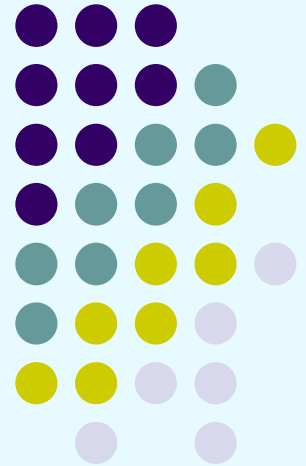
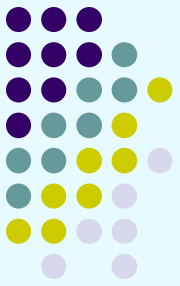


Modelado y animación de objetos 2D y 3D



Daniela Varela
Viterbo Rodríguez
Pablo Rasilla

Índice



1 – Modelado y animación 2D



2 – Uso de modelos 2D

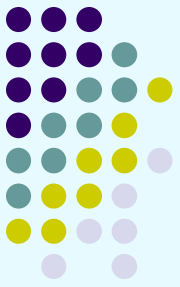


3 – Modelado y animación 3D

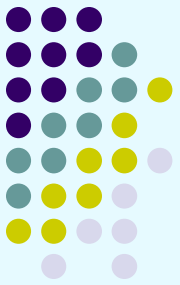


4 – Uso de modelos 3D

1 – Modelado y animación 2D



Introducción

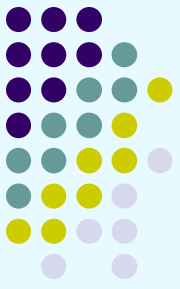


“El arte es una colaboración entre Dios y el artista, y mientras menos haga el artista; mejor.” – Andre Gide (1869 – 1951)

El primer paso es dibujar los personajes para tener una referencia al animarlo, para ello consideraremos:

- Importancia de un dibujo sólido
- Construcción de un personaje
- Conceptos básicos de la animación
- Animación cíclica
- Herramientas para el dibujo y la animación

Importancia de un dibujo sólido



- **Volumen**

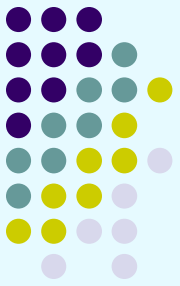
- Evitar que los dibujos queden planos.
- Se piensa en 3D.
- Perspectiva?



- **Proporciones**

- Cartoon vs. realista.
- No excederse en las deformaciones.
- No agregar peso. Conservar el volumen inicial.

Importancia de un dibujo sólido



- **Equilibrio**

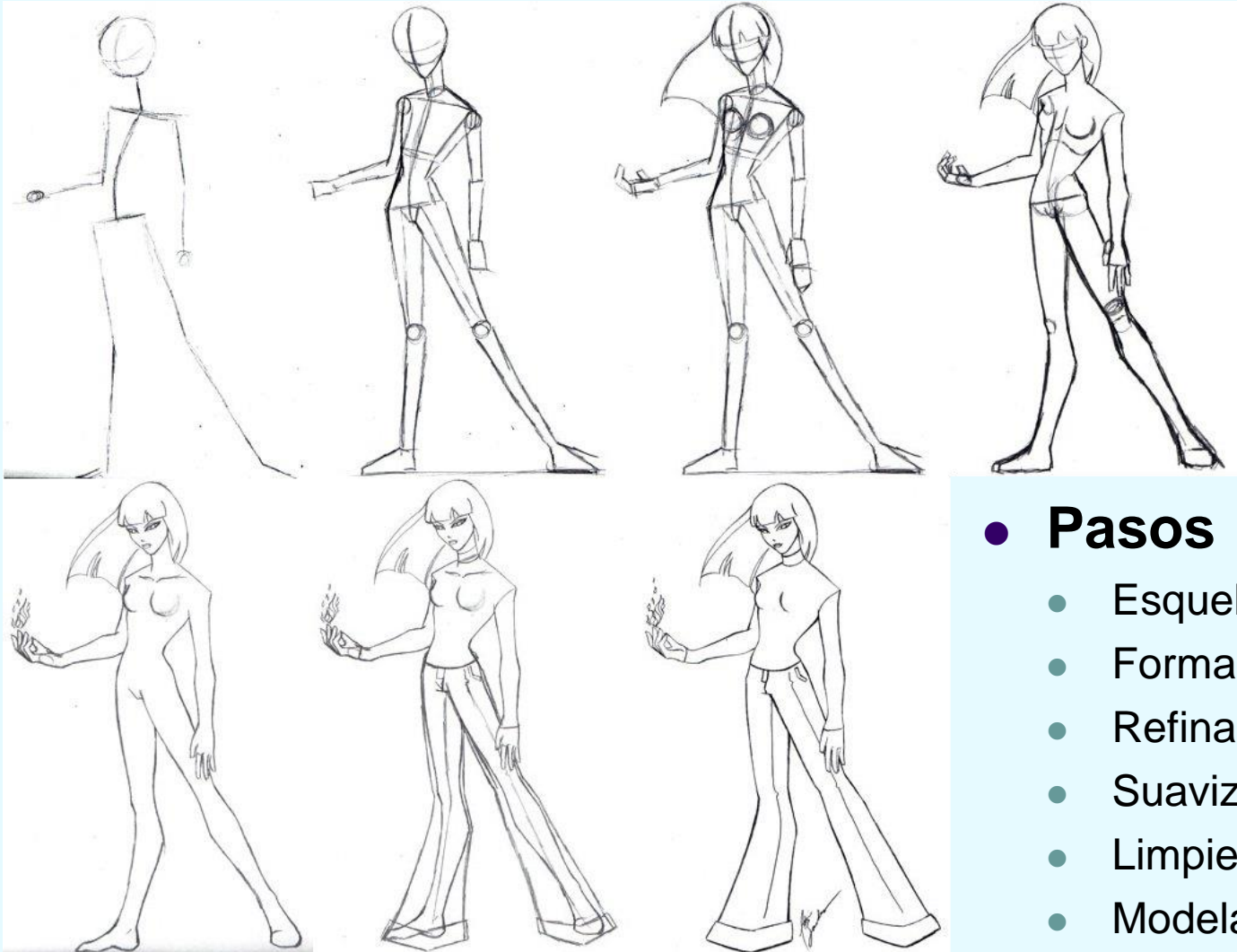
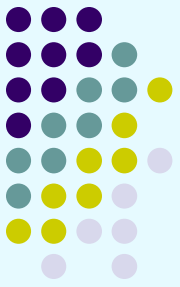
- Mantener el centro de masa apoyado para las poses estáticas.
- Saber romper el equilibrio para lograr el movimiento.



- **Peso**

- El movimiento no es más que trasladar peso de un lado a otro.
- Mover el peso con coherencia es la base para una buena animación.

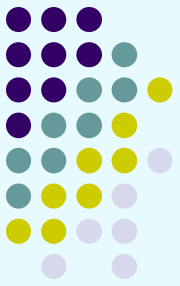
Construcción de un personaje



● Pasos

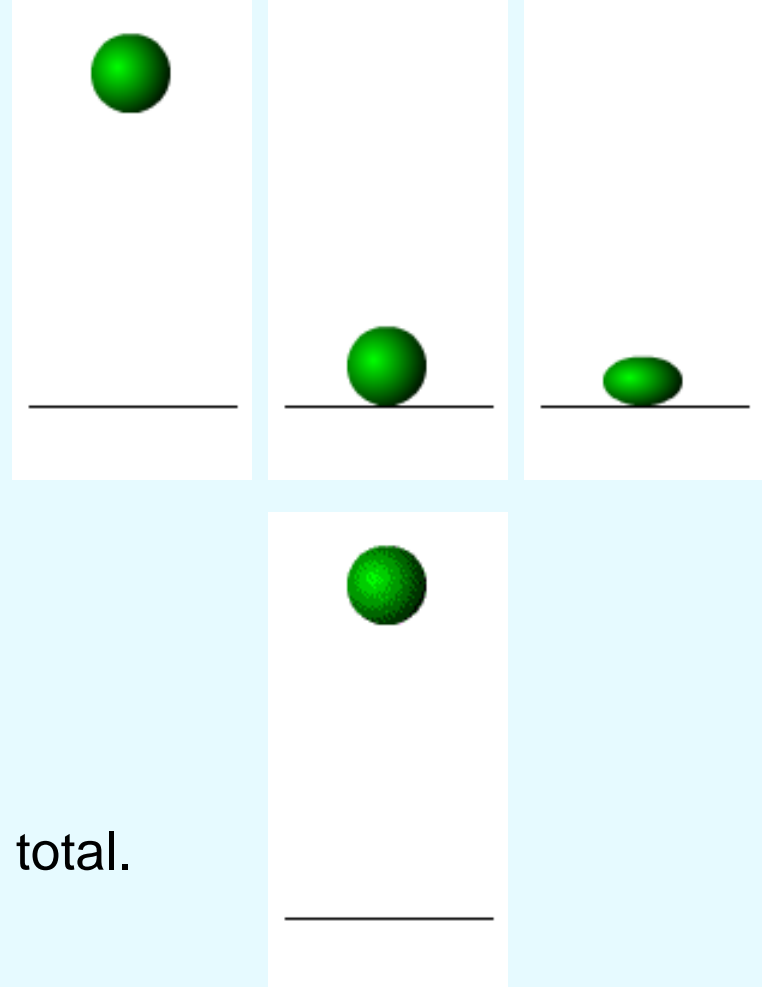
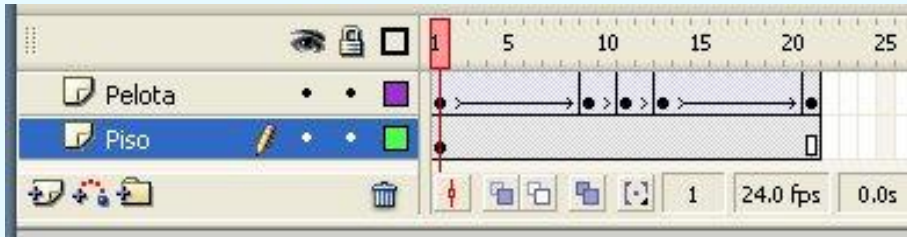
- Esqueleto
- Formas básicas
- Refinamiento
- Suavizar curvas
- Limpieza y detalles
- Modelado de la ropa
- Limpieza final

Conceptos básicos de la animación



- **Key Frames**

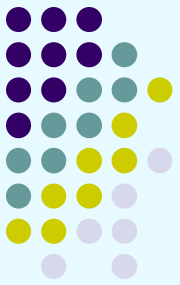
- Momentos clave de la animación.
- No necesariamente equiespaciadas en el tiempo.



- **Intercalación**

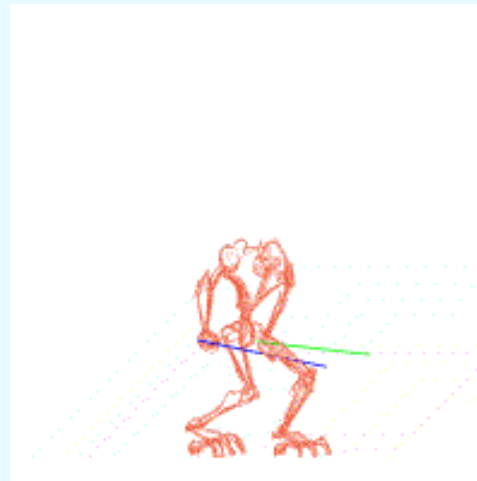
- Proceso automatizable?
- Restringido por la cantidad de cuadros total.

