**APOLLO**

**MUSIC DESIGN  
SPECIFICATIONS**

VERSION 2.0

**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 |

1. **Music Direction**
   1. *The Style* 2
   2. *The Medium* 2
2. **Quest Music**
   1. *Adaptive Structure* 3
   2. *Theme Modes* 3
   3. *Awareness Mode* 4
   4. *Stealth/Fear Mode* 4
   5. *Combat* 5
   6. *Manual Overrides* 5
3. **City Music**
   1. *Description* 6
   2. *District Themes* 6
   3. *Emotions* 6
   4. *Ambient Crime* 6
4. **Investigation Screen (Batcomputer) Music**
   1. *Basic Structure* 7
   2. *“Boss du Jour” Variations* 7
5. **Interrogation Music**
   1. *Adaptive Structure* 8
   2. *Player Actions* 9
   3. *Special Attacks* 9

**6. Combat Music**

*6.1 Basic Combat Themes* 11

*6.2 Advanced Combat Themes* 11

*6.3 Stun State and Finishing Moves* 12

**7. Cinematics**

*7.1 Basic Structure* 14

*7.2 Production Process* 14

# 1. Music Directon

***1.1 The Style***

The dark, brooding, cinematic scores of Batman Begins and The Dark Knight are the closest existing fit to the aesthetic direction of Apollo's game design. The combination of gritty realism, theatrical characters, a majestic urban environment, and a self-sacrificing protagonist intuitively pushes the music in this direction. These would be the first scores to inquire about licensing, and for our original content they provide an aesthetic core to work outward from.

* 1. ***The Medium***

The score is mixed-media: a combination of live and electronic sources, with an extensive use of sampling and processing. Ensemble recording for Apollo's original content is best managed in sections (strings, brass, etc.), recorded separately and mixed with other source in a digital environment, as a mixed media score such as this is not bound by a traditional orchestra/concert hall format. For Apollo, ensemble recording will be mandatory for at least brass and strings in order to be compatible with the film scores' production level. Aside from this, we expect that many percussion elements and textures will be sample based, and there will be a lot of creative use of sources from synthesis, non-traditional instrument soloists, sounds from field recordings, and more.

**2. Quest Music**

***2.1 Adaptive Structure***

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 should first and foremost be adaptive to AI Awareness states. The four AI states in Apollo are:

* Sleeping
* Relaxed (AI Idles, Patrols, etc.)
* Suspicious (AI Investigates disturbances with curiosity)
* Alert (AI Investigates disturbances with urgency)

*Fear States (AI):*

In addition to Awareness States, Apollo also incorporates a set of Fear States for the AI. The design docs for Fear currently include the following four states:

* Calm
* Nervous
* Scared
* Panicked

*Stealth (Player):*

Thirdly, the player can choose to operate stealthily by remaining in shadows, moving quietly, etc. affecting the game AI’s ability to sense the player’s presence.

*Combat:*

Finally, overriding Awareness States, Fear states, and Stealth, there’s Combat, which should be supported with a distinct, high-energy arrangement.

* Combat

***2.2 Theme Modes:***

Each Quest theme provides several arrangement variations to support AI and player-driven state changes. In theory, we could make the music support the whole complex matrix of states (Alert/Scared, Suspicious/Panicked, etc.), but in practice I think they would conflict with one-another. A clearer way to approach AI Awareness, Stealth, and Fear in a single theme would be to define two distinct “modes” for the theme: 1) Awareness and 2) Stealth/FEAR.

***2.3 Awareness Mode:***

Awareness is the default mode for the Quest Theme. Each arrangement variation corresponds to a specific AI Awareness state, increasing in intensity to match the AI’s activity:

Alert

Sleep/Relaxed

Suspicious

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.

***2.4 Stealth/Fear Mode:***

“Stealth/Fear mode” overrides the default Awareness mode, and is activated by the player simply moving into a stealth position. Once the Quest Theme moves to “Stealth/Fear” mode, shifts in intensity are now controlled by the AI’s Fear states, rather than their Awareness states. This way the player’s style, whether stealthy or in-your-face, is rewarded with a unique experience.

