

## Tema 2 - Arquitectura y modelos para entornos virtuales.

### 2.4 Modelos de generación procesal.

Germán Arroyo, Juan Carlos Torres

5 de febrero de 2021

### Tema 2: Arquitectura y modelos para entornos virtuales.

2.1 Grafos de escena y modelos jerárquicos.

2.2 Métodos básicos de representación.

2.3 Sistemas básicos de iluminación y cámaras.

2.4 Modelos de generación procesal.

## 2.4 Modelos de generación procesal

Hay varios objetos que se pueden generar mediante algoritmos que basados en sólidos o mallas.

Revolución (*lathe*).

Extrusión y soleados (*extrusion/loft*).

Operaciones booleanas (*boolean*).

## Revolución (I)

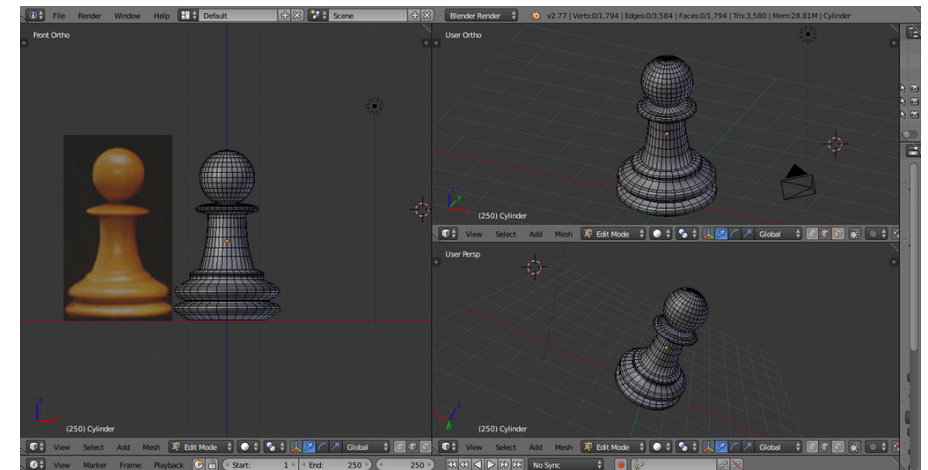


Figura 1: Objeto creado por revolución.

## Revolución (II)

Tomar el vector del eje.

Desplazar un vértice del centro del eje y rotar a incrementos (nº de caras) hasta completar el nº de grados deseado (normalmente 360º).

Siguiente vértice:

- ▶ Añadir cierta altura ( $1/altura\_total\_modelo$ )
- ▶ Si la iteración es par, sumar ángulo de giro, en otro caso restar. Repetir 2.

Construir las caras:  $(v_{j,i}, v_{(j+1) \bmod N,i}, v_{j,(i+1) \bmod N})$ ,  $N$  es el numero de pasos.

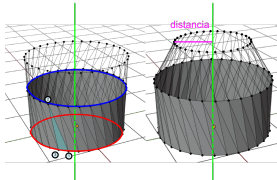


Figura 2: Pasos para realizar la revolución.

## Revolución (III)

Matriz de rotación en eje arbitrario:

<http://ksuweb.kennesaw.edu/~plaval/math4490/rotgen.pdf>

$$T = \begin{pmatrix} tu_x^2 + C & tu_xu_y - Su_z & tu_xu_z + Su_y & 0 \\ tu_xu_y + Su_z & tu_y^2 + C & tu_yu_z - Su_x & 0 \\ tu_xu_z - Su_y & tu_yu_z + Su_x & tu_z^2 + C & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Donde:

$\hat{r} = (u_x, u_y, u_z)$  = eje de rotación, y  $\Theta$  ángulo de rotación (360º).

