

# Einführung zu



Torsten Kammer  
@zcochrane

# Was ist das?

- iOS, OS X, alle anderen
- API für sehr teure Soundkarten
- Die keiner hat
- Und die es auch nicht für Mac gibt
- Aber auch sehr nützlich ohne das



# Das Problem



**PENG!**



**VROOM!**



**HINTERGRUNDMUSIK**



**MASCHINEN-  
BRUMMEN**

# Die Lösung

- ✦ OpenAL verwaltet Soundquellen und Zuhörer
- ✦ Gibt Ton aus, mit
  - ✦ Doppler-Effekt
  - ✦ Entfernungsabhängig usw.
- ✦ In Stereo, 5.1 und so weiter

# Was ist es nicht?

- ✧ Ersatz für Core Audio
  - ✧ basiert darauf
  - ✧ kann weniger, aber einfacher
- ✧ Allgemeine Soundverarbeitung
  - ✧ Speziell auf das eine Problem ausgelegt

# Das Prinzip



**PENG!**



**VROOM!**



**HINTERGRUNDMUSIK**



**MASCHINEN-  
BRUMMEN**

# Das Prinzip



**PENG!**



**VROOM!**

Listener (1x)

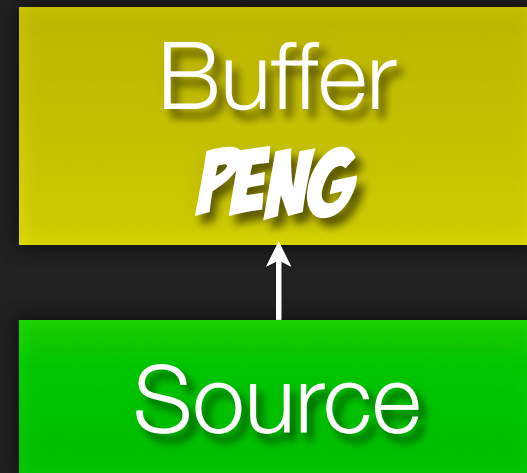
- Position
- Orientation
- Velocity...

**HINTERGRUNDMUSIK**



**MASCHINEN-  
BRUMMEN**

# Das Prinzip



Listener (1x)

- Position
- Orientation
- Velocity...

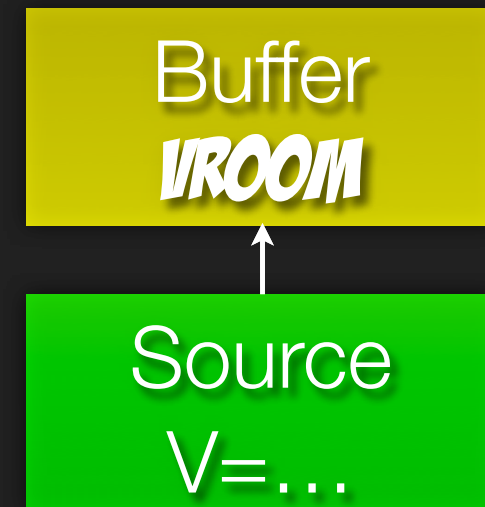
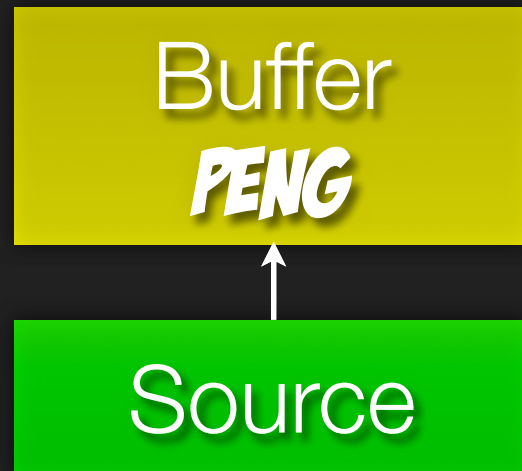
**HINTERGRUNDMUSIK**

**VIROOM!**

**MASCHINEN-  
BRUMMEN**



# Das Prinzip



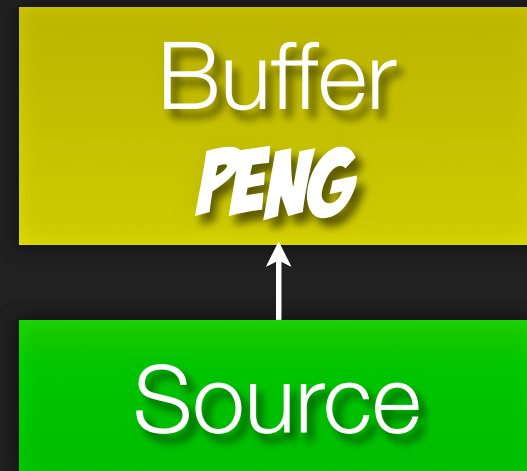
Listener (1x)

- Position
- Orientation
- Velocity...

