

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

SEDE VÍCTOR LEVI SASSO

FOLLETO DE HERRAMIENTAS DE COMPUTACIÓN GRÁFICA
INCLUYE PRUEBAS SUMATIVAS Y PRESENTACIONES DEL CONTENIDO

DR. CARLOS A. ROVETTO

MAYO 2021



Universidad Tecnológica de Panamá (UTP)
Esta obra está licenciada bajo la Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.

Para ver esta licencia:
<https://creativecommons.org/licenses/by-nc-sa/4.0>

Contenido

Índice de figuras	5
Índice de Tablas	9
Introducción.....	10
Capítulo I: Generalidades de la computación gráfica	11
Generalidades y evolución histórica	11
Definición y conceptos	11
Principales hitos en la evolución histórica de la computación gráfica	12
Funcionamiento de dispositivos para el manejo y despliegue gráfico	13
Monitores	13
Impresoras	16
Escáneres	17
Joysticks	18
Teclados	19
Mouse	20
Formatos de archivos gráficos	22
Capítulo II: Manejo de elementos gráficos en entornos digitales	24
Algoritmos de discretización de líneas.....	24
Algoritmo DDA	24
Algoritmo de Bresenham.....	28
Algoritmo de discretización de circunferencias	32
Algoritmo de discretización de elipses	34
Algoritmo de relleno de figuras	38
Métodos de transformación	41
Traslación	41
Rotación.....	42
Escalado	43
Simetría y recorte.....	44
Sistema de coordenadas homogéneas.....	49
Capítulo III: Fundamentos de animación 2D	51
Introducción al bosquejo de la figura humana	51
Dimensionamiento por tamaño de cabeza.....	51

Figura masculina y femenina	51
Rostro	51
Manos y pies	52
Caricatura	52
Evolución emocional	55
Reglas de tercios	56
Teoría del color	57
Principios de animación	58
Synfig	59
Capítulo IV: Herramientas de generación de imágenes 3D	60
Introducción a la herramienta	60
Secciones del área de trabajo	61
Layout	61
Vistas	67
Paleta de objetos y herramientas	67
Transformaciones	68
Traslación	68
Rotación	69
Escalados	69
Formas básicas	69
Operaciones binarias	73
Fuentes de luz	77
Point light	77
Spot light	78
Area light	78
Sun light	78
Tipos de cámaras	79
Formas basadas en splines	81
Materiales y texturas	84
Efectos especiales aplicados a los objetos	87
Efectos atmosféricos	95
Bibliografía	101

Anexos	102
ANEXO 1: PRUEBAS RÁPIDAS	119
ANEXO 2: PRESENTACIONES.....	129

Índice de figuras

Figura 1. Disciplinas de la computación gráfica.	11
Figura 2. Monitor de tubos de rayos catódicos.....	14
Figura 3. Pantallas con tecnología LCD.....	15
Figura 4. Ejemplo de pantalla plasma..	15
Figura 5. Ejemplo de pantalla LED.....	16
Figura 6. Representación de la estructura general de un escáner.....	18
Figura 7. Izquierda: modelo del diseño inicial del joystick; derecha: mando inalámbrico introducido por Nintendo..	18
Figura 8. Teclado QWERTY.....	19
Figura 9. Teclado QWERTZ.....	20
Figura 10. Teclado AZERTY.	20
Figura 11. Teclado DVORAK.	20
Figura 12. Modelo de mouse vendido con las Xerox.	21
Figura 13. Ejemplo de mouse gamer, incluye dos botones adicionales y el lateral izquierdo.....	21
Figura 14. Mapa de bits.....	22
Figura 15. Vectorial.	22
Figura 16. Dimensionamiento por tamaño de cabeza.....	51
Figura 17. Figura masculina (derecha) y femenina (izquierda).	51
Figura 18. Rostro.....	51
Figura 19. Manos y pies.	52
Figura 20. Se trazan ejes horizontales y verticales en los círculos para mantener la proporción de los rasgos del rostro.	52
Figura 21. En las caricaturas es común que la forma de la cabeza de los personajes varíe.	53
Figura 22. Los elementos del rostro brindan infinidad de personalidades a los personajes.	53
Figura 23. Dibujos de rostros de caricaturas.....	53
Figura 24. Guías de dirección a la que estará mirando el personaje.	54
Figura 25. Direcciones de visión del personaje.....	55
Figura 26. Base para las expresiones.....	55
Figura 27. Expresiones obtenidas con distintos estilos de cejas y boca.	55
Figura 28. Opción que activa la función de cuadrícula.....	56
Figura 29. Puntos fuertes.....	56
Figura 30. Ejemplo de aplicación de regla de tercios.....	57
Figura 31. Círculo cromático.....	58
Figura 32. Synfig.....	59
Figura 33. Pantalla de inicio de Blender.....	61
Figura 34. Ventana Layout.....	62

Figura 35. Ventana Modeling.	62
Figura 36. Ventana Sculpting.	63
Figura 37. Ventana UV Editing.	63
Figura 38. Ventana Texture Paint.	64
Figura 39. Ventana Shading.	64
Figura 40. Ventana Animation.	65
Figura 41. Ventana Rendering.	65
Figura 42. Ventana Compositing.	66
Figura 43. Ventana Scripting.	66
Figura 44. Opción de emular teclado numérico.	67
Figura 45. Lista de objetos en Blender.	68
Figura 46. Herramientas para la modificación de objetos.	68
Figura 47. Figuras básicas en Blender.	69
Figura 48. Añadir círculo desde el menú Add.	70
Figura 49. Selección de trabajo por aristas.	70
Figura 50. Aplicación del comando Fill sobre un círculo.	71
Figura 51. Inserción de una cara dentro del círculo.	71
Figura 52. Izquierda: Extruir mediante la barra de herramientas. Derecha: Extruir con el comando E sobre el eje Z (azul).	72
Figura 53. Vista ortogonal derecha.	72
Figura 54. Escalado de los bordes del objeto.	72
Figura 55. Subdivisión de caras y escalado.	73
Figura 56. Objeto modificado.	73
Figura 57. Ícono para acceder a los modificadores de los objetos.	74
Figura 58. Objetos seleccionados. El último objeto seleccionado está resaltado por un tono amarillo.	74
Figura 59. Panel de opciones del modificador booleano.	75
Figura 60. Selección del tipo de operación y objeto del que se heredarán las características.	75
Figura 61. Cambio del viewpoint.	75
Figura 62. Resultado de la operación diferencia.	76
Figura 63. Resultado de la operación Unión.	76
Figura 64. Resultado de operación Intersección.	76
Figura 65. Opción para añadir una nueva fuente de luz.	77
Figura 66. Point light.	77
Figura 67. Spot light.	78
Figura 68. Area light.	78
Figura 69. Sun light.	79
Figura 70. Menú de cámara.	79
Figura 71. Render con lente en perspectiva.	80

Figura 72. Render con lente ortogonal.	80
Figura 73. Render con lente panorámico.	81
Figura 74. Inserción de una curva de Bézier.	81
Figura 75. Curva de Bézier en modo edición.	82
Figura 76. Cambio de Fill Mode.	82
Figura 77. Ajustes de los valores de Depth y Resolution.	83
Figura 78. Resultado de cambio en la configuración de la geometría de la curva.	83
Figura 79. Paso para convertir la curva en Mesh.	84
Figura 80. Menú Material.	84
Figura 81. Menú de un material nuevo.	85
Figura 82. Cambio de color base.	85
Figura 83. Opciones de color: Brick Texture.	86
Figura 84. Menú Image Texture.	86
Figura 85. Ejemplo de textura insertada.	87
Figura 86. Corrección del despliegue de la textura insertada.	87
Figura 87. Modificadores de objetos.	87
Figura 88. Colocación de los objetos en escena.	88
Figura 89. Adición del modificador Cloth al plano.	88
Figura 90. Cambio de frame.	89
Figura 91. El plano con el modificador cloth atraviesa el cubo.	89
Figura 92. Modificador collision al cubo.	90
Figura 93. El plano se detiene sobre el cubo.	90
Figura 94. Subdivisión del plano para corregir a apariencia del plano sobre el cubo.	90
Figura 95. Resultado del modificador cloth sobre un plano con subdivisiones.	91
Figura 96. Plano y cubo modificado.	91
Figura 97. Añadir modificador Dynamic Paint al plano seguido por Add Canvas.	91
Figura 98. Cambio del tipo de superficie de la canva.	92
Figura 99. Dynamic paint tipo Brush.	92
Figura 100. Resultado al aplicar el modificador Dynamic paint.	92
Figura 101. Elementos necesarios para la animación.	93
Figura 102. Ubicación en el frame 0 antes de insertar el primer keyframe.	93
Figura 103. Selección de opción de keyframe LocRot.	93
Figura 104. Posición del objeto en el frame 0.	94
Figura 105. Panel en donde se pueden modificar de forma más precisa los ángulos de rotación y posición del objeto.	94
Figura 106. Recorrido del objeto.	94
Figura 107. Plano por encima de los ejes Y y X.	95
Figura 108. Botón para añadir el modificador de partículas.	95
Figura 109. Acción del modificador partículas sobre el plano al avanzar en la línea de tiempo.	96

Figura 110. Modificador de la intensidad de emisión de 1000 a 3000.	96
Figura 111. Personalización de la forma de las partículas.	97
Figura 112. Emisión de partículas con la forma de una gota.	97
Figura 113. Al modificar el objeto, los cambios se verán reflejados en las partículas... ..	98
Figura 114. Quick Smoke.....	98
Figura 115. Caja de dominio de las partículas de humo.	99
Figura 116. Añadir Force Field.....	99
Figura 117. Force Field tipo wind.	99
Figura 118. Modificación de la fuerza.....	100
Figura 119. Cambio de dirección de las partículas de humo.....	100

Índice de Tablas

Tabla 1. Evolución histórica de la computación gráfica.....	13
Tabla 2. Tipos de formatos más comunes en el entorno gráfico.....	23
Tabla 3. Vistas en Blender.	67

Introducción

El conjunto de conocimientos requeridos para la abstracción de procesos a través de sistemas formales es el fundamento de la teoría de la computacional. Esto permite modelar procesos de forma abstracta considerando las limitaciones de los dispositivos que procesan la información dentro de la computadora y que son la base para la aplicación de las ciencias de la computación en las diversas áreas del conocimiento. Algunos de los temas que se abordan en este folleto están soportados por conceptos de lógica y razonamiento, teoría de conjuntos, la teoría de autómatas, entre otras teorías y que permite estandarizar los procesos realizados por la computadora y así darles solución desde un enfoque teórico.

La teoría de computacional proporciona métodos de descripción de procesos que realiza la computadora que encuentran aplicaciones en áreas como compiladores, gráficos y lenguajes de programación. Sabemos que los lenguajes de programación tienen una semántica y un estudio formal riguroso por lo cual es importante conocer sobre los sistemas axiomáticos. Un sistema axiomático consiste en un conjunto de axiomas que son verdades evidentes que no necesitan ser demostradas y que son tomados por todos los individuos y funcionan como modelos conforme a los cuales razonamos. Un sistema axiomático puede tener expresados sus axiomas de manera formal o informal, siendo la axiomatización formal la que nos interesa en este folleto debido a que utiliza un lenguaje formal y en él cada axioma es una cadena finita de signos de un alfabeto del lenguaje formal, siguiendo reglas que hacen de la secuencia una fórmula bien formada.

El razonamiento axiomático nos ayuda a abordar temas relacionados con el álgebra booleana que se utiliza para expresar los efectos de las compuertas lógicas y se utilizan para producir circuitos lógicos. En este folleto se aborda el estudio de la combinación de las compuertas lógicas, que son de gran utilidad en el diseño de los circuitos lógicos mediante el álgebra booleana. Finalmente se aborda el uso de funciones lógicas mediante diagrama de Karnaugh y la simplificación de estas funciones utilizando el concepto de redundancia como minitérmino o maxitérmino.

Capítulo I: Generalidades de la computación gráfica

Generalidades y evolución histórica

Definición y conceptos

La computación gráfica es una ciencia enfocada en la comunicación visual del arte asistida por una computadora, en lo que también juegan un rol de importancia los dispositivos que permiten interactuar con dicho arte (Hughes, Van Dam, & Mcguire, 2014). El arte puede ser la representación de la realidad o de objetos abstractos.

Se justifica a la computación gráfica como ciencia debido a que en ella convergen distintas disciplinas. Estas disciplinas actúan en conjunto para generar un mensaje visual, que al final entra en contacto con un último e importante factor: la percepción humana. Estas disciplinas se observan en la **Figura 1**:



Figura 1. Disciplinas de la computación gráfica.

- **Física:** Se utiliza para hacer modelos realistas en los que se involucran los espectros de luz y movimiento dentro de las animaciones.
- **Matemáticas:** Se emplean para describir las formas y objetos que interactúan en la animación.
- **Ingeniería:** Procura optimizar los procesos de asignación del ancho de banda, memoria y procesador; es decir, contempla los aspectos a nivel computacional

que influyen sobre las tareas de renderizado con el principal objetivo de economizar recursos y tiempo sin afectar la calidad del producto final.

- **Arte, diseño gráfico y la interacción humano – computador:** Persiguen lograr una comunicación efectiva con el usuario.

En la computación gráfica se contemplan dos direcciones o sentidos en la comunicación:

- **Comunicación computador a usuario:** Se presenta en una pantalla la idea o mensaje de forma atractiva y entendible al usuario.
- **Comunicación usuario a computador:** Se genera un flujo de información que retorna a la máquina. Esta información es obtenida a través de la utilización de dispositivos que permiten al usuario la interacción con el contenido mostrado en pantalla.

Principales hitos en la evolución histórica de la computación gráfica

En la siguiente tabla se resumen algunos de los acontecimientos que marcaron etapas en la evolución de la computación gráfica según (Hartfoushi, Obiedat, Al-Sayyed, & Al-Shboul, 2013):

Décadas	Acontecimientos
1950	<ul style="list-style-type: none"> • Primer monitor basado en tecnología Cathode Ray Tube (CRT). • Ben Laposky diseñó el primer equipo que mostró imágenes gráficas. Consistía en un osciloscopio que trabajaba con vigas electrónicas y registraba todo en una película. • El primer sistema de computación gráfica que manejaba video, Whirlwind, fue desarrollado por Jay Forrester y Robert Everett, en el Massachusetts Institute of Technology (MIT). • Derivado de Whirlwind, surge SAGE (Semi-Automatic Ground Equipment). Programa que mediante gráficos vectoriales simples, mostraba imágenes de radar, convirtiéndolo en pieza clave del sistema estadounidense de defensa contra misiles. • Surge el primer sistema de software comercial de manufactura asistida por computadora (CAM). Diseñado por el Dr. Patrick J. Hanratty, padre de los gráficos asistidos por computadora. • Su software era una herramienta de programación de control numérico. • General Motors y IBM desarrollaron el Diseño Aumentado por Computadora -1, con el fin de ayudar a los ingenieros con el diseño de los automóviles.
1960	<ul style="list-style-type: none"> • Ivan Sutherland presenta como parte de su disertación de Ph. D. el proyecto "Sketchpad". Utilizó una computadora TX-2. Sketchpad es reconocido como el primer software CAD.

	<ul style="list-style-type: none"> • Syntha Vision presenta el primer software para el modelado de sólidos. • La NASA desarrolló un procesador de imágenes llamado VICAR (Video Image Communication and Retrieval) sobre el mainframe de IBM. Con este programa se procesaron las imágenes capturadas del espacio. • Se implementaron los primeros simuladores de vuelo. • Tektronix, primer computador CRT. • Se funda SIGGRAPH (Association for Computing Machinery's Special Interest Group on Computer Graphics and Interactive Techniques).
1970	<ul style="list-style-type: none"> • Pierre Bezier da a conocer las curvas de Bezier, que más adelante constituirían una de las principales herramientas en el dibujo vectorial. • Se introduce la tecnología ráster (matrices de celdas conocidas como píxeles). • Surgimiento de la técnica de mapeo de imágenes y textura por Blinn.
1980	<ul style="list-style-type: none"> • Quantel desarrolló Paintbox, herramienta de gran impacto entre los productores de programas de televisión, debido a que les permitía editar y manipular digitalmente lo grabado. • Primer computador Apple Macintosh con gráficos a color. • TRON – Película de Disney elogiada por sus gráficos tridimensionales.
1990	<ul style="list-style-type: none"> • Se introducen los monitores VGA y SVGA. • Marc Andreessen desarrolló el primer buscador web, Mosaic, con texto e imágenes. • Se libera la primera versión de Adobe Photoshop y PaintShop. • Se sientan las especificaciones OpenGL. • El World Wide Web Consortium (W3C) introduce el formato SVG (scalable vector graphics) para la utilización de imágenes de mayor tamaño en los sitios web. • Aparecen los primeros dispositivos móviles con pantallas táctiles.

Tabla 1. Evolución histórica de la computación gráfica.

Funcionamiento de dispositivos para el manejo y despliegue gráfico

Monitores

El monitor es el elemento principal dentro de un sistema de computación gráfica, dado que es el responsable de mostrar los gráficos resultantes al usuario. Todos los monitores, desde sus inicios, han empleado el modo de color rojo, verde y azul, RGB, por sus siglas en inglés.

A continuación, se explican de forma breve las características y funcionamiento de los monitores desde las primeras versiones hasta los que predominan actualmente en el mercado.

- **Tubos de rayos catódicos (CRT)**

La tecnología de Tubos de Rayos Catódicos, CRT por sus siglas en inglés, tuvo su origen en 1897 por Karl Ferdinand Braun; sin embargo, obtuvo su mayor auge al ser utilizado para la creación de los primeros televisores a finales de la década de 1940 (Figura 2).

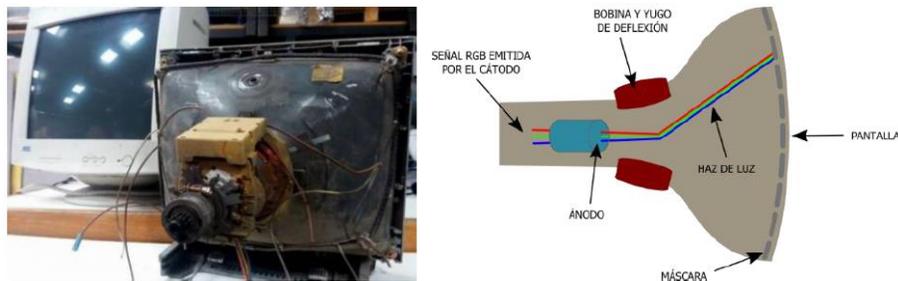


Figura 2. Monitor de tubos de rayos catódicos.

Una señal RGB era generada por un cañón de electrones, formado por un cátodo y un ánodo. El cátodo emitía los electrones que el ánodo se encargaba de manipular. Posteriormente el haz de luz era redirigido por el yugo de desviación, con el propósito de que llegara en un barrido horizontal a la película de fósforo que recubría la pantalla.

Es de esperar que este diseño original haya variado para hacer más cómoda su producción y elevar la calidad de las imágenes transmitidas. Los monitores RTC a colores incluyeron el uso de varias películas de fósforo entre las que se contempló una máscara de sombra (Hearn & Baker, 2006).

- **Pantalla de cristal líquido (LCD)**

El cristal líquido fue descubierto en un principio por Friederich Reinitzer en el año 1888, posterior a él investigadores como Richard Williams y George Heilmeyer realizaron nuevos descubrimientos relacionados al efecto electro-óptico que posee el material al ser sometido a un voltaje.

George Heilmeyer, investigador de los laboratorios RCA, fue el que llevó a cabo la construcción de la primera pantalla basada en la tecnología de cristal líquido, a mediados de la década de 1960.

Internamente, según (Martínez Riachi, Carreño, Constable, Tarabaín, & Freites, 2002), una pantalla LCD (Figura 3) consta de las siguientes capas:

- Fuente de luz blanca.
- Un polarizador, que permite el paso de un solo haz de luz.

- Capa de cristal líquido, formada por dos placas transparente recubiertas de óxido de estaño entre las que se encuentra la sustancia conocida como cristal líquido.
- Filtro de longitudes de onda.
- Un analizador.



Figura 3. Pantallas con tecnología LCD. Fuente: Pixabay.

- **Pantalla plasma**

Las pantallas de plasma (Figura 4) se conocen también con el nombre de pantallas de descarga de gas. Su principio funcional está basado en convertir la energía eléctrica en luz. Los primeros registros que se tienen de las pantallas de plasma datan del año 1964, en donde los científicos e investigadores Donald Bitzer, Gene Slottow y, en ese entonces estudiante recién graduado, Robert Wilson.



Figura 4. Ejemplo de pantalla plasma. Fuente: Pixabay.

Este tipo de pantallas constan de dos placas de cristal separadas por un espacio en el que se inyectan gases nobles. Se adhieren tiras de material conductor de forma horizontal a una placa y de forma vertical a la otra. Al suministrar tensión eléctrica en las intersecciones formadas por las tiras, el gas reacciona y genera luz.

- **Pantalla de diodo emisor de luz (LED)**

Este tipo de monitores utilizan una tecnología desarrollada en **1962** por **Nick Holonyak**. Holonyak desarrolló el primer **LED que emitía luz roja**. Luego en **1967**, **George Craford**, estudiante de Holonyak, consiguió la creación del **LED verde** y finalmente, en **1994**; **Isamu Akasaki, Hiroshi Amano y Shuji Nakamura**, lograron obtener el **LED azul**.



Figura 5. Ejemplo de pantalla LED. Fuente: Pixabay.

En las pantallas de tipo LED (Figura 5) se recurre a matrices de diodos que dan origen a los pixeles. De igual manera que los anteriores, se maneja en base a tensiones de voltaje para producir una imagen.

Otras tecnologías previas a las mencionadas son las siguientes:

- Monochrome Display Adapter (**MDA**).
- Color graphics Adapter (**CGA**).
- Enhanced Graphics Adapter (**EGA**).
- Video Graphics Array (**VGA**).
- Super Video Graphics Array (**SVGA**).

Impresoras

Los primeros registros de una impresora mecánica se remontan a los años 1800, en donde también se crea la primera máquina de escribir, siendo considerada la precursora de los teclados e impresoras de la actualidad.

Teniendo estas bases con el pasar del tiempo se diseñaron e implementaron distintos modelos y técnicas de impresión, como las que se listan a continuación:

- Impresora de impacto.
- Impresora de matriz de puntos.
- Impresora de chorro de tinta.
- Impresora láser.

Un elemento para recordar en este punto es que, contrario al modo de color que utilizan los monitores, las impresoras usan el modo **CYMK** (cyan, yellow, magenta y key). Esto

es un factor importante al momento de hacer un diseño, pues, si el diseño será para exposición en pantallas, debe ser creado con el formato RGB, mientras que, si el objetivo es ser impreso, debe ser diseñado en CYMK. Esto se hace para garantizar la calidad y fidelidad final de los colores del diseño.

- **Impresora 3D**

La iniciativa de impresión en 3D surge ante el deseo de llevar los **diseños en planos a un entorno tridimensional** permitiendo una evaluación de los diseños y hacer pruebas antes de invertir en la producción de estos. Los modelos para impresión son creados utilizando un software tipo CAD.

Los aportes a la tecnología de impresión en 3D iniciaron en 1981 con Hideo Kodama, cuyo invento utilizaba luces ultravioletas para endurecer polímeros para crear objetos sólidos. Este principio fue tomado por Charles Hull, en 1984. Se atribuye a Hull la creación de la estereolitografía (**SLA**). Otros dos procesos para la impresión 3D son el sinterizado selectivo por láser (**SLS**) y el modelado por disposición fundida (**FDM**).

- ✓ **SLA**

El proceso conocido como estereolitografía o SLA por sus siglas en inglés, fue el primero en estar detrás de la posibilidad de imprimir un objeto a partir de datos en formato digital. Los objetos se imprimen por capas, eran bañados por un solvente y endurecidos con luces ultravioletas.

- ✓ **SLS**

El sinterizado selectivo por láser o SLS, como su nombre indica, está basado en el uso de polímeros en polvo, que son fundidos por un láser y colocados por capas para crear formas más complejas que con el SLA.

- ✓ **FDM**

El modelado por disposición fundida o FDM es la técnica de impresión que predomina en estos días. Este proceso calienta un cable de material termoplástico hasta convertirlo en líquido, lo que permite extraerlo y colocarlo en capas.

Escáneres

Un escáner es comúnmente utilizado para digitalizar documentos impresos, obteniendo así imágenes o texto, que se puede mostrar y almacenar en una computadora. El término digitalizar hace referencia a la transcripción de información analógica a lenguaje binario, es decir, ceros y unos (Fontanills & Franquesa, 2011).

El funcionamiento general de un escáner inicia con la colocación del documento que se quiere digitalizar en la superficie destinada para esta tarea (Figura 6). Cada modelo de

escáner cuenta con sensores de luz, que son capaces de identificar los tonos monocromos como los colores. Un escáner posee una fuente de luz que recorre el documento y basado en la intensidad del reflejo de esa luz, se obtiene la información referente a la coloración de cada píxel. Toda esta información es recolectada por la herramienta de software del escáner, quién la procesa y posteriormente ofrece la opción de almacenarla en el equipo.

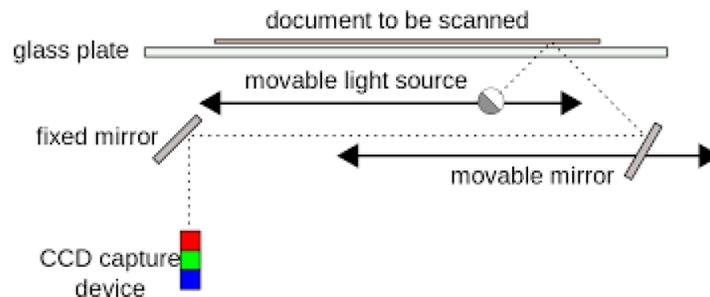


Figura 6. Representación de la estructura general de un escáner.

Joysticks

Para este dispositivo de entrada, conocido también como palanca de mando, no se tienen datos certeros sobre su origen; no obstante, tuvo su primera aplicación en la aviación, en donde estas palancas eran utilizadas para controlar los alerones de las naves hasta el presente.

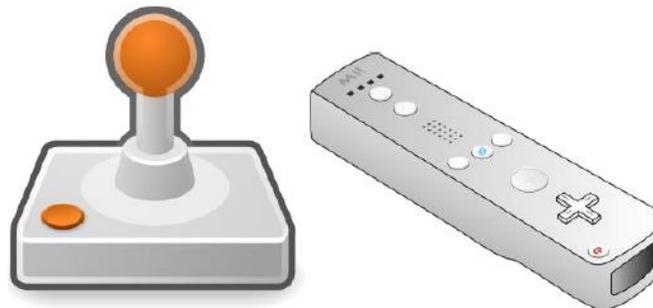


Figura 7. Izquierda: modelo del diseño inicial del joystick; derecha: mando inalámbrico introducido por Nintendo.
Fuente: Pixabay.

La siguiente aplicación que resaltar fue dentro del **área de entretenimiento digital**, específicamente en la rama de los **videojuegos**. Sus primeras versiones utilizaban interruptores para las direcciones horizontal y vertical por separado. Posteriormente se reemplazaron estos interruptores por un solo control que activara las cuatro direcciones gracias a la implementación de un pequeño circuito (Figura 7, izquierda).

Los joysticks han sido clasificados en dos grupos: **analógicos** y **digitales**. En el primer grupo entran los que siguen fiel a la incorporación de una palanca como tal en el control. En el segundo grupo están los que, en lugar de una palanca, poseen cuatro botones

posicionados formando una cruz, que representan las direcciones arriba, abajo, izquierda y derecha, respectivamente.

Como era de esperarse los joysticks no se limitaron sólo a botones, hoy en día incluyen otras tecnologías como **sensores de movimiento y vibración**; no obstante, su función sigue siendo la misma, ser un **señalador de posición en una pantalla**.

Teclados

Los teclados indistintamente de su modelo y ergonomía tienen como función principal servir como dispositivo de entrada, específicamente para información de tipo texto y comandos que permiten acceder a funciones tanto básicas como avanzadas dentro de las herramientas de software (Hearn & Baker, 2006).

Los teclados de forma general se encuentran subdivididos de la siguiente forma:

- Alfanumérico
- Numérico
- Teclas de cursor
- Funciones especiales

En cuanto a disposición de teclas se tienen las siguientes variantes:

- **QWERTY**: Es el de uso estándar. Heredado de las máquinas de escribir. Está patentado por Christopher Latham Sholes, quién luego lo vendió a la compañía E. Remington and Sons. **Ver Figura 8.**

°	!	"	#	\$	%	&	/	()	=	?	'	~	←
	-	1	2	3	4	5	6	7	8	9	0	,	.	Backspace
Tab	↔	Q	W	E	R	T	Y	U	I	O	P	,	*	↵
		@										.	+ ~	↵
Caps Lock	⬆	A	S	D	F	G	H	J	K	L	Ñ	[]	
												{	}	
Shift	⬆	>	Z	X	C	V	B	N	M	;	:	-		Shift
		<								,	.	=		⬆
Ctrl		Win Key	Alt								Alt Gr	Win Key	Menu	Ctrl

Figura 8. Teclado QWERTY.

- **QWERTZ**: Utilizado en Alemania. La variante representativa es el cambio de posición entre la letra Y y Z. **Ver Figura 9.**

°	!	"	§	§	%	&	/	()	=	?	'	←
^	1	2	3	4	5	6	7	{	[]	0	ß	↩
↔	Q	W	E	R	T	Z	U	I	O	P	Ü	*	↵
↓	A	S	D	F	G	H	J	K	L	Ö	Ä	'	↵
↑	>	Y	X	C	V	B	N	M	;	:	-	↵	
	<							µ	,	.	-	↵	
Strg	(Win)	Alt						Alt Gr	(Win)	(Menu)	Strg		

Figura 9. Teclado QWERTZ.

- **AZERTY:** Utilizado en Francia y Bélgica. **Ver Figura 10.**

2	1	2	3	4	5	6	7	8	9	0	°	+	←
	&	é	#	'	{	-	è	-	ç	à	@)	Backspace
Tab	A	Z	E	R	T	Y	U	I	O	P	^	£	Enter
Caps Lock	Q	S	D	F	G	H	J	K	L	M	%	µ	↵
Shift	>	W	X	C	V	B	N	?	:	/	\$	Shift	
	<							,	:	!	!	↵	
Ctrl	Win Key	Alt						Alt Gr	Win Key	Menu	Ctrl		

Figura 10. Teclado AZERTY.

- **DVORAK:** Patentado por August Dvorak y William Dealey en el año 1936, versión que posteriormente sufrió cambios hasta ser estandarizada en 1982 por el ANSI. **Ver Figura 11.**

°	!	"	·	§	%	&	/	()	=	?	¿	←
o	1	2	3	4	5	6	7	8	9	0	:	i	↩
↔	:	;	Ñ	P	Y	F	G	C	H	L	^	*	↵
	-	,									+		↵
↑	A	O	E	U	I	D	R	T	N	S	..	Ç	↵
		€									ç	Ç	
↵	>	-	Q	J	K	X	B	M	W	V	Z		↵
	<												
Ctrl	Win	Alt						Alt Gr	Win	Menu	Ctrl		

Figura 11. Teclado DVORAK.

Mouse

Segundo dispositivo de entrada utilizado comúnmente en los sistemas de computación gráfica, el primero es el teclado.

El invento principal que dio paso a la futura generación del mouse como se conoce en la actualidad, es el del Trackball por parte de la fuerza Marina Canadiense, en el año 1952. Poco después en 1963 nace el primer prototipo a manos de Douglas Engelbart.

El modelo diseñado por Engelbart fue modificado, siendo la característica más notoria la aparición de una esfera en el interior del mouse (Figura 12), encargada de dictar las

coordenadas en la que se encontrara el cursor. Este aporte fue realizado por Telefunken, una empresa alemana.



Figura 12. Modelo de mouse vendido con las Xerox.

En el ámbito comercial, los primeros equipos personales en incluir una interfaz gráfica que permitía la interacción con un mouse fueron las Xerox cerca del año de 1973. Estas computadoras integraron el mouse entre sus componentes.

El siguiente en aparecer en escena fue Apple en 1983. Su modelo de mouse reemplazó la esfera de goma por una de metal, y en cuanto a apariencia, difería de los modelos anteriores en la cantidad de botones, debido a que solo contaba con un único botón mientras que otros contaban con tres botones en la parte superior.

La mayor desventaja del modelo de mouse con la esfera en su interior era la acumulación de suciedad. Este problema fue resuelto en 1980 por Microsoft, quien reemplazó la esfera por un LED infrarrojo y añadió la rueda entre los botones. Posteriormente, este LED infrarrojo fue reemplazado por un láser, siendo la tendencia que ha llegado hasta el presente.

La apariencia del mouse ha persistido con el pasar de los años, dominando la presencia de los dos botones y la rueda; sin embargo, a este modelo se le han aplicado modificaciones para crear diseños enfocados en tareas más específicas, un ejemplo claro son mouse para gamers, que incluyen botones adicionales para beneficiar al jugador (**Figura 13**).



Figura 13. Ejemplo de mouse gamer, incluye dos botones adicionales y el lateral izquierdo. Fuente: Pixabay.

Formatos de archivos gráficos

Antes de entrar en detalles sobre los formatos de archivos gráficos, es importante recordar los dos tipos de imágenes con los que se puede trabajar: **mapa de bits** y **vectorial**.

- **Mapa de bits:** También denominados imágenes ráster, tiene como principal característica estar compuesta por píxeles. En este tipo de imágenes juega un rol de gran importancia la llamada resolución, debido a que, a mayor cantidad de píxeles, mejor será la calidad de la imagen. Su mayor problema se presenta en el momento de agrandar la imagen, dado que es común que se presente un pixelado y se pierda la calidad. Este tipo de gráfico se utiliza mayormente para fotografías, imágenes escaneadas o imágenes con alto nivel de detalles. **Ver Figura 14.**
- **Vectorial:** Este tipo de imágenes se componen de curvas obtenidas mediante cálculos matemáticos generados por la herramienta de diseño. A diferencia de los mapas de bits, en las imágenes vectoriales no es necesario preocuparse por la resolución. Se utilizan con mayor frecuencia para el diseño de logotipos, iconos o gráficos destinados a entornos web. **Ver Figura 15.**

Las siguientes figuras muestran ejemplos de los dos tipos de imágenes que se han descrito:

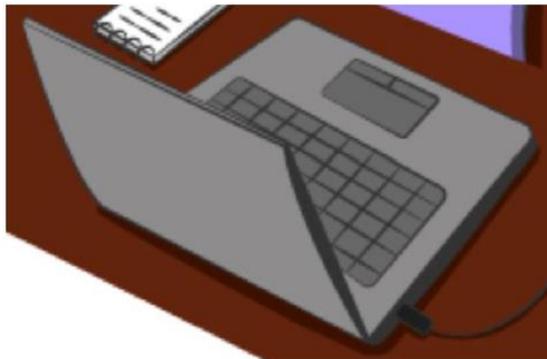


Figura 14. Mapa de bits.

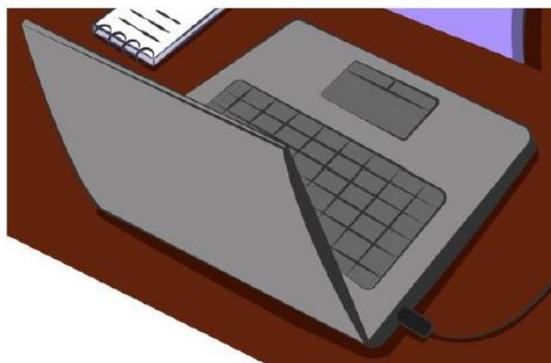


Figura 15. Vectorial.

En la siguiente tabla se describen los tipos de formatos más comunes en el entorno gráfico:

NOMBRE	EXTENSIÓN	CARACTERÍSTICAS
<i>Joint Photographic Experts Group</i>	JPEG	<ul style="list-style-type: none"> • Permite alta compresión, a cambio de calidad de imagen. • Utilizada mayormente para impresión. • Pierde calidad cada vez que se hagan y guarden cambios sobre una misma imagen. • Enfocada únicamente en imágenes. • Propicia para imágenes con colores complejos y sombreados.
<i>Windows Bitmap</i>	BMP	<ul style="list-style-type: none"> • Propio del programa Paint de Windows. • Buena calidad de imagen. • Ocupa mucho espacio de almacenamiento. • No admite fondos transparentes.
<i>Graphics interchange formato</i>	GIF	<ul style="list-style-type: none"> • Soporta 256 colores. • Admite animaciones que no requieren software externo. • Formato de compresión LZW que impide pérdida de calidad.
<i>Portable Network Graphics</i>	PNG	<ul style="list-style-type: none"> • Puede utilizar canales RGB. • Admite fondos transparentes y con degradados. • Imágenes simples pero alta calidad. • Ocupan mucho espacio de almacenamiento.
<i>Tagged Image File Format</i>	TIFF	<ul style="list-style-type: none"> • Usado para impresiones de calidad. • No realiza compresión al guardarse. • Archivos muy grandes.
<i>Portable Document Format</i>	PDF	<ul style="list-style-type: none"> • Utilizado con fines de exposición o impresión.
<i>Encapsulate PostScript</i>	EPS	<ul style="list-style-type: none"> • Formato de impresión. Las impresoras que utilizan este formato deben ser PostScript. • Maneja gráficos vectoriales y mapas de bits.
<i>Scalable Vector Graphics</i>	SVG	<ul style="list-style-type: none"> • Visible en navegadores. • Basado en XML.