$C = \cos \Theta$

$S = \sin \Theta$

$t = 1 - C$

## Extrusión y solevados (I)

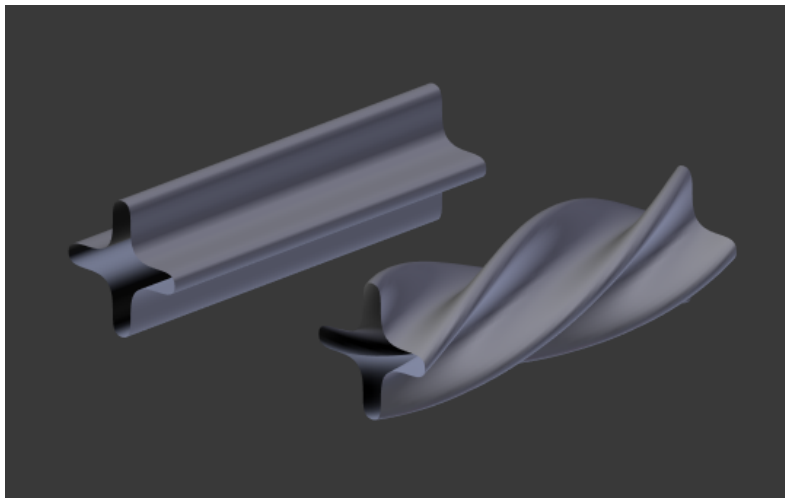


Figura 3: Objeto creado mediante extrusión (izquierda) y solevado (derecha).

## Extrusión y solevados (II)

$i = 0$

$\vec{k}_i$  es el primer vértice de la curva de recorrido.

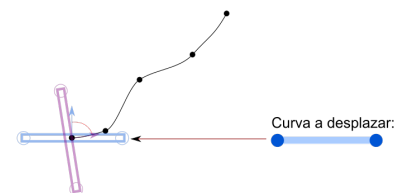
$\hat{d} = \text{norm}(k_{i+1} - k_i)$ .

Desplazar la curva al punto  $\vec{k}$ , girarla  $\Theta$  grados (punto de pivote):

$\Theta = \arccos[(\hat{d} \cdot \hat{N}) / (|\hat{d}| \cdot |\hat{N}|)] = \arccos(\hat{d} \cdot \hat{N})$

Incrementar  $i$  y repetir 2.

Construir los triángulos como en la revolución.



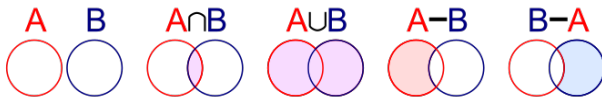
## Operaciones booleanas (I)

Conjunto de operaciones booleanas entre sólidos en  $\mathbb{R}^3$ :

Unión:  $A \cup B = \{x \in A \text{ or } x \in B\}$

Intersección:  $A \cap B = \{x \in A \text{ and } x \in B\}$

Diferencia:  $A - B = \{x \in A \text{ and } x \notin B\}$



## Operaciones booleanas (II)

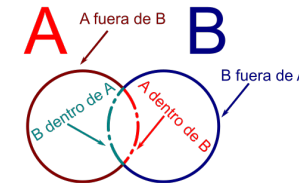
Algoritmo:

Encontrar las intersecciones entre  $A$  y  $B$ .

Dividir ambos modelos por las intersecciones, y clasificar las regiones.

Decidir qué regiones se borran.

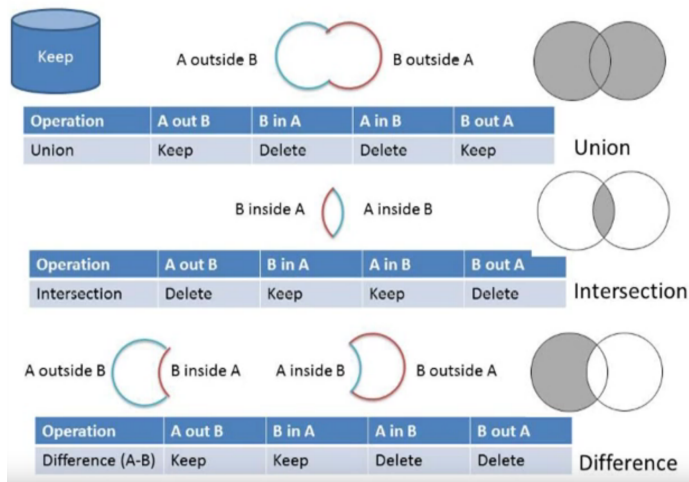
Unir entre sí las regiones restantes.



