Curvas

Polinomios de Berstein

Los polinomios de Berstein forman una base para los polinomios de grado n. Cualquier polinomio de grado n se puede describir como una combinación lineal de polinomios de Berstein .

$$B_i^n(u)=\binom{n}{i}\sum_{i=0}^{n} u^i(1-u)^{n-i}$$

Para $i=0$ y $u=0$ considerar $u^i=1$.
Para $i=n$ y $u=1$ considerar $(1-u)^{n-i}=1$.

Se pueden expresar en forma recursiva.
$B_i^n(u)=(1-u)B_i^{n-1}(u)+u\,B_{i-1}^{n-1}(u)$, iniciando la recurrencia con: $B_0^0(u)=1$.

Propiedades:

1. $\sum_0^n B_i^n(u)=1$, es una base.
2. $B_i^n(u)\geq 0$, es convexa.
3. ∀ $u\in[0,1], 0\leq B_i^n(u)\leq 1$

Curvas de Bezier

Una curva de Bezier es una combinación afÌn convexa de sus puntos de control. Para cada valor del par·metro u, se obtienen los coeficientes de la combinación como los valores que asumen los polinomios de Berstein para el valor de u.

$$C(u)=\sum_{i=0}^{n} p_i B_i^n(u)$$

Propiedades:
1.El grado de la base de polinomios es uno menos que la cantidad de puntos de control.
2.El primer y ˙ltimo punto de la curva coinciden con el primer y ˙ltimo punto del grafo de control.
3.El vector tangente en los extremos de la curva tiene la misma dirección que el primer y ˙ltimo segmento del grafo de control respectivamente.
4.Tienen control global.

En particular para este trabajo practico nos interesan las curvas cubicas, sus tangentes y sus normales.

Curvas de Bezier cúbicas

$$C(u)=\sum_{i=0}^{3} p_i B_i^3(u)$$

donde la bases de Berstein cubicas son:

$B_0^3(u)=(1-u)^3$
$B_1^3(u)=3\mathrm{u}\times(1-u)^2$
$B_2^3(u)=3\mathrm{u}^2\times(1-u)$
$B_3^3(u)=3\mathrm{u}^3$

<u>Derivadas</u>

Las derivadas de las curvas de Bezier, se obtiene derivando los polinomios de Berstein. Los polinomios de Berstein se derivan como:

$$\frac{dB_i}{dt^r}(u)=\frac{n!}{(n-r)!}\sum_{i=0}^{n-r} \Delta^r p_i B_i^{n-r}(t)$$

donde:

$\Delta^r p_i=\Delta^{r-1} p_{i+1}-\Delta^{r-1} p_1$, con $\Delta^0 p_i=p_i$

Entonces

$$\frac{dC_i^r}{dt^r}(u)=\frac{n!}{(n-r)!}\sum_{i=0}^{n} p_i(\sum_{j=0}^{n-r} \Delta^r p_j B_j^{n-r}(t))$$

Derivada Primera de la curva cubica – Tangente

$$\frac{dC_i}{dt}(u)=3\times\sum_{i=0}^{3} p_i(\sum_{j=0}^{2} \Delta p_j B_j^2(t))$$

donde la bases de Berstein cuadradas son:

$B_0^2(u)=(1-u)^2$
$B_1^3(u)=2\mathrm{u}\times(1-u)$
$B_2^2(u)=u^2$

Derivada Segunda de la curva cubica – Normal

$$\frac{dC_i^2}{dt^2}(u)=6\times\sum_{i=0}^{3} p_i(\sum_{j=0}^{1} \Delta^2 p_j B_j(t))$$

donde la bases de Berstein lineales son:

$B_0(u)=(1-u)$
$B_1(u)=u$

Curvas de B-Spline

La curva expresada paramétricamente es:

$$C(u)=\sum_{i=0}^{n} p_i N_i^k(u)$$

donde:

- u : par·metro.
- N_i^k : i-esima base de orden k.
- p_i : i-esimo punto de control.
- n : n˙mero de puntos de control.
- k : orden de las bases.

En particular para este trabajo practico nos interesan las curvas cubicas uniformes, sus tangentes y sus normales.

Curvas de B-Spline cúbicas uniformes

Son curvas compuesta por segmentos polinomiales c˙bicos. Tienen una distribución uniforme del par·metro. Grado de continuidad C2.
Se forman como combinación de 4 bases de grado 3:

$$C(u)=\sum_{n=-2}^{1} p_i B_{i+n}(u)$$

donde $B_i(u)=N_i^4(u)$

Bases cubicas:

$$B_0^3(u)=\frac{1}{6}u^3$$

$$B_1^3(u)=\frac{1}{6}(-3\mathrm{u}^3+3\mathrm{u}^2+3\mathrm{u}+1)$$

$$B_{-1}^3(u)=\frac{1}{6}(3\mathrm{u}^3-6\mathrm{u}^2+4)$$

$$B_{-2}^3(u)=\frac{1}{6}(-u^3+3u^2-3\mathrm{u}+1)$$

Para obtener las tangentes y derivadas se procede como las curvas de Bezier, pero con las siguientes bases

Bases cuadr·ticas:

$$B_0^2(u)=\frac{1}{2}u^2$$

$$B_1^2(u)=\frac{1}{2}(-u^2+u+\frac{1}{2})$$

$$B^2_{-1}(u) = \frac{1}{2}(1-t)^2$$

Bases lineales:

$B_0(u) = (1-u)$

$B_1(u) = u$

Calculo de la posición final

El primer paso para obtener la ubicación de cada imagen sobre la curva de Bezier, fue obtener la longitud de dicha curva. Como para la aproximación de la curva se utilizan paso chicos, se uso:

$$longitud = \lim_{n \to \infty} \sum_{k=1}^{n} \sqrt{[(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2]}$$

Entonces, teniendo como dato el numero de imagen y su tamaño, mas la longitud de la curva puedo saber en que lugar deberÌa estar.