Tabla 2. Tipos de formatos más comunes en el entorno gráfico.

Capítulo II: Manejo de elementos gráficos en entornos digitales

Algoritmos de discretización de líneas

Los algoritmos de discretización de líneas, como su nombre lo indica, están enfocados en los cálculos requeridos para el trazado de líneas en pantalla, para lo cual es necesario contar con dos puntos que respondan a las coordenadas inicial y final de los extremos de la línea. Estos dos puntos son los primeros en mostrarse en pantalla y a partir de ella se obtienen posiciones aproximadas de los píxeles por los que pasará la línea. Estas posiciones siempre serán llevadas a valores enteros (Donald Hearn and M. Pauline Baker, 2006). Los cálculos de las posiciones se basan en la ecuación geométrica que determina a la línea:

$$y = mx + b \quad (1)$$

En la ecuación 1, **m** representa la pendiente y **b** la intercepción con el eje y.

Algoritmo DDA

Algoritmo Analizador Diferencial Digital, DDA, por sus siglas en inglés, se basa en la ecuación 2 para el cálculo de las posiciones.

$$\Delta y = m\Delta x \quad (2)$$

Este algoritmo inicia obteniendo los valores de Δx y Δy , que a su vez se utilizan para calcular la pendiente m . Esto implica que es necesario conocer las coordenadas de los extremos entre los que se va a trazar la línea; el punto inicial (x_i , y_i) y el punto final (x_f , y_f).

Despejando la pendiente de la ecuación 2, se tiene que:

$$m = \frac{\Delta y}{\Delta x} = \frac{y_f - y_i}{x_f - x_i} \quad (3)$$

El valor de la pendiente determina las ecuaciones que se deben utilizar para realizar el determinar las siguientes posiciones.

- **Pendiente $m \leq 1$**

$$y_{k+1} = y_k + m \quad (4)$$

- **Pendiente $m \geq 1$**

$$x_{k+1} = x_k + \frac{1}{m} \quad (5)$$

El subíndice **k** indica la **cantidad de pasos** o **iteraciones** que se deben realizar para obtener todas las posiciones de los píxeles; k se determina evaluando el mayor entre los **valores absolutos** de Δx y Δy ; es decir:

Si $|\Delta x| > |\Delta y|$ entonces $k = \Delta x$. Si no, entonces $k = \Delta y$.

Ejemplo 1: Trazar la recta que va del punto (3, 4) hasta (13, 11) mediante el algoritmo DDA.

- **Paso 1:** Calcular Δx y Δy .

$$\begin{aligned}\Delta x &= x_f - x_i \\ \Delta x &= 13 - 3 \\ \Delta x &= \mathbf{10}\end{aligned}$$

$$\begin{aligned}\Delta y &= y_f - y_i \\ \Delta y &= 11 - 4 \\ \Delta y &= \mathbf{7}\end{aligned}$$

- **Paso 2:** Calcular la pendiente **m**. Se verifica las ecuaciones a utilizar dado el **valor de m**.

$$\begin{aligned}m &= \frac{\Delta y}{\Delta x} = \frac{7}{10} \\ m &= \mathbf{0.7}\end{aligned}$$

$$\begin{aligned}|m| &\leq 1 \\ |0.7| &\leq 1\end{aligned}$$

Se cumple la condición de la ecuación 4.

- **Paso 3:** Determinar la cantidad k de pasos.

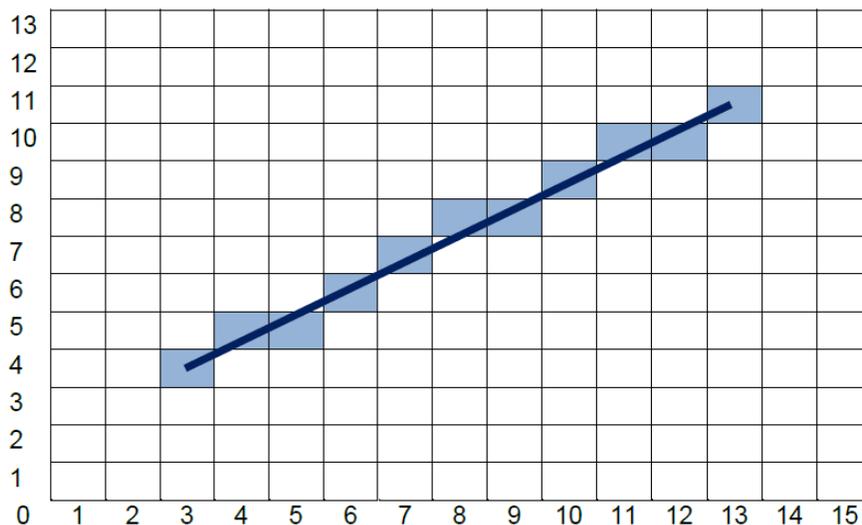
$$\begin{aligned}|\Delta x| &> |\Delta y| \\ |10| &> |7| \\ k &= 10\end{aligned}$$

- **Paso 4:** Se procede al cálculo del resto de los puntos. El **valor inicial de k es 0** y los valores de x_0 y y_0 corresponden al **punto inicial** suministrado al inicio del problema.

Nota: Para este ejemplo los valores de **x** tendrán un incremento de 1, mientras que los nuevos valores de **y** serán determinados por la **ecuación 4**. En caso de que el valor de **y** contenga cifras decimales, es necesario aplicar redondeo y llevarlo al entero más cercano. Ejemplo: 3.2 -> **3**; 3.5 -> **4**; 3.8 -> **4**.

k	x_k	y_k	Redondeo	
0	3	4	4	
1	$3 + 1 =$	4	$4 + 0.7 =$ 4.7	5
2	$4 + 1 =$	5	$4.7 + 0.7 =$ 5.4	5
3	$5 + 1 =$	6	$5.4 + 0.7 =$ 6.1	6
4	$6 + 1 =$	7	$6.1 + 0.7 =$ 6.8	7
5	$7 + 1 =$	8	$6.8 + 0.7 =$ 7.5	8
6	$8 + 1 =$	9	$7.5 + 0.7 =$ 8.2	8
7	$9 + 1 =$	10	$8.2 + 0.7 =$ 8.9	9
8	$10 + 1 =$	11	$8.9 + 0.7 =$ 9.6	10
9	$11 + 1 =$	12	$9.6 + 0.7 =$ 10.3	10
10	$12 + 1 =$	13	$10.3 + 0.7 =$ 11	11

- **Paso 5: Graficar los píxeles resultantes.** Las posiciones no se ubican en las intersecciones de los ejes.



Ejemplo 2: Trazar la recta que va del punto (2, 3) hasta (7, 12) mediante el algoritmo DDA.

- **Paso 1:** Calcular Δx y Δy .

$$\Delta x = x_f - x_i$$

$$\Delta x = 7 - 2$$

$$\Delta x = 5$$

$$\Delta y = y_f - y_i$$

$$\Delta y = 12 - 3$$

$$\Delta y = 9$$

- **Paso 2:** Calcular la pendiente m . Se **verifica las ecuaciones** a utilizar dado el **valor de m** .

$$m = \frac{\Delta y}{\Delta x} = \frac{9}{5}$$

$$m = 1.8$$

$$|m| \leq 1$$

$$|1.8| \leq 1$$

Se cumple la condición de la ecuación 5.

- **Paso 3:** Determinar la cantidad **k** de pasos.

$$|\Delta x| < |\Delta y|$$

$$|5| < |9|$$

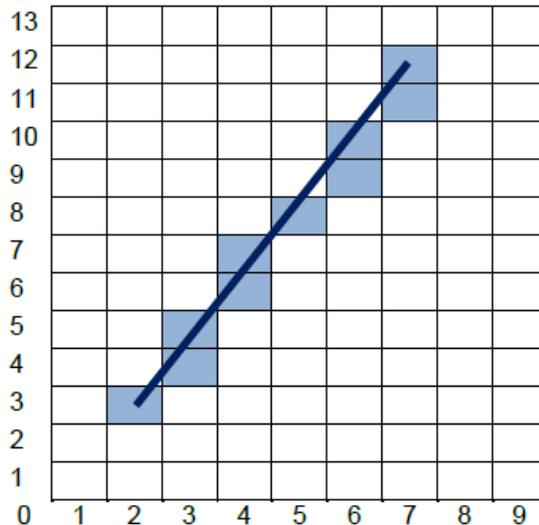
$$k = 9$$

- **Paso 4:** Se procede al cálculo del resto de los puntos. El **valor inicial de k es 0** y los valores de **x₀** y **y₀** corresponden al **punto inicial** suministrado al inicio del problema.

Nota: Para este ejemplo los valores de **y** tendrán un incremento de 1, mientras que los nuevos valores de **x** serán determinados por la **ecuación 5**. En caso de que el valor de **x** contenga cifras decimales, debe llevarse al entero más cercano.

k	x _k	Redondeo	y _k
0	2	2	3
1	$2 + \frac{1}{1.8} = 2.6$	3	3 + 1 = 4
2	$2.6 + \frac{1}{1.8} = 3.2$	3	4 + 1 = 5
3	$3.2 + \frac{1}{1.8} = 3.8$	4	5 + 1 = 6
4	$3.8 + \frac{1}{1.8} = 4.4$	4	6 + 1 = 7
5	$4.4 + \frac{1}{1.8} = 5$	5	7 + 1 = 8
6	$5 + \frac{1}{1.8} = 5.6$	6	8 + 1 = 9
7	$5.6 + \frac{1}{1.8} = 6.2$	6	9 + 1 = 10
8	$6.2 + \frac{1}{1.8} = 6.8$	7	10 + 1 = 11
9	$6.8 + \frac{1}{1.8} = 7.4$	7	11 + 1 = 12

- **Paso 5: Graficar los píxeles resultantes.** Las posiciones **no** se ubican en las intersecciones de los ejes.



Algoritmo de Bresenham

Algoritmo creado por **Jack E. Bresenham** en **1965**. Es un proceso de mayor precisión para el trazado de líneas, que además puede ser adaptado para la generación de curvas. De forma similar al algoritmo DDA requiere de un punto inicial y un punto final, a partir de los que se obtienen la pendiente **m** y las constantes: **Δx , Δy , $2\Delta x$, $2\Delta y$** (Donald Hearn and M. Pauline Baker, 2006).

Este algoritmo utiliza una variable llamada parámetro de decisión que generalmente se representa con una letra **p** minúscula con un subíndice que inicia en 0. El valor inicial de **p** se calcula en función de la pendiente, es decir:

- Si $0 < |m| < 1$:

$$p_0 = 2\Delta y - \Delta x \quad (6)$$

- Si $|m| > 1$:

$$\Delta x = -\Delta y \quad (7)$$

$$p_0 = 2\Delta x - \Delta y \quad (8)$$

En las ecuaciones 6 y 8 se aprecia que, de acuerdo al valor resultante de la pendiente **m**, se intercambian **Δx** y **Δy** . Este cambio aplica de forma similar en el cálculo del incremento de **p** para cada paso o iteración. El incremento de **p** se obtiene con las siguientes condiciones:

- Si $p_k < 0$ entonces
 x_{k+1}, y_k //Se incrementa *x* en una unidad y se mantiene el valor de *y*.

$$p_{k+1} = p_k + 2\Delta y$$

- Si $p_k > 0$ entonces

$x_k + 1, y_k + 1$ //Se incrementan los valores de x e y en una unidad.

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

Nota: Se observa que, si el valor absoluto de la pendiente es mayor a 1, se debe cambiar el signo de Δy .

De esta manera se tienen dos casos dentro del algoritmo de Bresenham:

- **Caso 1** ($0 < |m| < 1$): Se incrementa x de manera unitaria, mientras que se evalúan los valores de y .
- **Caso 2** ($|m| > 1$): Se intercambian x e y . Se incrementa y de manera unitaria (si la pendiente es negativa, se decreta unitariamente), mientras que se evalúan los valores de x .

La cantidad de pasos se determina de la siguiente forma:

$$\text{Si } |\Delta x| > |\Delta y| \text{ entonces } k = \Delta x$$

$$\text{Si no, entonces } k = \Delta y$$

Ejemplo 1: Mediante el algoritmo de Bresenham, trazar la línea que va del punto (1, 2) a (10, 8).

- **Paso 1:** Calcular las constantes Δx , Δy , $2\Delta x$ y $2\Delta y$.

$$\Delta x = x_f - x_i$$

$$\Delta x = 10 - 1$$

$$\Delta x = 9$$

$$2\Delta x = 2(9)$$

$$2\Delta x = 18$$

$$\Delta y = y_f - y_i$$

$$\Delta y = 8 - 2$$

$$\Delta y = 6$$

$$2\Delta y = 2(9)$$

$$2\Delta y = 12$$

- **Paso 2:** Calcular y evaluar la pendiente m .

$$m = \frac{\Delta y}{\Delta x} = \frac{6}{9}$$

$$m = 0.7$$

$$0 < |m| < 1$$

$$0 < |0.7| < 1$$

Se cumple la condición del **caso 1**.

- **Paso 3:** Determinar el valor de p_0 y k . Se utiliza la ecuación 6 dado que $0 < |0.7| < 1$.

$$p_0 = 2\Delta y - \Delta x$$

$$p_0 = 12 - 9$$

$$p_0 = 3$$

$$|\Delta x| > |\Delta y|$$

$$|9| > |6|$$

$$k = 9$$

- **Paso 4:** Calcular los puntos.

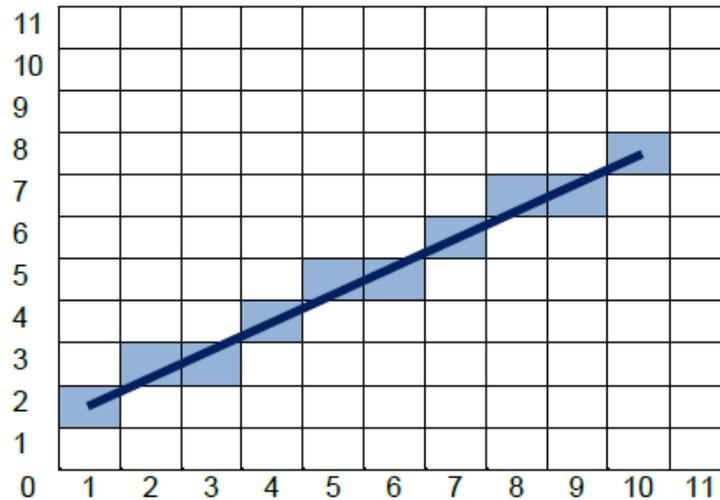
$$2\Delta y - 2\Delta x = 12 - 18$$

$$2\Delta y - 2\Delta x = -6$$

$$2\Delta y = 12$$

k	x_k	y_k	p_k	$p_k < 0$ (S/N)
0	1	2	3	N
1	1+1=2	2+1=3	3-6=-3	S
2	2+1=3	3+0=3	-3+12=9	N
3	3+1=4	3+1=4	9-6=3	N
4	4+1=5	4+1=5	3-6=-3	S
5	5+1=6	5+0=5	-3+12=9	N
6	6+1=7	5+1=6	9-6=3	N
7	7+1=8	6+1=7	3-6=-3	S
8	8+1=9	7+0=7	-3+12=9	N
9	9+1=10	7+1=8		

• **Paso 5: Graficar.**



Ejemplo 2: Mediante el algoritmo de Bresenham, trazar la línea que va del punto (1, 10) a (7, 3).

• **Paso 1:** Calcular las constantes Δx , Δy , $2\Delta x$ y $2\Delta y$.

$$\begin{array}{llll} \Delta x = x_f - x_i & 2\Delta x = 2(6) & \Delta x = y_f - y_i & 2\Delta y = 2(-7) \\ \Delta x = 7 - 1 & 2\Delta x = 12 & \Delta x = 3 - 10 & 2\Delta y = -14 \\ \Delta x = 6 & & \Delta y = -7 & \end{array}$$

• **Paso 2:** Calcular y evaluar la pendiente m .

$$m = \frac{\Delta y}{\Delta x} = \frac{-7}{6}$$

$$m = 0.7$$

$$|m| > 1$$

$$|-1.2| > 1$$

Se cumple la condición del **caso 2**.

$$\Delta y = -\Delta y$$

$$* \Delta y = -(-7) = 7$$

$$* 2\Delta y = 2(7) = 14$$

- **Paso 3:** Determinar el valor de p_0 y k . Se utiliza la ecuación 8 dado que $|-1.2| < 1$.

$$p_0 = 2\Delta y - \Delta y$$

$$p_0 = 12 - 7$$

$$p_0 = 5$$

$$|\Delta x| > |\Delta y|$$

$$|6| > |7|$$

$$k = 7$$

- **Paso 4:** Calcular los puntos.

$$2\Delta y - 2\Delta x = 12 - 14$$

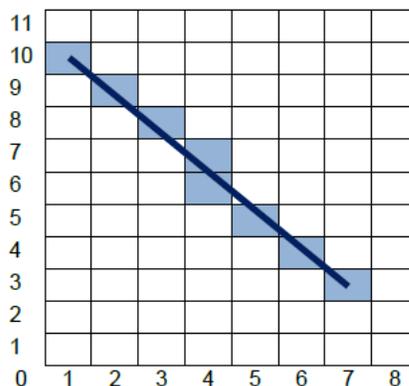
$$2\Delta y - 2\Delta x = -2$$

$$2\Delta x = 12$$

Nota: La pendiente de esta recta es negativa, por lo que los valores de y disminuirán de forma unitaria.

k	x_k	y_k	p_k	$p_k < 0$ (S/N)
0	1	10	5	S
1	1 + 1 = 2	10 - 1 = 9	5 - 2 = 3	S
2	2 + 1 = 3	9 - 1 = 8	3 - 2 = 1	S
3	3 + 1 = 4	8 - 1 = 7	1 - 2 = -1	N
4	4 + 0 = 4	7 - 1 = 6	-1 + 12 = 11	S
5	4 + 1 = 5	6 - 1 = 5	11 - 2 = 9	S
6	5 + 1 = 6	5 - 1 = 4	9 - 2 = 7	S
7	6 + 1 = 7	4 - 1 = 3		

- **Paso 5:** Graficar.



Algoritmo de discretización de circunferencias

Para la discretización de circunferencias existen al menos tres variantes de algoritmos: el básico, el de **Bresenham** y el de **punto medio**. Esta sección se enfoca en la aplicación de algoritmo de Bresenham para el dibujo de circunferencias.

El algoritmo requiere que sean suministrados el radio r y el punto en el que estará centrada la circunferencia (x_c, y_c) .

El punto inicial (x_0, y_0) está dado por el radio r , de la siguiente manera:

$$(x_0, y_0) = (0, r)$$

Se observa que el valor de x_0 es **cero**, mientras que el de y_0 es el **radio** de la circunferencia. Lo siguiente es el cálculo del **parámetro de decisión**, mediante la ecuación 9:

$$p_{k=0} = 3 - 2r \quad (9)$$

Al evaluar el resultado de p_k se pueden dar dos situaciones:

- Si $p_k < 0$ entonces
 $x_k + 1, y_k$ //Se **incrementa x** en una unidad y se **mantiene** el valor de **y**.
 $p_{k+1} = p_k + 4x_k + 6$
- Si $p_k > 0$ entonces
 $x_k - 1, y_k - 1$ //Se **incrementa x** una unidad, mientras que **y** se decrementa.
 $p_{k+1} = p_k + 4(x_k - y_k) + 10$

Las iteraciones se detienen cuando $x \geq y$. Para circunferencias con su centro en el origen, los valores de x_k y y_k se grafican de forma directa; no obstante, para aquellas circunferencias que su centro sea **distinto a (0, 0)** se debe realizar un paso adicional para cada x_k y y_k calculado:

$$x = x_k + x_c \quad (10)$$

$$y = y_k + y_c \quad (11)$$

Las ecuaciones **10** y **11** indican que para cada x calculada se le **suma** el valor de la **coordenada x de su centro**; lo mismo aplica en el caso de **y**.

La circunferencia se subdivide en octantes (ocho partes). Los puntos obtenidos con el algoritmo descrito corresponden al primer octante del cuadrante 1. Luego de haber calculado los puntos, se requiere determinar la simetría para el resto de los octantes y así completar la circunferencia.

Ejemplo 1: Dada una circunferencia con radio = 6 y su centro en el origen (0, 0). Obtenga las posiciones para el primer octante en el cuadrante 1 y gráfíquelos.

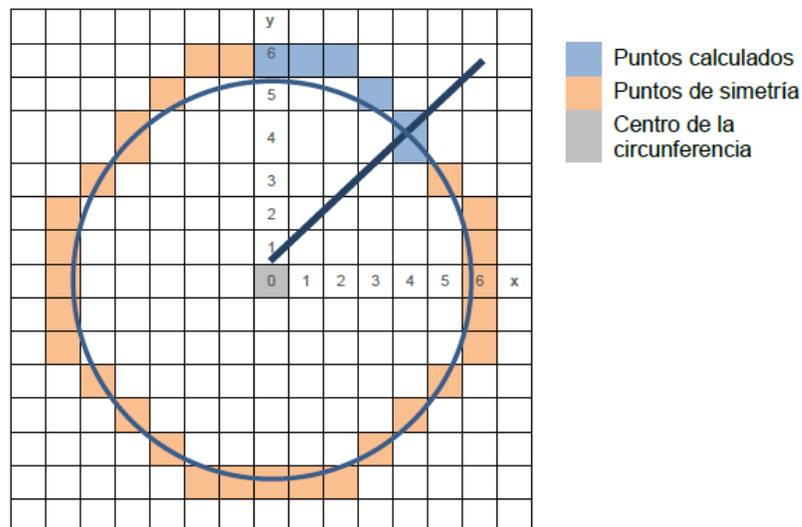
- **Paso 1:** Definir el punto inicial (x_0, y_0) y el valor inicial del parámetro de decisión p .

$$\begin{aligned} (x_0, y_0) &= (0, r) & p_0 &= 3 - 2r \\ (x_0, y_0) &= (0, 6) & p_0 &= 3 - 2(6) \\ & & p_0 &= 3 - 12 \\ & & p_0 &= -9 \end{aligned}$$

- **Paso 2:** Calcular las posiciones de los píxeles

k	x_k	y_k	p_k	$p_k < 0$ (S/N)
0	0	6	-9	S
1	$0 + 1 = 1$	$6 - 0 = 6$	$-9 + 4(0) + 6 = -3$	S
2	$1 + 1 = 2$	$6 - 0 = 6$	$-3 + 4(1) + 6 = 7$	N
3	$2 + 1 = 3$	$6 - 1 = 5$	$7 + 4(2 - 6) + 10 = 1$	N
4	$3 + 1 = 4$	$5 - 1 = 4$	$1 + 4(3 - 5) + 10 = 3$	N

- **Paso 3:** Graficar los puntos obtenidos y completar la circunferencia.



Ejemplo 2: Dada una circunferencia con radio = 5 y su centro en (3, 5). Obtenga las posiciones para el primer octante en el cuadrante 1 y gráfíquelos.

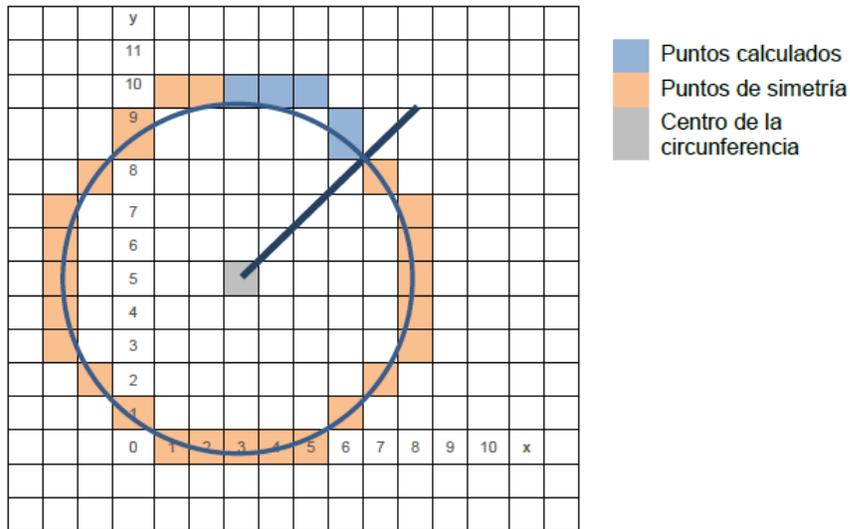
- **Paso 1:** Definir el punto inicial (x_0, y_0) y el valor inicial del parámetro de decisión p .

$$\begin{aligned} (x_0, y_0) &= (0, r) & p_0 &= 3 - 2r \\ (x_0, y_0) &= (0, 5) & p_0 &= 3 - 2(5) \\ & & p_0 &= 3 - 10 \\ & & p_0 &= -7 \end{aligned}$$

- **Paso 2:** Calcular las posiciones de los píxeles, considerando que $x_c=3$ y $y_c=5$.

k	x_k	y_k	p_k	$p_k < 0$ (S/N)	$x_k + x_c$	$y_k + y_c$
0	0	5	-7	S	$0 + 3 = 3$	$5 + 5 = 10$
1	1	$5 - 0 = 5$	$-7 + 4(0) + 6 = -1$	S	$1 + 3 = 4$	$5 + 5 = 10$
2	2	$5 - 0 = 5$	$-1 + 4(1) + 6 = 9$	N	$2 + 3 = 5$	$5 + 5 = 10$
3	3	$5 - 1 = 4$	$9 + 4(2 - 5) + 10 = 7$	N	$3 + 3 = 6$	$4 + 5 = 9$
4	4	$4 - 1 = 3$				

- **Paso 3:** Graficar los puntos obtenidos y completar la circunferencia.



Algoritmo de discretización de elipses

La elipse se caracteriza por ser una **circunferencia alargada y simétrica con respecto a dos ejes**, por ende, posee dos radios o distancias diferentes desde su centro. Como resultado de las modificaciones aplicadas a los algoritmos para dibujar circunferencias se obtiene el llamado **algoritmo de punto medio para el dibujo de elipses**, en el cual se divide el primer cuadrante en **dos regiones**.

Similar a la circunferencia se deben suministrar los radios denominados como r_x y r_y ; además del punto central de la elipse (x_c, y_c) . A continuación, se presentan las condiciones y cálculos para cada región:

- **Región 1:**

El punto inicial (x_0, y_0) está dado por el radio r_y :

$$(x_0, y_0) = (0, r_y)$$

El **valor inicial del parámetro de decisión pa** se obtiene con la ecuación 12:

$$pa_{k=0} = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2 \quad (12)$$

En donde para cada pa_k se verifican las siguientes condiciones, mientras $2r_y^2 x \geq$

$2r_x^2y$:

Si $pa_k < 0$ entonces

$x_k + 1, y_k$ //Se **incrementa x** en una unidad y se **mantiene** el valor de **y**.

$$pa_{k+1} = pa_k + 2r_y^2x_{k+1} + r_y^2$$

Si $pa_k > 0$ entonces

$x_k + 1, y_k - 1$ //Se **incrementa x** una unidad, mientras que **y** se **decrementa**.

$$pa_{k+1} = pa_k + 2r_y^2x_{k+1} + r_y^2 - 2r_x^2y_{k+1}$$

- **Región 2:**

El punto inicial (x_0, y_0) está dado por los últimos puntos obtenidos en la región 1:

$$(x_0, y_0) = (0, y)$$

El **valor inicial del parámetro de decisión pb** se obtiene con la ecuación 13:

$$pb_{k=0} = r_y^2(x_0 + \frac{1}{2})^2 - r_x^2(y_0 + 1)^2 - r_x^2r_y^2 \quad (13)$$

En donde para cada pb_k se verifican las siguientes condiciones, mientras $y > 0$:

Si $pb_k < 0$ entonces

$x_k + 1, y_k - 1$ //Se **incrementa x** una unidad, mientras que **y** se **decrementa**.

$$pb_{k+1} = pb_k + 2r_y^2x_{k+1} - 2r_x^2y_{k+1} + r_x^2$$

Si $pb_k > 0$ entonces

$x_k, y_k - 1$ //Se **mantiene** el valor de **x** y se **decrementa y** en una unidad.

$$pb_{k+1} = pb_k + 2r_x^2y_{k+1} + r_x^2$$

Ejemplo 1: Dados los radios $r_x = 7$ y $r_y = 3$, con el centro en el origen, determine los puntos que conforman la elipse. Considere que:

$$\begin{array}{ll} r_x^2 = 7^2 & r_y^2 = 3^2 \\ r_x^2 = 49 & r_y^2 = 9 \\ 2r_x^2 = 98 & 2r_y^2 = 18 \end{array}$$

Región 1:

- **Paso 1:** Determinar el valor de pa_0 y el punto inicial (x_0, y_0) .

$$pa_0 = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2 \quad (x_0, y_0) = (0, r_y)$$

$$pa_0 = 3^2 - 7^2(3) + \frac{1}{4}(7^2) \quad (x_0, y_0) = (0, 3)$$

$$pa_0 = -126$$

- **Paso 2:** Calcular los valores de x_k , y_k y pa_k para cada iteración mientras $2r_y^2 x \geq 2r_x^2 y$. En donde:

k	(x_k, y_k)	pa_k	$2r_y^2 x$	$2r_x^2 y$
0	(0,3)		0	294
1	(1,3)	$-126 + 18(1) + 9 =$	18	294
2	(2,3)	$-99 + 18(2) + 9 =$	36	294
3	(3,3)	$-54 + 18(3) + 9 =$	54	294
4	(4,2)	$9 + 18(4) + 9 - 98(2) =$	72	196
5	(5,2)	$-106 + 18(5) + 9 =$	90	196
6	(6,2)	$-7 + 18(6) + 9 =$	108	196
7	(7,1)	$110 + 18(7) + 9 - 98(1) =$	126	98

Región 2:

- **Paso 4:** Determinar el valor de pb_0 y el punto inicial (x_0, y_0) .

$$(x_0, y_0) = (x_k, y_k)$$

$$(x_0, y_0) = (7, 1)$$

$$pb_0 = r_y^2(x_0 + \frac{1}{2})^2 - r_x^2(y_0 + 1)^2 - r_x^2 r_y^2$$

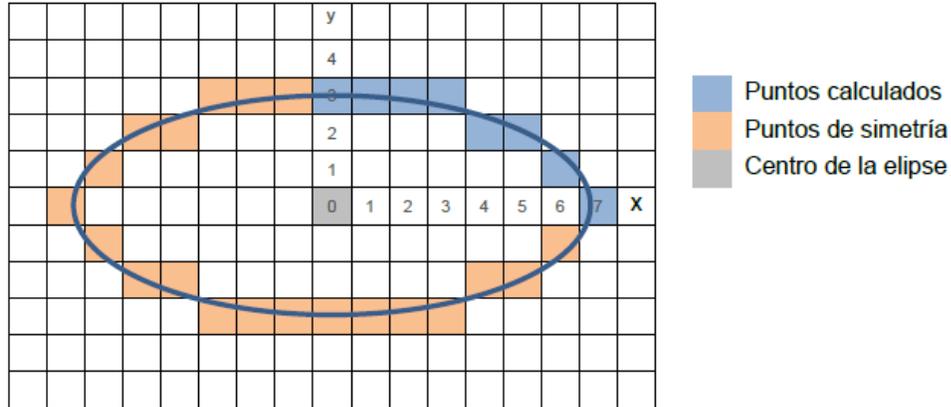
$$pb_0 = 3^2(7 + \frac{1}{2})^2 - 7^2(1 + 1)^2 - 3^2(7^2)$$

$$pb_0 = 65$$

- **Paso 5:** Calcular los valores de x_k , y_k y pb_k para cada iteración mientras $y > 0$.

k	(x_k, y_k)	pb_k
0	(7,1)	65
1	(7,0)	$65 - 98(0) + 49 =$ 114

- **Paso 6:** Graficar.



Ejemplo 2: Dados los radios $r_x = 4$ y $r_y = 6$, con el **centro en (1,2)**, determine los puntos que conforman la elipse. Considere que:

$$\begin{aligned} r_x^2 &= 4^2 & r_y^2 &= 6^2 \\ r_x^2 &= 16 & r_y^2 &= 36 \\ 2r_x^2 &= 32 & 2r_y^2 &= 72 \end{aligned}$$

Región 1:

- Paso 1:** Determinar el valor de pa_0 y el punto inicial (x_0, y_0) .

$$\begin{aligned} pa_0 &= r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2 & (x_0, y_0) &= (0, r_y) \\ & & (x_0, y_0) &= (0, 6) \\ pa_0 &= 6^2 - 4^2(6) + \frac{1}{4}(4^2) \\ pa_0 &= -56 \end{aligned}$$

- Paso 2:** Calcular los valores de x_k , y_k y pa_k para cada iteración mientras $2r_y^2 x \geq 2r_x^2 y$. En donde: $x_c = 1$ y $y_c = 2$.

k	(x_k, y_k)	pa_k	$2r_y^2 x$	$2r_x^2 y$	$x_k + x_c$	$y_k + y_c$	
0	(0,6)	-56	0	192	1	8	
1	(1,6)	$-56 + 72(1) + 36 =$	52	72	2	8	
2	(2,5)	$52 + 72(2) + 36 - 32(5) =$	72	144	3	7	
3	(3,4)	$72 + 72(3) + 36 - 32(4) =$	196	216	128	4	6

Región 2:

- Paso 4:** Determinar el valor de pb_0 y el punto inicial (x_0, y_0) .

$$\begin{aligned} (x_0, y_0) &= (x_k, y_k) \\ (x_0, y_0) &= (3, 4) \\ pb_0 &= r_y^2 \left(x_0 + \frac{1}{2}\right)^2 - r_x^2 (y_0 + 1)^2 - r_x^2 r_y^2 \end{aligned}$$

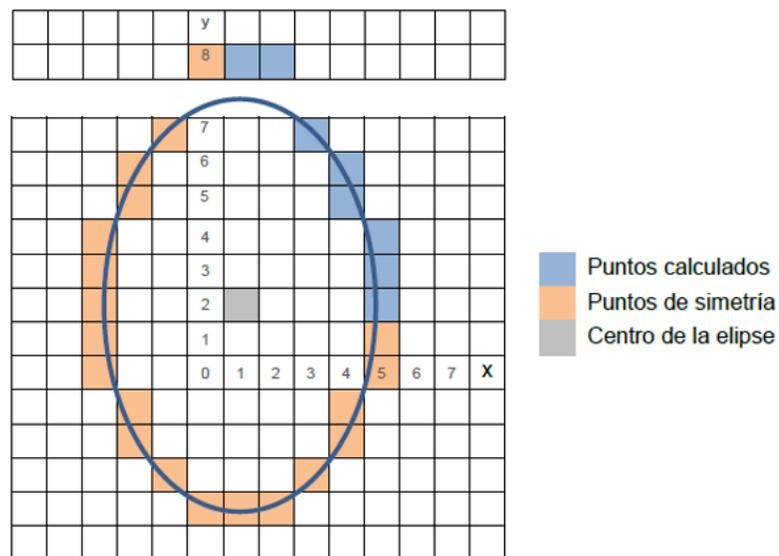
$$pb_0 = 6^2(3 + \frac{1}{2})^2 - 4^2(4 - 1)^2 - 4^2(6)$$

$$pb_0 = 9$$

- **Paso 5:** Calcular los valores de x_k , y_k y pb_k para cada iteración mientras $y > 0$.

k	(x_k, y_k)	pb_k	$x_k + x_c$	$y_k + y_c$
0	(3,4)	9	4	6
1	(3,3)	$9 - 32(3) + 16 =$	4	5
2	(4,2)	$-71 + 72(4) - 32(2) + 16 =$	5	4
3	(4,1)	$169 - 32(4) + 16 =$	5	3
4	(4,0)		5	2

- **Paso 6:** Graficar.



Algoritmo de relleno de figuras

Los algoritmos de relleno de figuras son los utilizados para otorgar o reemplazar el color existente dentro de las celdas o píxeles que componen una figura (**mapa de bits**). Ya con un enfoque a herramientas de diseño gráfico, esta acción la suele realizar la opción conocida como **“bote de pintura”**, que, con hacer clic dentro de una región, cambia al color especificado.