Animación Cíclica

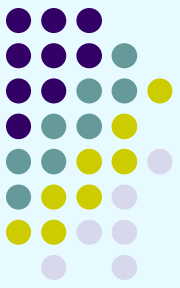


- **Aspectos a tener en cuenta**

- Loop (primer y último frame deben tener fluidez).
- Balance fluidez vs. cantidad de cuadros.
- Todos son key frames?
- Herramientas? (Onion Skin)



Herramientas para el dibujo y la animación

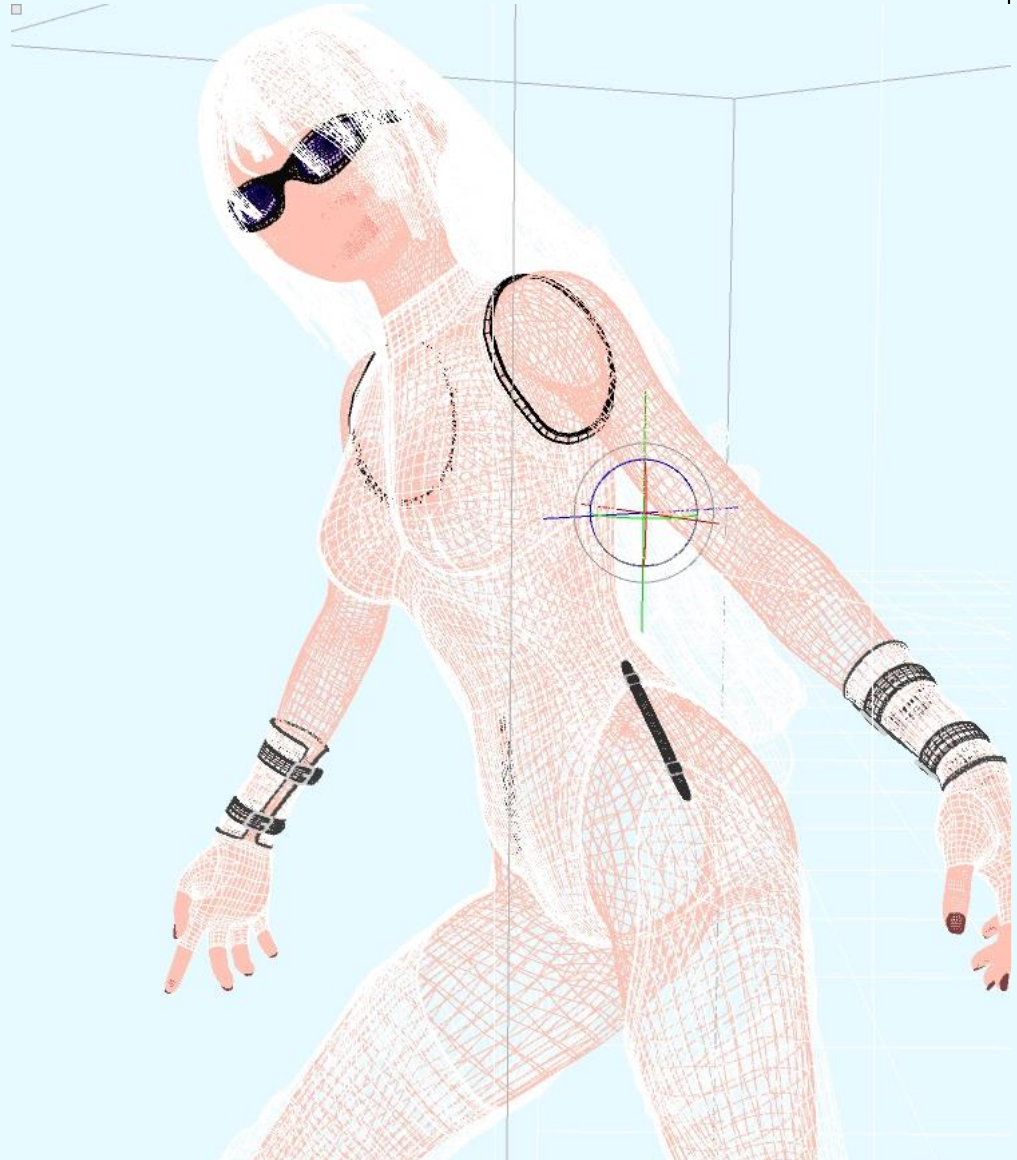
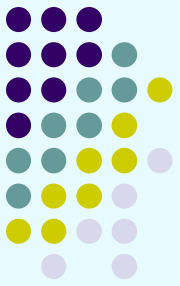


- Herramientas comerciales

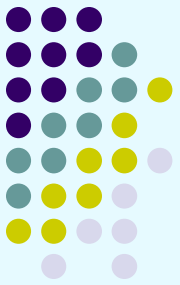


- Herramientas gratuitas

2 – Uso de modelos 2D



Introducción

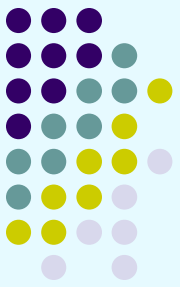


“La animación puede explicar lo que sea que la mente del hombre pueda concebir.” – Walt Disney (1901 – 1966)

Ahora hay que saber cuales cuadros se necesitan para las animaciones.

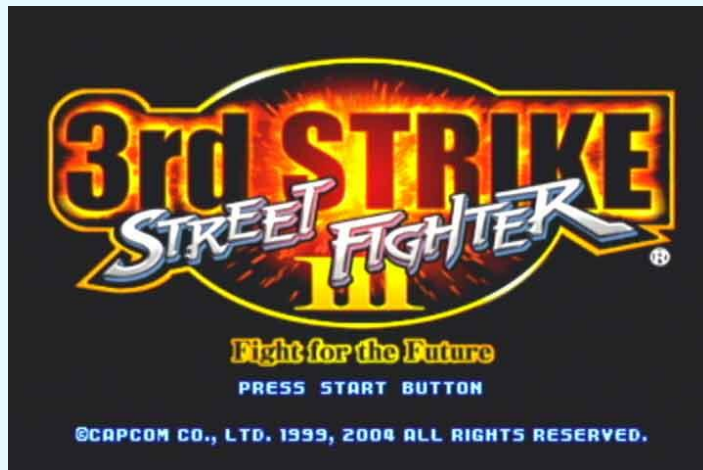
- Distintas animaciones de un personaje
- Interacción: programador – artista
- Efecto 3D con movimiento de capas

Distintas animaciones de un personaje

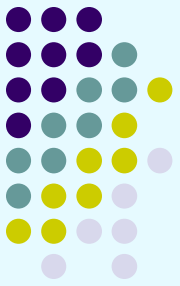


- **Típicamente aparecen:**

- Stand (básico)
- Walk / Run
- Jump / Fall
- Knee
- Attack / Action
- Damage / Dead
- Combinaciones
- Que necesita el programador ?



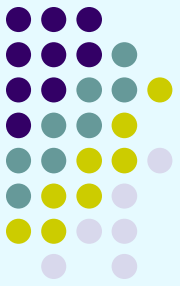
Interacción: programador – artista



- **Que nos exporta el artista ?**
 - Lista de cuadros (archivos separados) para cada animación
 - Formato: calidad, canal alpha (típicamente png)
 - Es importante definir una nomenclatura en los nombres
 - Ej: Ken_Stand_1.png, Ken_Stand_2.png, etc...
 - Punto de registro en la imagen
 - Duración de cada cuadro



Interacción: programador – artista

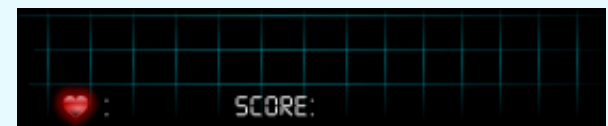


- **Cajas y Offsets**

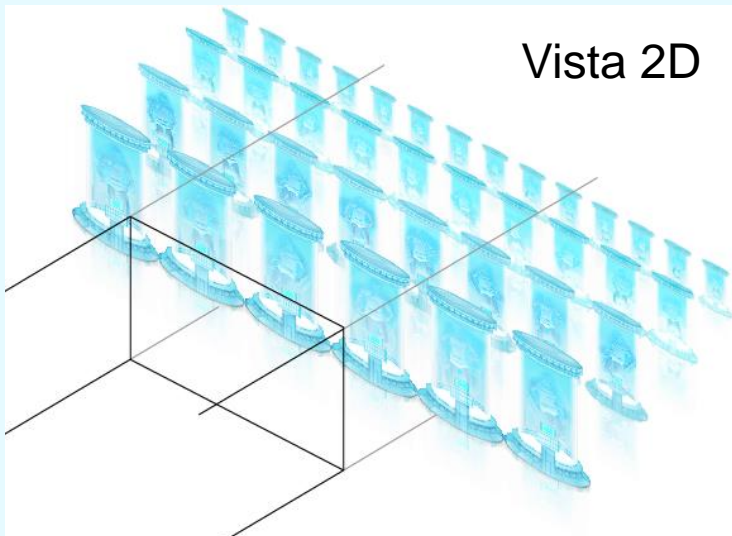
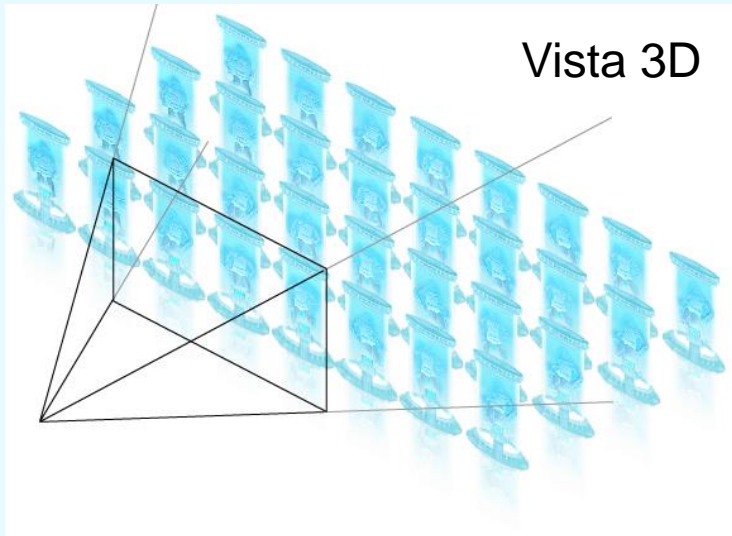
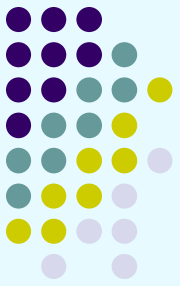
- El artista tiende a usar cajas que ajusten perfectas a la imagen.
- La ubicación de cada imagen en pantalla quien la define?
- Soluciones:
 - Todas las cajas iguales
 - Cajas justas pero con números redondos



