ANIMACIÓN

Visualización II - 2010

Animación y Simulación

- Animación

  “Hacer que los objetos cambien su comportamiento de acuerdo a acciones prefijadas”
- Simulación

  “Predecir como los objetos cambian su comportamiento de acuerdo a leyes físicas”
- **Que necesito para programar ?**
- **Lenguaje con Eventos de tiempo**

Contenido

- Sistemas Dinámicos:
  - Animaciones procedurales
  - Partículas
  - Simulación
- Keyframing
- Modelos articulados y cinemática:
  - Jerarquías
  - Forward Kinematics
  - Inverse Kinematics
- Algunas aplicaciones

Ciclo de render

- Por cada ciclo de reloj(*fps*)
  - Recorrer geometría
  - Aplicar transformaciones del objeto (traslación, escala)
  - Aplicar transformaciones de la vista (perspectiva, punto de vista)
  - Aplicar modelo de luz
  - Dibujar en pantalla

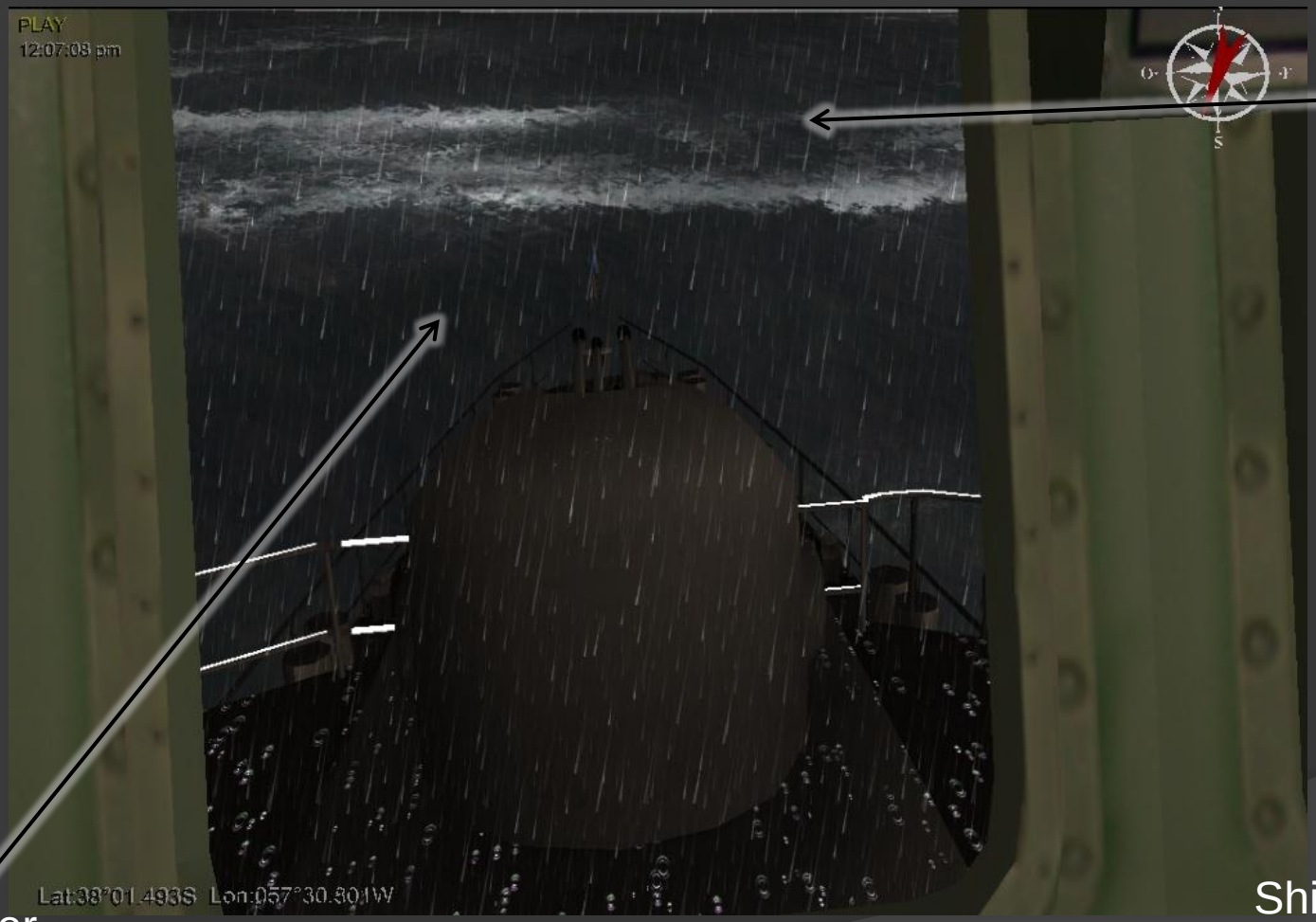
La geometría va a cambiar

Animación

Caso 1

Animación

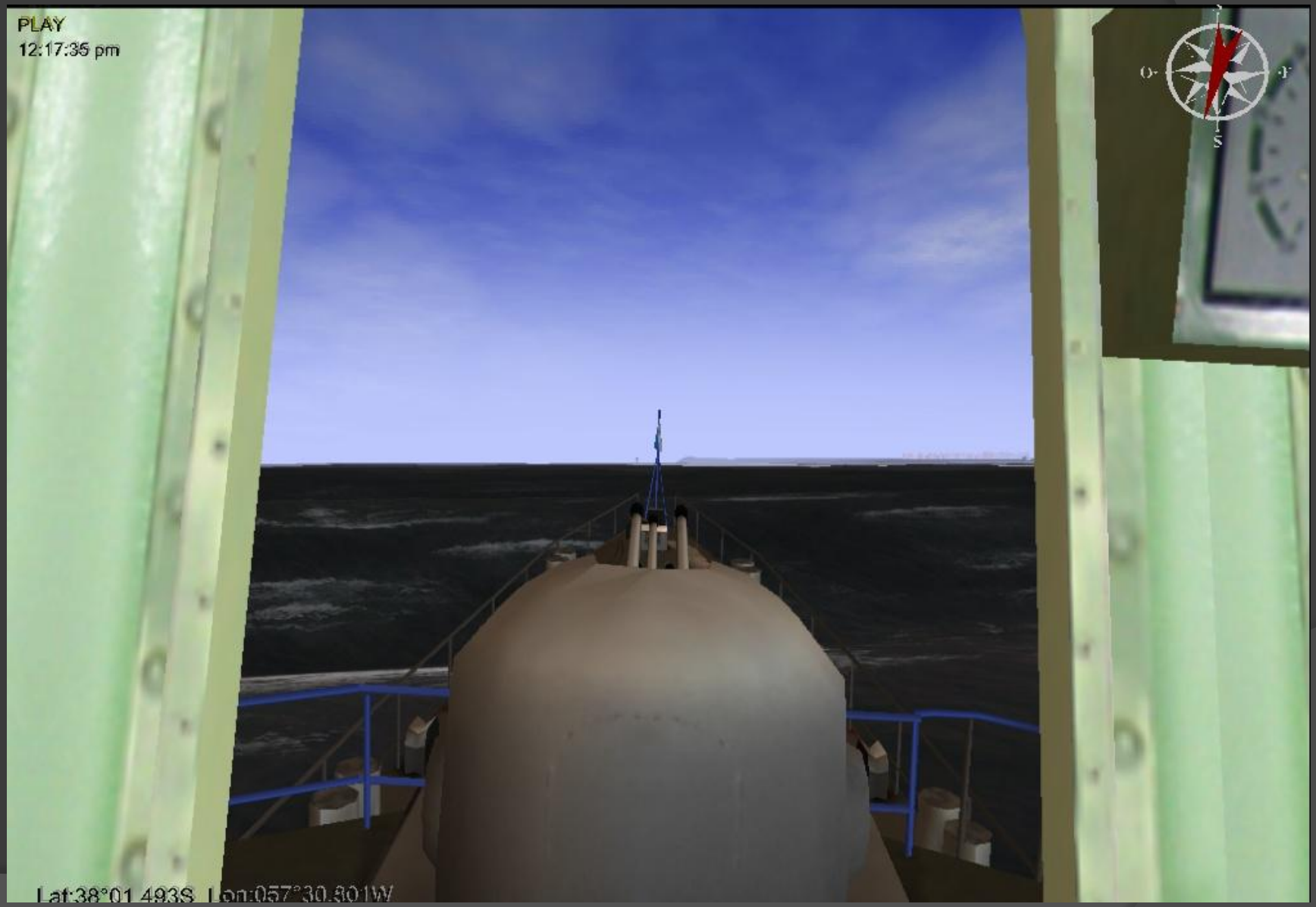
Caso 1



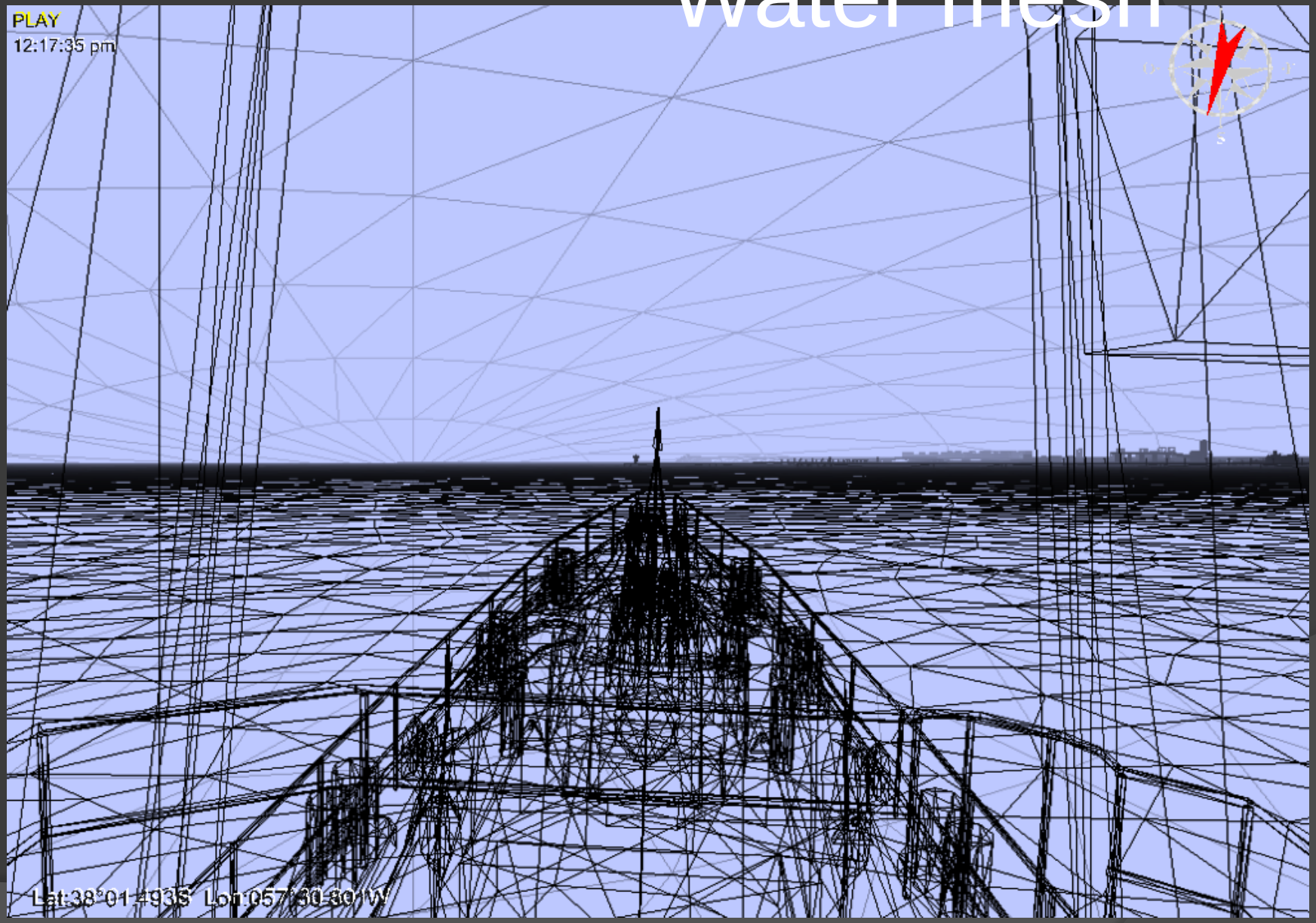
Rain

Water

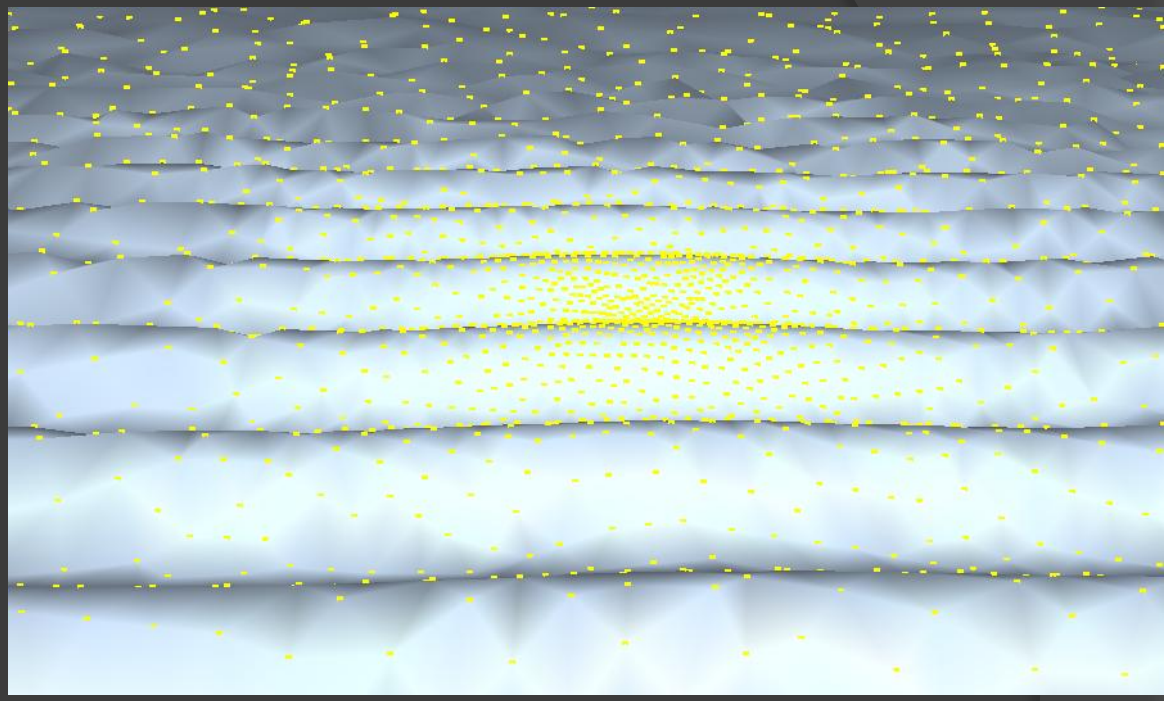
Ship

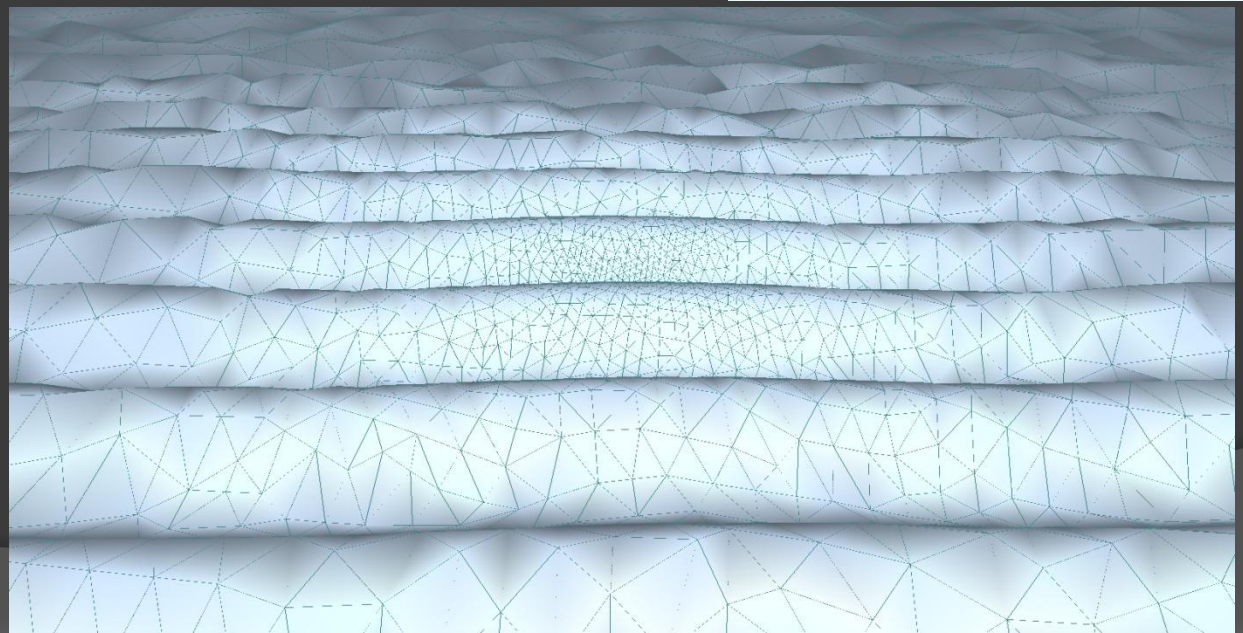

PLAY
12:17:35 pm
Lat:38°01.493S Lon:057°30.801W

Water mesh



Water mesh





Animación procedural

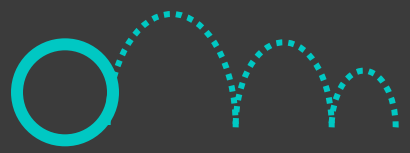
- Describe el movimiento de forma algorítmica
- la animación es una función con el ***tiempo*** como parámetro
- **Animar el agua**

