

IMAGEN QUE REPRESENTA LA HORA DEL DÍA

Abstract

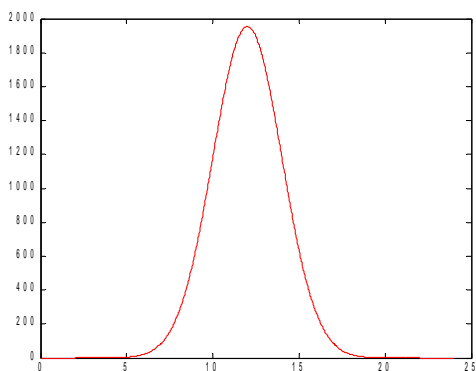
En este artículo se presenta una forma de simular una curva de insolación teórica y cómo mapearla sobre una imagen, de tal forma que en dicha imagen sea de noche o de día en función de la hora.

1. Introducción

La aplicación que se presenta a continuación muestra una manera sencilla de trabajar con funciones matemáticas y obtener una respuesta visual. Podemos aplicarla como fondo de escritorio en nuestro dispositivo móvil, de tal forma que nuestra imagen de fondo reaccione en función de las horas de luz. Así, si arrancamos la aplicación por la mañana la escena será muy luminosa y por la noche estará a oscuras.

La técnica consiste en asociar la opacidad de una imagen (rango de 0 a 100) a una función que describa la curva de insolación (las horas de luz diarias, en un rango de 0 a 24 horas).

Una curva de insolación térmica tiene aproximadamente esta evolución con el tiempo (en abscisas se muestran las horas y en ordenadas los vatios por metro cuadrado):

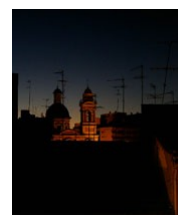


La opacidad (o transparencia, según se mire) de una imagen viene controlada en ActionScript por la propiedad `_alpha` -que no tiene nada que ver

con el canal alfa de una imagen, aunque suene parecido-. Esta opacidad varía linealmente en un rango de 0 (completamente transparente) a 100 (completamente opaca). Lo que vamos a hacer es cambiar esa linealidad por una función que tenga un comportamiento similar a la curva de insolación teórica.

2. Antes de empezar

Para conseguir ese efecto debemos realizar dos fotografías de la misma escena, una en claridad y otra en penumbra.



Estas dos imágenes las convertimos en movieClips, dándole un nombre de instancia identificativo a cada una; por ejemplo `dia_mc` y `noche_mc`.

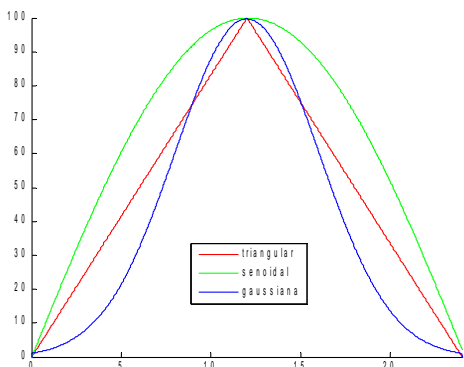
En el panel de Acciones creamos una variable que almacene la hora en que arranca la aplicación. Para ello usaremos la clase `Date`:

```
var fecha:Date = new Date();  
var h:Number = fecha.getHours();  
var m:Number = fecha.getMinutes();  
var s:Number = fecha.getSeconds();  
var tiempo:Number = h + m/60 + s/3600;
```

Si antes no hemos colocado la imagen diurna encima de la nocturna, podemos hacerlo mediante código:

```
dia_mc.swapDepths(5);
```

Y ya por último definimos la función para aplicar a la imagen de día. Vamos a hacer tres aproximaciones, de más burda a más refinada. De lo que se trata es de conseguir que a las 12 del mediodía la imagen tenga su máxima luminosidad, y vaya desvaneciéndose simétrica y progresivamente. La primera aproximación será mediante una función triangular, seguidamente se empleará una función sinusoidal y finalmente una distribución gaussiana.

