

スカイドーム

ゲームの背景となる、空や宇宙を簡易的に表現できます。

3Dゲームは、閉鎖的なゲームステージでない限りは、
オープンワールドのように広大なマップを用意しないと、見栄えが悪くなります。

『P.T.』



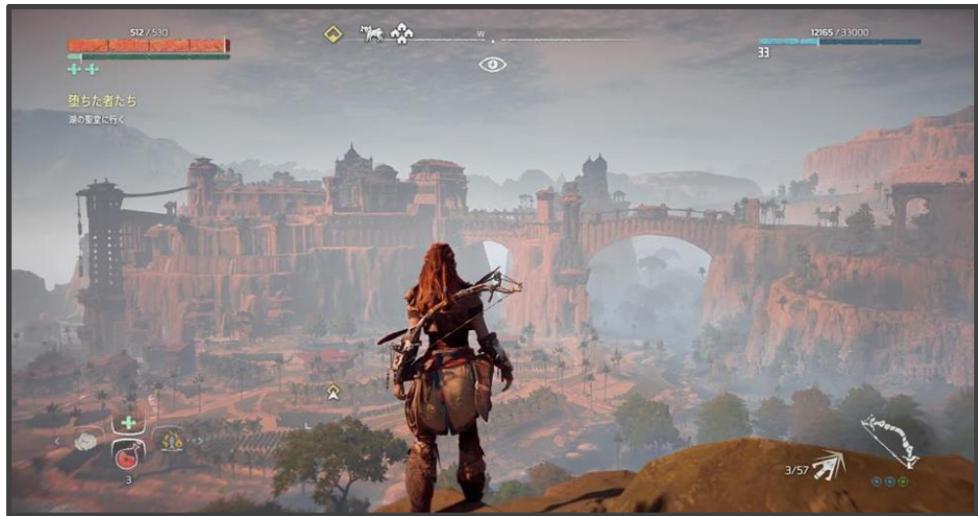
それなりに建物は広いですが、外が見えないので、広大なマップを作らなくて良い。

『ディアブロⅢ』



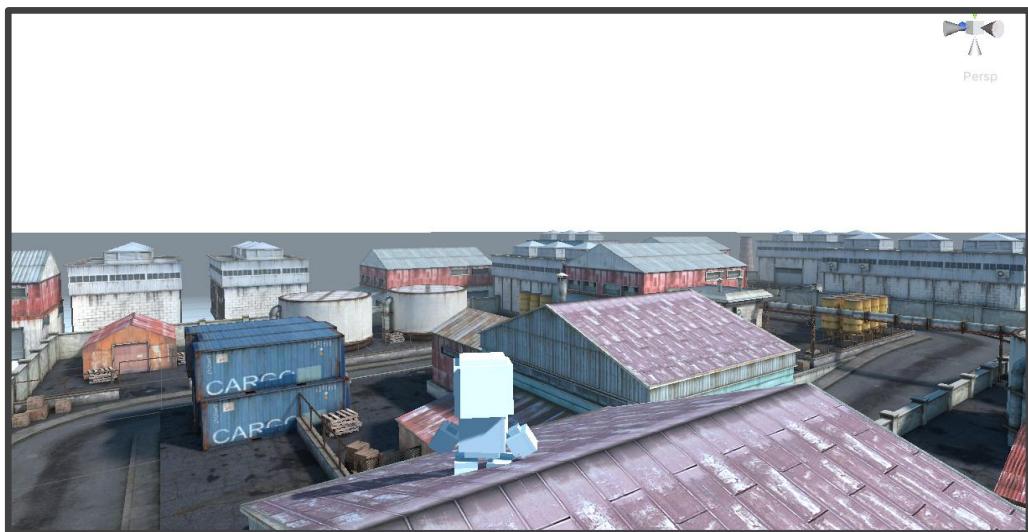
ダンジョンは、とても広大ですが、
カメラが固定で見下ろし型であるため、遠くの景色を写さなくて良い。

