Tema 2 - Arquitectura y modelos para entornos virtuales.

2.4 Modelos de generación procesal.

Germán Arroyo, Juan Carlos Torres

5 de febrero de 2021

Contenido del tema

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 genenar mediante algorimos que basados en sólidos o mallas.

- Revolución (*lathe*).
- Extrusión y solevados (*extrusion*/*loft*).
- Operaciones booleanas (*boolean*).

Revolución (I)

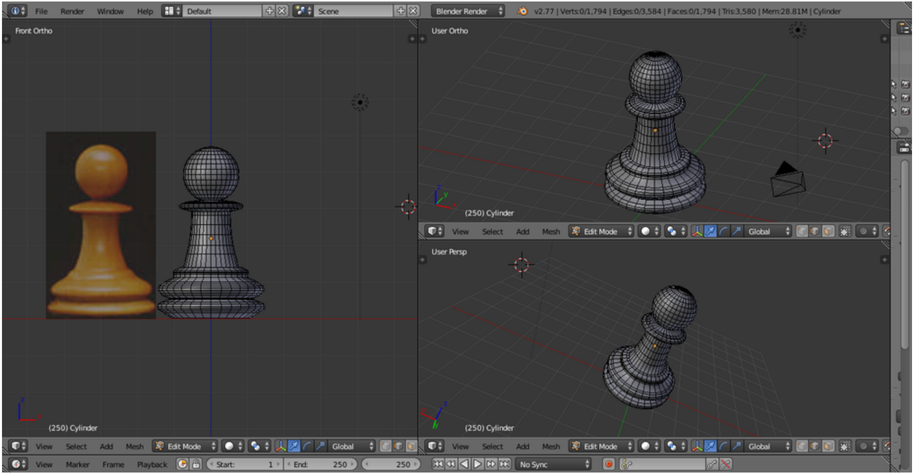


Figura 1: Objeto creado por revolución.

Revolución (II)

1. Tomar el vector del eje.
2. Desplazar un vértice del centro del eje y rotar a incrementos (nº de caras) hasta completar el nº de grados deseado (normalmente 360º).
3. Siguiente vértice:
   - Añadir cierta altura $(1/altura\_total\_modelo)$
   - Si la iteración es par, sumar ángulo de giro, en otro caso restar. Repetir 2.
4. Construir las caras: $(v_{j,i}, v_{(j+1) \bmod N,i}, v_{j,(i+1) \bmod N})$, N es el numero de pasos.

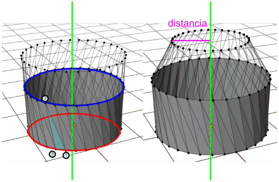


Figura 2: Pasos para realizar la revolución.

Revolución (III)

Matriz de rotación en eje arbitrario:

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

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

Donde:

- $\hat{r} = (u_x, u_y, u_z)$ = eje de rotación, y Θ ángulo de rotación (360°).
- $C = \cos \Theta$
- $S = \sin \Theta$
- $t = 1 - C$

Extrusión y solevados (I)

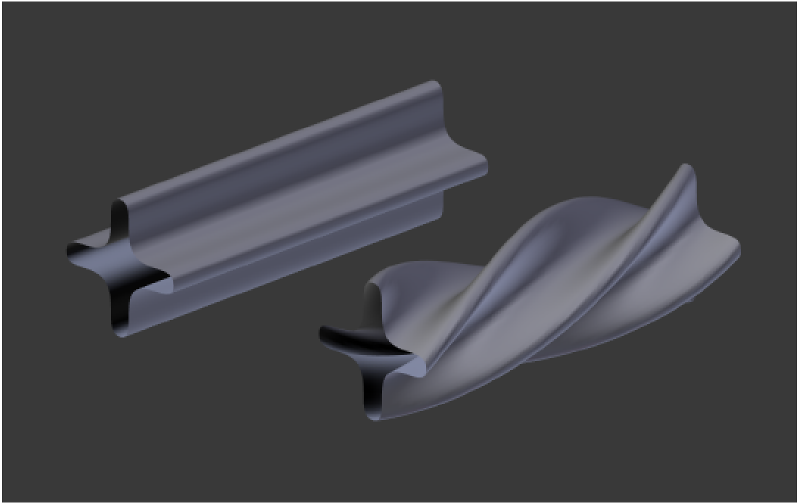
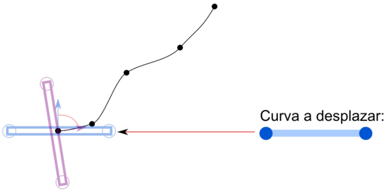


Figura 3: Objeto creado mediante extrusión (izquierda) y solevado (derecha).