Por cada vértice v

*v.Y =(Amp1 * sin(frec1 * (v.x + Time))*

+

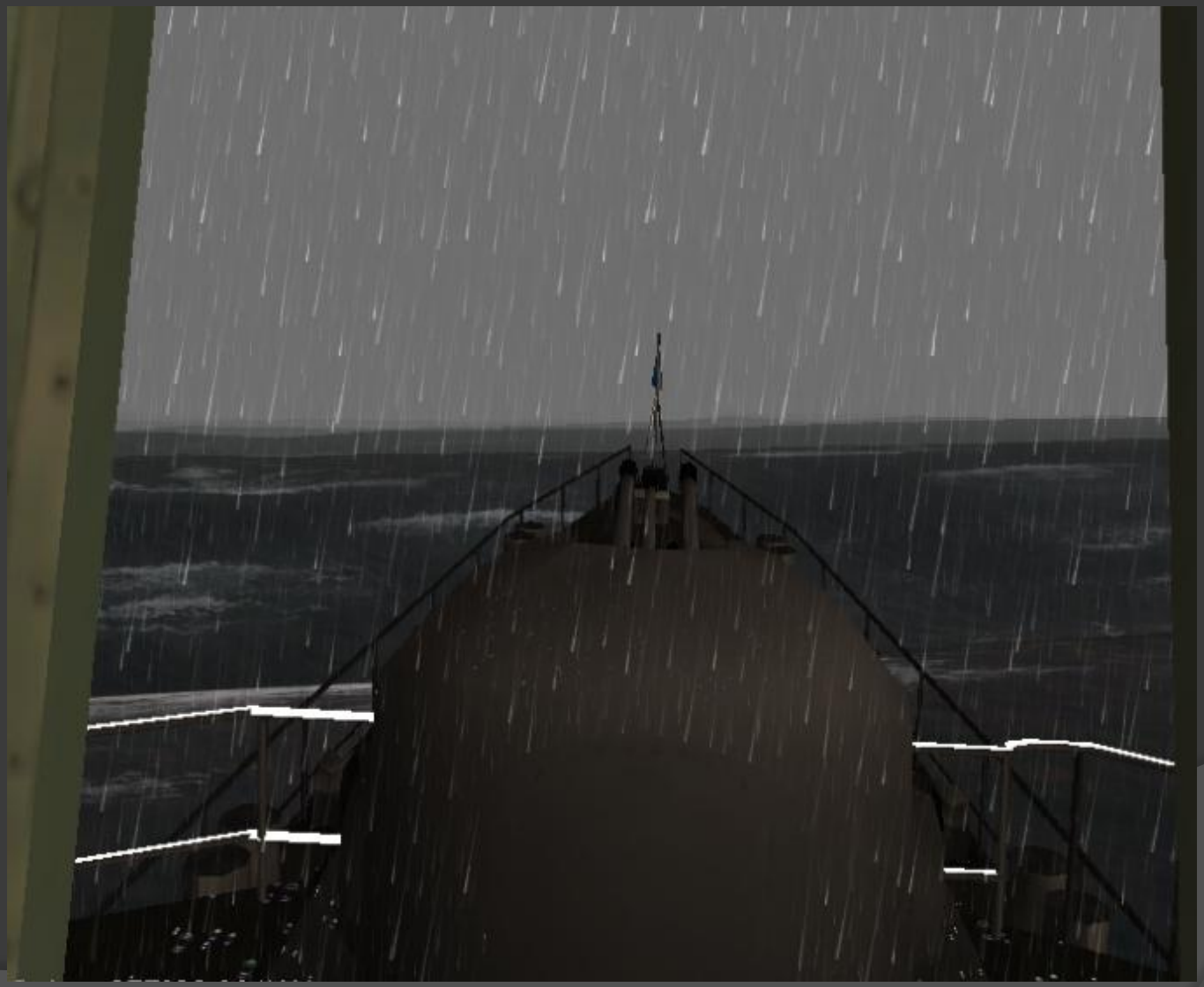
*Amp2 * cos(estadoDeMar.frec2 * v.z);*



Problemas ??

