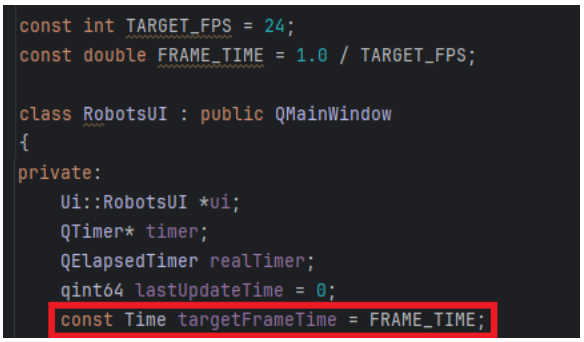
Evaluation of the group 7 by the group 6

• The group carefully met the deadline for submitting the "Ramping-up" project, demonstrating good time management and respect for the given timelines.

• All the requested features were successfully implemented. During our evaluation, all the questions we raised about the functioning of certain elements were answered clearly and precisely, except for one specific query: "In the file robotsui.h, why is the constant targetFrameTime equal to the constant FRAME_TIME? Can one of them be removed, or is there a reason they both exist?"



```
const int TARGET_FPS = 24;
const double FRAME_TIME = 1.0 / TARGET_FPS;

class RobotsUI : public QMainWindow
{
private:
    Ui::RobotsUI *ui;
    QTimer* timer;
    QElapsedTimer realTimer;
    qint64 lastUpdateTime = 0;
    const Time targetFrameTime = FRAME_TIME;
```

• A project demonstration was presented to us without any issues. All the requested features were properly implemented.

• Reading JSON files, such as timeline and state, was performed without any problems.

• The world is displayed correctly in an interactive window. The graphical elements, such as the buttons for opening and saving timelines and states, as well as the playback control buttons, are present and functional. Additionally, the window resizing works smoothly.

• Error and warning management was effectively handled. Error messages are correctly displayed in the console, accompanied by a beep sound.

• During playback, the trajectories of the robots following circular arcs, as well as their speeds and rotations, were implemented according to the formulas defined in the specifications.

• The animation refresh rate was set to 24 frames per second by default, as requested.

• The playback speed set by the user is respected without any lag.

• The provided source code has not been modified, which is in line with the requirements.

• The project was correctly tagged with the name "ramping up" on the GitHub fork, as per the final instructions.

• The project build is fully functional, with no errors or warnings, ensuring successful compilation and execution from start to finish.

• The tests we conducted using the provided test data passed successfully, confirming that all features are correctly implemented.