

# 地形にプレイヤー立たせて丸影

地形にプレイヤー立たせて丸影を落とします。  
毎フレーム、レイキャストとかやっているとしんどいので、地形データを使って高さを補間で算出してプレイヤーを立たせます。  
(よく考えるとレイキャストとあんまり計算量変わらないかな。。。)

- [まずは、プレイヤーをつくる](#)

# まずは、プレイヤーをつくる

これは、いつものモデル読んでGameObject継承したPlayerクラス作る感じでいいと思う。  
必要な情報はあとで追加していこう。  
最低限必要なところで考えると、以下のようなクラスになりそう。

☒ 例：クラス構成の改善案（関数分割）

```
class Player : public GameObject {
public:
    void Initialize() override;
    void Update() override;
    void Draw() override;
    void Release() override;

private:
    void HandleInput(); // 入力処理まとめ
    void ApplyGroundAdjustment(); // 地面に沿って傾き・高さ補正
    void SetTPSCamera(); // TPS視点カメラ制御

    int hPlayer = -1;
    const float moveSpeed = 5.0f;
    const float rotateSpeed = 90.0f;
    XMVECTOR forward_ = {0, 0, -1, 1.0};
};
```

## ☒ 小学生でもわかる要約文（ここまでの内容）

- プレイヤーは「矢印キーで動く人形」です。
- 地面に合わせてちゃんと立つようにしてます。
- 地形の高さを調べて、プレイヤーが浮いたり埋まったりないようにします。
- プレイヤーの後ろにカメラがくっついて、いつも後ろから見えるようにしてます。
- そのためのいろんな処理を、関数にわけてスッキリさせたいです。

## ☒ 1. Initialize() : プレイヤーの初期化

### ☒ 目的：

プレイヤーのモデルを読み込み、初期位置と向きを設定します。また、TPS（後ろから追いかけるカメラ）の初期位置もここでセットします。

### ☒ 実装と説明：

```
void Player::Initialize()
{
    hPlayer = Model::Load("Player.fbx"); // モデル読み込み
    Model::SetAnimFrame(hPlayer, 0, 42, 1.0); // アニメーション範囲設定（0~42）

    // 初期向き（Z+方向を向くようにするために180度回転）
    transform_.rotate_ = { 0.0f, 180.0f, 0.0f };

    // 初期位置（地形の中心あたりに置く想定）
    transform_.position_ = { 0.0f, 0.0f, 0.0f };

    // 初期の前向きベクトルもZ+方向（右手系の場合、奥に向かって前進）
    forward_ = { 0.0f, 0.0f, 1.0f };

    // カメラ初期設定
    SetTPSCamera();
}
```