Extrusión y solevados (II)

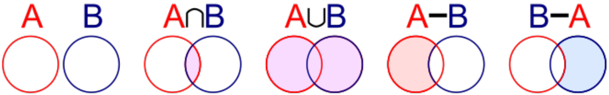
1. $i = 0$
2. $\vec{k}_i$ es el primer vértice de la curva de recorrido.
3. $\hat{d} = \text{norm}(k_{i+1} - k_i)$).
4. Desplazar la curva al punto $\vec{k}$, girarla Θ grados (punto de pivote):
   $\Theta = \arccos[(\hat{d} \cdot \hat{N})/(|\hat{d}| \cdot |\hat{N}|)] = \arccos(\hat{d} \cdot \hat{N})$
5. Incrementar i y repetir 2.
6. Construir los triángulos como en la revolución.



Operaciones booleanas (I)

Conjunto de operaciones booleanas entre sólidos en R^3:

- Unión: $A \cup B = \{x \in A$ or $x \in B\}$
- Intersección: $A \cap B = \{x \in A$ and $x \in B\}$
- Diferencia: $A - B = \{x \in A$ and $x \notin B\}$



Operaciones booleanas (II)

Algoritmo:

1. Encontrar las intersecciones entre A y B.
2. Dividir ambos modelos por las intersecciones, y clasificar las regiones.
3. Decidir qué regiones se borran.
4. Unir entre sí las regiones restantes.

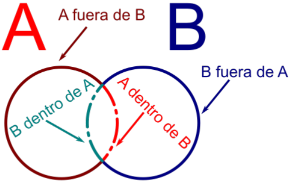
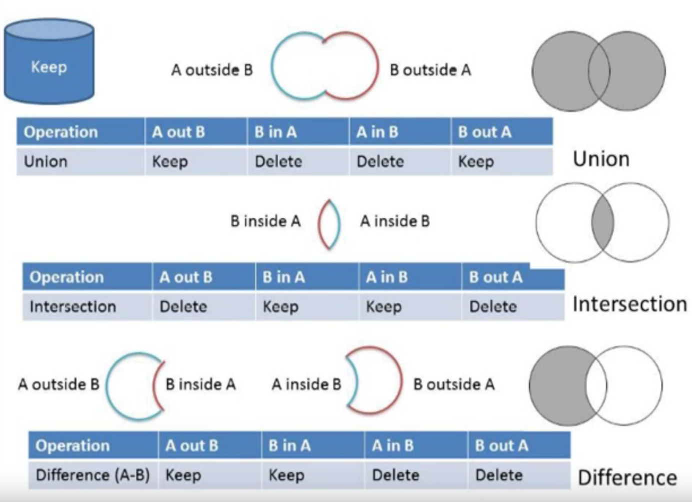


Figura 4: Clasificación de regiones.

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

Operaciones booleanas (III)



Operation	A out B	B in A	A in B	B out A
Union	Keep	Delete	Delete	Keep

Operation	A out B	B in A	A in B	B out A
Intersection	Delete	Keep	Keep	Delete

Operation	A out B	B in A	A in B	B out A
Difference (A-B)	Keep	Keep	Delete	Delete

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**.

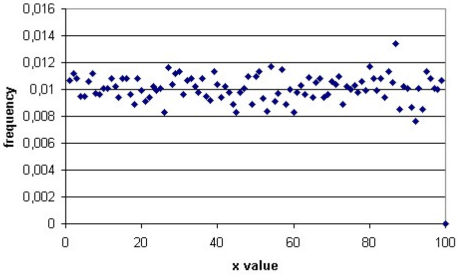


Figura 6: Generación de números pseudoaleatorios.

