

ゲームグラフィックス特論

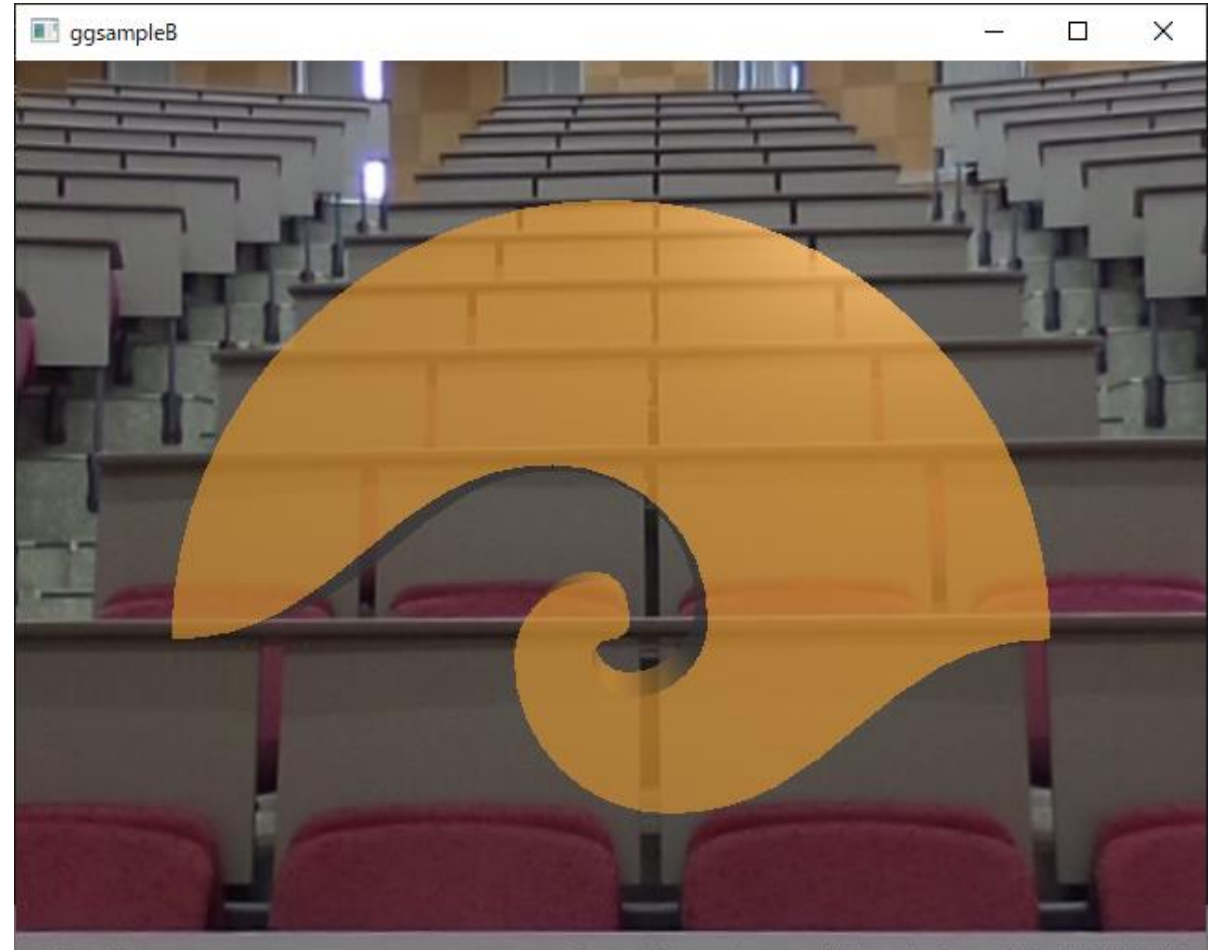
課題 B

課題

このプログラムは右図のような表示を行います

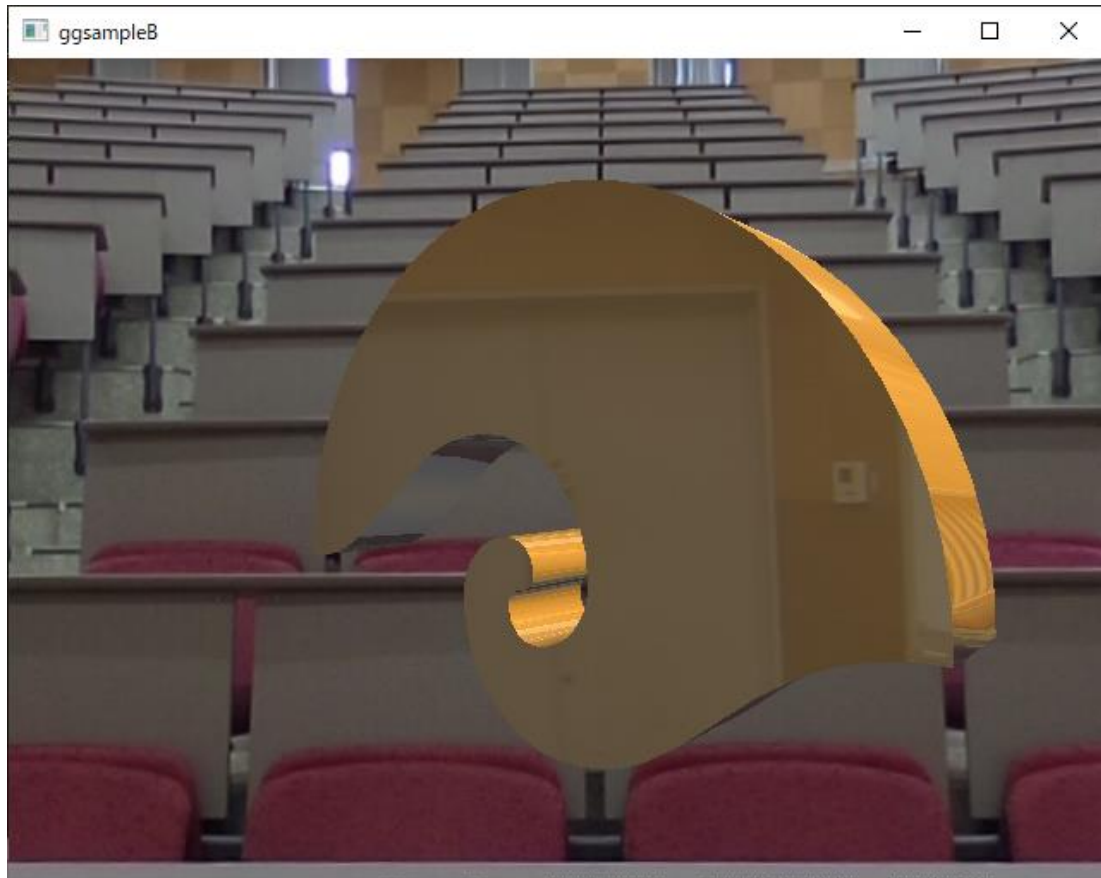
- ① このフラグメントシェーダのソースプログラム `simple.frag` に映り込み処理 (B第3回) を実装してください
- ② このフラグメントシェーダのソースプログラム `simple.frag` に屈折処理 (B第5回) を実装してください