El algoritmo mayormente conocido es el **FloodFill**. Es un método recursivo que trabaja utilizando una **pila** en la que almacena todas las posiciones que conforman la zona a colorear.

De forma general, el algoritmo FloodFill trabaja de la siguiente forma, **mientras la pila no esté vacía:**

- a) Se ingresa un punto dentro de la figura, este es el punto inicial. Se cambia su color al deseado.
- b) Se realiza una verificación de los puntos vecinos, ya sea tomando cuatro direcciones (norte, sur, este, oeste) u ocho direcciones (norte, sur, este, oeste, noreste, sureste, noroeste, suroeste) y se almacenan en la pila.
- c) Se saca el último elemento de la pila, se cambia su color, y nuevamente se verifican sus vecinos.

Ejemplo: Mediante el algoritmo de FloodFill reemplace el color celeste por el color amarillo en la siguiente figura, teniendo en consideración 8 direcciones. Tome como punto inicial **(2,3)** (fila, columna).

	0	1	2	3	4	5
0						
1						
2				x		
3						
4						
5						

Visitados:
(2,3)

Pila (Vecinos):
[(1,4) (1,3) (1,2) (2,2) **(3,3)**]

	0	1	2	3	4	5
0						
1						
2						
3				x		
4						
5						

Visitados:
(2,3) (3,3)

Pila (Vecinos):
[(1,4) (1,3) (1,2) (2,2) **(4,3)**]

(2,3) ya no es vecino porque ha cambiado de color.

	0	1	2	3	4	5
0						
1						
2						
3						
4				x		
5						

Visitados:
(2,3) (3,3) (4,3)

Pila (Vecinos):
[(1,4) (1,3) (1,2) **(2,2)**]

(4,3) no tiene vecinos, se pasa al siguiente punto (2,2).

	0	1	2	3	4	5
0						
1						
2			x			
3						
4						
5						

Visitados:
(2,3) (3,3) (4,3) (2,2)

Pila (Vecinos):
[(1,4) (1,3) (1,2) (1,3) (1,2) (2,1) **(3,1)**]

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Visitados:

(2,3) (3,3) (4,3) (2,2) (3,1)

Pila (Vecinos):

[(1,4) (1,3) (1,2) (1,3) (1,2) (2,1) (2,1) (2,0) **(3,0)**]

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Visitados:

(2,3) (3,3) (4,3) (2,2) (3,1) (3,0)

Pila (Vecinos):

[(1,4) (1,3) (1,2) (1,3) (1,2) (2,1) (2,1) (2,0) (2,0) **(2,1)**]

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Visitados:

(2,3) (3,3) (4,3) (2,2) (3,1) (3,0) (2,1)

Pila (Vecinos):

[(1,4) (1,3) (1,2) (1,3) (1,2) (2,1) (2,1) (2,0) (2,0) (2,0) **(2,1)**]

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Visitados:

(2,3) (3,3) (4,3) (2,2) (3,1) (3,0) (2,1) (1,2)

Pila (Vecinos):

[(1,4) (1,3) (1,2) (1,3) (1,2) (2,1) (2,1) (2,0) (2,0) (2,0) **(1,3)**]

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Visitados:

(2,3) (3,3) (4,3) (2,2) (3,1) (3,0) (2,1) (1,2) (1,3)

Pila (Vecinos):

[(1,4) (1,3) (1,2) (1,3) (1,2) (2,1) (2,1) (2,0) (2,0) (2,0) **(1,4)**]

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Visitados:

(2,3) (3,3) (4,3) (2,2) (3,1) (3,0) (2,1) (1,2) (1,3) (1,4)

Pila (Vecinos):

[(1,4) (1,3) (1,2) (1,3) (1,2) (2,1) (2,1) (2,0) (2,0) **(2,0)**]

	0	1	2	3	4	5
0						
1						
2	x					
3						
4						
5						

Visitados:

(2,3) (3,3) (4,3) (2,2) (3,1) (3,0) (2,1) (1,2) (1,3) (1,4) (2,0)

Pila (Vecinos):

[(1,4) (1,3) (1,2) (1,3) (1,2) (2,1) (2,1) (2,0) (2,0) (2,0)]

Los puntos restantes ya han sido visitados, dejando así la pila vacía y finalizando el algoritmo.

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Visitados:

(2,3) (3,3) (4,3) (2,2) (3,1) (3,0) (2,1) (1,2) (1,3) (1,4) (2,0)

Pila (Vecinos):

[]

Proceso completado.

Métodos de transformación

En esta sección se explican los métodos de transformación o **transformaciones geométricas** aplicables a las figuras u objetos que se encuentran dentro de una escena. Las transformaciones más comunes son las que implican **mover**, **rotar** o **escalar elementos**. Las transformaciones vistas en los siguientes puntos están enfocadas en un plano **bidimensional**.

Traslación

La traslación por definición consiste en mover o trasladar un objeto de una posición **P** a otra **P'** siguiendo una trayectoria en línea recta (Stegman, 2015). Las posiciones corresponden a la ubicación del objeto dentro de un sistema de coordenadas. La distancia en la que se traslada el objeto es denotada por la letra **t**. Sabiendo esto, tenemos la siguiente formula:

$$P' = (x', y') = P + t \quad (14)$$

La ecuación 14 se desglosa en dos ecuaciones; una para cada eje de coordenadas:

$$x' = x + t_x \quad (15)$$

$$y' = y + t_y \quad (16)$$

En las ecuaciones 15 y 16 se observa que se tiene una distancia **t** para cada eje. En computación gráfica, la ecuación 14 se representa mediante una suma de matrices de la siguiente forma:

$$\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Ejemplo: Traslade cinco (5) unidades a la izquierda y cuatro (4) unidades arriba, al punto ubicado en (6, 1).

Utilizando las ecuaciones 15 y 16, este problema se resuelve de la siguiente manera:

$$x' = x + t_x$$

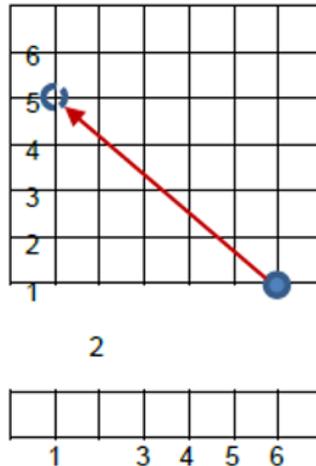
$$y' = y + t_y$$

Mover el punto a la izquierda o abajo implica un valor negativo.

Mover el punto a la derecha o arriba implica un valor positivo.

$$\begin{aligned} x' &= 6 + (-5) \\ x' &= 1 \end{aligned}$$

$$\begin{aligned} y' &= 1 + 4 \\ y' &= 5 \end{aligned}$$



Rotación

La rotación por su parte consiste en mover los objetos sobre un punto de pivote (x, y) , dado un ángulo θ (Steezman, 2015). Esto obedece el hecho de que, si el ángulo es positivo, el giro será en sentido contrario a las manecillas del reloj; y si es negativo, va en dirección de las manecillas del reloj.

La ecuación general para esta transformación es la siguiente:

$$p' = (x', y') = r \cdot P \quad (17)$$

En donde r representa el ángulo de rotación que se aplicará sobre el punto P . Para efectos de representación y cálculo se recurre al uso de las identidades trigonométricas. Desglosando la ecuación 17, se obtiene lo siguiente:

$$x' = x \cos\theta - y \operatorname{sen}\theta \quad (18)$$

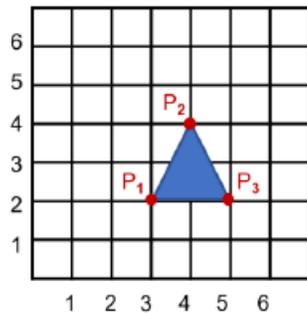
$$y' = x \operatorname{sen} \theta - y \operatorname{cos} \theta \quad (19)$$

La representación matricial para la ecuación 17 es:

$$\begin{pmatrix} \operatorname{cos} \theta & -\operatorname{sen} \theta \\ \operatorname{sen} \theta & \operatorname{cos} \theta \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix} = P'$$

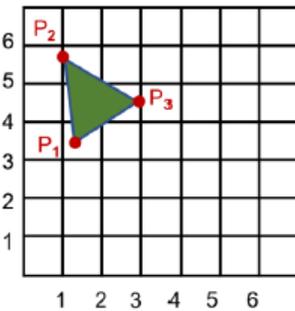
Ejemplo: Rote la siguiente figura en un ángulo $\theta = 35^\circ$. La figura está compuesta por tres puntos principales P_1 , P_2 y P_3 .

- **Paso 1:** Aplicar las ecuaciones 18 y 19 a los puntos principales del objeto o figura.



$P_1 = (3,2)$	$x' = 3 \operatorname{cos} 35 - 2 \operatorname{sen} 35 = 1.31$ ≈ 1.3	$P'_1 = (1.3, 3.4)$
	$y' = 3 \operatorname{sen} 35 + 2 \operatorname{cos} 35 = 3.35$ ≈ 3.4	
$P_2 = (4,4)$	$x' = 4 \operatorname{cos} 35 - 4 \operatorname{sen} 35 = 0.98$ ≈ 1.0	$P'_2 = (1.0, 5.6)$
	$y' = 4 \operatorname{sen} 35 + 4 \operatorname{cos} 35 = 5.57$ ≈ 5.6	
$P_3 = (5,2)$	$x' = 5 \operatorname{cos} 35 - 2 \operatorname{sen} 35 = 2.95$ ≈ 3.0	$P'_3 = (3.0, 4.5)$
	$y' = 5 \operatorname{sen} 35 + 2 \operatorname{cos} 35 = 4.51$ ≈ 4.5	

- **Paso 2:** Graficar los puntos obtenidos en con las ecuaciones.



Nótese que la figura ha sido **rotada en su totalidad** con respecto al **origen**.

Si se desea aplicar la rotación sobre un **punto en específico** de la figura, por ejemplo, el P_1 , este punto ha de ser **trasladado al origen** y luego se aplican las ecuaciones sobre el resto de los puntos de la figura.

Escalado

Las transformaciones anteriores, traslación y rotación, consistían en un movimiento de puntos o de una figura **sin alterar sus proporciones**; no obstante, durante el escalado **sí es posible que se presenten deformaciones**.

Un factor de escalado menor a uno, reducirá el tamaño en el eje que se esté aplicando; por otra parte, un factor mayor a uno aumenta o expande la figura en el eje indicado. **Los factores de escalado no necesariamente deben ser iguales para ambos ejes** (Stegman, 2015).

A continuación, se muestran las ecuaciones para escalar las dimensiones de un punto:

$$x' = x \cdot S_x \quad (20)$$

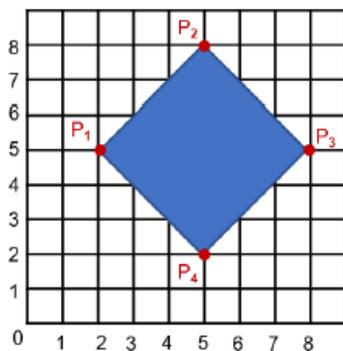
$$y' = y \cdot S_y \quad (21)$$

$$\begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

En donde S_x y S_y son los factores de escalado. En forma matricial:

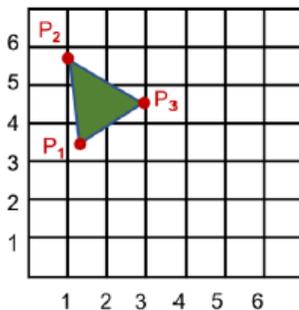
Ejemplo: Dado un factor de (0.5, 0.8) escale la siguiente figura:

- **Paso 1:** Multiplique las x e y de los puntos que conforman la figura por el factor de escalado correspondiente. Se recuerda que $S_x = 0.5$ y $S_y = 0.8$.



$P_1 = (2,5)$	$x' = 2(0.5) = 1$ $y' = 5(0.8) = 4$	$P'_1 = (1, 4)$
$P_2 = (5,8)$	$x' = 5(0.5) = 2.5$ $y' = 8(0.8) = 6.4$	$P'_2 = (2.5, 6.4)$
$P_3 = (8,5)$	$x' = 8(0.5) = 4$ $y' = 5(0.8) = 4$	$P'_3 = (4, 4)$
$P_4 = (5,2)$	$x' = 5(0.5) = 2.5$ $y' = 2(0.8) = 1.6$	$P'_4 = (2.5, 1.6)$

- **Paso 2:** Se grafican los puntos obtenidos.



El resultado es una figura con proporciones distintas a la inicial, debido a que los factores de escalado no eran iguales para ambos ejes.

Se observa también que hubo un desplazamiento de la figura, para corregir esto se recurre muchas veces a la traslación.

Simetría y recorte

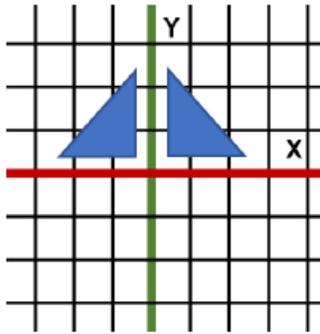
Al hablar de simetría se hace referencia a una condición particular en la que un objeto o gráfico en 2D al ser dividido por una línea, **la porción de un lado de la línea es un reflejo de la porción del otro lado** (Jo, 1999).

El principio de simetría es de gran utilidad en el entorno de computación gráfica, principalmente durante el modelado de objetos, como jarrón o un auto; puesto que solo se requiere modelar y definir detalles de un solo lado, para entonces aplicar alguna

herramienta que copie y genere un reflejo de la parte trabajada, completando así la figura en menor tiempo (Jo, 1999).

De esta forma se cuenta con cuatro tipos de simetría con su representación matricial (Lopez, 2005):

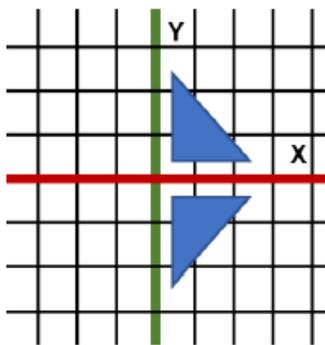
a) Simetría con **respecto al eje y**:



Para obtener los puntos de simetría con respecto al eje y, se deben multiplicar los puntos por la siguiente matriz:

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

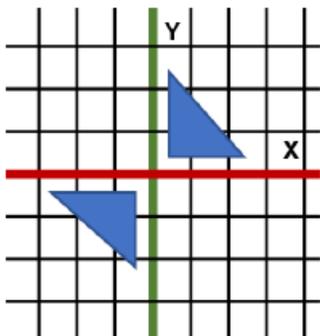
b) Simetría con **respecto al eje x**:



Los puntos de simetría con el eje x se obtienen al multiplicar los puntos por la siguiente matriz:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

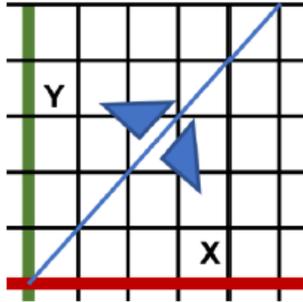
c) Simetría con **respecto al origen**:



La simetría con respecto al origen se da al multiplicar los puntos por la matriz:

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

d) Simetría con **respecto a la recta x=y**:



Los puntos de simetría en relación a una recta donde los valores de x son iguales a los de y, se obtienen al multiplicar por:

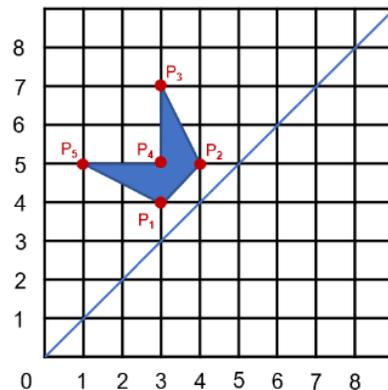
$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

La **simetría** que se da **con respecto a un eje**, también se conocen con el nombre de **reflexión**.

Nota: Si se requiere obtener la simetría de una **recta distinta** a las mencionadas, se debe evaluar la aplicación de **traslación** y **rotación** según sea necesario, sobre dicha recta de modo que se asemeje a uno de los cuatro casos anteriores.

Ejemplo: Complete la siguiente figura considerando que es simétrica con respecto a la recta $x=y$.

- **Paso 1:** Identificar los puntos principales **P** que componen la figura y su representación en forma de **vector (x y)**.



$$P_1 = (3,4) \quad (3 \ 4)$$

$$P_2 = (4,5) \quad (4 \ 5)$$

$$P_3 = (3,7) \quad (3 \ 7)$$

$$P_4 = (3,5) \quad (3 \ 5)$$

$$P_5 = (1,5) \quad (1 \ 5)$$

- **Paso 2:** Multiplicar los vectores por la matriz de simetría correspondiente para este caso.

$$(3 \ 4) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = [(3)(0) + (4)(1) \quad (3)(1) + (4)(0)] = (4 \ 3)$$

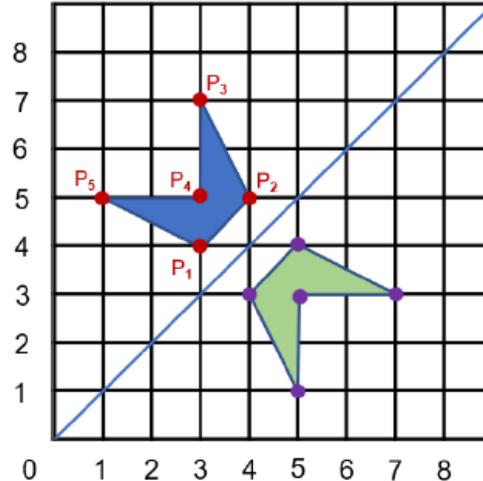
$$(4 \ 5) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = [(4)(0) + (5)(1) \quad (4)(1) + (5)(0)] = (5 \ 4)$$

$$(3 \ 7) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = [(3)(0) + (7)(1) \quad (3)(1) + (7)(0)] = (7 \ 3)$$

$$(3 \ 5) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = [(3)(0) + (5)(1) \quad (3)(1) + (5)(0)] = (5 \ 3)$$

$$(1 \ 5) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = [(1)(0) + (5)(1) \quad (1)(1) + (5)(0)] = (5 \ 1)$$

- **Paso 3:** Se grafican los puntos obtenidos.



En cuanto al tema de recorte, como en el caso de la discretización de líneas y circunferencias, existen distintos algoritmos, cada uno con sus ventajas y desventajas. Esta sección se enfocará en el **algoritmo de Weiler-Atherton**.

Este algoritmo de recorte se caracteriza por devolver más de un polígono resultante, a diferencia del algoritmo de Sutherland-Hodgeman, en el que caso como un polígono cóncavo, genera la aparición de líneas inexistentes en la figura original. Bajo determinadas condiciones se mueve por los bordes de la **ventana de recorte**.

Una ventana de recorte es un rectángulo que delimita la zona que se desea mantener, es decir, que todo lo que quede fuera del área de la ventana será eliminado.

Es importante señalar que se dan tres casos posibles durante el recorte:

- La figura es **visible**: La figura completa queda dentro de la ventana.
- La figura es **parcialmente visible**: Una sección de la figura queda dentro de la ventana.
- La figura **no es visible**: Ninguna sección de la figura queda dentro de la ventana.

El algoritmo de recorte de Weiler-Atherton **siempre inicia desde un punto externo o fuera de la pantalla**.

Ejemplo: Se desea recortar la siguiente figura. El algoritmo de Weiler-Atherton **conserva lo que se encuentra dentro de la ventana** y descarta el resto. Se debe tomar un **punto externo para comenzar**, en este caso será el **P₁**.

	<p>El borde color morado simula la ventana de recorte que se ha colocado sobre la figura verde.</p>
	<p>Se inicia el recorrido en P₁ hacia P₂, no obstante, en el camino se encuentra con una intercepción entre el segmento P₁P₂ y el borde de la ventana de recorte.</p> <p>Se identifica este punto y se almacena en una lista junto con P₂.</p> <p>El punto P₁ se descarta dado que se encuentra fuera de la ventana.</p> <p style="text-align: center;">$\{t_1, P_2\}$</p>
	<p>Se continúa con el segmento de P₂ a P₃. P₃ se encuentra dentro de la ventana, por lo tanto, se almacena en la lista.</p> <p style="text-align: center;">$\{t_1, P_2, P_3\}$</p>
	<p>Sigue el trayecto hacia P₄, sin embargo, antes de llegar a ese punto, existe otra intercepción, lo que indica que P₄ es un punto externo.</p> <p>Se almacena la intercepción t₂ en la lista.</p> <p style="text-align: center;">$R_1 = \{t_1, P_2, P_3, t_2\}$</p> <p>El recorrido ha salido de la ventana de recorte, lo que indica que la lista anterior contiene los puntos que conforman la sección de la figura que se va a conservar.</p>

	<p>El algoritmo inicia de nuevo, pero esta vez desde P_4 que fue el último punto visitado y además es un punto externo.</p> <p>En el camino a P_5 aparece una nueva intercepción.</p> <p>Se descarta P_4. Se almacena la intercepción encontrada y P_5 en una nueva lista.</p> <p style="text-align: center;">$\{t_3, P_5\}$</p>
	<p>El último recorrido va de P_5 a P_6. Se encuentra una nueva intercepción, que es almacenada junto a los puntos anteriores.</p> <p style="text-align: center;">$R_2 = \{t_3, P_5, t_4\}$</p> <p>Para este ejemplo sólo se generaron dos polígonos durante el proceso de recorte, que corresponde a la sección parcialmente visible de la figura.</p>
	<p>Finalmente se redibujan los polígonos resultantes de acuerdo a las listas de puntos.</p> <p style="text-align: center;">$R_1 = \{t_1, P_2, P_3, t_2\}$ $R_2 = \{t_3, P_5, t_4\}$</p>

Sistema de coordenadas homogéneas

En los puntos anteriores, relacionados con las transformaciones en 2D, se mostraron las representaciones matriciales para cada movimiento, no obstante, los cálculos se realizaban directamente con las ecuaciones presentadas. El sistema de coordenadas homogéneas tiene como propósito utilizar directamente estas matrices para obtener los resultados deseados, facilitando la multiplicación entre matrices.

Las matrices presentadas en los puntos anteriores son de tamaño 2x2 (2 filas y 2 columnas); en el sistema de coordenadas homogéneas se convierten a matrices 3x3.

Esto se logra al representar los puntos mediante un vector con la forma (hx, hy, h) . En donde hx y hy son los puntos x e y conocidos; mientras que h suele tomar el valor 1.

- **Ejemplo:** El punto **(2, 3)** en coordenadas homogéneas quedaría como **(2, 3, 1)**.

Esta notación otorga a un mismo punto distintas representaciones:

- **Ejemplo:** $(2,3,1) \Rightarrow (1, 1.5, 0.5) \Rightarrow (8, 12, 4)$.
- Nótese que el **valor original de x e y** es alterado de acuerdo **al valor de h** que se elija.

Así solo queda utilizar las representaciones matriciales en sistema de coordenadas homogéneas de las transformaciones de traslación, rotación y escalado. Lo mismo ocurre con las matrices de simetría.

- Matrices de transformación en 2D:

$$\begin{array}{c} \text{Traslación} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \text{Rotación} \\ \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \text{Escalado} \\ \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

- Matrices de simetría:

$$\begin{array}{c} \text{Eje x} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \text{Eje y} \\ \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \text{Origen} \\ \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \text{x = y} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

El uso de un **sistema de coordenadas homogéneas** facilita la aplicación de **transformaciones consecutivas** a un punto o figura.

Capítulo III: Fundamentos de animación 2D

Introducción al bosquejo de la figura humana

Dimensionamiento por tamaño de cabeza



Figura 16. Dimensionamiento por tamaño de cabeza.

Figura masculina y femenina

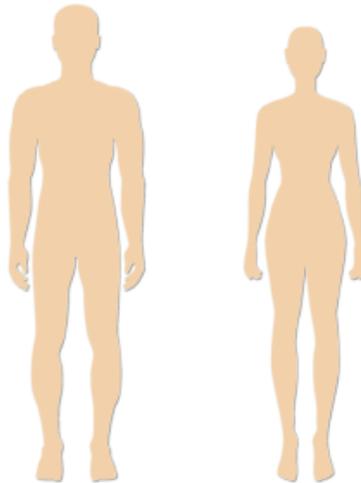


Figura 17. Figura masculina (derecha) y femenina (izquierda).

Rostro

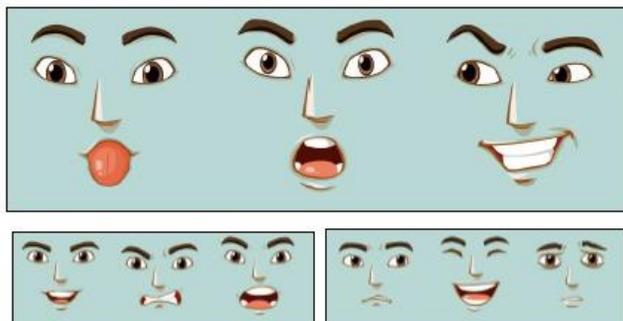


Figura 18. Rostro.

Manos y pies

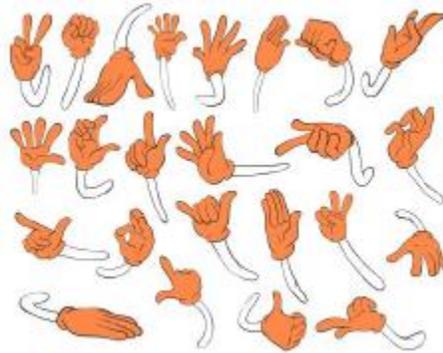


Figura 19. Manos y pies.

Caricatura

Una caricatura es una representación simplificada de un dibujo realista, utilizada normalmente en el sector de entretenimiento para la transmisión de ideas o mensajes con un toque exagerado y divertido, llegando incluso al sarcasmo. Este término se asocia con mayor frecuencia a los cortos animados o “cómicás” como se conocen comúnmente la secuencia de dibujos en 2D que dan como resultado una escena.

Los personajes en estilo caricatura se distinguen por sus rasgos exagerados: cabezas de mayor proporción respecto al cuerpo, ojos, nariz y orejas grandes; además de la simplicidad de elementos como el cabello.

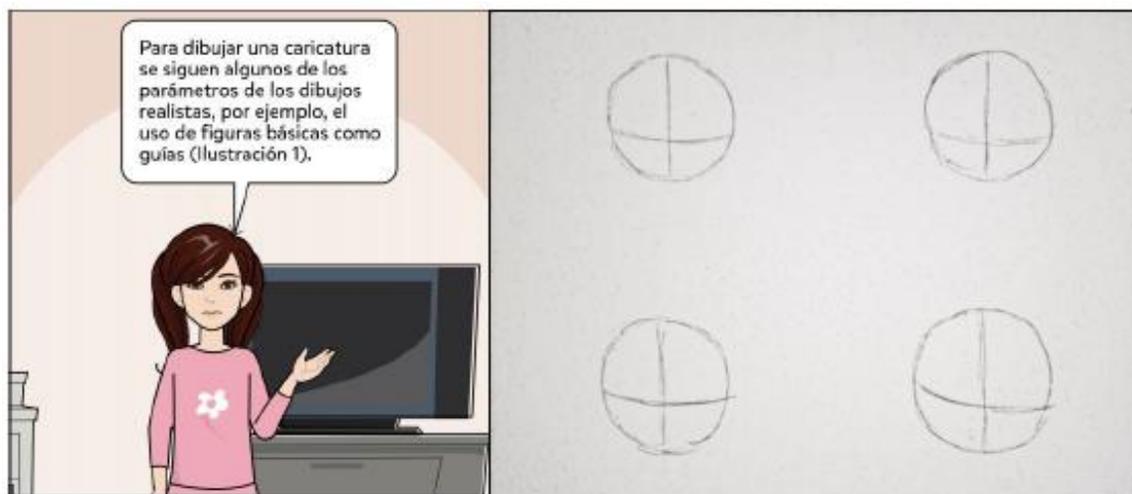


Figura 20. Se trazan ejes horizontales y verticales en los círculos para mantener la proporción de los rasgos del rostro.

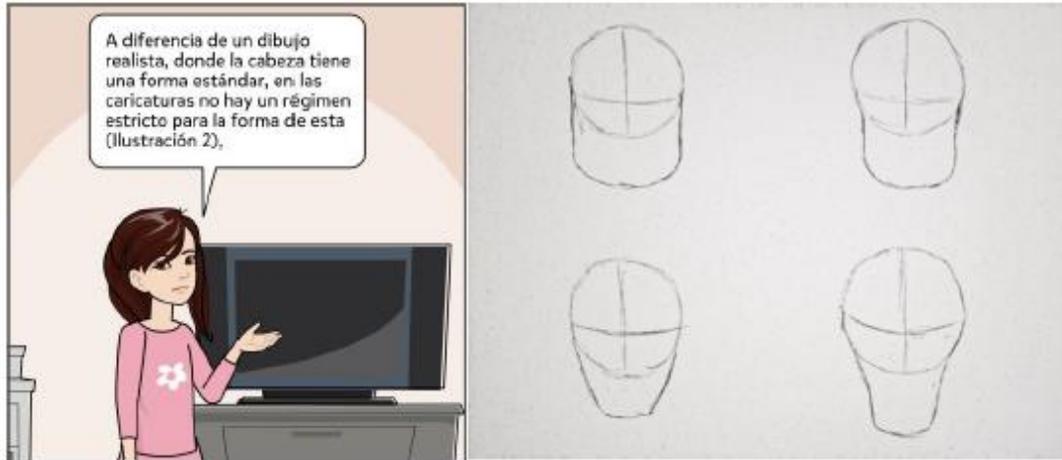


Figura 21. En las caricaturas es común que la forma de la cabeza de los personajes varíe.

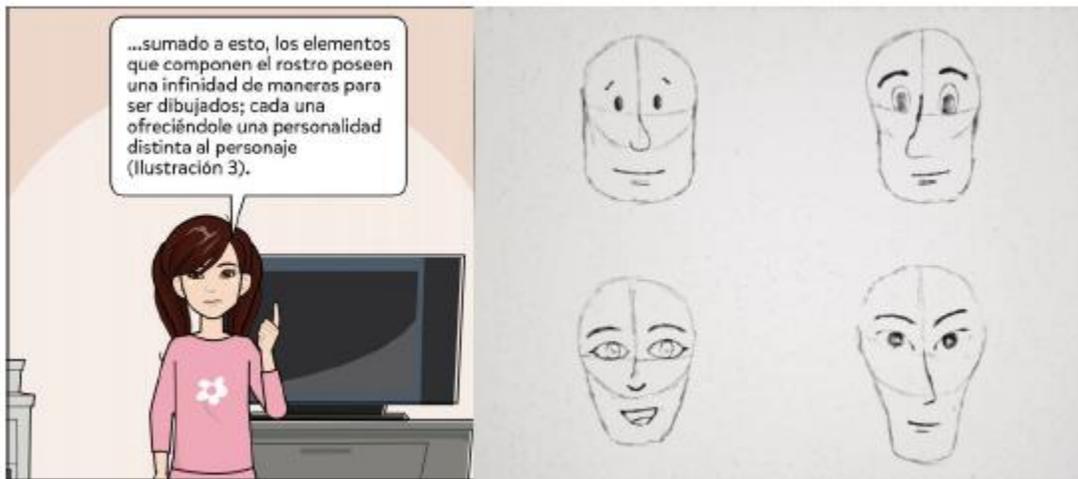


Figura 22. Los elementos del rostro brindan infinidad de personalidades a los personajes.

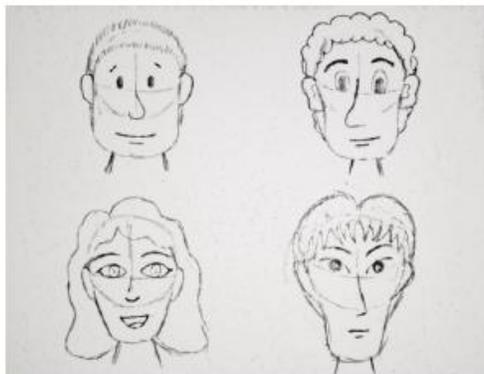


Figura 23. Dibujos de rostros de caricaturas.

El cuerpo en las caricaturas clásicas suele ser de menor proporción con respecto a la cabeza; no obstante, la forma de una pera como base para el cuerpo es una técnica utilizada comúnmente.

En las ilustraciones anteriores se observa la existencia de dos ejes en el círculo principal de la cabeza. Estos dos ejes son una herramienta útil al momento en que el personaje “cambia” la dirección hacia la que está viendo.

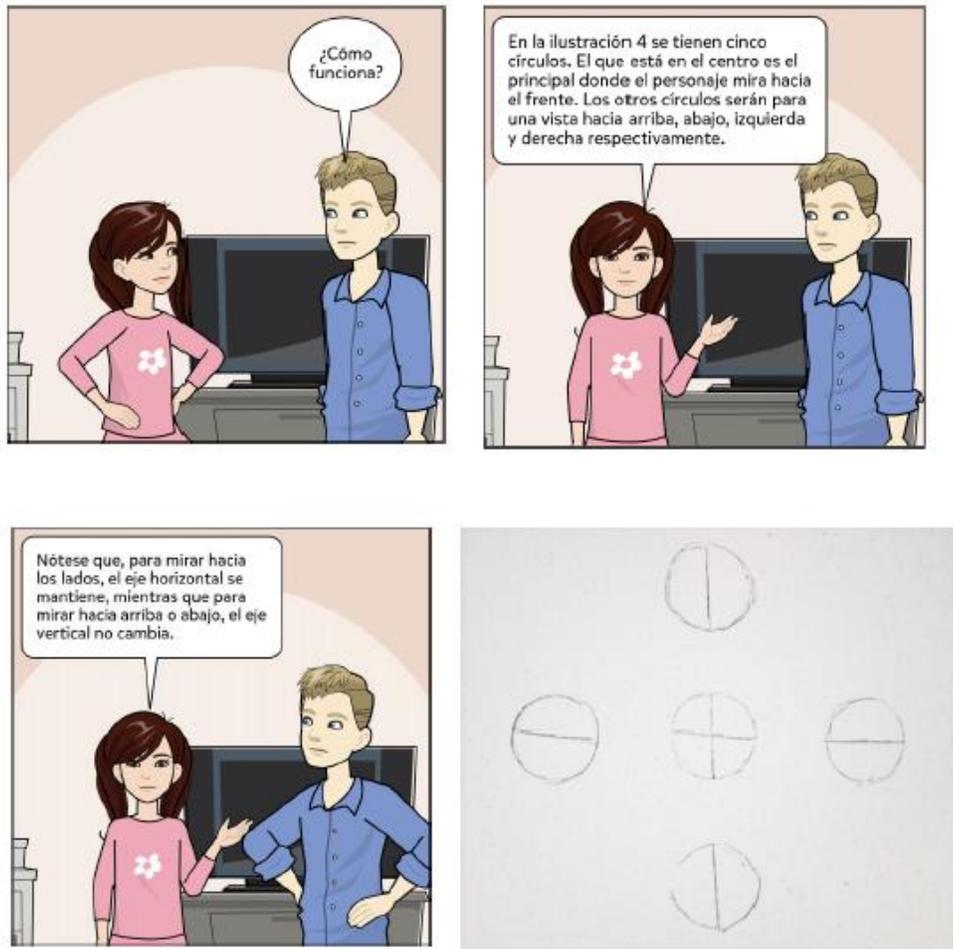


Figura 24. Guías de dirección a la que estará mirando el personaje.



Figura 25. Direcciones de visión del personaje.

Evolución emocional

Las emociones involucran cambios de estados de ánimo causados por una infinidad de factores. Estas emociones pueden ser plasmadas en dibujos, tanto realistas como caricaturas, mediante pequeños cambios en la expresión de los personajes. Los elementos que permiten transmitir esa evolución emocional son las cejas, boca e incluso la nariz.

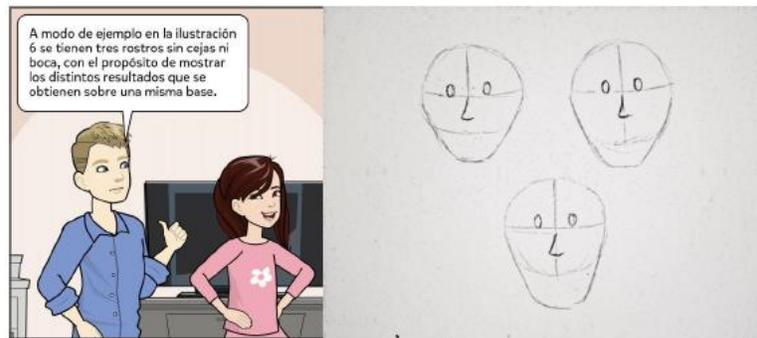


Figura 26. Base para las expresiones.



