

# Motos プログラム説明書

トライデントコンピュータ専門学校

ゲームサイエンス学科

深草 直斗

## 内容

---

自分で作成したソース	1
使用させてもらった、ほぼそのまま使用したソース	2
工夫した点	2
グラフィック系管理クラス（ライブラリ）	2
リソース管理クラス（ライブラリ）	2
当たり判定クラス（ライブラリ）	2
UI のアニメーション（ゲーム）	3
Controller クラス（ゲーム）	3

## 自分で作成したソース

---

Games ファイル内のソースファイル（Library/Common フォルダ内のファイルは除く）

## 使用させてもらった、ほぼそのまま使用したソース

---

Games/Library/Common フォルダ内のソースファイル

DeviceResources : DirectX11 のデバイス関連のオブジェクトを管理する

StepTimer : フレーム制御

## 工夫した点

---

### グラフィック系管理クラス (ライブラリ)

---

PrimitiveManager2D, PrimitiveManager3D, Sprite クラスは比較的簡単に図形や画像を描画することが出来るクラスです。

DX ライブラリをモデルにしました。

### リソース管理クラス (ライブラリ)

---

ModelManager, ShaderManager, TextureManager クラスはリソースの多重読み込みを防ぐためのクラスです。

STL の map クラスを使い、Key をファイル名に設定することで多重読み込みを防いでいます。

対応しているモデルデータは PMD, PMX

### 当たり判定クラス (ライブラリ)

---

CollisionManager は登録されたコライダーの衝突判定を管理するクラスです。

衝突時の処理の変化をインターフェイスを用いてコントロールすることが出来るようにしました。

衝突判定は毎フレーム行うため数が増えるにつれて処理が重くなってしまいます。  
これを解決するために登録時にコライダーのグループを分け、Unity の Layer Collision Matrix をイメージした設定を間に挟むことで総当たりの衝突判定からグループごとの当たり判定が出来るようにしました。  
対応しているコライダーの図形は立方体と球体です。

## UI のアニメーション (ゲーム)

---

ゲームの開始時に画面右の UI を音に合わせてアニメーションをさせました。  
実装を優先したためプログラムが汚くなってしまったが今後ツールや外部データの読み込みに改造しようと思っています。

## Controller クラス (ゲーム)

---

動作系のクラスとして分解することで改造や調整をしやすくするためにこの設計にしました。