Interacción

Rain

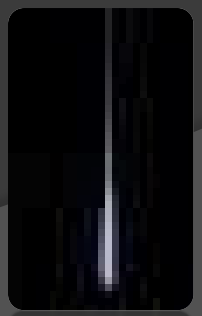


Rain

Rain

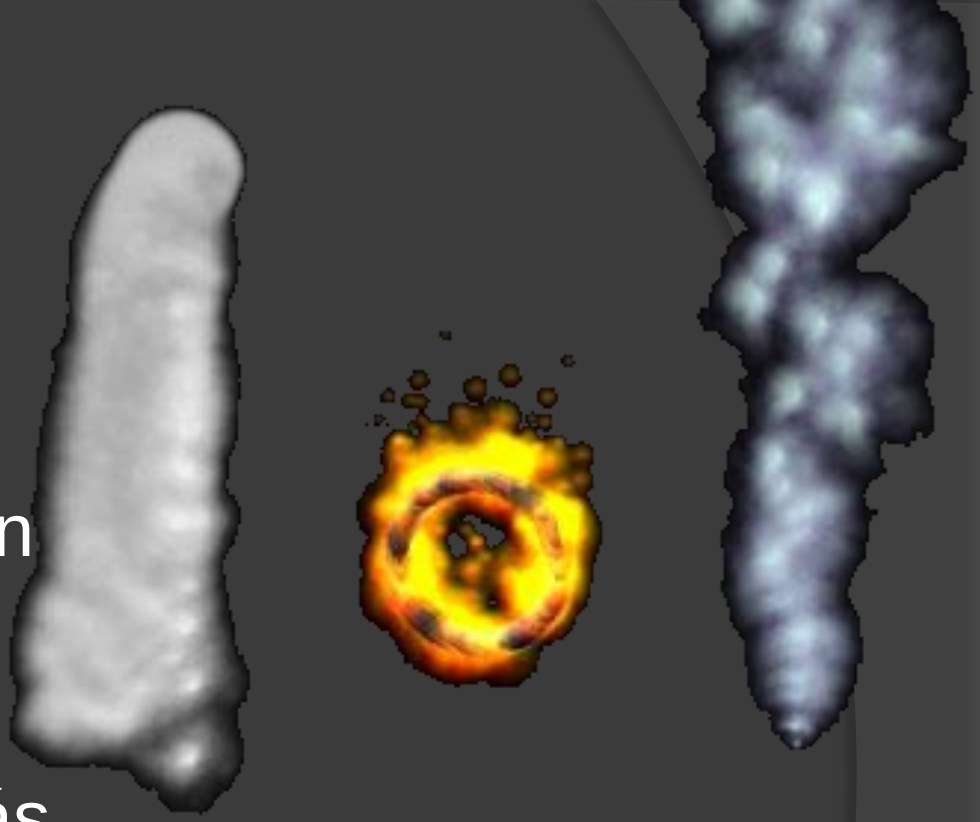
Sistemas de Partículas

- Las partículas son objetos básicos(rectángulos)
- Una partícula tiene:
  - Una posición y un tiempo de vida
  - Una regla de movimiento (física)
  - Características para ***render*** (textura, transparencia)
- En cada instante de tiempo
- $P_{pos} = P_{pos} + P_{vel} * Time$
- $P_{VEL} = P_{VEL} + P_{AC} * Time$



Aplicaciones

- Variando los parámetros, se pueden crear muchos efectos
- Para dibujar una partícula la técnica más común es *billboarding*