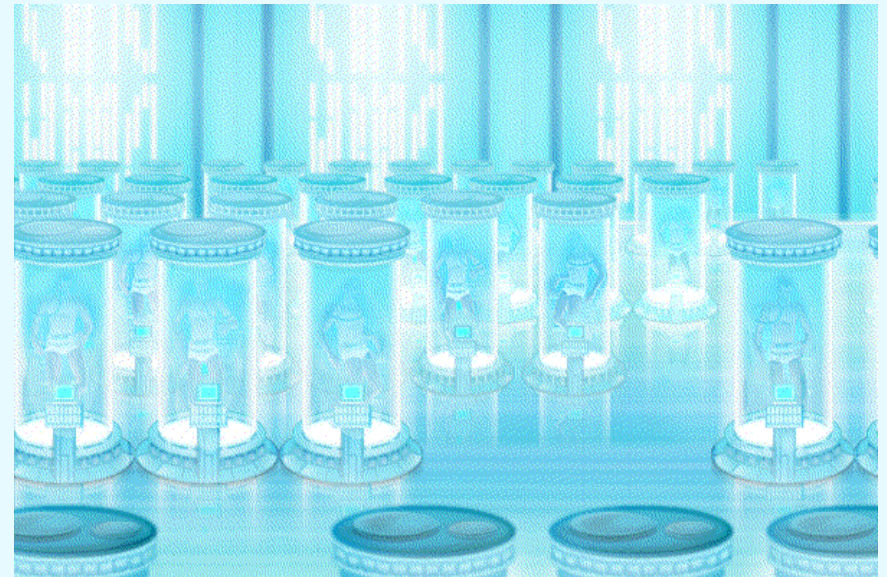
Referencia



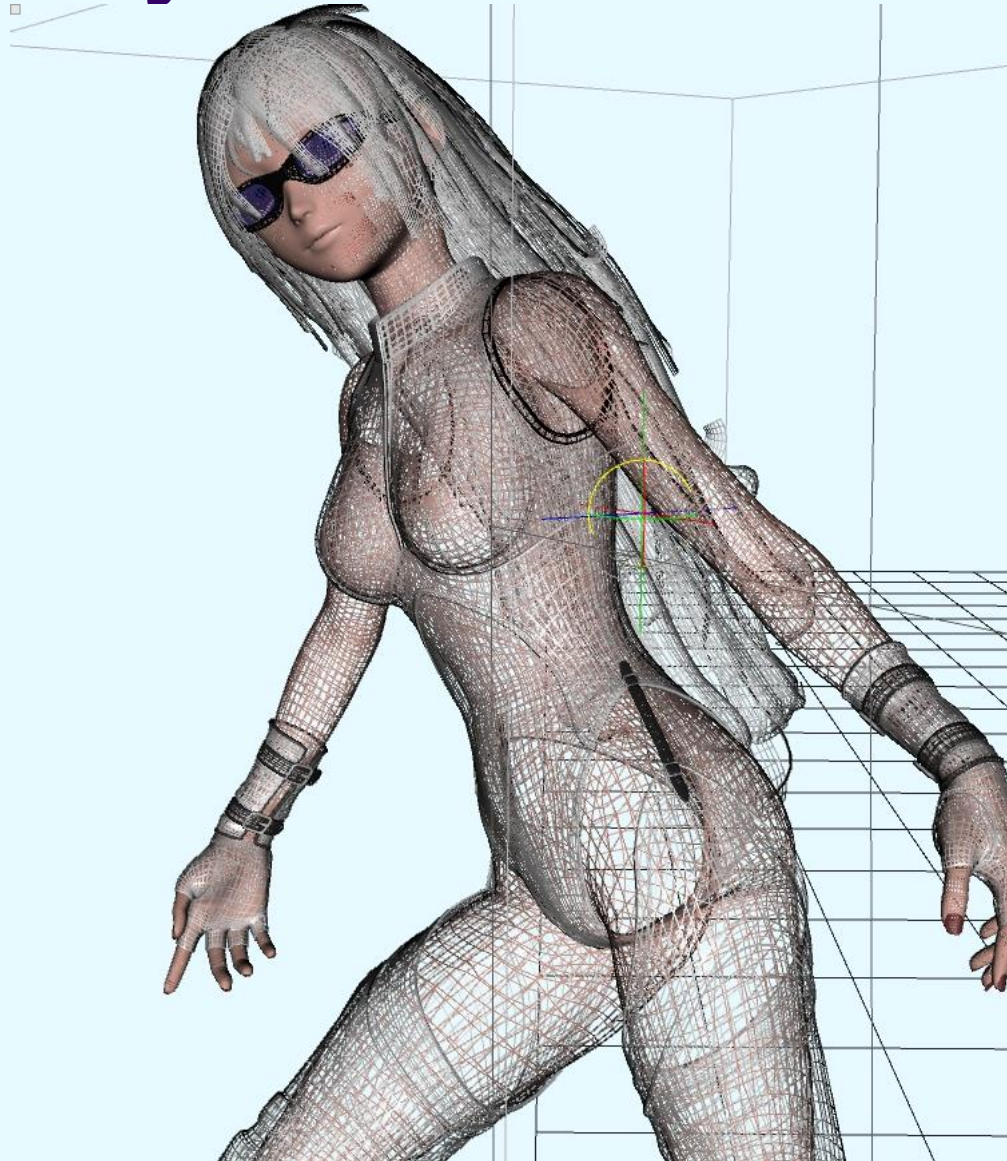
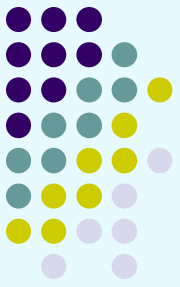
Efecto 3D con movimiento de capas



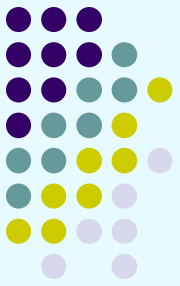
- Se capturan los conceptos básicos de una realidad 3D con perspectiva.
- Se representa en 2D tales conceptos.
- Evitar puntos de fuga.
- Evitar dependencia entre capas.



3 – Modelado y animación 3D



Introducción

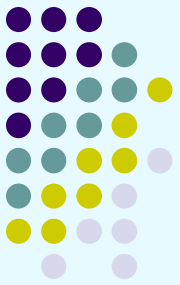


“Un buen plan, violentamente ejecutado ahora, es mejor que un plan perfecto la semana que viene.” - George S. Patton (1885 - 1945)

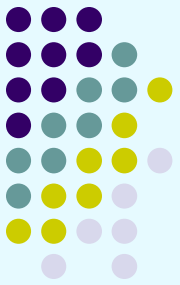
Independientemente de la aplicación a usar el proceso de creación y animación de un modelo tridimensional sigue un plan asociado a su estructura:

- Plan y diseño
- Modelado
- Texturado
- Animación

Plan y Diseño



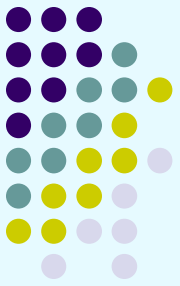
- Plan: lo primero es saber que objeto se modelará. Las técnicas y requerimientos de modelar una persona y un auto son distintas.
 - Ej.: proporciones, la persona debe ser proporcionada con respecto a sí misma, el auto con respecto a una persona que, eventualmente, pueda conducirlo.
- Diseño: conviene hacer un borrador en papel del objeto a modelar, especialmente si es un personaje que aparecerá prominentemente frente al usuario. El borrador debería rebosar detalle, las limitaciones del modelo se encargarán de reducir su complejidad.



Modelado

- El modelado siempre parte de un modelo base. Al modelar se le agrega detalle hasta lograr la terminación deseada.
- El modelo base puede ser tan complejo como se desee. Si se posee una estructura básica humana, se le puede agregar detalle para delinear un personaje en particular.
- Si se empieza de cero, la técnica es la misma. Se utiliza un poliedro como base, por ej. un cubo, y se le agrega complejidad hasta terminar el modelo.

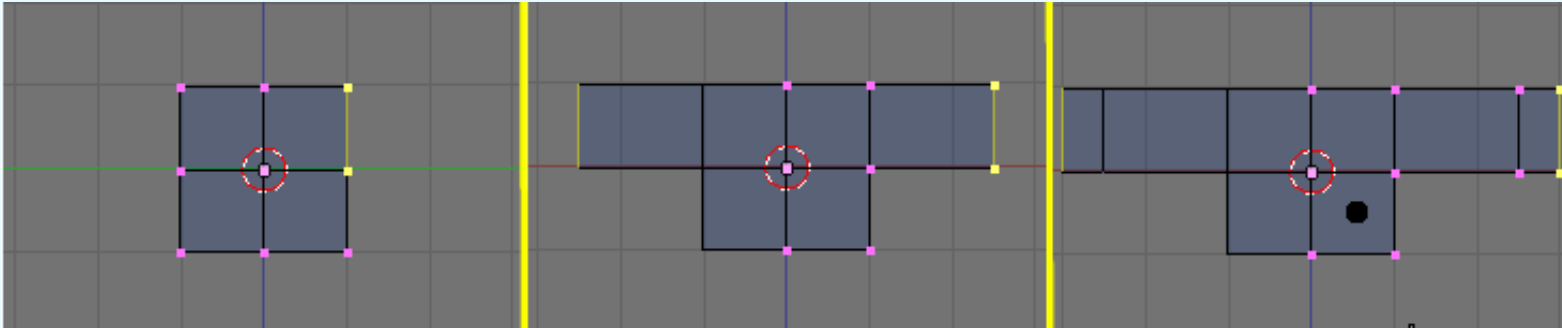
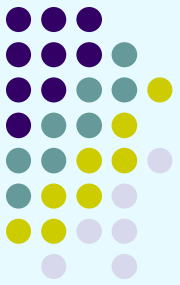
Modelado



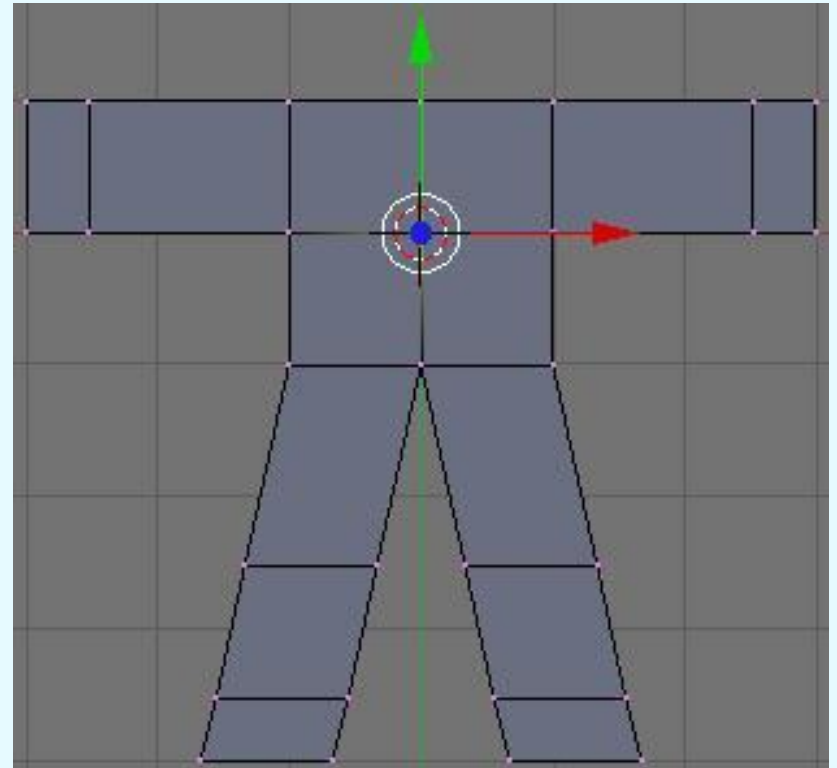
- El plan es modelar un hombre de pan de jengibre, 'gingerbread man', así podremos obviar el proceso el diseño.
- El modelo se puede realizar en cualquier programa con el que se esté familiarizado, en este caso usaremos Blender.
- Los programas de modelado disponen de herramientas para facilitar la realización del modelo.



