This tool is intended for kids, which means that we need to handle rather short commands written in simple language and possibly containing spelling mistakes. The fact that commands are very short and have almost no context affects NLP processing both positively and negatively.

I used the following NLP tools and techniques:

1) Spell checker
   I briefly tested the spell checker provided by SpaCy, but it seemed to be quite inaccurate to me. Therefore, I decided to use a highly-rated model from HuggingFace. It worked better, but I still had to add some phrases for correction manually, e.g. mov > move.

2) POS tagging
   In order to extract verbs from the input texts, I used POS-tagging. The problem here was the lack of context, e.g. if the command was 'jump up', the tagger would label 'jump' as a noun. In order to overcome this, I added 'Please to ' to every input. Even though it produced incorrect sentences it improved the performance of the analyzer and the sentences were not displayed anyway.

3) Word embeddings
   Despite the fact that we expect to get simple commands as input there is still room for some variation. I added phrases "the man goes forward" and "the frog jumps up and hops over" as anchors for the verbs of jumping (e.g. jump, hop, leap) or movement from side to side (e.g. go, move, slide).

The main challenge for me was to find balance between the using of rigid rules and applying tools that allow more flexibility like embeddings. In the beginning, it seems that the number of situations to handle is almost fixed and one needs just a bunch of rules and conditions to make the whole program work. After diving deeper into the task, it appears that there are more variables that one can think of and there will be even more if one needs to scale the project.