Vertonung von Computerspielen

TONSEMINAR WS13/14

MICHAEL DMOCH
MD079

Interaktive vs lineare Medien

- Problem: Wir wissen (fast*) nicht
  - was
  - wann
  - in welcher Reihenfolge
  - wie oft

  der Spieler tun wird

*fast:
es gibt Wege, dazu später mehr

Bild: Quelle 5

Unterschiede zu linearen Medien

- andere Herangehensweise:
  - Wegen ihrer Nonlinearität erfordern Vertonungen von Computerspielen eine ganz andere Herangehensweise als die von linearen Medien
- begrenzte visuelle Referenz:
  - Der Sounddesigner „fliegt blind“, er hat nicht wie beim Film einen Rohschnitt auf den er seine Sounds anlegen kann
- höherer Aufwand:
  - Eine überzeugende Vertonung erfordert viel Aufwand
- 50-80% Implementierung:
  - Je nach Art des Projektes liegt 50 bis 80% des Arbeitsaufwandes des Sounddesigners in der Implementierung (6)

Flexible Vertonung

Lösung: Die Vertonung muss flexibel auf das Spielgeschehen reagieren können

- aufwendige Implementierung
- Realismus wird bei der Implementierung abgestimmt, nicht im Tonstudio
- aufwendige Komposition
- begrenzte visuelle Referenz

Was kann/soll Vertonung erreichen?

Vor allem

- Immersion!
  - das Eintauchen in die Spielwelt
  - Identifikation mit der Spielfigur
  - alles um sich herum zu vergessen
- Stimmungen/Gefühle erzeugen bzw. verstärken

aber auch

- Kontinuität und Struktur schaffen
- Ästhetik prägen
- Bedeutung verleihen
- Assoziationen herstellen
- als zusätzliche Wahrnehmungsebene fungieren
- motivieren

aus: Quelle 3

Was kann/soll Vertonung erreichen?

Zu diesem Zweck sollen

- die Vertonung stimmig sein zum Spielgeschehen und User Input
- der Soundtrack musikalisch schlüssig sein / keine ungewollten Brüche enthalten:
  - Änderungen im Ablauf erfolgen stets im Takt
  - Fades und Übergänge nur zwischen harmonisch passendem Material
- sich wiederholende Sounds variieren:
  - Spiele werden Dutzende oder Hunderte Stunden gespielt, Aktionen oft wiederholt
  - Wenn man z.B. nur 2 Samples für einen häufigen Sound wie Fußschritte einsetzt, fällt das schnell negativ auf

Diegese

Diegese beschreibt, ob etwas in einer Szene stattfindet, also ob es auch die Charaktere hören, und nicht nur der Zuschauer/Spieler

- Szenische Vertonung (diegetisch)
  - Dialog
  - Geräusche
  - Atmo
  - Musik (z.B. Autoradio)
- Dramaturgische Vertonung (nichtdiegetisch)
  - Soundtrack
  - emotionale Geräusche (z.B. Low Frequency Effects -> Unwohlsein, Gefahr)
  - Erzähler
- Audification
  - Vertonung des Interface des Spiels (Buttons etc.)

Szenische Vertonung

- Ziel: (Pseudo-)Realismus
  - trotz unrealistischer Konzepte wie z.B. „Lebensenergie“ herstellende „Medikits“
- implementiert als 3D-Sound
  - Pseudo-Auralisation: Keine wirkliche Auralisation, sondern eine Annäherung mittels Variation von Richtung, Lautstärke, Höhendämpfung, Hallanteil, etc.
- virtuelles Mikro („Audio Listener“)
  - meist an virtuelle Kamera oder an Spielerfigur geknüpft
- virtuelle Klangquellen („Audio Source“)
  - in der Regel unmittelbar an Objekte im Spiel geknüpft

