una decisión en cuanto a qué se debe ejecutar basándonos en una respuesta de verdadero o falso.

La ejecución de las instrucciones incluyendo una estructura de control como el condicional funciona de esta manera:

- Las instrucciones comienzan a ejecutarse de forma secuencial (en orden) y cuando se llega a una estructura condicional, se decide qué camino tomar dependiendo del resultado de la condición (falso o verdadero).
- Cuando se termina de ejecutar este bloque de instrucciones se reanuda la ejecución en la instrucción siguiente a la instrucción condicional.

Ciclos

Las Sentencias de Iteración o Ciclos son estructuras de control que repiten la ejecución de un grupo de instrucciones. Básicamente, una sentencia de iteración es una estructura de control condicional, ya que dentro de la misma se repite la ejecución de una o más instrucciones mientras que una condición específica se cumpla. Muchas veces tenemos que repetir un número definido o indefinido de veces un grupo de instrucciones por lo que en estos casos utilizamos este tipo de sentencias.

En C++ los ciclos o bucles se construyen por medio de las sentencias `for`, `while` y `do - while`. La sentencia `for` es útil para los casos en donde se conoce de antemano el número de veces que una o más sentencias han de repetirse. Por otro lado, la sentencia `while` es útil en aquellos casos en donde no se conoce de antemano el número de veces que una o más sentencias se tienen que repetir.

Clases

Una clase es una construcción que le permite al programador crear sus tipos de datos personalizados, agrupando las variables de otros tipos, métodos y eventos. Una clase es como un plano, define los datos y el comportamiento de un tipo.

Los objetos son instancias de una clase, por ejemplo, se puede tener una clase `Aspirante` y tú serías una instancia de esa clase.

Motor de juego

Al igual que ocurre en otras disciplinas en el campo de la informática, el desarrollo de videojuegos se ha beneficiado de la aparición de herramientas que facilitan dicho desarrollo, automatizando determinadas tareas y ocultando la complejidad inherente a muchos procesos de bajo nivel.

El término motor de juego surgió a mediados de los años 90 con la aparición del famosísimo juego de acción en primera persona Doom, desarrollado por la compañía id Software bajo la dirección de John Carmack. Esta afirmación se sustenta sobre el hecho de que Doom fue diseñado con una arquitectura orientada a la reutilización mediante una separación adecuada en distintos módulos de los componentes fundamentales, como por ejemplo el sistema de renderizado gráfico, el sistema de detección de colisiones o el sistema de audio, y los elementos más artísticos, como por ejemplo los escenarios virtuales o las reglas que gobernaban al propio juego.

Este planteamiento facilitaba enormemente la reutilización de software y el concepto de motor de juego se hizo más popular a medida que otros desarrolladores comenzaron a utilizar diversos módulos o juegos previamente licenciados para generar los suyos propios. En otras palabras, era posible diseñar un juego del mismo tipo sin apenas modificar el núcleo o motor del juego, sino que el esfuerzo se podía dirigir directamente a la parte artística y a las reglas del mismo.

Arquitectura de juegos

Las aplicaciones modernas de interfaz gráfica de usuario son sorprendentemente similares a los antiguos juegos de aventuras una vez que les quitas la piel. Por lo general, su procesador de textos se queda allí sin hacer nada hasta que presiona una tecla o hace clic en algo.

La principal diferencia es que en lugar de comandos de texto, el programa está esperando eventos de entrada del usuario: clics del mouse y pulsaciones de teclas. Todavía funciona básicamente como las antiguas aventuras de texto donde el programa se bloquea mientras espera la entrada del usuario, lo cual es un problema.

A diferencia de la mayoría de los demás software, los juegos se siguen moviendo incluso cuando el usuario no está proporcionando información. Si te sientas mirando la pantalla, el juego no se congela. Las animaciones siguen animando. Los efectos visuales bailan y brillan. Si no tienes suerte, ese monstruo sigue mordiendo a tu héroe.