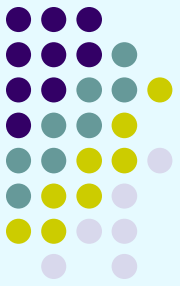
Modelado



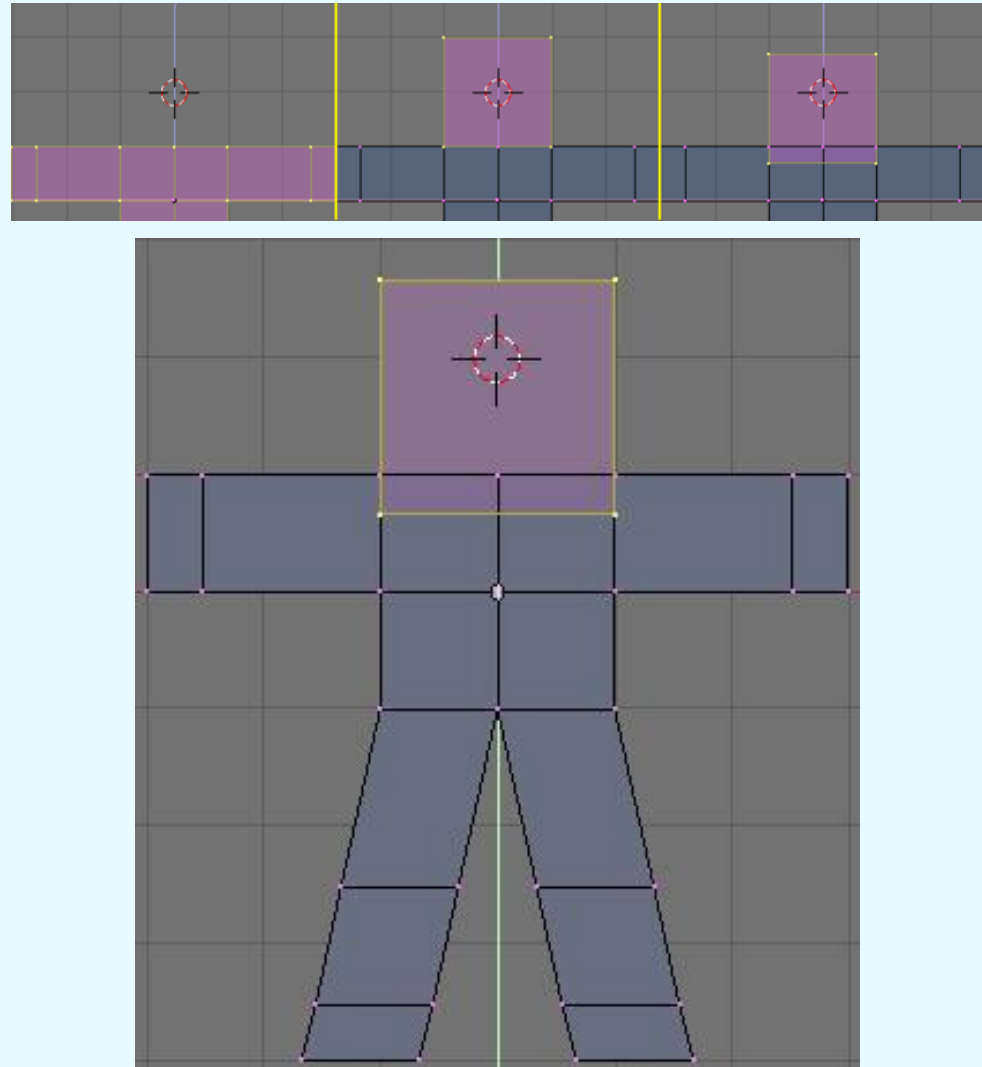
- Para construir el cuerpo usaremos tres herramientas:
 - subdivisión: a partir de un cubo, agregaremos un vértice al medio de cada arista
 - espejo: modelaremos sólo medio cuerpo, la otra mitad se modela automáticamente
 - extrusión: creamos nuevos polígonos estirando caras (en este caso) del original



Modelado

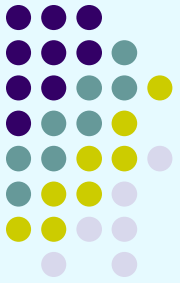
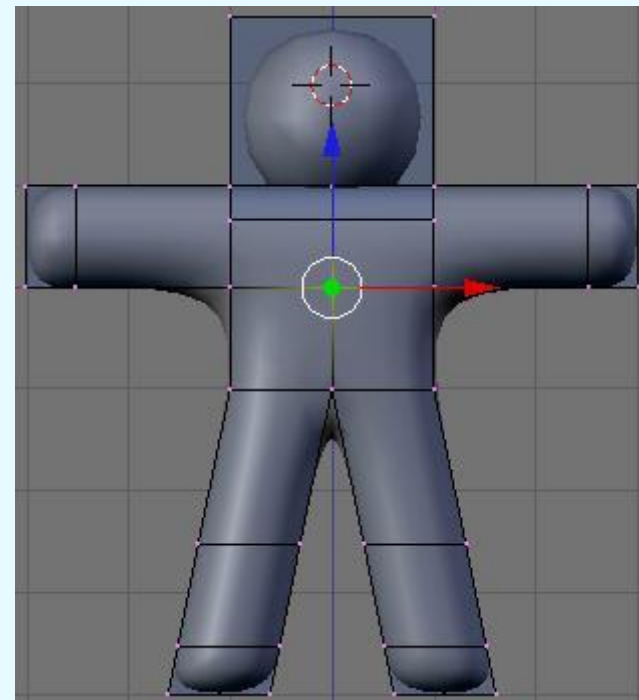
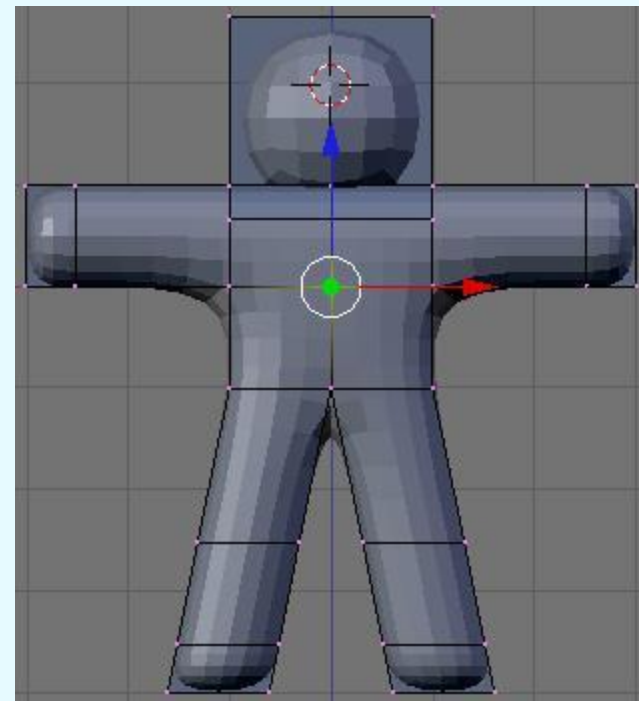


- La cabeza, un simple cubo, debe ubicarse conectada al cuerpo. Para ello se utiliza una traslación. Las traslaciones pueden realizarse fijando un eje.
- En la imagen, el cubo sólo se traslada a través de la línea vertical de color claro.



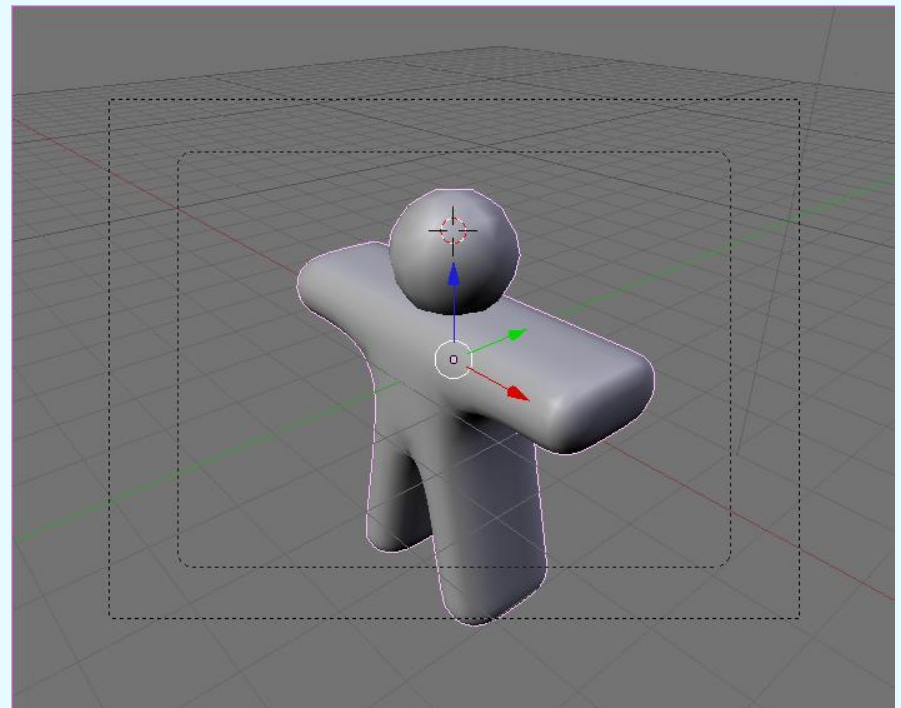
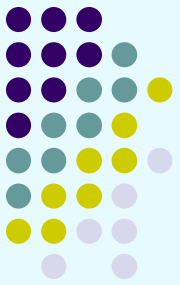
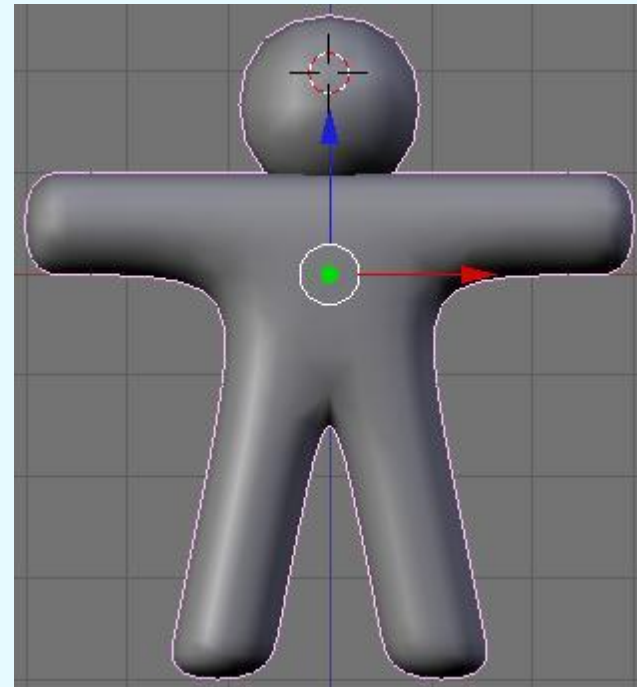
Modelado

- La herramienta subsurf permite refinar el modelo para que se asemeje a una figura humana.
- El programa automáticamente agrega vértices, aumentando la densidad de la malla. La densidad varía según el nivel de detalle que se requiera.



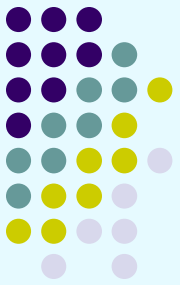
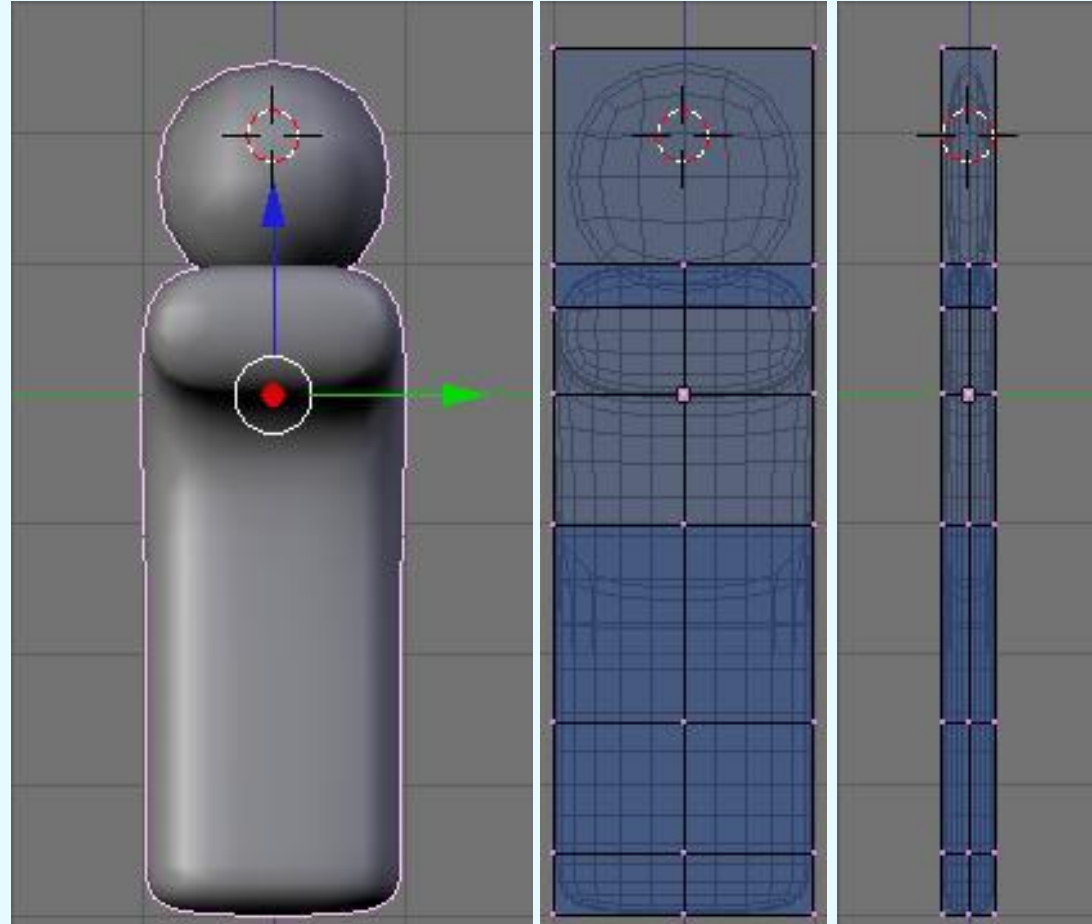
Modelado

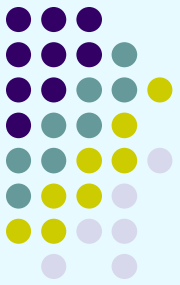
- Una vez aumentada la densidad de la malla, el modelo se puede suavizar. Para ello, el material se asigna como “suave”.
- Puede ser necesario recalcular las normales de las caras si hay defectos en la superficie.