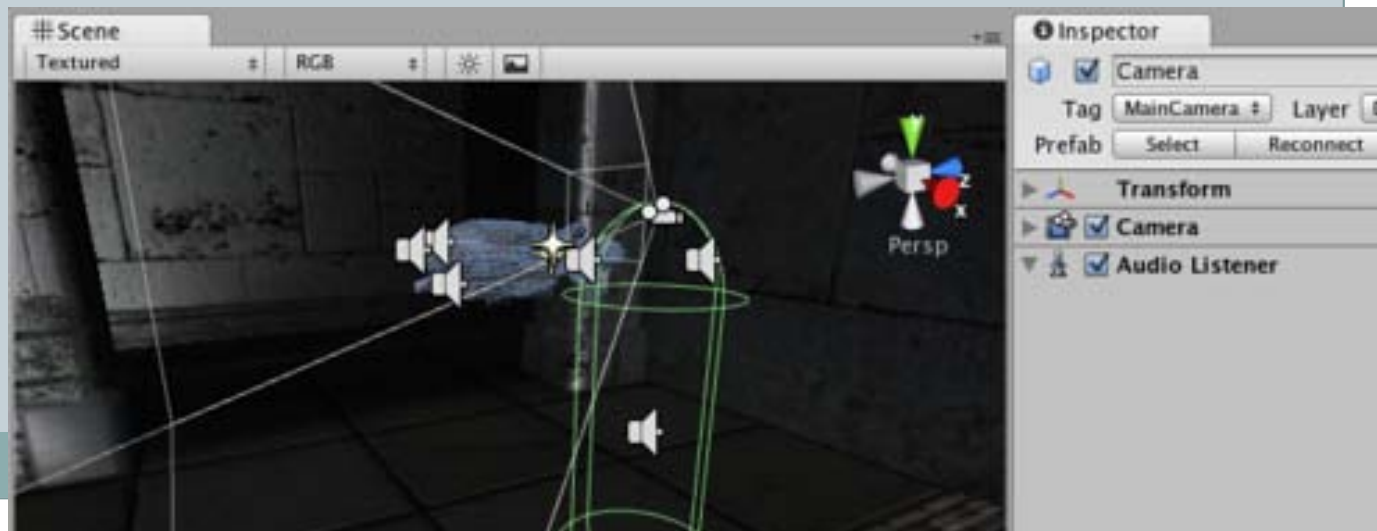


Bild: Quelle 7

Dramaturgische Vertonung

- implementiert als nicht-3D-Sound
  - z.B. Richtung aus der Musik kommt ändert nicht wenn Spielfigur Blickrichtung ändert
- in der Regel gescriptet, nur mittelbar an Objekte/Aktionen gekoppelt

Tools

- Game Engine
  - Editor für Erstellung des Spiels
  - Beliebte Game Engines: Unity, Unreal Engine, Cry Engine, ...
- Middleware / Authoring Tools
  - Editor für Implementierung der Sounds
  - Sounddesigner braucht keine Programmierkenntnisse
  - kann seine Implementierungen im Kontext ausprobieren, ohne auf einen Programmierer angewiesen zu sein
  - Beliebte Middleware: FMOD, DirectMusic, ISACT, XACT, Miles, Unreal Engine, Wwise
- PlugIns von Drittanbietern
  - z.B. Weather Machine
- proprietäre Lösungen (Eigenentwicklungen)
  - wenn man sich Features wünscht, die keine Middleware bietet