**HINTERGRUNDMUSIK**

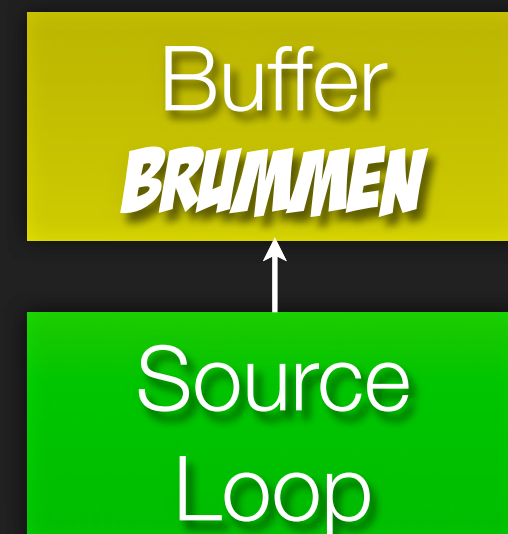
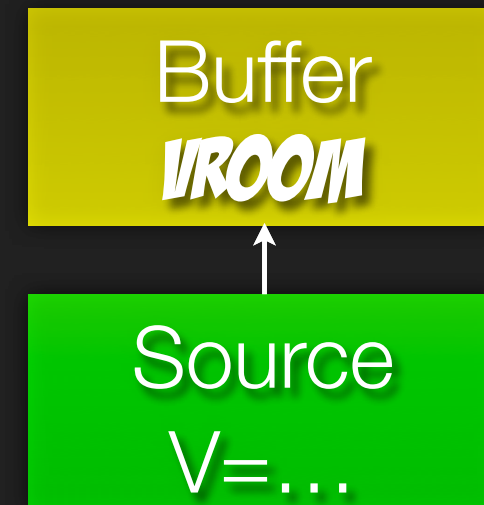
**MASCHINEN-  
BRUMMEN**

# Das Prinzip



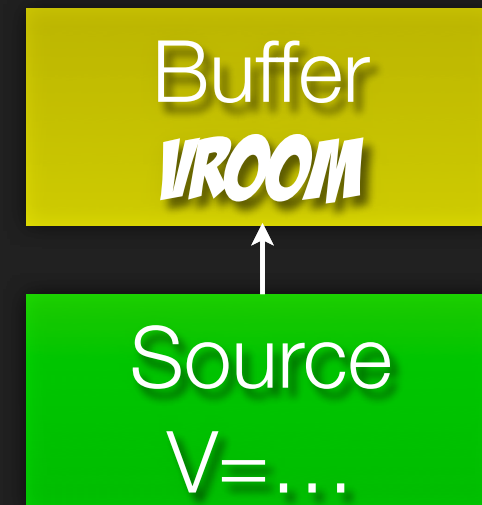
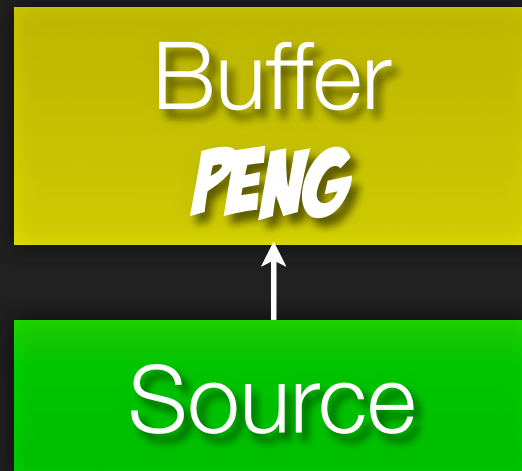
Listener (1x)

- Position
- Orientation
- Velocity...



