

エンタメチーム

最終プレゼン

メンバー

- 氷室 貴浩
- 栗島 裕大
- 児玉 識彰
- 野本 太陽
- 坂谷 隼

概要

新・次世代型ペンライトについて(2分)

開発経緯及び背景(2分)

各成果物開発担当者による成果物の解説(10分)

まとめ(1分)

実演・質疑応答(5分)

成果物

新・次世代型ペンライトと

その実用例を示す

演出システム

成果物 | 詳細

- 盛り上がり度に応じて色が変わるペンライト
- 会場の色を認識して盛り上がり度を算出するシステム
- 盛り上がり度に応じ会場の演出を変化させるシステム

イメージ図



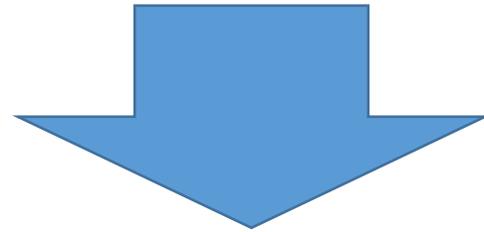
開発経緯・背景

既存のシステム：次世代型ペンライト FreFlow(フリフラ)



開発経緯・背景

問題：次世代型ペンライトは観客が**主催側に利用されている**



- ・観客主体でアピール・応援をするべき
- ・普通のペンライトより効果的なアピールがあるはず

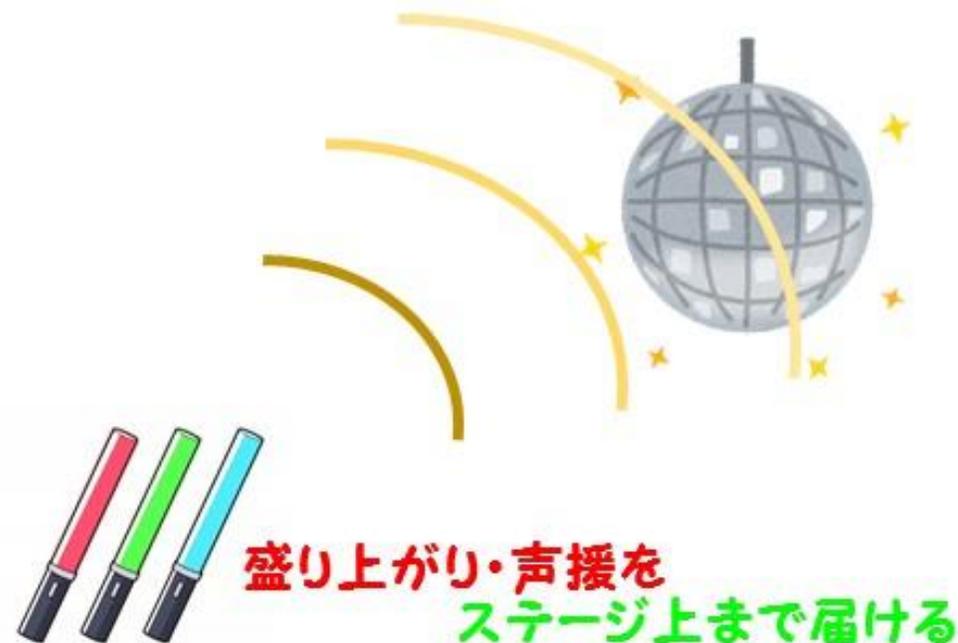
新・次世代型ペンライトのポイント

コンセプト

- ・あくまで観客のアピールである
- ・観客がライブをより盛り上げる

新規性

- ・ペンライトで観客の盛り上がりを表現できる
- ・観客の盛り上がりがステージに反映される



盛り上がり・声援を
ステージ上まで届ける

システム全体の概要



盛り上がりを集計し舞台演出まで反映させる

ハードウェア班1

~ハードウェア班で達成したいこと~

~ハードウェア班の目的~

観客の「盛り上がり」を数値化する

「盛り上がり」を示す参考になりそうな要素

- ・心拍数
- ・汗
- ・会場全体の振動
- ・**ペンライトの動き(頻度,大きさ)**

~授業内で達成したこと~

ペンライトが振りの強度を判断して結果を光で表現し
PC(後述のサーバー班の成果)へ送る

今回はペンライトを一定値以上の強度で

振っている→**盛り上がっている**

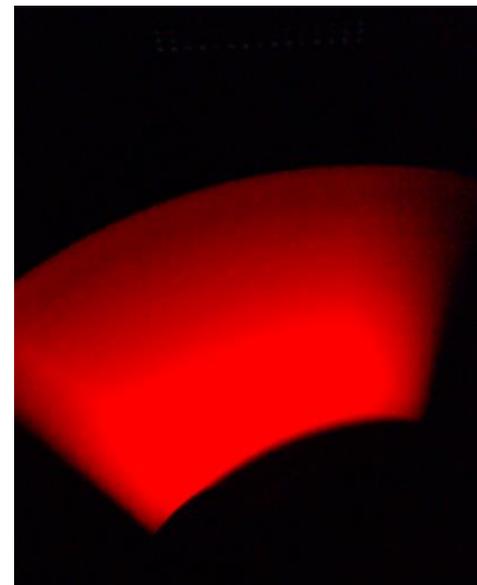
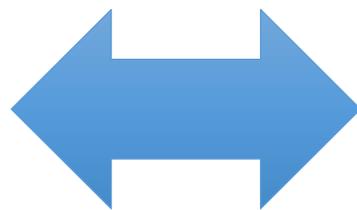
降っていない→**盛り上がっていない**