Esta es la primera parte clave de un ciclo de juego real: procesa la entrada del usuario, pero no la espera. El ciclo siempre sigue girando.

Un ciclo de juego se ejecuta continuamente durante el juego. Cada vuelta del ciclo procesa la entrada del usuario sin bloquearse, actualiza el estado del juego y lo renderiza. Rastrea el paso del tiempo para controlar la velocidad de juego.

Diseño de juegos

Los diseñadores de juego son los responsables de diseñar el contenido del juego, destacando la evolución del mismo desde el principio hasta el final, la secuencia de capítulos, las reglas del juego, los objetivos principales y secundarios, etc. Evidentemente, todos los aspectos de diseño están estrechamente ligados al propio género del mismo. Por ejemplo, en un juego de conducción es tarea de los diseñadores definir el comportamiento de los coches adversarios ante, por ejemplo, el adelantamiento de un rival.

Los diseñadores suelen trabajar directamente con los ingenieros para afrontar diversos retos, como por ejemplo el comportamiento de los enemigos en una aventura. De hecho, es bastante común que los propios diseñadores programen, junto con los ingenieros, dichos aspectos haciendo uso de lenguajes de scripting de alto nivel, como por ejemplo LUA o Python.

Como ocurre con las otras disciplinas previamente comentadas, en algunos estudios los diseñadores de juego también juegan roles de gestión y supervisión técnica.

Finalmente, en el desarrollo de videojuegos también están presentes roles vinculados a la producción, especialmente en estudios de mayor capacidad, asociados a la planificación del proyecto y a la gestión de recursos humanos. En algunas ocasiones, los productores también asumen roles relacionados con el diseño del juego. Así mismo, los responsables de marketing, de administración y de soporte juegan un papel relevante. También resulta importante resaltar la figura de publicador como entidad responsable del marketing y distribución del videojuego desarrollado por un determinado estudio. Mientras algunos estudios tienen contratos permanentes con un determinado publicador, otros prefieren mantener una relación temporal y asociarse con el publicador que le ofrezca mejores condiciones para gestionar el lanzamiento de un título.

Digamos que un diseñador de juegos siente la necesidad de crear un juego basado en las historias específicas de la mitología griega. Esto comenzaría a partir de una historia. Inmediatamente esto limita el tipo de juego al que irá. Lo más probable es que se haya eliminado un juego de estrategia estilo "Civilization", ya que ese tipo de juego en realidad no tiene nada que ver con las historias clásicas de Zeus, Heracles, Ares, etc. Tampoco se puede jugar un juego de estrategia en tiempo real, ya que no es bueno contar historias con solo unos pocos protagonistas. Un simulador de vuelo de alta gama probablemente tampoco funcionará. El diseñador podría, sin embargo, perseguirlo a través de un juego de acción, un juego de rol o un juego de aventura. Del mismo modo, la tecnología es limitada. Para contar la historia de los dioses griegos, el diseñador necesitará alguna forma de comunicarle al jugador una gran cantidad de información de trasfondo. Por lo tanto, será necesario contar con tecnología que permita esto.

Además, si el diseñador elige la tecnología que utilizará el juego en este punto, esto tendrá un impacto aún mayor en el tipo de juego que será posible.

Por ejemplo, elegir un motor 2D isométrico se presta mejor a un RPG o un juego de aventura en lugar de un juego de acción, a menos que uno pretenda ser deliberadamente "retro" y opte por una aventura de acción 2D en el espíritu de "Crusader: No Remorse". Si utilizo la tecnología 3D, para contar la historia de la mitología griega de forma adecuada tendrá que admitir entornos interiores y exteriores, lo que elimina inmediatamente muchos motores de juegos en 3D.