『ホライゾン ゼロドーン』



物凄く遠くまで見える。オープンワードなゲームは、
ユーザが見飽きないように、広大なマップを作る必要があります。

『もし、自作で頑張って作っても』



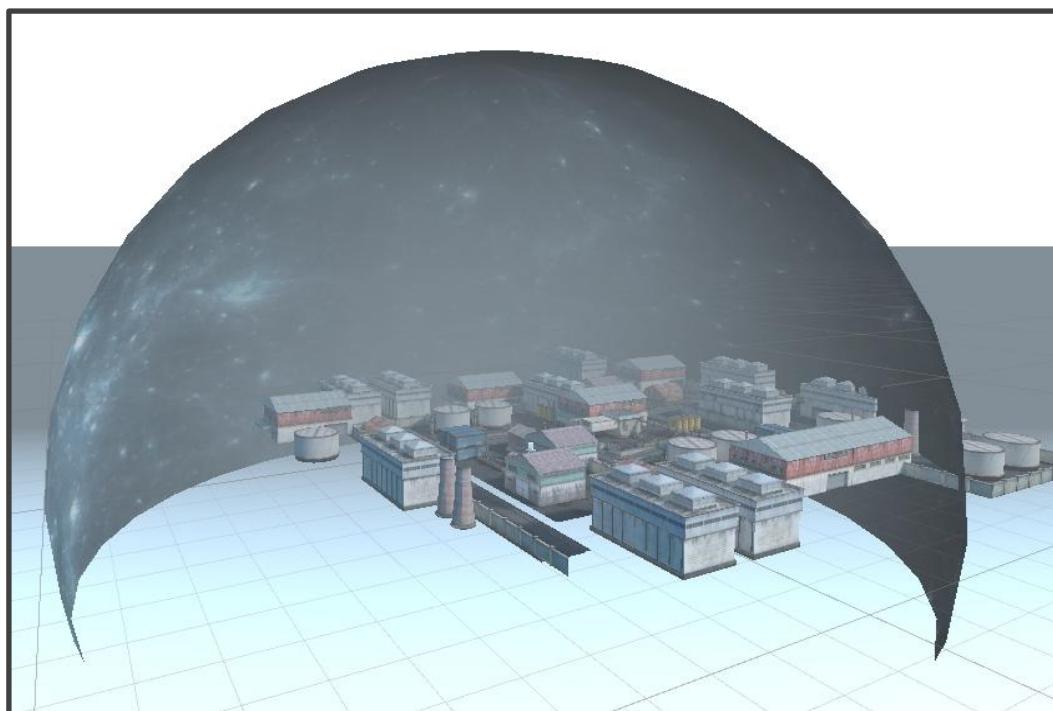
頑張って、建物を並べても、遠くの景色まで手が回らず、
背景の真っ白さにがっかりすることになるでしょう。
私たちは、デザイナーではありませんし、そこまで労力はかけたくないありません。