Modelado

- El modelo necesita achatarse si desea aproximarse a una galleta.
- Es necesario escalar horizontalmente el modelo.
Al escalar también se puede fijar un eje sobre el que se realiza la operación.



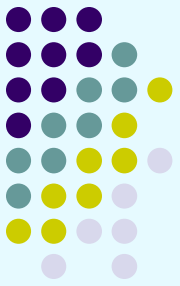


Modelado

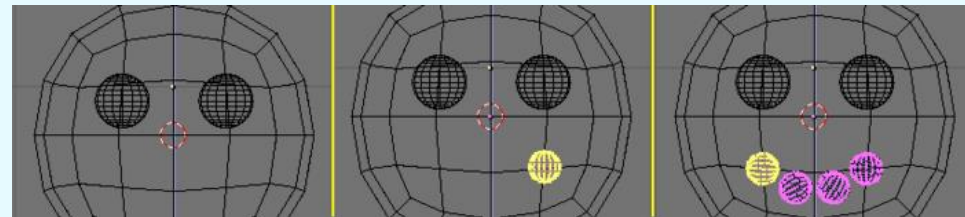
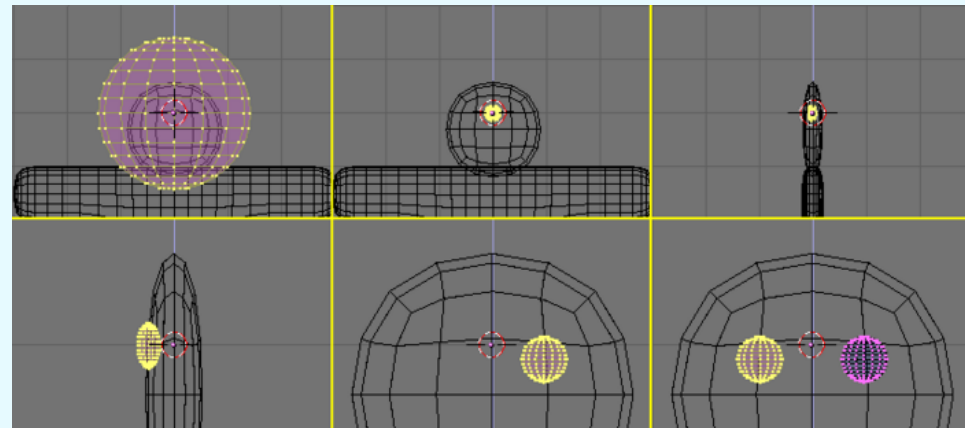
- Para tener una idea de cómo está quedando el modelo se puede armar una escena:
 - agregamos un plano como piso
 - configuramos la cámara para que siga al modelo
 - configuramos un par de luces, spot y hemi, para que iluminen al modelo. Usar dos luces permite simular la luz real con mayor fidelidad.



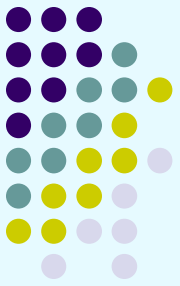
Modelado



- Para dar mayor verosimilitud al modelo agregaremos ojos, boca y botones.
- La herramienta para duplicar permitirá, a partir de un ojo (una esfera escalada horizontalmente), crear el otro ojo y los botones. Las copias se ubican con traslaciones.
- Merece especial mención el uso de la herramienta spin dup (spin duplicate) para, a partir de una esfera, ubicar sus copias automáticamente y generar la boca.

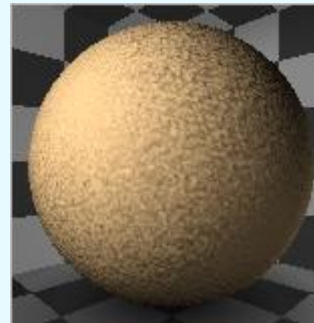


Texturado

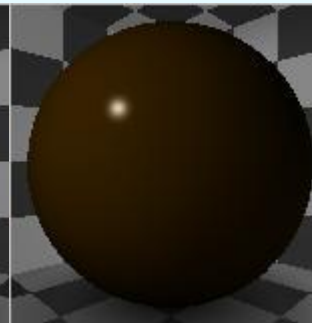


- El uso de materiales permite agregar color al modelo.
- Necesitamos seis materiales:
 - cuerpo de galleta
 - ojos de chocolate
 - boca de azúcar
 - tres botones de azúcar de diferente color
- El uso de 'bump maps' (mapas de altura) permite simular la rugosidad de los elementos.

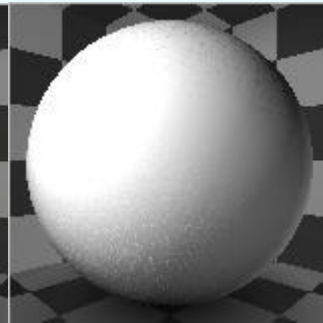
GingerBread



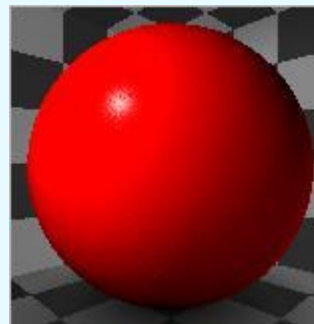
Chocolate



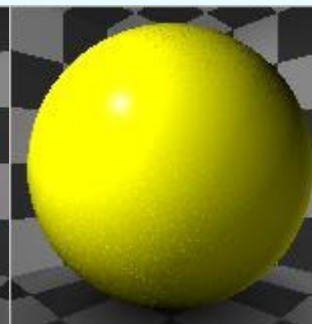
WhiteSugar



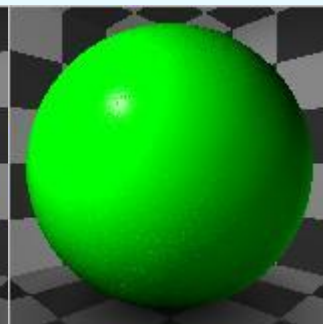
RedSugar



YellowSugar

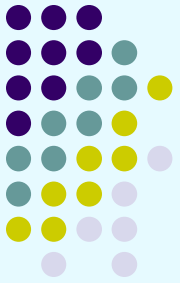
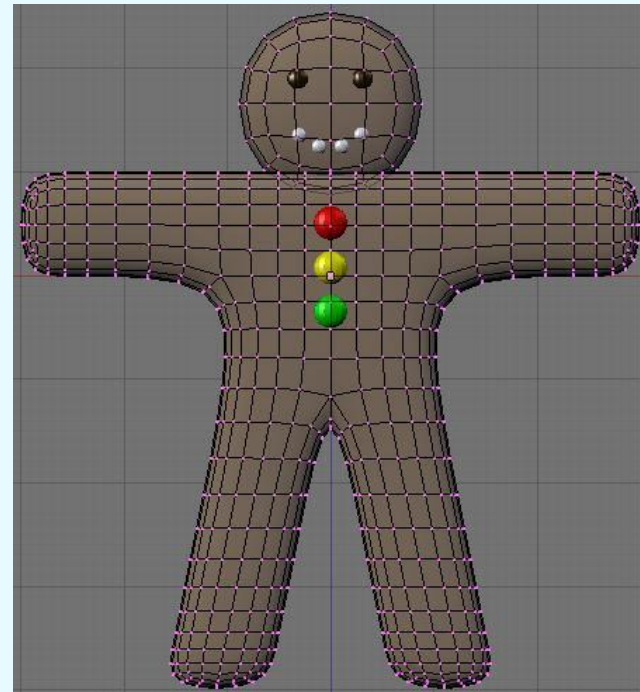
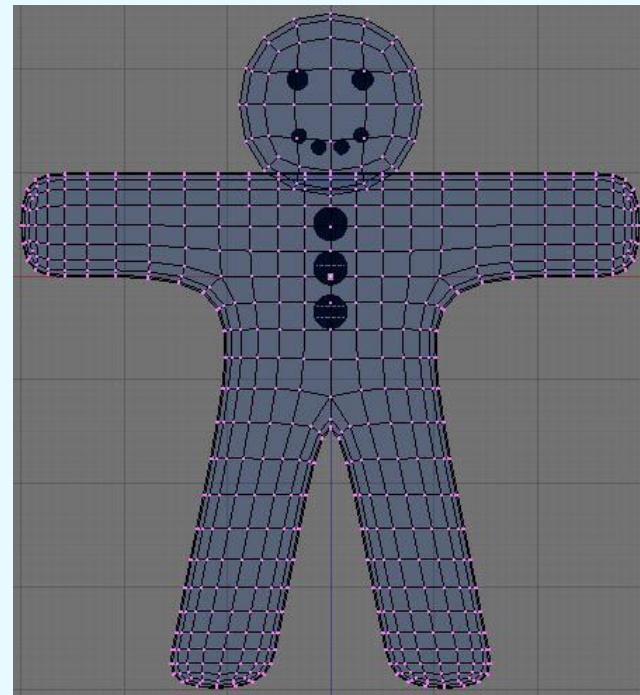


GreenSugar

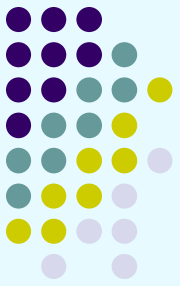


Texturado

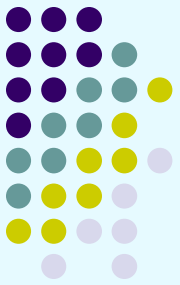
- Los 'bump maps' consisten en texturas que modifican la dirección de la normal.
Utilizamos dos texturas:
 - 'stucci' con ruido suave, para el cuerpo de galleta
 - ruido estándar, para el azúcar.
- Una textura se puede asociar a muchos otros elementos del modelo.
- Para mejorar la apariencia de galleta usaremos la textura de ruido estándar para modificar la reflectividad del cuerpo.