の二値で表現

図? 今回の授業で作成した改造ペンライト

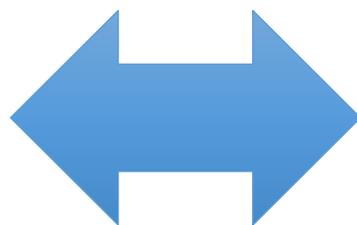


ペンライトの光の色



ペンライトの動き

「振られていない」



「振られている」

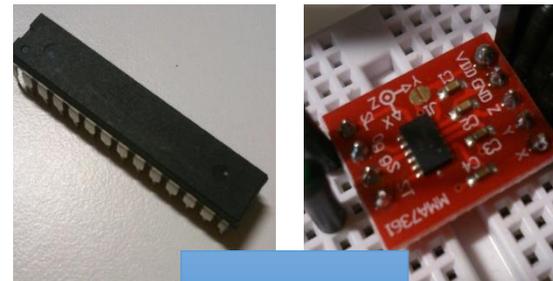
成果物：リアルタイムで光の色が変化するペンライト

仕組み

ペンライトの中身

- ・制御用マイコン(1)
- ・3軸加速度センサ(2)
- ・LED(3)

(1)マイコン (2)加速度センサ



内蔵



(3)LEDペンライト

動作の流れ

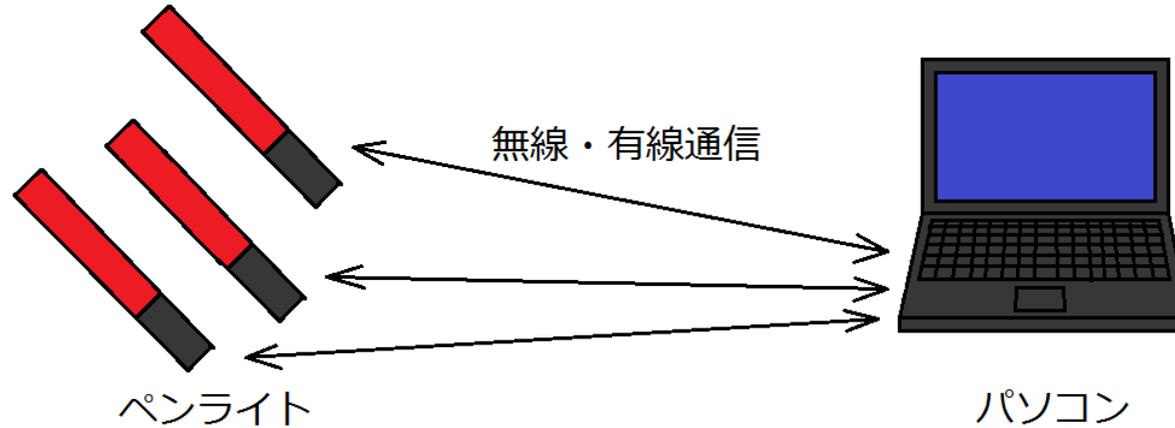
A,マイコンがペンライトの状態を監視(1,2)

B,LEDに結果を出力(光の色)(3)

→光の色は白もしくは赤になる

マイコンではA,Bの処理を3秒間隔程度でループ

サーバの目的



1. ペンライトが取得した情報の収集・解析
2. ソフトウェアへの解析結果の送信

サーバプログラムの問題点

1.コスト面

ペンライトに無線通信モジュールを載せると
高コストとなる(少なくとも2500円玉)

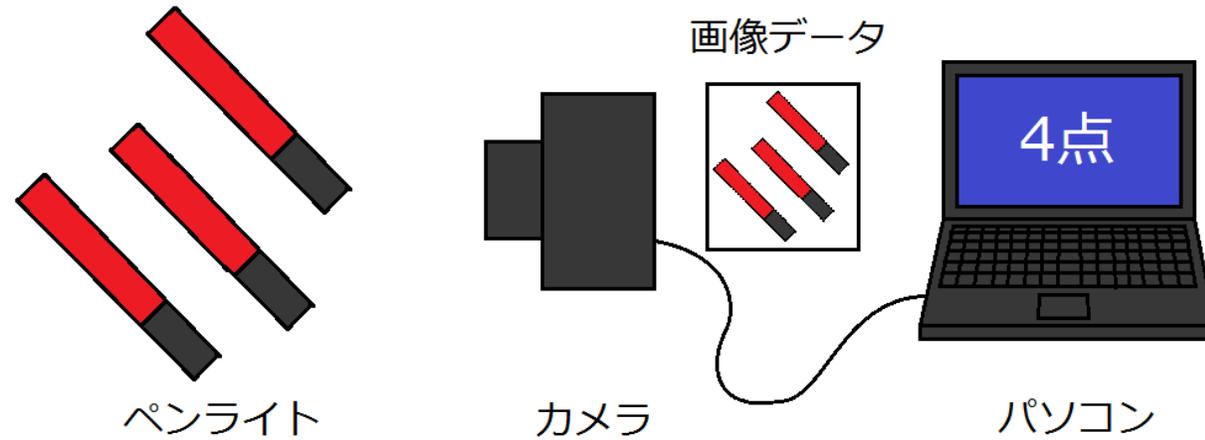
2.技術面

通信のプログラムはペンライトの数の増加により作成が難化する

3.利便性

有線通信は通信線がペンライトの使用時に障害となる

解決策:カメラを用いたサーバシステム



データの流れ

- 1.カメラによるペンライトの**画像の取得**
- 2.取得した**画像の点数化**
- 3.点数データのソフトウェアへの**送信**

システムの利点と欠点

・システムの利点

1. カメラで撮影するため無線・有線通信モジュールが不要
→ 低コスト、ペンライトの軽量化
2. ペンライトの数に依らない画像解析プログラム
→ プログラム作成が比較的容易