aus: Quelle 3

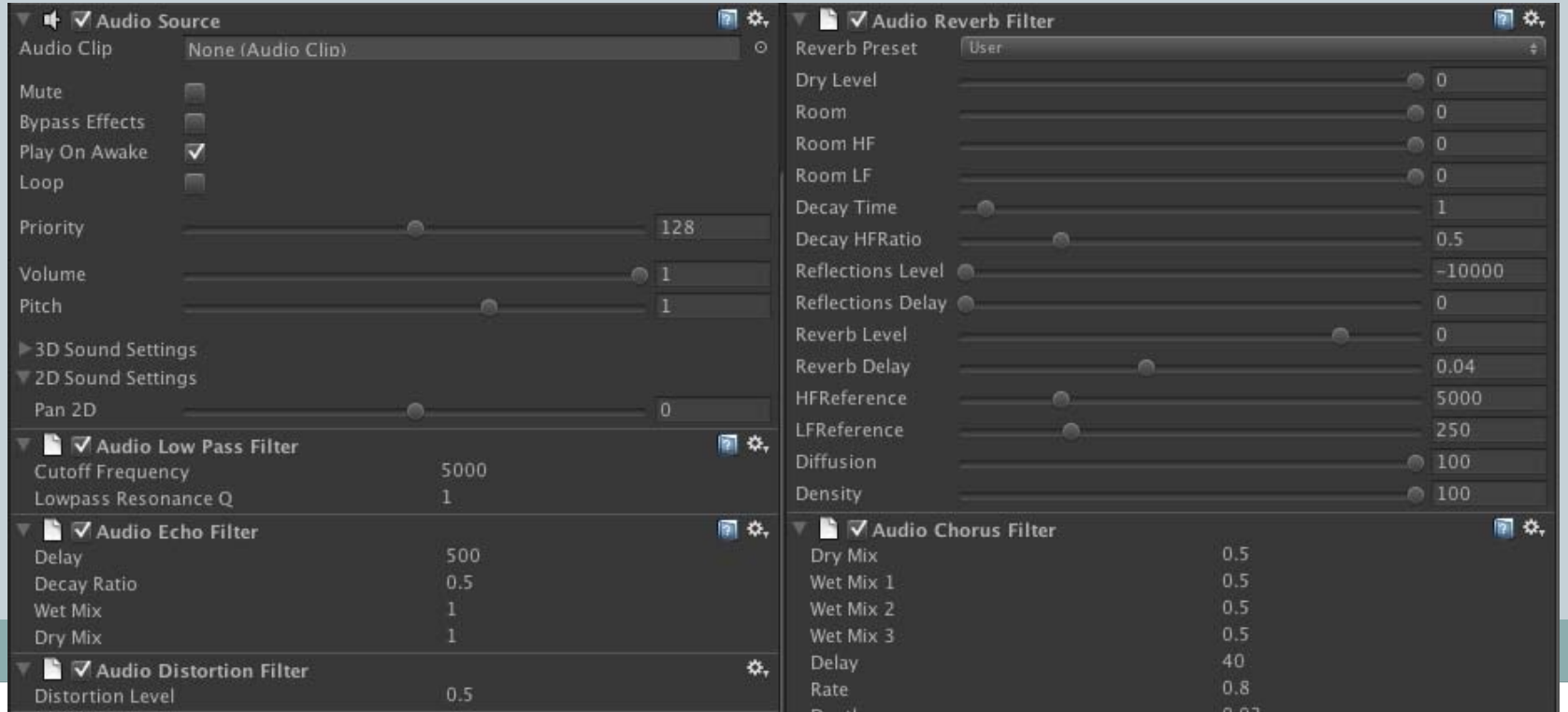
Beispiel für Middleware: FMOD

- Soundausgabe-Programmbibliothek + Middleware
- Cross-Plattform: Gleichzeitig für mehrere Plattformen entwickeln
  - Win, MacOS, iOS, Linux, PS2/3/PSP, Xbox, Xbox360, Gamecube, Wii
- Viele Formate: WAV, MP3, OGG, AIF, FLAC, Midi, MOD, ...
- Streaming von Festplatte oder Internet (und natürlich *Play from RAM*)
- Bietet zur Laufzeit DSP-Effekte:
  - Reverb (Nur Pro-Version)
  - LPF mit Resonanz
  - Delay/Echo
  - Chorus
  - Distortion
  - Doppler-Effekt
- Einbindung von VST-PlugIns zur Laufzeit möglich
- Engine Designer (für Fahrzeugmotoren), Weather System
- Editor für adaptive Musik

Unity: Inspector

- für Audio Sources lassen sich viele Klangparameter einstellen und dynamisch steuern
  - für Realismus
  - oder als Gestaltungsmittel

Bild: Quelle 7



Geräusche / Foley

- Pseudo-Realismus durch:
  - *Distance rolloffs* (Vol, LPF, Hall,...) = Steuerung abhängig von Distanz des virtuellen Mikrofons sowie der eingestellen Rolloff-Kurve
  - *Distance crossfade* (unterschiedliche Samples für nah und fern)
  - Simulation des Doppler-Effekts
  - Sample Layering
  - Abwechslung:
    - viele zufällig getriggerte, ähnliche Samples
    - zufällige Modulation (Vol, Pitch, ...)