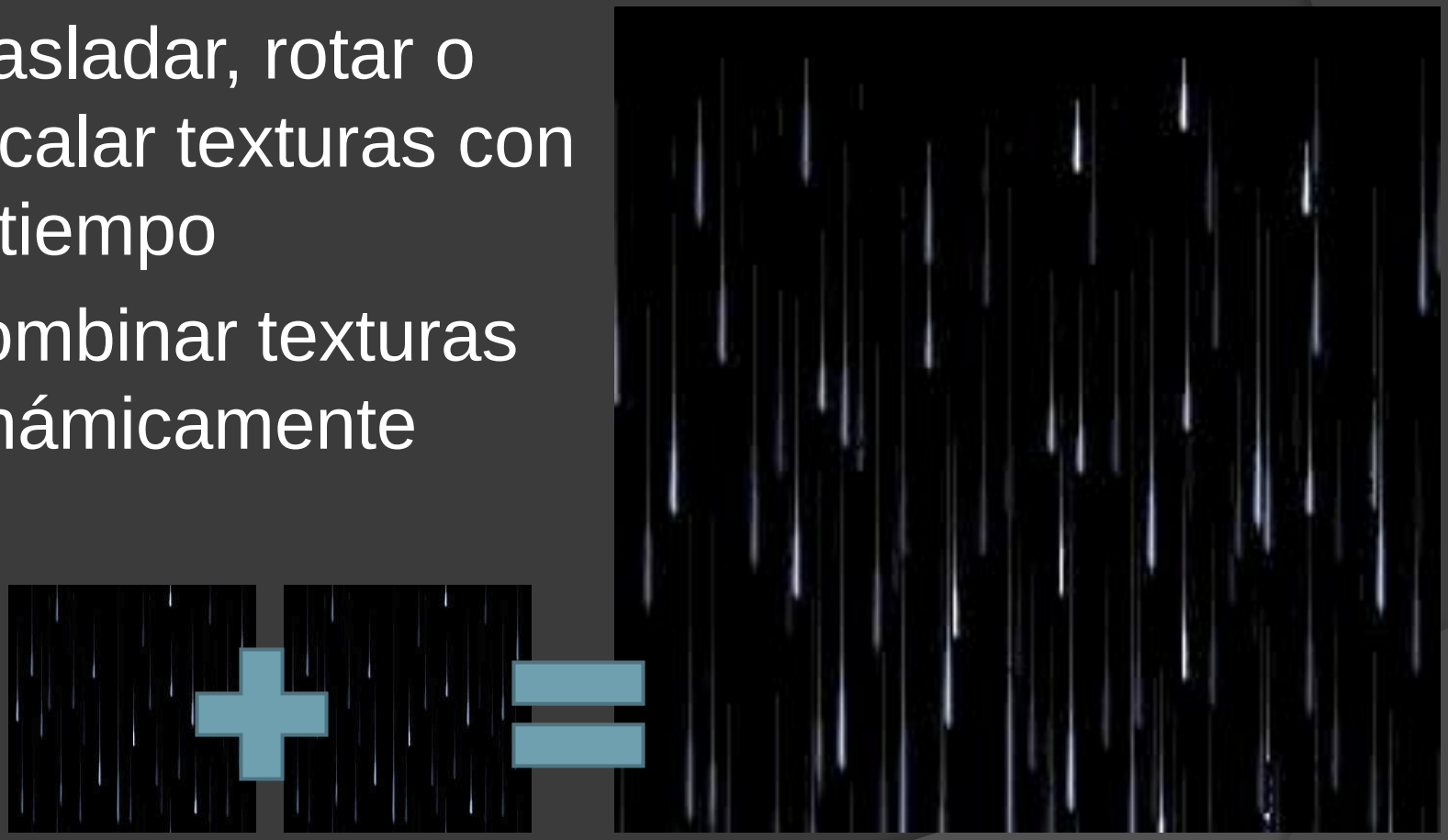


Problemas ?
Resolver colisiones

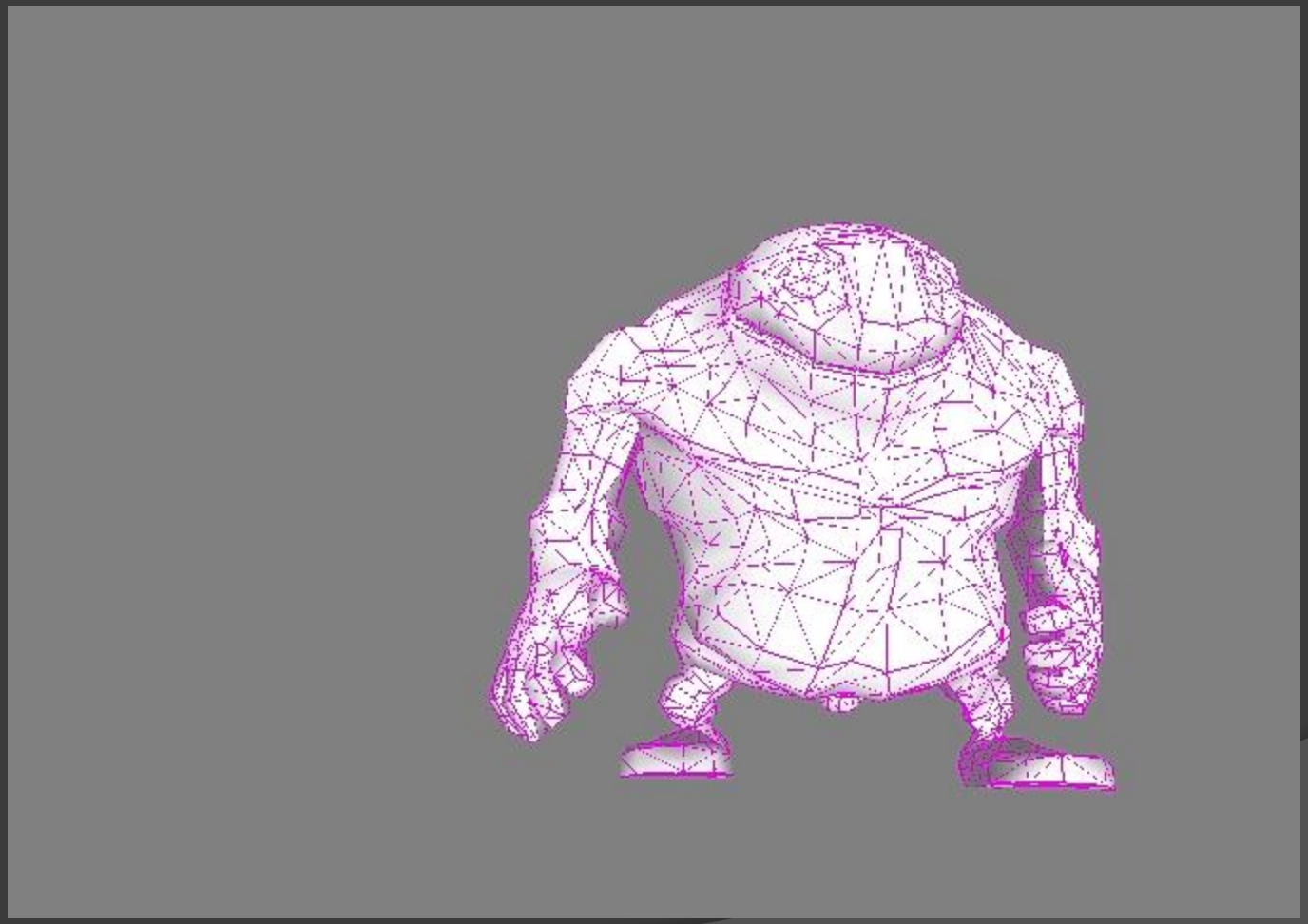
Rain

Texture animation

- Trasladar, rotar o escalar texturas con el tiempo
- Combinar texturas dinámicamente



Caso 2



Principios

- Define las poses de un personaje en determinados puntos de tiempo llamados : “keyframes” (análogo a la animación cuadro por cuadro)

Frame 0

Frame 20

Keyframing

Principios

- El animador crea los *keyframes*