・欠点

1. 一つ一つのペンライトの識別が困難
2. 解析結果に確率的なゆらぎが生じる

サーバプログラムの要件(1)

1.複数のペンライトの状態の検出について

- ペンライトが示す状態(**白**と**赤**、赤が高得点)を検知し、出力する**得点に反映すること**
- 画像の1ピクセル毎に輝度データにアクセス、条件判断を行う
- スピードを重視するため**条件判断の数を減らす**
- ペンライト以外の色をなるべく**取り除く**

→**赤**の輝度データと**青**の輝度データを評価に使用する

サーバプログラムの要件(2)

2.ペンライトの状態を表す点数について

- 点数は1~5点の5段階であること

→輝度データを下式で点数に変換する。

$$\text{点数} = \{(\text{赤の輝度}0\sim255) - (\text{青の輝度}0\sim255)\} / 40$$

(点数が0点以下、6点以上となる場合は0点とする)

サーバプログラムの要件(3)

3.ソフトウェアへの点数の送信方法について

- サーバのプログラムとソフトウェアが別の言語で作成されている。

→ソフトウェアと共有するテキストファイルに点数を書き込む

開発環境(windows7)の決定(1)

- C/C++による開発環境
→VisualC++
- 高スペックのPCでなくてもデバッグ等が容易
→VisualC++2008

開発環境(windows7)の決定(2)

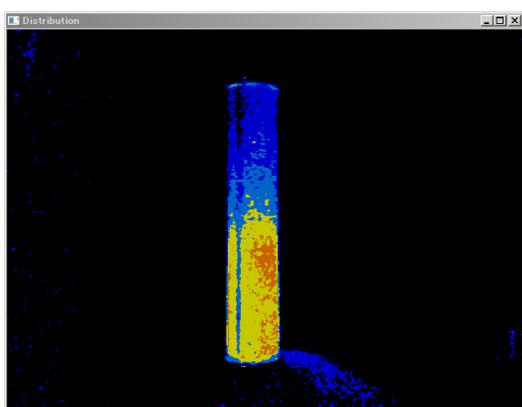
- カメラを操作して画像を取得、点数化できる
オープンソースのC++用ライブラリ
→OpenCV
- バグについての情報の豊富さ、ライブラリの容量の小ささ、
移植する際のコードの互換性
→OpenCV ver2.4.3

プログラムの動作(1)

画像

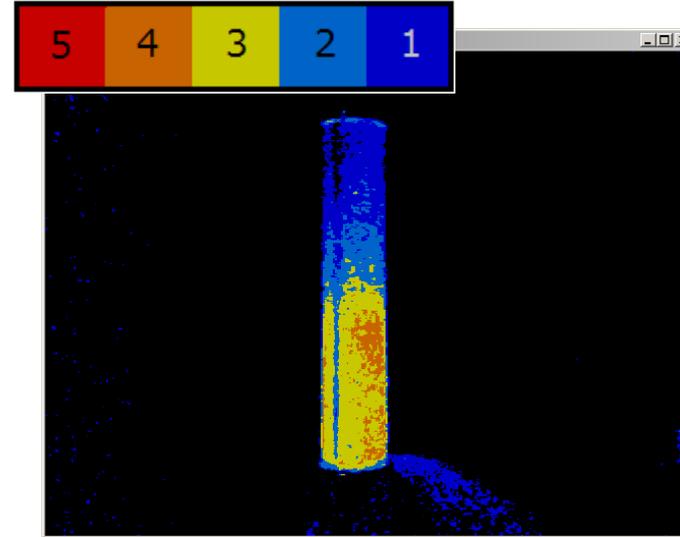


検出



- 外付けのwebカメラで画像を撮影する
- サイズは640x480
- 撮影は最大30fps
- 魚眼レンズを用いた視野の広域化を検討中

プログラムの動作(2)

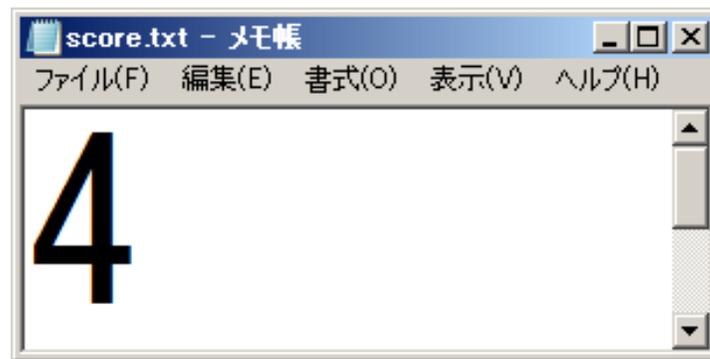


- 画像の1ピクセル毎にアクセスし、輝度データを取り出す
- 輝度データを下式で点数に変換する

$$\text{点数} = \{(\text{赤の輝度}0\sim255) - (\text{青の輝度}0\sim255)\} / 40$$

(点数が0点以下、6点以上となる場合は0点とする)

プログラムの動作(3)



- 得点の総和を点数化を行った総ピクセル数で割り平均点数を求める
- 求めた平均点数(1~5)を
ソフトウェアと共有しているテキストファイルに書き込む
- テキストファイルには約2秒ごとに点数を上書きする

プログラム作成の問題点と解決策

- 取得した画像へのポインタが参照してはいけない領域を参照し
プログラム実行時にエラーが発生する
 - ポインタによる参照をC++の関数からC言語による直接指定する方法に変える
- プログラムを移した別のパソコン(windows8.1)で動作しない
 - 開発環境をVisualStudio2015、OpenCV ver2.4.11に変更、コードを移植
- プログラム起動後1秒間にキャプチャされる画像のサイズが用意した
メモリの大きさを超える(カメラの設定が反映されていない)
 - カメラの設定が反映されるまでの画像を破棄する

