POLITECNICO
MILANO 1863

AVATAR
Advanced Virtual and Augmented Reality Toolkit for Learning

Development of a 3D environment for extended reality

Smart manufacturing laboratory

Eren Enver - 10947639
Saverio Rocchi - 10631378

Introduction

The context

AVATAR, an Erasmus+ project

Laboratory objective

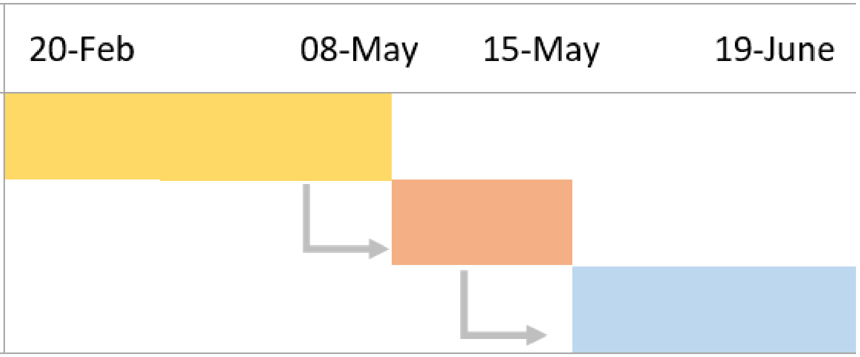
- Learn about VR/AR technologies
- Understand how to generate a workflow

How?

1. Create a guide for the Joint Learning Lab
2. Participate the Joint Learning Lab
3. Feedback and improvements

Introduction

Phase	Task	Start	Finish
1	Creation of the guide	20/2	8/5
2	Joint Learning Lab	8/5	15/5
3	Updating the guide	15/5	19/6

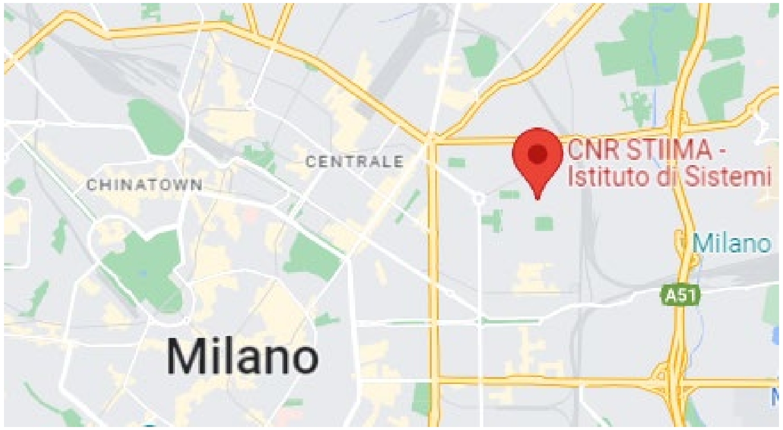


Guide creation – part 1

Guide

Contents of the guide

How to create a virtual representation of STIIMA - CNR robotic laboratory using free to use tools.



CNR STIIMA -
Istituto di Sistemi e...
Politecnico
di Milano

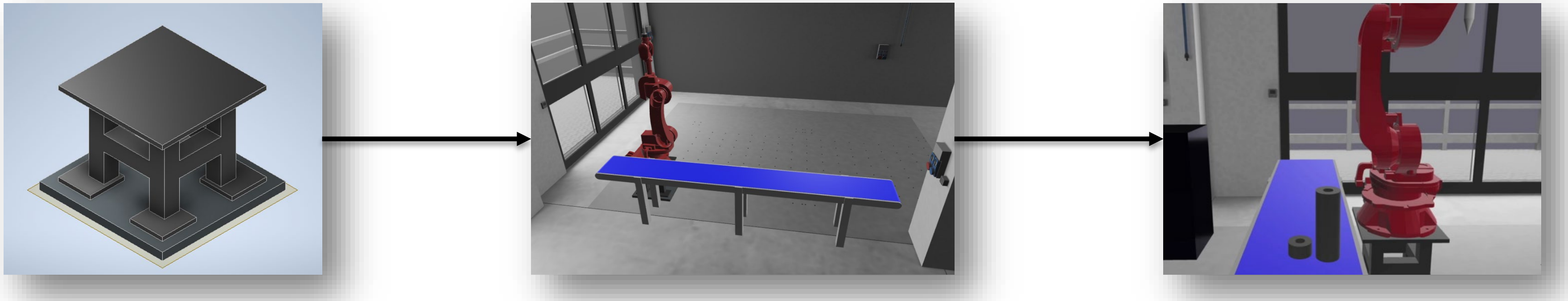
https://www.stiima.cnr.it/

Guide

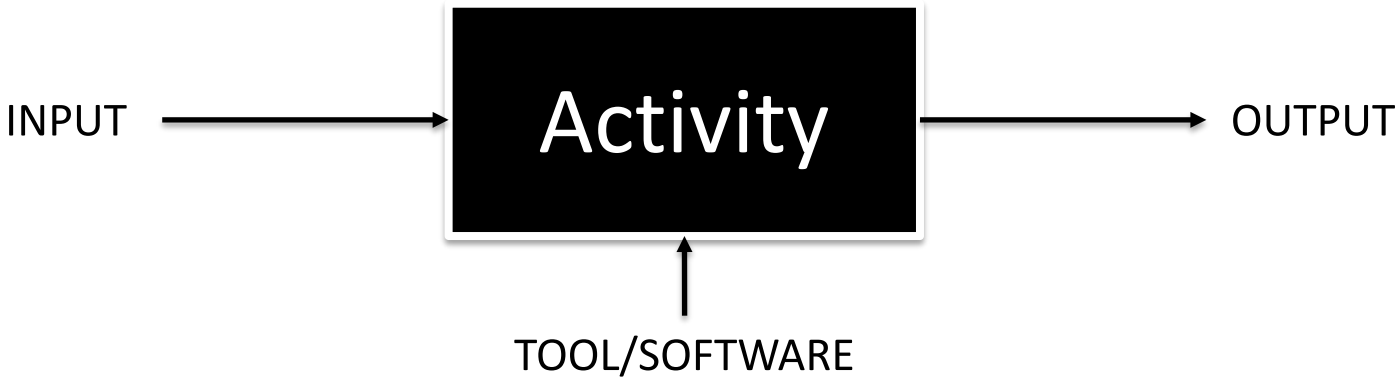
Models

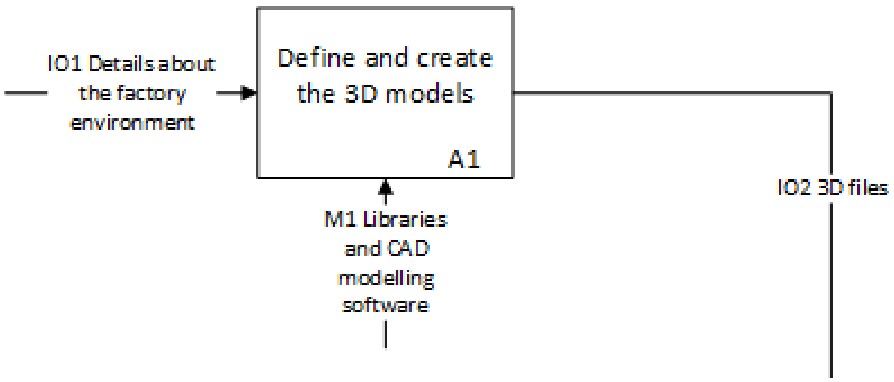
Assembly

Animate



The workflow




IO1 Details about the factory environment
Define and create the 3D models
A1
M1 Libraries and CAD modelling software
IO2 3D files

https://github.com/savixy/AVATARrepository/blob/main/images/documentation.md

Activities explained in the guide

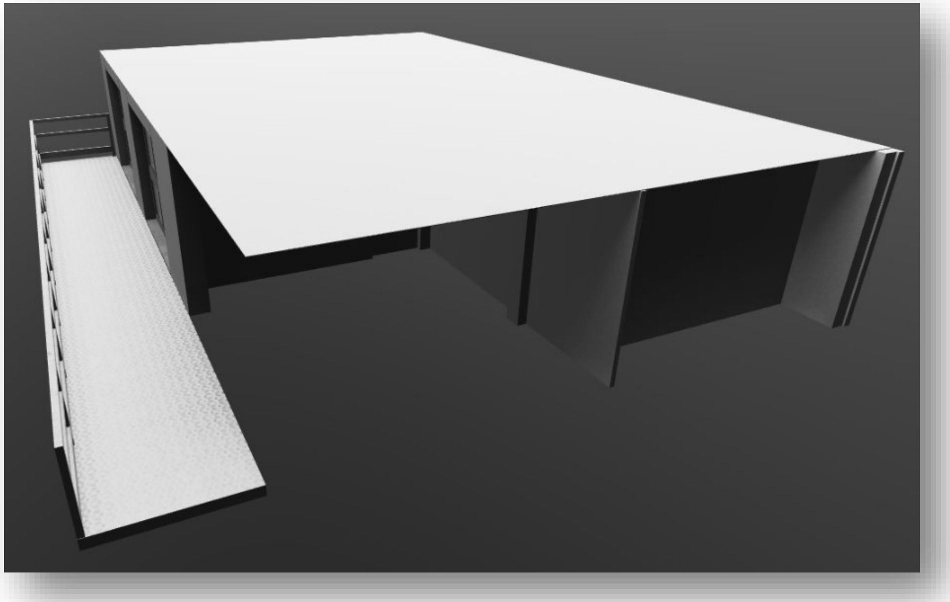
Activities – Define and create the 3D models

1) List of the models
   - Dimensions

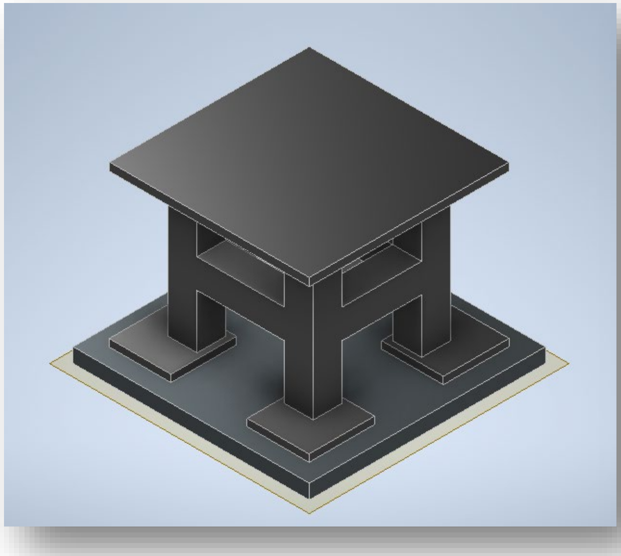
2) Already existing models:
   - Laboratory structure
   - Robot