- Los *keyframes* tienen la información del estado de la animación en un punto del tiempo (posición de Vértices)
- Para tener una animación fluida necesito muchos Frames
  - => mayor espacio en memoria

```
// Seudo-código Animación con Keyframing
If (Repetir_Animacion)
   _Actual= Frames[Tiempo mod #Frames]
Else
  _Actual = Frames[Tiempo]
```

Frames faltantes?

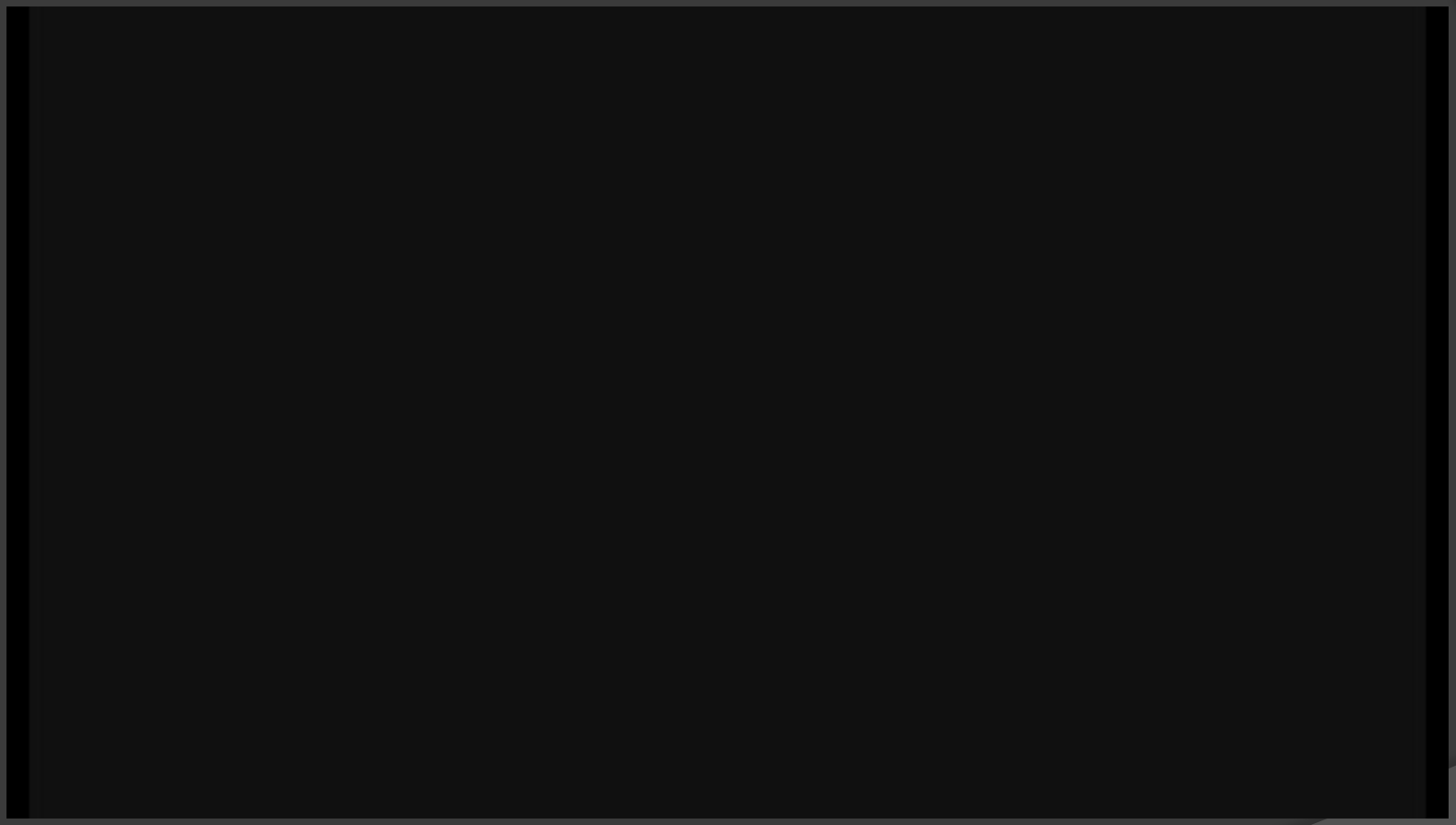
- Los *frames* pueden haber sido generados con otra frecuencia (por ej. 10 fps)
- Como hacer para renderizar a otro *frame rate ?*
  - Se interpola linealmente