ソフトウェア開発

以下の機能を実装したアプリケーションを開発

- 予め作製した動画を**自動**で切り替えることができる
- 外部機器（ハード・ソフト）との**通信**ができる
- **単独**で稼働でき、機能の**拡張**が**容易**である

開発言語

Processingでアプリケーションを設計



特徴

- 電子アートとビジュアルデザインに特化した総合開発環境
- Javaを単純化しグラフィック機能に特化させた言語

開発環境

OS : Windows10 (64bit)

言語 : Processing 2.2.1

プログラム構造

動画ファイルの
パスを登録 (5つ)

動画とレベルを紐つけ

ウィンドウサイズ・
フレームレートを設定

初期レベルを 1 に設定

Setup()
プログラムの最初に
1 度だけ読み込み

レベルに対応する
動画を再生する
(それ以外は停止)

3 秒ごとに Switcher
関数を実行

Draw()
毎秒 30 回実行

テキストファイルに
格納されたスコアから
レベルを算出

現在のレベルを変更

Switcher()
呼び出しごとに実行され
演出用動画の
切り替え制御を管理

工夫点

- 動画の遷移の自動制御ができる
- スコアの閾値が可変である
- レベルの拡張が容易である
- 出力に関して様々な拡張ができる

動作環境

- Windows / Mac / Linux
- Processing.js

演出構成の概要

- ソフトウェア上での1から5のレベルに対応する動画を動画作成ツールを用いて作製
- 動画の長さは3分

地味



派手

演出の設定

- ステージ舞台を模したモデルをスクリーンに映す
- 音楽が流れ観客がペンライトを振り
ソフトウェア上でのレベルに応じて次々と演出が変わる
- 1 から 5 へレベルが上がるにつれて動画内の演出が派手になる

使用したソフトウェア①

MikuMikuDance(Ver.9.26)

通称：MMD

3Dのモデルを作成・操作でき、動画を抽出することも可能



自分の好きなキャラクターを
自由に動かした動画が多数

(ニコニコ動画より)

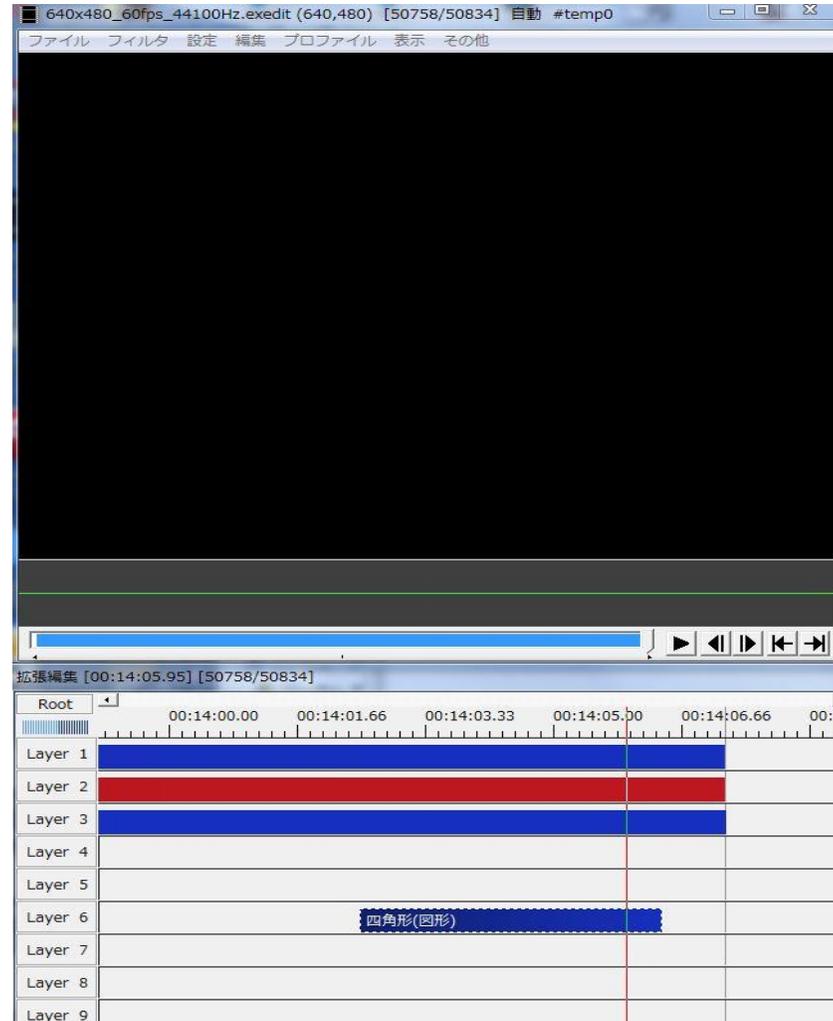
使用したソフトウェア②

AviUtl

動画編集ソフト

今回は動画のビット調整や

容量の軽減に利用した



(編集画面)