Por cada decisión que tome el diseñador sobre el juego que espera crear, debe entender cómo eso limita el juego. Si el diseñador intenta adaptarse a un tipo de juego en torno a un motor inadecuado, el juego sufrirá al final. Tratando de hacer un "simulador de dios" de estilo "Populous" utilizando un motor 3D de estilo irreal en primera persona es un gran error. Al igual que si uno tratara de contar la historia de los dioses griegos a través del juego de simulación de vuelo, el juego simplemente no funcionaría. Aquí radica la dificultad con muchas ideas de "alto concepto", a menudo las mentes de los especialistas en marketing que quieren capturar mercados dispares con un producto. Si las piezas no funcionan juntas, no importa cuántos mercados cubra el concepto: ningún jugador estará interesado en jugar el juego final.

Matemáticas para videojuegos

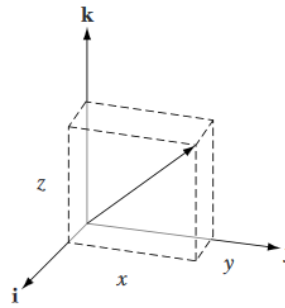
Vectores

Los vectores son de fundamental importancia en cualquier motor de juegos en 3D. Se utilizan para representar puntos en el espacio, como la ubicación de objetos en un juego o los vértices de una malla triangular. También se utilizan para representar direcciones espaciales, como la orientación de la cámara o las normales de superficie de una malla triangular. Comprender cómo manipular vectores es una habilidad esencial del programador 3D exitoso.

Para los matemáticos, un vector es una lista de números. Los programadores reconocerán el sinónimo de arreglo de términos. Observa que la clase de matriz de plantillas STL en C++ se llama vector, y la clase básica de contenedor de matriz de Java es `java.util.Vector`. Entonces, matemáticamente, un vector no es más que un arreglo de números.

Podemos representar de forma única cualquier vector v en el espacio 3D mediante el uso de la combinación lineal $v=xi+yj+zk$

Podemos pensar en x , y y z como las cantidades que movemos en las direcciones i , j y k , desde la cola de v hasta su punta. Como los vectores i , j y k son conocidos y fijos, simplemente almacenamos los valores x , y y z y los usamos para representar nuestro vector numéricamente. De esta forma, un vector 3D v se representa por una tripleta ordenada (x, y, z) . Estos son conocidos como los componentes del vector. Nuestros vectores base i , j y k se representarán con componentes $(1, 0, 0)$, $(0, 1, 0)$ y $(0, 0, 1)$, respectivamente.



Producto punto de vectores

El producto punto es omnipresente en la programación de videojuegos, útil en todo, desde gráficos hasta simulación e Inteligencia Artificial.

El producto escalar de dos vectores, también conocido como producto escalar o producto interno, es una de las operaciones más utilizadas en gráficos 3D porque proporciona una medida de la diferencia entre las direcciones en las que apuntan los dos vectores.

El producto escalar de dos vectores n-dimensionales P y Q, escrito como $P \cdot Q$, es la cantidad escalar dada por la fórmula

$$P \cdot Q = \sum_{i=1}^n P_i Q_i.$$

Esta definición establece que el producto escalar de dos vectores viene dado por la suma de los productos de cada componente. En tres dimensiones, tenemos

$$P \cdot Q = P_x Q_x + P_y Q_y + P_z Q_z$$

Dos vectores P y Q son perpendiculares si y sólo si $P \cdot Q = 0$. Esto viene de hecho que la función coseno es cero en un ángulo de 90 grados y el producto punto está relacionado a esta función. Los Vectores cuyo producto punto es cero son llamados ortogonales.

El signo del producto punto nos dice que tan cercanos dos vectores apuntan en la misma dirección. Por ejemplo, basándonos en un vector P, cualquier vector que produzca un producto punto positivo está apuntando al mismo lado que P, mientras que cualquier vector que al multiplicarlo por P produzca un valor negativo estará en el lado contrario.

