Dawson Harris & Anders Choy - CSE 190C: P2 Description

1. The problem we formulated for part B was Instant Insanity.
2. Our state includes four separated lists that represent the four cubes correspondingly. Each cube list consists of 6 elements, which represent the different sides of the cube [left, front, right, back, top, bottom]. The numbers in the lists represent the four different colors: 0 = red, 1 = green, 2 = blue, 3 = yellow. On the other hand, the goal state checks the first four elements (sides: left, front, right, back) to see if each element (each side) of each list (each cube) is different from each other. Therefore, there can be no same color on the same side for every cube.
3. In Instant Insanity, our method uses 24 different operators. These operators essentially include every possible rotation of each cube to change their orientation. Each cube can be rotated left, right, up, down, clockwise, and counterclockwise. We have initially thought about using two layers of operators to allow the player to choose which cube to rotate before the move. This method, however, is more complicated and might not be compatible with the given SOLUZION clients, because it would require 2 inputs from the user. We went with our choice because it was the simplest and most efficient.
4. Yes, we did provide a visualization for our Instant Insanity code. The visualization lays out the four cubes with their six sides in a table with 4 columns and 15 rows. The four boxes in each middle row represent the four sides (left, front, right, back) that are checked for a correct solution (goal state). The orientation of colors changes with the moves (operators) the user selects. A textual state is also provided at the bottom of the window.
5. We worked on both parts at the same time and we would both assist each other if we ever got stuck. For example, we were both working on FamerFox at the same time, and when Dawson got stuck on trying to figure out how to check for legal operation moves, Anders would come from a different area of the code to focus on this single problem so we could quickly continue to complete the rest of the project. We used the same working method when programming the Instant Insanity puzzle and our visualizations.
6. The most challenging parts to us were preventing illegal moves in the FarmerFox program, as well as simply trying to solve the InstantInsanity puzzle. It was also a bit difficult to figure out how to represent all the steps and states.