Lesson 3: Example Program - Map Coloring

What's It All About?

1. The four color theorem is introduced. The problem of coloring the countries of South America with four colors is considered.
2. A program to color the South American countries with four colors so that no two bordering countries are the same color is written. The program shows that the Prolog inferencing engine can be used to perform search.

Map Coloring

In general, the **map coloring problem** is to color a given map with some number of colors. This problem has intrigued mathematicians for a very long time!

The **four color theorem** states that it is possible to color any map with four colors in such a way that no adjacent regions are the same color. The very long standing problem of proving this theorem was finally solved in 1976 with the help of a computer program. The proof was very controvercial, because the program used rendered the proof infeasible to check by hand. Eventurally, the proof has come to be accepted by most everyone. The Wikipedia article on the four color theorem provides a sense of the significance of the four color problem to mathematicians.

`https://en.wikipedia.org/wiki/Four_color_theorem`

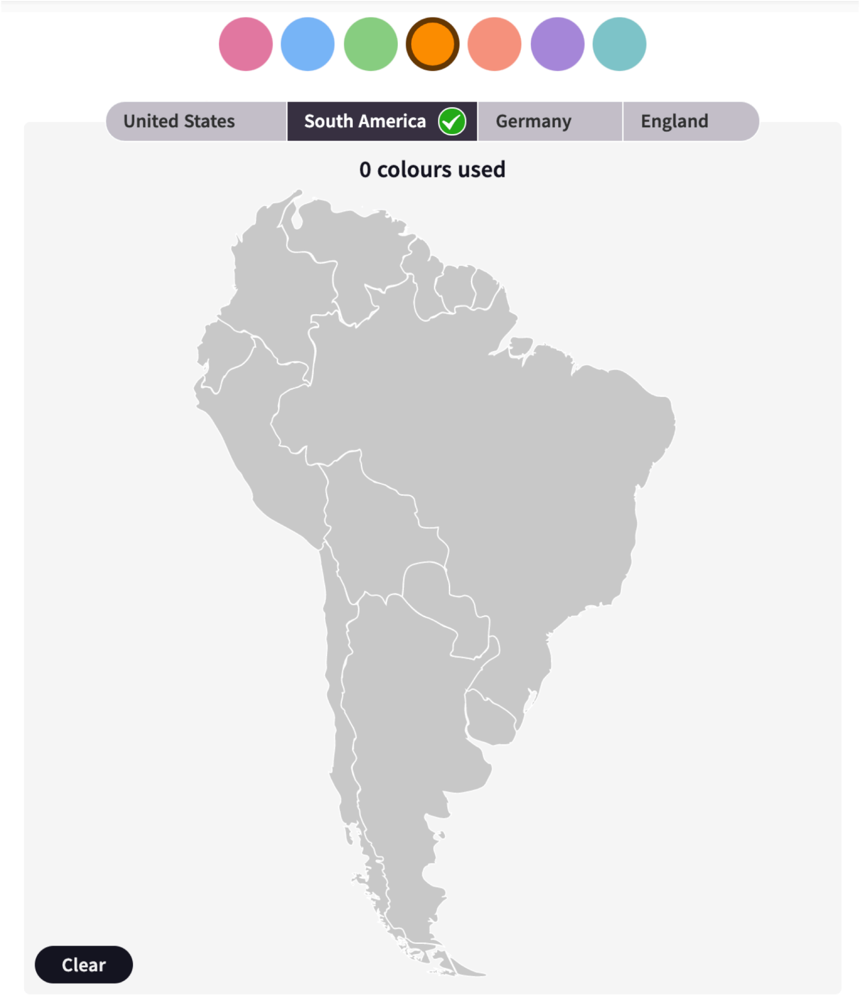
The following site will let you color maps in a convenient, fun way. You might like to spend a little time making friends with the site.