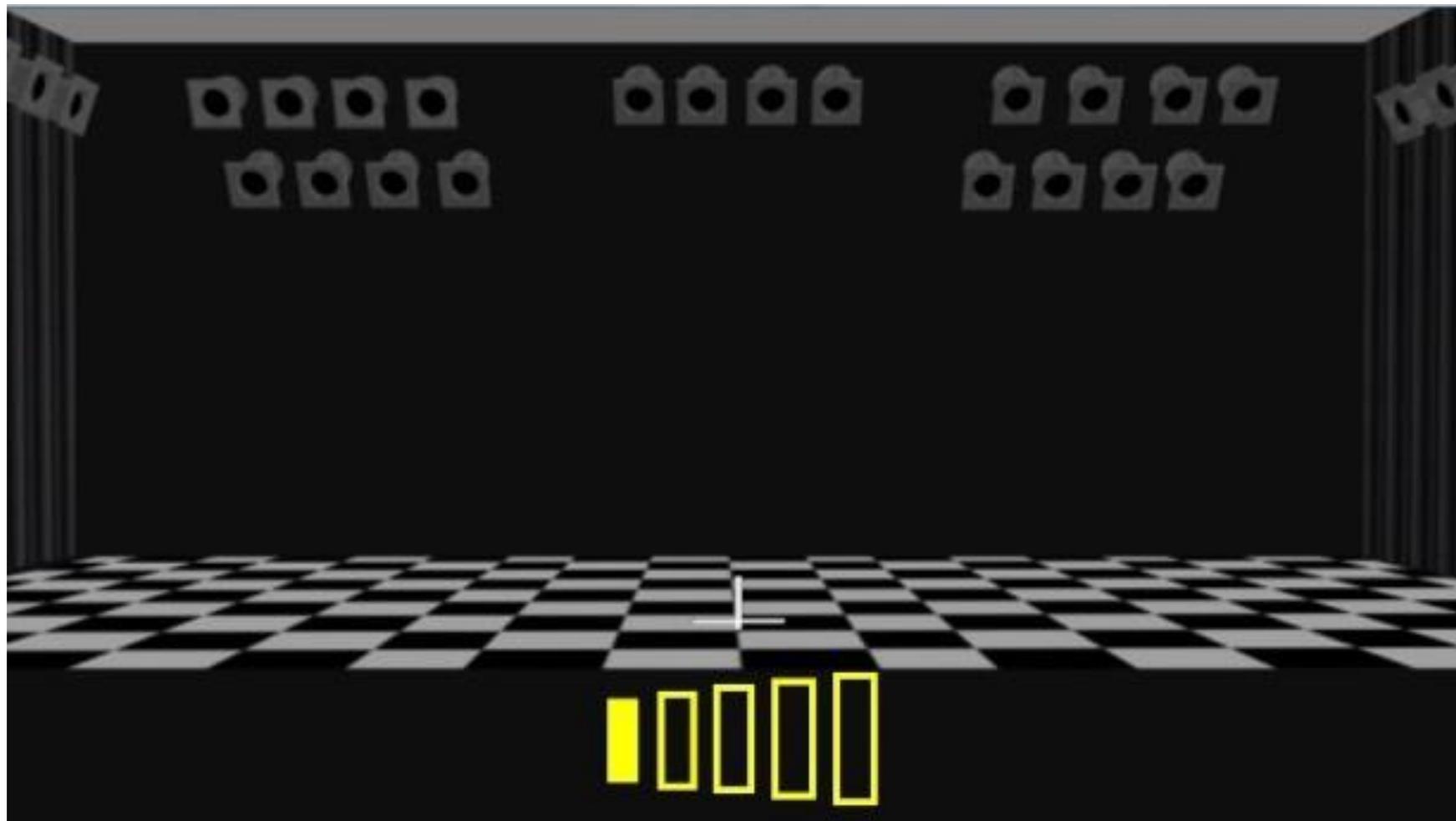
使用した素材

- 小さめライブハウス風ステージPMX化
(<http://bowlroll.ne./up/dl16287>)
- ミラーボール
(<https://bowlroll.net/file/51978>)
- ドラゴン花火
(<http://3dphoto.jp/>)

