Práctica 2

Autores

- 100383530 Daniel Alcaide Nombela
- 100383497 Alejandro de la Cruz Alvarado

Entrega final

- Fichero `practica2.png`

Descripción de la imagen

En la imagen se ven:

- Un terreno
- Una «ciudad»
- Un diamante flotante
- Objetos de la práctica anterior

Estructura del árbol de trabajo

Directorios:

- `ciudad`: procedimiento para la generación de la ciudad
- `fields`: mapas de alturas de la ciudad y el terreno
- `previos`: objetos anteriores incluidos en la escena
- `render`: escenas extra
- `sierp`: definición de la pirámide de Sierpinsky usada en el diamante
- `terreno`: procedimiento para la generación del terreno
- `test`: algunas pruebas con algoritmos de generación de estructuras

Ficheros:

- `precalc.sh`: script de generación de mapas de alturas (ver Generación de estructuras procedurales)
- `genfield.py`: script utilitario para convertir matrices de python en imágenes png, usado para generar los mapas de alturas

Renderizado de la imagen

La escena principal está definida en el fichero `practica2.pov`. Para su construcción se usaran los mapas de alturas definidos en el directorio `fields`:

- La ciudad utiliza el mapa `fields/ciudad.png`
- El terreno utiliza el mapa `fields/terreno.png`

El terreno utilizado es una versión replicada del generado originalmente, que está en `fields/terreno_original.png`. Para generar la escena basta con ejecutar el siguiente comando en la raíz del árbol de trabajo:

```
$> povray practica2.pov
```

Descripción de estructuras procedurales

Las estructuras procedurales son objetos fractales generados por algoritmos vistos en clase. Sus implementaciones en POV-Ray están en los ficheros `terreno/precalc.pov` y `ciduad/precalc.pov`. Como salida, imprimen por la salida `Debug` una array de python que representa el campo de alturas.

- El **terreno**, de apariencia natural, es generado por el algoritmo «diamond-square», que es de tipo *midpoint-displacement*.

- El algoritmo que genera la **ciudad** es una variación del *midpoint displacement* en la que no se calculan las medias entre los puntos:

```
1) Se divide un cuadrado inicial en 4 partes de igual superficie.
2) Cada parte se eleva o hunde una cantidad aleatoria
3) Se divide cada cuadrado en 4 partes de igual superficie
4) Vuelta al paso 2
```

En cada iteración se disminuye la variación de la altura aplicada. Este algoritmo genera una estructura fractal con apariencia escalonada/cuadriculada.

Generación de estructuras procedurales

Para generar tanto la ciudad como el terreno se usa el script `precalc.sh`. Para generar el terreno en un entorno POSIX, ejecutar:

```
$> sh precalch.sh terreno
```

o bien:

```
$> ./precalch.sh terreno
```

Para generar la ciudad, basta con sustituir `terreno` por `ciudad`. Para funcionar, python ha de contar con la biblioteca `Pillow` instalada:

```
$> pip install Pillow
```