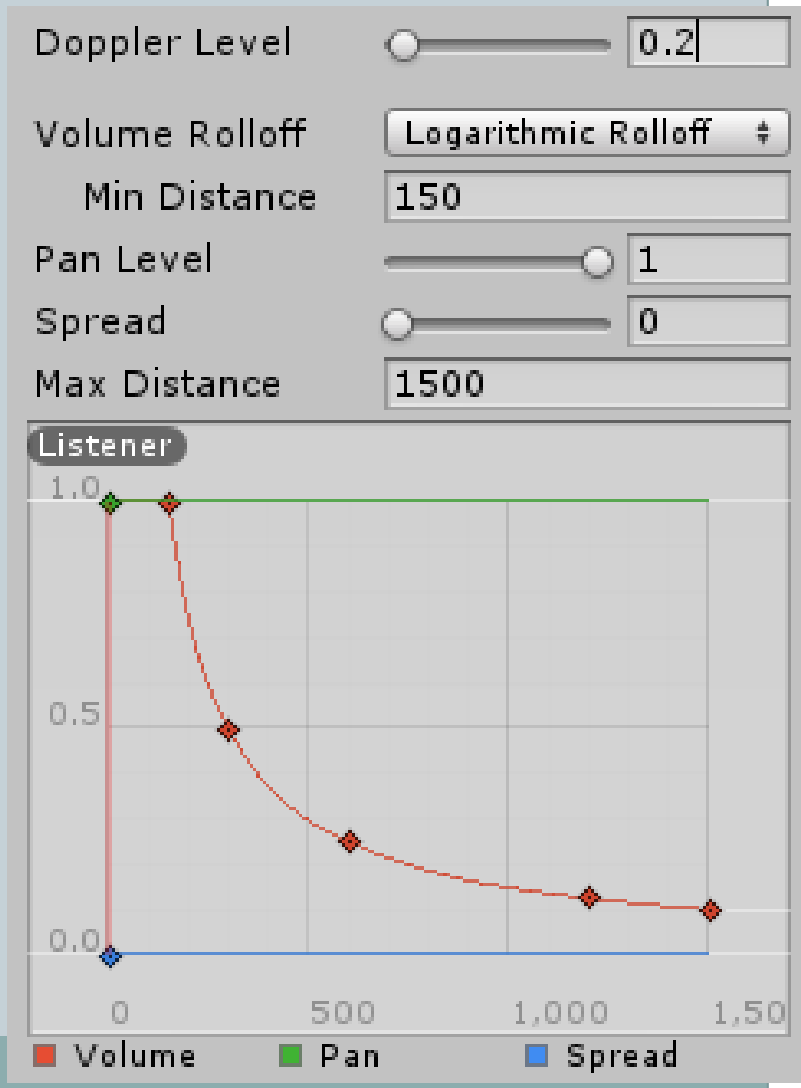


Bild: Quelle 7

Adaptive Musik

- PCM vs Midi:
  - Dank gesteigerter Leistung aktueller Systeme fast nur noch PCM
  - Obwohl Midi mehr Flexibilität bei adaptiver Musik ermöglicht
  - Midi noch relevant wo Systemleistung und Bandbreite knapp sind (Mobile Games, Browser Games)
- Musik soll
  - kohärent sein
  - offensichtliche Wiederholungen vermeiden
  - relativ kurzfristig auf Events reagieren können:
    - Musik reagiert zu träge: Musik und Spielgeschehen wirken zusammenhanglos
    - Musik reagiert zu schnell / bei geringfügigen Events: Gefahr des Mickymousing-Effekts
- Logik
  - steuert horizontale / vertikale Aspekte der Musik
  - abhängig von Events

