Ref. Ares(2023)327378 - 16/01/2023

This project is funded by the European Union
G.A No 101016906

Project Number: 101016906
Start Date of Project: 2021/01/01
Duration: 48 months

Type of document D3.4 – V1.1

Simulated Farming Environment and Basic Robotic Components

Dissemination level	PUBLIC
Submission Date	2021-08-31
Resubmission Date	2023-01-16
Work Package	WP3
Task	T3:7
Type	Report
Version	1.1
Author	PaleBlue
Approved by	Consortium

DISCLAIMER:
The sole responsibility for the content of this deliverable rests with the authors. It does not necessarily reflect the opinion of the European Union. Neither the Directorate-General for Communications Networks, Content and Technology, Artificial Intelligence and Digital Industry nor the European Commission are responsible for any use that may be made of the information contained therein.

This project is funded by the European Union G.A No 101016906

History of changes

Changes in original text	New version	Old Version
Paragraph 4 BEM, p. 18-19	Details on BEM mechanism based on the "Request for Revision Comment".	N.A.

This project is funded by the European Union G.A No 101016906

Executive Summary

The simulation environment will be set up to match the properties of the real testing environment as closely as possible. This will enable the highest chance for realistic testing. The first version of the simulation environment will be relatively simple but will still have the needed realism.

The main focus for this phase is the environment with the ground, support structure, grapevines with grapes and the light conditions. Next to that we will provide a ROS-controlled mobile base which can drive around the plot in a similar way as it would in the real testing ground.

Table of Content

1 Introduction 6
1.1 Purpose of this document 6
1.2 Document overview 6
2 Simulation environment 7
2.1 The simulation engine 7
2.2 Ground and environment 7
2.2.1 Ground 7
2.2.2 Environment 8
2.3 Support structure 8
2.4 Vegetation 10
2.4.1 Grapevines 10
2.4.2 Grapes 11
2.5 Environmental conditions 12
2.5.1 Light 12
2.5.2 Weather 14
3 Mobile base platform 15
3.1 Model 15
3.2 Dynamics 15
3.2.1 Collisions 16
3.2.2 Tracks 16
3.3 ROS Control 16
3.3.1 Actuator drivers 16
4 Box Exchange Mechanism 18
4.1 Boxes with grapes 19
5 VR Environment 20

Abbreviations and Acronyms

BEM	Box Exchange Mechanism
ROS	Robot Operating System (see https://www.ros.org/)
URDF	Unified Robot Description Format (see http://wiki.ros.org/urdf)

This project is funded by the European Union G.A No 101016906

1 Introduction

Training in simulation and then transfer of the acquired models from the simulation to the real environment is a preferred alternative to training models directly on a real robotic platform because of many reasons, including the wear and tear of the physical hardware, the difficulties of running experiments on the field, duration of design iteration, and safety of the virtual environments.

An effective VR farming environment capable of resembling the physical world as much as possible is being developed to support the development of the robotic platforms and algorithms in CANOPIES.

1.1 Purpose of this document

This document describes the requirements for the simulation environment and how these requirements will be fulfilled in the first version of the simulation environment.

For subsequent work packages we expect that additions and changes will be included the simulation environment to support the functionalities developed in those work packages.

1.2 Document overview

The document is split in three main sections:

1. The simulation environment:
   Including the ground, the support structure in the vineyard, the grape plants and the weather conditions;
2. The mobile base:
   Without the dual arm or box exchange mechanism;
3. The box exchange mechanism:
   For exchanging boxes with grapes between two mobile bases.

Within the scope of this delivery, there are no human users in the simulation. Human user interaction will be part of deliverable D6.4: VR Human-Robot-Interaction Component.

2 Simulation environment

This chapter describes the simulation environment in which the virtual representations of the robots and users will work together. It will describe the following parts:

1. The simulation engine;
2. The ground and environment;
3. The support structure;
4. The vegetation:
    a. The grapevines;
    b. The grapes.
5. Environmental conditions:
    a. Light;
    b. Weather.