`https://mathigon.org/course/graph-theory/map-colouring`

Sample Problem

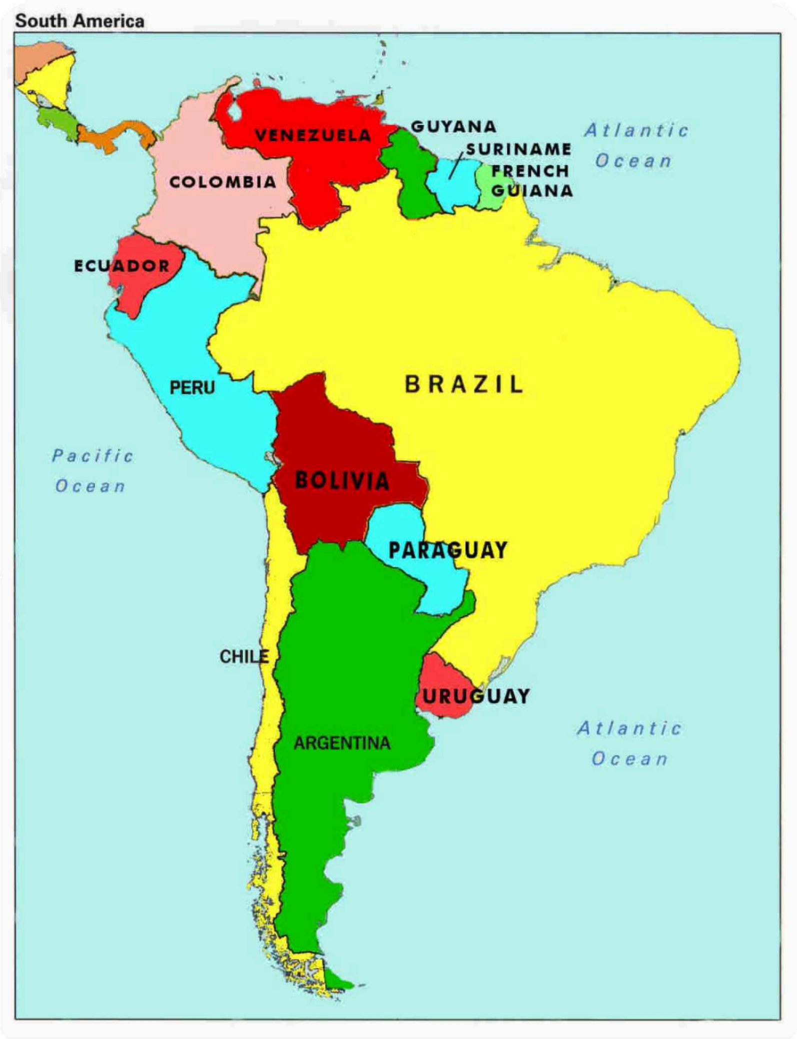
Color a map representing the countries of South America with four colors.

The uncolored map:



0 colours used

Map of South America - with Names



Source for the Map Coloring Solver

```
% ---------------------------------------------------------------------------
% File: map_coloring.pro
% Line: Program to find a 4 color map rendering for South American coutries.
% More: The colors used will be red, blue, green orange.
% More: The standard abbrieviations are used to stand for the countries.

% ---------------------------------------------------------------------------
% different(X,Y) :: X is not equal to Y

different(red,blue).
different(red,green).
different(red,orange).
different(green,blue).
different(green,orange).
different(green,red).
different(blue,green).
different(blue,orange).
different(blue,red).
different(orange,blue).
different(orange,green).
different(orange,red).

% ---------------------------------------------------------------------------
% coloring(AR,UY,BR,CL,PARA,BO,PERU,ECUA,VE,COL,GY,SR,GF) :: The South
% American coutries represented by their standard abbrieviations are colored
% so that none of the countries sharing a border are the same color.

coloring(AR,UY,BR,CL,PARA,BO,PERU,ECUA,VE,COL,GY,SR,GF) :-
  different(AR, CL),
  different(AR, UY),
  different(AR, BR),
  different(AR, PARA),
  different(AR, BO),
  different(CL, BO),
  different(CL, PERU),
  different(UY, BR),
  different(PARA, BR),
  different(PARA, BO),
  different(BO, BR),
  different(BO, PERU),
  different(PERU, BR),
  different(PERU, ECUA),
  different(PERU, COL),
  different(ECUA, COL),
  different(COL, BR),
  different(COL, VE),
  different(VE, BR),
  different(VE, GY),
  different(GY, BR),
  different(GY, SR),
  different(SR, BR),
  different(SR, GF),
  different(GF, BR).
```

