

Pensieve header: Planet Earth centered at some specific point. Much obliged to pensieve://Projects/-PlanetHopf/.

```
(AllCountries = Union[Take[CountryData["Countries"], All], {"Antarctica"}]);
```

AllCountries

```
{Afghanistan, Albania, Algeria, AmericanSamoa, Andorra, Angola, Anguilla, Antarctica, AntiguaBarbuda, Argentina, Armenia, Aruba, Australia, Austria, Azerbaijan, Bahamas, Bahrain, Bangladesh, Barbados, Belarus, Belgium, Belize, Benin, Bermuda, Bhutan, Bolivia, BosniaHerzegovina, Botswana, Brazil, BritishVirginIslands, Brunei, Bulgaria, BurkinaFaso, Burundi, Cambodia, Cameroon, Canada, CapeVerde, CaymanIslands, CentralAfricanRepublic, Chad, Chile, China, ChristmasIsland, CocosKeelingIslands, Colombia, Comoros, CookIslands, CostaRica, Croatia, Cuba, Curacao, Cyprus, CzechRepublic, DemocraticRepublicCongo, Denmark, Djibouti, Dominica, DominicanRepublic, EastTimor, Ecuador, Egypt, ElSalvador, EquatorialGuinea, Eritrea, Estonia, Ethiopia, FalklandIslands, FaroeIslands, Fiji, Finland, France, FrenchGuiana, FrenchPolynesia, Gabon, Gambia, CapeVerde, CaymanIslands, Georgia, Germany, Ghana, Gibraltar, Greece, Greenland, Grenada, Guadeloupe, Guam, Guatemala, Guernsey, Guinea, GuineaBissau, Guyana, Haiti, Honduras, HongKong, Hungary, Iceland, India, Indonesia, Iran, Iraq, Ireland, IsleOfMan, Israel, Italy, IvoryCoast, Jamaica, Japan, Jersey, Jordan, Kazakhstan, Kenya, Kiribati, Kosovo, Kuwait, Kyrgyzstan, Laos, Latvia, Lebanon, Lesotho, Liberia, Libya, Liechtenstein, Lithuania, Luxembourg, Macau, Macedonia, Madagascar, Malawi, Malaysia, Maldives, Mali, Malta, MarshallIslands, Martinique, Mauritania, Mauritius, Mayotte, Mexico, Micronesia, Moldova, Monaco, Mongolia, Montenegro, Montserrat, Morocco, Mozambique, Myanmar, Namibia, Nauru, Nepal, Netherlands, NewCaledonia, NewZealand, Nicaragua, Niger, Nigeria, Niue, NorfolkIsland, NorthernMarianaIslands, NorthKorea, Norway, Oman, Pakistan, Palau, Panama, PapuaNewGuinea, Paraguay, Peru, Philippines, PitcairnIslands, Poland, Portugal, PuertoRico, Qatar, RepublicCongo, Reunion, Romania, Russia, Rwanda, SaintHelena, SaintKittsNevis, SaintLucia, SaintPierreMiquelon, SaintVincentGrenadines, Samoa, SanMarino, SaoTomePrincipe, SaudiArabia, Senegal, Serbia, Seychelles, SierraLeone, Singapore, SintMaarten, Slovakia, Slovenia, SolomonIslands, Somalia, SouthAfrica, SouthKorea, SouthSudan, Spain, SriLanka, Sudan, Suriname, Svalbard, Swaziland, Sweden, Switzerland, Syria, Taiwan, Tajikistan, Tanzania, Thailand, Togo, Tokelau, Tonga, TrinidadTobago, Tunisia, Turkey, Turkmenistan, TurksCaicosIslands, Tuvalu, Uganda, Ukraine, UnitedArabEmirates, UnitedKingdom, UnitedStates, UnitedStatesVirginIslands, Uruguay, Uzbekistan, Vanuatu, VaticanCity, Venezuela, Vietnam, WallisFutuna, WestBank, WesternSahara, Yemen, Zambia, Zimbabwe}
```

```
Area[reg_Region] := Abs[Dot[
  Sin[List @@ Last /@ reg],
  ((RotateLeft[#] - #) &) @ (List @@ First /@ reg)
]];

```

```
Area[c_CountryShape] := (Plus @@ Area /@ c);
```

```
Area[s_String] := Area[CountryShape[s]];