2.1 The simulation engine

The simulation will use the Unity 3D engine (see https://unity.com/). The engine brings all models and users/robots together in a virtual 3D environment. It is a highly optimized 3D engine which is used to create games, but it also extensively used for professional 3D solutions.

The engine includes the visual rendering of the models, the physics interaction between objects, animations and networking between multiple clients.

The Unity engine was chosen because it has most functionality required for this project which includes:

- Virtual reality support for a wide range of headsets
- Multiple networking options
- Multiple physics engine options
- ROS support

The options in VR, networking and physics solutions provides us with the flexibility needed to cope with the developing requirements for this project.

2.2 Ground and environment

2.2.1 Ground

The terrain in the simulated environment should enable the following:

1. Provide a non-flat surface which resembles the actual ground;
2. Have a similar look as the real ground;
3. Enable the slipping of the mobile base tracks in case it cannot move.

These requirements can be fulfilled with the use of a Terrain component which is part of the standard components in the Unity Engine we use.

The terrain is a deformable polygon mesh which can be modelled in the same way as the real ground surface.

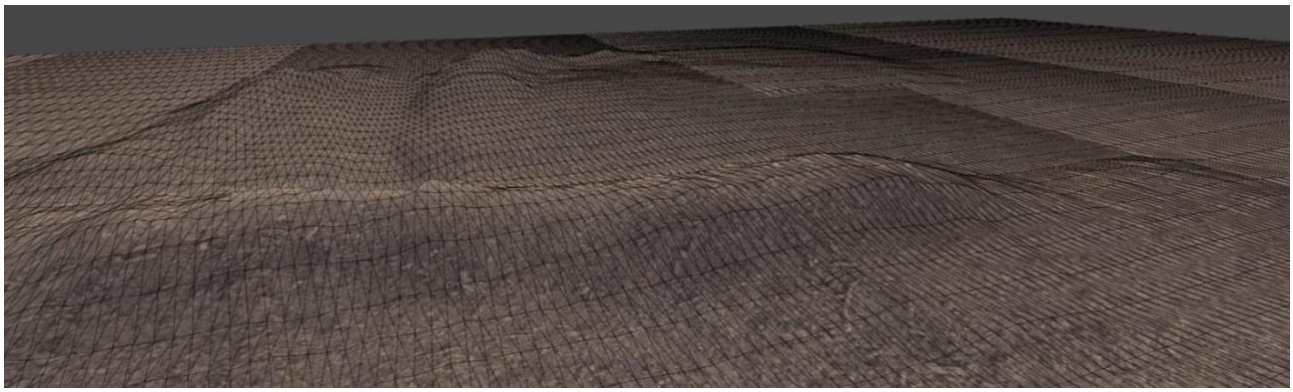


Figure 1 - Terrain model

The look is determined by using one or multiple ground images (Textures) to match the looks of the actual ground.

The interaction of the tracks of the mobile base with the ground is modelled by a Physic Material which defines the static and dynamic friction of the ground. These friction values together with the physics properties of the mobile base tracks (see section Dynamics) determine the slipperiness of the ground and how much of the torque from the mobile base motors will actually be transformed in the motion of the base platform.

Optionally, loose objects like rubble and rocks can be added to the ground to further mimic the real conditions of the environment.

If necessary small plants and grass can be placed at realistic places around the plot.

2.2.2 Environment

The environment of the simulated testing ground will be surrounded by visible or invisible walls to prevent the testing subjects from leaving the test ground.

Visible walls will act like real walls: they will be visible for optical sensors and the mobile base or humans will not be able to go through them.

Invisible walls will not be visible for the optical sensors, but the mobile bases or humans will not be able to go through.

2.3 Support structure

The support structure is a permanent structure to support the growth of the vines. It consists of the following components:

1. Vertical and diagonal concrete poles to guide the vine trunks and support for the next items;
2. Iron wires to support the vine's branches, the irrigation system and the cover structure;
3. Tunnel shaped plastic cover supported by the arches;
4. Irrigation system.