Matrices

En el álgebra lineal, una matriz es una cuadrícula rectangular de números dispuestos en filas y columnas. Recordando nuestra definición anterior de vector como una matriz unidimensional de números, una matriz también puede definirse como una matriz bidimensional de números. (Los

"dos" en "matriz bidimensional" provienen del hecho de que hay filas y columnas, y no deben confundirse con vectores o matrices 2D.) Entonces un vector es una matriz de escalares, y una matriz es una matriz de vectores.

En general, una matriz cuadrada puede describir cualquier transformación lineal. Una transformación lineal conserva líneas rectas y paralelas, y no hay traslación, es decir, el origen no se mueve. Sin embargo, otras propiedades de la geometría, tales como longitudes, ángulos, áreas y volúmenes, posiblemente se vean alteradas por la transformación. En un sentido no técnico, una transformación lineal puede "estirar" el espacio de coordenadas, pero no "curva" ni "deforma". Este es un conjunto muy útil de transformaciones, incluyendo: rotación, escala, proyección ortográfica, reflexión y "shear".

Apéndice B. Principios de animación

Estirar y encoger

El estirar y encoger tiene el propósito es dar una sensación de peso y flexibilidad al dibujar objetos. Puede ser aplicado a objetos simples, como una pelota rebotando o algo mucho más complejo como la musculatura de un rostro humano. Si lo llevamos al extremo, una figura estirada o encogida con cierto grado de exageración puede tener un efecto cómico. Sin embargo en las animaciones realistas lo más importante de este principio es el hecho de que el volumen de un objeto no cambia cuando se estira o se encoge. Por ejemplo una pelota se estira en su longitud de manera horizontal, su ancho(en caso de tres dimensiones, también la profundidad) debe contraerse en horizontal como corresponde.

Anticipación

La anticipación se usa para preparar a la audiencia para una acción y que esta parezca más realista. Un bailarín saltando fuera del piso tiene que doblar sus rodillas antes, o un golfista haciendo swing antes debe balancear el palo hacia atrás. Esta técnica también puede ser usada para cuestiones que involucran sucesos, como un personaje mirando hacia afuera de la pantalla para anticipar la llegada de alguien, o que preste atención a un objeto que está a punto de recoger.

Puesta en Escena

Este principio es similar a lo que en teatro y en cine se conoce como puesta en escena. Su propósito es dirigir la atención de la audiencia y dejar en claro que es lo más importante dentro de la escena; Johnston y Thomas lo definen como "El presentar cualquier idea de forma completamente clara, sin lugar a dudas." esa idea puede ser una acción, una expresión, una personalidad o un humor. Esto puede hacerse de distintas formas, como con el posicionamiento de un personaje dentro del cuadro, el uso de luz y sombra, o la angulación y posición de la cámara. La esencia de este principio es mantener la atención en lo que es relevante, evitando los detalles innecesarios.

Animación directa y Pose a pose

Estas son dos aproximaciones distintas al proceso de dibujar una animación. La "animación directa" significa dibujar directamente una escena cuadro por cuadro de principio a fin, mientras que la animación "pose a pose" implica empezar el proceso de animación dibujando poses clave y posteriormente llenando los espacios. La animación directa crea la ilusión de fluidez y dinamismo en el movimiento, por lo que es mejor para producir secuencias realistas. Por otra parte esta técnica dificulta el mantener las proporciones y la creación de poses precisas o convincentes. La animación "pose a pose" funciona mejor para escenas dramáticas o emotivas donde la composición y la relación con el ambiente son de gran importancia. Por lo mismo se usa con frecuencia una combinación de ambas.

Acción complementaria y Acción superpuesta

Se les denomina acción complementaria y acción superpuesta a estas técnicas que se relacionan estrechamente, ya que ambas ayudan a que el movimiento sea más realista y dé la ilusión de que el personaje se mueve de acuerdo a las leyes de la física, sobre todo el principio de la inercia. "Acción complementaria" significa que las piezas desvinculadas del cuerpo deben continuar su movimiento incluso después de que el personaje se dejó de mover, y deben continuar mostrando varios grados de movimiento oscilatorio con el objetivo de amortiguar la dirección de donde proviene el personaje hacia el centro de masa. "Acción superpuesta" es cuando las partes del cuerpo tienden a moverse en diferentes rangos (un brazo se va a mover en diferente tempo que la cabeza, etc.). El "arrastar" puede ser considerado como una tercera técnica, en la cual el personaje comienza a moverse y a algunas partes de su cuerpo les toma algunos cuadros alcanzarlo.

