

bakarrik marraztuz, eta bestetik, triangeluen barne aldea beteta marraztuz. Bigarren erara testura ere kalkulatu behar zaio triangelu barneko puntu bakoitzari. Lehenengo era dagoeneko kodetuta dago, beraz, triangeluak ertzen bidez marraztuta ikus daitezke ezer programatu gabe. Tekla honek era batetik bestera pasatzen du.

- **o**: Triangelu-multzoak osorik marrazteko edo triangelu-multzoan aukeratuta dagoen triangelua bakarrik marrazteko agintzen du tekla honek. Azken finean triangelu-multzoa objektu bezala hartu behar den ala ez esaten du tekla honek.
- **d**: triangelu-multzoak objektutzat hartzen badira, objektu denak ala bakarria marrazteko balio du tekla honek. Hau da, ez aukeratutako objektua bakarrik, baizik eta guztiak batera marrazteko eskaera da. Tekla berriz sakatzean denak marraztetik bakarria marraztera aldatuko da berriz ere.
- **r**: aldaketa mota aukeratzen du: biraketa eragingo da hemendik aurrera.
- **t**: aldaketa mota aukeratzen du: leku aldaketa eragingo da hemendik aurrera.
- **g**: aldaketarako erreferentzia-sistema aukeratzen du: aldaketa bi erreferentzia-sistematan eragin daiteke, alde batetik objektuaren erreferentzia-sisteman (lokalean) eta, bestetik, munduko erreferentzia-sisteman (globalean) biraketa eragingo da hemendik aurrera. Tekla honen bidez, sakatzen den bakoitzean, batetik bestera aldatuko da. Hasieran erreferentzia-sistema lokalean eragingo dira.
- **x**, **y**, eta **z**: ardatz horietako baten arabera aldaketa eragiteko agintzen dute tekla hauek.
- **X**, **Y**, eta **Z**: ardatz horietako baten arabera aldaketa eragiteko agintzen dute tekla hauek, baina alderantzizko norabidean.
- **u**: azken aldaketa desegiteko agindua.

2 Enunciado

Se pretende implementar una aplicación que dibuje triángulos con una textura sobre los mismos. El triángulo se define mediante 3 puntos tridimensionales por lo que cada uno tiene las coordenadas (x, y, z) , con $x, y, z \in (-1, 1)$

y, además, a cada vértice se le ha añadido (u, v) que representan las coordenadas de textura, donde $u, v \in (0, 1)$.

La textura a utilizar es una foto en formato **ppm**, que se cargará en la aplicación desde el fichero **foto.ppm**. En este documento se ha utilizado la imagen mostrada en Figura 1.

Además de la textura, la aplicación leerá desde un fichero de texto la definición de los triángulos que se definan en el fichero: cada línea del fichero que comience con el carácter **t** define un triángulo mediante 15 números, cinco para cada vértice del triángulo; los tres primeros representan las coordenadas (x, y, z) , coordenadas espaciales del triángulo, y los otros dos representan las coordenadas de textura (u, v) que definen la parte de la foto que hay que mapear sobre el triángulo. Una línea del fichero tiene esta forma:

t x1 y1 z1 u1 v1 x2 y2 z2 u2 v2 x3 y3 z3 u3 v3

Por ejemplo, dada la foto de la Figura 1, si tenemos un triángulo en la parte inferior izquierda de la ventana de dibujo, y mapeamos sobre ese triángulo la parte inferior izquierda de la foto, es decir, dados los puntos con las coordenadas indicadas en la siguiente tabla:

| | Koordenatuak | testura |
|-----------|---------------|----------|
| 1. puntua | $(-1, -1, 0)$ | $(0, 0)$ |
| 2. puntua | $(1, -1, 0)$ | $(1, 0)$ |
| 3. puntua | $(-1, 1, 0)$ | $(0, 1)$ |

obtendremos la imagen mostrada en la Figura 2.

Se pueden cargar varios ficheros de triángulos en la aplicación, y cada fichero se considerará como un objeto. Para cargar un nuevo fichero hay que pulsar la tecla **<f>** y escribir el nombre del fichero; la aplicación tiene una función que lee los triángulos del fichero de texto e introduce la información del fichero en una cadena. El primer objeto (fichero) de la cadena lo apunta la variable **foptr**, y el objeto seleccionado es el apuntado por **sel_ptr**.