**Figura 4:** Clasificación de regiones.

<https://www.youtube.com/watch?v=QWtknlm5kn8>

## Operaciones booleanas (III)



**Figura 5:** Esquema de decisiones para eliminar piezas.

## Modelado procedural

**Modelado procedural:** el modelo 3D se describe usando código en lugar de datos

Ventajas:

Generación automática.

Fácil parametrización y continuidad.

Menor tamaño en memoria y disco.

Mayor cantidad de contenido.

Variedad (aleatoriedad).

## Generación de números pseudoaleatorios

Se basan en una **semilla**, siempre obtendremos los mismos valores **si el algoritmo y la semilla coinciden**.

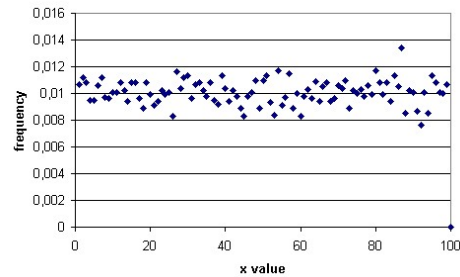


Figura 6: Generación de números pseudoaleatorios.

## Ejemplo: Perlin Noise

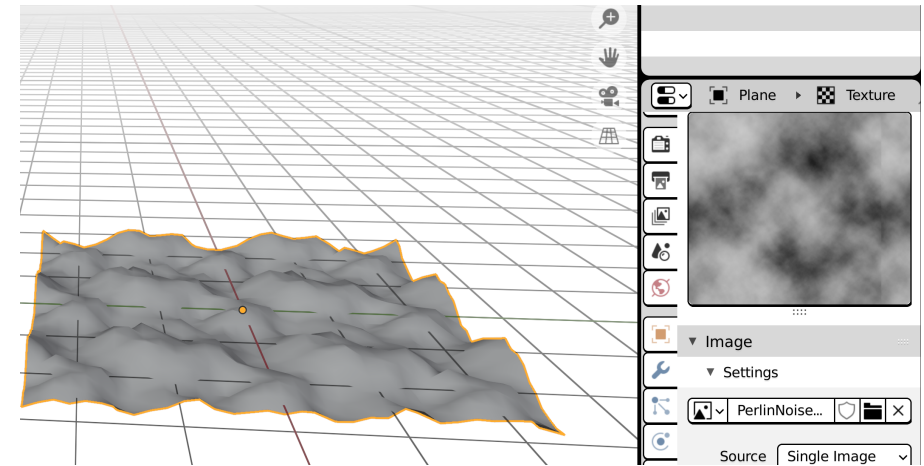
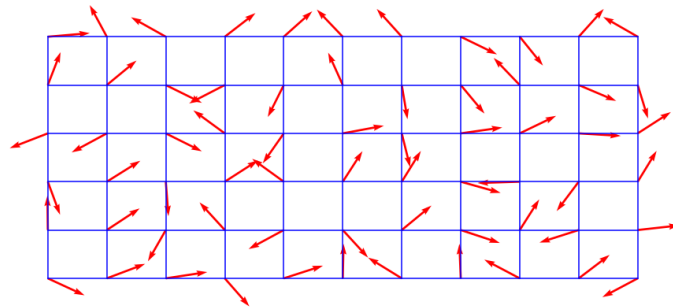


Figura 7: Ejemplo de ruido Perlin 2D.

## Algoritmo de Perlin (I)

Definir una rejilla de  $N > 1$  dimensiones:

Cada intersección está asociada con un vector de valores aleatorios.



