

Sonido en videojuegos

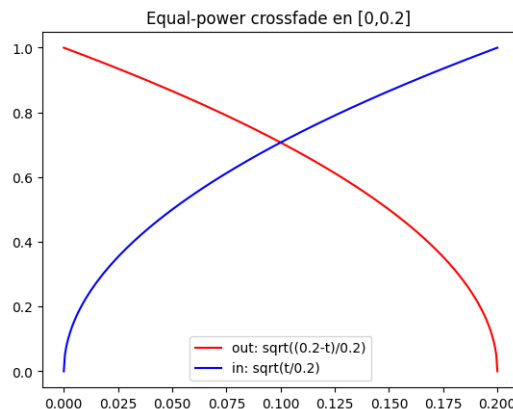
Unity Audio 3

El ejercicio marcado con **Evaluable** debe entregarse a través del CV. Se subirá un único archivo con el script pedido con el nombre de los alumnos **NombreApellidos1-NombreApellidos2.cs**

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 para solapar y mezclar las muestras para suavizar la transición entre ellas. La carpeta `muestras/disparo` contiene:

- `head_[1-2]`: fragmentos iniciales de disparo, con 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 las 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 azul. En la leyenda se muestra la función que genera cada una de las curvas, para un intervalo temporal de 0.2 segundos (este valor se parametrizará en el script). En nuestro caso, aplicaremos la envolvente de fadeout en el clip `head` al final del mismo (en los últimos 0.2 segundos) y la envolvente de fadein al clip de `tail`, al principio. Para solapar las muestras el clip de `tail` comenzará a reproducirse 0.2 segundos antes de que termine `head`. Para aplicar las envolventes pueden reemplazarse los fragmentos afectados. Para ello, el método `AudioClip.GetData` permite obtener un fragmento dado, aplicar la envolvente correspondiente y volver a incorporar dicho fragmentos al clip original con `AudioClip.SetData` (véase Unity documentation).

Modificaremos el script `Slice.cs` incluyendo una atributo público `overlapTime` para parametrizar el tiempo de solapamiento (0.2 en el ejemplo de arriba) y dos métodos `FadeIn(AudioClip clip)`, `FadeOut(AudioClip clip)` para aplicar las envolventes correspondientes a un clip 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.
2. Utilizar la misma técnica de crossfades del ejercicio anterior e implementar otro script para hacer fundidos entre pistas de música. El script cargará un conjunto de archivos de música de una carpeta previamente seleccionada. Las pistas se almacenarán en un array de clips de audio y se comenzará a reproducir una de ellas elegida aleatoriamente. Cuando se pulse la tecla '1' seleccionará otra pista de música y hará un fundido con un tiempo especificado para reproducir la nueva pista.
¿Puedes hacer esto mismo cargando las pistas en streaming en vez utilizar clips de audio?