それぞれの `simple.frag` を Moodle に **アップロード** してください

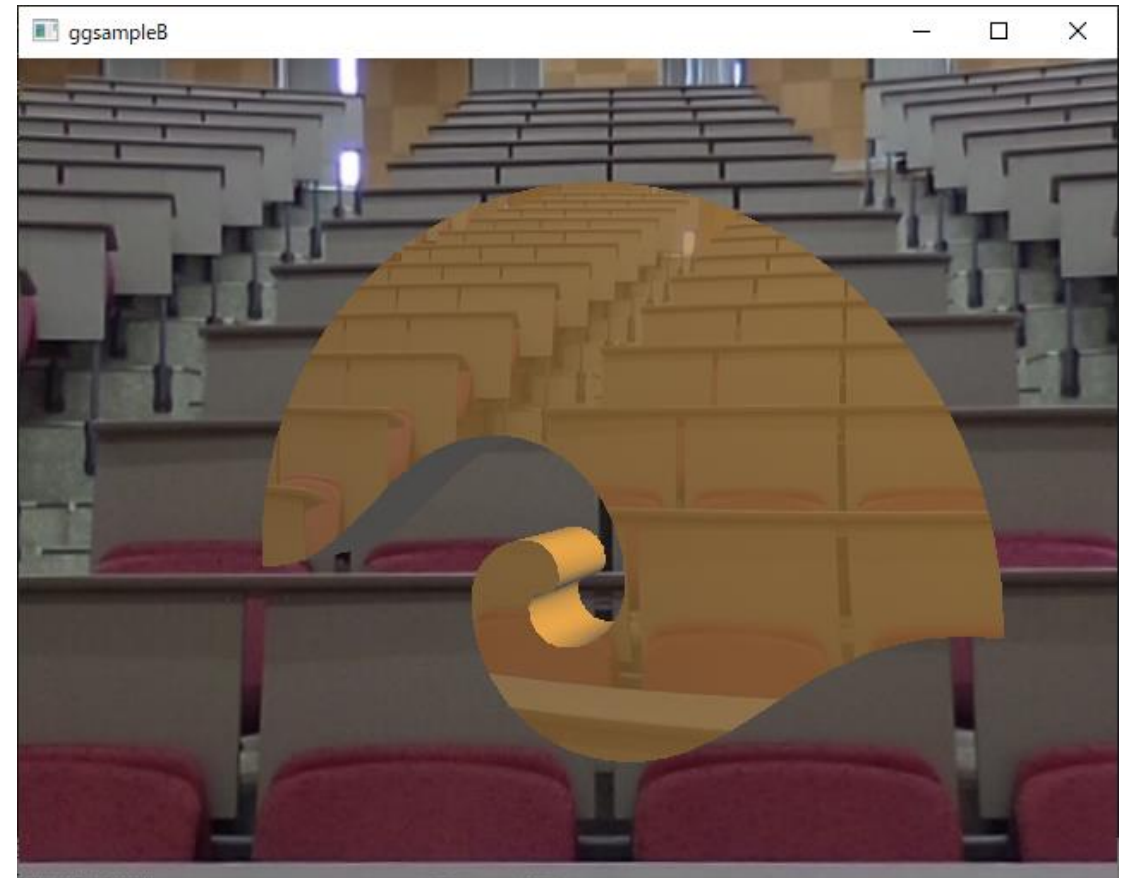


実行例

環境マッピング



屈折マッピング



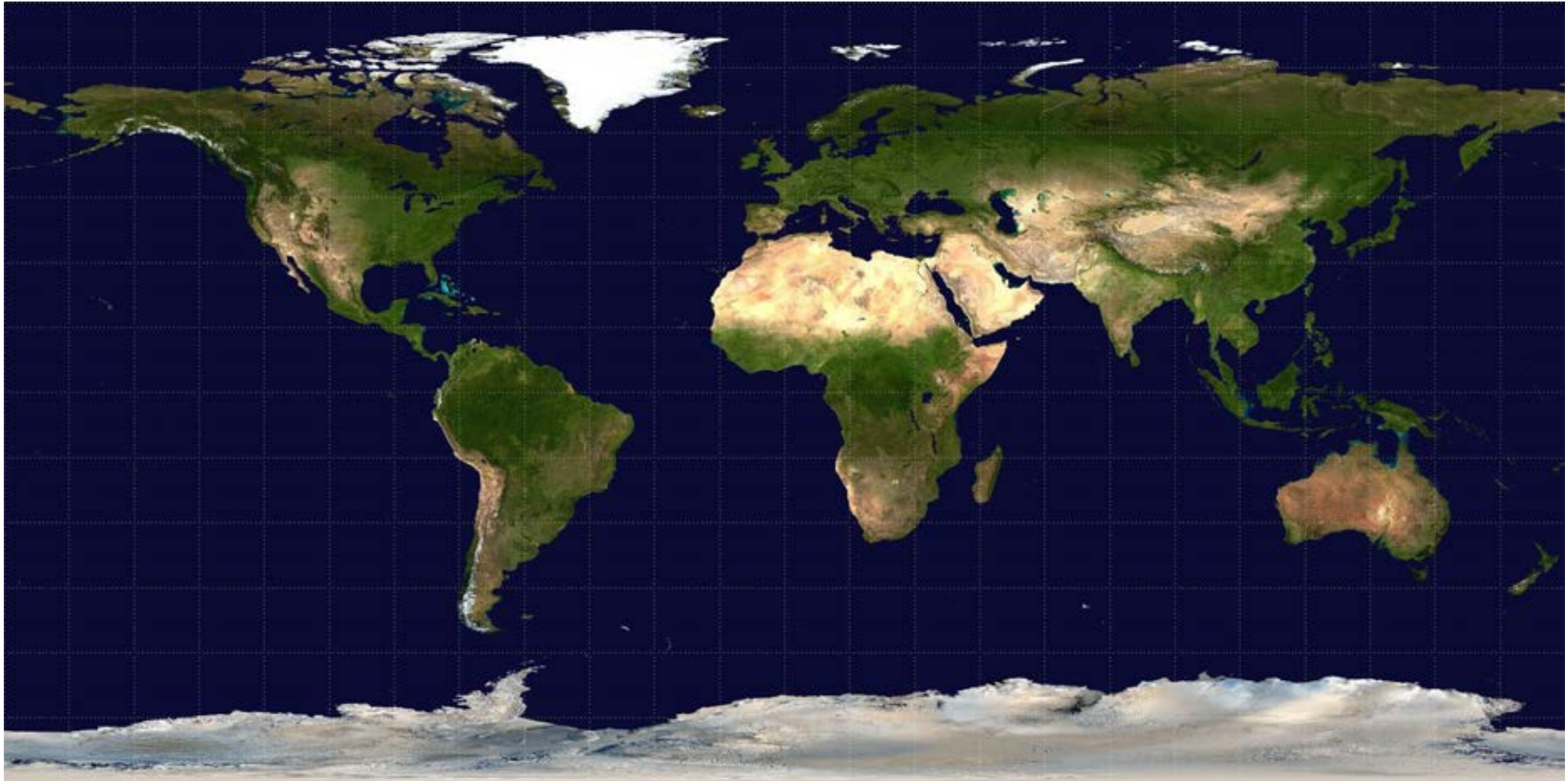
テクスチャに全方位 (全天球) 画像を使う