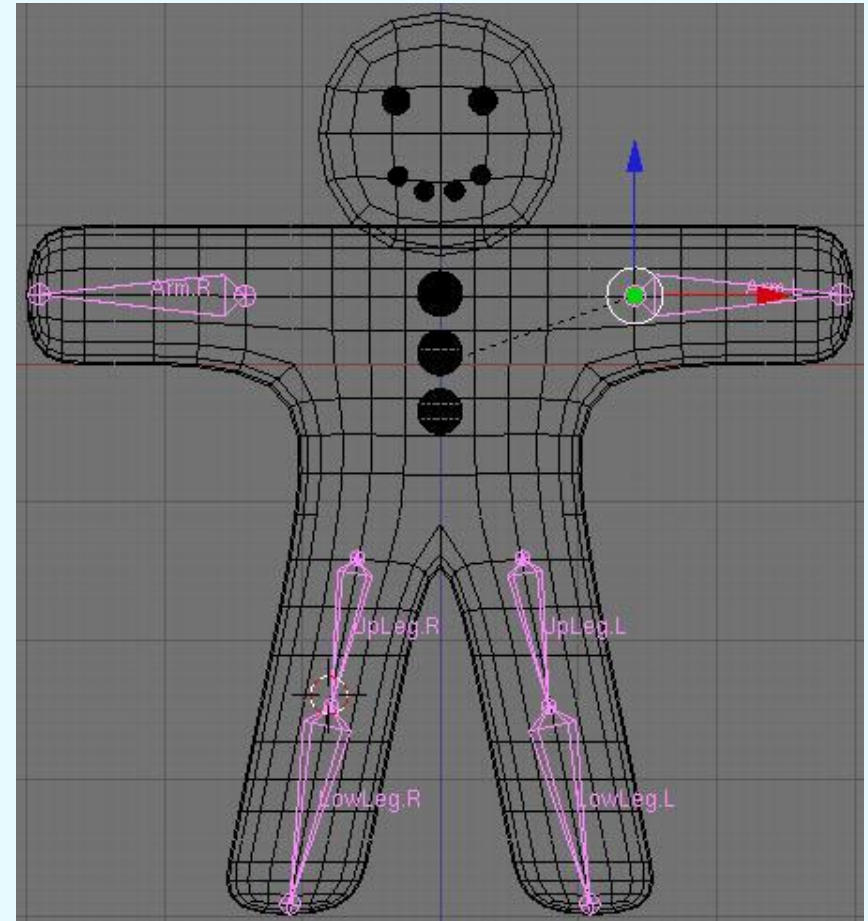
Texturado



Animación

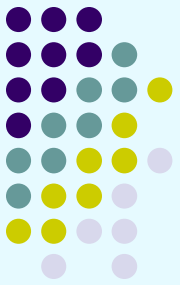
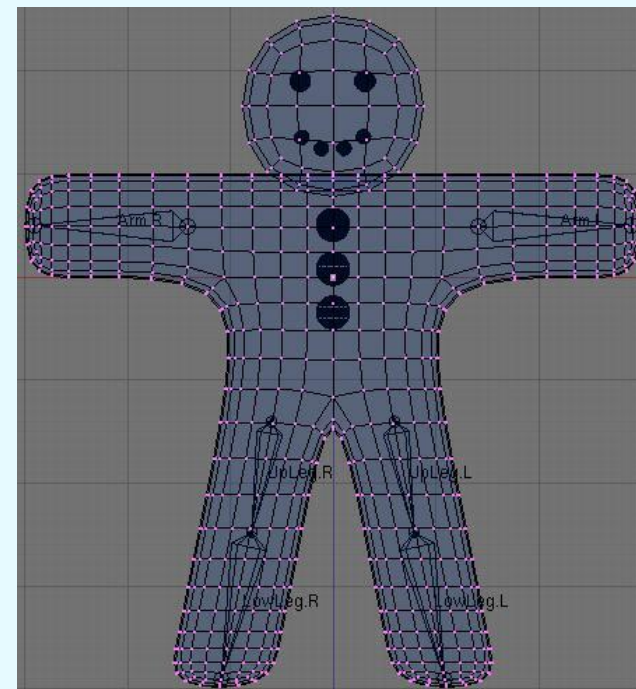
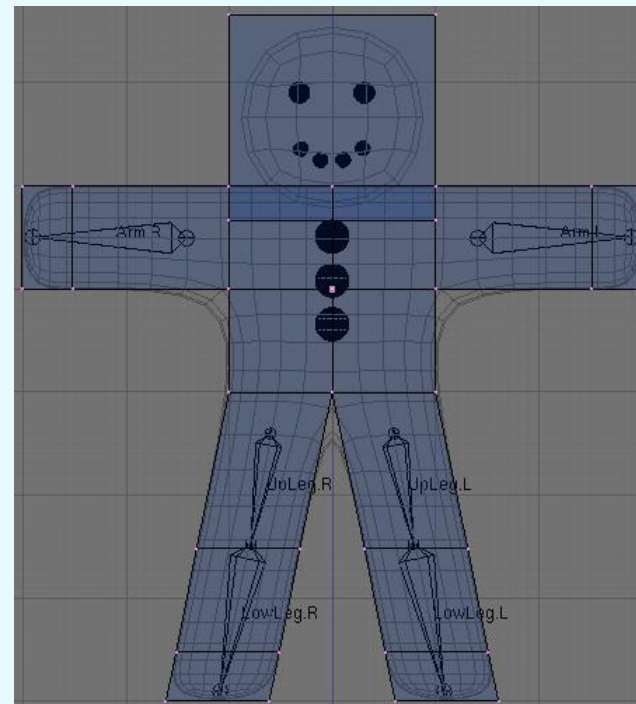


- Para animar el modelo se necesita una armazón o esqueleto.
 - Cada hueso corresponde a un miembro del modelo.
 - Cada pierna es una cadena: dos o más huesos unidos.
- El sufijo '.L', '.R' en el nombre de los huesos facilitará la animación.



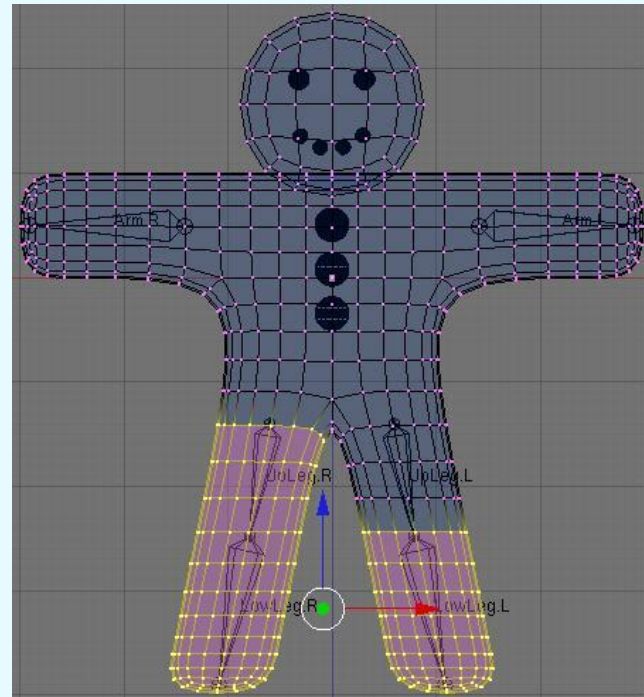
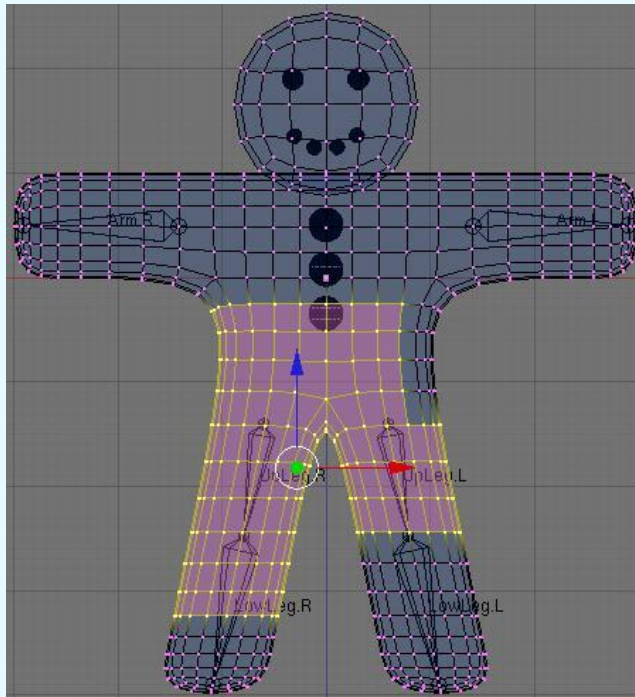
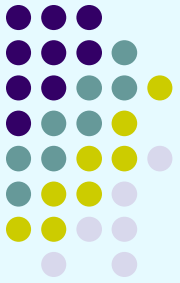
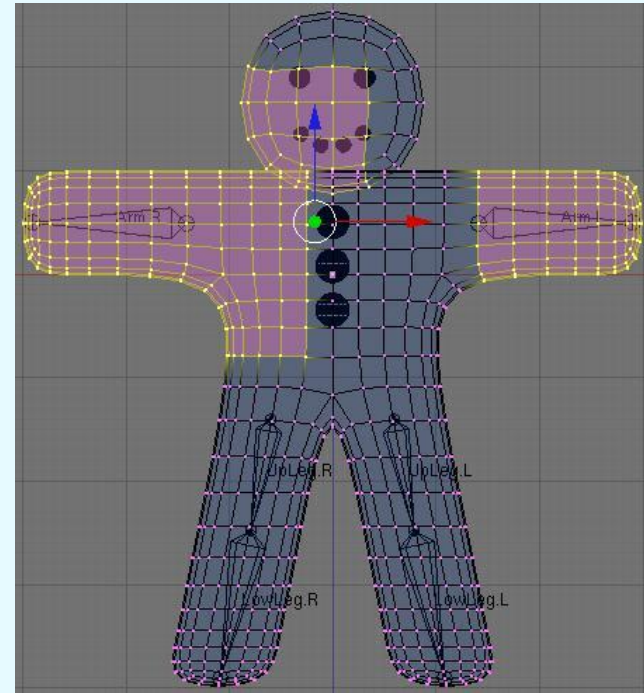
Animación

- Ahora hay que asignar los vértices que cada hueso deformará.
Blender puede hacerlo automáticamente, pero en general, es necesario asignarlos a mano.
- Para mejorar el control de la deformación del modelo es recomendable aplicar el modificador subsurf. Así se puede trabajar sobre los vértices de la malla suavizada.

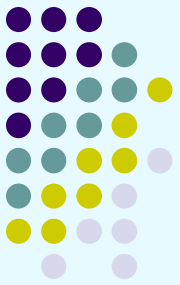


Animación

- La diferencia entre usar la malla simple y la densa se puede ver en los grupos de vértices que están seleccionados.
Izquierda: simple; derecha: densa



Animación



Hay dos formas de animar un modelo:

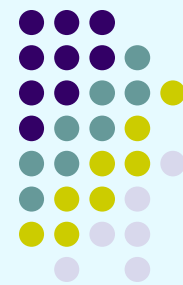
- **Cinemática inversa**

- Se define la posición del último hueso en la cadena, conocido como 'efectuador final'.
- El resto de los huesos asume una posición que evite interrupciones en la cadena.
- Las posiciones son automáticamente calculadas por el 'IK solver'.
- Esto permite ubicar las manos y pies con mayor precisión.

- **Cinemática directa**

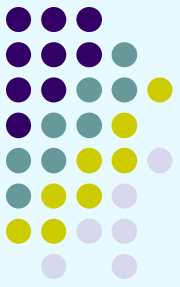
- Se mueve el primer hueso de la cadena, los demás siguen su movimiento.
- Los huesos, a excepción del primero, sólo pueden ser rotados con respecto al anterior.
- Es fácil de seguir pero complica el ubicar el último hueso de la cadena.