Events

- Levels / Räume / Zonen (mittels Collidern)
- Flags, z.B.
  - Schalter aktiviert
  - Gegner besiegt
  - Spieler tot
- Parameter, z.B.
  - Lebensenergie des Gegners
  - Tageszeit
- Meta-Events (mehrere Flags/Parameter kombiniert) z.B.
  - hohe Erfolgsaussicht (Spieler stark + Gegner schwach)
  - Flugzeug stürzt ab (Sprit alle + schneller Höhenverlust)
- Game Engine kann auch auf Audio Events warten
  - Beispiel: http://youtu.be/-XuClagw6IQ?t=36s

aus: Quelle 3

Layering (vertikal)

Beim vertikalen Prinzip laufen mehrere Spuren (Stems) parallel im Loop.

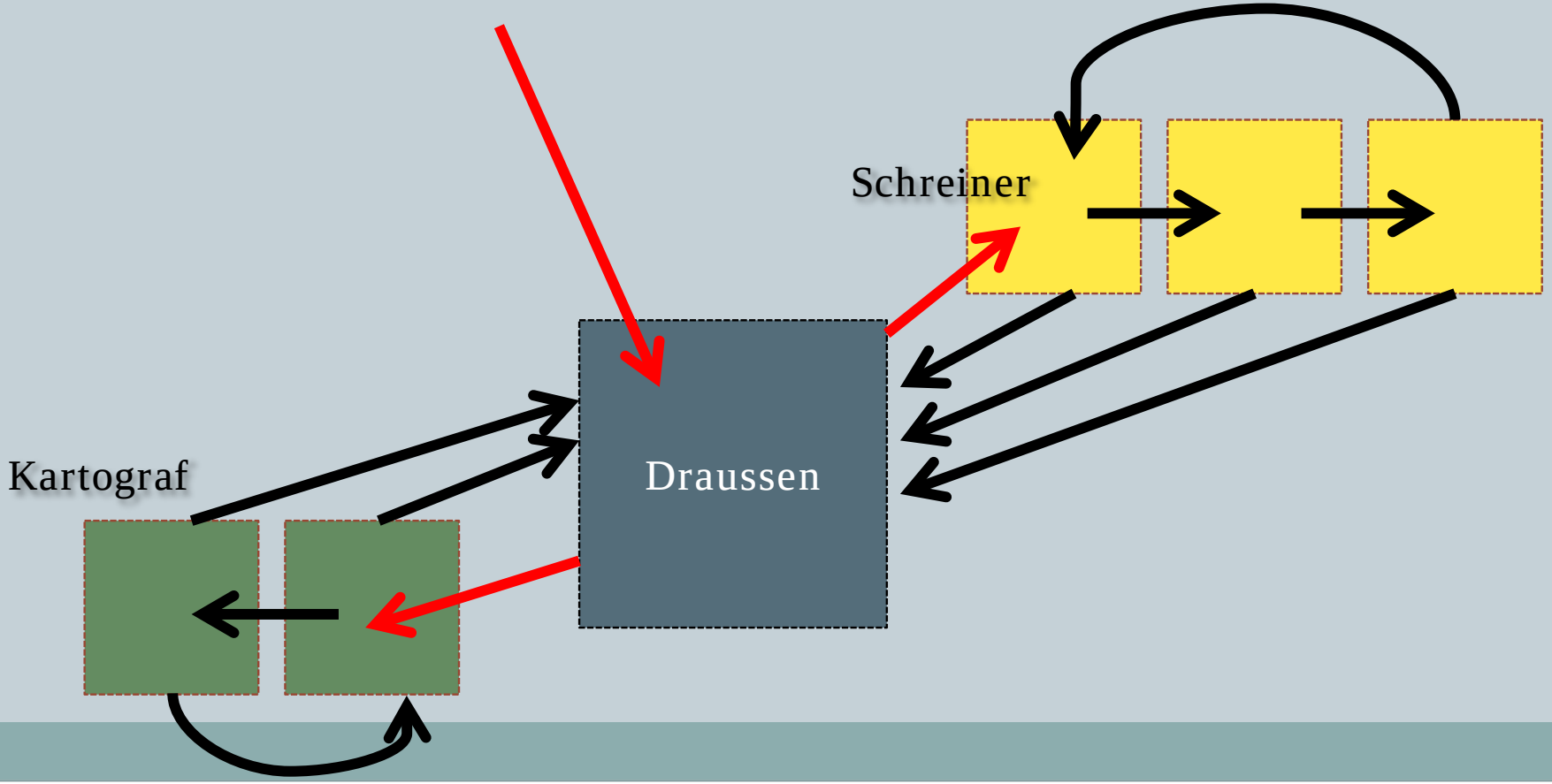
- Unmuting-Verfahren:
  - die Spuren enthalten zusätzliche/alternative Instrumente
  - werden dazugemischt (unmute), wenn z.B. musikalische Steigerung gewünscht ist
- Crossfade-Verfahren:
  - die Spuren bestehen jeweils aus kompletten Musikstücken mit gleicher Struktur und unterschiedlichem Feeling, die aber harmonisch stets kompatibel zueinander sind, so daß jederzeit problemlos zwischen ihnen übergeblendet werde kann