正距円筒図法

全方位（全天球）画像の利用

- このプログラムでは背景に全方位（全天球）画像を使用している
 - 1点から見たすべての方向の光景を1枚の画像に収めたもの
 - キューブマッピング（第10回）でも同じことが可能だが画像が6枚必要
- 正距円筒図法（Equirectangular）の画像を用いる
 - 正距円筒図法は球面を平面に展開する手法の一つ
 - 横軸に経度、縦軸に緯度を取る
 - 経線と緯線は直交するがメルカトル図法と違って角度が等間隔になっている
 - RICOH THETA などの全天球カメラや Google Street View などでも用いられる

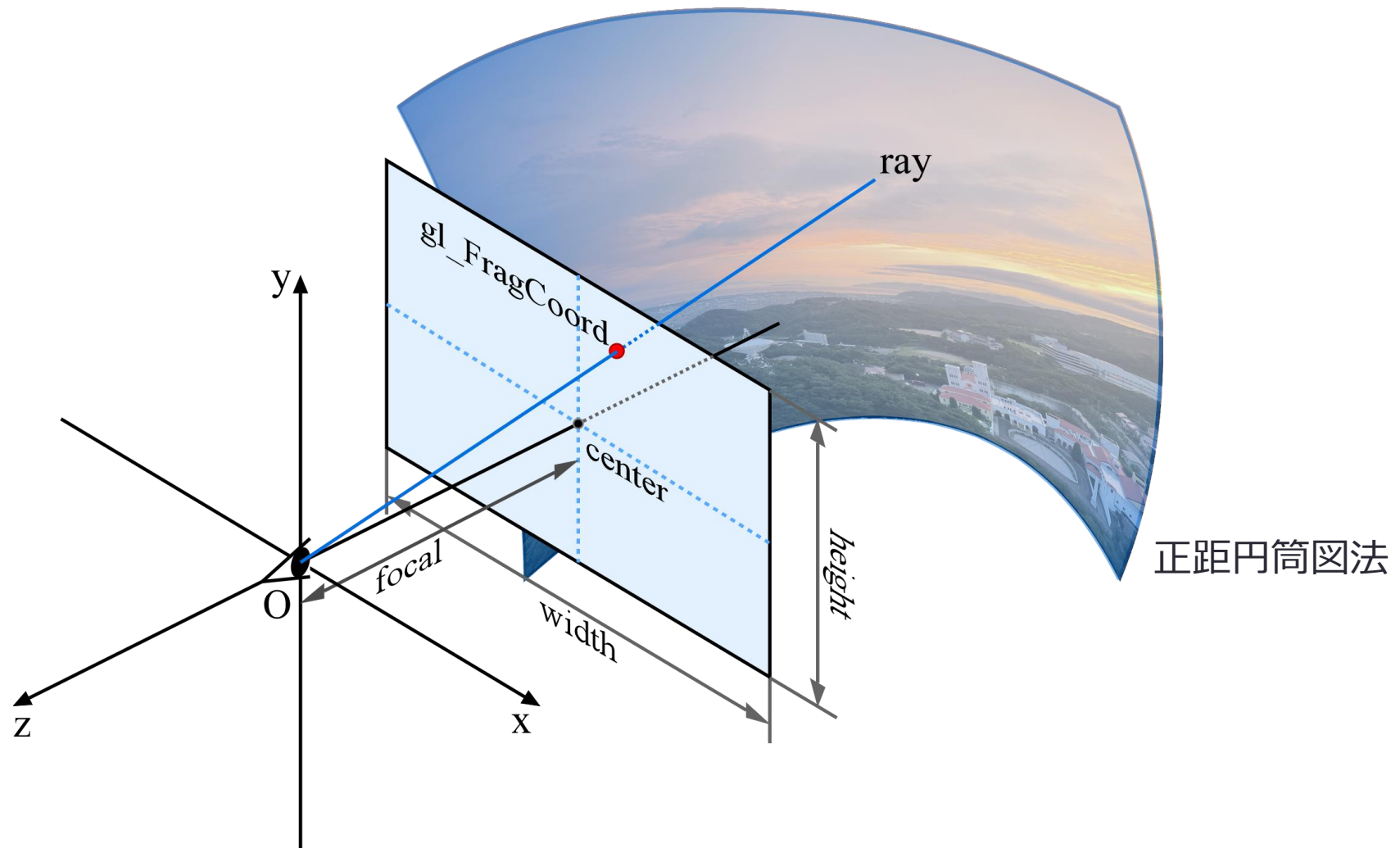
正距円筒図法の画像 (Wikipedia より)