All these components will be modelled as a static structure. This means that it is a permanent, unmovable structure. It will stop the movements of any object colliding with this structure without

movement. This includes collisions with the mobile base, the dual arm, humans, grape vines and/or grape clusters.

The support structure will be designed according to the Flat-roof Pergola system depicted in Figure 1 in [D2.1] with the difference that the plastic cover is folded out.

Depending on conditions variants of the support structure can be created. An example of this may be during the pruning when the plastic cover is removed.

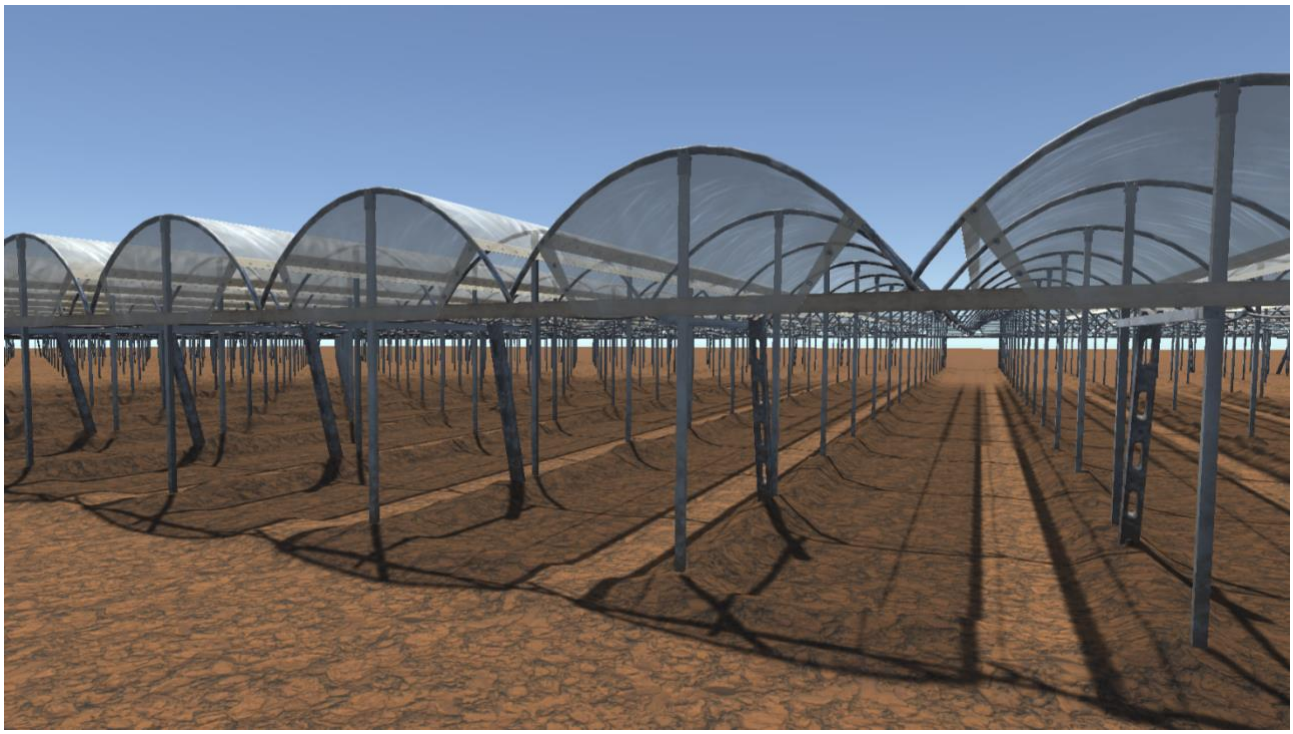


Figure 3 - Environment model

2.4 Vegetation

The vegetation consists of the grapevines and the attached grape clusters. We distinguish between them because of the specific requirements set for the clusters.

2.4.1 Grapevines

The plants will be modelled based on the specifications given in Section 2.3 - Pruning in [D2.1].