Acelerar y desacelerar

El movimiento del cuerpo humano y el de la mayoría de los objetos necesita tiempo para acelerar y desacelerar. Por esta razón, la animación luce más realista si tiene más dibujos cerca del inicio y el fin de una acción, enfatizando las poses clave, y sin acentuar las del medio. Este principio aplica para personajes moviéndose de una pose clave a otra, así como sentarse y pararse, pero también para objetos inanimados en movimiento, como una pelota rebotando.

Arcos

La mayoría de las acciones naturales tienden a seguir una trayectoria en forma de arco, por lo que la animación debe apegarse a este principio creando estos "arcos" para dar la sensación de realismo. Esta técnica puede aplicarse a un miembro en movimiento rotando la articulación, o al lanzar un objeto en una trayectoria parabólica. La excepción de este principio aplica para el movimiento mecánico, o movimientos fuertes, ya que normalmente estos suceden en líneas rectas.

Acción Secundaria

Agregar acciones secundarias a la acción principal le da más vida a la escena y ayuda a respaldarla. Una persona caminando puede simultáneamente balancear sus brazos, mantenerlos en sus bolsillos, hablar o silbar, o mostrar emociones a través de sus expresiones faciales. Lo importante sobre las acciones secundarias es que enfatizan, no desvían la atención de la acción principal. Si la acción secundaria opaca a la acción principal, es mejor prescindir de ella. Por ejemplo, en un movimiento dramático las expresiones faciales pasan desapercibidas. En estos casos es mejor incluirlos sólo al principio y al final del movimiento, no durante este.

Timing

Timing se refiere al número de dibujos o cuadros que se utilizan para una acción dada, lo que en cine se traduce en la velocidad de una acción. A un nivel puramente físico, el timing correcto hacer que los objetos parezcan obedecer las leyes de la física; por ejemplo, el peso de un objeto determina con qué impulso va a reaccionar, como al ser empujado. El timing es vital para establecer el humor, las emociones y reacciones de un personaje. También puede ser útil para comunicar aspectos de la personalidad del personaje.

Exageración

La exageración es un efecto usado especialmente en la animación, ya que el imitar la realidad de forma perfecta puede lucir estático o aburrido en las caricaturas. El nivel de exageración depende de cuanto realismo busques o de un estilo en particular, como el tipo de dibujo o animación de un artista en específico. Otras formas de exageración, pueden involucrar cuestiones sobrenaturales o surrealistas, como alteraciones en los atributos físicos de un personaje, o elementos en la narrativa.

Dibujo Sólido

El principio del dibujo sólido significa el tomar en cuenta que se está haciendo referencia a formas de un espacio tridimensional, por lo que se les debe dar volumen y peso. El animador necesita ser un habilidoso dibujante y tiene que entender las bases de las figuras tridimensionales, anatomía, peso, balance, luz, sombra, etc.

Atractivo

El atractivo de un personaje de caricatura corresponde a lo que se conoce como el carisma de un actor. Un personaje que es atractivo, no es necesariamente simpático, los villanos y los monstruos también pueden ser atractivos, lo importante es que el espectador tenga la sensación de que el personaje es real e interesante. Existen muchos trucos para hacer que el personaje se conecte con la audiencia; para personajes simpáticos, el uso de un rostro simétrico o parecido al de un bebé, tiende a ser efectivo. Un rostro complicado, duro, o difícil de leer, va a carecer de atractivo. Puede ser descrito con más precisión como "cautivador" en la composición de la pose, o en el diseño del personaje.