Sleep/Relaxed

Calm/Nervous

Combat

Panicked

Scared

Suspicious

Alert

Player moves into/out of stealth

Transitions in and out of stealth need to sound smooth and non-repetitive. At the theme’s base level (sleep/relaxed – calm/nervous) it may make sense to “remix” multiple stems on the fly depending on which mode the player is in, or at least have two bounced full mixes to fade between that share the same form. This would allow a longer looped piece to cycle through all of its content even if the player goes in and out of stealth frequently.

*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/fear mode)

***2.5 Combat***

Finally, the two paths would merge for a common combat theme. Combat trumps all AI states in either mode. For more information on combat music structure, see *6. Combat Music.*

***2.6 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.

**3. City Music:**

***3.1 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.

***3.2 District Themes:***

* TBD

***3.3 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.

***3.4 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 *6. Combat Music.*

**4. Investigation Screen (Batcomputer) Music:**

***4.1 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).

***4.2 “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.

**5. Interrogation Music:**

**5.1 Adaptive Structure:**

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: Panicked

Question: Scared

Question: Nervous

Answer: Panicked

Answer: Scared

Answer: Nervous

Answer: Confident

Question: Confident

**5.2 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.

Question: Panicked

Heart Rate

Light Attack Attack

Question: Confident

Question: Nervous

Question: Scared

Heavy Attack

Answer: Confident

**5.3 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 cross-fades 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.

Q: Nervous

Q: Confident

Q: Panicked

Special Attack

Answer: Confident

Q: Scared

Duration increases heart rate

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: Panicked

Special Attack: Scared

Special Attack: Nervous

Special Attack: Confident

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..

**6. Combat Music:**

***6.1 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.

***6.2 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:*

These are moves that call for changes to the combat arrangement itself:

* 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.

There are a few limitations to how Wwise currently handles randomization of variations in triggered stingers that would ideally be addressed either by Audiokinetic or by us before production starts:

* A "no repeat" clause like we implemented for Condemned 2's stingers. This allows similar moves in potentially rapid succession to not retrigger the same variation twice.
* The ability to flag certain variations to play over certain sections of the combat track. This ensures the stingers are always harmonically compatible in cases where the combat track cycles through distinct key/chord changes.

***6.3 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

Combat Arrangement

Stun Arrangement

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

Player stuns enemy Player finishes enemy

Combat Arrangement

Post-Combat State Arr.

Finishing Segment

Stun Arrangement

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

**7. Cinematics:**

***7.1 Basic Structure:***

Apollo will include many cinematics to advance the game’s story. Apollo’s cinematics are game-engine driven, so it’s imperative that their soundtracks are able to fire off at their absolute start points with no timing inconsistencies.

To economize on memory, reduce synchronization risks, and maintain a streamlined collaborative process between music and sound design, all music and sound effects for each cinematic are to be mixed into a single, interleaved audio file. The only audio that remains separate for cinematics is dialogue, so that individual lines can be swapped out for localization purposes without having to remix the rest of the audio.

Cinematic audio tracks should automatically fade out all existing music and sound effects in the game, and transition seamlessly back to the in-game audio once they are finished.

A comprehensive list of cinematic scenes for Apollo is TBD.

***7.2 Production Process:***

An ideal music production process for Apollo Cinematics would be something similar to Condemned and Condemned 2, where all dialogue and animations get to final timing in a solid pass before textures and effects are finalized. These black and white “playblasts” (with synchronized dialogue included for reference) can be submitted to the composer in a linear digital video format (AVI or Quicktime) for a scoring pass. The composer can then start scoring with confidence that there won’t be radical timing shifts along the way, but still get in the process early enough to iterate and avoid a crunch.

Music and sound effects passes can be done separately, and some sound effects are dependent on final textures and visual effects to design properly. A final mix pass of all the elements--along with any necessary mastering treatment--comes last. This is done in sync to video captures of the final in-game cinematics.