Animación

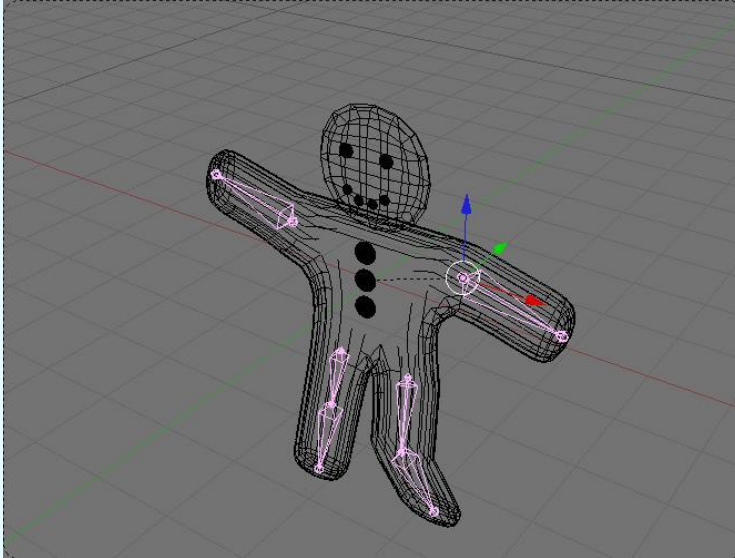


- Usaremos cinemática directa para lograr que el modelo camine.
- Definiremos cinco cuadros de la animación, Blender automáticamente rellenará los demás cuadros para lograr una animación fluida.
- Para crear los cuadros inversos, Blender cambia las rotaciones de los huesos con sufijo '.L' por '.R' y viceversa.
- Los cuadros serán:
 - 1: dando un paso con la pierna izquierda
 - 11: apoyando los dos pies, izquierdo adelante
 - 21: dando un paso con la pierna derecha, inverso del cuadro 1
 - 31: apoyando los dos pies, derecho adelante, inverso del cuadro 11
 - 41: fin del ciclo, igual a 1

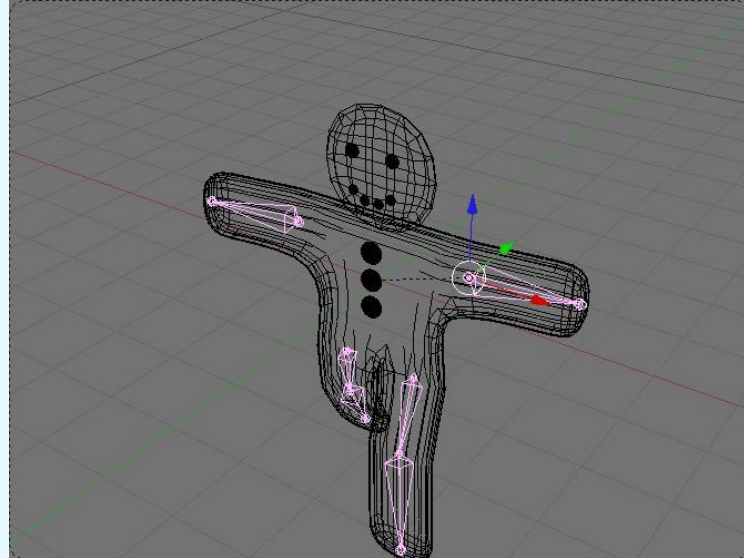
Animación



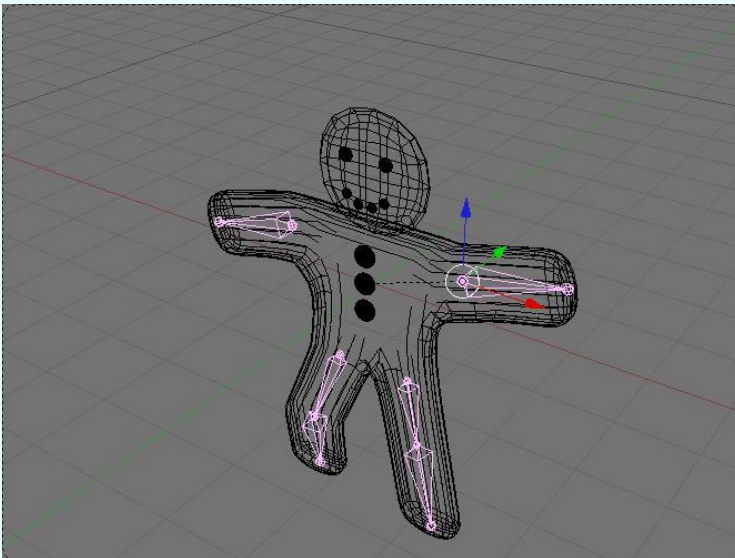
cuadro
1, 41



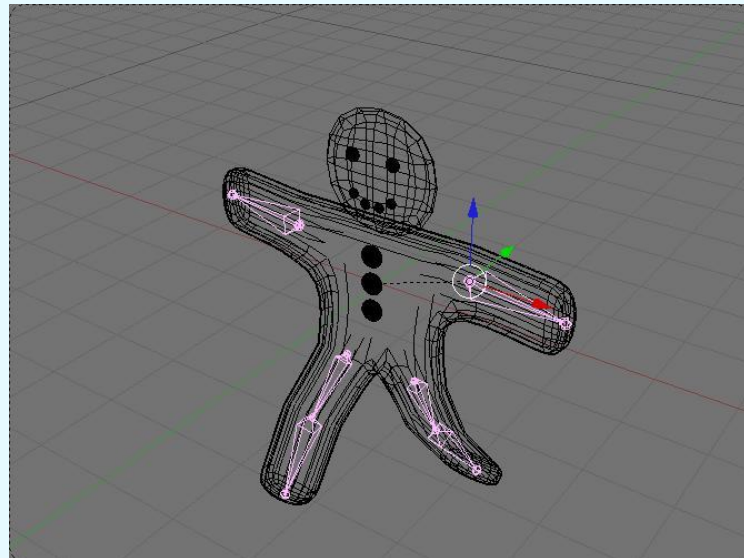
cuadro
11



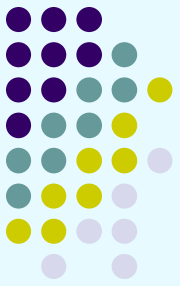
cuadro
21



cuadro
31



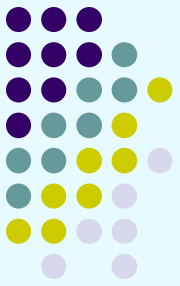
Animación



- Falta agregar una traslación a los cuadros de animación para que pueda desplazarse.
- Si deseáramos realizar un video con Blender, es imprescindible para mostrar el movimiento.
- Si el modelo será exportado a un motor gráfico, este puede encargarse de desplazar el modelo y sólo se necesita la animación.



Animación



cuadro
1, 41



cuadro
11



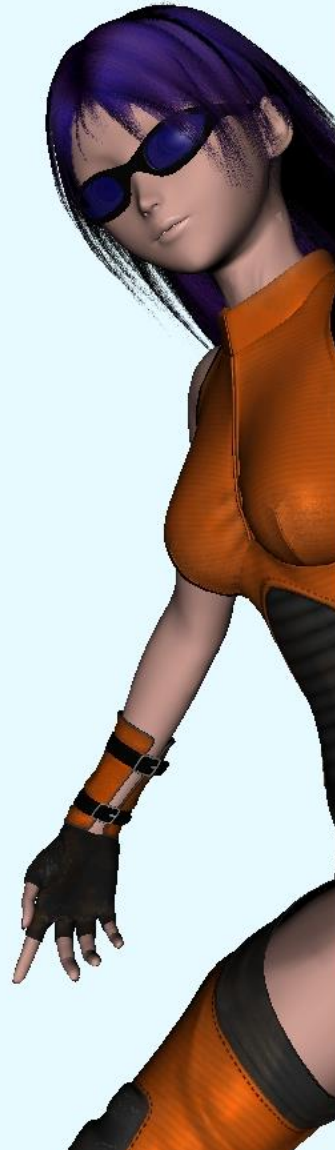
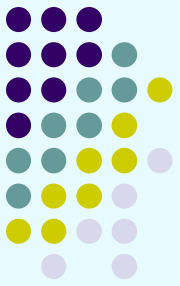
cuadro
21

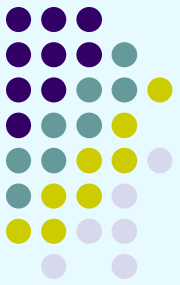


cuadro
31



4 – Uso de modelos 3D





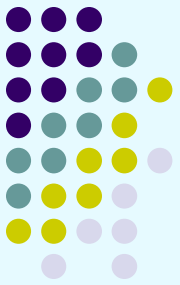
Introducción

“Finalmente aprendí que quiere decir ‘compatible hacía arriba’.
Significa que podemos conservar todos nuestros viejos errores” –
Dennie van Tassel

Para utilizar un modelo se necesita elegir una API (Application
Programable Interface) y un formato para exportar el modelo.

- ¿Qué es OpenGL?
- ¿Qué es un 3ds?
- ¿Cómo cargar un 3ds en OpenGL?

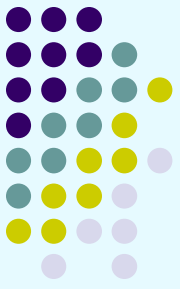
¿Qué es OpenGL?



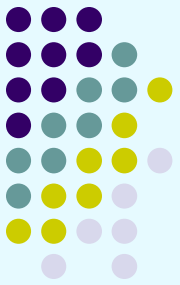
- OpenGL es una interface de software para hardware gráfico independiente de la plataforma.
- Consiste en 150 comandos diferentes para especificar los objetos y operaciones necesarias para crear aplicaciones 3D interactivas.



¿Qué es OpenGL?



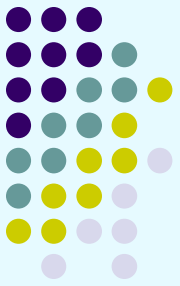
- Se divide en tres partes funcionales:
 - La biblioteca OpenGL, que proporciona todo lo necesario para acceder a las funciones de dibujo de OpenGL.
 - La biblioteca GLU (OpenGL Utility Library), una librería de utilidades que proporciona acceso rápido a algunas de las funciones más comunes de OpenGL, a través de comandos de nivel inferior de OpenGL.
 - GLX (OpenGL Extension to the X Window System) proporciona un acceso a OpenGL para poder interactuar con un sistema de ventanas X Window, y está incluido en la propia implementación de OpenGL (su equivalente en Windows es la librería WGL, externa a la implementación de OpenGL).



¿Qué es OpenGL?

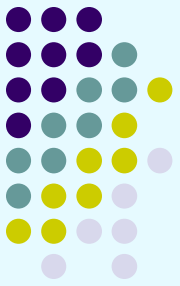
- Cada objeto esta especificado por vértices:
 - **glVertex3f (2.0, 4.1, 6.0);**
// especifica un vértice en la coord. x,y,z (2.0, 4.1, 6.0)
// “3f” significa 3 coordenadas flotantes.
 - **glVertex2i (4, 5);** // 2 enteros para x, y. z = 0.
 - **glVertex3fv (vector);** // float vector[3] = {5.0, 3.2, 5.0};
- Color actual afecta todos los vértices
 - **glColor3f (0.0, 0.5, 1.0);**
- Los vértices se especifican solamente entre glBegin(*mode*) y glEnd(), generalmente se especifican en un orden de agujas del reloj.
 - **glBegin (GL_TRIANGLES);**
 glVertex2i (0, 0);
 glVertex2i (2, 0);
 glVertex2i (1, 1);
 glEnd();

¿Qué es un 3ds?



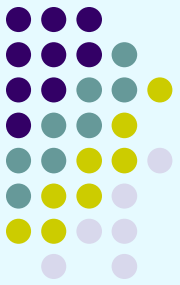
- Un archivo 3ds contiene la información para describir los detalles de la escena que compone.
- Esta formado por una serie de bloques llamados 'chunks' que contienen todo lo necesario para describir la escena:
 - Nombre de cada objeto.
 - Coordenadas de los vértices
 - Coordenadas para mapeo.
 - Lista de polígonos
 - Colores de las caras
 - Keyframes de la animación

¿Qué es un 3ds?



- Un chunk esta formado por 4 campos:
 - **Identificador:** Número hexadecimal de 2 bytes de largo que identifica al chunk.
 - **Largo del chunk:** Número de 4 bytes que es la suma del largo del chunk y de los sub-chunks que contiene.
 - **Chunk data:** Contiene la informacion de la escena.
 - **Sub-chunks**

¿Qué es un 3ds?



MAIN CHUNK 0x4D4D

3D EDITOR CHUNK 0x3D3D

OBJECT BLOCK 0x4000

TRIANGULAR MESH 0x4100

VERTICES LIST 0x4110

FACES DESCRIPTION 0x4120

FACES MATERIAL 0x4130

MAPPING COORDINATES LIST 0x4140

SMOOTHING GROUP LIST 0x4150

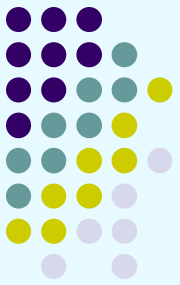
LOCAL COORDINATES SYSTEM 0x4160

LIGHT 0x4600

SPOTLIGHT 0x4610

CAMERA 0x4700

¿Qué es un 3ds?



