Game Music

# Loop based

A piece of music is limited in time, it lasts only so many minutes. A game level lasts as long as the player is playing it, if it’s 2 minutes, or 2 hours. So the music must be able to cover any amount of time.

That is why loops are used. But there are two ways of creating loops.

* A piece consisting of 2 minutes is looped over from start to end. To create variations, several music files can be played seamlessly after each other.
* A longer piece, for example 10 minutes, or with a configuration file telling the software where the music can restart. This music file then would contain all the variations of the theme.

# Horizontal re-sequencing

A piece of music is constructed as a series of chords. These chords can be played in different ways,with different rhythms, the notes played in a different order. In horizontal re-sequencing, several versions of these chords are created, for different situations.

For example:

Chord A:default(verion1) - Chord A:default(version2)

Chord A:action(version1) - Chord A:action(version2)

Chord B:default(version1) - Chord B:default(version2)

Chord B:action(version1) - Chord B:action(version2)

These pieces only consist of a few measures. There are different versions for each circumstance (default, action, ...), to allow variation. When a player moves from default game play to action, the Chord A:default can be followed by Chord B:action in a seamless way, when the action continues, it goes to ChordA:action, if the action then is finished the progression goes to ChordB:default.

In this way of working there is only one layer of music, where the different parts can be re-ordered (re-sequenced) to the need of the game.

This means a lot of short chunks of music, where it can quickly adapt to different circumstances. One big drawback is that long melodies are not possible, because they can span different chunks of music. This limits the creativity and possibly the quality of the music.

# Vertical re-sequencing

Here we work as we’ve started now. Several layers of music are created for different circumstances. But there are two types:

## Additive

The layers are written to be added or subtracted when the gameplay requires it. Different volume settings of the different layers, fade-ins, fade-outs can accommodate for the build up or decrease a certain mood in the game.

## Interchangeable

Here we work in several layers too, but each layer can be used separately. In other words, they can be added on top of each other, but also interchanged. This adds more variation, but does present quite some challenges for the composer. It’s a real puzzle to make.

# Stingers

These are separate pieces of music to signal some gameplay events like:

* Victory stinger: for example when a combat ends successfully
* Defeat/Death stinger: the opposite, when the player loses/dies
* Transition stinger: for example transition from exploration to combat mode
* Hint stinger: for example when a certain object is to be found, or a character to have a dialogue with.
* Prize stinger: for example when an object is found, or when the player gains a new ability/power

# Cinematics and Cut Scenes

These are essentially animations, video’s. The music does not have to be limited by it’s ability to loop. This music can be composed in a single layer of music, since there is no need for interactivity.

# Scripted Events

These can be compared with cinematics and cut scenes, The difference is in the graphics, the video is not rendered, but a scripted animation where the player has not the ability to interact.