Ejemplo: Perlin Noise

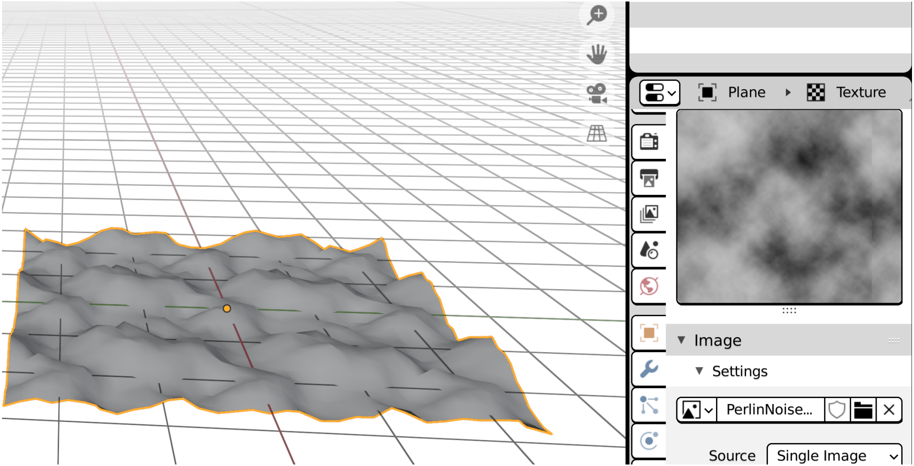
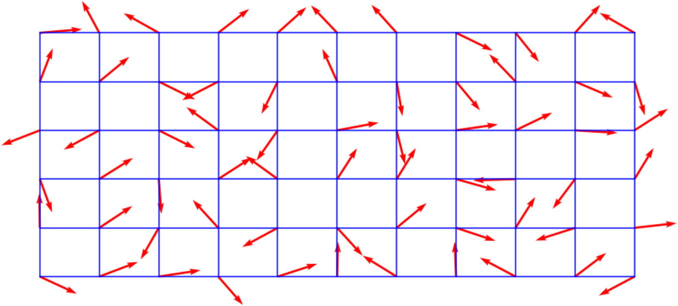


Figura 7: Ejemplo de ruido Perlin 2D.

Algoritmo de Perlin (I)

1. Definir una rejilla de $N > 1$ dimensiones:
   - Cada intersección está asociada con un vector de valores aleatorios.

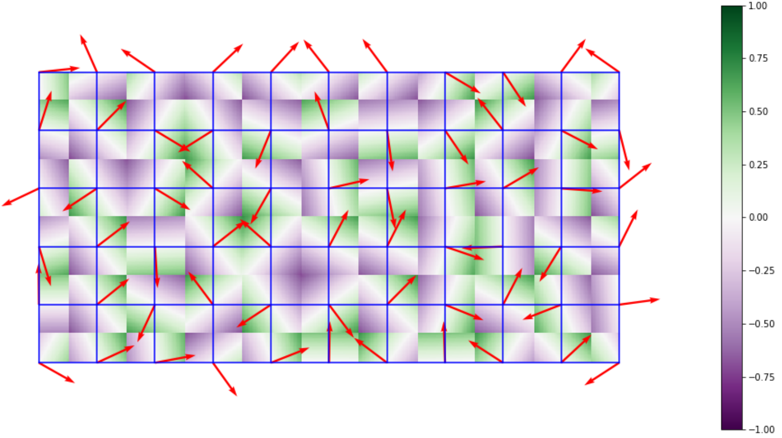


Algoritmo de Perlin (II)

2. 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)

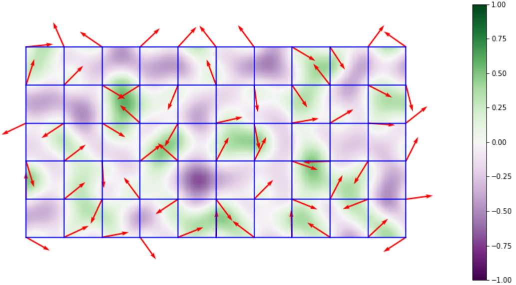
- Calcular el producto escalar del vector de gradiente y los vectores de desplazamiento.



Algoritmo de Perlin (IV)

3. 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 linear según avanzamos hacia las esquinas.



Fractales determinísticos

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

Figura 8: Ejemplo de fractal determinístico.

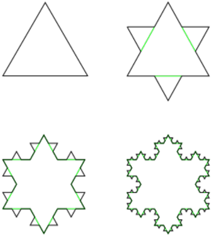


Figura 9: Otro ejemplo de fractal determinístico.

Generación de terrenos sencillos automática

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

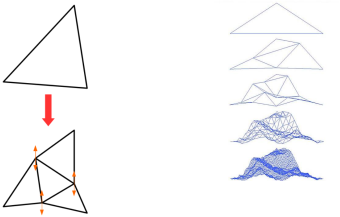


Figura 10: Generación de terreno mediante fractal determinístico.

https://upload.wikimedia.org/wikipedia/commons/6/6d/Animated_fractal_mountain.gif

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:** (ϵ) 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 →b

Clasificación de gramáticas (I)

La clasificación de Chomsky define 4 tipos de gramáticas.

