# WILL.A.I

Team:  Nicolò Donati and Riccardo Grassi

# Time management

In order to keep the choreography under the 3 min time limit we measured how much time  the execution of mandatory positions take.
The available time is the result of the subtraction.

We divided the available time in smaller equal intervals, one for each step.

A step is composed by the set of moves from one mandatory position to the next one.
At completion of each step we add the spare time to the next.

To compute the execution time for a move, we executed them in various conditions and averaged results.

# Constraints

## Time

Check if there is enough time in the current step to chose this move.

## Preconditions

Check if the preconditions are satisfied. (standing/sitting)

## Creativity

Check if the move is different from the last five in the choreography.

# Entropy Constraint

In order to have a diverse choreography in each step we defined a function to compute entropy, the implementation is based on Claude Shannon definition.

This constraint is checked only for feasible set of moves.

During the testing phase we tuned this as a parameter, making tradeoffs between finding reliably a solution and having a creative one.

# Search Algorithm

We chose a  best first search strategy, in particular A*.

Best-first means that the chosen node is the one considered as the most desirable by

$$f(n) = g(n) + h'(n).$$

```
def h(self, node):
    # We implemented a simple heuristic that estimates the cost to reach the goal
    # by multiplying the number of remaining moves to the avg_time of our moves set
    state_dict = from_state_to_dict(node.state)
    goal_dict = from_state_to_dict(self.goal)
    return (goal_dict['moves_done'] - state_dict['moves_done']) * self.avg_time
```

The avg_time of the moves set is a simple mean over the estimated time of execution for each move.

# Let’s dance