## Algoritmo de Perlin (II)

Para calcular el valor de cada punto:

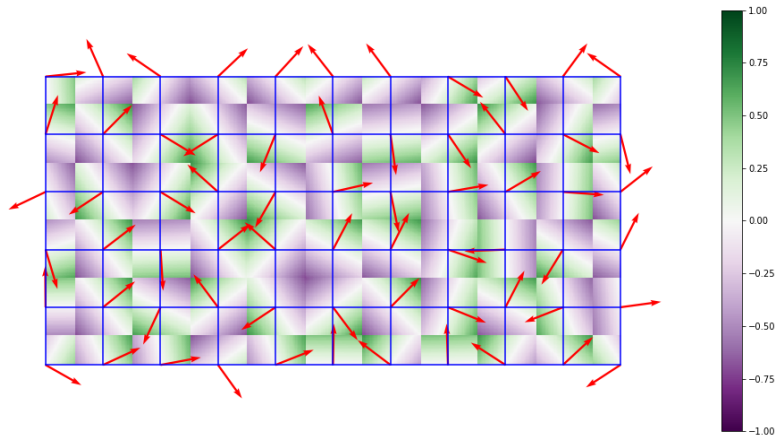
Cada punto de intersección, desplazarse en la dirección aleatoria dada y ver en que celda cae.

Identificar los puntos de las esquinas de la celda y sus direcciones.

Calcular el vector de desplazamiento (desde el punto candidato a cada esquina).

## Algoritmo de Perlin (III)

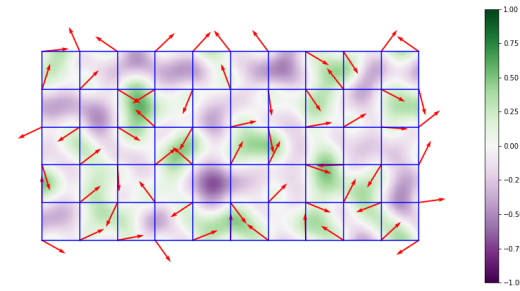
Calcular el producto escalar del vector de gradiente y los vectores de desplazamiento.



## Algoritmo de Perlin (IV)

El último paso es la interpolación de los valores de las esquinas.

En muchos casos la función es de tipo sigmoide (*smoothstep*), pero cualquier función cuya primera derivada (y posiblemente la segunda) sean 0 es válida. Por ello, se suele aproximar a una interpolación lineal según avanzamos hacia las esquinas.



## Fractales determinísticos

Se parte de una forma inicial (iniciador) y una regla de sustitución. En cada interacción se sustituye una parte usando la regla de sustitución.



Figura 8: Ejemplo de fractal determinístico.

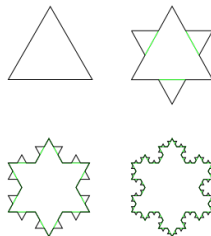


Figura 9: Otro ejemplo de fractal determinístico.

## Generación de terrenos sencillos automática

La regla puede contener números aleatorios (ej. Perlin Noise):

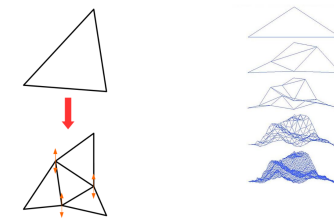


Figura 10: Generación de terreno mediante fractal determinístico.

[https://upload.wikimedia.org/wikipedia/commons/6/6d/Animated\\_fractal\\_mountain.gif](https://upload.wikimedia.org/wikipedia/commons/6/6d/Animated_fractal_mountain.gif)

## Gramáticas formales

Una gramática formal consiste en una serie de **reglas de producción** ( $A \rightarrow B$ ), donde cada lado contiene una serie de símbolos.

**Símbolo inicial:** si podemos reducir hasta aquí significa que los símbolos pertenecen al lenguaje.

**Símbolos no terminales:** puede aplicarse alguna regla de producción.

**Símbolos terminales:** no se puede aplicar más reglas.

**Cadena vacía:** ( $\epsilon$ ) representa la ausencia de símbolos, y se considera terminal.

Ejemplo:

$S \rightarrow AB$

$S \rightarrow \epsilon$  (de forma simplificada:  $S \rightarrow AB|\epsilon$ )

$A \rightarrow aS$

$\$B \rightarrow b$

## Clasificación de gramáticas (I)

La clasificación de Chomsky define 4 tipos de gramáticas.