MATERIAL BLOCK 0xAFFF

MATERIAL NAME 0xA000

AMBIENT COLOR 0xA010

DIFFUSE COLOR 0xA020

SPECULAR COLOR 0xA030

TEXTURE MAP 1 0xA200

BUMP MAP 0xA230

REFLECTION MAP 0xA220

[SUB CHUNKS FOR EACH MAP]

MAPPING FILENAME 0xA300

MAPPING PARAMETERS 0xA351

KEYFRAMER CHUNK 0xB000

MESH INFORMATION BLOCK 0xB002

SPOT LIGHT INFORMATION BLOCK 0xB007

FRAMES (START AND END) 0xB008

OBJECT NAME 0xB010

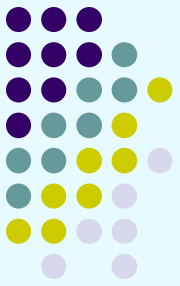
OBJECT PIVOT POINT 0xB013

POSITION TRACK 0xB020

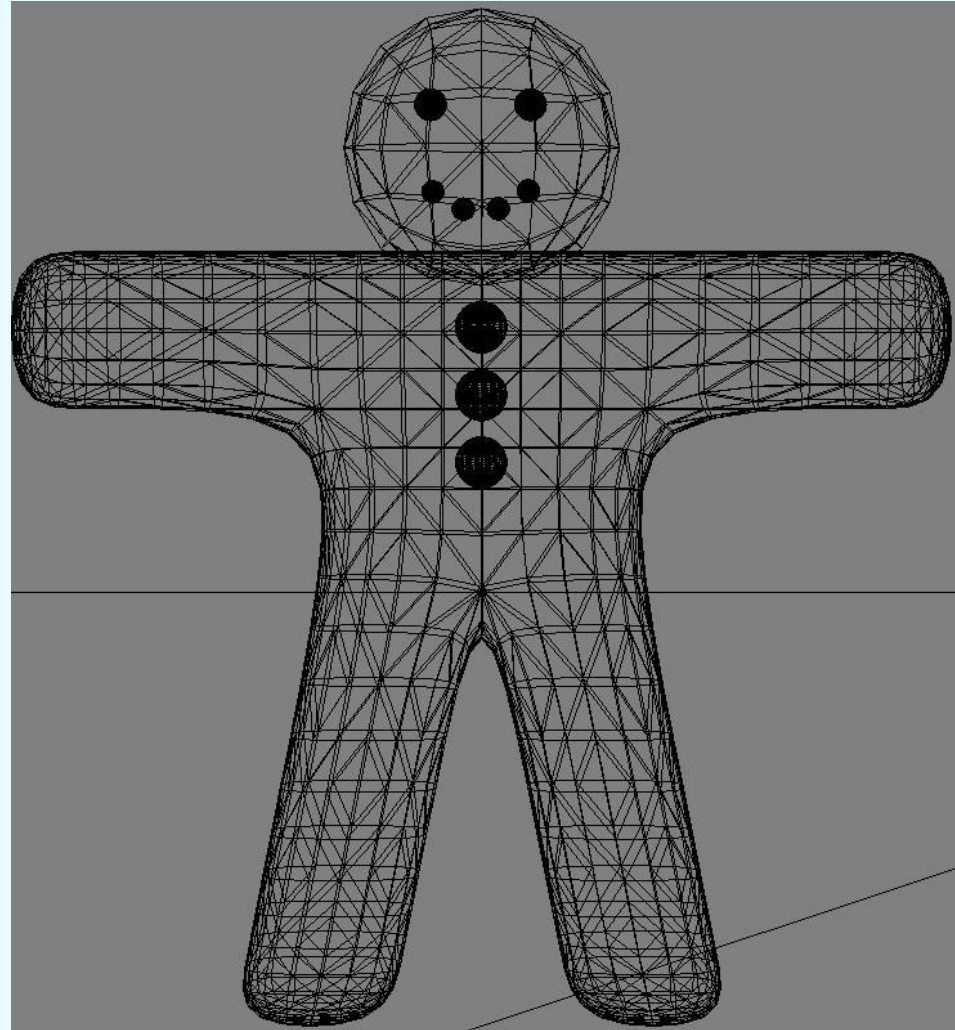
ROTATION TRACK 0xB021

SCALE TRACK 0xB022

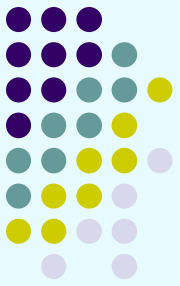
¿Cómo cargar un 3ds en OpenGL?



- Implementar un loop que procese todo el archivo.
- En cada iteración leer el id del chunk y el largo.
- Analizar el contenido del chunk.
- Saltear los sub-chunks que no necesito leer.

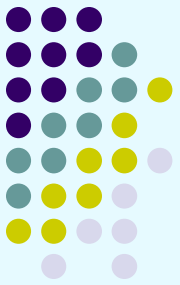


¿Cómo cargar un 3ds en OpenGL?



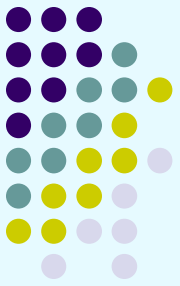
- Como primer paso debo definir la cantidad de vértices y polígonos que se van a manejar:
 - `#define MAX_VERTICES 8000`
 - `#define MAX_POLYGONS 8000`

¿Cómo cargar un 3ds en OpenGL?



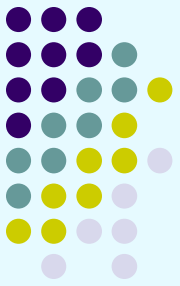
- Luego debo crear la rutina para cargar el 3ds:
- `char Load3DS (obj_type_ptr p_object, char *p_filename)`
`{`
 `int i;`
 `FILE *l_file;`
 `unsigned short l_chunk_id;`
 `unsigned int l_chunk_length;`
 `unsigned char l_char;`
 `unsigned short l_qty;`
 `unsigned short l_face_flags;`
`}`
- Esto acepta dos parametros: puntero a la estructura de datos y nombre del archivo.

¿Cómo cargar un 3ds en OpenGL?



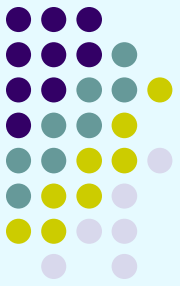
- El paso siguiente es abrir el archivo y leerlo.
- Según el identificador leído del bloque de datos se leerá de diferente forma.
- Al leer cada chunk se debe leer los datos asociados a el y la información que brindan los chunks hijos.

¿Cómo cargar un 3ds en OpenGL?

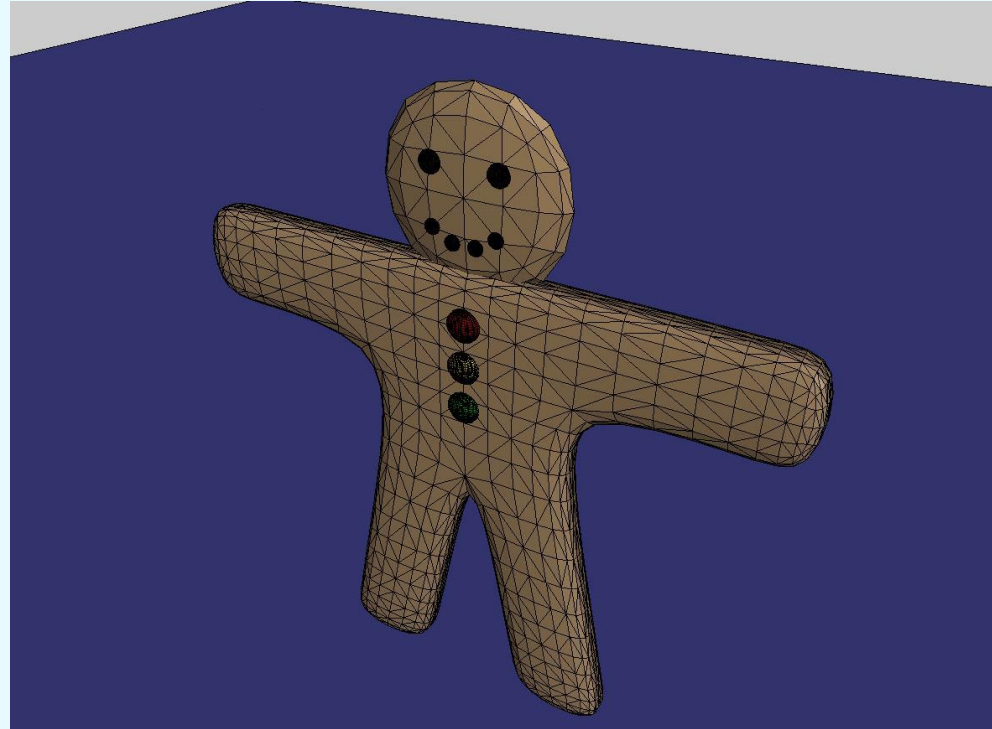


- A continuación se listan algunos de los chunks importantes para la lectura del 3DS:
 - OBJECT BLOCK contiene el nombre del objeto.
 - VERTICESLIST contiene todos los vertices del modelo.
 - FACESDESCRIPTION contiene la lista de polígonos.
 - MAPPING COORDINATES LIST contiene las coordenadas. .

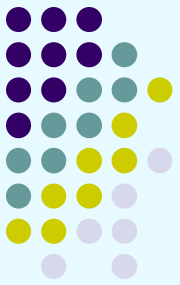
¿Cómo cargar un 3ds en OpenGL?



- Luego de leer todos los chunks del 3ds se cierra el archivo y el modelo quedara cargado.
- El paso final es utilizar la rutina para cargar el modelo:
 - Load3DS
(`&object, "spaceship.3ds"`);
- Y por último en la función display cargar la información del objeto.

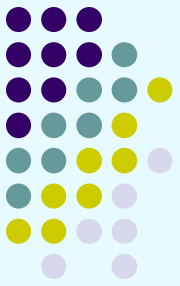


¿Cómo cargar un 3ds en OpenGL?



- Ejemplo de carga de un vértice:
 - // Texture coordinates of the first vertex
`glTexCoord2f(object.mapcoord[object.polygon[l_index].a].u,
object.mapcoord[object.polygon[l_index].a].v);`
 - // Coordinates of the first vertex
`glVertex3f(object.vertex[object.polygon[l_index].a].x,
object.vertex[object.polygon[l_index].a].y, object.vertex[
object.polygon[l_index].a].z);`

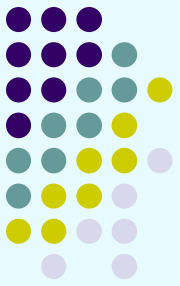
¿Cómo cargar un 3ds en OpenGL?

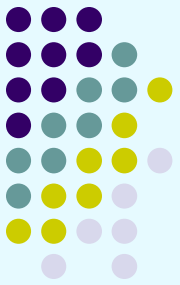


- Finalmente, para animar una figura se debe:
 - Actualizar los datos de la figura
 - Borrar la pantalla
 - Dibujar la figura
 - Repetir



¿Preguntas?





Bibliografía

- Conceptos básicos del dibujo: <http://www.dibunet.com/indicetutorial.php>
- Conceptos básicos de Animación: <http://animation.about.com/od/otherusefultutorials/u/animationbasics.htm>
- Street Fighter Sprites: <http://www.zweifuss.com/>
- Clone Wars Game: http://theclonewars.cartoonnetwork.com/games/game_02.html
- Modeling a Character in 3DS Max, Paul Steed
- Blender Manual: <http://wiki.blender.org/index.php/Manual/Manual>
- Blender 2.47: www.blender.org
- 3D Art Tutorial: <http://www.3dfantasies.info/res.html>
- DAZ Studio: www.daz3d.com
- OpenGL Documentation: <http://www.opengl.org/documentation>
- <http://www.exa.unicen.edu.ar/catedras/visdat2/tutoriales.html>
- Ingeniería de Sistemas y Computación Universidad Tecnológica de Pereira
- Revista Electrónica Neoteo: <http://www.neoteo.com/>
- Tutorial de como cargar un archivo 3DS: http://www.spacesimulator.net/tut4_3dsloader.html