Tipos de grámáticas:

- 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.
- α, β, γ = cadenas de terminales y/o no terminales.
  - α, β pueden ser ϵ
  - $\gamma \neq \epsilon$

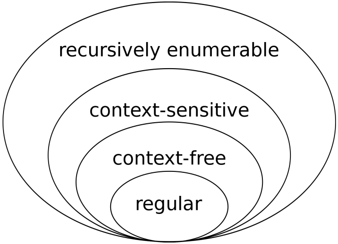


Figura 11: Clasificación de 1920 dada por Chomsky.

Clasificación de gramáticas (II)

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

Clasificación de gramáticas (III)

Gramática	Lenguaje	Autómata	Reglas de producción
Tipo-2	Libre de contexto	Autómata con pila	$A \to \alpha$
Tipo-3	Regular	Autómata de estados finitos	$A \to a$, $A \to aB$

Sistemas de Lindenmayer

Las gramáticas L-system son gramáticas usadas para generar plantas y organismos.

$\mathbf{G} = \{V, S, \omega, P\}$

Donde:

- V = símbolos no terminales.
- S = símbolos terminales.
- $\omega \in V$ = define el estado inicial del sistema.
- P = conjunto de reglas de producción.

Ejemplo: números de Fibonacci

- $V = \{A, B\}$
- $S = \{\}$
- $\omega = \{A\}$
- $P = \{(A \rightarrow B), (B \rightarrow AB)\}$

Ejemplo de secuencia de Fibonnacci:

- A, B, AB, BAB, ABBAB, BABABBAB, ABBABBABABBAB, BABABBABABBABBABABBAB, ...
- Las longitudes dan: 1, 1, 2, 3, 5, 8, 13, 21, ...

Ejemplo: curva de Koch (I)

- $V = \{F\}$
- $S = \{+, -\}$ ($+$ = giro izquierda 90º, $-$ = giro derecha 90º)
- $\omega = \{F\}$ (F = dibujar 1 paso hacia delante)
- $P = \{(F \to F + F - F - F + F)\}$

Ejemplo de secuencia de curva de Koch:

- F; F+F-F-F+F; F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F;
  F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F+
  F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F-
  F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F-
  F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F+
  F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F; …

Ejemplo: curva de Koch (II)

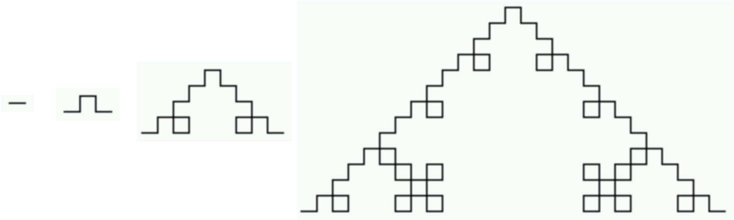


Figura 12: Curva de Koch generada con gramática L-system.

Ejemplo: árbol 2D

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

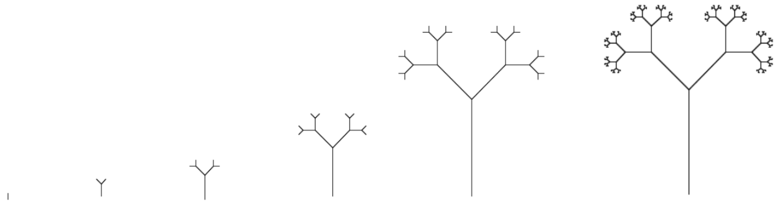


Figura 13: Ejemplo de gramática L-system para la generación de árbol 2D.

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.

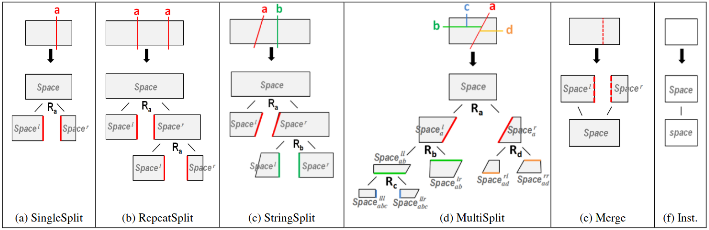


Figura 14: Generación de interiores de edificios con gramáticas L-system.