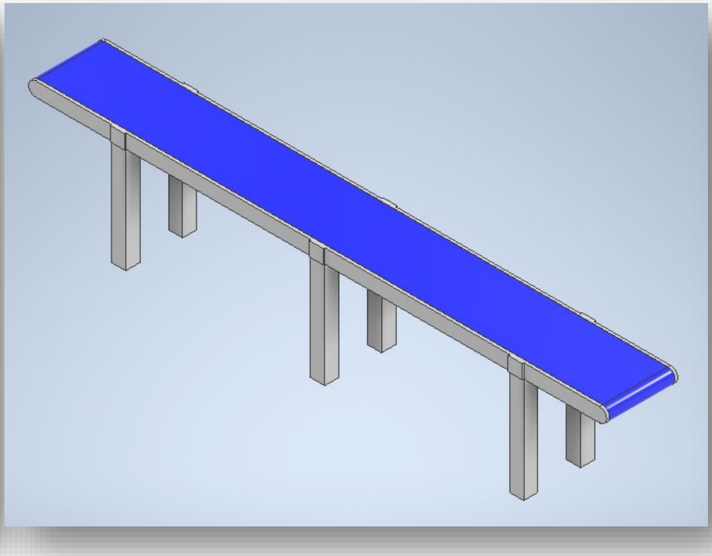
3) From scratch:
   - Conveyor
   - Robot Base
   - Tool

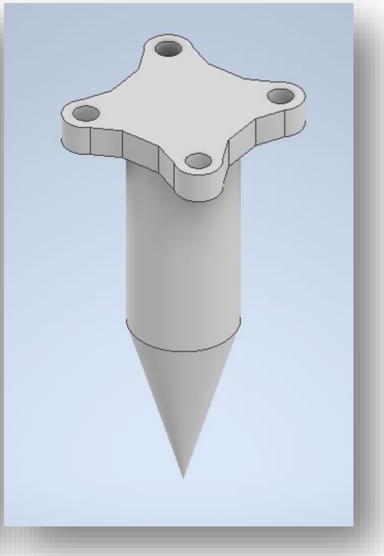
Activities – Define and create the 3D models



Activities – Define and create the 3D models







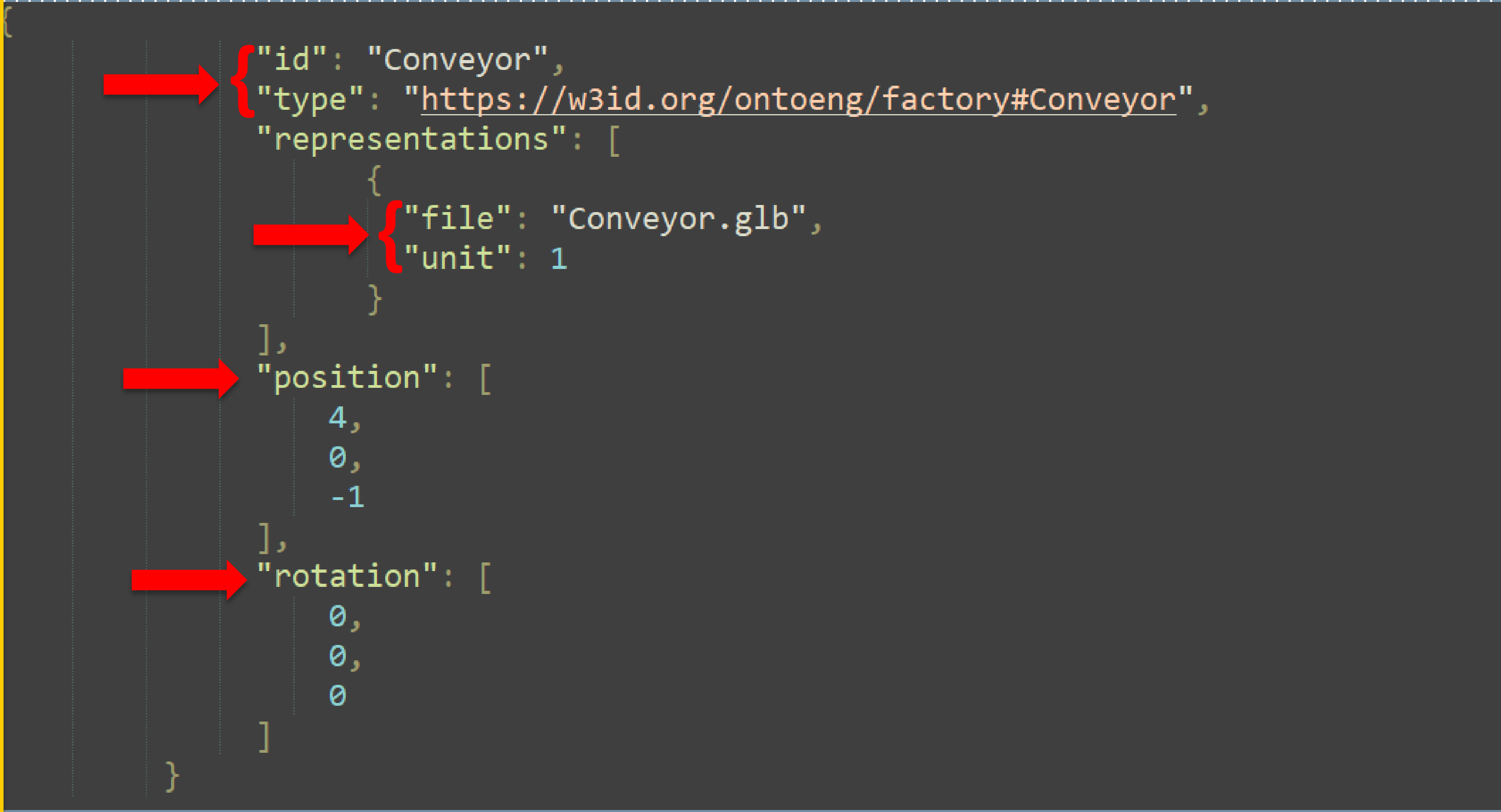
Activities – import in the environment

To import the models:

1) Use of Excel file provided

2) Programming of a json file

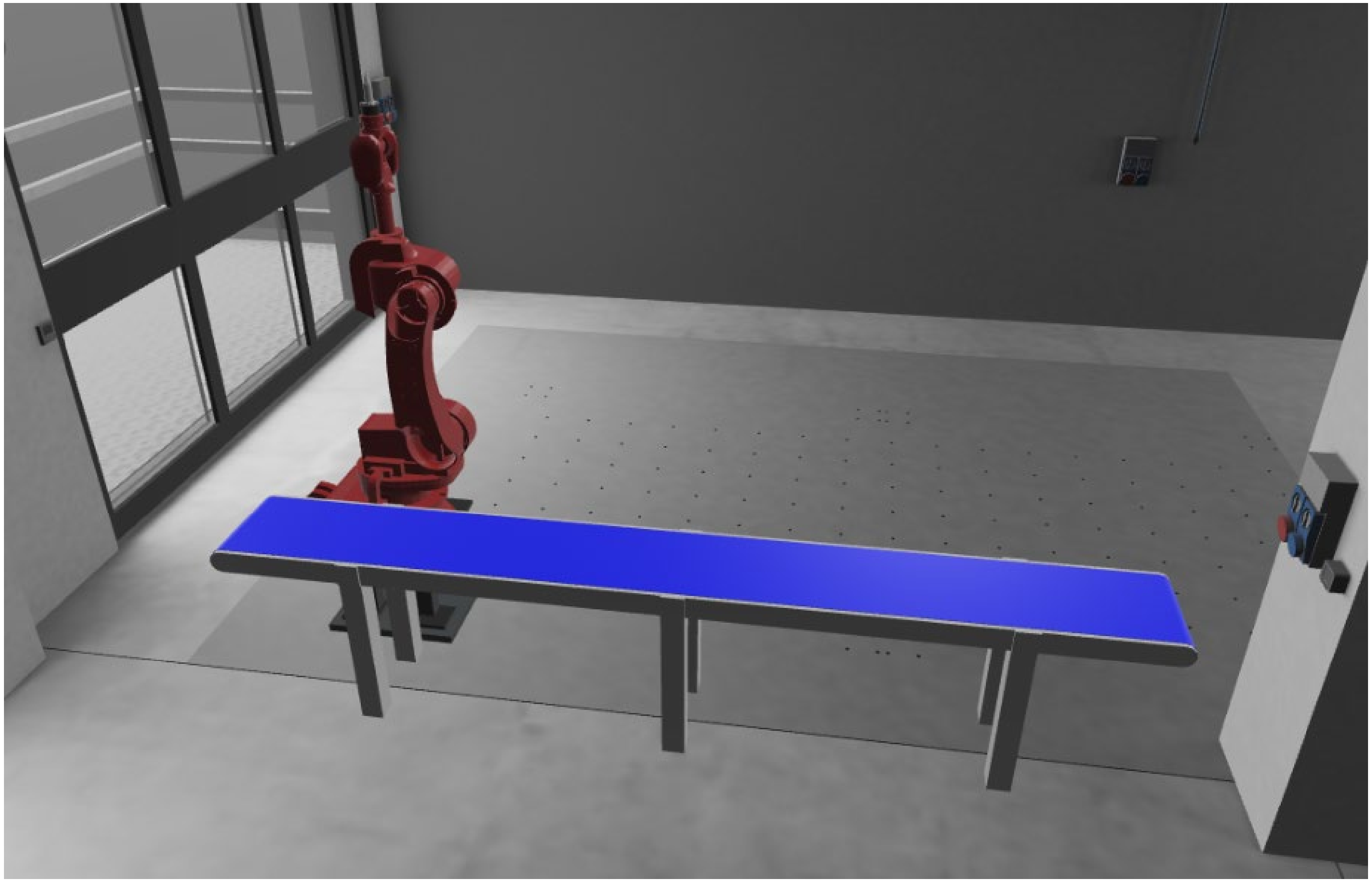
Activities – import in the environment



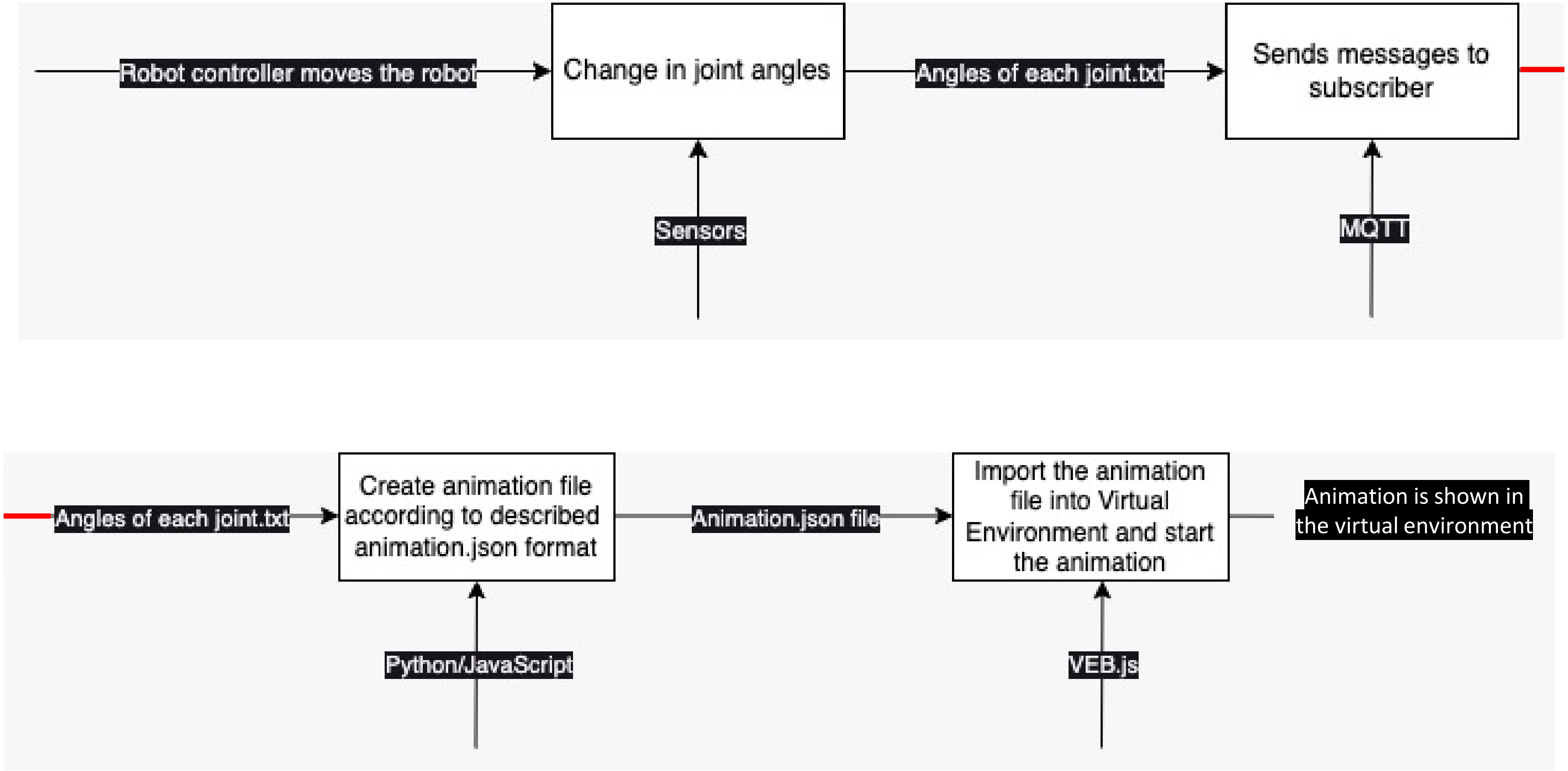
```
{
        }
            "id": "Conveyor",
            "type": "https://w3id.org/ontoeng/factory#Conveyor",
            "representations": [
                {
                    "file": "Conveyor.glb",
                    "unit": 1
                }
            ],
            "position": [
                4,
                0,
                -1
            ],
            "rotation": [
                0,
                0,
                0
            ]
        }
```

Activities – Upload and check the 3D environment

Position check and Json code updating



Activities – Animations
(Workflow for conversion and crucial points)



Robot controller moves the robot

Change in joint angles

Sensors

Angles of each joint.txt

Sends messages to subscriber

MQTT

Angles of each joint.txt

Create animation file according to described animation.json format

Python/JavaScript

Animation.json file

Import the animation file into Virtual Environment and start the animation

VEB.js

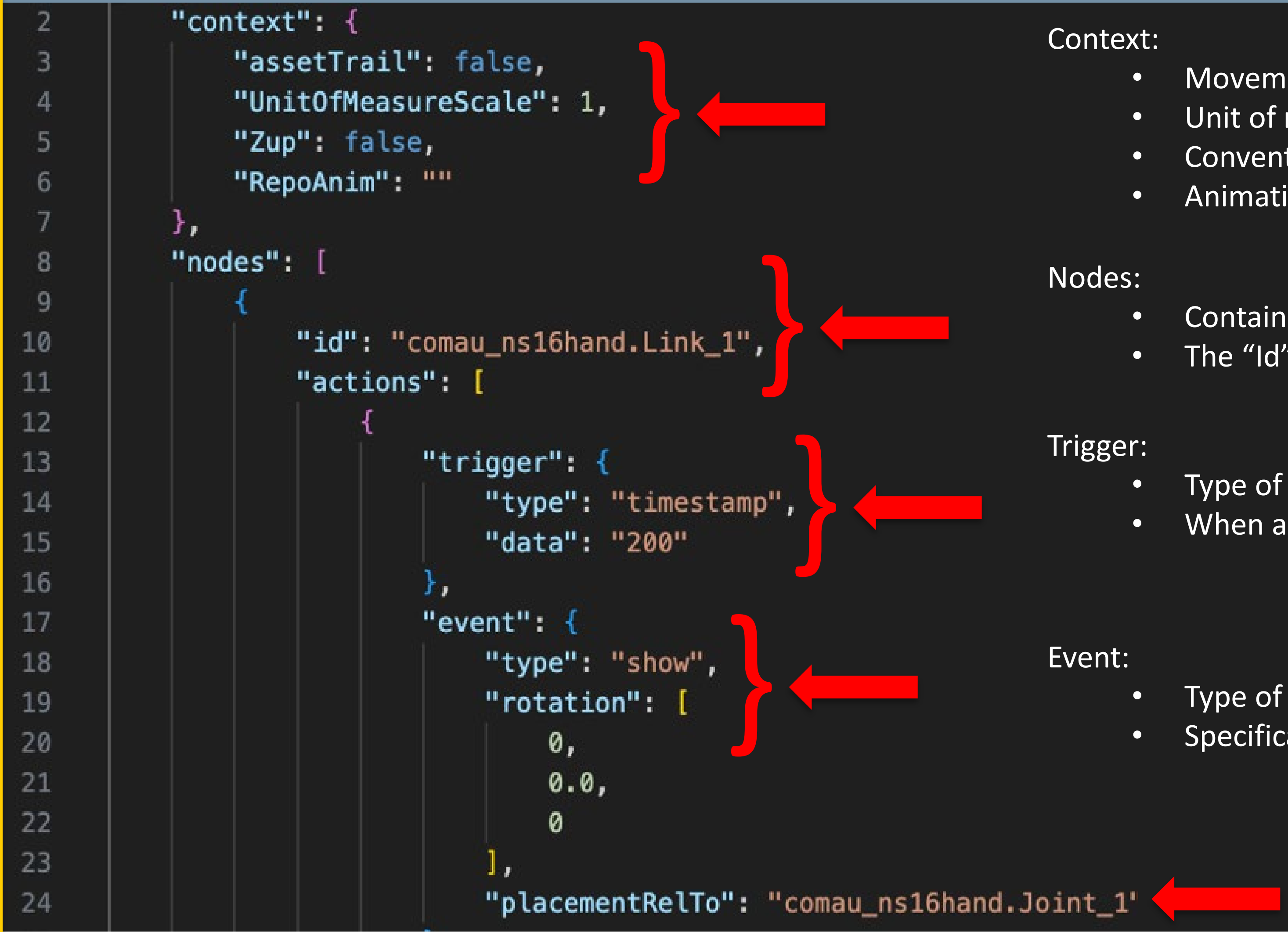
Animation is shown in the virtual environment

Activities – Animations
(Conventions for coordinate plane)

Different conventions	
VEB.js convention: Yup	**URDF** convention: Zup
Computer graphics applications, including game engines, modelling software, and physics simulations.	Geospatial applications, such as geographic information systems (GIS) and mapping software, where the elevation or altitude is important.

It's important to **be aware of** the **convention** being used in a particular context to ensure proper orientation and alignment of objects in a 3D space.

Activities – Animations
(Information about form of anim.json file)



```
    "context": {
        "assetTrail": false,
        "UnitOfMeasureScale": 1,
        "Zup": false,
        "RepoAnim": ""
    },
    "nodes": [
        {
            "id": "comau_ns16hand.Link_1",
            "actions": [
                {
                    "trigger": {
                        "type": "timestamp",
                        "data": "200"
                    },
                    "event": {
                        "type": "show",
                        "rotation": [
                            0,
                            0.0,
                            0
                        ],
                        "placementRelTo": "comau_ns16hand.Joint_1"
```

Context:

- Movement tracking
- Unit of measure
- Convention
- Animation files location

Nodes:

- Contains all the moving parts
- The “Id” identify the moving part

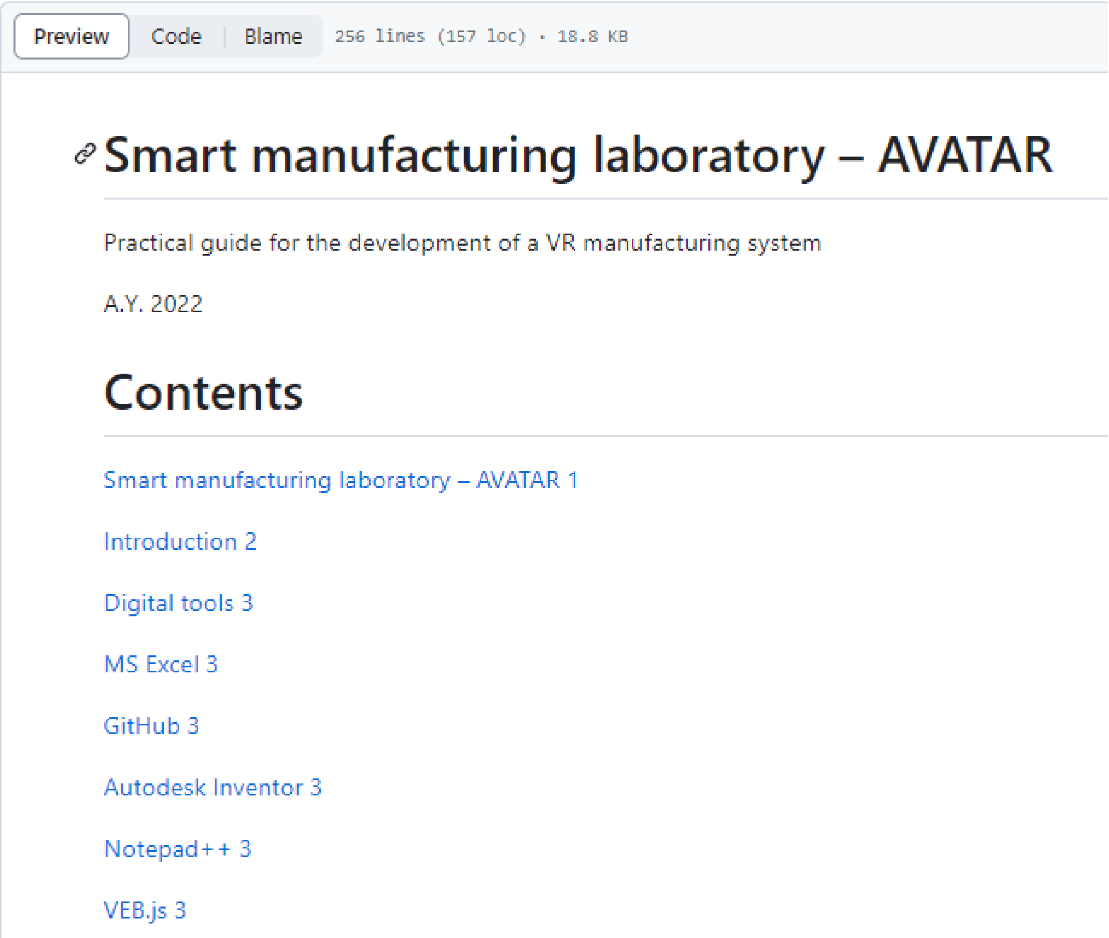
Trigger:

- Type of trigger
- When action will start

Event:

- Type of event
- Specifications of the event

Documentation



Smart manufacturing laboratory – AVATAR

Practical guide for the development of a VR manufacturing system

A.Y. 2022

Contents

https://github.com/savixy/AVATARrepository/blob/main/images/documentation.md

Joint Learning Lab – part 2

Timetable

	Monday May 8th 2023	Tuesday May 9th 2023	Wednesday May 10th 2023	Thursday May 11th 2023	Friday May 12th 2023
	POLIMI	**STIIMA-CNR**	**STIIMA-CNR**	**POLIMI**	**POLIMI**
09:15 – 10:15	**Welcome Introduction Lectures** Room Sala Ovale	**Visit to the laboratories** (VR lab, Robotics lab)	**Groupwork** with tutors (Meeting rooms)	**XR Laboratory Activities** Room L.04	**Groupwork** Room MEL LAB 1
10:15 – 11:15					
11:15 – 12:15	Visit to the VR Cave	**Groupwork** with tutors (Meeting rooms)			
12:15 – 13:15					
13:15 – 14:15	LUNCH	LUNCH	LUNCH	LUNCH	LUNCH
14:15 – 15:15	**Lectures** Room L.09	**Groupwork** with tutors (Meeting rooms)	**Groupwork** with tutors (Meeting rooms)	**XR Laboratory Activities** Room L.04	**Presentations of the Groupworks** Room MEL LAB 1
15:15 – 16:15			**Visit to the Robotics lab**		
16:15 – 17:15			**Group work** with tutors (Meeting rooms)		**Conclusions Farewell** Room MEL LAB 1
17:15 – 18:15					

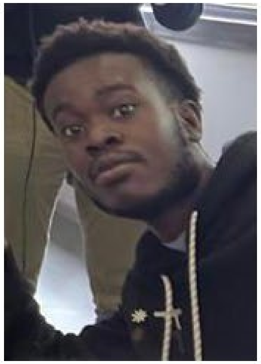
Teams

Group C

Matija Žuža

Kristina Golo

Enver Eren



Kan Kouakou

Marco Varisco

Alessandra Lupo

Group D

Kaveh Bekhrad

Nađa Belić

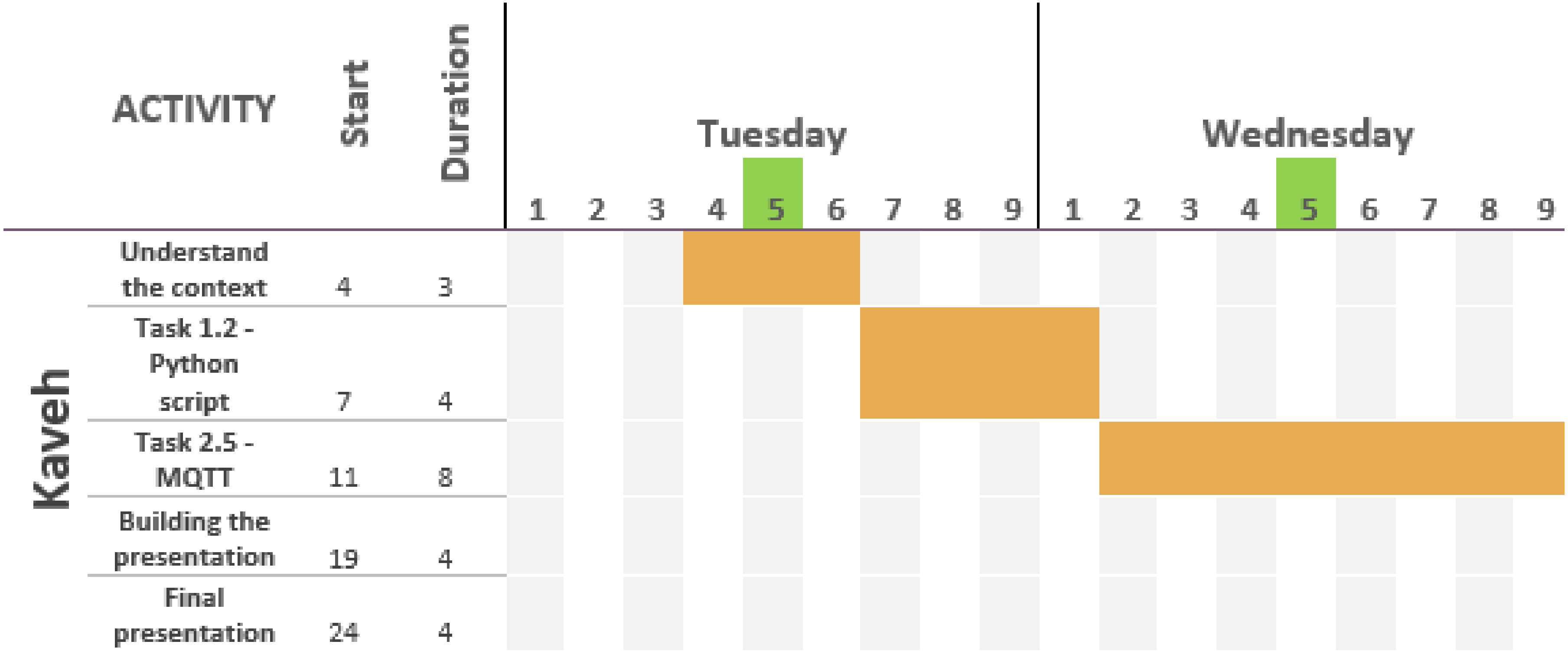
Aleksandar Ćosić

Leonardo Lomacci

Saverio Rocchi

Matteo Speranza

Gantt



	ACTIVITY	Start	Duration	Tuesday									Wednesday								
				1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
Kaveh	Understand the context	4	3				■	■	■												
	Task 1.2 - Python script	7	4							■	■	■	■								
	Task 2.5 - MQTT	11	8											■	■	■	■	■	■	■	■
	Building the presentation	19	4																		
	Final presentation	24	4																		

Challenges

Challenge #1 - Visualize Trajectories

1. Visualize the scene
2. Elaborate trajectories
3. Visualize trajectories
4. Check trajectories effectiveness

Challenge #2 - Receive a Trajectory via MQTT

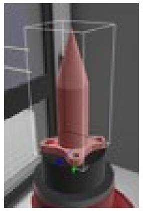
1. Receive and check the trajectory
2. Create an MQTT client

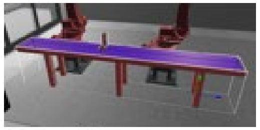
Challenge #3 - Generate a Trajectory

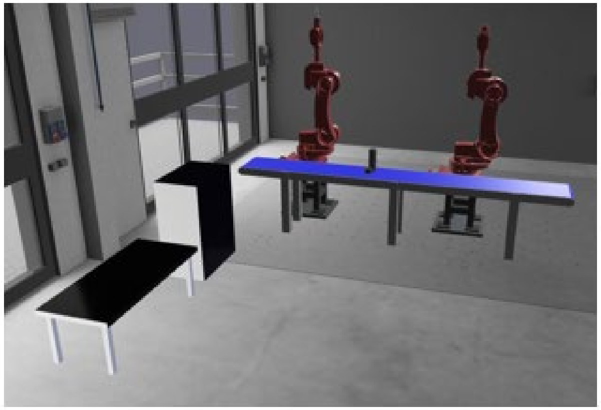
1. Given starting and target position
2. Check the generated trajectory