Cámara

En la aplicación la c·mara realiza 3 clases de movimiento: rotación en torno al origen usando el mouse, zoom al origen usando el mouse y movimiento autom·tico al cambiar entre la vista de grilla y la vista de curva.

Rotación:

Para realizar este movimiento se setean variables ante un evento de click izquierdo, que permite obtener la posición x del mouse en ese instante. Luego mediante el evento de movimiento de mouse (y mientras el botón esté presionado) se calcula la diferencia de posiciones y asÌ se determina hacia donde rotar y con que velocidad. Adem·s de deshabilito esta función al hacer click sobre el panel de edición para que no interfiera al momento de dibujar curvas (siempre y cuando el panel este visible, de lo contrario se puede rotar haciendo click en cualquier punto de la pantalla).

Para implementar la rotación en sÌ se rotó todo el modelo alrededor del origen en sentido contrario al movimiento del mouse y asÌ tener la ilusión de que se esta rotando la c·mara.

Zoom:

El procedimiento de captura del mouse es equivalente solo que se dispara ante el evento de click derecho.

Para implementar el zoom simplemente se acerco la c·mara al centro.

Automático:

Al presionar la tecla para intercambiar de vistas se genera una curva bspline entre el punto inicial de la c·mara (mirando hacia la grilla, agregado 3 veces para que sea interpolado), dos puntos intermedios, y la posición final de la c·mara (mirando hacia la curva, y también agregado 3 veces). La curva que se genera es una par·bola que se aleja y se acerca al eje z a medida que se eleva (o desciende, seg˙n el caso).

Luego se inicia la animación de la c·mara mediante el uso de un timer y asÌ ir posicion·ndola en los distintos puntos obtenidos con Bspline.

Panel de Edición

Para editar la curva de Bezier fue necesario capturar los clicks realizados sobre el panel de edición. Para esto se uso los limites del viewport 2D para considerar si el punto de control debÌa ser agregado o no y su posición en el mundo 2D.

Para la visualización 3D de la curva simplemente se usó el mismo modelo pero dentro del entorno 3D y con un factor de escala ya que la escena 3D es mucho m·s grande que la 2D.

Si se quiere dibujar una curva nueva se deben borrar los puntos de control y volver a clickear sobre la pantalla de edición.

Carga y Procesamiento de imágenes:

Para realizar la carga de las im·genes a utilizar en las texturas de la aplicación, se decidió utilizar la librerÌa SDL, junto con el complemento SDL-image de la misma. Estas librerÌas no son parte de la instalación por defecto de la mayorÌa de las distribuciones de linux est·ndar, por lo cual se las debe instalar, para ellos puede utilizarse un gestor de paquetes, por ejemplo, synaptic. Una vez instaladas, a la hora de compilar el programa deben pasarse como flags: -lSDL -lSDL_image para de este modo poder utilizar dichas librerÌas.

Para cargar las im·genes desde la clase Imagen, se utilizó, dentro del método cargarImagen, al cual se le pasa como par·metro la ruta de la imagen a cargar, y nos devuelve un SDL_Surface*[1], la función IMG_Load(ruta). En caso de producirse un error al abrir la imagen, el mismo se informa por salida est·ndar y el método retorna NULL.

Externamente a esta clase, por cada imagen cargada, se valida el formato de datos ingresado y se acomoda el valor del mismo al correspondiente seg˙n la imagen cargada antes de crear con la misma la textura.

1 La estructura SDL_Surface está conformada de la siguiente manera:

```
typedef struct SDL_Surface {
        Uint32 flags;
        SDL_PixelFormat *format;
        int w, h;
        Uint16 pitch;
        void *pixels;
        SDL_Rect clip_rect;
        int refcount;
} SDL_Surface;
```

Tutorial:

A continuación, se detalla muy brevemente los controles del teclado y mouse respectivamente para poder utilizar correctamente la aplicación y todas sus funcionalidades.

* Teclado *

- a: muestra los ejes
- b: intercambia entre la vista de grilla y la vista de curva
- c: borra la curva editada
- e: muestra/oculta la ventana de edición
- g: muestra/oculta la grilla
- t: muestra/oculta todas las curvas y polígonos de control
- ESC: sale del programa

* Mouse *

- Rotar la cámara: botón izquierdo apretado y mover mouse hacia los costados
- Zoom: botón derecho presionado y mover mouse arriba (zoomIn) o hacia abajo (zoomOut)