演出をより盛り上げるために

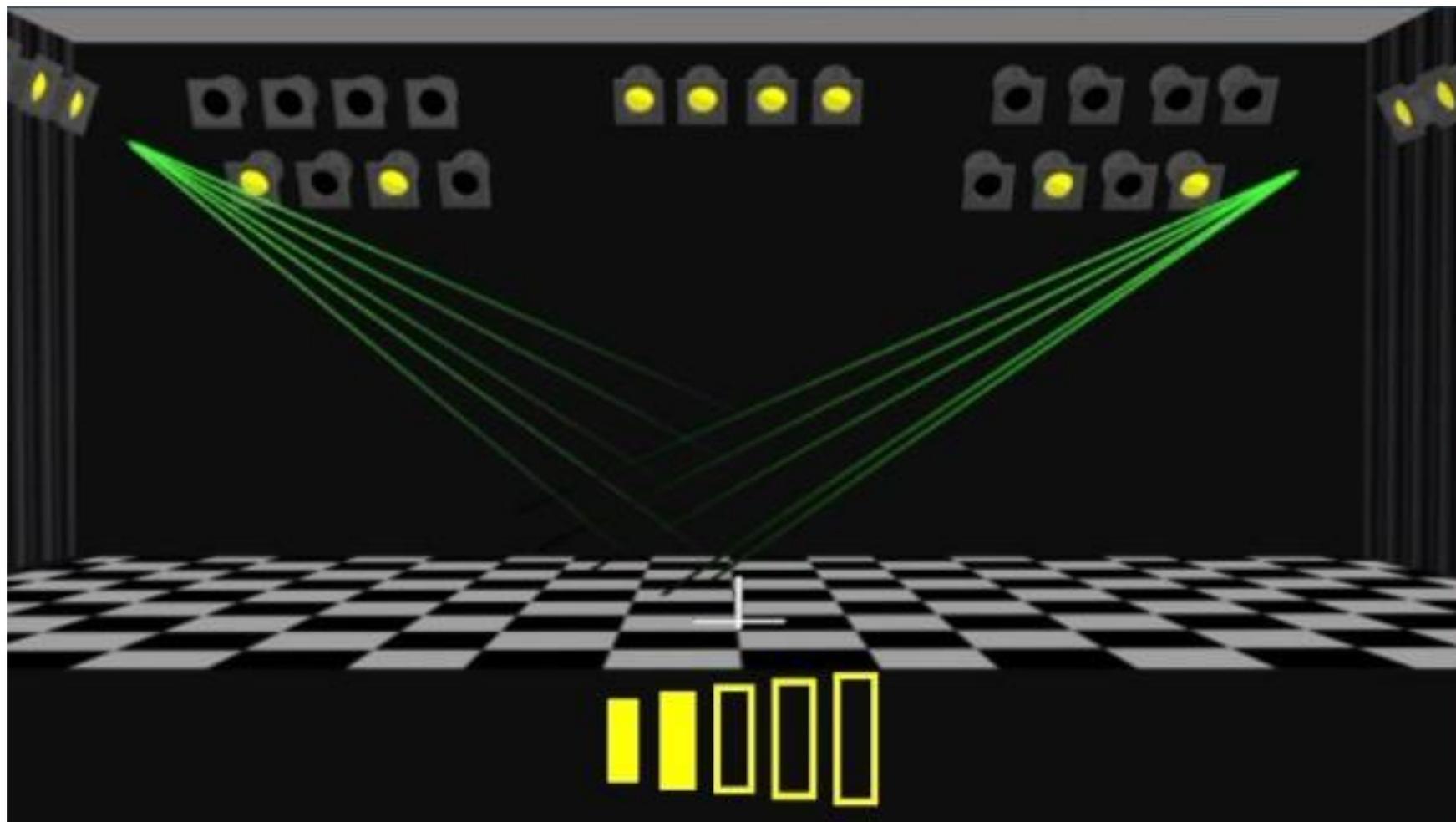
- 3Dモデリングの動画を用いることで
より**臨場感**を出し、盛り上がる動画を目指した
- 演出のイメージとした楽曲
 - Dance My Generation / ゴールデンボンバー

動画の詳細（レベル1）



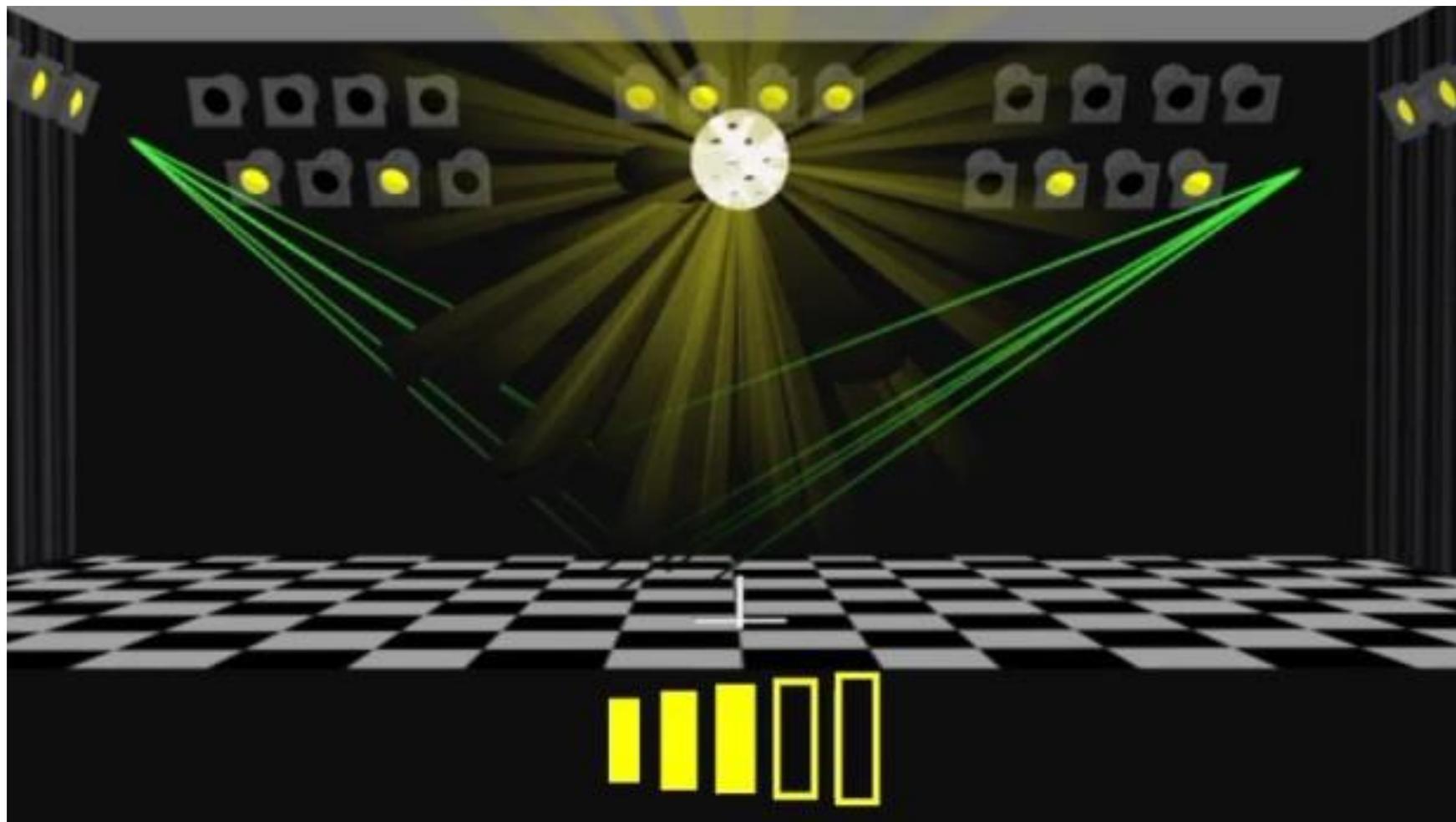
- ステージのみ

動画の詳細 (レベル2)