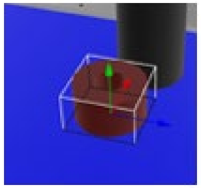
Challenge #1 – Scene visualization

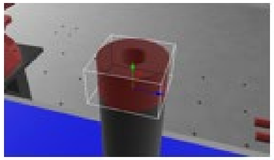
VEB.js


Tool

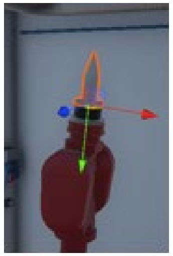

Conveyor

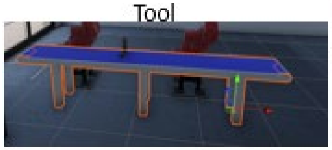

Scene

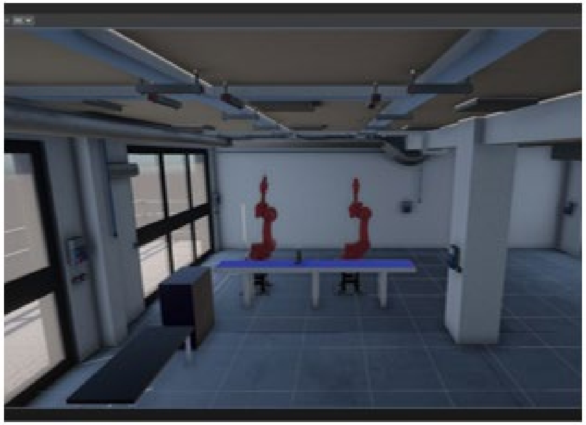

Workpiece 1

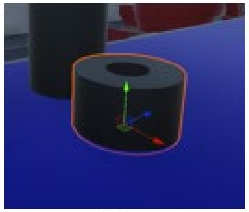

Workpiece 6

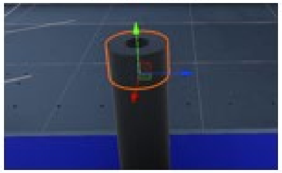
Unity


Tool

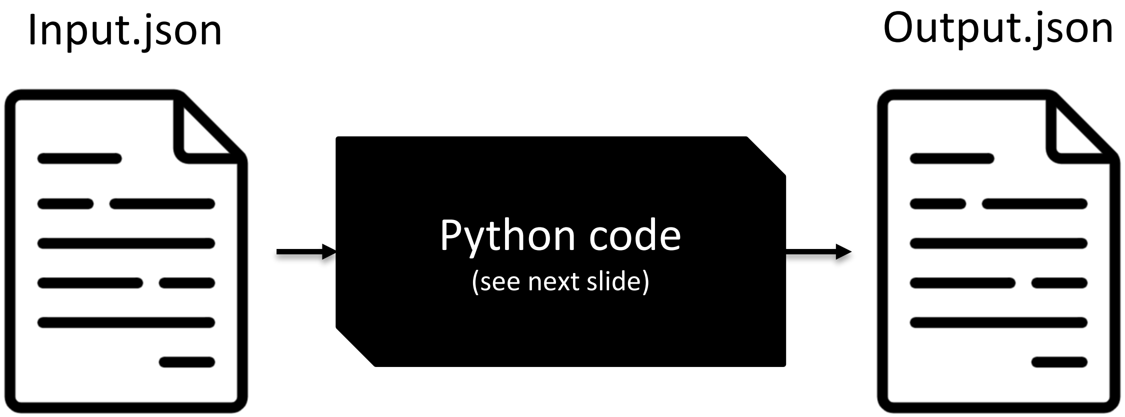

Conveyor


Scene


Workpiece 1

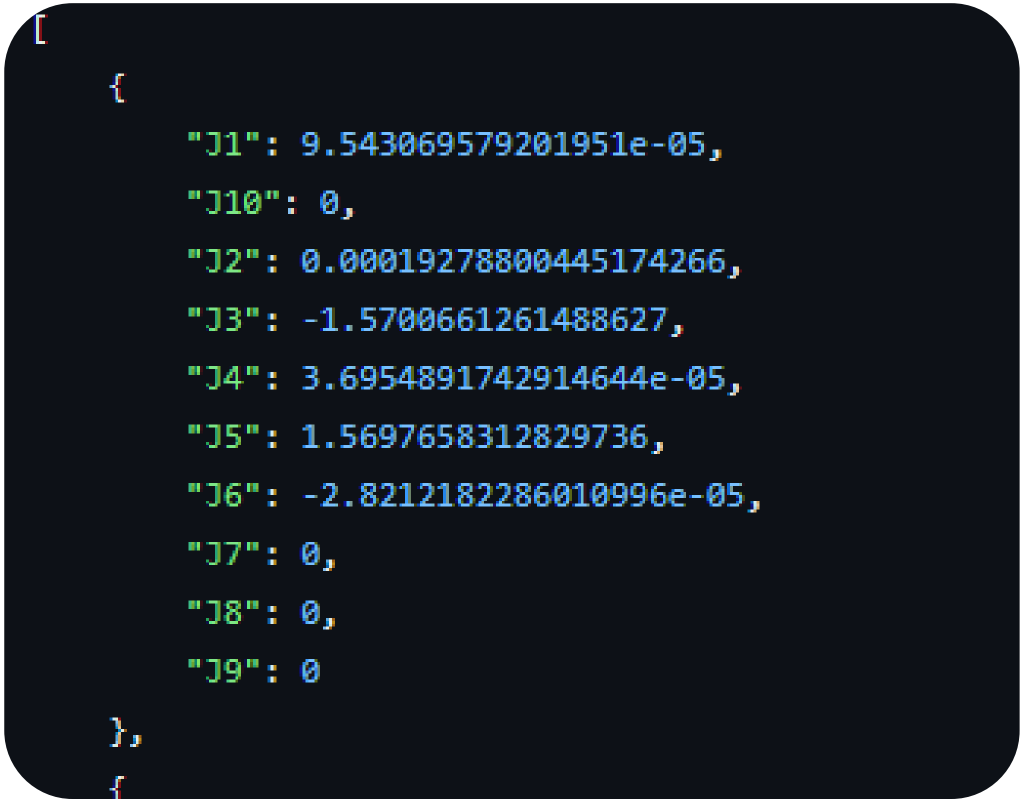

Workpiece 6

Challenge #1 – Elaboration of trajectories



Challenge #1 – Elaboration of trajectories

Content of the input file

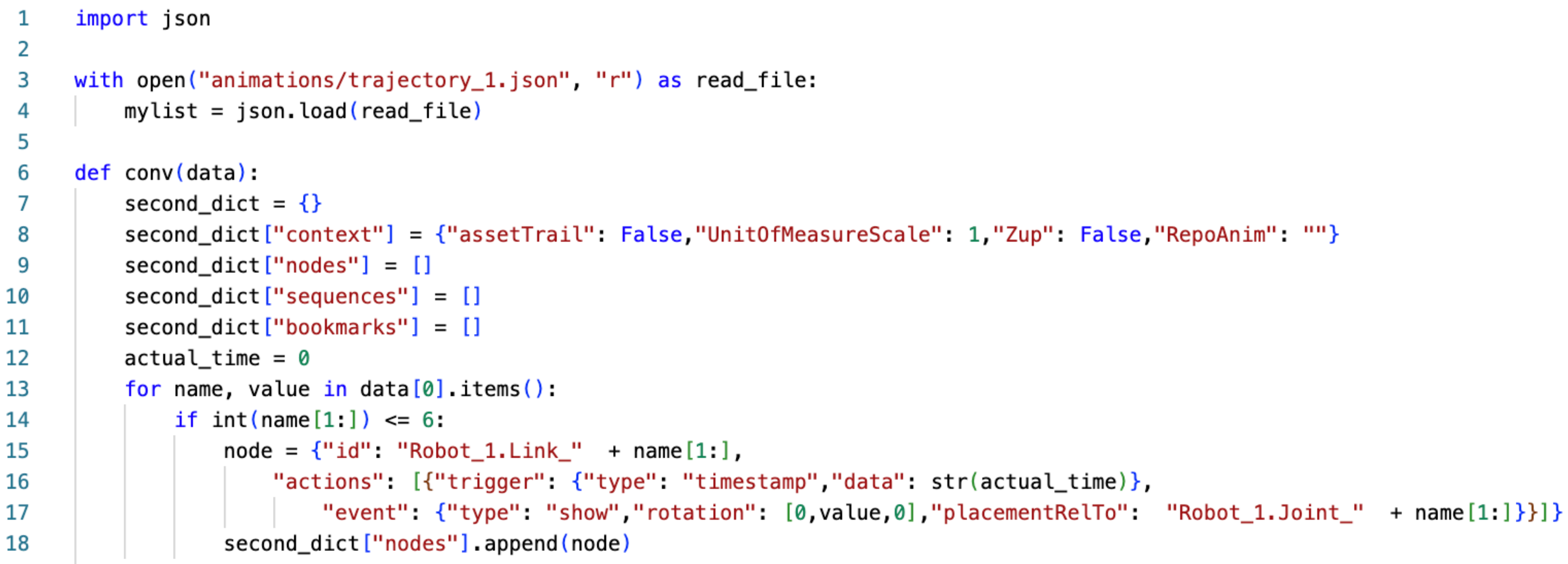


```
[
    {
        "J1": 9.543069579201951e-05,
        "J10": 0,
        "J2": 0.00019278800445174266,
        "J3": -1.5700661261488627,
        "J4": 3.6954891742914644e-05,
        "J5": 1.5697658312829736,
        "J6": -2.8212182286010996e-05,
        "J7": 0,
        "J8": 0,
        "J9": 0
    },
    {
```

DF/docs/AVATAR-JLL/files/trajectory_1.json at main · difactory/DF · GitHub

Challenge #1 – Elaboration of trajectories

Part 1 – Python code



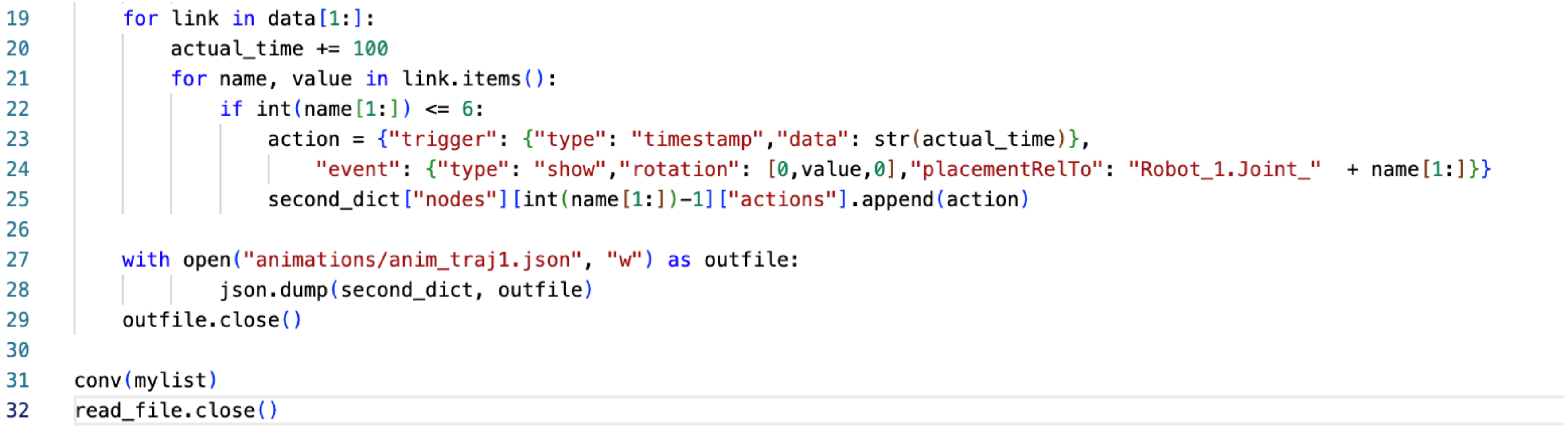
```python
import json

with open("animations/trajectory_1.json", "r") as read_file:
    mylist = json.load(read_file)

def conv(data):
    second_dict = {}
    second_dict["context"] = {"assetTrail": False,"UnitOfMeasureScale": 1,"Zup": False,"RepoAnim": ""}
    second_dict["nodes"] = []
    second_dict["sequences"] = []
    second_dict["bookmarks"] = []
    actual_time = 0
    for name, value in data[0].items():
        if int(name[1:]) <= 6:
            node = {"id": "Robot_1.Link_"  + name[1:],
                "actions": [{"trigger": {"type": "timestamp","data": str(actual_time)},
                    "event": {"type": "show","rotation": [0,value,0],"placementRelTo":  "Robot_1.Joint_"  + name[1:]}}]}
            second_dict["nodes"].append(node)
```

Challenge #1 – Elaboration of trajectories

Part 2 – Python code



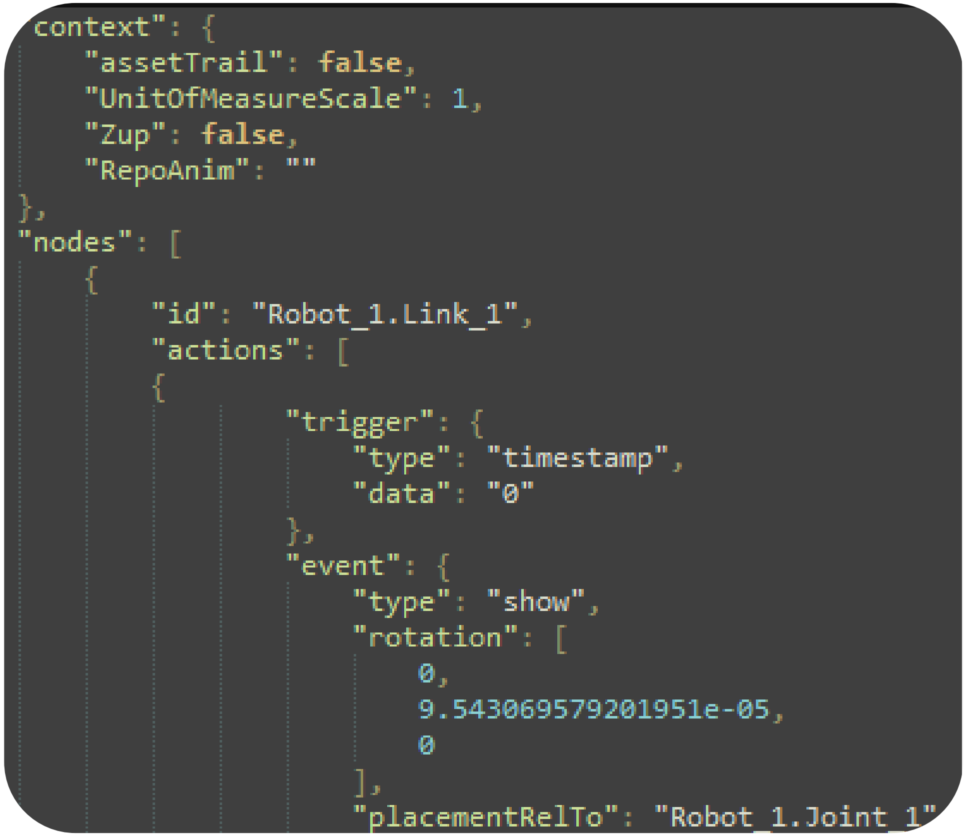
```python
    for link in data[1:]:
        actual_time += 100
        for name, value in link.items():
            if int(name[1:]) <= 6:
                action = {"trigger": {"type": "timestamp","data": str(actual_time)},
                    "event": {"type": "show","rotation": [0,value,0],"placementRelTo": "Robot_1.Joint_"  + name[1:]}}
                second_dict["nodes"][int(name[1:])-1]["actions"].append(action)

    with open("animations/anim_traj1.json", "w") as outfile:
            json.dump(second_dict, outfile)
    outfile.close()

conv(mylist)
read_file.close()
```