Figura 27. Expresiones obtenidas con distintos estilos de cejas y boca.

Reglas de tercios

La regla de tercios es una técnica utilizada comúnmente en fotografía, incluso, las cámaras digitales y teléfonos inteligentes incluyen una opción con un nombre similar a “cuadrícula” o “grid”, ilustración 28.

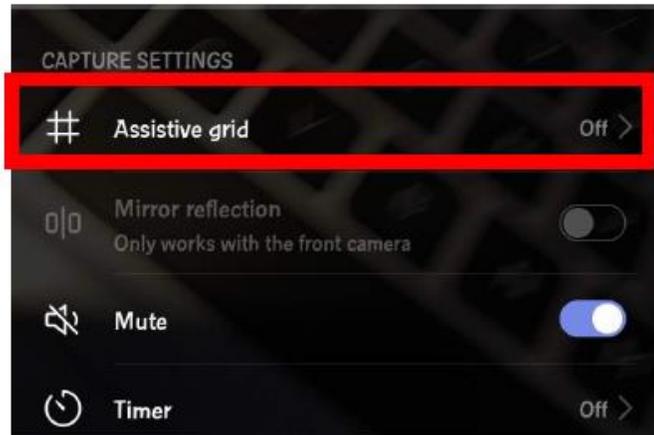


Figura 28. Opción que activa la función de cuadrícula.

Esta herramienta lo que hace es dividir la pantalla en tres tercios horizontales y tres tercios verticales, obteniendo así nueve cuadrantes. Las intersecciones de estas divisiones se les llaman **puntos fuertes**, ilustración 29.

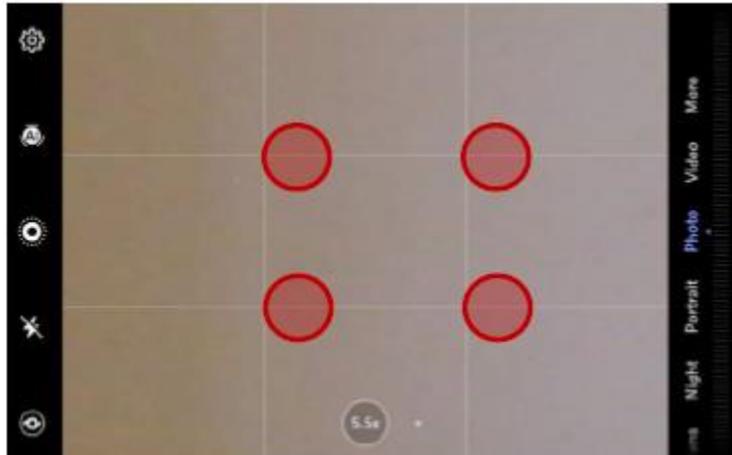


Figura 29. Puntos fuertes.

La técnica consiste en ubicar el centro de interés de la imagen en uno de esos cuatro puntos, ilustración 30. En caso de que existan dos puntos de interés, lo recomendable es situarlos en puntos diagonalmente opuestos.



Figura 30. Ejemplo de aplicación de regla de tercios.

Teoría del color

La teoría del color es un conocimiento básico para los artistas y diseñadores, pero no solo se limita a ellos ya que incluso en la vida cotidiana es común recurrir a este conocimiento hasta para el simple hecho de combinar las piezas de vestir para asistir a la universidad, una entrevista o reunión familiar; con el propósito de adecuarlo a cada ocasión.

La teoría del color se enfoca en los lineamientos para combinar los colores, considerando una serie de atributos como **matiz**, **luminosidad** y **saturación**; y para esto, se vale de una herramienta conocida como **círculo cromático**.

El círculo cromático es una paleta de colores en la que se incluyen los colores primarios rojo, azul y amarillo; de los cuales se derivan los colores secundarios, terciarios, ilustración 31. En este círculo también se subdividen los colores en cálidos y fríos.

Las combinaciones comunes son las siguientes:

- **Complementarios:** Se combinan los colores que se encuentran opuestos en el círculo cromático. Ejemplo: rojo y verde en la ilustración 31.
- **Análogos:** Se refiere a tres colores situados de manera consecutiva en el círculo cromático. Ejemplo: verde, turquesa y azul, de la ilustración 31.
- **Tríadicas:** Es la combinación resultante de tres colores equidistantes, como el verde naranja y violeta marcados por los lados del triángulo negro en la ilustración 31.

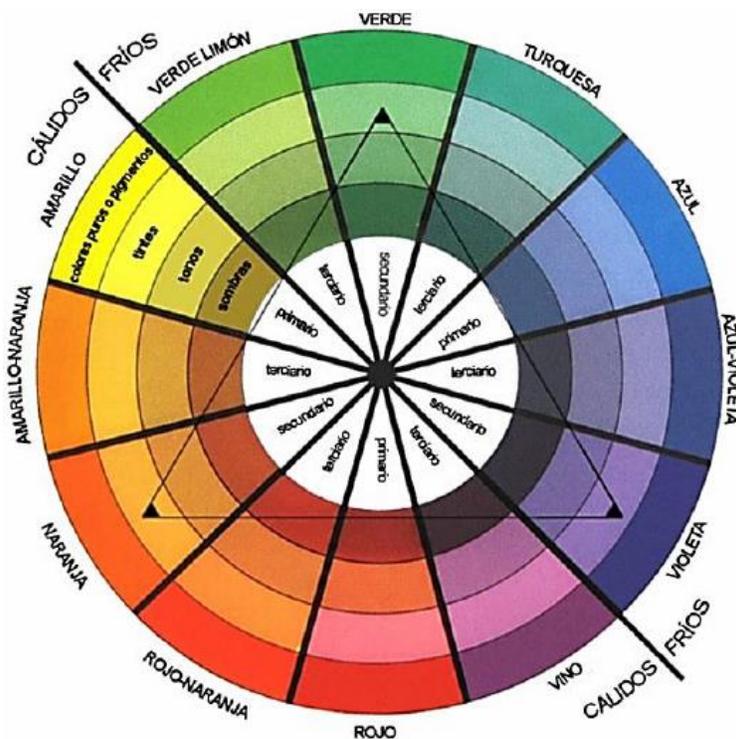


Figura 31. Círculo cromático.

Principios de animación

Los principios básicos de animación tienen por autores a los animadores Ollie Johnston y Frank Thomas, ambos de la compañía Disney. Estos principios se dieron a conocer en la obra: *The illusion of life: Disney Animation*.

Estos principios son una colección de técnicas para animar personajes, sin embargo, tienen aplicaciones en otros campos. Presentaremos en resumen cada uno de estos principios de animación:

1. **Estirar y encoger:** Aumento y disminución del tamaño de los cuerpos.
2. **Anticipación:** Permite anticipar la acción a realizar.
3. **Puesta en escena:** Resaltar la presencia en escena del objeto y resaltar su forma.
4. **Acción directa y pose a pose:** Secuencia de pasos que conducen a una acción no esperada.
5. **Acción continuada y superposición:** Se mezclan movimientos para resaltar la acción.
6. **Entradas lentas y salidas lentas:** La acción se divide en entrada, centro y salidas. La entrada y salidas son ralentizadas mientras que el centro se acelera la acción.
7. **Arcos:** Las trayectorias de arcos son más realistas y naturales.
8. **Acción secundaria:** Movimientos que complementan la acción principal para resaltarla.

- 9. Ritmo (Timing):** Número de eventos o cuadros que se utilizan para una acción en un tiempo dado. Contribuye a la realidad de los objetos.
- 10. Exageración:** Incrementar el efecto de una acción.
- 11. Dibujos sólidos:** Resaltar las perspectivas de los objetos para dar volumen y peso.
- 12. Personalidad o apariencia:** Garantizar que el objeto se coherente con sus movimientos.

Synfig

El software de código libre Synfig es la abreviatura de Synfig Studio y permite editar gráficos vectoriales para posteriormente animar estos gráficos. Su creador fue Robert Quattlebaum y ha sido creado para el diseño de películas animadas con el mínimo de recursos para realizarlo.



Figura 32. Synfig.

Capítulo IV: Herramientas de generación de imágenes 3D

Introducción a la herramienta

En el capítulo anterior se trataron las técnicas y principios básicos que dieron origen a las representaciones gráficas que existen en la actualidad. Vivimos ahora en la era digital, en donde los gráficos no se limitan a las propiedades de largo y anchura, sino que también incluyen profundidad, un atributo que da paso a la gama de las imágenes tridimensionales o 3D. Estas animaciones y diseños en 3D, son las que ofrecen mayor realismo de las representaciones a los espectadores; beneficios aprovechados por otras áreas como la medicina, robótica, arquitectura, entre otras.

Existen múltiples herramientas creadas para el modelado y desarrollo de animaciones en 3D, algunas de software libre y otros de pago. Algunos ejemplos de estas herramientas de diseño son:

- AutoCAD
- Sketchup,
- Maya,
- Revit,
- 3Ds Max
- Blender

En este capítulo se hará referencia al uso esencial de Blender para la generación de gráficos en 3D.

Blender es el resultado de un proyecto colaborativo de software libre y que está dotado de las funcionalidades necesarias para la creación de gráficos 3D de calidad: modelado, animación, simulación, renderizado, composición, rastreo de movimiento, edición de video e incluso en sus últimas versiones ha añadido la posibilidad de trabajar animaciones en 2D.

Una vez instalada la aplicación, cuando se procede a abrir por primera vez, la pantalla inicial que se muestra, es la que a continuación se presenta en la Figura 33; un aspecto interesante de esta versión (2.8), es que permite la posibilidad de seleccionar la manera en que se realicen las selecciones dentro del entorno, esto debido a que, en versiones anteriores, tenía la peculiaridad de requerir del botón derecho del mouse para tareas como selección, arrastre y otros; lo que podía generar confusión para quienes utilizaran esta herramienta por primera vez.

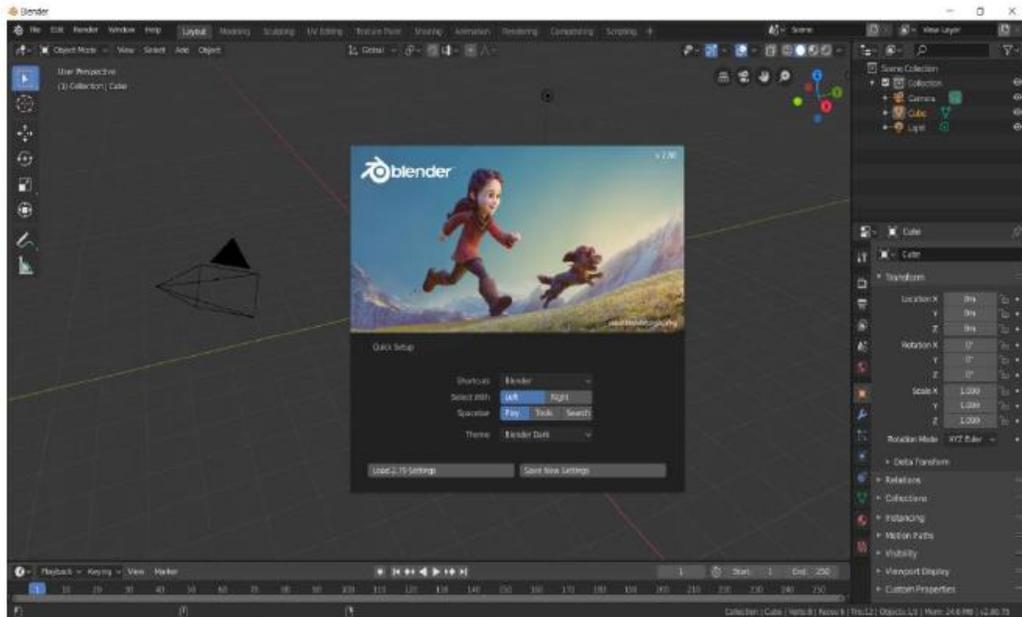


Figura 33. Pantalla de inicio de Blender.

Para efectos del curso mantendremos las configuraciones por defecto de esta versión (Select with **Left**), haciendo clic en **Save New Settings**. Para desaparecer la pequeña ventana de presentación solo es necesario hacer clic en cualquier otro punto de la ventana.

Secciones del área de trabajo

Las secciones de trabajo se encuentran más organizadas en la versión 2.8 en comparación con las anteriores; no obstante, en todas se mantiene la barra de menú con las opciones comúnmente conocidas, a excepción de la sección Render que ofrece acceso rápido para el renderizado de imágenes o videos. Pueden apreciar que, al lado de la barra de menú, se observan una serie de pestañas, que son accesos directos a los espacios de trabajo predefinidos. Vale destacar que estas pestañas se pueden modificar, así como la distribución de las opciones de cada una de ellas, para comodidad del usuario. A continuación, se describen brevemente los espacios de trabajo predefinidos por Blender.

Layout

La ventana que se abre por defecto al ingresar a Blender es **Layout** y se muestra en la Figura 34. Esta ventana contiene todas las herramientas y modificadores de los objetos como **cuerpos sólidos**, es decir, se trabaja el objeto como un todo y no por los elementos que lo componen (caras, aristas y vértices). Cuenta con un panel a la derecha en la que se visualiza un listado de los elementos que están presente en la escena; mientras que debajo de este, se encuentran distintas opciones entre la que destaca la de las transformaciones básicas, localización y escala de los objetos.

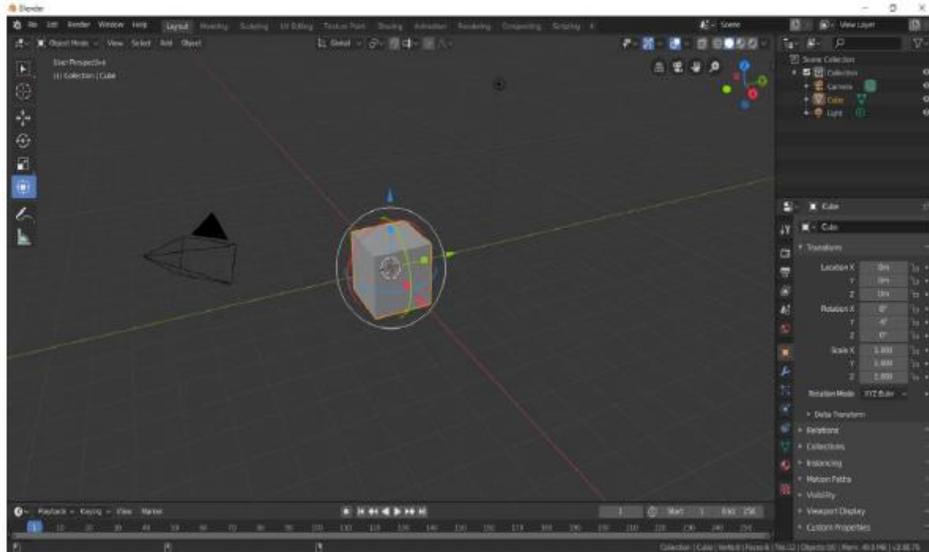


Figura 34. Ventana Layout.

- **Modeling**

Este modo habilita la edición puntual de los componentes de los objetos. En el lado izquierdo de la pantalla se ofrecen un conjunto de opciones como insertar caras, hacer cortes, suavizar bordes, entre otros. Los objetos en este modo se tornan de un tono amarillo como se muestra en la Figura 35.

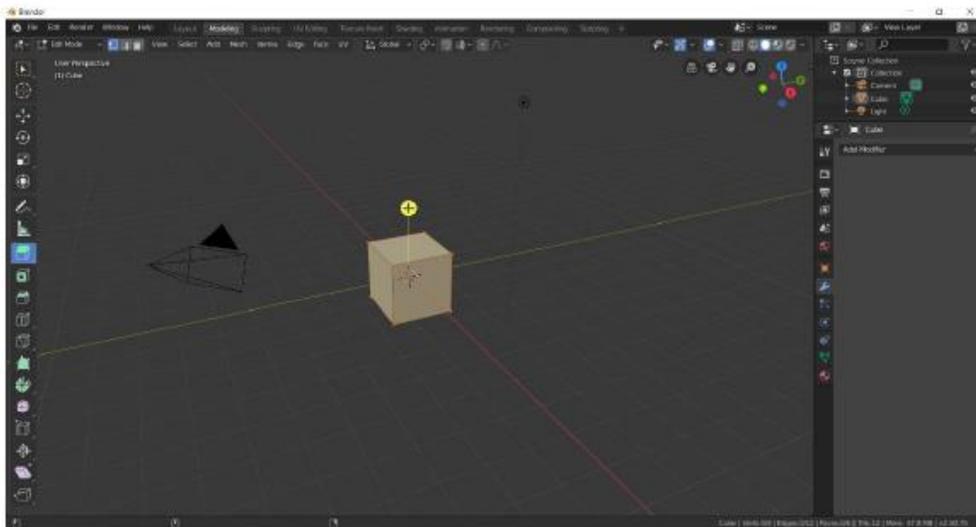


Figura 35. Ventana Modeling.

- **Sculping**

Ofrece una serie de herramientas para alterar la forma original de los objetos desde un modo sólido (Figura 36). Los efectos obtenidos se asocian a relieve sobre la superficie

de los objetos. Entre sus opciones está la posibilidad de trabajar de forma simétrica los cambios, lo que es ideal para objetos que presenten esta propiedad.

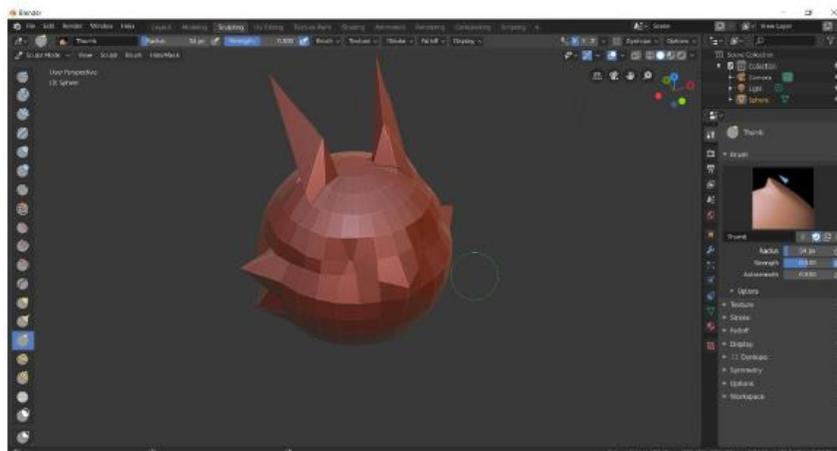


Figura 36. Ventana Sculpting.

- **UV Editing**

En el mundo de digital se cuenta con una serie de recursos gráficos entre las que destacan las imágenes con texturas. En el modo UV editing se ofrece la alternativa de modificar la distribución de una imagen de textura sobre un determinado objeto (ver Figura 37) cuando la presentación obtenida luego de aplicado un material no es la esperada.

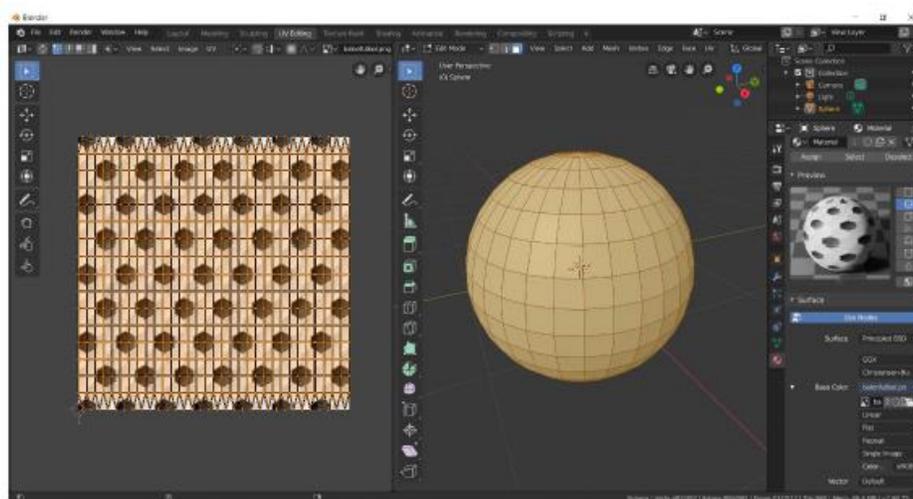


Figura 37. Ventana UV Editing.

- **Texture Paint**

Complemento del modo UV editing. Muchos diseñadores prefieren añadir un toque personal en el material gráfico que trabajan, como se muestra en la Figura 38, en este modo se pueden hacer modificaciones libres a la textura aplicada sobre los objetos.

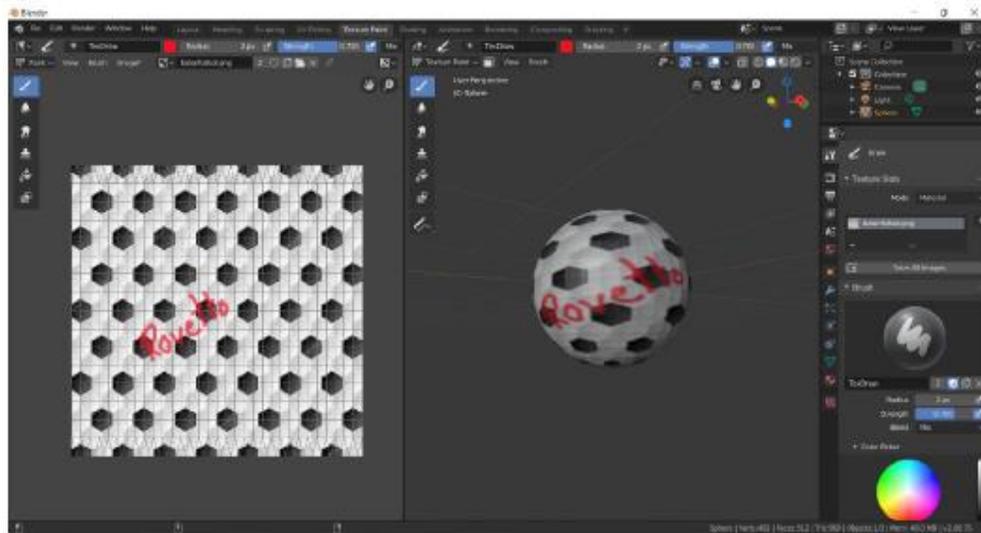


Figura 38. Ventana Texture Paint.

- **Shading**

El shading se refiere a los efectos visuales que tendrá el objeto dentro de la escena 3D. En esta ventana (Figura 39) se modifican los atributos de los objetos para obtener los efectos deseados.



Figura 39. Ventana Shading.

- **Animation**

Modo en el que se modifican las posiciones y tamaño de los objetos en el tiempo con el propósito de generar secuencias de cuadros (frame) que al renderizarse produzcan una

animación. Su característica más sobresaliente es la sección inferior en la que se encuentra una línea de tiempo. En ella se colocan los “keys” (puntos) para fijar la posición de un objeto en un determinado frame (Figura 40).

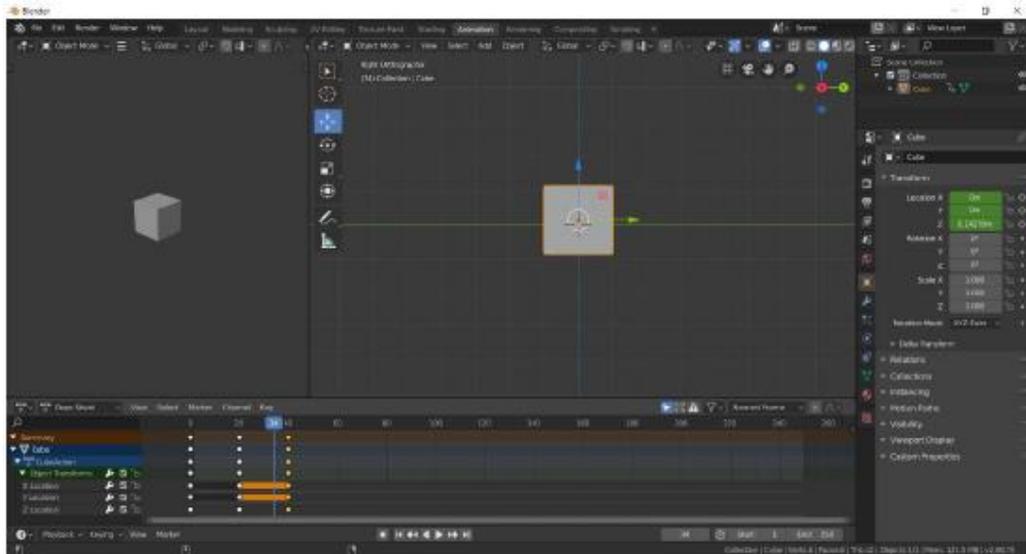


Figura 40. Ventana Animation.

- **Rendering**

Ventana en la que se puede observar el resultado del renderizado (Figura 41).



Figura 41. Ventana Rendering.

- **Compositing**

Alternativa para el tratamiento de imágenes o películas a través de uso de nodos (Figura 42). Una misma configuración puede ser aplicada a una o todas las imágenes de una película con respecto al tiempo.

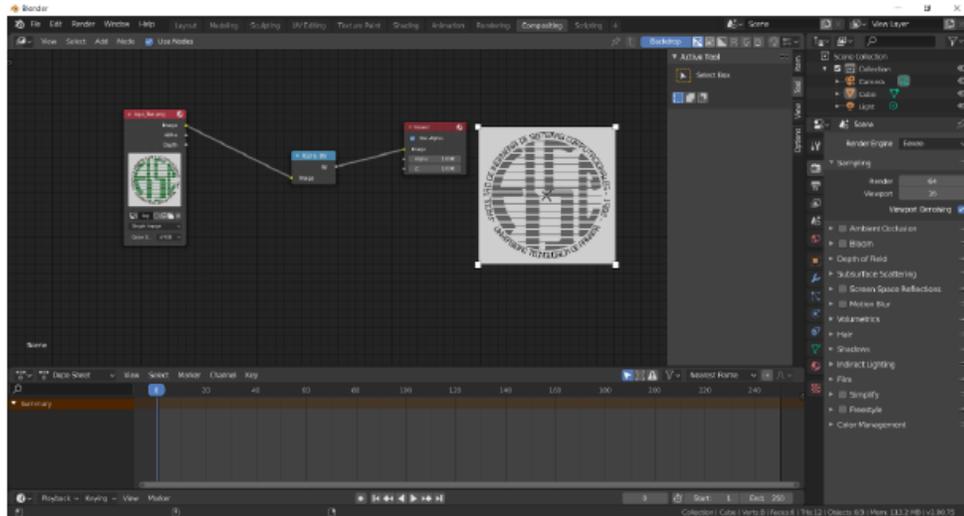


Figura 42. Ventana Compositing.

- **Scripting**

Blender incluye un editor de texto (Figura 43) con el que usuarios avanzados pueden crear, modificar, animar o gestionar procesos complejos sobre los objetos mediante la creación de scripts en Python.

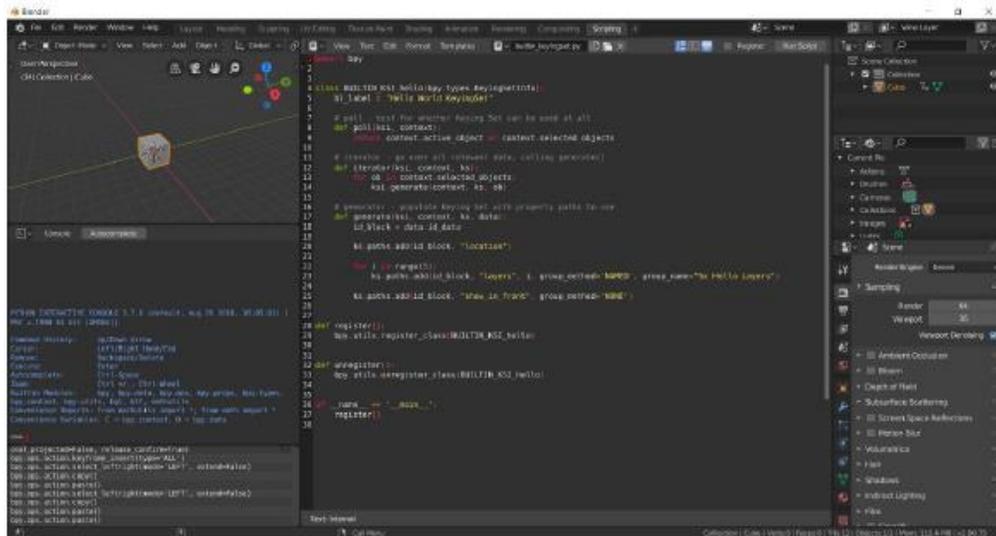


Figura 43. Ventana Scripting.

Vistas

Al hablar de vistas, Blender ofrece dos opciones: User Perspective y User Orthographic. Se cambia entre una y otra con el botón 5 del teclado numérico. En caso de no poseer un teclado numérico, Blender ofrece la alternativa de “emular” estos botones; para esto se accede en el menú **Edit > Preferences > Input** y se marca la opción de “**Emulate Numpad**” (Figura 44). En la Tabla 1 se recopilan las vistas básicas.

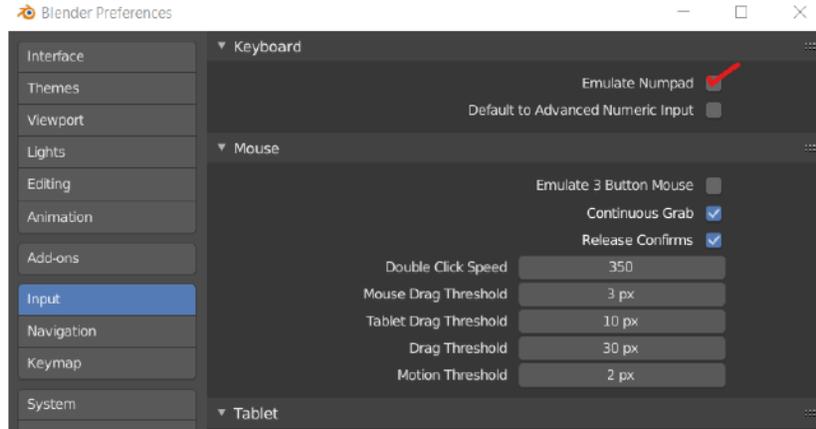


Figura 44. Opción de emular teclado numérico.

VISTA	TECLAS NUMPAD
User Perspective / User Orthographic	5
Front	1
Back	Ctrl + 1
Right	3
Left	Ctrl + 3
Top	7
Bottom	9

Tabla 3. Vistas en Blender.

Paleta de objetos y herramientas

Blender posee una gama de objetos (Figura 45) con distintas finalidades. Se encuentran en la opción Add. Los objetos que se verán en este curso serán Mesh, Curve, Light y Camera. Por su parte, a estos objetos se les aplican transformaciones, efectos espejo, duplicación, uniones, entre otros (Figura 46).

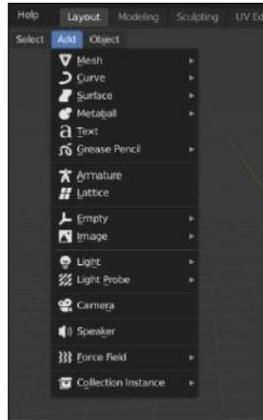


Figura 45. Lista de objetos en Blender.

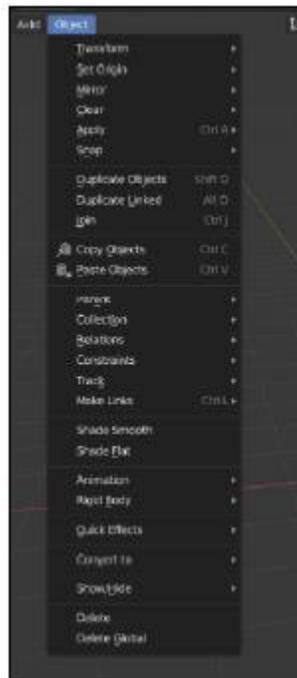


Figura 46. Herramientas para la modificación de objetos.

Transformaciones

Las siguientes transformaciones son las básicas para cualquier objeto y se utilizan mucho al momento de crear animaciones.

Traslación

La Traslación es el movimiento sobre una figura en la que toda su estructura se desplaza en la misma dirección. Una vez realizada la acción tenemos otra figura idéntica, pero en otra posición diferente a la inicial. Cuando el movimiento de traslación se repite en múltiples ocasiones en una misma dirección en un tiempo constante se genera una ilusión óptica de movimiento.

Rotación

La rotación es un giro alrededor de un punto dentro de la figura mientras que se conserva su forma y tamaño original. En el movimiento de rotación influyen factores como la amplitud de la rotación que vendrá determinada por el ángulo. De igual forma está el sentido de la rotación que puede ser en dirección de las agujas de reloj o contrario. Finalmente está el punto de rotación o centro.

Escalados

El escalado de figuras cambia el tamaño en los planos vertical u horizontal. El cambio de la escala se realiza con respecto a un punto de referencia que varía según el plano de escalado que elija.

Formas básicas

La forma básica que Blender muestra por defecto al ingresar al entorno es un cubo; no obstante, posee otras formas y se encuentran dentro de la categoría de **Mesh** en el menú **Add** (Figura 47).

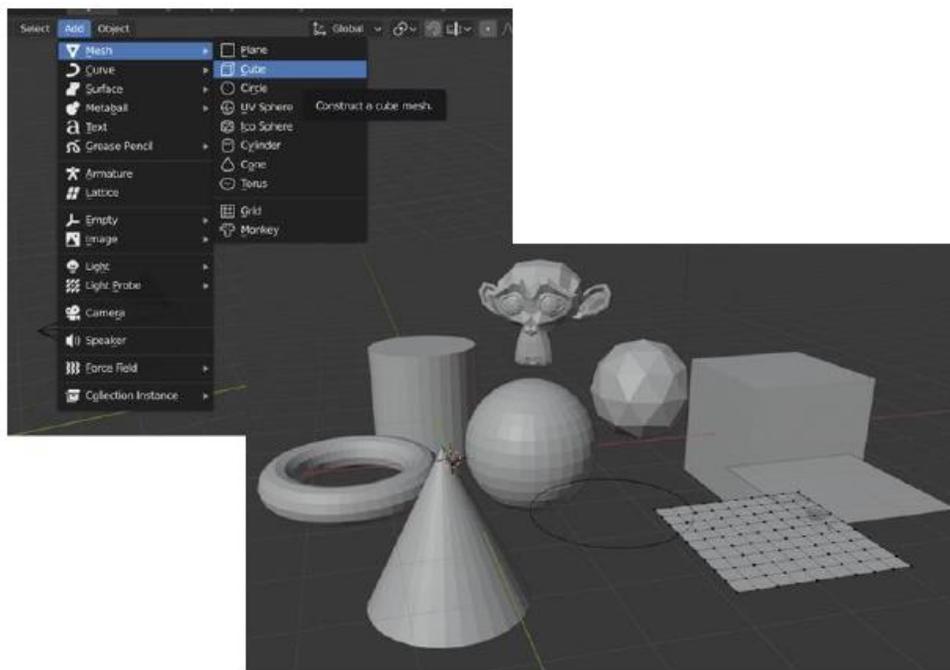


Figura 47. Figuras básicas en Blender.

En la **Figura 47** se observa que dentro de Mesh se encuentran las formas: plano, cubo, círculo, esfera, cilindro, cono, torus, un plano subdividido e incluso una plantilla del rostro de un mono. Estas formas básicas pueden ser modificadas en el **modo edición o pestaña de modelling**, que ofrece otras opciones además de las transformaciones básicas vistas en el video de la sección 1.4.

- **Ejemplo 1:**

- a. Se elimina el cubo que muestra Blender por defecto y se añade un círculo como se muestra en la **Figura 48**.

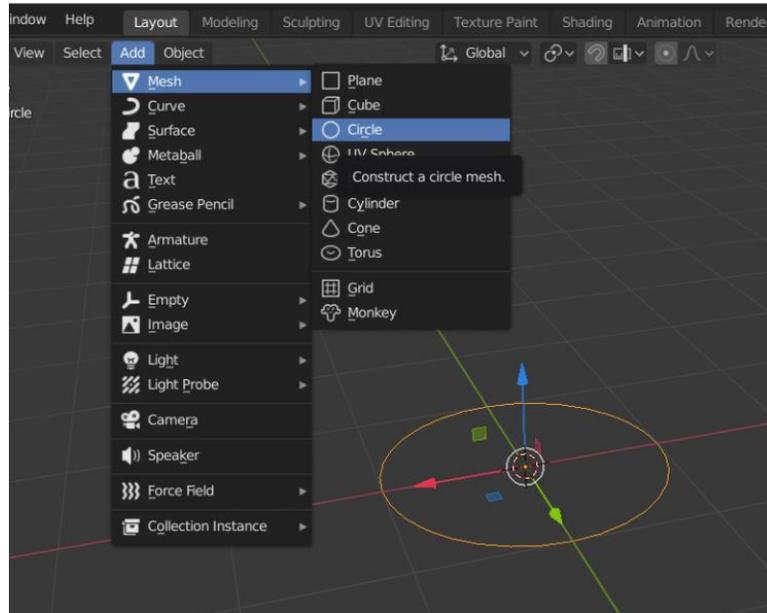


Figura 48. Añadir círculo desde el menú Add.