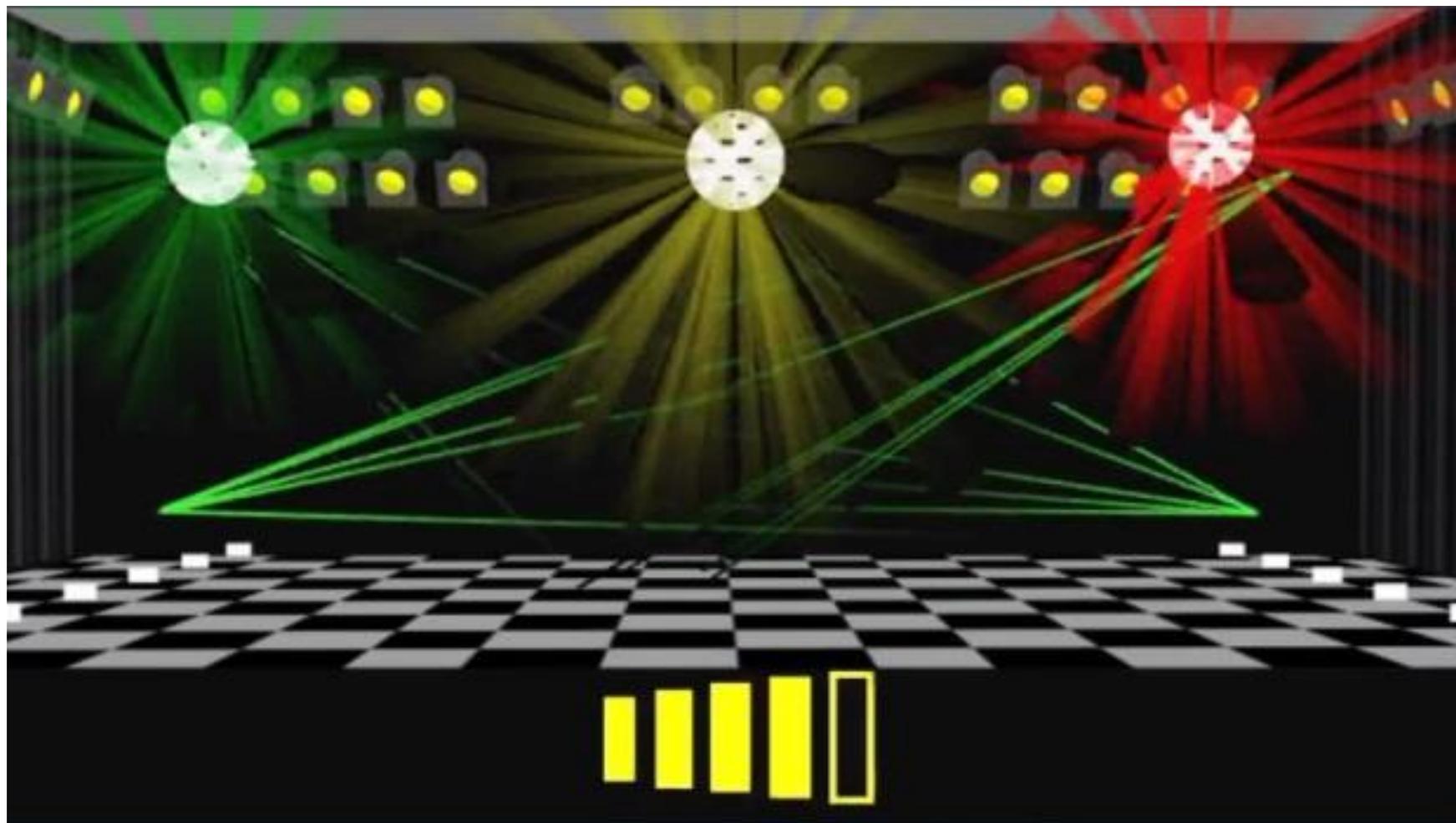
- 照明が一部点灯
- レーザー (2箇所)

動画の詳細 (レベル3)