視線ベクトルの算出

フラグメントを通る視線

フラグメントの位置 `gl_FragCoord` と視線ベクトル `ray`



gl_FragCoord から視線ベクトル ray を求める

- gl_FragCoord はフラグメントシェーダが出力するフラグメント位置
 - vec4 型、 $(x, y, z, 1/w)$
 - (x, y) はウィンドウの左下を原点とするビューポート上のフラグメントの位置
 - z はそのフラグメントにおける図形の透視深度（デプステストの対象）
- 視線ベクトル ray
 - ray は $(\text{gl_FragCoord.x} - \text{center.x}, \text{gl_FragCoord.y} - \text{center.y}, -\text{focal})$

gl_FragCoord から ray を求める

// 視点からこのフラグメントに向かう視線を求める

```
vec3 ray = mat3(rotate) * vec3(gl_FragCoord.xy - center.xy, center.z);
```

マウスの右ドラッグ
による回転

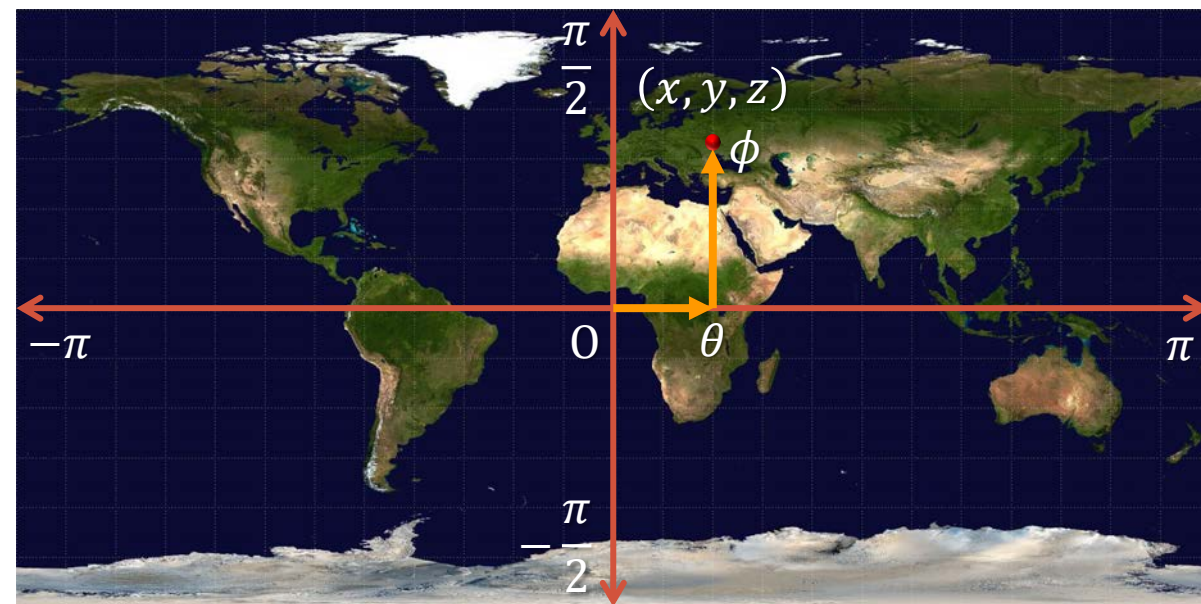
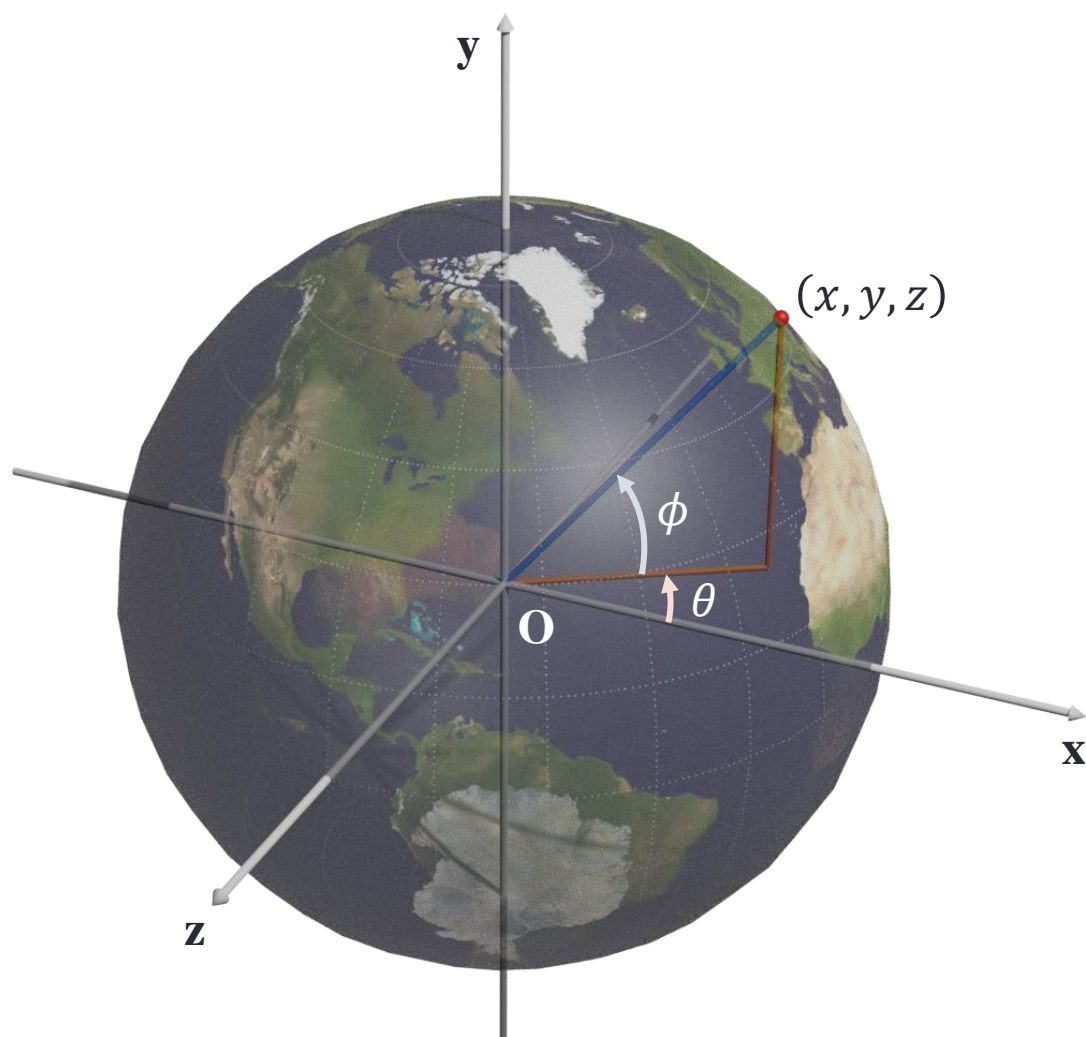
vec2