- b. Se cambia a la ventana o pestaña **Modelling**.
- c. En este modo es posible modificar la forma a través de sus **vértices**, **aristas** o **caras**. En la **Figura 49**, dentro del cuadrante rojo, se ha activado la opción de trabajar por aristas, dado que el círculo añadido no posee ninguna cara.

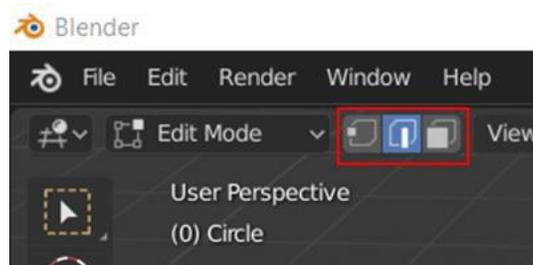


Figura 49. Selección de trabajo por aristas.

- d. Se seleccionan **todas** las aristas. Se toca una de las aristas y luego se introduce la combinación **Ctrl+L**.
- e. El siguiente paso será el añadir una cara al círculo mediante el comando Fill. Luego de seleccionadas las aristas se presiona la tecla **F**. El resultado debe ser como el mostrado en la **Figura 50**.

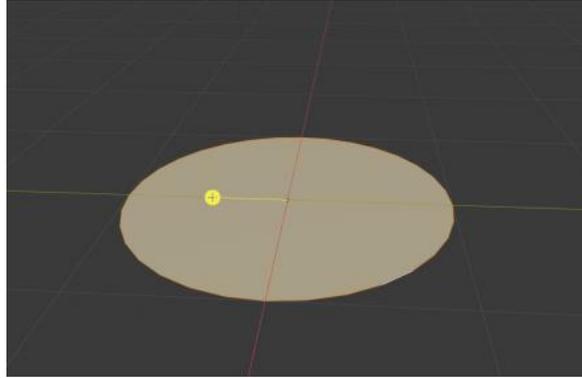


Figura 50. Aplicación del comando Fill sobre un círculo.

- f. A continuación, se procede a insertar una cara dentro del círculo con la tecla **I**. Luego de presionada la tecla se mueve el ratón hacia el centro de la figura y se presiona la tecla **Enter** (Figura 51).

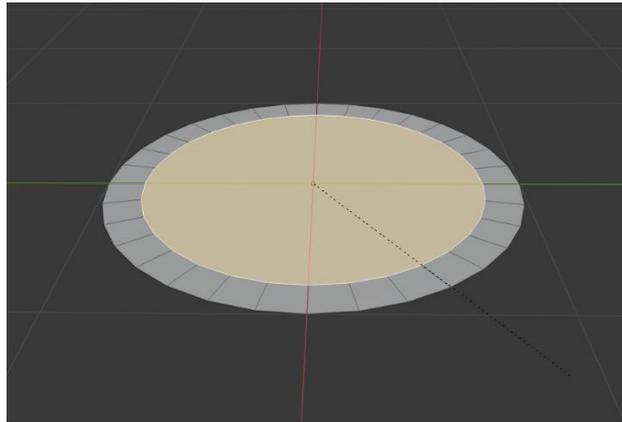


Figura 51. Inserción de una cara dentro del círculo.

- g. Se cambia el modo de trabajo a **caras** de la misma forma que se hizo en el punto c.
- h. Lo siguiente será extruir todas las caras de la figura. Esto se puede hacer de dos formas, mediante la tecla **E** o con el botón que se encuentra en la barra de la izquierda en la ventana de Modelling (Figura 52). En el caso de hacerlo mediante short cut es recomendable especificar un eje (**X**, **Y** o **Z**) luego de la **E**.

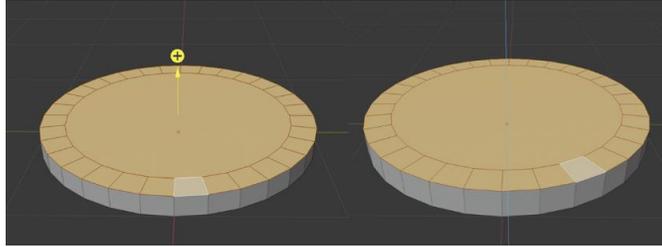


Figura 52. Izquierda: Extruir mediante la barra de herramientas. Derecha: Extruir con el comando E sobre el eje Z (azul).

- i. Ahora se cambiará de vista al ingresar la tecla 3 (right orthographic). Debe mostrarse el objeto como se ve en la **Figura 53**.

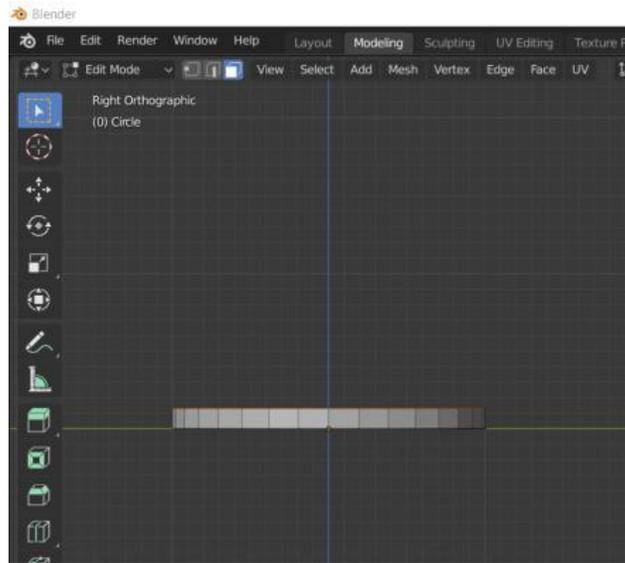


Figura 53. Vista ortogonal derecha.

- j. Se extruyen los bordes del objeto en el eje Z y se escalan como se muestra en la **Figura 54**.

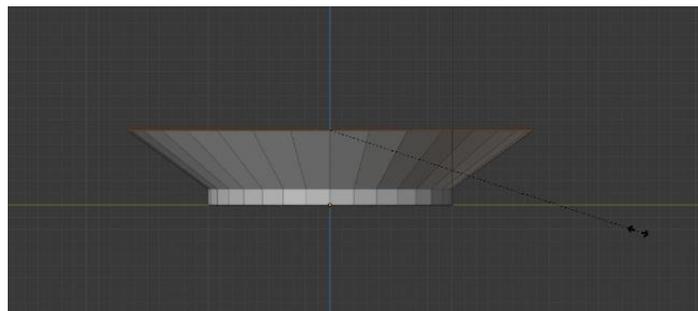


Figura 54. Escalado de los bordes del objeto.

- k. Otra acción aplicable a este objeto es la creación de cortes en las caras existentes, por ejemplo, en las que se acaban de extruir. Con la combinación **Ctrl+R** aparecerá una línea divisoria color amarillo (si se gira la rueda del ratón aparecerán más) se presiona **Enter**. Blender queda en

espera para que ubique la división donde se desea (líneas naranjas). Se presiona **Enter** nuevamente. Si se requiere se escala la división creada como se muestra en la **Figura 55**.

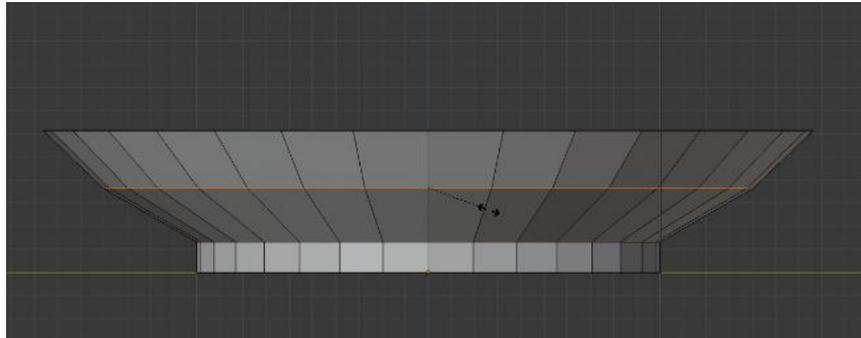


Figura 55. Subdivisión de caras y escalado.

- I. De esta manera se pueden obtener objetos distintos a los predefinidos por Blender (Figura 56).

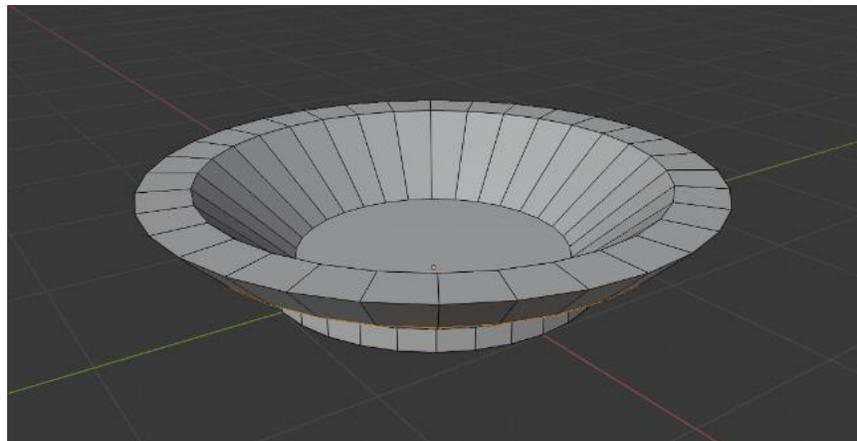


Figura 56. Objeto modificado.

Operaciones binarias

Las operaciones binarias u operaciones booleanas en Blender son las mismas que las tratadas con los diagramas de Venn: unión, diferencia e intersección.

Estas operaciones en Blender se añaden o aplican los objetos mediante el uso de modificadores. Estos se encuentran en el panel derecho y su icono es una llave (**Figura 57**).

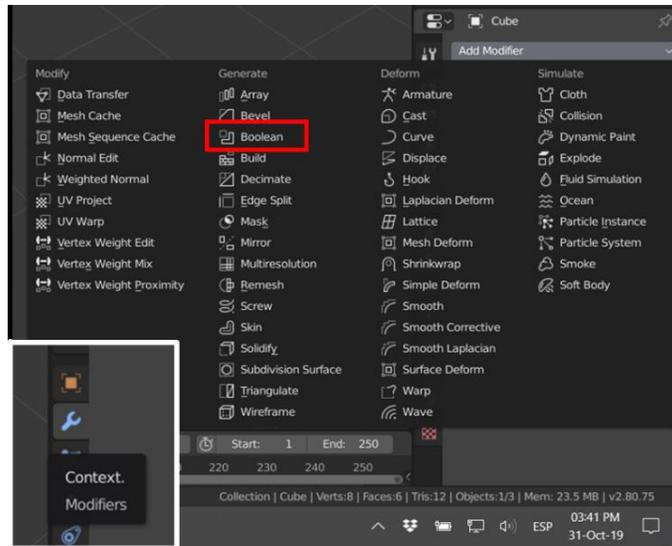


Figura 57. Ícono para acceder a los modificadores de los objetos.

- **Ejemplo 2:**

- Se añaden dos objetos, preferiblemente distintos.
- La regla general para aplicar las operaciones booleanas es seleccionar de último el objeto que heredará o reflejará el resultado de la operación. En la Figura 58 se muestran dos objetos seleccionados. El primero está resaltado en un tono naranja y el segundo (el que hereda el resultado) en un tono amarillo.

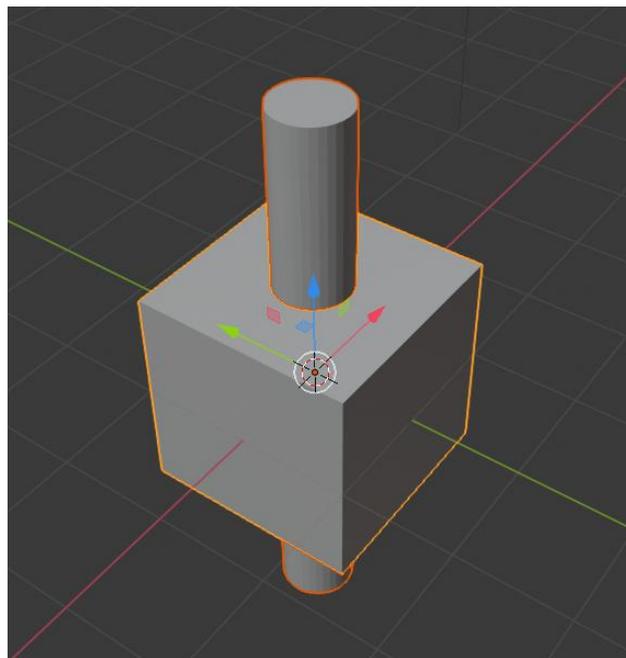


Figura 58. Objetos seleccionados. El último objeto seleccionado está resaltado por un tono amarillo.

- c. Al añadir un modificador de tipo Booleano, aparecen nuevas opciones que se observan en la Figura 59. Resaltado en rojo se encuentra la sección en la que se define el tipo de operación y en amarillo, el objeto del que se obtendrán las características.

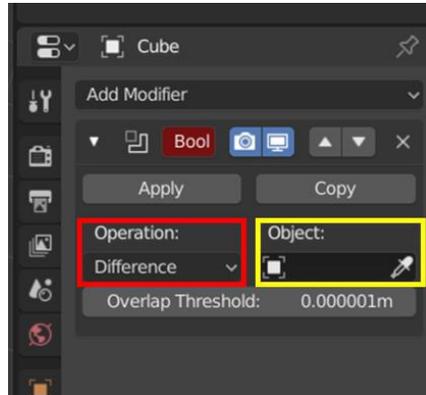


Figura 59. Panel de opciones del modificador booleano.

- d. Para este ejemplo, el cubo heredará las características del cilindro.

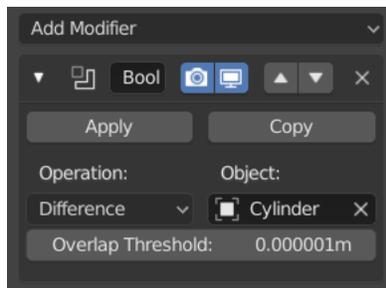


Figura 60. Selección del tipo de operación y objeto del que se heredarán las características.

- e. Una forma de verificar si la operación se está calculando correctamente es cambiando el **viewport de sólido a wireframe (Figura 61)**.

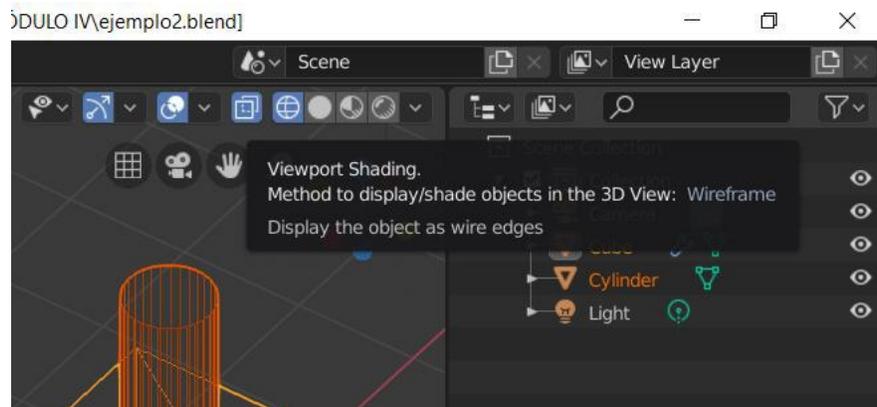


Figura 61. Cambio del viewpoint.

- f. Finalmente, se le da clic al botón **Apply**. Se elimina el objeto original (cilindro) y se regresa el viewport a sólido. El resultado se observa en la Figura 62.

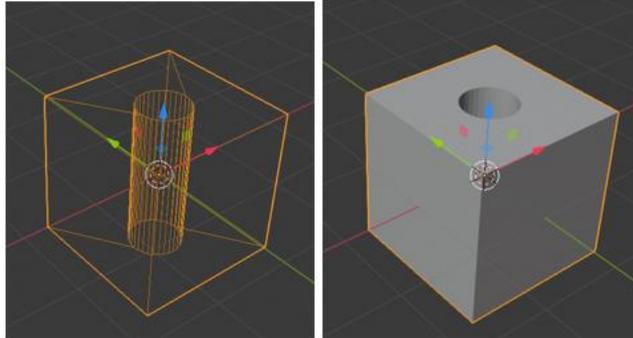


Figura 62. Resultado de la operación diferencia.

- g. De esta misma forma se aplican las otras dos operaciones. En la **Figura 63** se muestra el resultado de unión; mientras que, en la **Figura 64** se muestra el resultado de la intersección. Nótese que en este último se ha cambiado el orden de selección de los objetos, de manera que quien recibe el resultado es el cilindro y no el cubo.

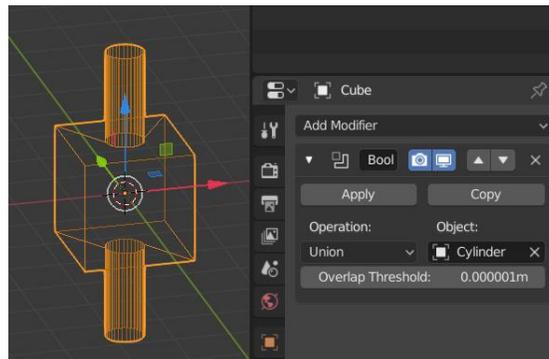


Figura 63. Resultado de la operación Unión.

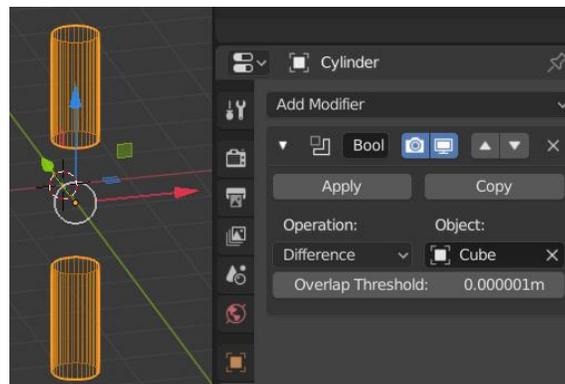


Figura 64. Resultado de operación Intersección.

Fuentes de luz

Blender proporciona opciones de iluminación para dar distintos efectos a las escenas que se estén trabajando. Se añaden a la escena como cualquier otro objeto desde el menú **Add -> Light**.

A cada lámpara se le puede personalizar su color y nivel de intensidad de acuerdo con lo que se requiera.

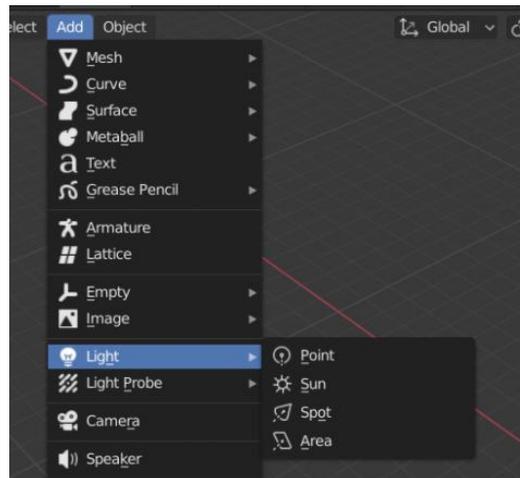


Figura 65. Opción para añadir una nueva fuente de luz.

Point light

El point light o punto de luz, como su nombre lo indica, es un tipo de lámpara que emite la luz en forma **radial**, lo que significa que emite la misma cantidad de luz en todas direcciones. Su apariencia en la escena es de un punto rodeado de dos circunferencias (**Figura 66**). Su intensidad se incrementa dependiendo de su cercanía con el objeto a iluminar.

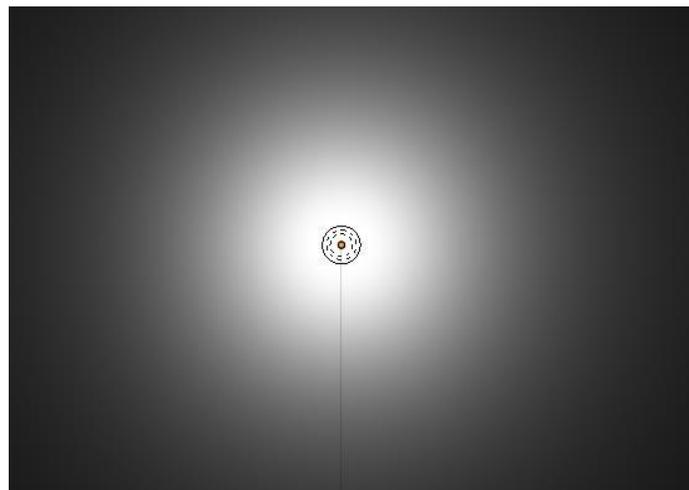


Figura 66. Point light.

Spot light

Es un tipo de lámpara con forma cónica que emite luz en una dirección. Su apariencia dentro de la escena se asemeja a la de un reflector (**Figura 67**).

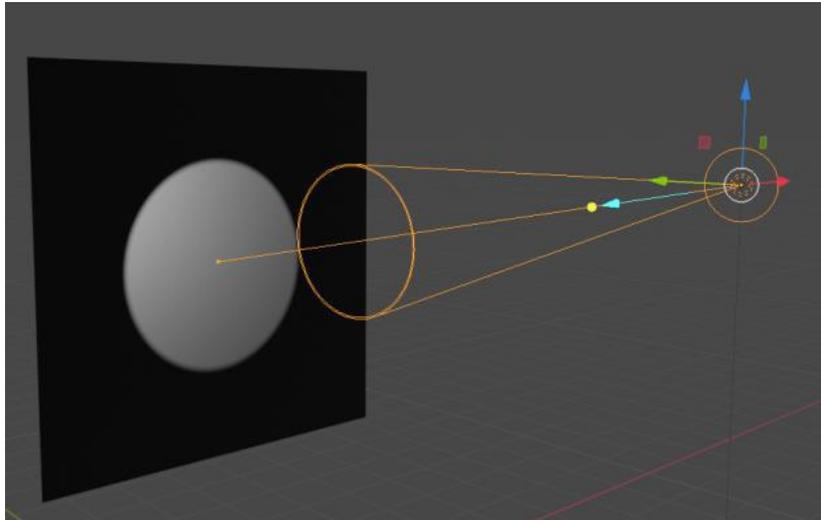


Figura 67. Spot light.

Area light

Lámpara que ofrece un efecto de distribución de luz más realista debido a que no solo ilumina los objetos que tenga cerca, sino que también proporciona una iluminación menos intensa a los objetos cercanos al principal (Figura 68).

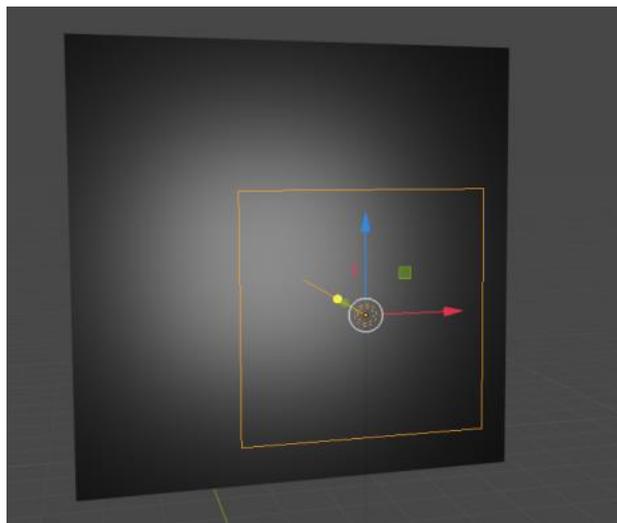


Figura 68. Area light.

Sun light

Tiene un mayor alcance que las anteriores. No se ubica dentro de la escena, no obstante, se controla la dirección en la que proviene la luz que ilumina los objetos (**Figura 69**).

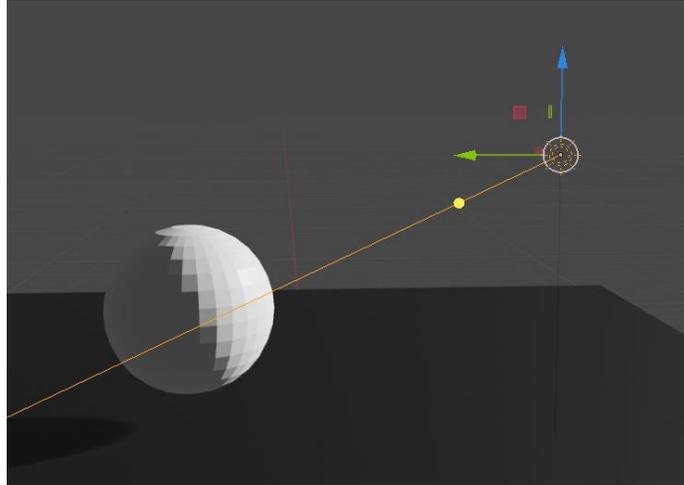


Figura 69. Sun light.

Tipos de cámaras

Para personalizar la cámara o los tipos de cámaras a utilizar para renderizar, Blender posee una opción en su panel lateral derecho, cuyo ícono es una filmadora color verde (Figura 70). Al activar esta opción, se muestra un menú en el que se ofrecen tres tipos de lentes: Perspectiva (Figura 71), Ortogonal (Figura 72) y Panorámico (Figura 73).

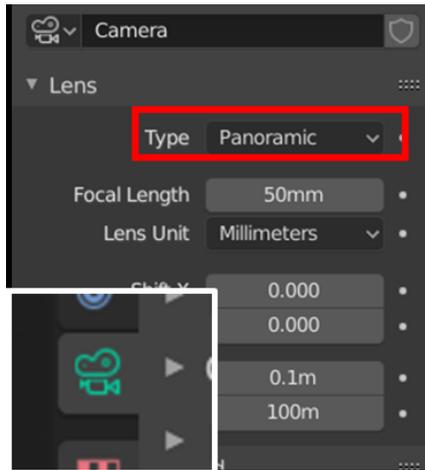


Figura 70. Menú de cámara.



Figura 71. Render con lente en perspectiva.



Figura 72. Render con lente ortogonal.



Figura 73. Render con lente panorámico.

Se observa que no existe mucha diferencia entre el lente en perspectiva y el panorámico debido a que la escena renderizada es sencilla, por otra parte, el lente ortogonal le resta la sensación de realismo al objeto.

Formas basadas en splines

Los splines son una herramienta vista anteriormente en las herramientas para dibujos en 2D y en Blender tienen la misma función: crear curvas. Identificadas como curvas de Bézier son estructuras que permiten definir curvas que de otra forma sería muy complejo modelar en 3D. Similar a otros objetos, se añaden desde el menú **Add -> Curve -> Bézier (Figura 74)**.

Al ingresar al modo edición, se aprecia la aparición de las extensiones en algunos vértices, indispensables para el manejo de las curvas (**Figura 75**). En el panel izquierdo se aprecian otras opciones entre las cuales está la creación de una curva a mano libre.

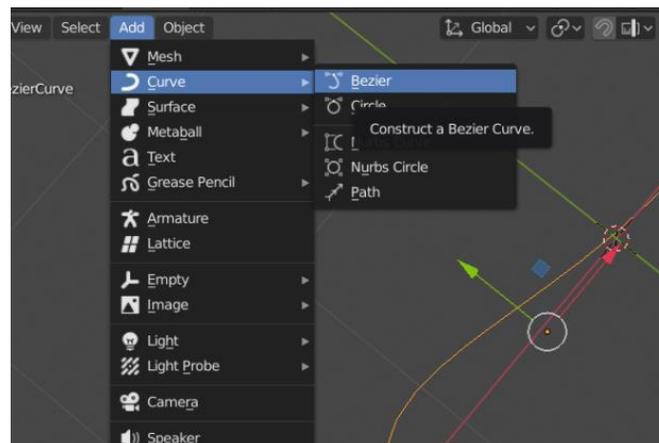


Figura 74. Inserción de una curva de Bézier.

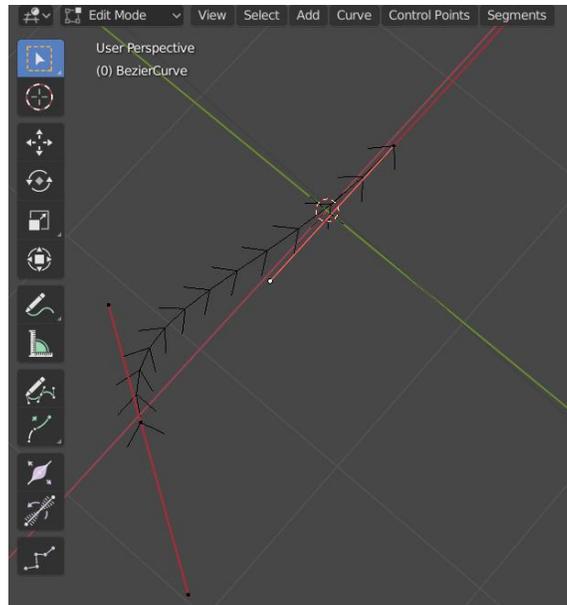


Figura 75. Curva de Bézier en modo edición.

Definida la curva, el siguiente paso es darle aspecto 3D, para esto, en el modo objeto, se selecciona la curva y en el panel lateral derecho se encontrará un botón que despliega las opciones para las curvas de Bézier. Dentro de ese panel se configura la opción **Fill Mode a Full (Figura 76)**. Seguido a esto, en la pestaña de **Geometry -> Bevel**, se cambiará el valor de la **Depth** para darle volumen a la curva (**Figura 77**).

Nota: El valor de Resolution debe ser bajo, de lo contrario al transformar la curva a Mesh, tendrá demasiados vértices y será difícil manejar el objeto.

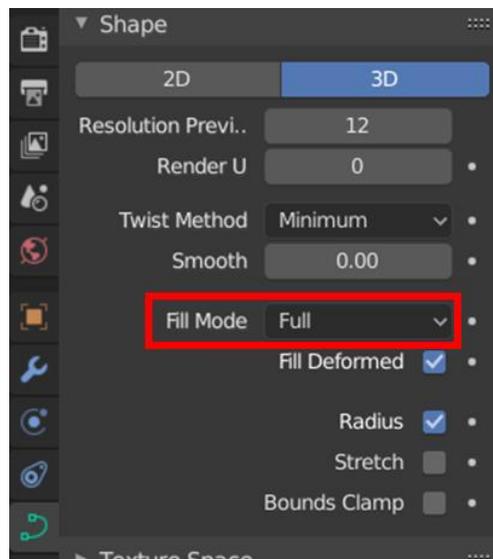


Figura 76. Cambio de Fill Mode.

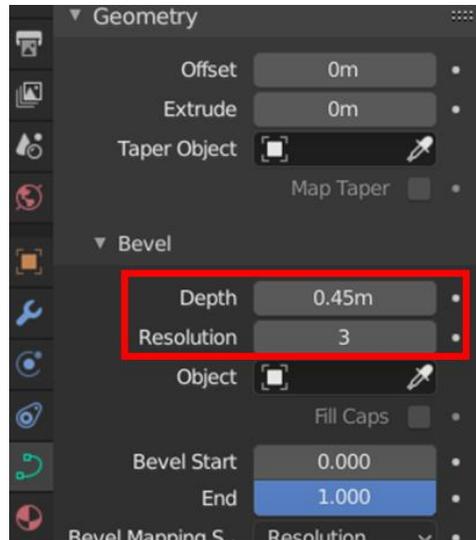


Figura 77. Ajustes de los valores de Depth y Resolution.

El resultado para esta configuración debe ser parecido al que se muestra en la **Figura 78**.

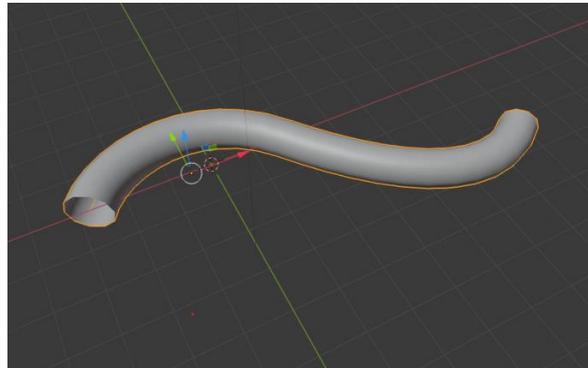


Figura 78. Resultado de cambio en la configuración de la geometría de la curva.

El último paso es convertir la curva en Mesh. Estando en el modo objeto (Layout) se selecciona la curva y con clic derecho se busca la opción **Convert to Mesh (Figura 79)**.

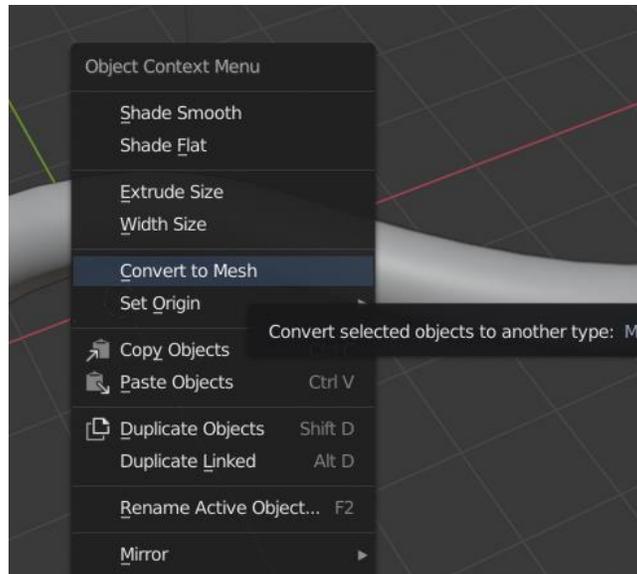


Figura 79. Paso para convertir la curva en Mesh.

Materiales y texturas

En esta sección se tratará la creación de materiales y texturas, elemento posterior a la creación de modelos en 3D con el objetivo de brindarles una mayor sensación de realismo. La etapa seguida a la creación de objetos en 3D es la asignación de colores o texturas que, teniendo el objeto seleccionado, se puede realizar de las siguientes maneras:

- a. **Creación de materiales.** Se accede desde el panel inferior derecho en el botón que se muestra en la Figura 80. Esto libera la opción para crear un nuevo material, que a su vez despliega otro menú de opciones (Figura 81). Para añadir un color, se hace clic sobre el parámetro **Base Color** (Figura 82), y se selecciona el color deseado; por el contrario, si se desea otro tipo de aspecto, este mismo parámetro ofrece otras opciones a las que se accede al hacer clic en el pequeño círculo al lado derecho del cuadrante de color (Figura 83).

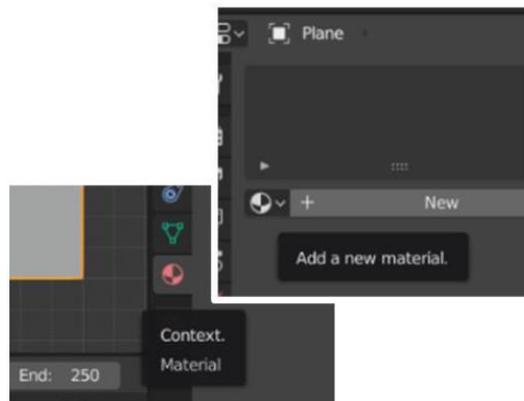


Figura 80. Menú Material.

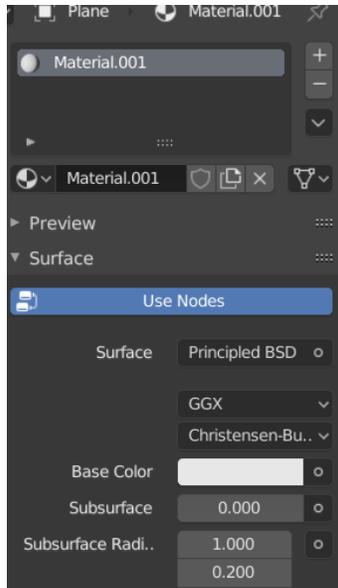


Figura 81. Menú de un material nuevo.



Figura 82. Cambio de color base.