Tipos de gramáticas:

Tipo-0: incluyen todas las gramáticas formales.

Tipo-1: generan lenguajes sensibles al contexto.

Tipo-2: generan lenguajes libres de contexto (contexto teórico para la mayoría de los lenguajes de programación).

Tipo-3: generan lenguajes regulares (expresiones regulares de búsquedas [grep]).

## Clasificación de gramáticas (II)

Significado de los símbolos de las tablas:

$a$  = Terminal;  $A, B$  = No terminales.

$\alpha, \beta, \gamma$  = cadenas de terminales y/o no terminales.

►  $\alpha, \beta$  pueden ser  $\epsilon$

►  $\gamma \neq \epsilon$

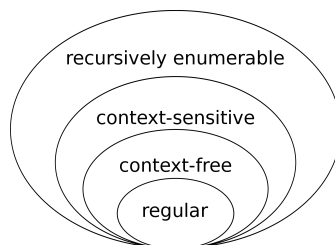


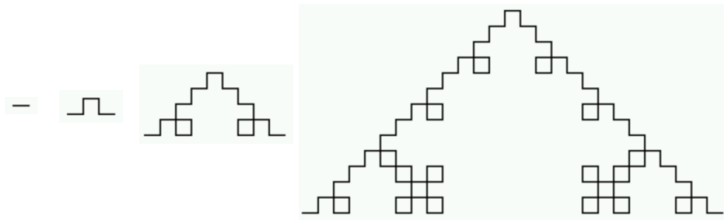
Figura 11: Clasificación de 1920 dada por Chomsky.

## Clasificación de gramáticas (II)

Gramática	Lenguaje	Autómata	Reglas de producción
Tipo-0	Enumerable recursivamente	Máquina de Turing	$\gamma \rightarrow \alpha$ (sin restricciones)
Tipo-1	Sensible a contexto	Autómata linealmente acotado	$\alpha A \beta \rightarrow \alpha \gamma \beta$



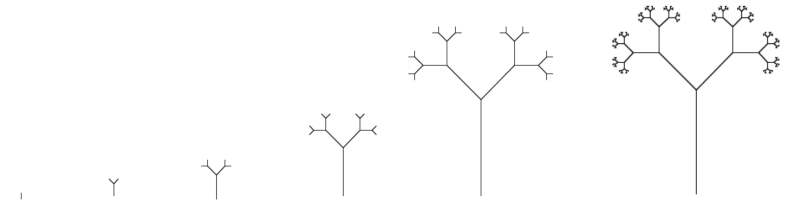
## Ejemplo: curva de Koch (II)



**Figura 12:** Curva de Koch generada con gramática L-system.

## Ejemplo: árbol 2D

$V = \{Y, T\}$  ( $Y$  = yema [mitad de tamaño que tallo],  $T$  = tallo)  
 $S = \{[, ]\}$  ( $[$  = bifurcación a izquierda,  $]$  = bifurcación a derecha)  
 $\omega = \{Y\}$   
 $P = \{(Y \rightarrow T[Y]Y), (T \rightarrow TT)\}$



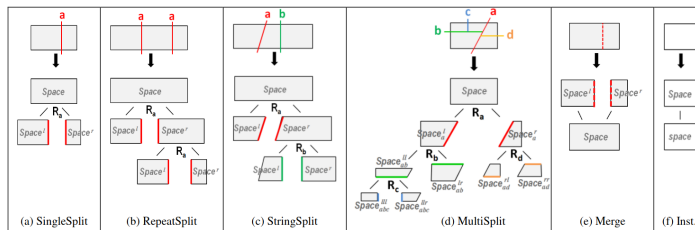
**Figura 13:** Ejemplo de gramática L-system para la generación de árbol 2D.

## Aún más ejemplos

Becker, S., Peter, M., Fritsch, D., Philipp, D., Baier, P., & Dibak, C. (2013).

**Combined grammar for the modeling of building interiors.** ISPRS

Ann. Photogramm. Remote Sens. Spatial Inf. Sci, 1-6.



**Figura 14:** Generación de interiores de edificios con gramáticas L-system.