```

```
CountryShape[c_String] := Sort[

```

```
  Select[
    Map[(LonLat @@ N[# Degree]) &,
      Apply[Region,
        CountryShape @@ First[CountryData[c, "SchematicPolygon"]],
        {1}
      ],
    {2}
  ],
  Area[#] > 10^(-5) &
],
Area[#] &
]
```

```
(* /. LonLat[ln_, lt_] => LonLat[-Pi/2+ln, lt] *)
```

```
MakeColourScheme[ac_List, newborders_List] := Module[
  {c, nbd, cnbd, addborders, k, oc, nc, changes, change},
  (nbd[#] = CountryData[#, "BorderingCountries"]) & /@ ac;
  addborders[{c1_, c2_}] := (
    nbd[c1] = Union[nbd[c1], {c2}];
    nbd[c2] = Union[nbd[c2], {c1}];
  );

```

```

);
addborders[l_List] := Do[
  addborders[{l[[i]], l[[j]]}],
  {i, 2, Length[l]}, {j, i}
];
addborders /@ newborders;
(CountryColour[#] = RandomReal[]) & /@ AllCountries;
changes = {}; change = 0;
Do[
  Do[
    cnbd = Sort[CountryColour /@ nbd[c]];
    If[cnbd ≠ {},
      AppendTo[cnbd, 1 + First[cnbd]];
      {k} = Ordering[Rest[RotateRight[cnbd] - cnbd], 1];
      oc = CountryColour[c];
      nc = FractionalPart[(cnbd[[k]] + cnbd[[k + 1]]) / 2];
      oc = Abs[oc - nc]; If[oc > 0.5, oc = 1 - oc];
      change += oc;
      CountryColour[c] = nc
    ],
    {c, AllCountries}
  ];
  AppendTo[changes, change];
  change = 0,
  {32}
];
Print["Country colouring process changes: ", changes]
];
MakeColourScheme[ac_] := MakeColourScheme[ac, {}];
LoadColourScheme[s_String] := Module[
  {rule},
  rule = Get[s <> ".m"];
  (CountryColour[#] = (# /. rule)) & /@ (First /@ rule);
];
SaveColourScheme[] := SaveColourScheme[
  DateString[{"YearShort", "Month", "Day", "-", "Hour24", "Minute", "Second"}]
];
LoadColourScheme[dt_String] := Module[
  {},
  Put[
    (# → CountryColour[#]) & /@ AllCountries,
    dt <> ".m"
  ];
];
Export[dt <> ".png",
  Graphics[{
    EdgeForm[White],
    {
      Hue[CountryColour[#]],
      CountryShape[#] /. {CountryShape → List,

```

```

        Region[pts___] := Polygon[{pts} /. LonLat -> List]]
    } & /@
    (AllCountries)
  },
  Background -> Black, ImageSize -> 640
]
]
]

```

```

MakeColourScheme[AllCountries, {
  {"Canada", "Greenland"}, {"Canada", "Russia"},
  {"UnitedStates", "Russia"}, {"Canada", "Mexico"},
  {"Malaysia", "PapuaNewGuinea"}, {"Japan", "Russia"}, {"Japan", "SouthKorea"},
  {"Kazakhstan", "Mongolia"}, {"Australia", "NewZealand"},
  {"Australia", "Indonesia"}, {"Australia", "PapuaNewGuinea"},
  {"China", "Japan"}, {"China", "SouthKorea"}, {"NorthKorea", "Japan"},
  {"UnitedStates", "Greenland"}, {"Greenland", "Russia"},
  {"Pakistan", "Tajikistan"}, {"Canada", "Japan"}, {"Canada", "Italy"},
  {"Italy", "Japan"}, {"Italy", "Germany"}, {"France", "UnitedKingdom"}
}]

```

Country colouring process changes:

```

{29.9257, 9.0729, 2.80438, 1.47158, 1.57599, 0.90749, 1.74371, 0.631762,
 0.49355, 0.401256, 0.32993, 0.274895, 0.234484, 0.204787, 0.189087, 0.181907,
 0.177293, 0.172619, 0.168004, 0.16348, 0.159041, 0.154668, 0.150675, 0.147762,
 0.145184, 0.142461, 0.139823, 0.137075, 0.134181, 0.131178, 0.12812, 0.125055}