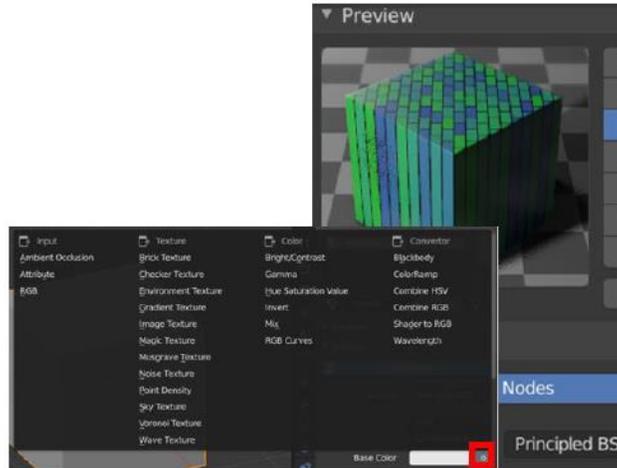


Figura 83. Opciones de color: Brick Texture.

- b. **Uso de texturas.** Dentro del menú de **Base Color** se encuentra la opción **image textura** (Figura 84). Esta opción permite la utilización de imágenes con texturas distintas a las predefinidas por Blender, por ejemplo, la textura de un dado para un cubo; no obstante, en la Figura 85, se observa que la textura no queda bien distribuida sobre el objeto. Esto se debe a que el **mapping** no se ha calculado correctamente. La distribución de la textura se puede corregir desde la pestaña **UV Editing**. Solo es necesario seleccionar el objeto y aplicar las transformaciones necesarias al **unwrap** del objeto para que coincida con los pliegues de la textura (Figura 86).

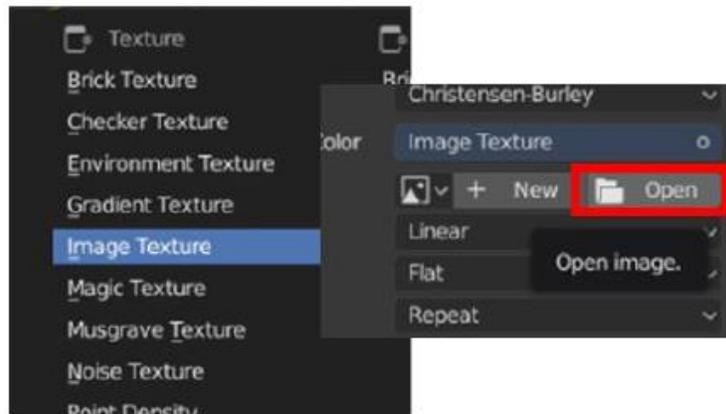


Figura 84. Menú Image Texture.

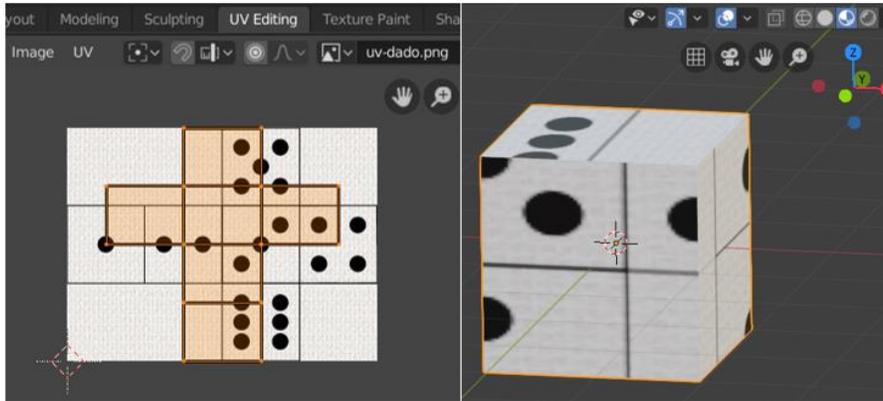


Figura 85. Ejemplo de textura insertada.

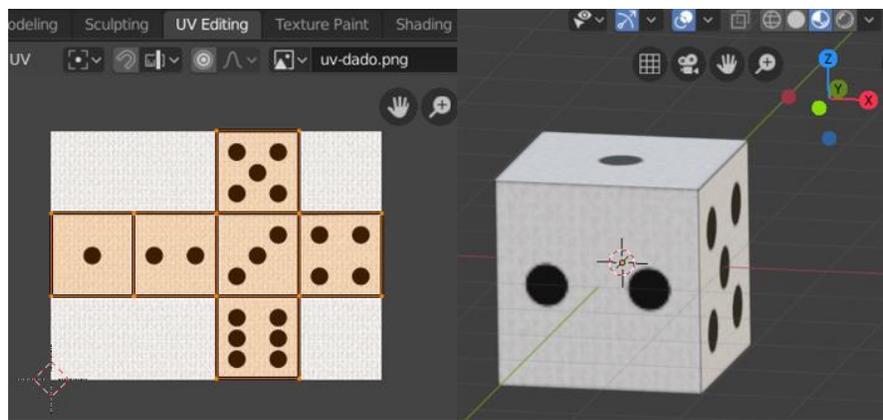


Figura 86. Corrección del despliegue de la textura insertada.

Efectos especiales aplicados a los objetos

Algunos de los efectos especiales de Blender se encuentran asociados a la opción de **Modifiers** en la sección de **Simulation** (Figura 87). Entre estos se encuentran **Cloth**, **Collision**, **Dynamic Paint**, **Explode**, **Fluid Simulation**, **Ocean**, **Particle instance**, **Particle System**, **Smoke** y **Soft Body**.



Figura 87. Modificadores de objetos.

Ejemplo 1: Simulación Cloth – Modificador Collision

Se añaden un cubo y un plano a la escena como se muestra en la Figura 88. Se selecciona el plano y se le coloca el modificador **Cloth** (Figura 89).

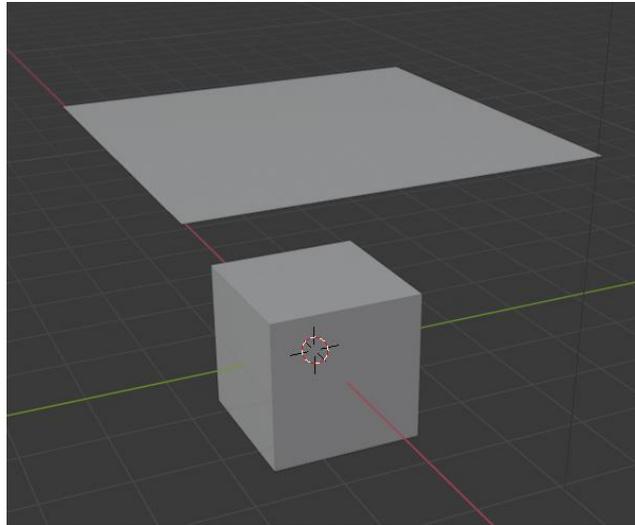


Figura 88. Colocación de los objetos en escena.

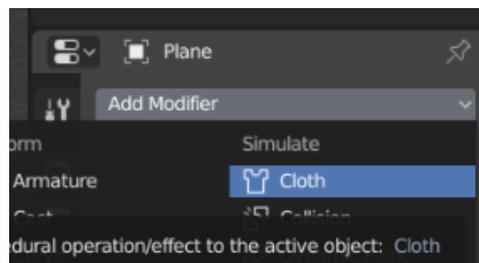


Figura 89. Adición del modificador Cloth al plano.

En la parte inferior del espacio de trabajo se aprecia una línea de tiempo en la que incluso se cuenta con botones de **play**, **pause**, entre otros. El valor inicial es el **frame 0**. Al arrastrar el cuadrante azul, el valor irá incrementando (Figura 90), y a su vez, el plano al que se le aplicó el modificador irá descendiendo sobre el cubo; sin embargo, en la Figura 91 se observa que el plano atraviesa el cubo.

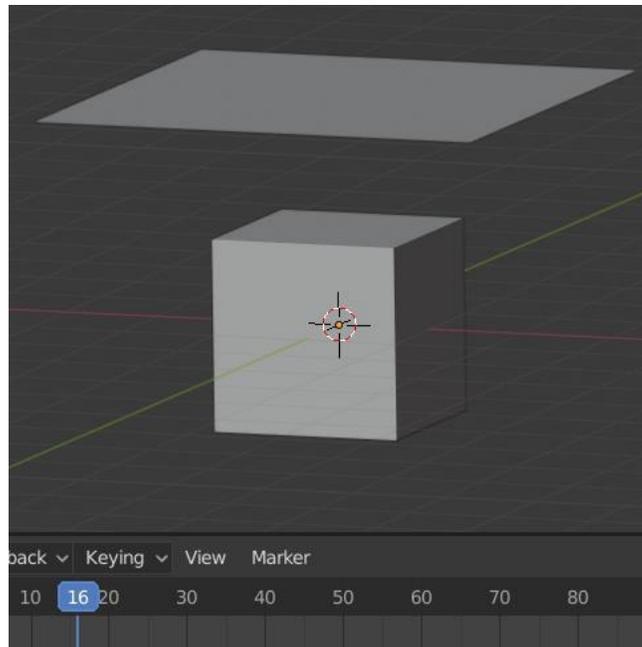


Figura 90. Cambio de frame.

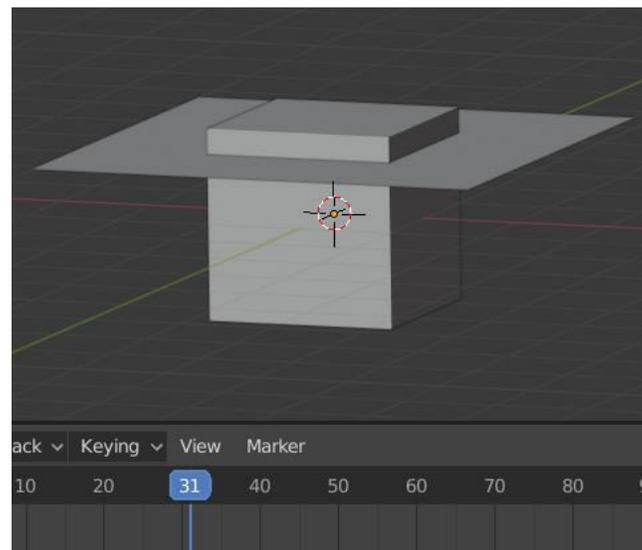


Figura 91. El plano con el modificador cloth atraviesa el cubo.

Para corregir lo anterior, se debe añadir otro tipo de modificador al cubo: **Collision**; como se muestra en la Figura 92. Esto impedirá que el plano atravesase el cubo (Figura 93). Nótese que anteriormente, en el frame 31 el plano ya había atravesado el cubo, mientras que luego de añadido el modificador al cubo, el plano se detiene sobre el cubo.

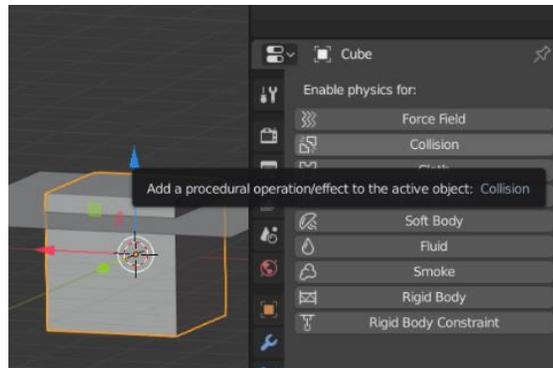


Figura 92. Modificador collision al cubo.

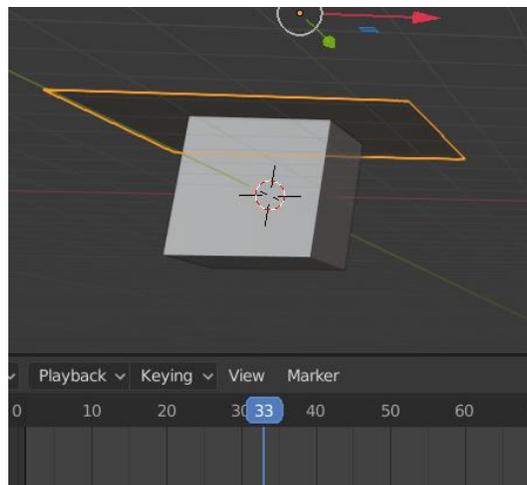


Figura 93. El plano se detiene sobre el cubo.

Otra observación importante en la Figura 94 es que el plano permanece rígido sobre el cubo, lo cual no es el comportamiento esperado en un trozo de tela. Este detalle se corrige al subdividir el plano varias veces dentro de la pestaña Modelling (Figura 95); con el plano seleccionado se hace clic derecho para desplegar las opciones entre las que aparece **Subdivide**. En la Figura 96 se observa el resultado tras la subdivisión del plano.

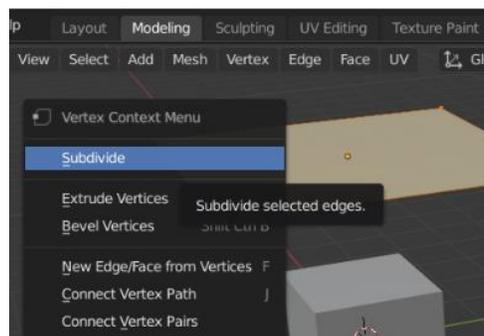


Figura 94. Subdivisión del plano para corregir a apariencia del plano sobre el cubo.

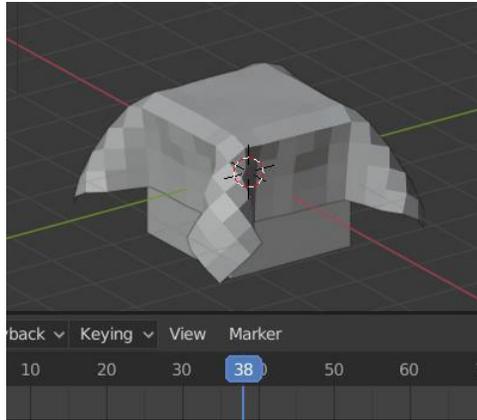


Figura 95. Resultado del modificador cloth sobre un plano con subdivisiones.

Ejemplo 2: Simulación Dynamic Paint – Animación básica

Para el siguiente ejemplo se requieren los objetos que se muestran en la Figura 96. **El plano debe estar subdividido como en el ejemplo anterior.**

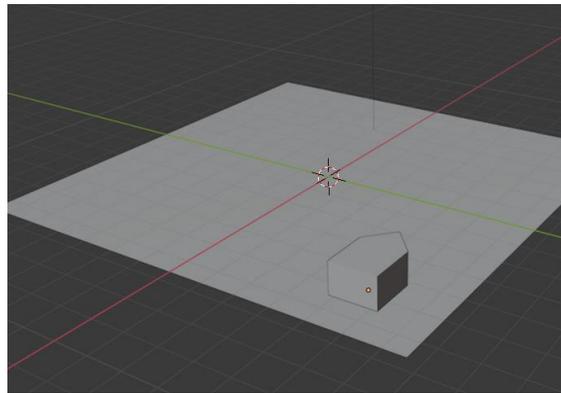


Figure 96. Plano y cubo modificado.

En esta ocasión se trabajará con el modificador **Dynamic Paint** (Figura 97). En el caso del plano, se añadirá del tipo **Canvas** con una superficie de tipo **Waves** (Figura 98).

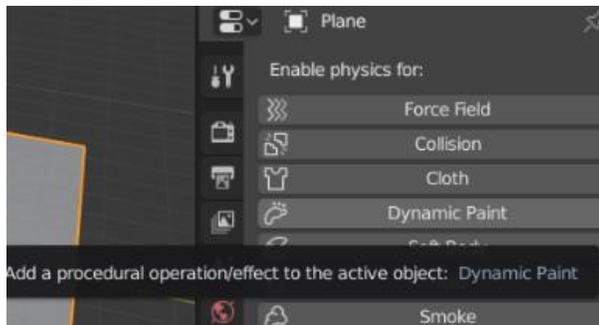


Figura 97. Añadir modificador Dynamic Paint al plano seguido por Add Canvas.

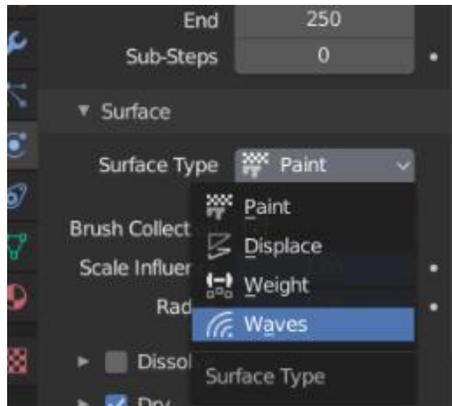


Figura 98. Cambio del tipo de superficie de la canva.

En el caso del cubo, se añadirá el mismo modificador, pero del tipo **Brush** (Figura 99). El resultado de la aplicación del modificador, al igual que con la simulación cloth, se aprecia al avanzar los frames en la línea de tiempo (Figura 100).

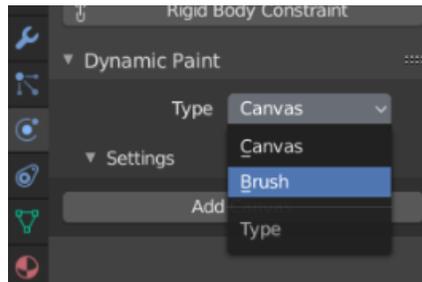


Figura 99. Dynamic paint tipo Brush.

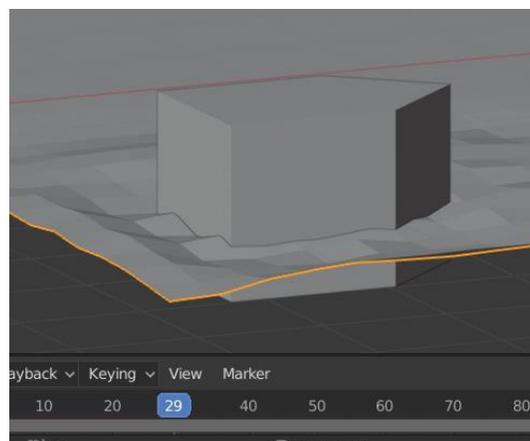


Figura 100. Resultado al aplicar el modificador Dynamic paint.

El siguiente paso es cambiar a la pestaña **Animation** con una perspectiva **Top Orthographic** como se muestra en la Figura 101.

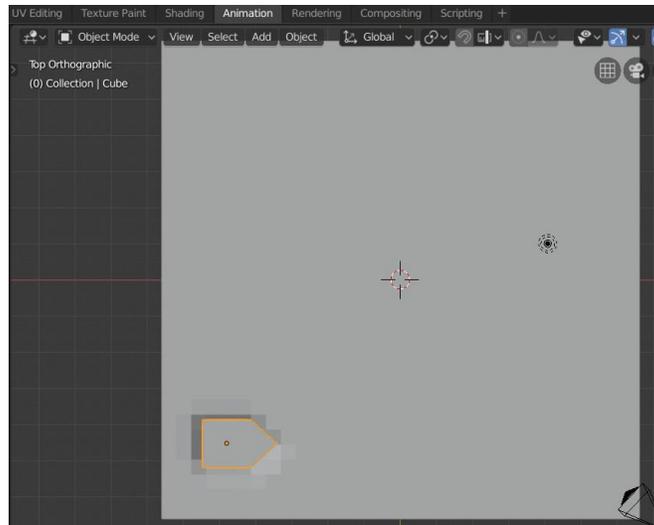


Figura 101. Elementos necesarios para la animación.

Verificando que el frame inicial sea el 0 (Figura 102), se hace clic derecho y se selecciona la opción Insert Keyframes. Se desplegará otra lista de opciones, de las que se toma LocRot (Localización y rotación), como se muestra en la Figura 103.

Nota: Si el objeto durante una animación sufre cambios en su escala, además de rotaciones y cambios de posición, lo recomendable es utilizar LocRotScale para fijar las posiciones del objeto en la línea de tiempo.

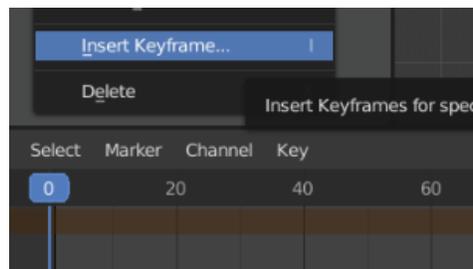


Figura 102. Ubicación en el frame 0 antes de insertar el primer keyframe.

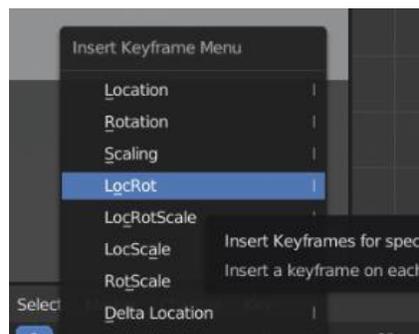


Figura 103. Selección de opción de keyframe LocRot.

El resultado de esta operación es la aparición de unos marcadores amarillos en la línea de tiempo, sobre el frame 0, como se aprecia en la Figura 104. Se repite el mismo proceso, cambiando el objeto de posición y ángulo, con el fin de obtener un recorrido. En el caso de las rotaciones, es posible indicar el grado de giro luego de ingresar la tecla **R** y se ve reflejado en el panel de la derecha.

Nota: Si el panel de transformaciones básicas no está visible, solo es necesario introducir la tecla **T**.

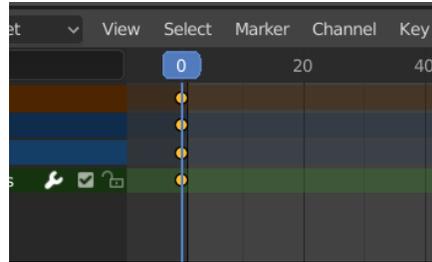


Figura 104. Posición del objeto en el frame 0.

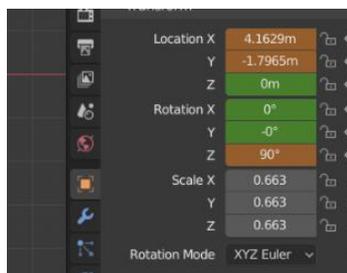


Figura 105. Panel en donde se pueden modificar de forma más precisa los ángulos de rotación y posición del objeto.

El resultado debe ser similar al que se observa en la Figura 106. Al hacer play a la animación, el objeto genera los frames entre las posiciones indicadas en la línea de tiempo.

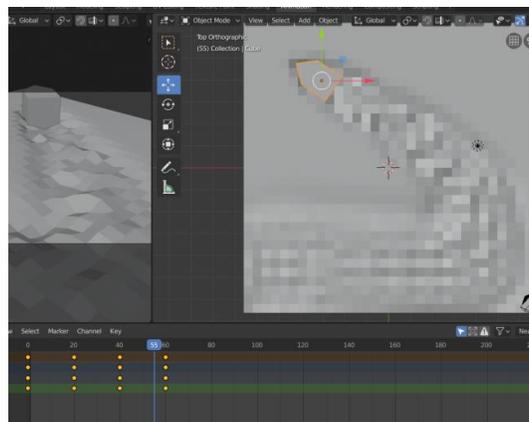


Figura 106. Recorrido del objeto.

Efectos atmosféricos

En cuanto a efectos atmosféricos, lo más comunes son escenas de día o de noche, sin embargo, estos solo requieren jugar con las fuentes de luz para lograr la sensación deseada. En esta sección se trabajará una simulación sencilla de lluvia mediante el uso de partículas.

Ejemplo: Lluvia – Partículas

Se inserta un plano en la escena y se eleva en el eje Z (Figura 107). En el panel de la derecha se selecciona el botón **Particle** (Figura 108) para añadir ese modificador al plano.

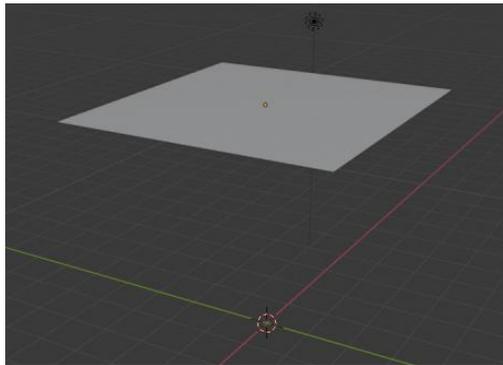


Figura 107. Plano por encima de los ejes Y y X.

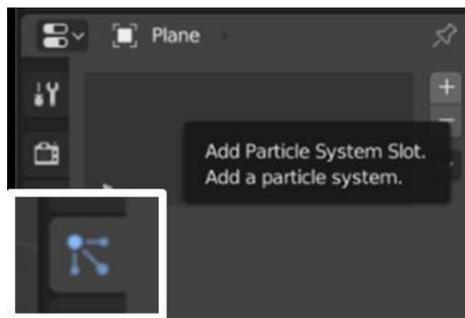


Figura 108. Botón para añadir el modificador de partículas.

Los efectos se aprecian al avanzar en la línea del tiempo, como es el caso de los modificadores vistos anteriormente. En la Figura 109 se observa como caen pequeños círculos desde el plano, lo que simula una llovizna. En la Figura 110 se nota un cambio de intensidad propiciado por un ajuste en el parámetro **Number** de la pestaña **Emission**.

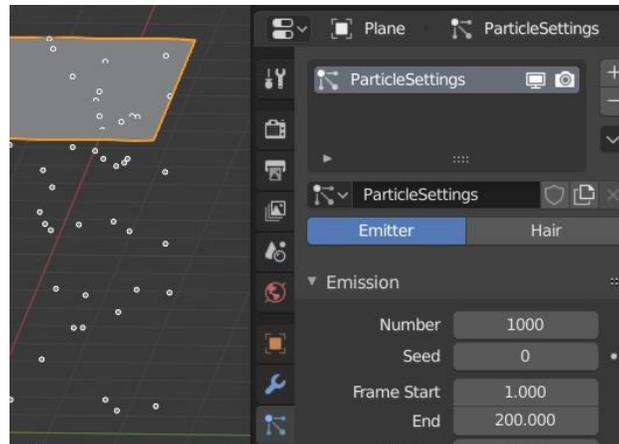


Figura 109. Acción del modificador partículas sobre el plano al avanzar en la línea de tiempo.

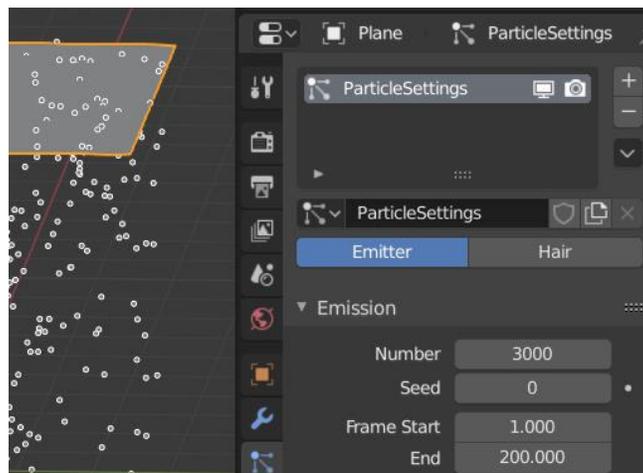


Figura 110. Modificador de la intensidad de emisión de 1000 a 3000.

La forma de las partículas puede ser enlazada a un objeto definido por el usuario. Para esto, se requiere acceder a la pestaña **Render** del menú de particle, cambiar la opción **Render as a Object** (Figura 79). Esto desplegará un menú en el que se debe seleccionar el nombre del objeto del cual las partículas heredarán su forma.

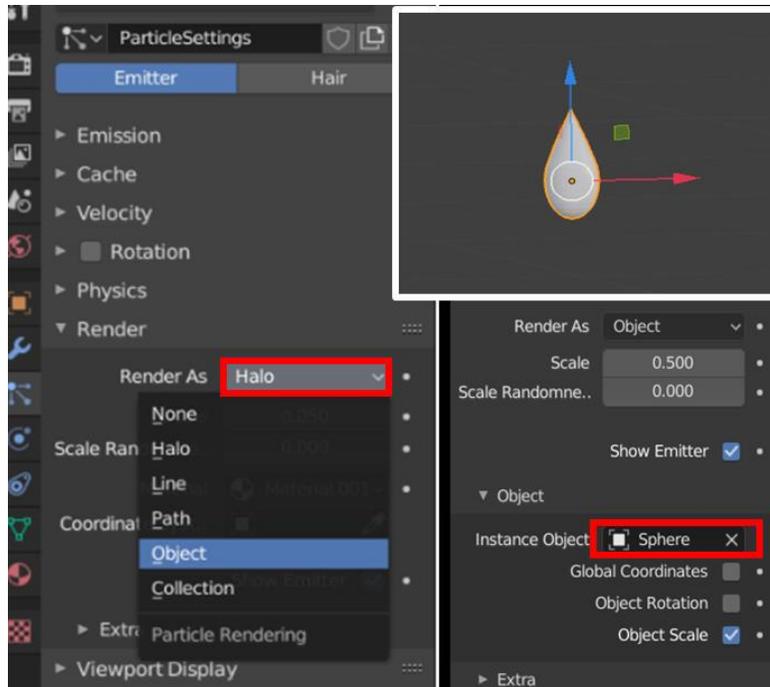


Figura 111. Personalización de la forma de las partículas.

El resultado de los pasos anteriores se observa en la Figura 112. La forma de las partículas ahora es la forma del objeto que se ha definido.

En la Figura 113 se observa que un cambio de escala en el objeto padre es heredado por las partículas en emisión. Esto facilita una personalización más detallada del efecto de lluvia.

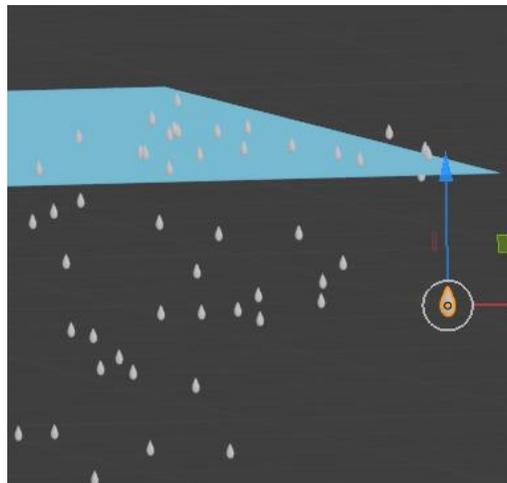


Figura 112. Emisión de partículas con la forma de una gota.

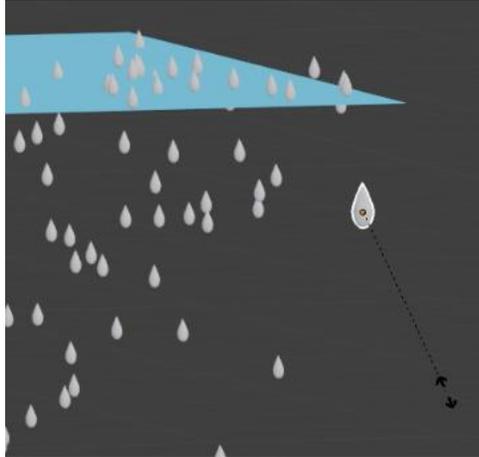


Figura 113. Al modificar el objeto, los cambios se verán reflejados en las partículas.

Ejemplo: Smoke – Force Fields

Para este ejemplo se requiere insertar un objeto en la escena que servirá como el emisor de las partículas de humo. Se selecciona el objeto, seguido por un clic derecho, se busca la opción **Quick Effects->Quick Smoke** (Figura 114). Esto hará aparecer una caja que representa el alcance de las partículas de humo como se muestra en la Figura 115.

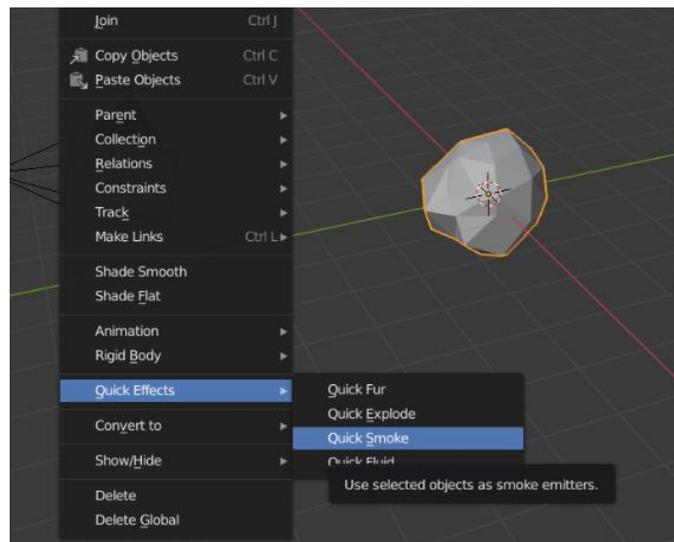


Figura 114. Quick Smoke.

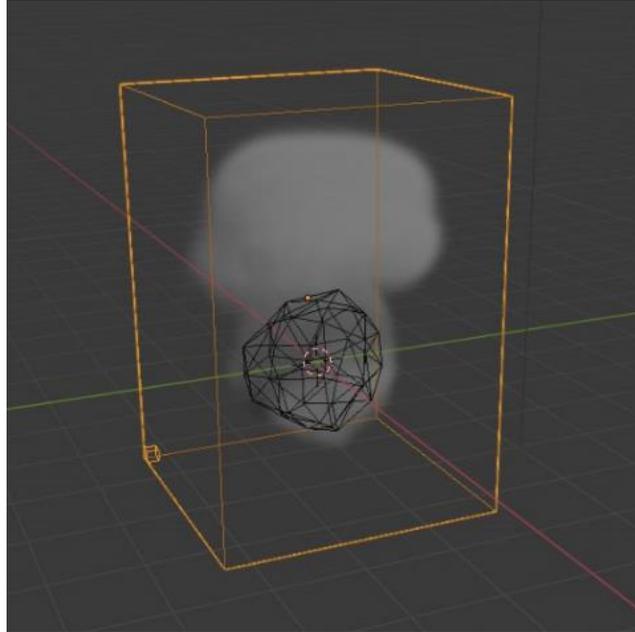


Figura 115. Caja de dominio de las partículas de humo.

Por defecto, el humo se dirige hacia arriba. Esto se puede modificar mediante el uso de **Force Fields** de tipo **wind** (Figura 116). Este objeto tiene la apariencia de tres círculos (Figura 117).

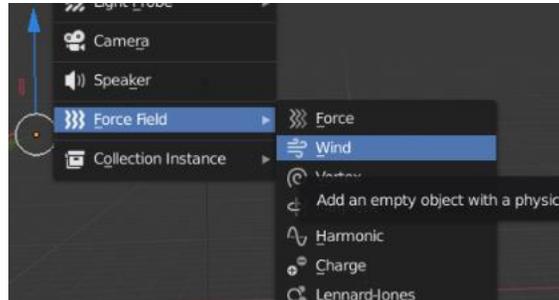


Figura 116. Añadir Force Field.

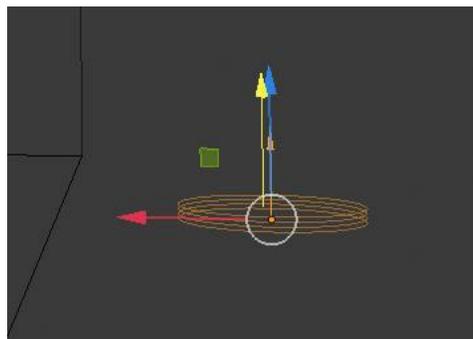


Figura 117. Force Field tipo wind.

En el panel derecho se modifica el valor de la fuerza que ejerce el objeto sobre las partículas de humo (Figura 118). En la Figura 119 se observa el cambio de dirección.

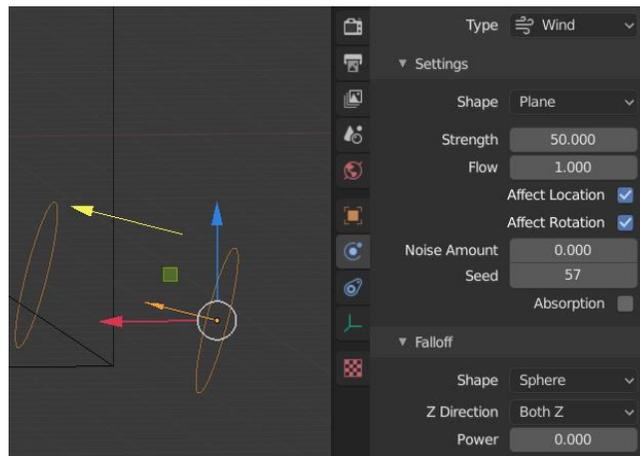


Figura 118. Modificación de la fuerza.