Challenge #1 – Elaboration of trajectories

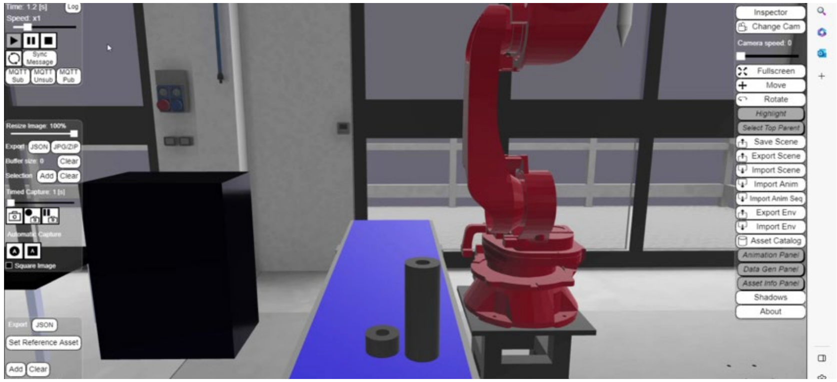
Content of the output file



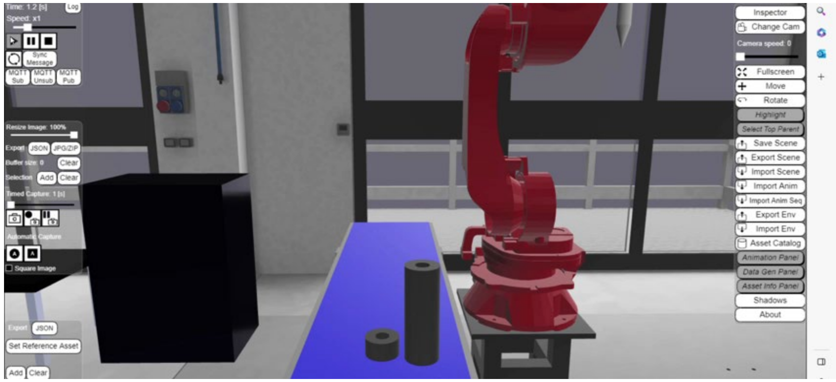
```
context": {
    "assetTrail": false,
    "UnitOfMeasureScale": 1,
    "Zup": false,
    "RepoAnim": ""
},
"nodes": [
    {
        "id": "Robot_1.Link_1",
        "actions": [
        {
                "trigger": {
                    "type": "timestamp",
                    "data": "0"
                },
                "event": {
                    "type": "show",
                    "rotation": [
                        0,
                        9.543069579201951e-05,
                        0
                    ],
                    "placementRelTo": "Robot_1.Joint_1"
```

Challenge #1 – Visualization of trajectories

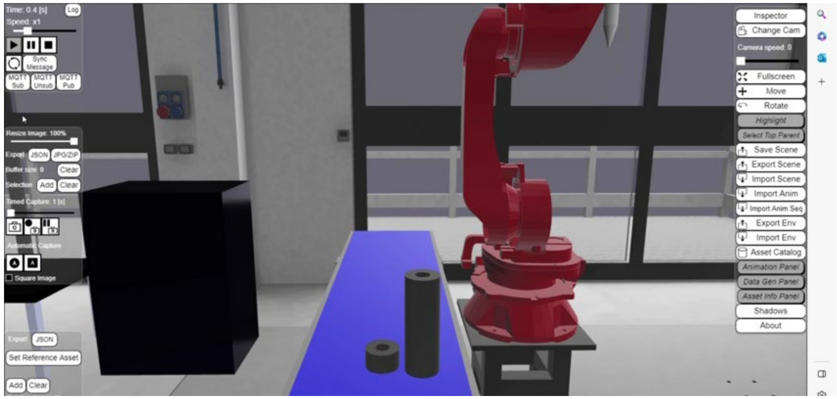
Trajectory 1



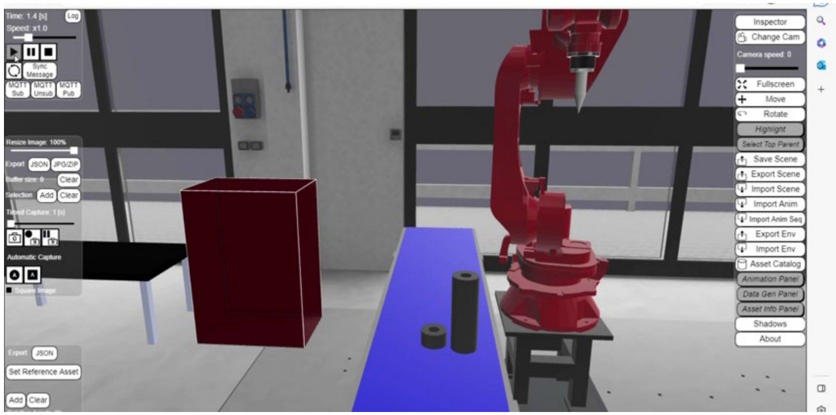
Trajectory 2



Trajectory 3



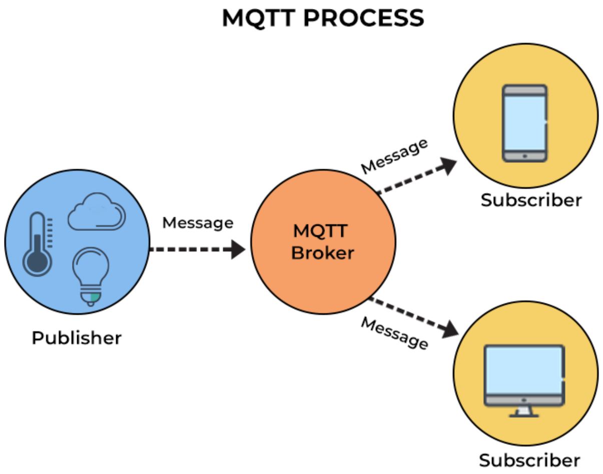
Trajectory 4

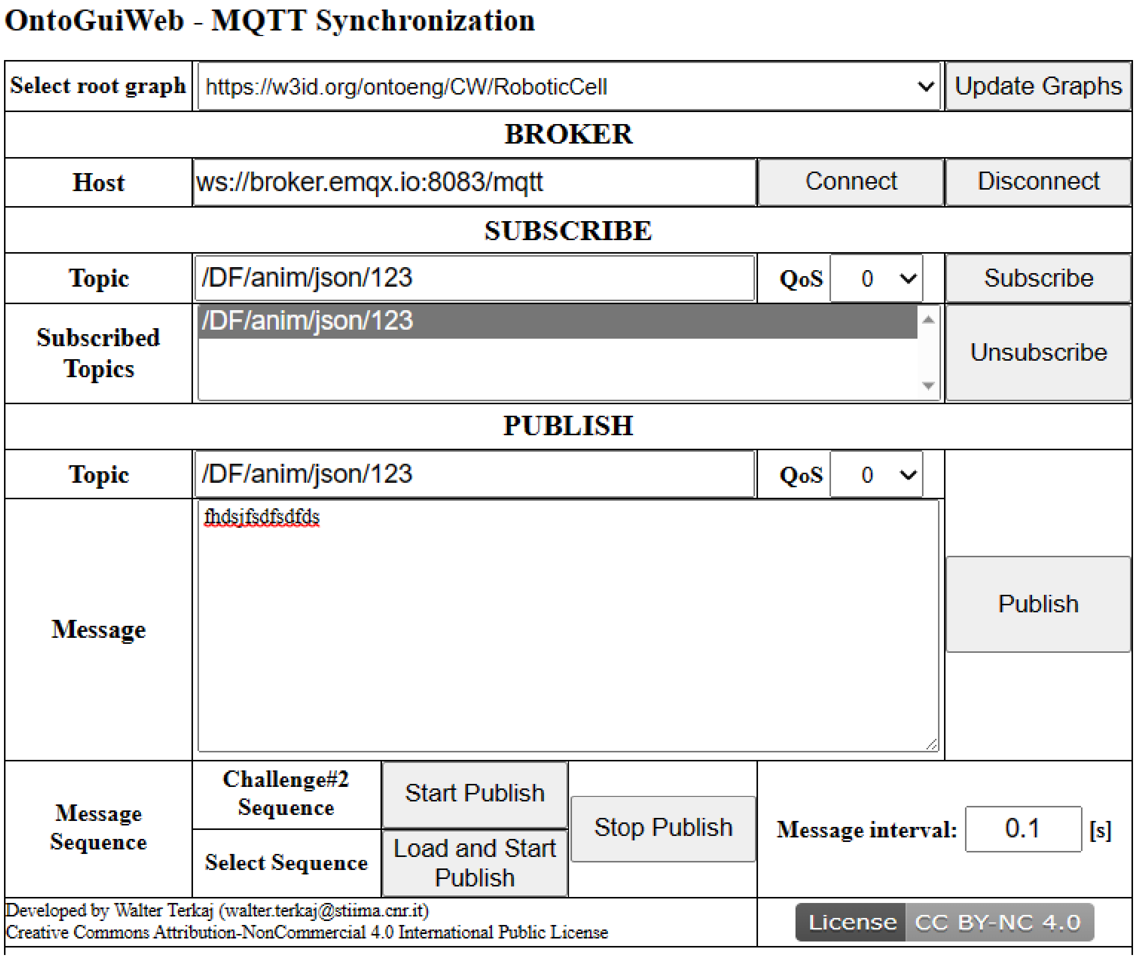


Challenge #1 – Elaboration of trajectories

	Tj.1	Tj.2	Tj.3	Tj.4
Have collision	yes	yes	no	no
Reach the goal	yes	no	yes	yes
Distance from target [m]	0,0368	0,1712	0,0368	0,0368