Branching (horizontal)

- Musik unterteilt in viele Segmente z.B.
  - Intro
  - Atmo
  - Suspense
  - Action
- Am Ende von jedem Segment entscheidet die Logik abhängig vom Spielgeschehen, ob
  - das Segment wiederholt wird
  - ein anderes Segment gespielt wird
- Segmente sind z.B. 8 Takte oder auch nur 1 Takt lang
  - kürzere Segmente -> Musik reagiert schneller
  - längere Segmente können einen größeren musikalischen Zusammenhang herstellen

Branching (horizontal)

- Musik unterteilt in viele Segmente
- z.B. bei Monkey Island 2: http://www.youtube.com/watch?v=7N41TEcjcvM



Spannungsbogen

- schlechte Übergänge ergeben einen schlechten Spannungsbogen

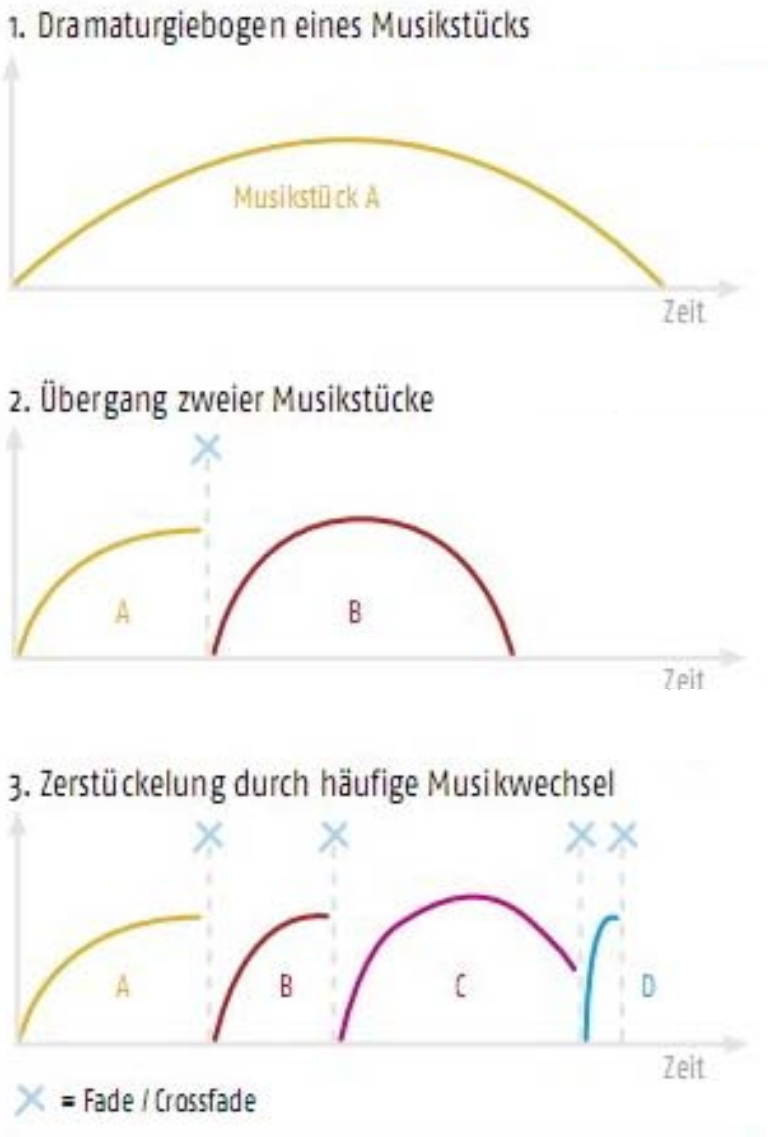
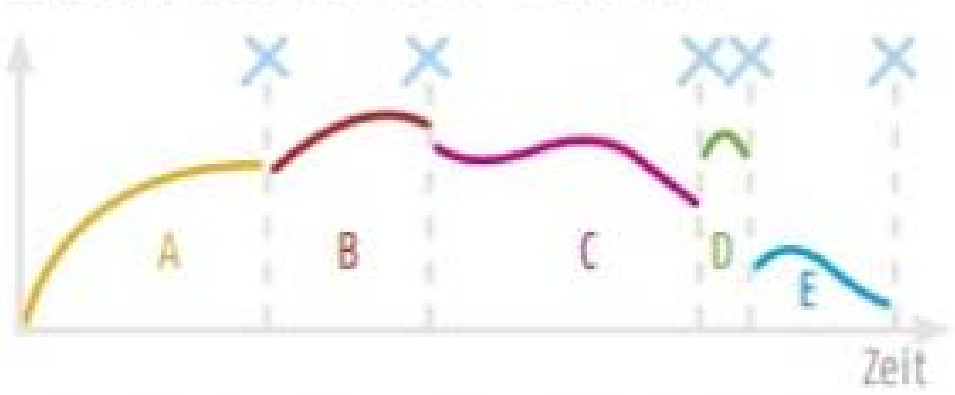


Bild: Quelle 5

Spannungsbogen

- geschickter Einsatz der gezeigten Techniken ergeben einen guten Spannungsbogen

horizontal



vertikal

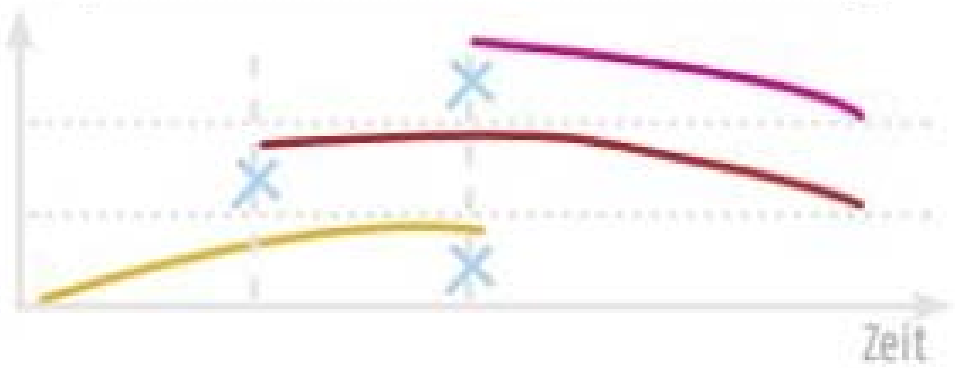


Bild: Quelle 5

Komplexität

- horizontale & vertikale Komposition ist kombinierbar
- dann wird es jedoch schnell sehr komplex und damit aufwändig
- was sich aber auch sehr lohnen kann ☺