A la hora de dibujar, la aplicación ofrecerá varias posibilidades de dibujar los triángulos:

1. Dibujar polígonos vacíos o rellenos (**lineak** == 1 o 0). En el caso de que haya que dibujar el triángulo vacío, no hay que plasmar la textura sobre el triángulo, y esa parte ya está implementada en el código inicial. Pero el caso de polígono con textura no está implementado.
2. Considerar cada fichero como un objeto o no (**objektuak** == 1 o 0). Si se considera que los triángulos de un fichero componen un objeto, esta opción permite dibujar todos los triángulos del objeto seleccionado, o incluso todos los triángulos de todos los objetos cargados, para ello

la variable **denak** debe tener valor 1. Si tiene 0 dibujará el objeto seleccionado.

3. Si no se consideran objetos, en ese caso dibuja un único triángulo, el indicado por la variable **indexx**. Dicha variable se controla mediante la tecla **<intro>**; cada vez que se pulse **<intro>** la variable **indexx** debe incrementarse para que se dibuje el siguiente triángulo (tras el último debe indicar el primero). Si la variable **denak** tiene valor 1 la aplicación no borra los triángulos previos, por lo que añadirá el triángulo al dibujo que ya existe, pero si **denak** vale 0 borrará lo dibujado y mostrará únicamente el triángulo indicado por la variable **indexx**.

Por defecto, **lineak** = 0, **objektuak** = 0 y **denak** = 0. Es decir, dibujará un único triángulo; el seleccionado del objeto seleccionado. Precisamente el triángulo indicado por la variable **indexx** en el objeto seleccionado (objeto indicado por la variable **sel_ptr**).

Mediante la tecla **<TAB>** se pasará a seleccionar el siguiente objeto (de nuevo al último le sigue el primero).

La aplicación terminará cuando se pulse **<ESC>**.

2.1 Ejemplos

La parte de la textura que queremos plasmar y el triángulo sobre el que queremos que se dibuje no tienen por qué tener la misma forma. Podemos mapear la misma parte de la foto que aparece en la Figura 2 sobre un triángulo rotado y escalado ligeramente, es decir, mantenemos las coordenadas de textura pero hacemos más pequeño y rotamos el triángulo donde mapeamos la foto. Si las coordenadas de los tres puntos del triángulo son: punto 1 en $(-0.92, 0.92, 0)$, punto 2 en $(0.92, -1, 0)$ y punto 3 en $(-0.84, 0.92, 0)$, obtendremos la imagen mostrada en la izquierda de la Figura 3. En la imagen de la derecha de la misma figura se puede ver la misma parte de textura mapeada sobre un triángulo aún más pequeño y mas rotado.

Las tres imágenes de las Figuras 2 y 3 mantienen las proporciones de la foto y el triángulo del lienzo (ambos son triángulos isósceles y rectángulos), pero podemos hacer que la forma del triángulo dibujado y la parte de la textura que mapeamos sobre el triángulo tengan diferentes formas, por ejemplo si utilizamos las mismas coordenadas de textura que en los anteriores dibujos (parte inferior izquierda de la foto) pero lo mapeamos sobre triángulos que no son isósceles, como los mostrados en la Figura 4, se puede ver cómo se deforma el dibujo mapeado.

En realidad podemos sobreponer cualquier triángulo de la foto sobre cualquier triángulo del lienzo, como se puede observar en la Figura 5.

2.2 Ficheros iniciales

Para facilitar la implementación de la aplicación se dispone del código **cargar-triángulos.c** donde se define la función que lee desde el fichero indicado desde el teclado las coordenadas de los tres vértices de cada triángulo así como las coordenadas de textura u, v asociadas a cada vértice,

Además de cargar la información de los triángulos, hay que cargar la textura. Para ello disponemos del fichero **cargar-ppm.c** donde se define la función que carga en la variable **bufferra** y en las variables **dimx** y **dimy** la información de la textura. El fichero que contiene la textura ha de llamarse **textura.ppm** (debe ser una imagen en formato ppm).