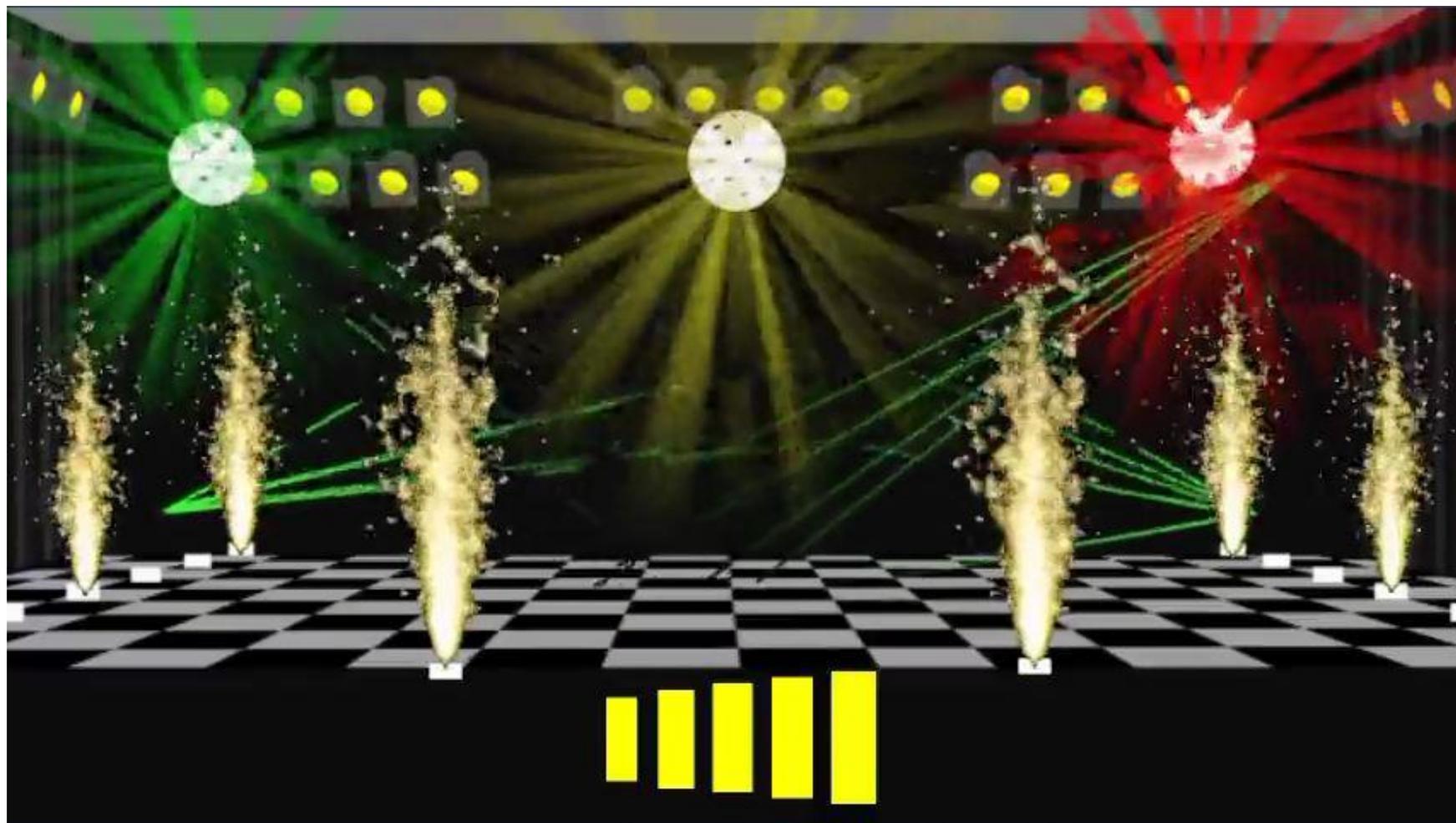
- ミラーボール (1つ)

動画の詳細 (レベル4)



- ミラーボール (3つ)
- 照明が全て点灯

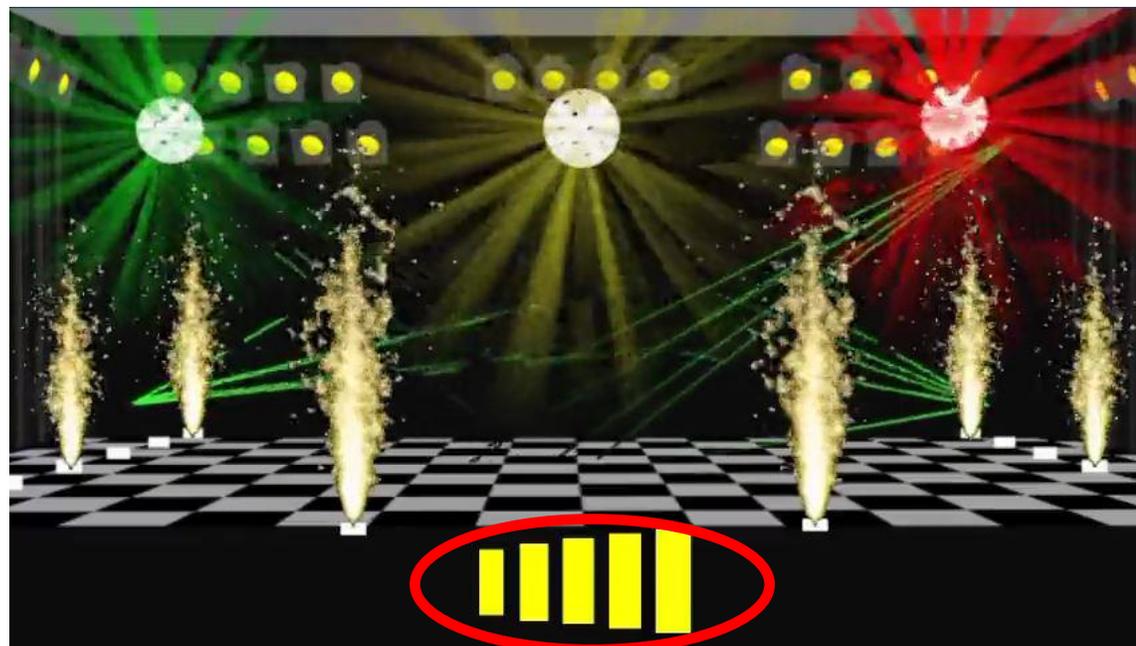
動画の詳細 (レベル5)



- ドラゴン花火

工夫点

- レベルの数に合わせるため、
また素材が多すぎて目が疲れないようにするため、
素材の種類を絞った
- 動画の下に目盛を置き、
今どのレベルの演出なのか
一目で分かるように配慮した



まとめ

- 各成果物はそれぞれ独立で稼働する
 - 各部品を改良しても他のシステムの仕様に影響しない



大きな拡張性を持った
総合的なシステム