focal (視点からスクリーンの中心
までの距離) を格納している

視線によるテクスチャのサンプリング

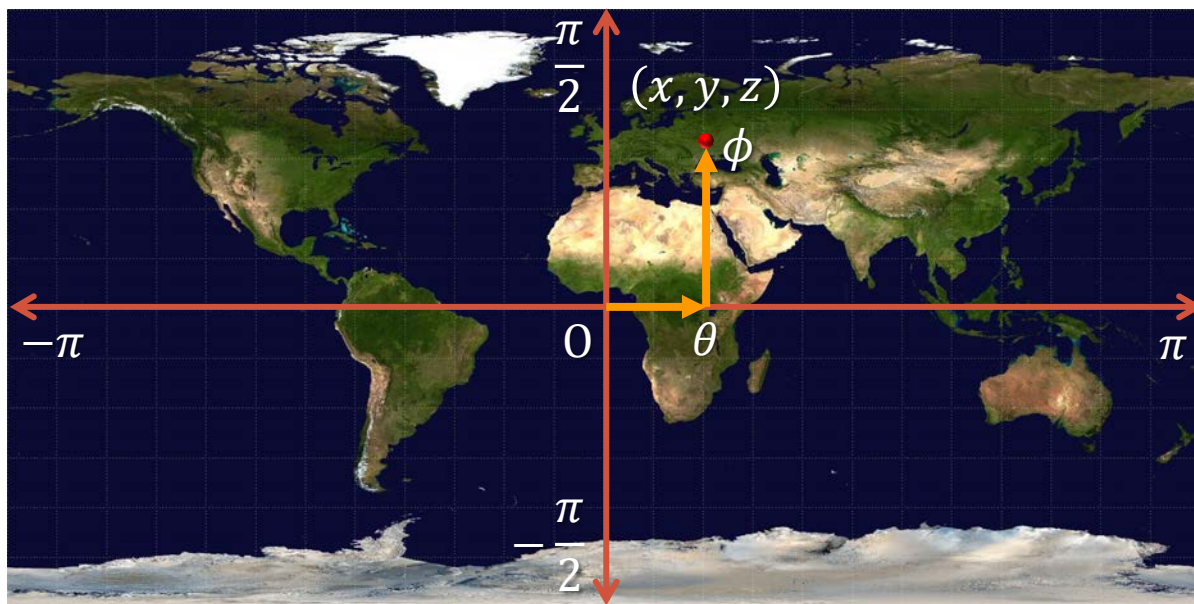
正距円筒図法テクスチャのテクスチャ座標

球面上の位置 (x, y, z) から方位角 θ と仰角 ϕ を求める



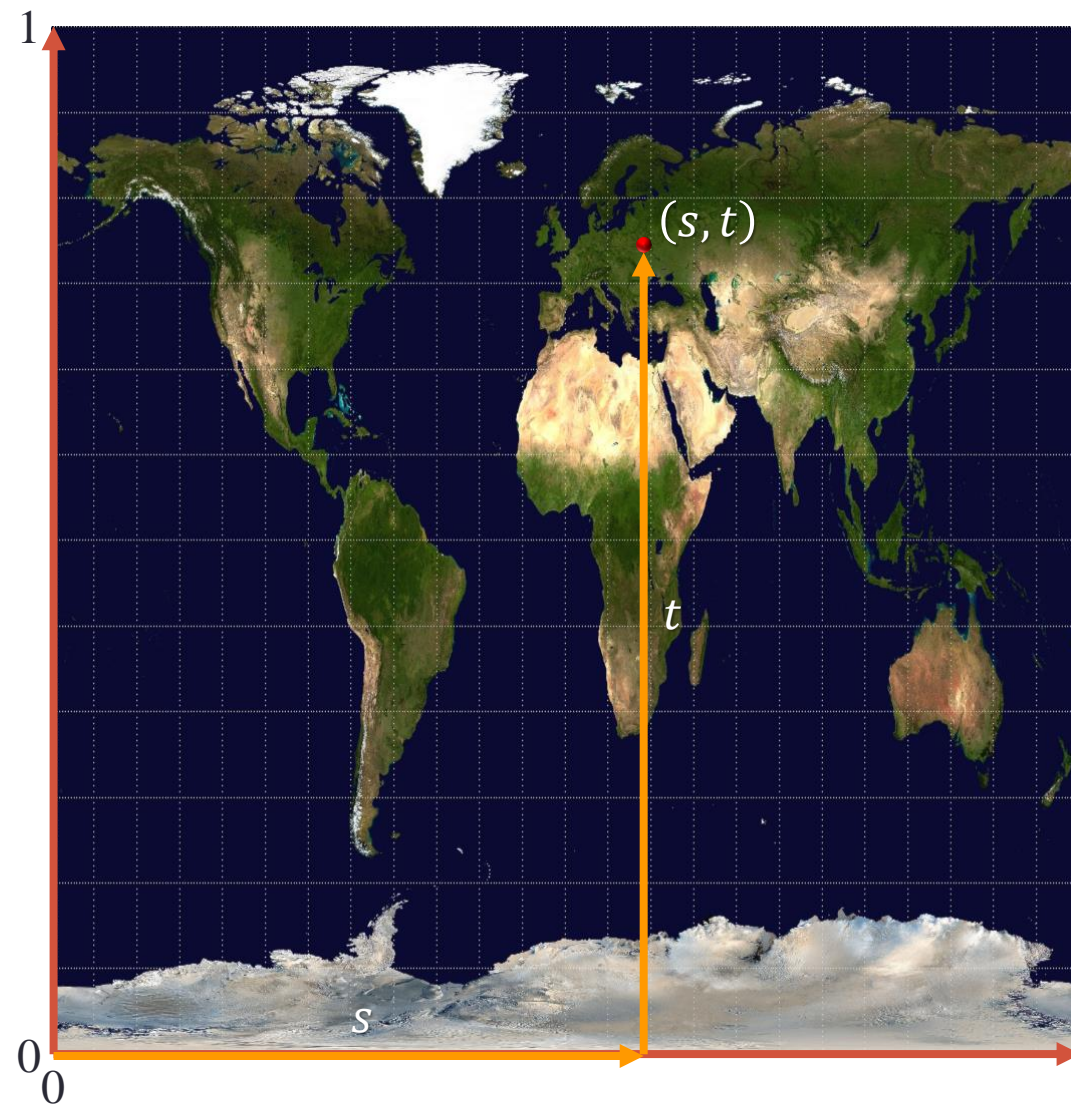
$$\theta = \tan^{-1} \frac{-z}{x} = -\tan^{-1} \frac{z}{x} \qquad \phi = \tan^{-1} \frac{y}{\sqrt{x^2 + z^2}}$$

方位角 θ と仰角 ϕ からテクスチャ座標 (s, t) を求める



$$s = \frac{\theta + \pi}{2\pi} = \frac{\theta}{2\pi} + \frac{1}{2}$$

$$t = \frac{\phi + \pi/2}{\pi} = \frac{\phi}{\pi} + \frac{1}{2}$$

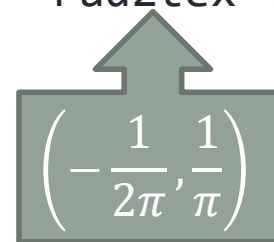


ray からテクスチャ座標 texcoord を求める

// 視線の方向から正距円筒図法のテクスチャ座標を求める

```
vec2 texcoord = atan(ray.xy, vec2(ray.z, length(ray.xz))) * rad2tex + 0.5;
```

$\tan^{-1} \frac{z}{x}$ と $\tan^{-1} \frac{y}{\sqrt{x^2+z^2}}$ を
一度に計算している


$$\left(-\frac{1}{2\pi}, \frac{1}{\pi} \right)$$

参考

reflect と refract

GLSL の組み込み関数

reflect

Name

reflect — calculate the reflection direction for an incident vector

Declaration

```
genType reflect( genType I,  
                  genType N );  
  
genDType reflect( genDType I,  
                  genDType N );
```

Parameters

I
Specifies the incident vector.

N
Specifies the normal vector.

<https://www.khronos.org/registry/OpenGL-Refpages/gl4/html/reflect.xhtml>

refract

Name

refract — calculate the refraction direction for an incident vector

Declaration

```
genType refract( genType I,  
                  genType N,  
                  float eta );  
  
genDType refract( genDType I,  
                  genDType N,  
                  float eta );
```

Parameters

I
Specifies the incident vector.

N
Specifies the normal vector.

eta
Specifies the ratio of indices of refraction.

これらは正規化されている必要がある

<https://www.khronos.org/registry/OpenGL-Refpages/gl4/html/refract.xhtml>