Además, disponemos de la aplicación inicial **dibujar-triángulos-y-objetos.c** que muestra la forma de dibujar en un lienzo los puntos que queramos. En ese código el color que se le asigna a cada pixel es el color que corresponde al primer pixel de la foto.

Para poder implementar la aplicación conviene tener una función que dadas las coordenadas (u, v) de la textura calcule el color que le corresponde:

```
unsigned char *color_textura(u,v)
```

Esta función devuelve el puntero a tres caracteres que definen el color (r, g, b) del pixel más cercano a las coordenadas (u, v) en la foto.

Además de la función que calcula el color de textura que corresponde a las coordenadas de textura, hay que implementar la función que partiendo de la información del triángulo (15 números) calcule todos los puntos internos al triángulo:

```
dibujar_triángulo(...)
```

Dicha función debe recibir la información de un triángulo que se habrá leído del fichero y que tiene la información de los tres puntos (cuyas coordenadas son (x_1, y_1, z_1) para el primer punto, (x_2, y_2, z_2) para el segundo... Además al punto 1 le corresponde el punto (u_1, v_1) de la foto, al punto 2 le corresponde el punto (u_2, v_2) de la foto...). Para cada punto interno del triángulo hay que calcular sus coordenadas (x, y, z) y las coordenadas (u, v) . Con las coordenadas (x, y, z) sabemos donde debemos dibujar el pixel, y con las coordenadas (u, v) podremos asignar el color al pixel.

2.3 Interface

Existen varias posibilidades a la hora de dibujar, todas ellas se modifican mediante teclas y es en la función **teklatura** donde se indica qué teclas son las que establecen cada opción o petición:

- **f**: se utiliza para cargar nuevo conjunto de triángulos desde fichero. Pedirá el nombre del fichero desde donde se cargará el nuevo conjunto de triángulos. Los triángulos de cada fichero forman un conjunto de triángulos que se guarda en la aplicación, y la aplicación mantiene en todo momento un único conjunto seleccionado, el indicado por la variable **sel_ptr**. Tras cargar un nuevo conjunto de triángulos este pasa a ser el conjunto seleccionado, y de entre sus triángulos el primero será el triángulo seleccionado.
- **<TAB>**: posibilita cambiar la selección de conjunto de triángulos. Los conjuntos de triángulos se guardan en una cadena y con esta tecla pasamos a seleccionar la siguiente de la cadena. Si la seleccionada fuera el último de la cadena, entonces el nuevo conjunto seleccionado pasará a ser el primero de la cadena. Además, reasigna la variable **indexx** al primer triángulo del conjunto seleccionado.
- **<Intro>**: posibilita cambiar de triángulo seleccionado. cada vez que pulsemos la tecla seleccionará el siguiente triángulo, y tras el último pasará al primero.
- **<ESC>**: finaliza la aplicación.
- **l**: modifica la forma de dibujar los triángulos; los triángulos se pueden dibujar de dos formas, por un lado dibujando las aristas del triángulo, y por otro lado, dibujando todos los puntos internos del triángulo. En esta segunda forma de dibujar hay que calcular la textura correspondiente a cada punto interno. La primera forma de dibujar está implementada. Esta tecla, cada vez que se teclee, cambia de una forma de dibujo a la otra.
- **o**: permite dibujar todo el conjunto de triángulos (como si se tratara de un objeto) o dibujar solamente el triángulo seleccionado del conjunto de triángulos. La tecla hace cambiar de una opción a la otra.
- **d**: Si estamos en modo de dibujar todo el conjunto de triángulos, mediante esta tecla se puede pedir que se dibujen todos los objetos o solo el seleccionado. Cada vez que pulsemos la tecla cambiaremos de una opción a la otra.
- **r**: establece el tipo de transformación: a partir de este momento se va a rotar el objeto.
- **t**: establece el tipo de transformación: a partir de este momento se va a trasladar el objeto.

- **g**: cambia el sistema de referencia en el que se aplican las transformaciones. Las transformaciones se pueden aplicar en el sistema de referencia del objeto (sistema local) o en el del mundo (sistema global). Mediante esta tecla cambiamos de un sistema al otro.
- **x, y, z**: petición de transformación para cada eje.
- **X, Y, Z**: petición de transformación para cada eje, pero en sentido contrario.
- **u**: petición de deshacer la última transformación.