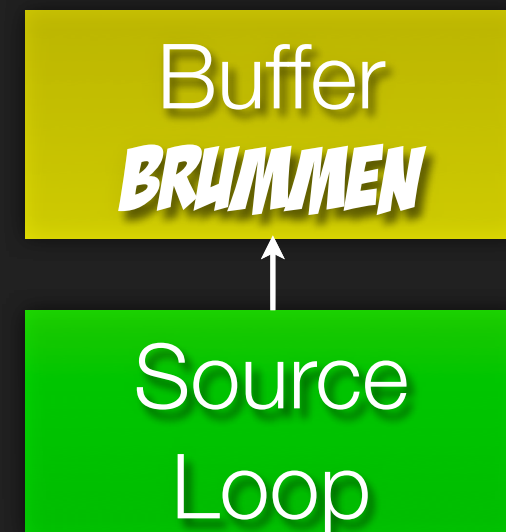
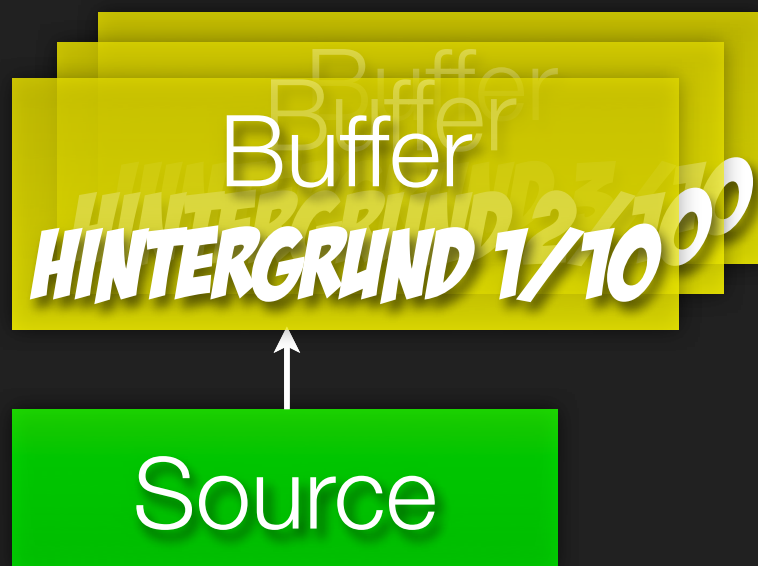
**HINTERGRUNDMUSIK**

# Das Prinzip



Listener (1x)

- Position
- Orientation
- Velocity...



# Kontext erzeugen

```
ALCdevice *device = alcOpenDevice(NULL);  
ALCcontext *ctx = alcCreateContext(device,  
NULL);  
alcMakeContextCurrent(ctx);
```

```
// Krach machen
```

```
alcMakeContextCurrent(NULL);  
alcDestroyContext(ctx);  
alcCloseDevice(device);
```

# Listener einstellen

```
alListenerf(AL_GAIN, 0.5);
```

```
float position[3] = {...}
```

```
alListenerfv(AL_POSITION, position);
```

```
float orientation = {f.x, f.y, f.z, u.x,  
u.y, u.z};
```

```
alListenerfv(AL_ORIENTATION, orientation);
```

```
float velocity[3] = {...}
```

```
alListenerfv(AL_VELOCITY, velocity);
```

# Erzeugen von Buffern

```
ALuint buffer;  
alGenBuffers(1, &buffer);
```

```
alBufferData(buffer, AL_FORMAT_  
{MONO,STEREO}{8,16}, data, length,  
samplingRate);
```

```
alDeleteBuffers(1, &buffer);
```

# Wo kommen die Daten her?

- ✦ Extended Audio Services
- ✦ Beispielcode: oalTouch
- ✦ Cross-Platform Alternative: Ogg
- ✦ Oder direkt WAV auslesen

# Sources

```
ALuint source;  
alGenSources(1, &source);
```

```
alSourcei(source, AL_BUFFER, buffer);  
alSourcei(source, AL_LOOPING, AL_TRUE);  
alSourcePlay(sourceID);  
alSourcefv(source, ..., ...);  
alSourceStop(sourceID);
```

```
alDeleteSources(1, &source);
```



# Hier nicht betrachtet

- ✧ OpenAL unterstützt Streaming (z.B. Hintergrundmusik)
- ✧ Verschiedene Effekte als Erweiterungen
- ✧ Aufnahme von Sound
- ✧ Verwenden von Ogg

# Mehr Information

- ✦ <http://www.openal.org/> - OpenAL 1.1 Specification
- ✦ Apple Dokumentation