Challenge #2 – Receive a trajectory via MQTT and check it





OntoGuiWeb - MQTT Synchronization

Select root graph: https://w3id.org/ontoeng/CW/RoboticCell

BROKER

Host: ws://broker.emqx.io:8083/mqtt

SUBSCRIBE

Topic: /DF/anim/json/123 QoS: 0

Subscribed Topics: /DF/anim/json/123

PUBLISH

Topic: /DF/anim/json/123 QoS: 0

Message: fhdsjfsdfsdfds

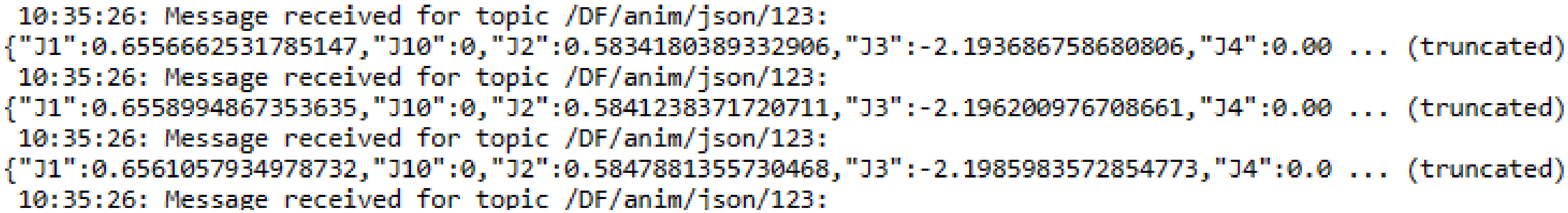
Message Sequence: Challenge#2 Sequence; Select Sequence

Message interval: 0.1 [s]

Developed by Walter Terkaj (walter.terkaj@stiima.cnr.it)
Creative Commons Attribution-NonCommercial 4.0 International Public License

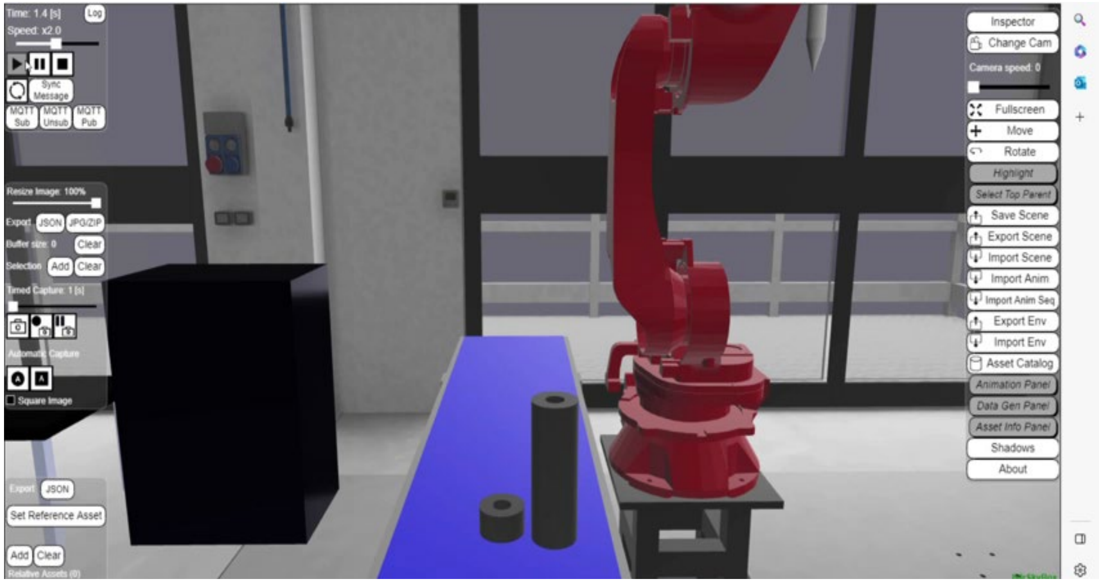
Challenge #2 – Receive a trajectory via MQTT and check it

MQTT message received



```
 10:35:26: Message received for topic /DF/anim/json/123:
{"J1":0.6556662531785147,"J10":0,"J2":0.5834180389332906,"J3":-2.193686758680806,"J4":0.00 ... (truncated)
 10:35:26: Message received for topic /DF/anim/json/123:
{"J1":0.6558994867353635,"J10":0,"J2":0.5841238371720711,"J3":-2.196200976708661,"J4":0.00 ... (truncated)
 10:35:26: Message received for topic /DF/anim/json/123:
{"J1":0.6561057934978732,"J10":0,"J2":0.5847881355730468,"J3":-2.1985983572854773,"J4":0.0 ... (truncated)
 10:35:26: Message received for topic /DF/anim/json/123:
```

Challenge #2 – Receive a trajectory via MQTT and check it

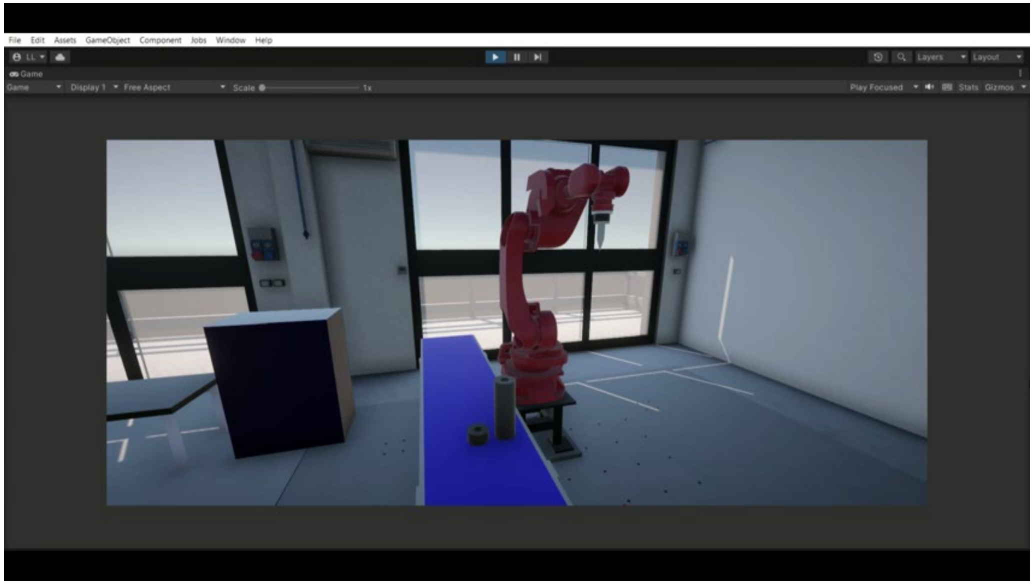


	Trajectory 5
Collision	No
Goal	No
Distance [m]	0,0799

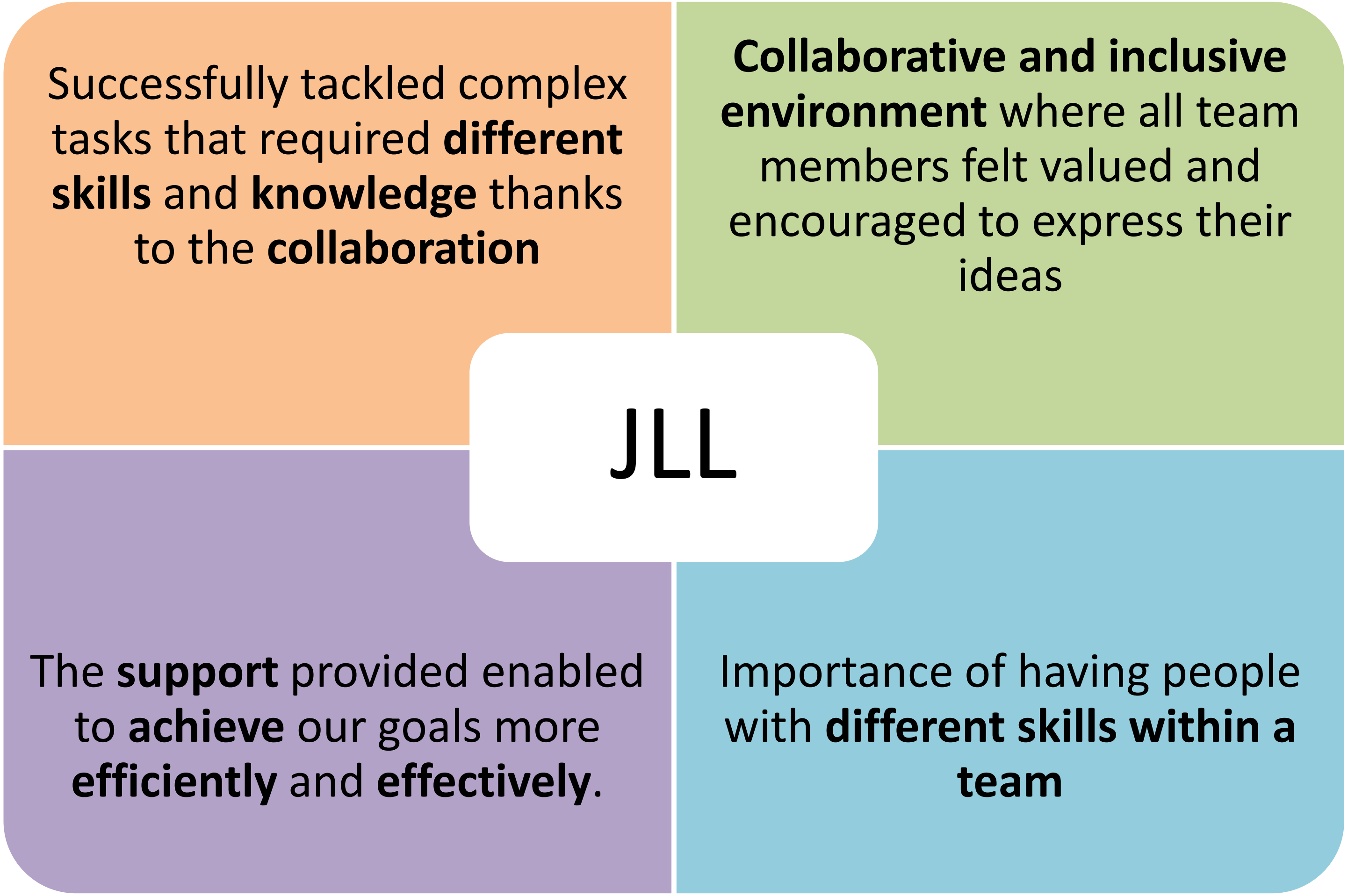
Challenge #3 – Create a trajectory

- Set starting position of the joints in Unity
- Set manually the final positions for each joint, so the robot is touching the workpiece
- Add a middle position to avoid collision
- Fill the animator and play
- Export the file with the .anim extension

	New trajectory
Goal	Yes
Collision	No
Distance	0,0455



JLL - outcome



JLL

Successfully tackled complex tasks that required **different skills** and **knowledge** thanks to the **collaboration**