```

```

MapCenteredAt[coord_String, excl___] :=
  MapCenteredAt[FromDMS[coord] Degree // N, excl];
MapCenteredAt[{BLat_, BLon_}] := MapCenteredAt[{BLat, BLon}, {}];
MapCenteredAt[{BLat_, BLon_}, excl_List] := Module[{M, P, Draw},
  M = RotationMatrix[

$$\begin{pmatrix} \text{Cos}[BLat] \text{Cos}[BLon] & 0 \\ \text{Cos}[BLat] \text{Sin}[BLon] & 0 \\ \text{Sin}[BLat] & 1 \end{pmatrix}$$
 // Transpose];
  P[LonLat[θ_, φ_]] :=

$$\left( \begin{array}{l} \{x, y, z\} = M.\{\text{Cos}[\phi] \text{Cos}[\theta], \text{Cos}[\phi] \text{Sin}[\theta], \text{Sin}[\phi]\}; \\ (*\sqrt{\frac{1}{2(1+z)}} \{x, y\}*) \frac{\text{ArcCos}[z]}{\pi \sqrt{1-z^2}} \{x, y\} \end{array} \right);$$

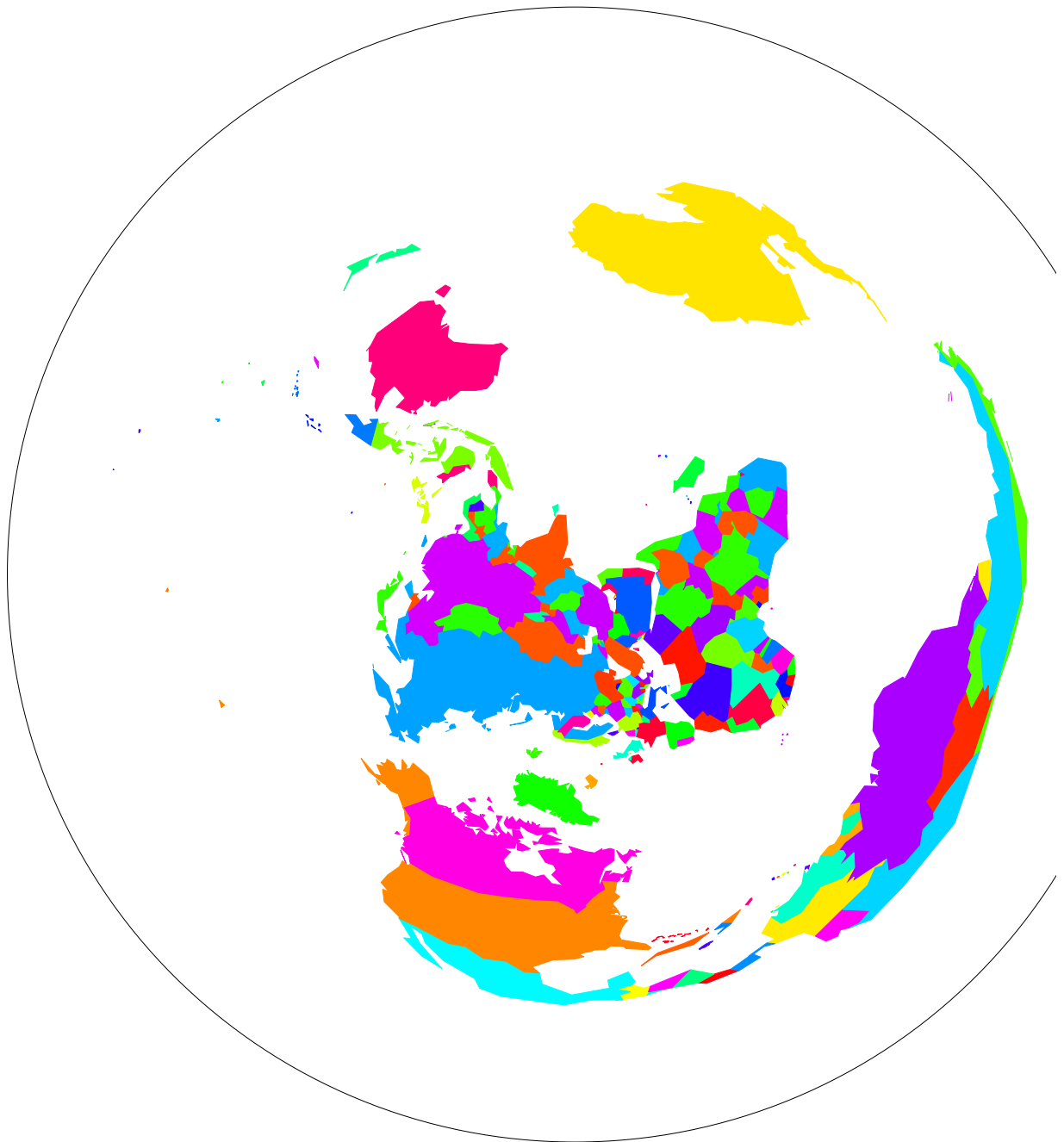
  Draw[c_String] := {Hue[CountryColour[c]], (List @@ CountryShape[c]) /.
    {Region[pts___] -> Polygon@{pts}} /. p_LonLat -> P[p]};
  Graphics[{Circle[{0, 0}, 1], Draw /@ Complement[AllCountries, excl]}]
]

