**APOLLO**

**MUSIC DESIGN  
SPECIFICATIONS**

VERSION 4.0

**Revision Tracking**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Revisions | By Whom | Date |
| 1.0 | Document Created | Nathan Grigg | 7.26.09 |
| 2.0 | Updated version | Brian Pamintuan | 9.08.09 |
| 3.0 | Updated version | Nathan Grigg | 10.19.09 |
| 4.0 | **Updated version** | **Nathan Grigg** | **12.01.09** |

Contents

[Revision Tracking 2](#_Toc243737496)

[1. Quest Music 4](#_Toc243737497)

[2. City Music: 7](#_Toc243737498)

[3. Investigation Screen (Batcomputer) Music: 8](#_Toc243737499)

[4. Interrogation Music: 9](#_Toc243737500)

[5. Combat Music 12](#_Toc243737501)

# 1. Quest Music

#### 1.1 Adaptivity

Each quest is assigned a Quest Theme that begins when the player enters a quest area. Quest themes are designed to adapt to the following inputs from the game:

*Awareness States (AI):*

Quest themes adapt to significant changes to AI awareness states. If the AI becomes aware of the player, the music should increase in intensity. Once the AI is engaged in combat with the player, the music intensity should shift further upward.

*Stealth (Player):*

Thirdly, the player can choose to operate stealthily by remaining in shadows, moving quietly, etc., which should affect the musical arrangement.

*Fear (AI)*

Apollo incorporates a fear level for the AI, which in turn affects the AI’s behavior. Since AI fear level is directly influenced by player actions, the player would benefit from musical feedback at a threshold where you would notice a significant shift in the AI’s behavior.

#### 1.2 AwarenesS:

There are three distinct arrangements per quest theme. Each arrangement variation corresponds to a specific AI Awareness state, increasing in intensity to match the AI’s activity level.

Base (Unaware)

Aware

Combat

Titles like NOLF2 and Tron 2.0, which were scored with DirectMusic, set global music states from a game level's most active AI. If you disturbed even one AI, raising it to Alert, the music would follow suit. Likewise if you were in combat and wiped out everyone within earshot, the music would follow a global AI state change back to Idle. This is a simple and effective framework that would be a good starting point for Apollo, particularly in quests.

#### 1.3 Stealth:

Stealth is player-driven. If the player successfully hides in shadows, he is given user feedback of a blue glow on the player model. The “Base” music arrangement is designed to shift to accompany this visual feedback. Two full mixes that share the same musical form smoothly fade between each other depending whether the player is visible or in shadow (blue glow). This allows a long looped piece to cycle through all of its content even if the player goes in and out of stealth frequently.

Aware

Combat

Stealth

Base (Unaware)

Player moves into/out of stealth

Since the AI’s awareness level hinges on whether the player is visible, “stealth” arrangement variations are not necessary for the “Aware” and “Combat” states.

*Stealth Moves:*

Stealth moves for Apollo would be supported by stingers. Stingers can include randomized variations to increase variety. It may be necessary to place limits on how many times these stingers can play in a single quest, since the moves will also be supported by sound effects, and the novelty of the stingers may wear off. A current list of stealth moves below:

* Shadow/Cover Dive
* Shadow Grab
* Shadow Punch
* Shadow Kick
* Betarang
* Dash (not exclusively for stealth but music stinger would only work in stealth mode)

#### 1.4 FEAR

When the AI reaches a certain threshold of fear as a result of player actions, a long, subtle stinger or temporary layer in the arrangement will play over the background music as a signal to the player that the fear shift has occurred. The specific content would be subject to change based on the arrangement, as it would need to be applied to every awareness-based arrangement in the Quest Theme, but a consistency of timbre across all themes--such as a reverberant choral effect or an eerie string texture--would keep its purpose clear.

Player triggers AI Fear threshold: trigger subtle stinger or layer over the arrangement that slowly subsides over time.

>Quest Arrangement (Base/Aware/Combat)>

#### 1.5 Manual Overrides

We would need to be able to “manually” override the state structure at times (via music command scripting in the world), like if we hit an in-game cinematic, or pertinent in-game dialogue that may require a compositional change rather than a simple channel duck.

