

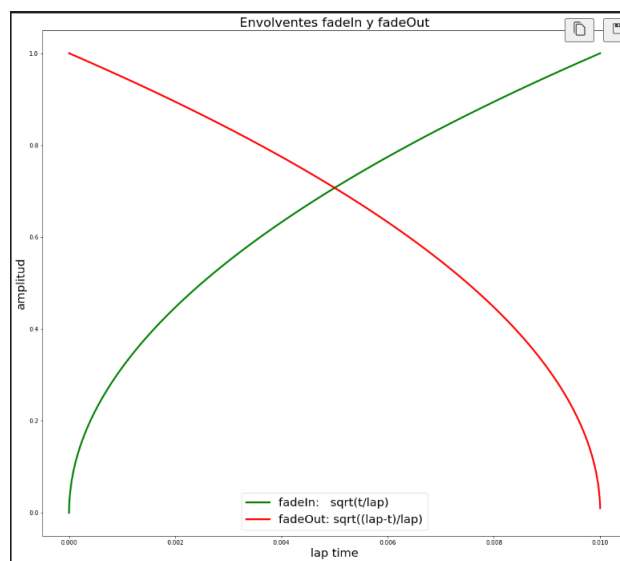
Sonido en videojuegos

Unity Audio 4

1. (**Evaluable**) En este ejercicio vamos a implementar un efecto disparo de rifle mediante la técnica de *slicing* vista en clase (script `Slice.cs`), pero añadiendo crossfades con solapamiento para suavizar la transición entre muestras. La carpeta `muestras/disparo` contiene:

- `head_[1-2]`: fragmentos iniciales de disparo, con el el "golpe" inicial de energía (transitorio).
- `tail_[1-2]`: fragmentos de cola de disparo, que sonarán después del transitorio.
- `casing_[1-4]`: muestras de casquillos contra el suelo.

Para reproducir un disparo seleccionaremos aleatoriamente una muestra de cada tipo (*head*, *tail*, *casing*). Reproduciremos la muestras *head* y *tail* de manera consecutiva, pero con pequeño solapamiento temporal, mezclándolas con envolventes de tipo *equal power crossfade*, según se muestra a continuación:



En este gráfico se muestra la envolvente para fadeout en rojo y la de fadein en verde. En la leyenda se muestra la función que genera cada una de las curvas, para un intervalo temporal de *lap* segundos que se parametrizará en el script. En este ejemplo se ha tomado 0.01 segundos y en general tiene que ser menor que la duración de las muestras implicadas.

Podemos asumir que todos los clips de entrada son mono, i.e., tienen un solo canal (se lanzará error en caso contrario). Para hacer el solapamiento prepararemos los clips previamente: aplicamos la envolvente de fadeout a los clips `head` al final (en los últimos *lap* segundos) y la envolvente de fadein a los clips `tail`, al principio. Esta operación debe hacerse una sola vez para cada clip (no se hace dinámicamente) ya que modifica los samples del clip original. Después, durante la ejecución, en el `Update`, para solapar las muestras el clip de `tail` comenzará a reproducirse *lap* segundos antes de que termine `head`.

Para aplicar las envolventes pueden reemplazarse los fragmentos afectados de cada clip (véase Unity documentation):

- El método `AudioClip.GetData` permite obtener un fragmento a partir de una posición dada (`offsetSamples`)
- Después se aplicará la envolvente correspondiente a ese fragmento.
- El método `AudioClip.SetData` permite reinsertar el fragmento obtenido al clip original en la posición correspondiente.

Modificaremos el script `Slice.cs` incluyendo una atributo público *overlapTime* para parametrizar el tiempo de solapamiento y dos métodos `FadeIn(AudioClip clip)`, `FadeOut(AudioClip clip)` para aplicar las envolventes correspondientes a un clip dado de audio. El método `Update` lanzará los clips (el segundo con el retardo correspondiente).

Para completar el efecto, añadiremos al final un sonido aleatorio de casquillo.

Opcional: la carpeta también contiene muestras *loop-[1-3]* con loop de disparo (ráfagas), que pueden activarse con otra tecla. Pueden además incluirse otras mejoras:

- Para enriquecer el sonido pueden incluirse pequeñas variaciones aleatorias en el volumen y el pitch de cada muestra (haciendo las correcciones oportunas en la duración de los samples para sincronizarlos correctamente).
- Variar el comienzo de las muestras *casing* en la cola de los disparos, tanto en el simple como en la ráfaga. En la ráfaga pueden incluirse pequeñas demoras temporales aleatorias entre disparos.
- Incluir una reverb parametrizada para simular el recinto en que se hacen los disparos.