そこで、スカイドームの登場です。



2Dの背景を背景に描画しただけではなく、球体のモデルを使用して、内側にテクスチャを貼っています。

イメージ図↓

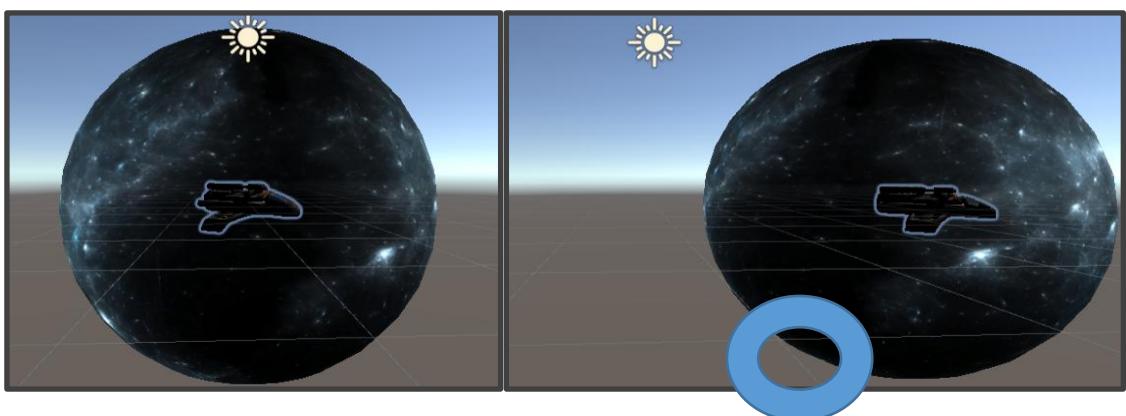
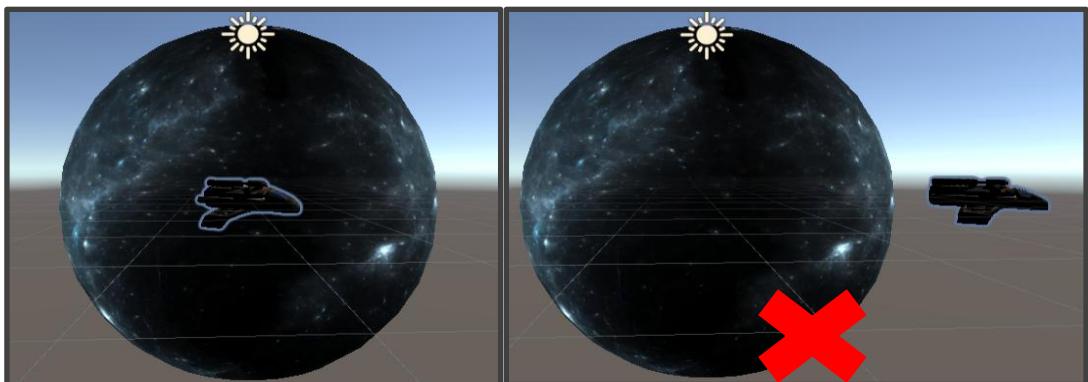


これで、かなり背景クオリティがアップします。

球体なので、色々なアングルでもそこそこ大丈夫。



あとでPlayerクラスを実装した後に行いますが、
操作キャラクターの移動に合わせて球体も移動させる必要があります。



学生時代のゲーム作品だったら、スカイドームで十分です。

【要件①】

ActorBaseを継承した、SkyDomeクラスを作成し、
タイトルシーンとゲームシーン上で、球体モデルを描画すること。

| | | |
|--------|---|-------------------------------------|
| リソース名 | : | SKY_DOME |
| リソース種別 | : | MODEL |
| 読み込み先 | : | PATH_MDL + "SkyDome/SkyDome.mdl" |
| スケール | : | 100.0 |
| ローカル回転 | : | Y軸180度 |
| 座標 | : | { 0.0f, 0.0f, 0.0f } |

ライト効果がかかると、部分的に影がかかり、
暗くなってしまいますので、
モデル描画の際には、下記の通り、ライトをオフにします。

```
void SkyDome::Draw(void)
{
    SetUseLighting(FALSE);
    MVIDrawModel(transform_.modelId);
    SetUseLighting(TRUE);
}
```