# 2. City Music:

#### 2.1 Description

When the player is en route to quest areas, or simply cleaning districts of ambient crime, the overall mood of the game needs to be supported with music. I would argue, however, that it shouldn’t be wall-to-wall music. One option would be a “welcome mat” approach where entry into each district plays a short theme (20 seconds or less) that subsides into the city ambience dynamically over its duration. Each district would have a trigger around its limits to play its theme once the player crosses its boundary. We may also consider putting in a “retrigger clause” for these--lock each district theme trigger for 20 seconds after the player hits it once, so he can’t play clumsy DJ by hopping back and forth across district lines.

#### 2.2 District Themes:

* TBD

#### 2.3 Emotions:

Each Gotham district has an emotional state that changes throughout the game based on the level of criminal activity. The player can affect a district’s emotion for the better by taking out a boss or cleaning the area of ambient crime. There are three City Emotion states: Happy, Neutral, and Angry, but the music supports them with only two modes.

* Angry
* Happy/Neutral

Using the same variation for Happy and Neutral emotion states will keep content requirements in line while supporting the overall atmosphere of the game, which is still dark and urban even when it’s not in complete chaos.

#### 2.4 Ambient Crime:

Ambient crime may require two music features:

* A simple stinger to alert the player that a crime is taking place within a certain radius (this simply plays over any existing music).
* A randomized collection of simple combat themes for when the player is fighting the ambient criminals. This would interrupt any district themes being played. For more information on combat music structure, see 5. Combat.

# 3. Investigation Screen (Batcomputer) Music:

#### 3.1 Basic Structure:

* A looping background track with possible variations in form and arrangement to decrease repetition. There should be motion in the arrangement, but it can’t dominate the scene.
* An ending piece to play as the player selects the next mission (beat or bar boundary).

#### 3.2 “Boss du Jour” Variations:

* The Batcomputer’s track changes whenever new bosses are unlocked, allowing the main figure on the player’s mind to permeate the mood as he’s selecting his next mission.

# 4. Interrogation Music:

#### 4.1 Adaptivity:

Interrogation scenes require tight scoring to match the intensity changes on a moment-to-moment basis.

The main game components affecting the score are 1) the player input and 2) the resulting affects of that input on the interrogated AI's "heart rate" - which is divided into four distinct zones:

*Player Input:*

* Ask Question
* Light Attack
* Heavy Attack
* Special Attack
* Exit

*AI Heart Rate Zones:*

* Confident (cocky - sample answer: "You won't kill me. You haven't got the guts!")
* Nervous (distressed but defiant - "I don't know anything, and I wouldn't tell you if I did!")
* Scared (AI answers truthfully - "Okay, Okay... I'll tell you everything!")
* Panicked (Maximum heart rate. He has been pushed too far, and is now talking nonsense. "Please don't kill me!")

The basic content structure for interrogation should include a series of looping pieces for questions and answers. Minimally, we'll want one question and one answer piece for each AI Heart Rate zone, increasing in intensity as the heart rate goes up, but several variations of each would be ideal.  Transitions from question to answer and back would be done with immediate cross-fades.  Answer pieces would cross-fade in with the AI's answer dialogue.  Question pieces would cross-fade in after player actions.

Question: Confident

Answer: Confident

Answer: Nervous

Answer: Scared

Answer: Panicked

Question: Nervous

Question: Scared

Question: Panicked

#### 4.2 Player Actions:

To affect the AI's heart rate, the player has three options: a Light Attack (such as a shake), a Heavy Attack (such as a strike in the face), and a Special Attack - which is a choke or throttle with a user-defined length.  Light and Heavy attacks are punctuated with one-shot stingers that play over the top of the Question/Answer material.  Underneath the stinger (which can be fired off from an action inside a Wwise event), a cross-fade to the appropriate Question piece will occur based on the resulting heart rate.

Answer: Confident

Heavy Attack

Question: Scared

Question: Nervous

Question: Confident

Light Attack Attack

Heart Rate

Question: Panicked

#### 4.3 Special Attacks:

Special Attacks work differently.  The player controls how long the special attack piece plays by how long he chokes the AI, so instead of an overdubbed stinger, a special attack piece quickly crossfades in when the player initiates the attack and sticks around until A) the player releases the Special Attack, or B) the AI passes out from stress. The moment the player releases the Special Attack, the music cross-fades to the appropriate Question theme for the current heart rate.

