First, the user picks a color as the target color, and then starting with one of the RGB primary colors (red, green, and blue) they try to describe the target color in a comparative way. The goal is for the user to guide the machine to change that primary color step by step to get closer and finally reach the target color; the sooner and the fewer steps, the better.

So the learning scenario here is ‘to learn to make a successful communication;’ a communication between the user and the machine. So it’s a learning process on both sides. The human side learns to express themselves in a more subjective, not subjective, but communicative way. And the machine side learns a bit of common sense in the field of color perception.

And the strategy is simply to read the expressions, so the user commands, and extract the only two important parts: an adjective and the adverb. So the part of the command that asks for changing the color in a specific way, would be the adjective. And the part that asks that change in a specific amount would be the adverb. So for example if a user says ‘make it a bit brighter’ the software is supposed to recognize ‘bright’ as the adjective and ‘a bit’ as the adverb. And change the color accordingly.

For doing that I took a naive approach, I’m using two dictionaries for possible adjectives and adverbs, with a reasonable size. So analyzing the user commands does not happen on a syntactic level, at least for now. It’s rather simply the software searching the dictionaries to find occurrences of the adjectives and the adverbs it already stored.

The code is mainly just a giant while loop, being repeated until the color gets close enough to the target color. So kind of simple and it’s already done, but there are difficult tasks like handling invalid answers, expanding the dictionaries to a decent size, and also the interface… I originally wanted to work with the MIT App Inventor so the product would be an APP, but I have to wait to make sure that all the Python libraries I’m gonna use are available in the App Inventor… But these are problems that are more time-consuming than essentially difficult.

Then I also need to be careful about the color perception principles to implement a reliable translation between, at least normal ways of describing colors and the RGB language. For example, what does it mean for a color to be cold or dusty… or if I’m going to warm a color a bit, would it be increasing red 5% or 10%? The optimal solution for these kinds of problems is to run experiments or read the literature but at some point, I am definitely going to use some made-up definitions, as well.
And I’m also having this more technical problem here that for example, we don’t have a fixed formula for making a color brighter. It is always different based on the color you’re working on.
Yet these are not huge problems.

But the most difficult task is to parse the commands (so working on the syntax). For example, when I say ‘a bit more blue,’ my simple look-up strategy won’t work anymore. Cause we have two opposing adverbs here *bit* and *more*. So I have to find a way that the

software recognizes that the *bit* is the relevant adverb but the *more* is only the comparative sign for the adjective.

But the task could get even more complicated when the software fails to recognize any adjective or adverb in a command. Like when a user says something like: ‘just a tad dirtier’ and I have neither tad nor dirty in my dictionaries.
The simplest solution is to ask the user to express themselves better (so learning a bit more on the human side) but I’m also trying to use NLP libraries to actually parse sentences so make the software capable of asking smarter questions. Like instead of ‘I didn’t get it, could you please clarify ‘ it asks ‘what does ‘tad’ mean?’ or ‘what does ‘dirty’ mean’ And the user answers something like ‘*tad* means *a bit*’ or ‘*dirty* means *dark* and *pale*’. And then the software adds these two new terms in its dictionaries. And this would be learning more on the machine side. And the most difficult part of the task.