    Por cada vértice v del objeto

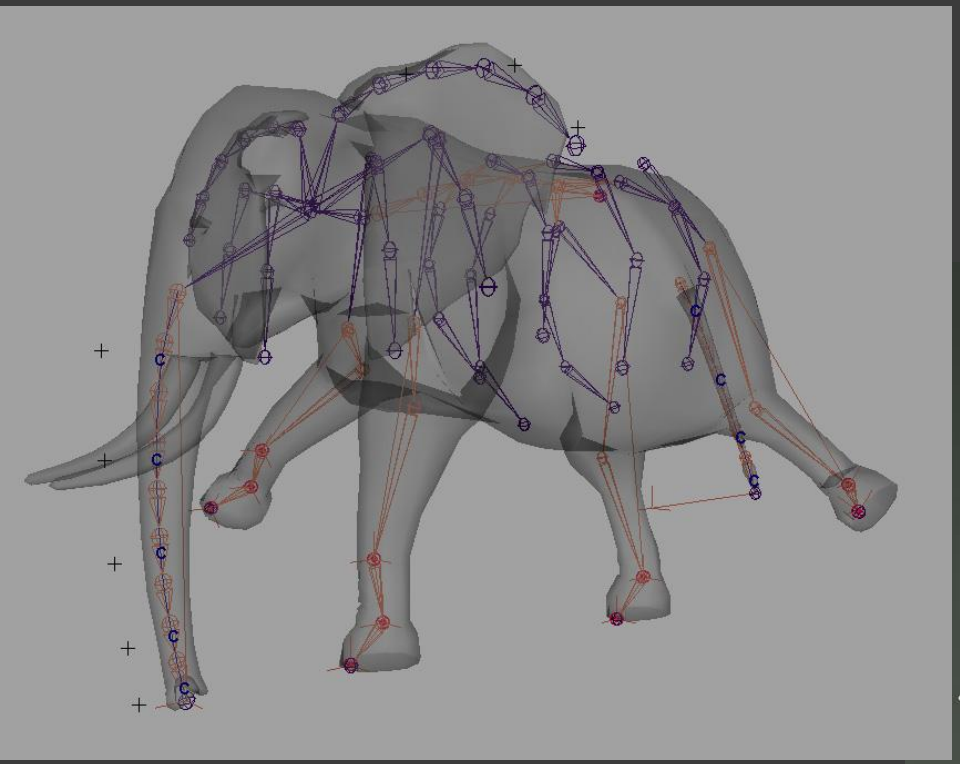
    v[t] = (v[t-1] + v[t+1]) / 2

Problemas ?
Mucho trabajo del Diseñador

FK + Physics + Particles

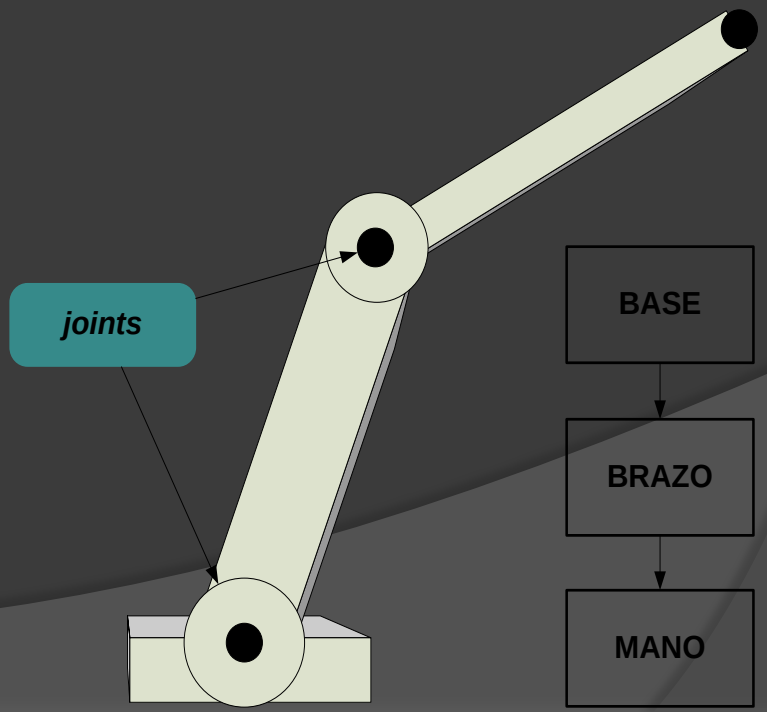


Modelos Articulados



Modelos Articulados

- Es una forma de mejorar el diseño de los Frames
- Las poses del personaje se describen por un conjunto de cuerpos rígidos (*bones*) unidas por articulaciones (*joints*)
  - Cada bone, se le asocian triángulos que representan la una piel (Skin)