【目標①】



【要件②】

時間経過で、空が流れる演出を行うため、
スカイドームをゆっくり回転させること。

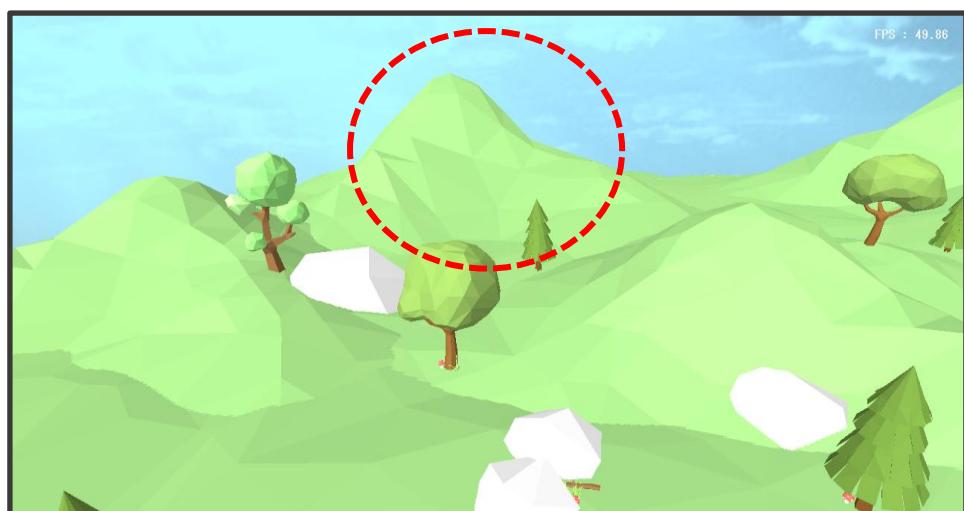
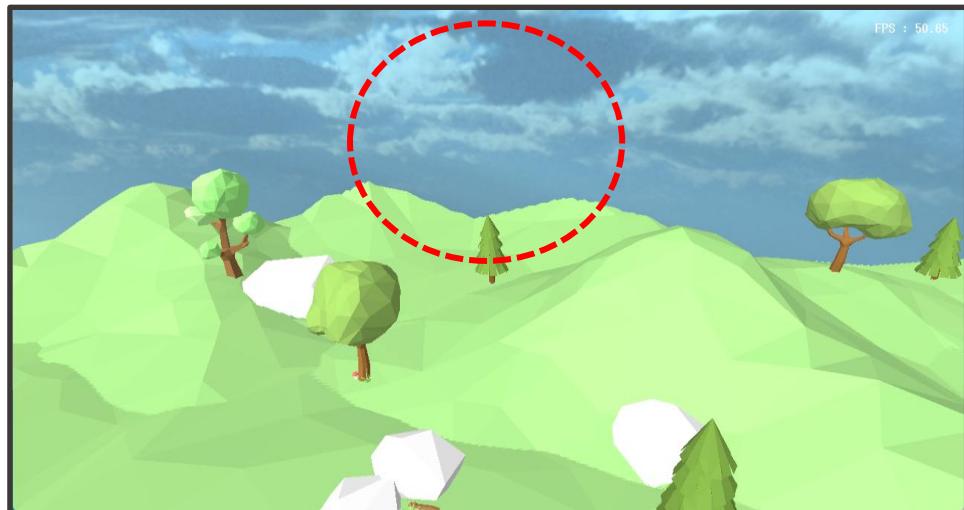
【目標②】

空(スカイドーム)がゆっくり流れるようになること。

もし、スカイドームより、遠くにオブジェクトを配置する場合

StageのZ座標を、仮で6000.0fに設定します。

すると、スカイドームとステージが重なってしまい、
モデルが切れてしまします。



スカイドームよりも、奥側にあるステージモデルは、
奥行き制御、Zバッファにより、当然ながら描画されません。

対策として、スカイドームのスケールを大きくして、
ステージがまるまる収まるようにしようとしがちですが、
大きくすればするほど、テクスチャが荒くなる分、解像度を上げる
必要がありますし、さらに遠い位置にあるモデルが出てきたら？
と、キリがありません。

そこで、スカイドームの描画時には、
Zバッファを参照せず、書き込みも行わないようにして、
この問題を解決することができます。

DxLibでのコードは以下の通りです。

```
void SkyDome::InitPost(void)
{
    // Zバッファ無効(突き抜け対策)
    MVISetUseZBuffer(transform_.modelId, false);
    MVISetWriteZBuffer(transform_.modelId, false);
}
```