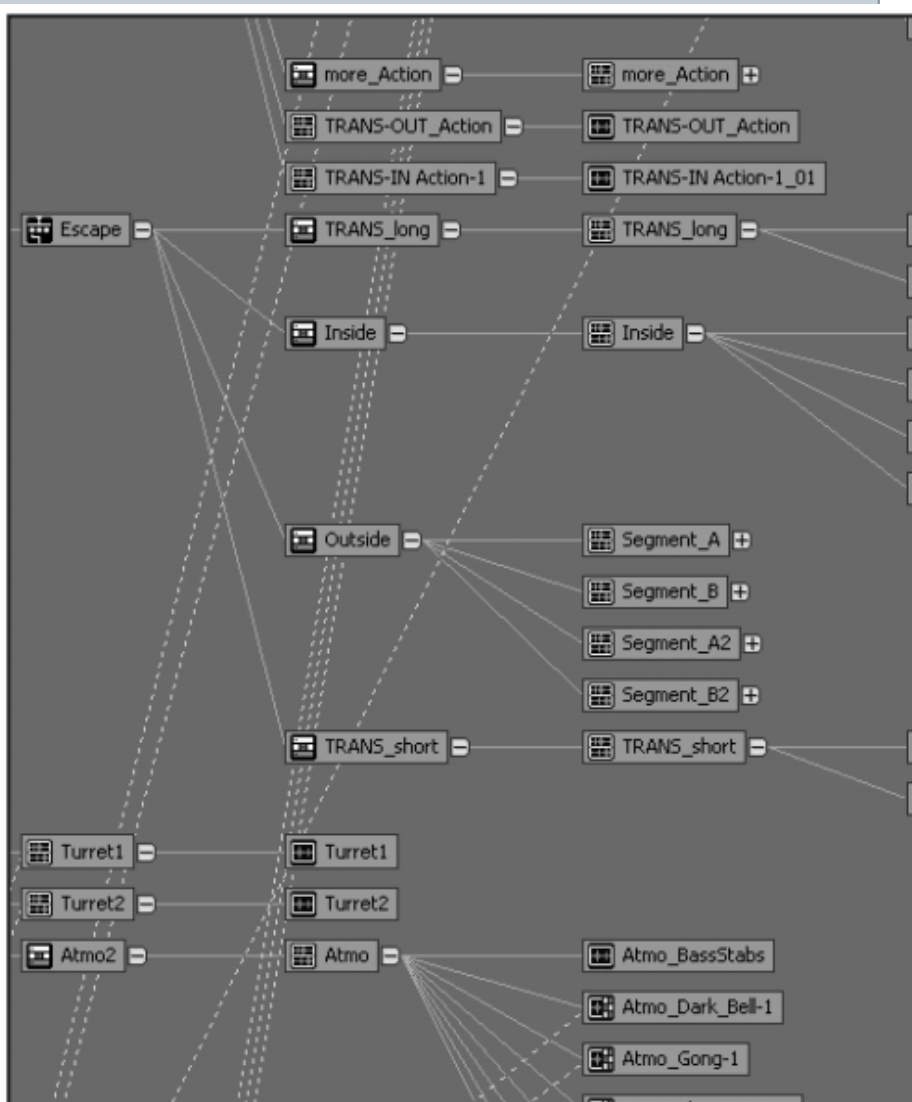


Bild: Quelle 3

Quellen

1. Matts Johan Leenders: „Sound für Videospiele“, 2012, Schüren Verlag Marburg
2. Oliver Szczypula & Jan Hofmann: „Game Sound“, 2008, VDM Verlag Dr Müller Saarbrücken
3. Benjamin Krause: „Adaptive Musik in Computerspielen“, 2008, Diplomarbeit HdM Stuttgart
4. www.gamasutra.com/view/feature/2263/gdc_2005_report_what_makes_music_.php
5. www.makinggames.de/index.php/magazin/575_interaktive_vertonung/2
6. designingsound.org/2009/11/rob-bridgett-special-the-role-of-an-audio-director-in-video-games/
7. www.unity3d.com