Demo of the Map Coloring Solver

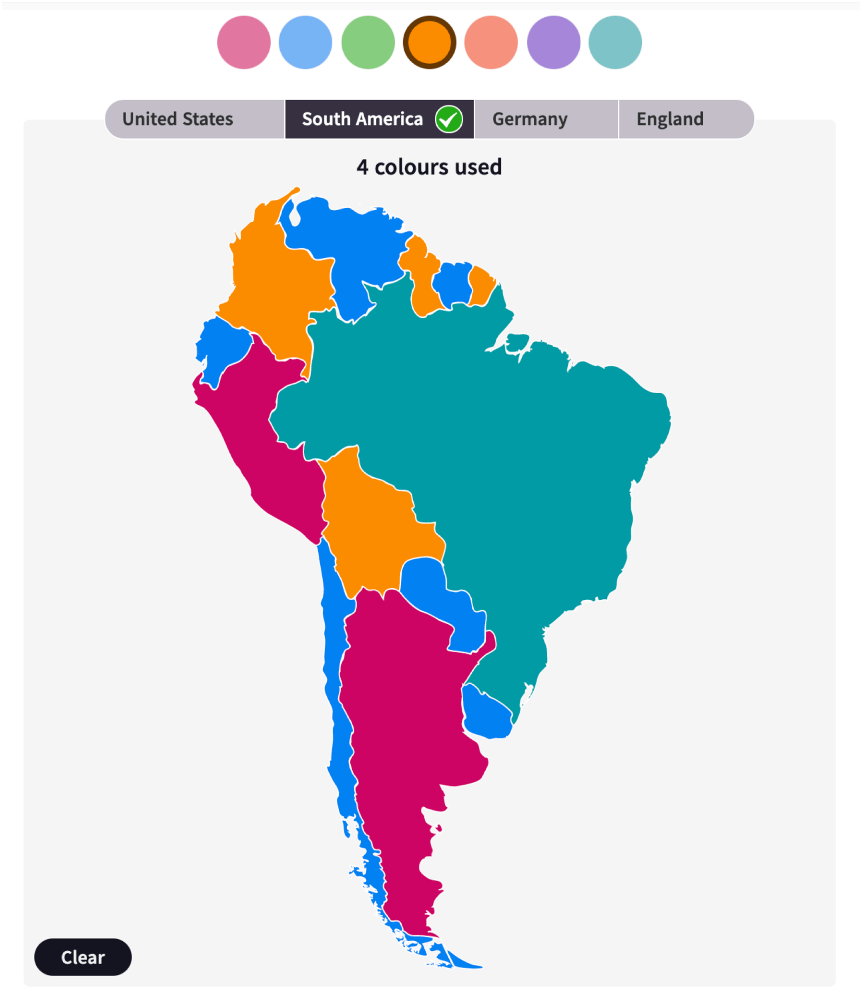
```
bash-3.2$ swipl
...

?- consult(’map_coloring.pro’).
% map_coloring.pro compiled 0.00 sec, 1 clauses
true.

?- coloring(AR,UY,BR,CL,PARA,BO,PERU,ECUA,VE,COL,GY,SR,GF).
AR = PERU, PERU = red,
UY = CL, CL = PARA, PARA = ECUA, ECUA = VE, VE = SR, SR = blue,
BR = green,
BO = COL, COL = GY, GY = GF, GF = orange

?- halt.
bash-3.2$
```

Image of the Solution



4 colours used