Figure 4 - Vine plant model

Each plant will have the following additional characteristics during the harvesting period:

1. 2-3 grape bunches per cane;
2. 2-3 canes per branch;
3. 4-6 branches per plant.

This results in 16 - 54 grape clusters per plant which is on average 32 clusters per plant.

Each cane will not be shortened and can therefore reach the length of about 3 meters.

2.4.2 Grapes

The grapes will be similar to the Pizzutello Nero grape variety.

Figure 5 - Simulated grape bunch

For the first version of the simulated farming environment two stages of ripeness will be included: one with the correct colour for ripeness and one with a (partially) green colour indicating that the cluster should not be harvested yet. We will achieve this by using two colour textures for a grape cluster model.

In later versions more colour textures may be needed to accommodate for the looks of the grapes for near infra-red sensor and other multi- or hyper-spectral sensors.

The grape clusters will be a single model which means that no movements between the grapes within a cluster will be possible. The peduncle connecting the grape cluster to the cane can be cut to remove the grape cluster from the cane.

If it is necessary to improve the dynamics for research activities, it is possible to model movements between parts of the bunch. This is not included in the first version because it adds complexity and increases the required computing power for the simulation. We try to keeps things simple where we can and add move advanced models where it is needed.

After severing the cluster from the plant, the grape cluster will be an independent object which can collide with objects in its environment. It's physical properties like mass will be modelled according to those of real grape bunches such that its behaviour while handling will be similar.

2.5 Environmental conditions

2.5.1 Light

The light conditions in the simulation environment will initially include the following two scenarios:

1. Bright morning day (around 7:00 AM);
2. Bright mid-day (around 12:00 PM).

For those scenarios we will use the sun's position in the testing ground in Aprilia, Italy:

Date:	02/07/2021 | GMT1	
coordinates:	41.5942016, 12.6485765	
location:	41.59420160,12.64857650	
hour	Elevation	Azimuth
05:39:35	-0.833°	57.59°
6:00:00	2.44°	60.96°
7:00:00	12.65°	70.33°
8:00:00	23.47°	79.37°
9:00:00	34.61°	88.71°
10:00:00	45.78°	99.37°
11:00:00	56.54°	113.29°
12:00:00	65.88°	134.73°
13:00:00	71.19°	170.29°
14:00:00	69.02°	211.16°
15:00:00	61.03°	238.34°
16:00:00	50.72°	254.91°
17:00:00	39.66°	266.7°
18:00:00	28.46°	276.47°
19:00:00	17.46°	285.55°
20:00:00	6.93°	294.7°
20:47:17	-0.833°	302.33°

Figure 6 - Sun positions in Aprilia, Italy

(source: https://www.sunearthtools.com/index.php)

The actual measured lux values at the testing ground are as follows:

Time of day	Open air	Under canopy
(Earlier times to be added)	(Values need to be corrected)	
11AM	40,000 lx	1,400 lx
12PM	65,000 lx	1,500 lx

Table 1 - Measured lux values at the testing ground

From the measurements it is visible that the amount of light underneath the canopy is around 3% compared to the open-air light intensity.