```

```
MapCenteredAt["43° 42' 0" N, 79° 24' 0" W"]
```



MapCenteredAt["24°51,36′N 67°0,36′E"]

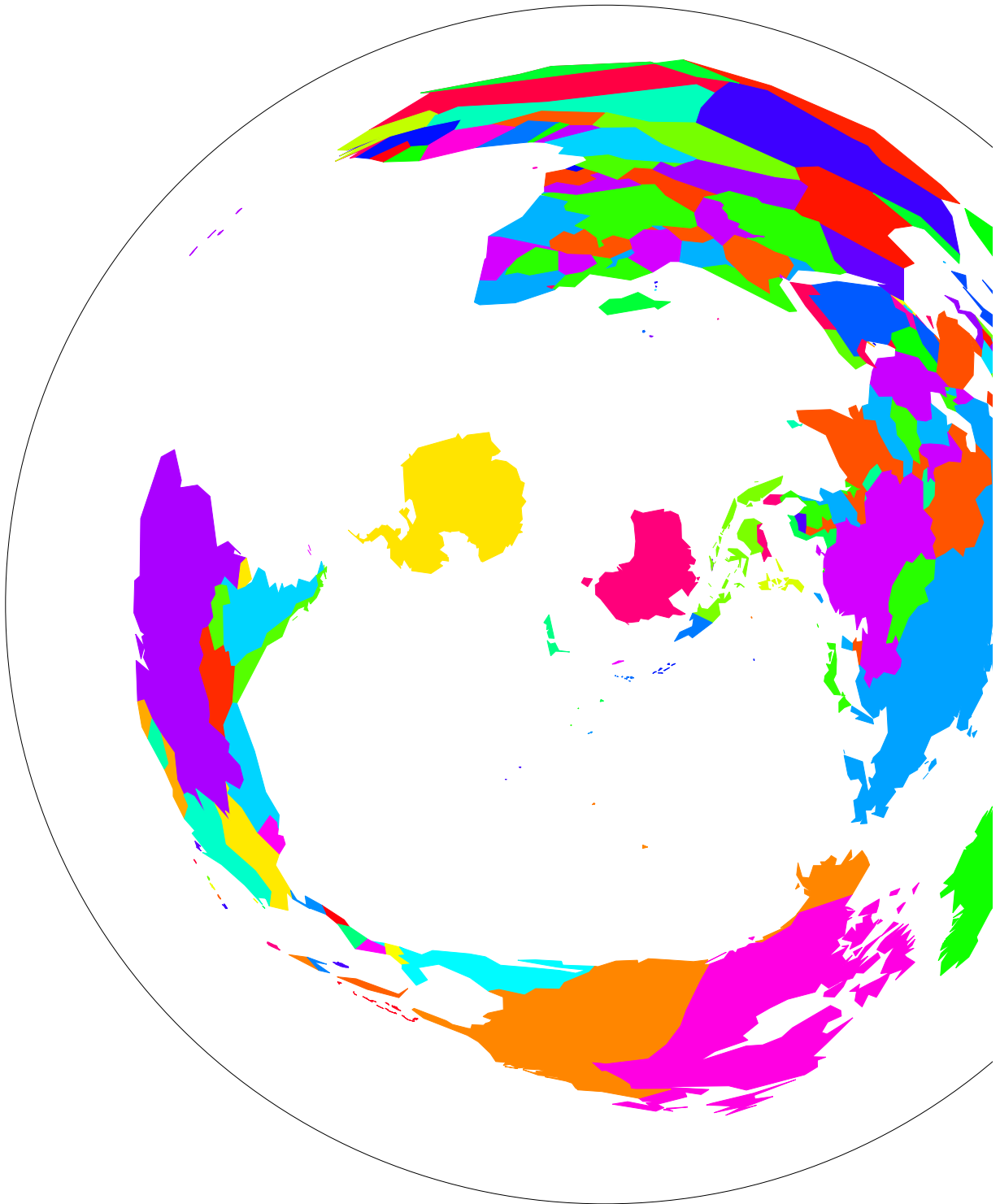


```
MapCenteredAt@"31° 47' 0" N,35° 13' 0" E"
```



```
CityData["Sydney", "Coordinates"]  
{-33.87, 151.21}
```

```
MapCenteredAt[CityData["Sydney", "Coordinates"] Degree]
```



```
CityData["Jerusalem", "Coordinates"]  
{31.78, 35.22}
```

```
MapCenteredAt[{-31.78`, 180 + 35.22`} Degree, {"Israel", "WestBank", "Jordan"}]
```