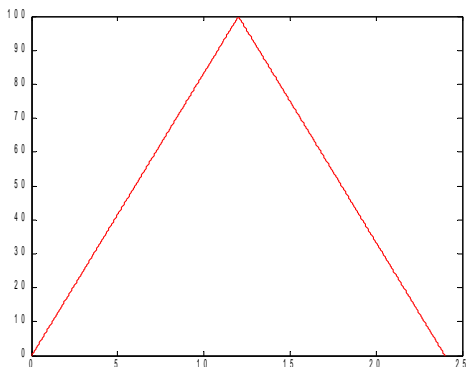


3. Función triangular

La función se define a trozos, según se muestra en la siguiente ecuación:

$$f(x) = \frac{100}{12}x \quad \text{para } 0 \leq x \leq 12$$

$$f(x) = 100 - \frac{100}{12}(x - 12) \quad \text{para } 12 < x \leq 24$$



La cual se define en ActionScript según siguiente sintaxis:

```
if (tiempo >= 0 && tiempo <= 12) {
    dia_mc._alpha = (100/12)*tiempo;
} else if (tiempo > 12 && tiempo <= 24) {
```

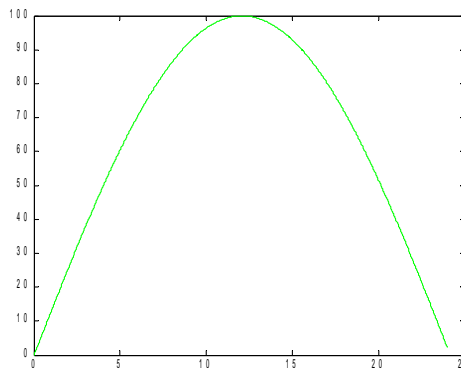
```
dia_mc._alpha = 100 - (100/12)*(tiempo-12);
}
```

Así, sólo falta publicar la película y observar el resultado.

4. Función sinusoidal

En este caso solamente nos interesa el área positiva de la función seno. Para que exista el punto de corte $(x, y) = (24, 0)$ y el máximo en $(12, 100)$, la función debe tener esta expresión:

$$f(x) = 100 \cdot \sin(0.13 \cdot x)$$



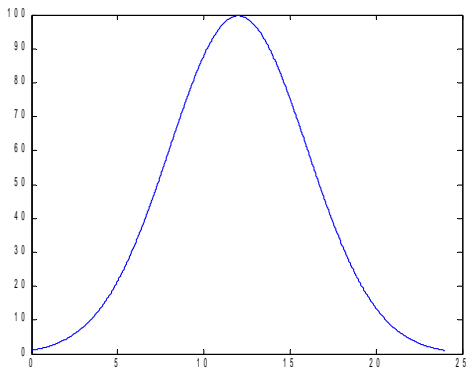
Que se define en ActionScript según:

```
dia_mc._alpha = 100 * Math.sin(.13 * tiempo);
```

5. Función gaussiana

La función de distribución gaussiana se representa mediante esta ecuación:

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2} \quad \forall x \in \mathbb{R}$$



En ActionScript la ecuación es:

```
var gauss:Number = (1/(sigma*Math.sqrt( →  
2*Math.PI))) * (Math.pow(Math.E, →  
-0.5*Math.pow((tiempo-media)/sigma, 2)));
```

Donde *sigma* representa la desviación típica y *media* es la media (ambos valores en horas). Para este ejercicio se ha centrado la media en 12 y se ha tomado una desviación típica de 4 horas, por lo que hay que corregir la amplitud para que el máximo coincida con el punto (12, 100). Así, la opacidad queda:

```
dia_mc._alpha = 1000*gauss;
```

6. Conclusión

Hemos visto una sencilla aplicación que puede resultar cuando menos curiosa de cara a la experiencia de usuario.

Para terminar, decir que si queremos evaluar durante toda la ejecución del SWF la hora, y ver así cómo evoluciona a lo largo del día, debemos encerrar los cálculos anteriores en un bucle `onEnterFrame`.

Referencias

[1] J.M Pinazo, E. Torrella. **Transferencia de calor**. Valencia 1988