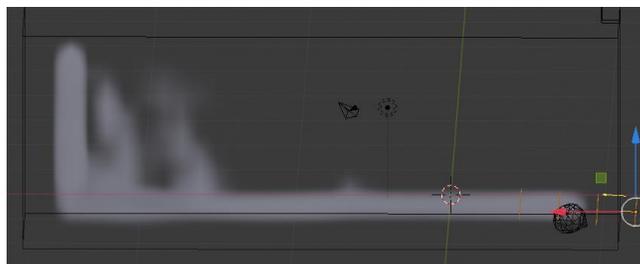


Figura 119. Cambio de dirección de las partículas de humo.

Bibliografía

- Fontanills, D. G., & Franquesa, A. F. (2011). Gráficos digitales. *Imagen y Lenguaje Audiovisual*.
- Hartfoushi, O., Obiedat, R., Al-Sayyed, R., & Al-Shboul, B. (2013). Impact of computer graphics on the engineering product design: Conceptual analysis. *Academic Journals*, 25(8), 1553-1561. doi:<https://doi.org/10.5897/SRE2013.5607>
- Hearn, D., & Baker, M. P. (2006). *Gráficos por Computadora con OpenGL*. Retrieved from <https://ingenieriayeducacion.files.wordpress.com/2013/12/graficosporcomputadorayopengl.pdf>
- Hughes, J., Van Dam, A., & McGuire, M. (2014). *Computer Graphics. Principles and Practice*.
- Martínez Riachi, S., Carreño, C., Constable, L., Tarabaín, P., & Freites, M. (2002). *Cristales Líquidos: un ejemplo fantástico de aplicación tecnológica de las propiedades de la materia* (Vol. 52).

Anexos

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Herramientas de Computación Gráfica

Asignación #1

Escoger la mejor respuesta. Encierre en un círculo la respuesta correcta.

1. El monitor es el elemento principal dentro de un sistema de computación gráfica. Desde sus inicios han empleado la configuración RGB, que significa que la imagen se envía a su monitor con un valor para cada píxel de los siguientes colores. Seleccione una:
 - a. Azul, Cian, Negro
 - b. Rojo, Gris, Blanco
 - c. Rojo, Verde, Azul

2. La tecnología de Tubos de Rayos Catódicos, CRT por sus siglas en inglés, tuvo su origen en 1897 por Karl Ferdinand Braun. Una señal RGB era generada por un cañón de electrones, formado por un cátodo y un ánodo. El ánodo emitía los electrones que el cátodo se encargaba de manipular. Seleccione una:
 - a. Verdadero
 - b. Falso

3. Pantalla Plasma se conocen también con el nombre de pantallas de descarga de gas. Seleccione una:
 - a. Verdadero
 - b. Falso

4. Las pantallas LEDs están formadas de pixeles mediante módulos o paneles de LEDs (diodos emisores de luz). Pueden ser monocromáticos, bicolor o policromáticos (RGB). Seleccione una:
 - a. Verdadero

- b. Falso**
- 5.** Los teclados de forma general se encuentran subdivididos de la siguiente forma: Alfanumérico, Numérico., Teclas de cursor y Funciones especiales. Las disposiciones de las teclas más utilizada es la...Seleccione una:
- a. Verdadero**
 - b. Falso**

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Herramientas de Computación Gráfica

Asignación #2

Un formato de archivo está diseñado específicamente para representar imágenes gráficas. Los formatos de archivo de gráficos pueden clasificarse ampliamente en dos grandes grupos que son los formatos de mapa de bits y los formatos vectoriales. Las imágenes gráficas se almacenan digitalmente usando una pequeña cantidad de formatos de archivos gráficos estandarizados, incluyendo mapa de bits, TIFF, JPEG, GIF, PNG; también se pueden almacenar como datos sin procesar y sin procesar.

Objetivo

Conocer los métodos de compresión de imágenes y los tipos de formatos de archivos para el almacenamiento de imágenes.

Indicaciones

Este trabajo busca conocer las diferencias entre los formatos y tener así criterios para discernir entre la mejor opción.

1. Explique en sus palabras los métodos de compresión de imágenes (Lossless vs Lossy) y los algoritmos más utilizados.
2. Explique en sus palabras las ventajas y desventajas de los diferentes formatos para almacenamiento de imágenes. Mencione las características más relevantes (formatos propietarios, independientes de la aplicación etc).
3. Su opinión debe llevar un orden coherente con la lectura de la unidad.
4. El contenido del documento debe estar limitado a 3000 palabras en **Microsoft Word**.
5. Agregue imágenes relacionadas a los atributos señalados.
6. El documento debe contener: Hoja de presentación, introducción, contenido, conclusión y referencias en formato APA que deben ser citadas dentro del documento.

Observación

- La asignación tiene una evaluación sumativa de 10 puntos.

Criterios de Evaluación

Se evaluará lo siguiente:

Criterios	Valoración					
	0	2	4	6	8	10
1. Estructura del documento						
2. Calidad del contenido						
3. Uso de citas en el documento						
4. Calidad de las imágenes						
5. Respeto a la ortografía						

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Herramientas de Computación Gráfica

Asignación #3

En los gráficos por computador, los puntos y segmentos de línea recta son los componentes geométricos más elementales. El analizador diferencial digital (DDA) permite construir líneas entre dos puntos y mostrarlos como una línea recta o polígono en la pantalla. La distancia entre dos puntos se describe mediante una ecuación diferencial donde las coordenadas del punto inicial y la del punto final se especifican en el software. El DDA se utiliza para la interpolación de variables en un intervalo entre el punto inicial y el punto final. El DDA se usa para rasterizar líneas, triángulos y polígonos, sin embargo, tienen muchas otras aplicaciones.

Objetivo

Implementar en lenguaje C++ el analizador diferencial digital (DDA; digital differential analyzer).

Indicaciones

En este trabajo usted debe:

1. Implementar a través del lenguaje de programación C++ el analizador diferencial digital (DDA; digital differential analyzer).
2. Elaborar un pequeño video de máximo cinco minutos explicando su implementación.
 - a. Debe mostrar su rostro en el video a fin de validar su trabajo
 - b. Debe habilitar comentarios para dejarle mi apreciación del trabajo
3. Subir el video en Youtube.
4. Enviar el enlace a través de la plataforma.

Observación

- La asignación tiene una evaluación sumativa de 10 puntos.

Criterios de Evaluación

Se evaluará lo siguiente:

Criterios	Valoración					
	0	2	4	6	8	10
1. Duración						
2. Calidad del contenido						
3. Originalidad y creatividad						
4. Vocabulario y Audio						
5. Estructura del código C++						

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Herramientas de Computación Gráfica

Asignación #4

El relleno de figuras recurre a la utilización de métodos recursivos como el algoritmo de relleno por difusión, también llamado floodfill (en inglés). El algoritmo de relleno por difusión o inundación determina el área formada por elementos contiguos en una matriz multidimensional. El algoritmo ayuda a visitar todos y cada uno de los puntos de un área determinada, conectada a una celda determinada en una matriz multidimensional. En este método, se selecciona un punto o semilla que está dentro de la región. Luego, se utilizan cuatro enfoques conectados u ocho enfoques conectados para rellenar.

Objetivo

Implementar lenguaje C++ el algoritmo floodfill.

Indicaciones

En este trabajo usted debe:

1. Implementar a través del lenguaje de programación C++ el algoritmo floodfill según el ejemplo de la clase (**Evite recurrir a librerías preprogramadas**).
2. Elaborar un pequeño video de máximo cinco minutos explicando su implementación.
 - a. Debe mostrar su rostro en el video a fin de validar su trabajo
 - b. Debe habilitar comentarios para dejarle mi apreciación del trabajo.
3. Subir el video en Youtube.
4. Enviar el enlace a través de la plataforma.

Observación

- La asignación tiene una evaluación sumativa de 10 puntos.

Criterios de Evaluación

Se evaluará lo siguiente:

Criterios	Valoración					
	0	2	4	6	8	10
1. Duración						
2. Calidad del contenido						
3. Originalidad y creatividad						
4. Vocabulario y Audio						
5. Estructura del código C++						

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Herramientas de Computación Gráfica

Asignación #5

La simetría preserva invariantes las medidas de los ángulos de una forma geométrica a pesar de las transformaciones. Entregar un documento con las formalidades de un documento escrito. Hoja de presentación, Introducción, Contenido, Conclusiones y Bibliografía.

Objetivo

Elaborar un ejemplo paso a paso considerando el material sobre escalado, simetría y recorte.

Indicaciones

1. Entregar un documento con las formalidades de un documento escrito. Hoja de presentación, Introducción, Conclusiones y Bibliografía en formato APA.
2. Elabore tres ejemplos considerando el material de la clase 2.3.3 sobre escalado y la clase 2.3.4 sobre simetría y recorte.

Observación

- La asignación tiene una evaluación sumativa de 10 puntos.

Criterios de Evaluación

Se evaluará lo siguiente:

Criterios	Valoración					
	0	2	4	6	8	10
1. Formalidades del documento						
2. Ejemplo de escalado						
3. Ejemplo de simetría						
4. Ejemplo de recorte						
5. Creatividad del trabajo						

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Herramientas de Computación Gráfica

Asignación #6

El rostro humano varía para cada persona y esto lo establece las proporciones que toman de referencia partes de la anatomía como boca, nariz, ojos, etc. La antropometría es la ciencia que estudia las medidas humanas para el diseño de los objetos utilizados por el hombre. Existe la muy conocida proporción dorada o áurea para establecer un ideal en las medidas del cuerpo humano.

Objetivo

Conocer principios utilizados para medir el rostro humano.

Indicaciones

1. Explique con sus palabras que es la proporción áurea.
2. Explique la relación entre la secuencia de números de Fibonacci.
3. Obtenga un análisis de su rostro con relación a la proporción áurea.
4. Tome una foto de frente tipo selfie para responder que tanto se acerca su imagen a la medida áurea realizando los siguientes pasos.
 - a. Haga seis copias de su imagen tipo selfie.
 - b. Imprima las imágenes.
 - c. Tome de referencia la presentación (d) del enlace para tomar las medidas indicadas en las líneas de color azul.
 - d. <https://www.slideshare.net/magvil/anlisis-armnico-de-las-proporciones-del-rostro>
 - e. Compare de dos en dos las medidas indicadas en azul dividiendo la de mayor longitud entre la de menor longitud
 - f. Escriba el número que obtuvo para cada par de medidas al lado de la imagen.

- g. Coloque las medidas en el documento a enviar y verifique que tanto se acerca a la medida áurea adjunto las imágenes que tomó.
5. Entregar un documento con las formalidades de un documento escrito. Hoja de presentación, Introducción, Contenido y Bibliografía en formato APA.

Observación

- La asignación tiene una evaluación sumativa de 10 puntos.

Criterios de Evaluación

Se evaluará lo siguiente:

Criterios	Valoración					
	0	2	4	6	8	10
1. Formalidades del documento						
2. Contenido de las respuestas						
3. Calidad de las imágenes						
4. Exactitud de los cálculos						
5. Creatividad del trabajo						

Universidad Tecnológica de Panamá
Facultad de Ingeniería de Sistemas Computacionales
Herramientas de Computación Gráfica
Asignación #7

Objetivo

Crear una animación en Synfig.

Indicaciones

Cree una animación de 30 segundos utilizando Synfig y envíe el archivo fuente para abrir y ejecutar la animación. Introduzca elementos genuinos para que su trabajo sea diferente al de los demás. Es preciso mencionar que Synfig es un software gratuito y de código abierto con licencia GNU GPL v3. Las fuentes están disponibles en GitHub. Enlace: <https://www.synfig.org/>

Universidad Tecnológica de Panamá
Facultad de Ingeniería de Sistemas Computacionales
Herramientas de Computación Gráfica

Asignación #8

Blender es un programa multiplataforma, dedicado especialmente al modelado, iluminación, renderizado, animación, creación de gráficos, edición de vídeo y pintura digital. El programa es software libre y es compatible con todas las versiones de Windows, macOS, GNU/Linux, Android entre otras.

Objetivo

Conocer el uso del software Blender.

Objetivos específicos

- Modelar objetos en 3D mediante el uso de los comandos básicos de Blender
- Aplicar las transformaciones básicas de rotación y traslación a un objeto en Blender.
- Visualizar las transformaciones mediante un clip animado.

Indicaciones

1. Diseñe una escena que contenga los siguientes objetos:
 - Una plataforma con tres niveles como la que se muestra en la figura. Esta debe contar con una textura de su preferencia.



- Un balón. Debe tener una textura.

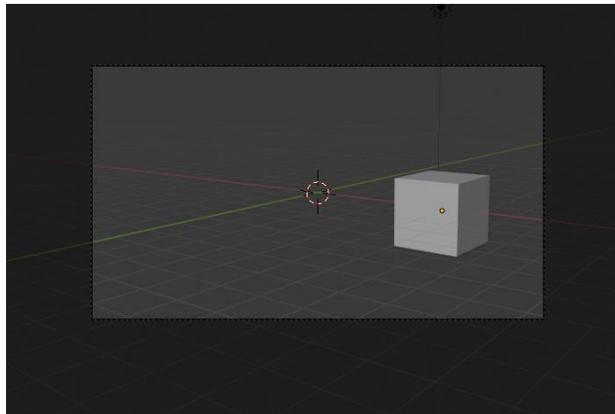
Para cada objeto debe documentar los pasos realizados: capturas de pantalla en el que se describa los comandos y herramientas utilizadas en la creación de ambos objetos.

2. Animación

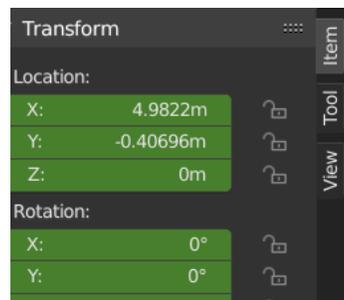
El balón debe rotar sobre el nivel superior y rebotar sobre los otros dos niveles. Para esto recuerde que debe fijar las posiciones en la línea de tiempo con la opción **Insert keyframe->LocRot**.

Proceso de renderizado:

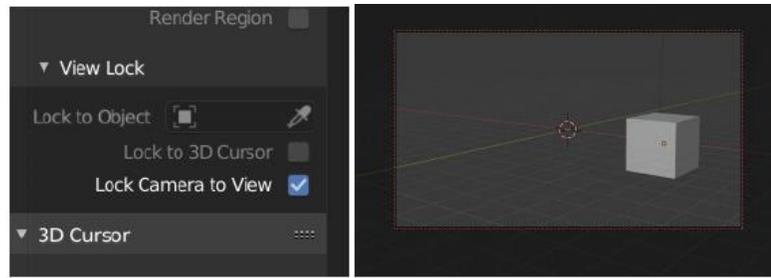
- Vista de cámara: Para efectos del renderizado de la animación, es necesario ajustar la vista de la cámara; para esto, dentro de la pestaña Animation, se ingresa la combinación Ctrl+Alt+Numpad0.



Dentro de este modo, se ingresa la tecla N, que libera el panel de transformaciones básicas, y en el cual se ubica la pestaña View.

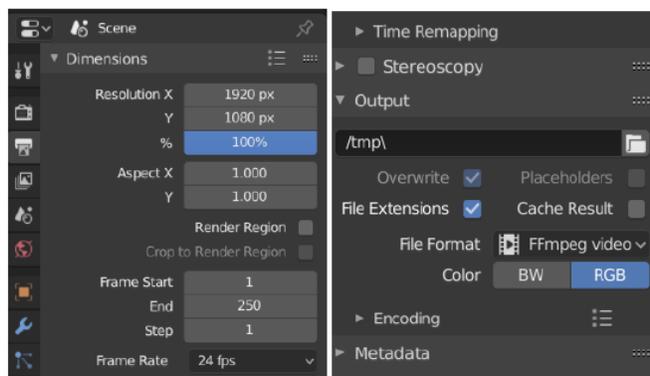


Dentro de esa pestaña, se marca la opción Lock Camera to View. El contorno del cuadrante de la cámara se tornará rojo. De esta forma es más sencillo ajustar el área que cubrirá la cámara.



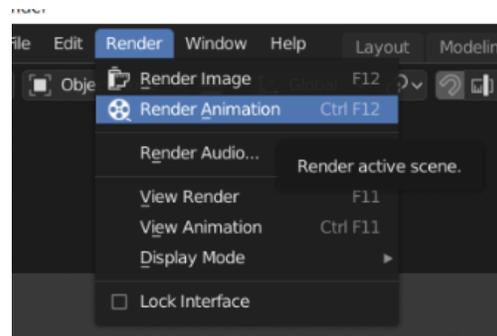
- **Renderizar**

En el panel derecho se encuentra el botón Scene. Dentro de él las opciones de los frame per second que tendrá la animación; así como la definición del frame inicial y el final de la animación. Estos dos últimos parámetros deben ser ajustados de acuerdo con la extensión de la animación trabajada. Con un frame rate de 24 fps, cada segundo de la animación tendrá 24 fps; es decir que un minuto tendrá $60 \times 24 = 1440$ frames.



En la opción de output, se escoge el directorio en donde se guardará el resultado del render. El formato de archivo para el render será **FFmpeg video**.

Finalmente, para renderizar la escena, en la barra de menú se accede al botón **Render->Render Animation**.



Observación

- Puede existir ligeras diferencias entre las pantallas brindadas por mí, debido al cambio de la versión de Blender.
- La asignación tiene una evaluación sumativa de 10 puntos.

Criterios de Evaluación

Se evaluará lo siguiente:

Criterios	Valoración					
	0	2	4	6	8	10
1. Creatividad						
2. Modelado						
3. Rotación						
4. Traslación						
5. Animación						

ANEXO 1: PRUEBAS RÁPIDAS

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Sistemas Basados en el Conocimiento

Prueba #1

Nombre: _____ Cédula: _____ Grupo: _____

Profesor: Dr. Carlos A. Rovetto

Puntos Obtenidos: _____ /

I Parte. Cierto (C) y falso (F).

1. ____ El elemento principal dentro de un sistema de computación gráfica es el mouse.
2. ____ La pantallas LED recurren a matrices de diodos que dan origen a los pixeles.
3. ____ El modo de color utilizado por los monitores desde sus inicios es el RGB.
4. ____ El primer sistema de computación gráfica que manejaba video se llamaba Whirlwind.
5. ____ La tecnología de tubos de rayos catódicos es la más reciente en el desarrollo de los monitores.
6. ____ Internamente, las pantallas LCD constan de tres capas.
7. ____ El principio funcional de las pantallas plasma está basado en convertir energía eléctrica en luz.
8. ____ En la década de 1960, se presentó el software Syntha Vision como el primer software para el modelado de sólidos.
9. ____ La computación gráfica es una ciencia que busca comunicar de forma visual el arte asistido por computadora.
10. ____ En la década de 1990, se liberó la primera versión de Adobe Photoshop.

II Parte. Desarrollo.

1. Mencione las capas que componen la estructura interna de una pantalla LCD.

Universidad Tecnológica de Panamá
Facultad de Ingeniería de Sistemas Computacionales
Sistemas Basados en el Conocimiento

Prueba #2

Nombre: _____ Cédula: _____ Grupo: _____

Profesor: Dr. Carlos A. Rovetto

Puntos Obtenidos: _____ /

I Parte. Cierto (C) y falso (F).

1. ____ Los teclados se subdividen solamente en alfanumérico y teclas de cursor.
2. ____ Uno de los dispositivos de entrada más utilizados en sistemas de computación gráfica es el mouse.
3. ____ Los dos tipos de imágenes con los que se puede trabajar son ráster y vectorial.
4. ____ Una imagen en formato JPEG perderá calidad cada vez que se le hagan y guarden cambios.
5. ____ El formato BMP admite fondos transparentes.
6. ____ La primera aplicación de los joysticks fue en la aviación.
7. ____ Digitalización es el proceso de convertir información análoga a lenguaje binario.
8. ____ Al igual que los monitores, las impresoras usan el modo de color RGB.
9. ____ La principal función de los joysticks es ser un señalador de posición en una pantalla.
10. ____ Los primeros registros de una impresora mecánica se remontan a los años 1900.

II Parte. Desarrollo.

1. Mencione tres variantes de la disposición de teclas en un teclado.
2. Mencione dos tipos de impresoras.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Sistemas Basados en el Conocimiento

Prueba #3

Nombre: _____ Cédula: _____ Grupo: _____

Profesor: Dr. Carlos A. Rovetto

Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Mencione cuatro disciplinas que actúan en conjunto con la computación gráfica:

_____, _____,
_____ y _____.

2. Tipo de imagen cuya principal característica son los píxeles:

_____.

3. Tipo de imagen que se compone de curvas obtenidas mediante cálculos matemáticos:

_____.

4. Mencione tres tipos de formatos comunes en el entorno gráfico:

_____, _____ y
_____.

5. Los joysticks se clasifican en dos grandes grupos que son:

_____ y _____.

II Parte. Desarrollo.

1. Describa las dos direcciones de comunicación que se distinguen en la computación gráfica.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Sistemas Basados en el Conocimiento

Prueba #4

Nombre: _____ Cédula: _____ Grupo: _____

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Condición particular en la que la porción de un lado de la línea de un gráfico es un reflejo del otro lado: _____.

2. Mencione tres métodos de transformación geométrica:
_____, _____ y _____.

3. Figura que tiene una circunferencia alargada y simétrica con respecto a dos ejes, posee dos radios: _____.

4. El algoritmo _____ es un método recursivo que trabaja utilizando una pila que almacena las posiciones que conforman la zona a colorear.

5. La _____ es la simetría que se da con respecto a un eje.

6. Durante el recorte de un gráfico se pueden dar tres casos:

- a. _____: La figura completa queda dentro de la ventana de recorte.
- b. _____: Una sección de la figura queda dentro de la ventana de recorte.
- c. _____: Ninguna sección de la figura queda dentro de la ventana de recorte.

II Parte. Desarrollo.

1. ¿Qué son los algoritmos de discretización de líneas?

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Sistemas Basados en el Conocimiento

Prueba #5

Nombre: _____ Cédula: _____ Grupo: _____

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Tres métodos de transformación comunes son:

- a. _____: Es posible que se presenten deformaciones en la figura. Escalado
- b. _____: Consiste en mover un objeto de una posición a otra siguiendo una trayectoria en línea recta. Traslación
- c. _____: Consiste en mover un objeto sobre un punto de pivote dado un ángulo θ

2. El uso de un sistema de coordenadas homogéneas facilita la aplicación de _____ a una figura.

3. Nombre de la estructura de datos que utiliza el algoritmo FloodFill: _____.

4. A los píxeles que componen una figura también se les conoce como: _____.

5. En los algoritmos de discretización de líneas, las posiciones aproximadas de los píxeles por los que pasará la línea, se calculan con la siguiente ecuación: _____.

6. Nombre del algoritmo que se utiliza para otorgar un color dentro de los píxeles que componen una figura: _____.

II Parte. Desarrollo.

1. ¿Qué es una ventana de recorte?

2. ¿Por qué el principio de simetría es de gran utilidad en la computación gráfica?

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Sistemas Basados en el Conocimiento

Prueba #6

Nombre: _____ Cédula: _____ Grupo: _____

Profesor: Dr. Carlos A. Rovetto

Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Tres combinaciones que pueden derivar de un círculo cromático son:
_____, _____ y
_____.

2. La teoría del color toma en cuenta los siguientes atributos:
_____, _____ y
_____.

3. Representación simplificada de un dibujo realista:
_____.

4. Una secuencia de dibujos en 2D dan como resultado una:
_____.

5. Es una técnica cuyo uso es común en fotografía:
_____.

6. Paleta de colores en la que se incluyen los colores primarios y los colores secundarios que derivan de ellos: _____.

II Parte. Desarrollo.

1. Mencione cinco principios de animación.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Sistemas Basados en el Conocimiento

Prueba #7

Nombre: _____ Cédula: _____ Grupo: _____

Profesor: Dr. Carlos A. Rovetto Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Indique el nombre de los siguientes principios de animación: 5

- a. _____: Aumento y disminución del tamaño de los cuerpos.
- b. _____: Número de eventos que se utilizan para una acción en un tiempo dado.
- c. _____: Garantiza que el objeto sea coherente con sus movimientos.
- d. _____: Incrementa el efecto de una acción.
- e. _____: Permite anticipar la acción a realizar.
- f. _____: Se mezclan movimientos para resaltar la acción.
- g. _____: Resalta la presencia en escena del objeto y resalta su forma.
- h. _____: Resaltar las perspectivas de los objetos para dar volumen y peso.

2. Combinación de colores que se encuentran opuestos en el círculo cromático:

_____.

3. Combinación de colores que resulta de tres colores equidistantes:

_____.

II Parte. Desarrollo.

1. Describa en qué consiste la regla de tercios.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Sistemas Basados en el Conocimiento

Prueba #8

Nombre: _____ Cédula: _____ Grupo: _____

Profesor: Dr. Carlos A. Rovetto

Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Mencione cuatro herramientas para el modelado y desarrollo de animaciones en 3D:

_____, _____,
_____ y _____.

2. Nombre de la sección en Blender que ofrece acceso rápido para el renderizado de imágenes o videos: _____.

3. Blender ofrece dos opciones para las vistas: _____ y _____.

4. Mencione cinco modos o secciones que ofrece la ventana Layout de Blender:

_____, _____,
_____, _____ y
_____.

II Parte. Desarrollo.

1. Mencione tres funcionalidades que Blender permite realizar al crear gráficos 3D.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Sistemas Basados en el Conocimiento

Prueba #9

Nombre: _____ Cédula: _____ Grupo: _____

Profesor: Dr. Carlos A. Rovetto

Puntos Obtenidos: _____ /

I Parte. Llene los espacios.

1. Blender ofrece una paleta de objetos, como: _____ y _____, a los que se les puede aplicar: _____ y _____.

2. Mencione las cuatro opciones de iluminación que Blender ofrece para aplicar efectos a escenas: _____, _____ y _____.

3. Para renderizar en Blender, existen tres tipos de cámaras o lentes que son: _____ y _____.

II Parte. Desarrollo.

1. Mencione cuatro tipos de efectos especiales que Blender permite aplicar a objetos.

ANEXO 2: PRESENTACIONES



1



2

Capítulo I: Generalidades de la computación gráfica

Generalidades y evolución histórica

Definición y conceptos

La computación gráfica es una ciencia enfocada en la comunicación visual del arte asistida por una computadora, en lo que también juegan un rol de importancia los dispositivos que permiten interactuar con dicho arte. El arte puede ser la representación de la realidad o de objetos abstractos.

- **Física:** Se utiliza para hacer modelos realistas en los que se involucran los espectros de luz y movimiento dentro de las animaciones.
- **Matemáticas:** Se emplean para describir las formas y objetos que interactúan en la animación.
- **Ingeniería:** Procura optimizar los procesos de asignación del ancho de banda, memoria y procesador.
- **Arte, diseño gráfico y la interacción humano – computador:** Persiguen lograr una comunicación efectiva con el usuario.

3

Principales hitos en la evolución histórica de la computación gráfica

En la siguiente tabla se resumen algunos de los acontecimientos que marcaron etapas en la evolución de la computación gráfica.

Décadas	Acontecimientos
1950	<ul style="list-style-type: none"> • Primer monitor basado en tecnología Cathode Ray Tube (CRT). • Ben Laposky diseñó el primer equipo que mostró imágenes gráficas. Consistía en un osciloscopio que trabajaba con vigas electrónicas y registraba todo en una película. • El primer sistema de computación gráfica que manejaba video, Whirlwind, fue desarrollado por Jay Forrester y Robert Everett, en el Massachusetts Institute of Technology (MIT). • Derivado de Whirlwind, surge SAGE (Semi-Automatic Ground Equipment). Programa que mediante gráficos vectoriales simples, mostraba imágenes de radar, convirtiéndolo en pieza clave del sistema estadounidense de defensa contra misiles. • Surge el primer sistema de software comercial de manufactura asistida por computadora (CAM). Diseñado por el Dr. Patrick J. Hanratty, padre de los gráficos asistidos por computadora. • Su software era una herramienta de programación de control numérico. • General Motors y IBM desarrollaron el Diseño Aumentado por Computadora -1, con el fin de ayudar a los ingenieros con el diseño de los automóviles.

4

Principales hitos en la evolución histórica de la computación gráfica

Décadas	Acontecimientos
1960	<ul style="list-style-type: none"> Ivan Sutherland presenta como parte de su disertación de Ph. D. el proyecto "Sketchpad". Utilizó una computadora TX-2. Sketchpad es reconocido como el primer software CAD. Synthia Vision presenta el primer software para el modelado de sólidos. La NASA desarrolló un procesador de imágenes llamado VICAR (Video Image Communication and Retrieval) sobre el mainframe de IBM. Con este programa se procesaron las imágenes capturadas del espacio. Se implementaron los primeros simuladores de vuelo. Tektronix, primer computador CRT. Se funda SIGGRAPH (Association for Computing Machinery's Special Interest Group on Computer Graphics and Interactive Techniques).
1970	<ul style="list-style-type: none"> Pierre Bezier da a conocer las curvas de Bezier, que más adelante constituirían una de las principales herramientas en el dibujo vectorial. Se introduce la tecnología ráster (matrices de celdas conocidas como píxeles). Surgimiento de la técnica de mapeo de imágenes y textura por Blinn. Quantel desarrolló Paintbox, herramienta de gran impacto entre los productores de programas de televisión, debido a que les permitía editar y manipular digitalmente lo grabado.
1980	<ul style="list-style-type: none"> Primer computador Apple Macintosh con gráficos a color. TRON – Película de Disney elogiada por sus gráficos tridimensionales.
1990	<ul style="list-style-type: none"> Se introducen los monitores VGA y SVGA. Marcus Andreessen desarrolló el primer buscador web, Mosaic, con texto e imágenes. Se libera la primera versión de Adobe Photoshop y PaintShop. Se sientan las especificaciones OpenGL. El World Wide Web Consortium (W3C) introduce el formato SVG (scalable vector graphics) para la utilización de imágenes de mayor tamaño en los sitios web. Aparecen los primeros dispositivos móviles con pantallas táctiles.

5

Funcionamiento de dispositivos para el manejo y despliegue gráfico

Monitores

El monitor es el elemento principal dentro de un sistema de computación gráfica, dado que es el responsable de mostrar los gráficos resultantes al usuario. Todos los monitores, desde sus inicios, han empleado el modo de color rojo, verde y azul, RGB, por sus siglas en inglés.

• Tubos de rayos catódicos (CRT)

La tecnología de Tubos de Rayos Catódicos, CRT por sus siglas en inglés, tuvo su origen en 1897 por Karl Ferdinand Braun

• Pantalla de cristal líquido (LCD)

El cristal líquido fue descubierto en un principio por Friederich Reinitzer en el año 1888, posterior a él investigadores como Richard Williams y George Heilmeyer realizaron nuevos descubrimientos relacionados al efecto electro-óptico que posee el material al ser sometido a un voltaje.

• Pantalla de diodo emisor de luz (LED)

Este tipo de monitores utilizan una tecnología desarrollada en 1962 por Nick Holonyak. Holonyak desarrolló el primer LED que emite luz roja.

Otras tecnologías previas a las mencionadas son las siguientes:

- Monochrome Display Adapter (MDA).
- Color graphics Adapter (CGA).
- Enhanced Graphics Adapter (EGA).
- Video Graphics Array (VGA).
- Super Video Graphics Array (SVGA).



6

Impresoras



Los primeros registros de una impresora mecánica se remontan a los años 1800, en donde también se crea la primera máquina de escribir, siendo considerada la precursora de los teclados e impresoras de la actualidad. Teniendo estas bases con el pasar del tiempo se diseñaron e implementaron distintos modelos y técnicas de impresión, como las que se listan a continuación:

- Impresora de impacto.
- Impresora de matriz de puntos.
- Impresora de chorro de tinta.
- Impresora láser.

• SLA

El proceso conocido como estereolitografía o SLA por sus siglas en inglés, fue el primero en estar detrás de la posibilidad de imprimir un objeto a partir de datos en formato digital. Los objetos se imprimen por capas, eran bañados por un solvente y endurecidos con luces ultravioletas.

• SLS

El sinterizado selectivo por láser o SLS, como su nombre indica, está basado en el uso de polímeros en polvo, que son fundidos por un láser y colocados por capas para crear formas más complejas que con el SLA.

• FDM

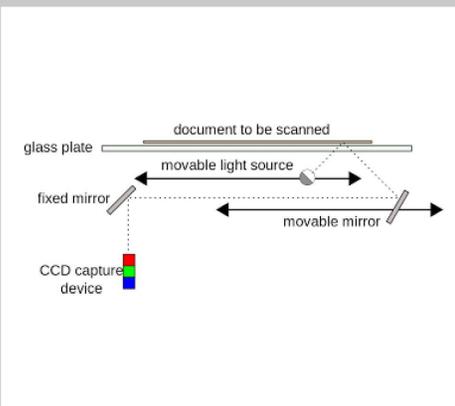
El modelado por disposición fundida o FDM es la técnica de impresión que predomina en estos días. Este proceso calienta un cable de material termoplástico hasta convertirlo en líquido, lo que permite extraerlo y colocarlo en capas.

7

Escáneres

Un escáner es comúnmente utilizado para digitalizar documentos impresos, obteniendo así imágenes o texto, que se puede mostrar y almacenar en una computadora. El término digitalizar hace referencia a la transcripción de información analógica a lenguaje binario, es decir, ceros y unos.

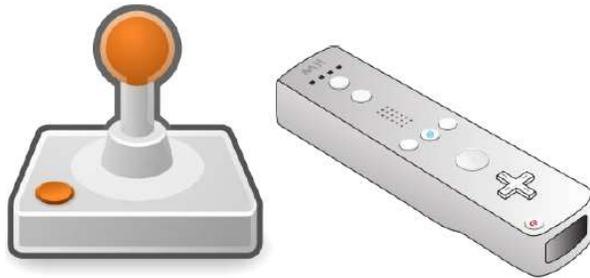
Un escáner posee una fuente de luz que recorre el documento y basado en la intensidad del reflejo de esa luz, se obtiene la información referente a la coloración de cada píxel. Toda esta información es recolectada por la herramienta de software del escáner, quién la procesa y posteriormente ofrece la opción de almacenarla en el equipo.



8

Joysticks

Para este dispositivo de entrada, conocido también como palanca de mando, no se tienen datos certeros sobre su origen; no obstante, tuvo su primera aplicación en la aviación, en donde estas palancas eran utilizadas para controlar los alerones de las naves hasta el presente.



9

Teclados

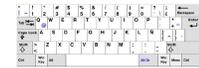
Los teclados indistintamente de su modelo y ergonomía tienen como función principal servir como dispositivo de entrada, específicamente para información de tipo texto y comandos que permiten acceder a funciones tanto básicas como avanzadas dentro de las herramientas de software.

Los teclados de forma general se encuentran subdivididos de la siguiente forma:

- Alfanumérico
- Numérico
- Teclas de cursor
- Funciones especiales

En cuanto a disposición de teclas se tienen las siguientes variantes:

- **QWERTY:** Es el de uso estándar. Heredado de las máquinas de escribir. Está patentado por Christopher Latham Sholes, quien luego lo vendió a la compañía E. Remington and Sons.
- **QWERTZ:** Utilizado en Alemania. La variante representativa es el cambio de posición entre la letra Y y Z.
- **AZERTY:** Utilizado en Francia y Bélgica.
- **DVORAK:** Patentado por August Dvorak y William Dealey en el año 1936, versión que posteriormente sufrió cambios hasta ser estandarizada en 1982 por el ANSI.



10

Mouse

Segundo dispositivo de entrada utilizado comúnmente en los sistemas de computación gráfica, el primero es el teclado.

El invento principal que dio paso a la futura generación del mouse como se conoce en la actualidad, es el del Trackball por parte de la fuerza Marina Canadiense, en el año 1952. Poco después en 1963 nace el primer prototipo a manos de Douglas Engelbart.

La apariencia del mouse ha persistido con el pasar de los años, dominando la presencia de los dos botones y la rueda; sin embargo, a este modelo se le han aplicado modificaciones para crear diseños enfocados en tareas más específicas, un ejemplo claro son mouse para gamers, que incluyen botones adicionales para beneficiar al jugador.



11

Formatos de archivos gráficos

Antes de entrar en detalles sobre los formatos de archivos gráficos, es importante recordar los dos tipos de imágenes con los que se puede trabajar: mapa de bits y vectorial.

- **Mapa de bits:** También denominados imágenes ráster, tiene como principal característica estar compuesta por píxeles. En este tipo de imágenes juega un rol de gran importancia la llamada resolución, debido a que, a mayor cantidad de píxeles, mejor será la calidad de la imagen. Su mayor problema se presenta en el momento de agrandar la imagen, dado que es común que se presente un pixelado y se pierda la calidad. Este tipo de gráfico se utiliza mayormente para fotografías, imágenes escaneadas o imágenes con alto nivel de detalles.
- **Vectorial:** Este tipo de imágenes se componen de curvas obtenidas mediante cálculos matemáticos generados por la herramienta de diseño. A diferencia de los mapas de bits, en las imágenes vectoriales no es necesario preocuparse por la resolución. Se utilizan con mayor frecuencia para el diseño de logotipos, iconos o gráficos destinados a entornos web.