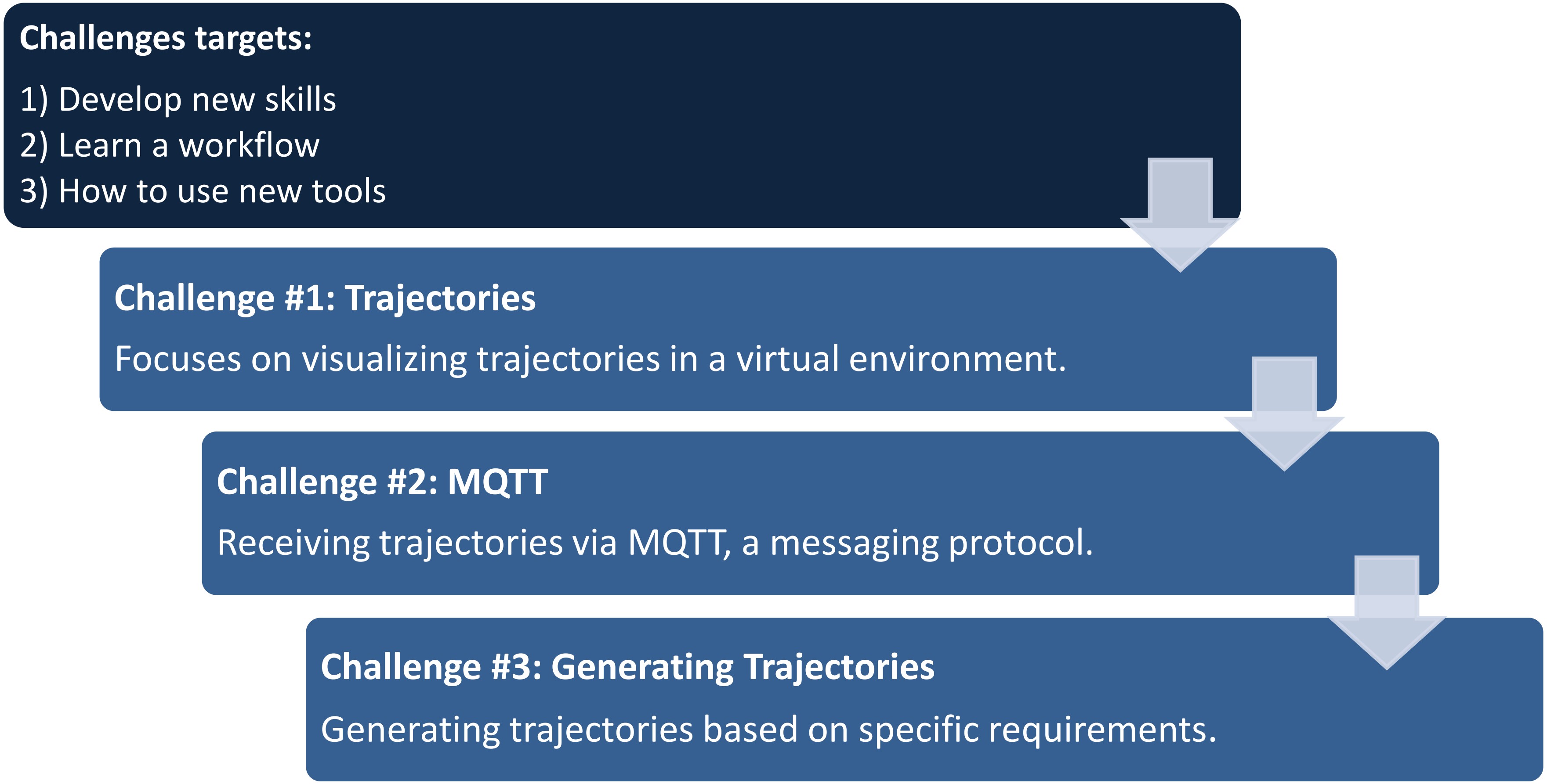
Collaborative and inclusive environment where all team members felt valued and encouraged to express their ideas

The **support** provided enabled to **achieve** our goals more **efficiently** and **effectively**.

Importance of having people with **different skills within a team**

Updating the Guide – part 3

Target of Challenges - Enhancing Skills and Understanding



Challenges targets:

1) Develop new skills
2) Learn a workflow
3) How to use new tools

Challenge #1: Trajectories

Focuses on visualizing trajectories in a virtual environment.

Challenge #2: MQTT

Receiving trajectories via MQTT, a messaging protocol.

Challenge #3: Generating Trajectories

Generating trajectories based on specific requirements.

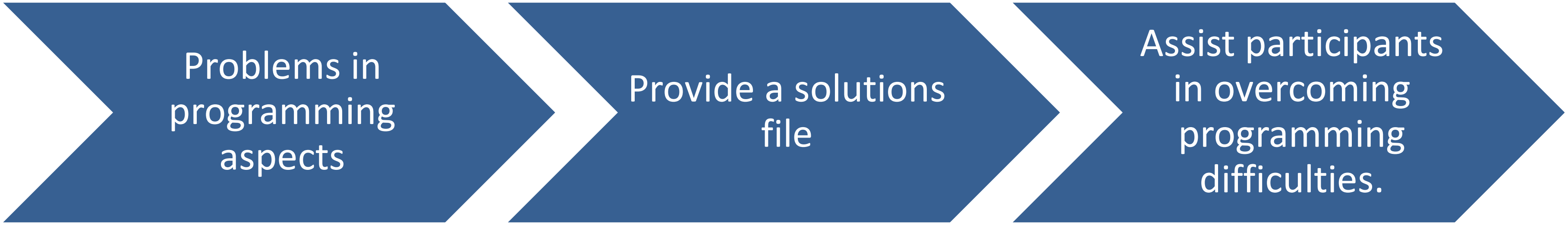
Feedback and Insights

1 Obtain valuable insights during the JLL about the difficulties participants faced.

2 Discussions within the groups to gather feedback on the challenges

Difficulties and Benefit of Solutions File

According to the feedbacks

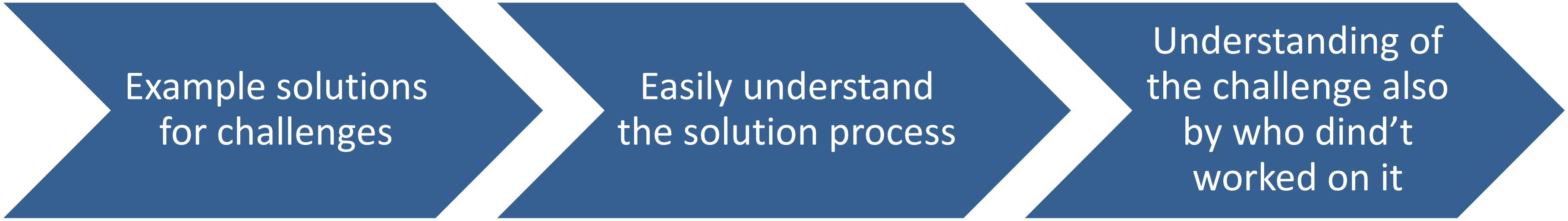


Problems in programming aspects

Provide a solutions file

Assist participants in overcoming programming difficulties.

Benefits of the provided solution file



Example solutions for challenges

Easily understand the solution process

Understanding of the challenge also by who dind’t worked on it

Updating Guide & Challenges Document

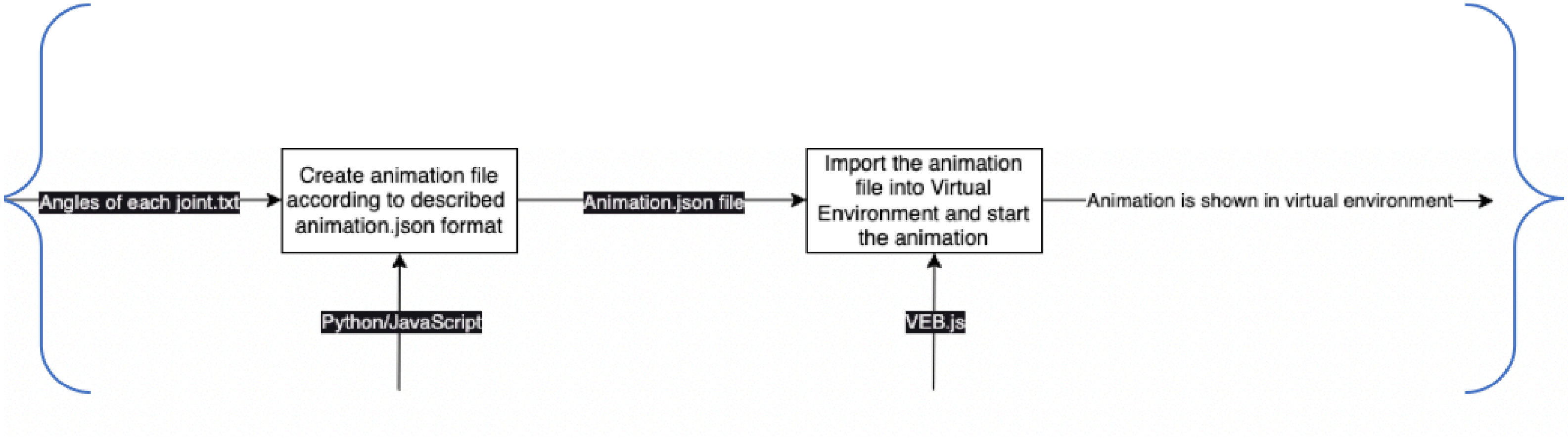
Add workflow diagram to understand the overall process

Incorporate additional clarifications in areas where we have identified potential ambiguities.

Broken links, corrected typos, and made other necessary updates to overall coherence and smoothness of the documentation.

Guide Workflows Added to Challenges

Challenge #1



Angles of each joint.txt

Create animation file according to described animation.json format

Python/JavaScript

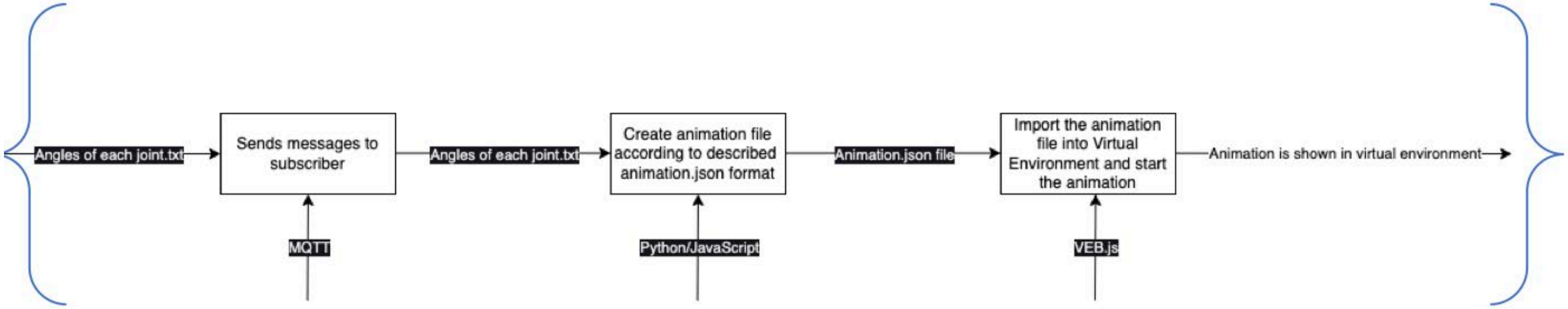
Animation.json file

Import the animation file into Virtual Environment and start the animation

VEB.js

Animation is shown in virtual environment

Challenge #2



Angles of each joint.txt

Sends messages to subscriber

MQTT

Angles of each joint.txt

Create animation file according to described animation.json format

Python/JavaScript

Animation.json file

Import the animation file into Virtual Environment and start the animation

VEB.js

Animation is shown in virtual environment

Thanks for your attention

POLITECNICO
MILANO 1863