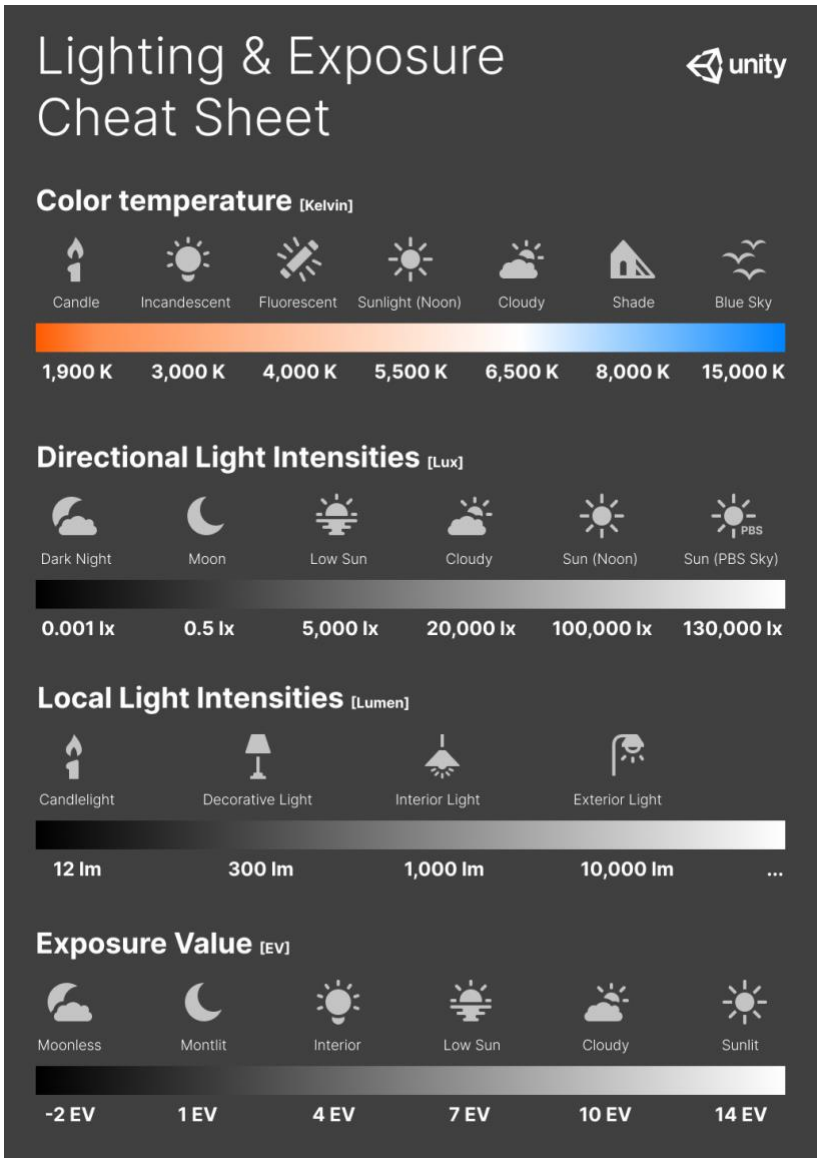


Figure 7 - Lighting values

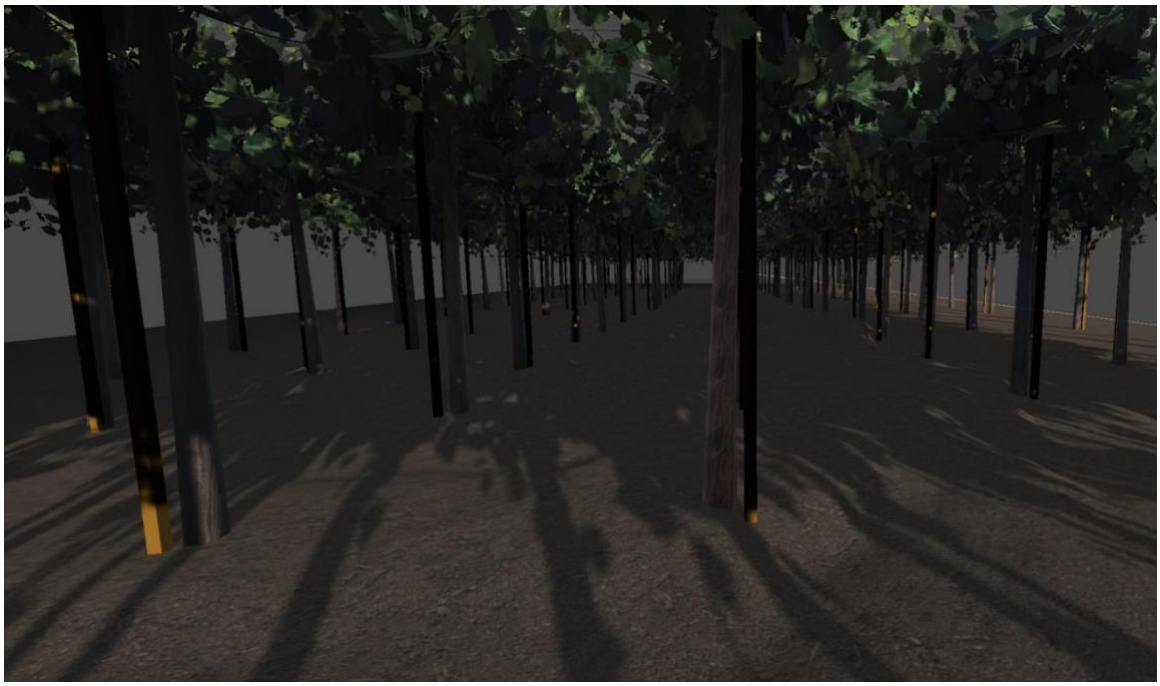


Figure 8 - Example lighting setup 7AM

Figure 9 - Example lighting setup 12PM (mid-day)

2.5.2 Weather

For the first version we plan no specific weather conditions: it will resemble an unclouded day with no wind and no precipitation.

3 Mobile base platform

3.1 Model

The mobile base platform is based on the Alitrak DCT-350P selected by the consortium.

With this model we need to fulfil the following:

1. Drive the platform across the environment, including physics interactions like ground bumps and collisions with objects;
2. Be able to support the dual arm and BEM;
3. Have the ability to support the necessary sensors;
4. Flexibility in design and interaction to follow the development of the real platform.

The model we use in the simulation environment is derived from the FBX model we received for this platform. The alternative would be to use URDF but this was not chosen because:

A. URDF is not available until the real mobile platform development is matured;
B. It is not compatible with the wheel simulation we need to fulfil goal 1 above.

Figure 10 - Simulated mobile base platform

3.2 Dynamics

The physical interaction of the mobile base with its environment can be divided into the following categories:

1. Collisions of the base with the environment (like poles and humans);
2. The interaction of the tracks with the ground.

3.2.1 Collisions

The mobile base should not be able to drive through fixed structures like the support poles in the vineyard. Furthermore, it would be able to push things around which are movable like small rocks and grape collection boxes on the ground.

This is all taken care of by the physics interaction for objects with wheels in the Unity engine.

3.2.2 Tracks

The tracks of the mobile base are assumed to be not deformable. In n the datasheets about the Alitrak DCT-350 it is documented that the tracks do not have suspension. Therefore, the shape of the tracks in the simulation is not influenced by ground or objects on the ground when it drives across.