12



13

Capítulo II: Manejo de elementos gráficos en entornos digitales

Algoritmos de discretización de líneas

Los algoritmos de discretización de líneas, como su nombre lo indica, están enfocados en los cálculos requeridos para el trazado de líneas en pantalla, para lo cual es necesario contar con dos puntos que responden a las coordenadas inicial y final de los extremos de la línea. Estos dos puntos son los primeros en mostrarse en pantalla y a partir de ella se obtienen posiciones aproximadas de los píxeles por los que pasará la línea. Estas posiciones siempre serán llevadas a valores enteros. Los cálculos de las posiciones se basan en la ecuación geométrica que determina a la línea:

$$y = mx + b \quad (1)$$

En la ecuación 1, **m** representa la pendiente y **b** la intercepción con el eje y.

14

Algoritmo DDA

Algoritmo Analizador Diferencial Digital, DDA, por sus siglas en inglés, se basa en la ecuación 2 para el cálculo de las posiciones.

$$\Delta y = m \Delta x \quad (2)$$

Este algoritmo inicia obteniendo los valores de Δx y Δy , que a su vez se utilizan para calcular la pendiente m . Esto implica que es necesario conocer las coordenadas de los extremos entre los que se va a trazar la línea; el punto inicial (x_i, y_i) y el punto final (x_f, y_f) .

Despejando la pendiente de la ecuación 2, se tiene que:

$$m = \frac{\Delta y}{\Delta x} = \frac{y_f - y_i}{x_f - x_i} \quad (3)$$

El valor de la pendiente determina las ecuaciones que se deben utilizar para realizar el determinar las siguientes posiciones.

15

Algoritmo DDA

Pendiente $m \leq 1$

$$y_{k+1} = y_k + m \quad (4)$$

Pendiente $m \geq 1$

$$x_{k+1} = x_k + \frac{1}{m} \quad (5)$$

El subíndice k indica la **cantidad de pasos o iteraciones** que se deben realizar para obtener todas las posiciones de los píxeles; k se determina evaluando el mayor entre los **valores absolutos** de Δx y Δy ; es decir:

Si $|\Delta x| > |\Delta y|$ entonces $k = \Delta x$. Si no, entonces $k = \Delta y$.

16

Algoritmo de Bresenham

Algoritmo creado por **Jack E. Bresenham** en 1965. Es un proceso de mayor precisión para el trazado de líneas, que además puede ser adaptado para la generación de curvas. De forma similar al algoritmo DDA requiere de un punto inicial y un punto final, a partir de los que se obtienen la pendiente m y las constantes: Δx , Δy , $2\Delta x$, $2\Delta y$.

Este algoritmo utiliza una variable llamada parámetro de decisión que generalmente se representa con una letra p minúscula con un subíndice que inicia en 0. El valor inicial de p se calcula en función de la pendiente, es decir:

- Si $0 < |m| < 1$:

$$p_0 = 2\Delta y - \Delta x \quad (6)$$

- Si $|m| > 1$:

$$\Delta y = -\Delta y \quad (7)$$

$$p_0 = 2\Delta x - \Delta y \quad (8)$$

17

Algoritmo de Bresenham

- En las ecuaciones 6 y 8 se aprecia que, de acuerdo al valor resultante de la pendiente m , se intercambian Δx y Δy . Este cambio aplica de forma similar en el cálculo del incremento de p para cada paso o iteración. El incremento de p se obtiene con las siguientes condiciones:

- Si $p_k < 0$ entonces

$x_k + 1, y_k$ // Se incrementa x en una unidad y se mantiene el valor de y .

$$p_{k+1} = p_k + 2\Delta y$$

- Si $p_k > 0$ entonces

$x_k + 1, y_k + 1$ // Se incrementan los valores de x e y en una unidad.

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

18



Algoritmo de discretización de circunferencias

Para la discretización de circunferencias existen al menos tres variantes de algoritmos: el básico, el de **Bresenham** y el de **punto medio**. Esta sección se enfoca en la aplicación de algoritmo de Bresenham para el dibujo de circunferencias.

El algoritmo requiere que sean suministrados el radio r y el punto en el que estará centrada la circunferencia (x_c, y_c) .

El punto inicial (x_0, y_0) está dado por el radio r , de la siguiente manera:

$$(x_0, y_0) = (0, r)$$

- Se observa que el valor de x_0 es **cero**, mientras que el de y_0 es el **radio** de la circunferencia. Lo siguiente es el cálculo del **parámetro de decisión**, mediante la ecuación 9:

$$p_{k=0} = 3 - 2r \quad (9)$$

Al evaluar el resultado de p_k se pueden dar dos situaciones:

- Si $p_k < 0$ entonces

$x_k + 1, y_k$ // Se **incrementa** x en una unidad y se **mantiene** el valor de y .

$$p_{k+1} = p_k + 4x_k + 6$$

19

Algoritmo de discretización de circunferencias

- Si $p_k > 0$ entonces

$x_k - 1, y_k - 1$ // Se **incrementa** x una unidad, mientras que y se decrementa.

$$p_{k+1} = p_k + 4(x_k - y_k) + 10$$

Las iteraciones se detienen cuando $x \geq y$. Para circunferencias con su centro en el origen, los valores de x_k y y_k se grafican de forma directa; no obstante, para aquellas circunferencias que su centro sea **distinto a (0, 0)** se debe realizar un paso adicional para cada x_k y y_k calculado:

$$x = x_k + x_c \quad (10)$$

$$y = y_k + y_c \quad (11)$$

Las ecuaciones **10** y **11** indican que para cada x calculada se le **suma** el valor de la **coordenada x de su centro**; lo mismo aplica en el caso de y .

La circunferencia se subdivide en octantes (ocho partes). Los puntos obtenidos con el algoritmo descrito corresponden al primer octante del cuadrante 1. Luego de haber calculado los puntos, se requiere determinar la **simetría** para el resto de los octantes y así completar la circunferencia.

20

Algoritmo de discretización de elipses

La elipse se caracteriza por ser una **circunferencia alargada y simétrica con respecto a dos ejes**, por ende, posee dos radios o distancias diferentes desde su centro. Como resultado de las modificaciones aplicadas a los algoritmos para dibujar circunferencias se obtiene el llamado **algoritmo de punto medio para el dibujo de elipses**, en el cual se divide el primer cuadrante en **dos regiones**.

Similar a la circunferencia se deben suministrar los radios denominados como r_x y r_y , además del punto central de la elipse (x_0, y_0) . A continuación, se presentan las condiciones y cálculos para cada región:

- **Región 1:**

El punto inicial (x_0, y_0) está dado por el radio r_x :
 $(x_0, y_0) = (0, r_y)$

El **valor inicial del parámetro de decisión pa** se obtiene con la ecuación 12:

$$pa_{k=0} = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2 \quad (12)$$

En donde para cada pa_k se verifican las siguientes condiciones, mientras $2r_y^2 x \geq 2r_x^2 y$:

Si $pa_k < 0$ entonces

$x_k + 1, y_k$ // Se incrementa x en una unidad y se mantiene el valor de y.

$$pa_{k+1} = pa_k + 2r_y^2 x_{k+1} + r_y^2$$

Si $pa_k > 0$ entonces

$x_k + 1, y_k - 1$ // Se incrementa x una unidad, mientras que y se decrementa.

$$pa_{k+1} = pa_k + 2r_y^2 x_{k+1} + r_y^2 - 2r_x^2 y_{k+1}$$

21

Algoritmo de discretización de elipses

Si $pb_k < 0$ entonces

$x_k + 1, y_k - 1$ // Se incrementa x una unidad, mientras que y se decrementa.

$$pb_{k+1} = pb_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_x^2$$

Si $pb_k > 0$ entonces

$x_k, y_k - 1$ // Se mantiene el valor de x y se decrementa y en una unidad.

$$pb_{k+1} = pb_k + 2r_x^2 y_{k+1} + r_x^2$$

- **Región 2:**

El punto inicial (x_0, y_0) está dado por los últimos puntos obtenidos en la región 1:
 $(x_0, y_0) = (0, y)$

El **valor inicial del parámetro de decisión pb** se obtiene con la ecuación 13:

$$pb_{k=0} = r_y^2 \left(x_0 + \frac{1}{2}\right)^2 - r_x^2 (y_0 + 1)^2 - r_x^2 r_y^2 \quad (13)$$

- En donde para cada pb_k se verifican las siguientes condiciones, mientras $y > 0$:

22

Algoritmo de relleno de figuras

Los algoritmos de relleno de figuras son los utilizados para otorgar o reemplazar el color existente dentro de las celdas o píxeles que componen una figura (mapa de bits). Ya con un enfoque a herramientas de diseño gráfico, esta acción la suele realizar la opción conocida como "bote de pintura", que, con hacer clic dentro de una región, cambia al color especificado.

El algoritmo mayormente conocido es el FloodFill. Es un método recursivo que trabaja utilizando una pila en la que almacena todas las posiciones que conforman la zona a colorear.

De forma general, el algoritmo FloodFill trabaja de la siguiente forma, mientras la pila no esté vacía:

- Se ingresa un punto dentro de la figura, este es el punto inicial. Se cambia su color al deseado.
- Se realiza una verificación de los puntos vecinos, ya sea tomando cuatro direcciones (norte, sur, este, oeste) u ocho direcciones (norte, sur, este, oeste, noreste, sureste, noroeste, suroeste) y se almacenan en la pila.
- Se saca el último elemento de la pila, se cambia su color, y nuevamente se verifican sus vecinos.

23

Métodos de transformación

En esta sección se explican los métodos de transformación o transformaciones geométricas aplicables a las figuras u objetos que se encuentran dentro de una escena.

Las transformaciones más comunes son las que implican mover, rotar o escalar elementos.

Las transformaciones vistas en los siguientes puntos están enfocadas en un plano bidimensional.

24

Traslación

La traslación por definición consiste en mover o trasladar un objeto de una posición **P** a otra **P'** siguiendo una trayectoria en línea recta (Steezman, 2015). Las posiciones corresponden a la ubicación del objeto dentro de un sistema de coordenadas. La distancia en la que se traslada el objeto es denotada por la letra **t**. Sabiendo esto, tenemos la siguiente fórmula:

$$P' = (x', y') = P + t \quad (14)$$

La ecuación 14 se desglosa en dos ecuaciones; una para cada eje de coordenadas:

$$x' = x + t_x \quad (15)$$

$$y' = y + t_y \quad (16)$$

En las ecuaciones 15 y 16 se observa que se tiene una distancia **t** para cada eje. En computación gráfica, la ecuación 14 se representa mediante una suma de matrices de la siguiente forma:

$$\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

25

Rotación

La rotación por su parte consiste en mover los objetos sobre un punto de pivote (x, y) , dado un ángulo θ . Esto obedece el hecho de que, si el ángulo es positivo, el giro será en sentido contrario a las manecillas del reloj; y si es negativo, va en dirección de las manecillas del reloj.

La ecuación general para esta transformación es la siguiente:

$$p' = (x', y') = r \cdot P \quad (17)$$

En donde r representa el ángulo de rotación que se aplicará sobre el punto P . Para efectos de representación y cálculo se recurre al uso de las identidades trigonométricas.

Desglosando la ecuación 17, se obtiene lo siguiente:

$$x' = x \cos \theta - y \sin \theta \quad (18)$$

$$y' = x \sin \theta + y \cos \theta \quad (19)$$

La representación matricial para la ecuación 17 es:

$$\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix} = P'$$

26

Escalado

Las transformaciones anteriores, traslación y rotación, consistían en un movimiento de puntos o de una figura sin alterar sus proporciones; no obstante, durante el escalado sí es posible que se presenten deformaciones.

Un factor de escalado menor a uno, reducirá el tamaño en el eje que se esté aplicando; por otra parte, un factor mayor a uno aumenta o expande la figura en el eje indicado. Los factores de escalado no necesariamente deben ser iguales para ambos ejes (Steezman, 2015).

A continuación, se muestran las ecuaciones para escalar las dimensiones de un punto:

$$x' = x \cdot S_x \quad (20)$$

$$y' = y \cdot S_y \quad (21)$$

$$\begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

En donde S_x y S_y son los factores de escalado. En forma matricial

27

Simetría y recorte

Al hablar de simetría se hace referencia a una condición particular en la que un objeto o gráfico en 2D al ser dividido por una línea, la porción de un lado de la línea es un reflejo de la porción del otro lado.

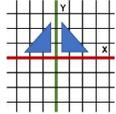
El principio de simetría es de gran utilidad en el entorno de computación gráfica, principalmente durante el modelado de objetos, como jarrón o un auto; puesto que solo se requiere modelar y definir detalles de un solo lado, para entonces aplicar alguna herramienta que copie y genere un reflejo de la parte trabajada, completando así la figura en menor tiempo.

De esta forma se cuenta con cuatro tipos de simetría con su representación matricial. La simetría que se da con respecto a un eje, también se conocen con el nombre de reflexión.

28

Simetría y recorte

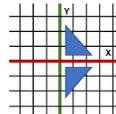
Simetría con **respecto al eje y**:



Para obtener los puntos de simetría con respecto al eje y, se deben multiplicar los puntos por la siguiente matriz:

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

Simetría con **respecto al eje x**:



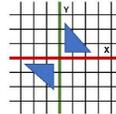
Los puntos de simetría con el eje x se obtienen al multiplicar los puntos por la siguiente matriz:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

29

Simetría y recorte

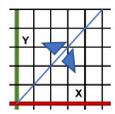
Simetría con **respecto al origen**:



La simetría con respecto al origen se da al multiplicar los puntos por la matriz:

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

Simetría con **respecto a la recta x=y**:



Los puntos de simetría en relación a una recta donde los valores de x son iguales a los de y, se obtienen al multiplicar por:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

30

Sistema de coordenadas homogéneas

En los puntos anteriores, relacionados con las transformaciones en 2D, se mostraron las representaciones matriciales para cada movimiento, no obstante, los cálculos se realizaban directamente con las ecuaciones presentadas. El sistema de coordenadas homogéneas tiene como propósito utilizar directamente estas matrices para obtener los resultados deseados, facilitando la multiplicación entre matrices.

Las matrices presentadas en los puntos anteriores son de tamaño 2x2 (2 filas y 2 columnas); en el sistema de coordenadas homogéneas se convierten a matrices 3x3. Esto se logra al representar los puntos mediante un vector con la forma (hx, hy, h). En donde hx y hy son los puntos x e y conocidos; mientras que h suele tomar el valor 1.

31



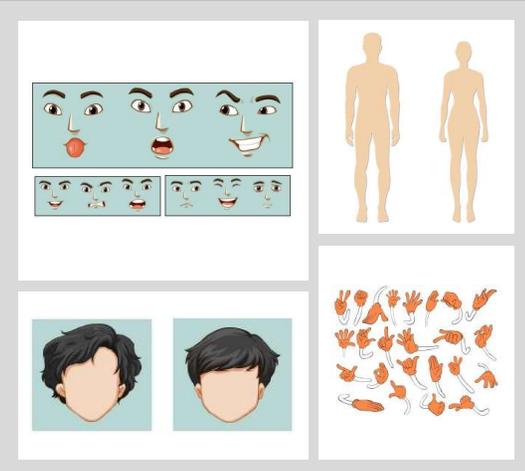
Capítulo III: Fundamentos de animación 2D

32

Capítulo III: Fundamentos de animación 2D

Introducción al bosquejo de la figura humana

- Dimensionamiento por tamaño de cabeza
- Figura masculina y femenina
- Rostro
- Manos y pies

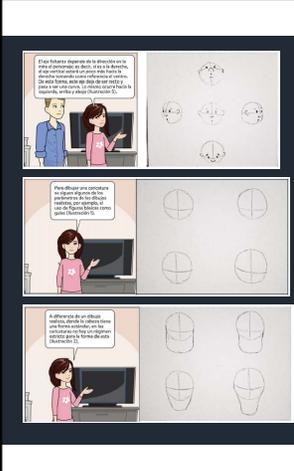


33

Caricatura

Una caricatura es una representación simplificada de un dibujo realista, utilizada normalmente en el sector de entretenimiento para la transmisión de ideas o mensajes con un toque exagerado y divertido, llegando incluso al sarcasmo. Este término se asocia con mayor frecuencia a los cortos animados o "cómicas" como se conocen comúnmente la secuencia de dibujos en 2D que dan como resultado una escena.

Los personajes en estilo caricatura se distinguen por sus rasgos exagerados: cabezas de mayor proporción respecto al cuerpo, ojos, nariz y orejas grandes; además de la simplicidad de elementos como el cabello.



34

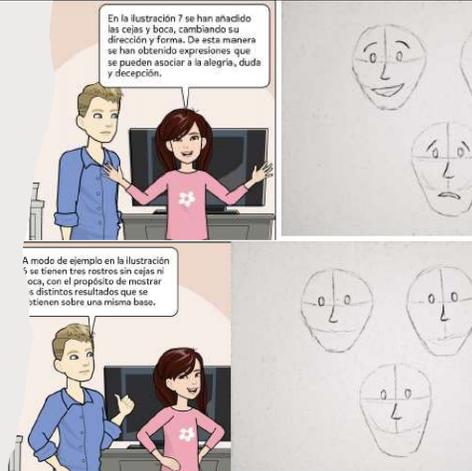
Evolución Emocional

Las emociones involucran cambios de estados de ánimo causados por una infinidad de factores. Estas emociones pueden ser plasmadas en dibujos, tanto realistas como caricaturas, mediante pequeños cambios en la expresión de los personajes.

Los elementos que permiten transmitir esa evolución emocional son las cejas, boca e incluso la nariz.

En la ilustración 7 se han añadido las cejas y boca, cambiando su dirección y forma. De esta manera se han obtenido expresiones que se pueden asociar a la alegría, duda y decepción.

A modo de ejemplo en la ilustración 5 se tienen tres rostros sin cejas ni oca, con el propósito de mostrar a distintos resultados que se obtienen sobre una misma base.



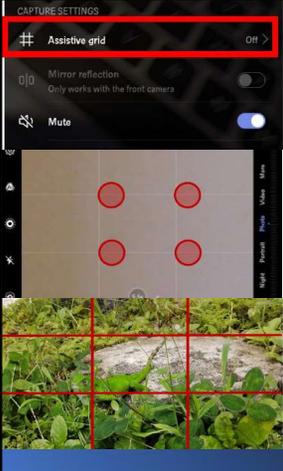
35

Reglas de Tercios

La regla de tercios es una técnica utilizada comúnmente en fotografía, incluso, las cámaras digitales y teléfonos inteligentes incluyen una opción con un nombre similar a "cuadrícula" o "grid".

Esta herramienta lo que hace es dividir la pantalla en tres tercios horizontales y tres tercios verticales, obteniendo así nueve cuadrantes. Las intersecciones de estas divisiones se les llaman puntos fuertes.

La técnica consiste en ubicar el centro de interés de la imagen en uno de esos cuatro puntos, ilustración 30. En caso de que existan dos puntos de interés, lo recomendable es situarlos en puntos diagonalmente opuestos.



36

Teoría de Color

La teoría del color es un conocimiento básico para los artistas y diseñadores, pero no solo se limita a ellos ya que incluso en la vida cotidiana es común recurrir a este conocimiento hasta para el simple hecho de combinar las piezas de vestir para asistir a la universidad, una entrevista o reunión familiar; con el propósito de adecuarlo a cada ocasión.

La teoría del color se enfoca en los lineamientos para combinar los colores, considerando una serie de atributos como matiz, luminosidad y saturación; y para esto, se vale de una herramienta conocida como círculo cromático.

El círculo cromático es una paleta de colores en la que se incluyen los colores primarios rojo, azul y amarillo; de los cuales se derivan los colores secundarios, terciarios, ilustración 31. En este círculo también se subdividen los colores en cálidos y fríos. Las combinaciones comunes son las siguientes:

- Complementarios: Se combinan los colores que se encuentran opuestos en el círculo cromático. Ejemplo: rojo y verde en la ilustración 31.
- Análogos: Se refiere a tres colores situados de manera consecutiva en el círculo cromático. Ejemplo: verde, turquesa y azul, de la ilustración 31.
- Triádicas: Es la combinación resultante de tres colores equidistantes, como el verde naranja y violeta marcados por los lados del triángulo negro

37

Principios de animación

Los principios básicos de animación tienen por autores a los animadores Ollie Johnston y Frank Thomas, ambos de la compañía Disney. Estos principios se dieron a conocer en la obra: The illusion of life: Disney Animation.

Estos principios son una colección de técnicas para animar personajes, sin embargo, tienen aplicaciones en otros campos. Presentaremos en resumen cada uno de estos principios de animación:

38

Principios de animación

Estirar y encoger: Aumento y disminución del tamaño de los cuerpos.	Anticipación: Permite anticipar la acción a realizar.	Puesta en escena: Resaltar la presencia en escena del objeto y resaltar su forma.	Acción directa y pose a pose: Secuencia de pasos que conducen a una acción no esperada.
Acción continuada y superposición: Se mezclan movimientos para resaltar la acción.	Entradas lentas y salidas lentas: La acción se divide en entrada, centro y salidas. La entrada y salidas son ralentizadas mientras que el centro se acelera la acción.	Arcos: Las trayectorias de arcos son más realistas y naturales.	Acción secundaria: Movimientos que complementan la acción principal para resaltarla.
Ritmo (Timing): Número de eventos o cuadros que se utilizan para una acción en un tiempo dado. Contribuye a la realidad de los objetos.	Exageración: Incrementar el efecto de una acción.	Dibujos sólidos: Resaltar las perspectivas de los objetos para dar volumen y peso.	Personalidad o apariencia: Garantizar que el objeto se coherente con sus movimientos.

39

Synfig

El software de código libre Synfig es la abreviatura de Synfig Studio y permite editar gráficos vectoriales para posteriormente animar estos gráficos. Su creador fue Robert Quattlebaum y ha sido creado para el diseño de películas animadas con el mínimo de recursos para realizarlo.

Synfig Studio

40



41

Capítulo IV: Herramientas de generación de imágenes 3D

Introducción a la herramienta

En el capítulo anterior se trataron las técnicas y principios básicos que dieron origen a las representaciones gráficas que existen en la actualidad. Vivimos ahora en la era digital, en donde los gráficos no se limitan a las propiedades de largo y anchura, sino que también incluyen profundidad, un atributo que da paso a la gama de las imágenes tridimensionales o 3D. Estas animaciones y diseños en 3D, son las que ofrecen mayor realismo de las representaciones a los espectadores; beneficios aprovechados por otras áreas como la medicina, robótica, arquitectura, entre otras.

Existen múltiples herramientas creadas para el modelado y desarrollo de animaciones en 3D, algunas de software libre y otros de pago. Algunos ejemplos de estas herramientas de diseño son:

- AutoCAD
- Sketchup,
- Maya,
- Revit,
- 3Ds Max
- Blender



42

Secciones del Área de Trabajo

Las secciones de trabajo se encuentran más organizadas en la versión 2.8 en comparación con las anteriores; no obstante, en todas se mantiene la barra de menú con las opciones comúnmente conocidas, a excepción de la sección Render que ofrece acceso rápido para el renderizado de imágenes o videos.

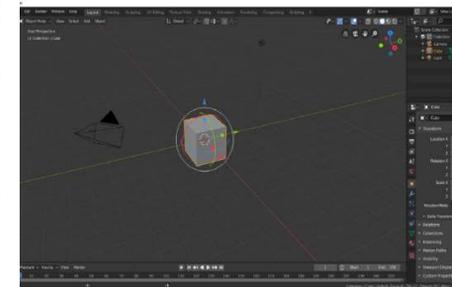
Pueden apreciar que, al lado de la barra de menú, se observan una serie de pestañas, que son accesos directos a los espacios de trabajo predefinidos. Vale destacar que estas pestañas se pueden modificar, así como la distribución de las opciones de cada una de ellas, para comodidad del usuario.

43

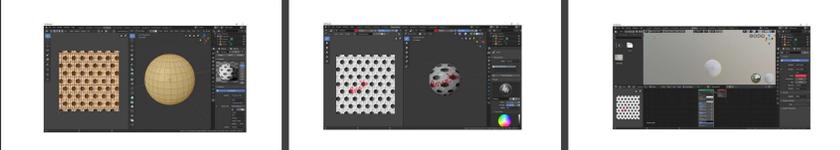
Layout

La ventana que se abre por defecto al ingresar a Blender es Layout. Esta ventana contiene todas las herramientas y modificadores de los objetos como cuerpos sólidos, es decir, se trabaja el objeto como un todo y no por los elementos que lo componen (caras, aristas y vértices).

Cuenta con un panel a la derecha en la que se visualiza un listado de los elementos que están presente en la escena; mientras que debajo de este, se encuentran distintas opciones entre la que destaca la de las transformaciones básicas, localización y escala de los objetos.



44

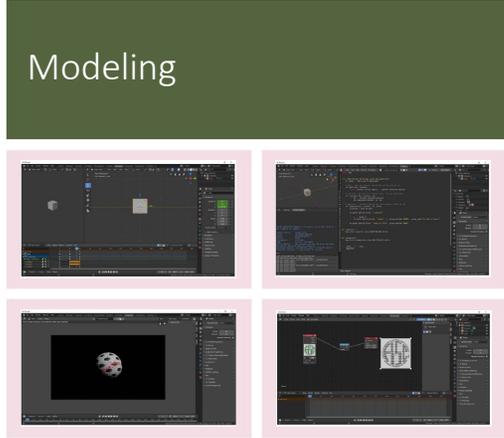


Layout

- **Modeling:** Este modo habilita la edición puntual de los componentes de los objetos. En el lado izquierdo de la pantalla se ofrecen un conjunto de opciones como insertar caras, hacer cortes, suavizar bordes, entre otros. Los objetos en este modo se tornan de un tono amarillo.
- **Sculpting:** Ofrece una serie de herramientas para alterar la forma original de los objetos desde un modo sólido. Los efectos obtenidos se asocian a relieve sobre la superficie de los objetos. Entre sus opciones está la posibilidad de trabajar de forma simétrica los cambios, lo que es ideal para objetos que presenten esta propiedad.
- **UV Editing.** En el mundo de digital se cuenta con una serie de recursos gráficos entre las que destacan las imágenes con texturas. En el modo UV editing se ofrece la alternativa de modificar la distribución de una imagen de textura sobre un determinado objeto (ver Figura 37) cuando la presentación obtenida luego de aplicado un material no es la esperada.
- **Texture Paint:** Complemento del modo UV editing. Muchos diseñadores prefieren añadir un toque personal en el material gráfico que trabajan, como se muestra en la Figura 38, en este modo se pueden hacer modificaciones libres a la textura aplicada sobre los objetos.
- **Shading:** El shading se refiere a los efectos visuales que tendrá el objeto dentro de la escena 3D. En esta ventana (Figura 39) se modifican los atributos de los objetos para obtener los efectos deseados.

45

Modeling

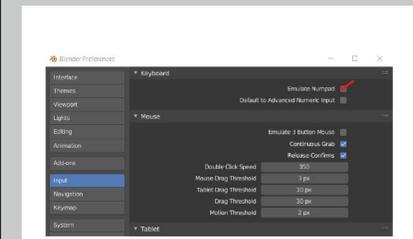


- **Animation:** Modo en el que se modifican las posiciones y tamaño de los objetos en el tiempo con el propósito de generar secuencias de cuadros (frame) que al renderizarse produzcan una animación. Su característica más sobresaliente es la sección inferior en la que se encuentra una línea de tiempo. En ella se colocan los "keys" (puntos) para fijar la posición de un objeto en un determinado frame.
- **Rendering:** Ventana en la que se puede observar el resultado del renderizado.
- **Compositing:** Alternativa para el tratamiento de imágenes o películas a través de uso de nodos. Una misma configuración puede ser aplicada a una o todas las imágenes de una película con respecto al tiempo.
- **Scripting:** Blender incluye un editor de texto con el que usuarios avanzados pueden crear, modificar, animar o gestionar procesos complejos sobre los objetos mediante la creación de scripts en Python.

46

Vistas

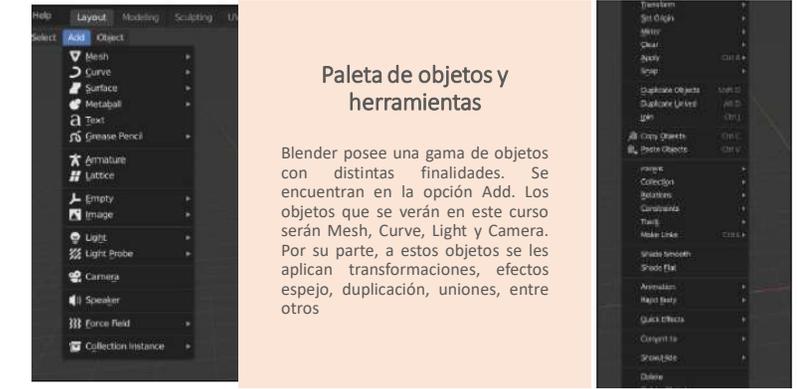
Al hablar de vistas, Blender ofrece dos opciones: User Perspective y User Orthographic. Se cambia entre una y otra con el botón 5 del teclado numérico. En caso de no poseer un teclado numérico, Blender ofrece la alternativa de "emular" estos botones; para esto se accede en el menú Edit > Preferences > Input y se marca la opción de "Emulate Numpad" En la Tabla 1 se recopilan las vistas básicas.



VISTA	TECLAS NUMPAD
User Perspective / User Orthographic	5
Front	1
Back	Ctrl + 1
Right	3
Left	Ctrl + 3
Top	7
Bottom	9

47

Paleta de objetos y herramientas



Blender posee una gama de objetos con distintas finalidades. Se encuentran en la opción Add. Los objetos que se verán en este curso serán Mesh, Curve, Light y Camera. Por su parte, a estos objetos se les aplican transformaciones, efectos espejo, duplicación, uniones, entre otros.

48

Transformaciones

Las siguientes transformaciones son las básicas para cualquier objeto y se utilizan mucho al momento de crear animaciones.

- **Traslación**

La Traslación es el movimiento sobre una figura en la que toda su estructura se desplaza en la misma dirección. Una vez realizada la acción tenemos otra figura idéntica, pero en otra posición diferente a la inicial.

- **Rotación**

La rotación es un giro alrededor de un punto dentro de la figura mientras que se conserva su forma y tamaño original. En el movimiento de rotación influyen factores como la amplitud de la rotación que vendrá determinada por el ángulo.

- **Escalados**

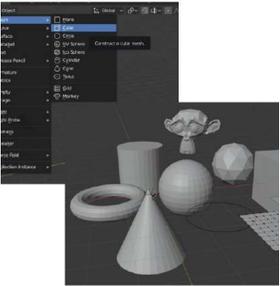
El escalado de figuras cambia el tamaño en los planos vertical u horizontal. El cambio de la escala se realiza con respecto a un punto de referencia que varía según el plano de escalado que elija.

49

Formas Básicas

La forma básica que Blender muestra por defecto al ingresar al entorno es un cubo; no obstante, posee otras formas y se encuentran dentro de la categoría de Mesh en el menú Add.

En la Figura se observa que dentro de Mesh se encuentran las formas: plano, cubo, círculo, esfera, cilindro, cono, torus, un plano subdividido e incluso una plantilla del rostro de un mono. Estas formas básicas pueden ser modificadas en el modo edición o pestaña de modelling, que ofrece otras opciones además de las transformaciones básicas vistas en el video de la sección 1.4.

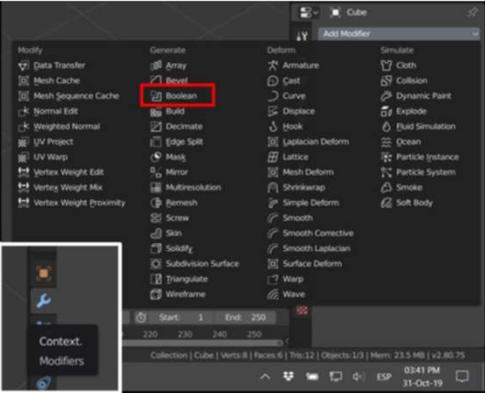


50

Operaciones Binarias

Las operaciones binarias u operaciones booleanas en Blender son las mismas que las tratadas con los diagramas de Venn: unión, diferencia e intersección.

Estas operaciones en Blender se añaden o aplican los objetos mediante el uso de modificadores. Estos se encuentran en el panel derecho y su icono es una llave.

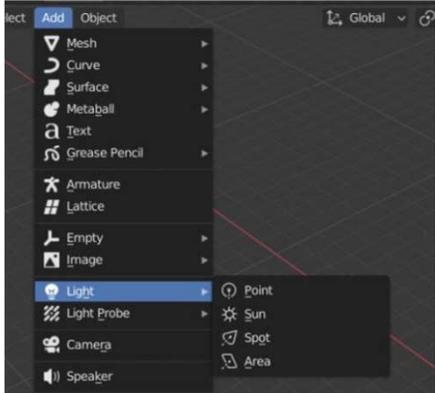


51

Fuentes de Luz

Blender proporciona opciones de iluminación para dar distintos efectos a las escenas que se estén trabajando. Se añaden a la escena como cualquier otro objeto desde el menú **Add -> Light**.

A cada lámpara se le puede personalizar su color y nivel de intensidad de acuerdo con lo que se requiera.



52

Fuentes de Luz

- **Point light**
El point light o punto de luz, como su nombre lo indica, es un tipo de lámpara que emite la luz en forma **radial**, lo que significa que emite la misma cantidad de luz en todas direcciones.
- **Spot light**
Es un tipo de lámpara con forma cónica que emite luz en una dirección. Su apariencia dentro de la escena se asemeja a la de un reflector
- **Area light**
Lámpara que ofrece un efecto de distribución de luz más realidad debido a que no solo ilumina los objetos que tenga cerca, sino que también proporciona una iluminación menos intensa a los objetos cercanos al principal
- **Sun light**
Tiene un mayor alcance que las anteriores. No se ubica dentro de la escena, no obstante, se controla la dirección en la que proviene la luz que ilumina los objetos

53

Tipos de Cámara

Para personalizar la cámara o los tipos de cámaras a utilizar para renderizar, Blender posee una opción en su panel lateral derecho, cuyo ícono es una filmadora color verde. Al activar esta opción, se muestra un menú en el que se ofrecen tres tipos de lentes: Perspectiva, Ortogonal y Panorámico.

54

Formas basadas en splines



Los splines son una herramienta vista anteriormente en las herramientas para dibujos en 2D y en Blender tienen la misma función: crear curvas. Identificadas como curvas de Bézier son estructuras que permiten definir curvas que de otra forma sería muy complejo modelar en 3D. Similar a otros objetos, se añaden desde el menú Add -> Curve -> Bézier

Al ingresar al modo edición, se aprecia la aparición de las extensiones en algunos vértices, indispensables para el manejo de las curvas. En el panel izquierdo se aprecian otras opciones entre las cuales está la creación de una curva a mano libre.

55

Materiales y texturas

En esta sección se tratará la creación de materiales y texturas, elemento posterior a la creación de modelos en 3D con el objetivo de brindarles una mayor sensación de realismo. La etapa seguida a la creación de objetos en 3D es la asignación de colores o texturas que, teniendo el objeto seleccionado, se puede realizar de las siguientes maneras:

- Creación de materiales. Se accede desde el panel inferior derecho en el botón
- Uso de texturas. Dentro del menú de Base Color se encuentra la opción image textura. Esta opción permite la utilización de imágenes con texturas distintas a las predefinidas por Blender, por ejemplo, la textura de un dado para un cubo; no obstante

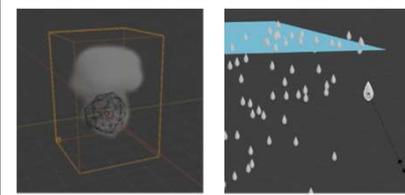
56

Efectos especiales aplicados a los objetos

Algunos de los efectos especiales de Blender se encuentran asociados a la opción de Modifiers en la sección de Simulation. Entre estos se encuentran Cloth, Collision, Dynamic Paint, Explode, Fluid Simulation, Ocean, Particle Instance, Particle System, Smoke y Soft Body.



57



Efectos Atmosféricos

En cuanto a efectos atmosféricos, lo más comunes son escenas de día o de noche, sin embargo, estos solo requieren jugar con las fuentes de luz para lograr la sensación deseada. En esta sección se trabajará una simulación sencilla de lluvia mediante el uso de partículas.

Ejemplo: Lluvia – Partículas

Ejemplo: Smoke – Force Fields



58