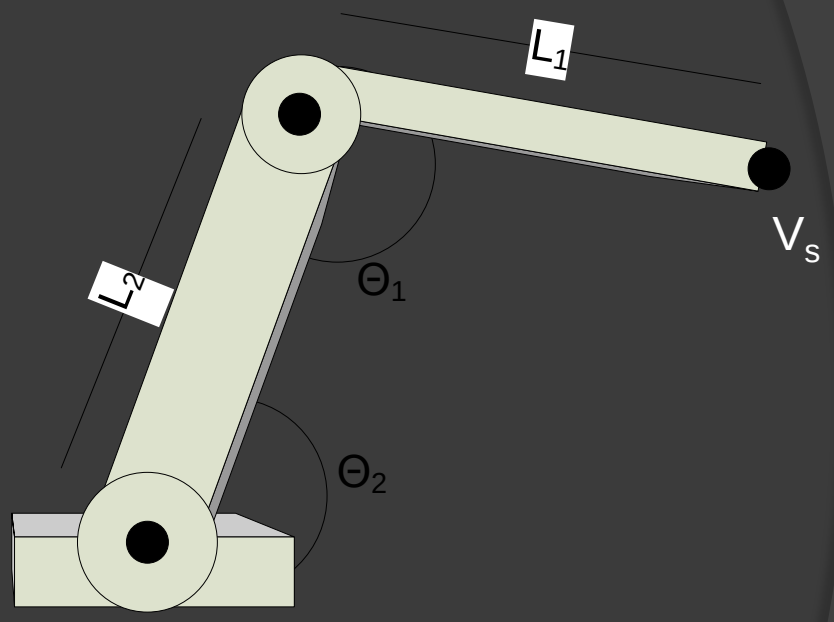


Jerarquía del esqueleto

- Cada transformación a un *bone* es relativa al padre en la jerarquía
- El diseñador gráfico modela el objeto básico, y trabaja sobre el *esqueleto*

Forward Kinematics (FK)

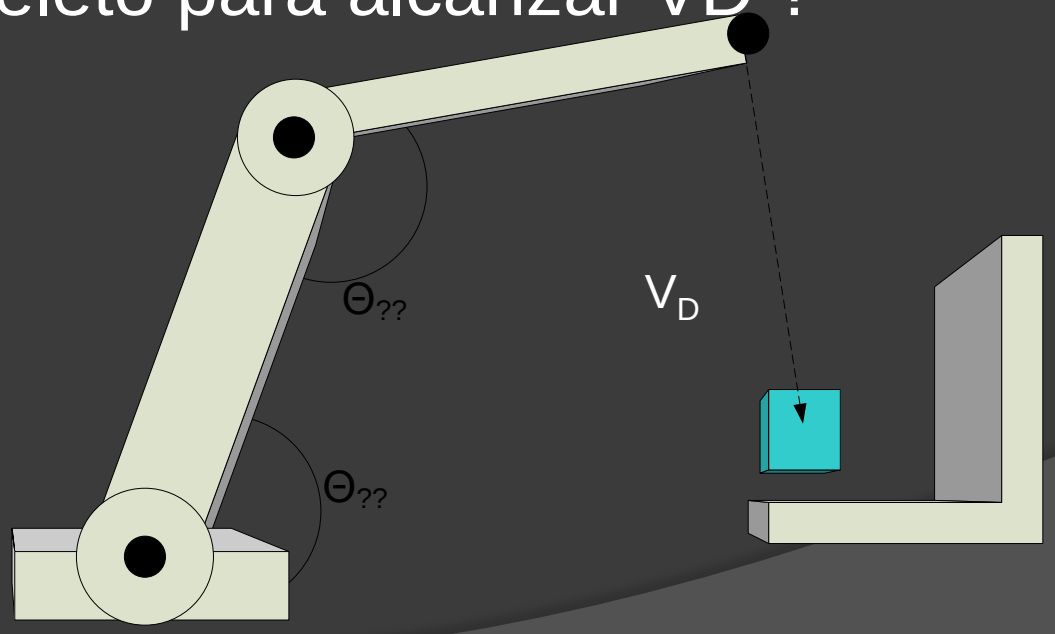
- Dado un esqueleto, con sus parámetros
  - Cuál es la posición de V_s en las coordenadas de locales?



$$V_s = (L_1 \cos \Theta_1 + L_2 \cos (\Theta_1 + \Theta_2) , L_1 \sin \Theta_1 + L_2 \sin (\Theta_1 + \Theta_2))$$

Inverse Kinematics (IK)

- Dado un esqueleto, un conjunto de parámetros iniciales VO y un resultado VD
  - Cuáles son los parámetros del esqueleto para alcanzar VD ?



FK & IK

- Las técnicas trabajan en conjunto
- Inverse kinematics
  - Movimiento basado en objetivo
  - Calcula las transformaciones necesarias para alcanzar el resultado
- Forward kinematics
  - Tiene parámetros pre definidos
  - Se calcula el “vector” posición por cada tiempo, en función de las transformaciones existentes

Como se hace en las películas y en los juegos?

- Controles jerárquicos
  - Gestos, sonrisas, pestañeo, etc.
  - IA
- Simulación física para aspectos secundarios
  - Pelo, ropa, agua
  - Sistemas de partículas

- *Se requiere mucho esfuerzo en construir los modelos 3D, los esqueletos y los controles*

Como se hace en los juegos?

Crowd Simulation en juegos

- Se aplican técnicas de IA para el movimiento en grupo : Flocking, Path finding
- Clone de objetos para reducir espacio en memoria

Age of Empires III

Recomendaciones

- Definir a priori el tipo de animación que vamos a usar para cada tipo de objeto "Mantener lo simple, simple"
- La visualización depende de la animación, pero deberìan estar *desacopladas*
  - Animar y luego visualizar (esquema serial)
  - Tener en cuenta el *Framerate* en la animación
  - Balancear la carga del CPU y del GPU

Links

- http://www-etud.iro.umontreal.ca/~clavetsi/physicsingraphics-details.html
- http://www.pixar.com
- http://www.renderman.org