

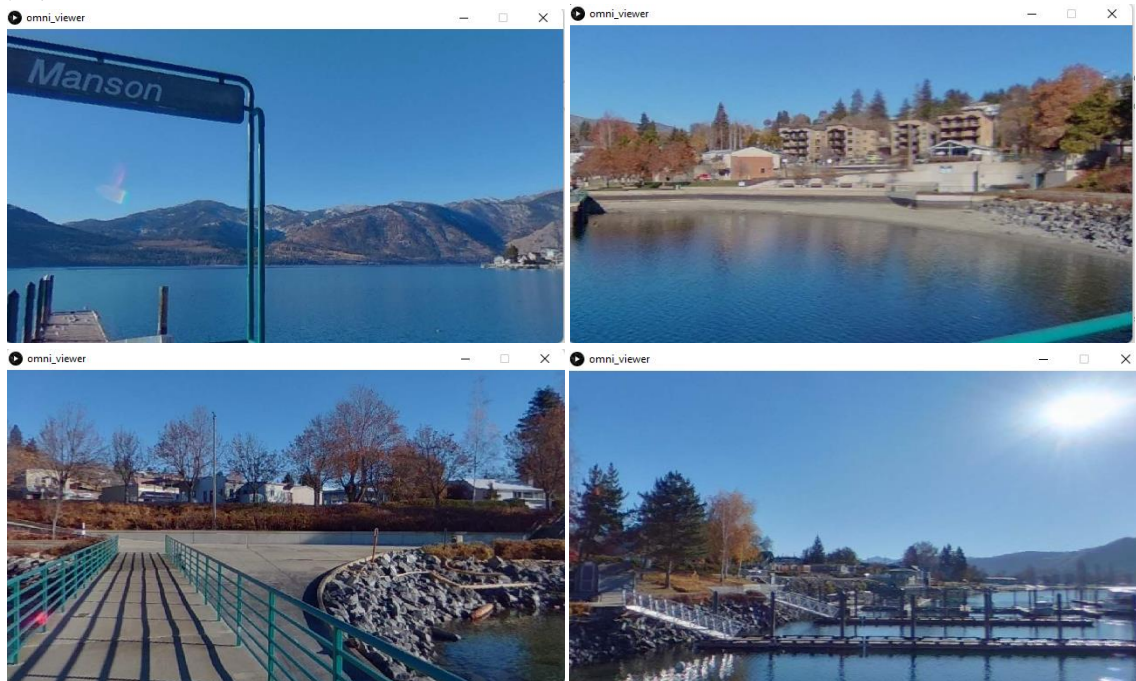
エルゴノミクスコンピューティングレポート 2

27019679 グレゴリウス ブライアン

使ったイメージは以下通りで、<https://unsplash.com/photos/luufnHoChRU> から入手した。



実行スクリーンショット：



動かし方（インタラクティブ）

キーボード操作で向きと zoom を変えられる。

WASD：向きを上下左右に動かす。

E：Zoom In

Q：Zoom Out

プログラムの構成

setup() :

テンプレートとそれほど変わらない。imgSrc に全天球画像を入れ、imgDst に仮想スクリーンとなる画像を生成。また現在の向きを保存する theta_v, phi_v, と水平方向の画角 theta_fovx を初期化する。

draw():

まず、**IO()** インタラクティブ入力処理関数を呼び出す。次に、imgSrc と imgDst のピクセル値をロードする。そして、theta_fovx を用いて、バーチャルスクリーンの幅と高さを決める。最後に、calcPixel() の関数を呼び出す。calcPixel() の中に各ピクセルの値を決める。

IO():

キーボードの入力を受け取り、対応するキーが押されたら phi_v, theta_v, theta_fovx を少し動かす。theta_v, phi_v の変化量を $\text{PI}/60$ で、theta_fovx の変化量を 1° とした。

calcPixel():

imgDst の各ピクセルについて次を繰り返す。

1. Z 軸を今向いている方向に固定し、新しくカメラ軸を作る。(繰り返しの中でする必要はないが、なぜかループの外ですれば、ベクトルの値が NaN になる)
2. ワールド軸の Z 方向にバーチャルスクリーンにおいて、そのスクリーンにおいて今のピクセルの位置ベクトルを決める。
3. 2 の位置ベクトルを参考に、現座標軸 (カメラ座標) の Z 軸方向にバーチャルスクリーンを作る。
4. バーチャルスクリーンにおけるピクセルを全天球の極座標で表す。ただし、ここで processing での atan2() は $-\text{PI}$ から PI の範囲をとるため、 $[-\text{PI}, 0]$ になったら 2PI を足して、 $[0, 2\text{PI}]$ の範囲にする。
5. 極座標に対応する imgsrc での位置を決める (浮動小数点)
6. (5) で得た位置を bilinear() に入れ、その位置に対応する色を取り出す。ただし、その位置が imgsrc 内になかったら、黒を描く。

bilinear():

x の位置と y の位置 (浮動小数点) を引数として受け取り、bilinear 法で重みを計算し、imgsearch のその位置に対応する色を返す。