The track interaction is in fact simulated by a series of synchronized wheels on both sides. A real track simulation is not readily available and our experiments showed that the series of wheels lead to a similar behaviour. The following interactions are possible:

- Straight forward/backward driving when both tracks are driven in the same direction;
- Turning on the spot when both tracks are driven in opposite directions;
- Drive left/right when one track is driven and the other stands still;
- Slipping tracks when the base collides with static structures of heavy objects.

The interaction between the ground and the track is determined by the properties of the wheel and the ground. The ground properties are described in the Ground section. For the track wheels the forward and sideward slip can be configured to match the correct behaviour of the track's behaviour.

The behaviour of the mobile base with the tracks is matched to the recordings of the real base. Recorded data (rosbag) of the real base will be compared to the resulting movements of the simulated base in a similar environment to check if the behaviour is correct.

3.3 ROS Control

The mobile base is autonomous, controlled by ROS. Section 2.1.6.1 of [D2.2] describes the relevant ROS packages used to interface with the (simulated) mobile base. Support for the sensors described there is part of work package 4. Within this work package we will discuss the mobile base actuators.

3.3.1 Actuator drivers

It is possible to control the mobile base through sending messages to a ROS control topic. The current implementation takes two torque values for each of the sides of the differential drive of the mobile base. These values are used to control the tracks on each side of the mobile base. The physics model of the tracks and the environment will then result in the movement of the mobile base.