Answer: Confident

Special Attack

Q: Panicked

Q: Confident

Q: Nervous

Duration increases heart rate

Q: Scared

Rather than a loop that sounds the same no matter what the heart rate is, the Special Attack content is best written as a single intensity-building crescendo that lasts as long as it takes to subdue an AI from its most confident state.  The crescendo can then be divided from there into four sequenced segments, one for each heart rate zone:

Special Attack: Confident

Special Attack: Nervous

Special Attack: Scared

Special Attack: Panicked

When the player initiates a Special Attack, the material cues from the appropriate segment for the current heart rate.

Knowing that the heart rate meter is more granular than this, an ending segment for the AI passing out from stress needs to be created separately from the Special Attack crescendo. This would be a simple cross-fade to a linear file matching the AI's pass-out animation.

# 5. Combat Music

#### 5.1 Basic Combat Themes

A combat arrangement at its most basic level contains a single looping track made up of shorter segment files that are either sequenced (for a fixed form) or randomized (for a variable form) depending on the content. The track is designed to play as long as the global awareness state = Combat. When the awareness state is lowered (due to the player and AI no longer being engaged in combat), a transition to the lowered state arrangement occurs. This could be an ending, a crossfade, or specific transition piece cued on the next musical boundary, depending on what the theme calls for.

#### 5.2 Enhanced Combat Themes

In addition to basic combat themes, there should be the option to score specific fights in the game--boss fights or other pivotal scenes--in more adaptive detail. Enhanced Combat Themes *shouldn't be used too often,* as adaptive music features become predictable when they are engine driven and always on. By design, we save them for the best moments, keeping the player immersed in the world, rather than the underlying form.

*Player Move Support:*

Enhance Combat Themes support most direct combat moves with stingers. This is a bit like the animation-cued stingers in select combat sequences in Condemned 2, only "smarter", in the sense that the Wwise platform allows us to cue these stingers on the next defined musical boundary such as "The next beat in the combat track." This allows us to synchronize stingers to rhythmic pieces. In testing, we can determine whether it's more critical to have some moves in better sync with the player animations (i.e. "immediate") or in better sync with the rest of the music ("next beat", "Next bar" etc.) and set each one appropriately.

Here's a list of Direct Combat moves in the game we would want to support in an Enhanced Combat Theme with stingers:

* Punch
* Kick
* Grab/Disarm
* Block
* Counter
* Combo

*Special Move Support:*

* Bear Hug/Choke Hold - music shifts to a tension building crescendo while the player either tries to button-mash or special-move himself free. Combat track resumes once the player has broken free. Special moves would be rewarded with a stinger timed with the animation.
* Grapple Gun - similar form to Bear Hug/Choke Hold - music shifts to tension building crescendo for tug-of-war button mashing sequence. Special "yank" move is rewarded with a stinger. Combat track resumes once the sequence ends, either through button mashing, special moves, or failure.
* Bonebreaker - the bonebreaker is handled like an ear-ring effect for music: the combat arrangement is filtered to a low-bandwidth for the duration of the "x-ray" effect, mixed with a high-pitched string tone, and a stinger fires off for the break. During the stinger, the filter sweeps back to full-bandwidth.

*Randomization:*

In addition to the randomizing of form that can potentially be applied to the combat track, the stingers can be randomized as well, meaning each move type (Punch, Kick, etc.) can be assigned a pool of variations to pick from when the move is activated, minimizing repetition.

Wwise does not currently support a "no repeat" condition for randomization of variations in triggered stingers. A "no repeat" randomization would allow similar moves in potentially rapid succession to not retrigger the same variation twice. This feature would ideally be addressed either by Audiokinetic or by us before production starts as it would reduce repetition in stinger polayback.

#### 5.3 Stun State and Finishing Moves

If Apollo implements an enemy stun/finishing move system for combat, this should be supported musically as well. When the player stuns an enemy, the combat track should transition to a linear tension-building piece that ramps during the duration of the stun effect.

If the player fails to finish the enemy, the enemy gets up and the combat piece resumes once the stun segment is finished playing.

Player stuns enemy

Combat Arrangement

Stun Arrangement

Combat Arrangement

If the player successfully performs a finishing move on an enemy mid-stun, an ending interrupts the stun segment, plays a finishing segment and cues up the arrangement assigned to the current AI awareness state.

Player stuns enemy Player finishes enemy

Combat Arrangement

Stun Arrangement

FinishingMove

AI Awareness Arrangement

In fights with multiple enemies, the feature would probably be best used only on the last AI standing.