The ROS control interface is still in development and may change in the future. One possible option is a kinematic control where the messages specify the linear and angular velocity of the mobile base. This is then be translated to the motor torques in order to drive the physics model. The ROS messages for controlling the platform will be formatted as geometry_msgs/Twist ROS message type.

The motor controllers receive input through CAN bus. Since ROS is used as the control interface, a ROS node will be developed to convert ROS messages into CAN bus messages. It is possible to give

the motor controllers an input, which ranges from -100% to 100%, where negative numbers are reverse control signals and positive numbers are forward control signals. The raw data input range is -32768 - 32767. Each motor controller has its own motor controller ID and has a separate input/output.

The platform and motor controllers have the following properties:

Motor rpm	rpm	2000
Reduction gear ratio	R	1:32
Forward speed	km/h	3,2
Reduced forward speed	km/h	2,6
Backward speed	km/h	3,2
Reduced backward speed	km/h	2,6

Table 2 – Motor controller properties.

The motor controllers are providing feedback with information regarding: motor position (0 - 4096, corresponding to 0-360 degrees), motor rpm (-2000 - 2000), voltage and fault codes. The motor rpm is the main interface between the simulation and the ROS control software.

4 Box Exchange Mechanism

As far as the box exchange mechanism (BEM) is concerned, it should be noticed that details concerning its simulation are described within the deliverable D7.3 VR Multi-Robot Component, being this component related to the research activities of the work package WP7 Multi-Robot Coordination.

Briefly speaking, the main approach in modelling and simulating the BEM is based on the consideration that the number of moving parts in the scenario for passing a box from a farming robot to a logistic one is limited. In this regard, dynamic physics objects is used for modelling the box (when it is full with grapes) that is transferred from farming to logistics robot.

BEM contains an internal mechanical structure for sorting and serving the boxes. This structure performs a number of motions internally. For the sake of simulation, we represent these with animated motions rather than physically simulated ones, as modelling physics in these BEM processes does not increase the value of the overall BEM simulation.

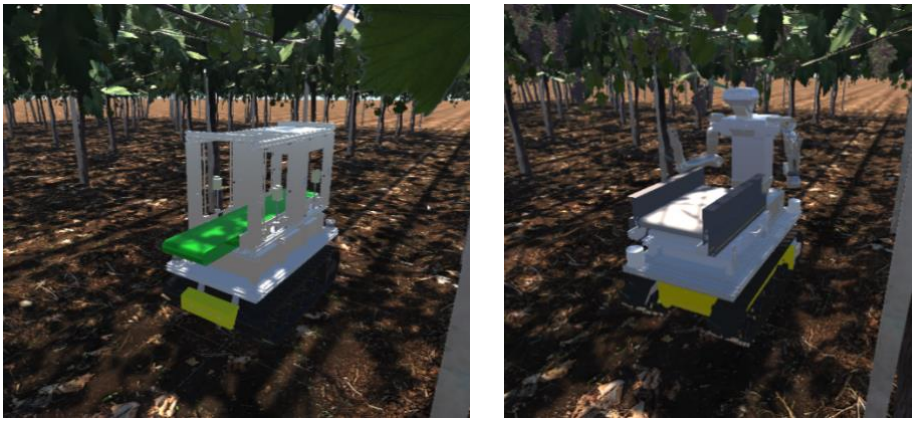


Figure 11 – BEM on the logistic and farming robots

The BEM architecture proposed by DTI is integrated into the simulation environment for both robots, with the respective additions to the farming robot (Figure 11, on the right) and the logistics robot (Figure 11, on the left).

Briefly, the core BEM mechanism for the exchange of a box is implemented as following. Boxes are represented as physical rigid bodies when placed onto the belt. As a matter of fact, the two robots have specialized 3-dimensional areas in space (volumetric areas) with applied forces, that causes the box to be passed for the belt of one robot to the other. In detail, when the mechanism is activated, a directed physical force is applied to a designated volumetric area directly above the belt, therefore pushing all the rigid body in the given direction. The transport belt of the receiving robot is performing likewise, and it also pushes rigid bodies that appear in the volumetric area into the receive area.

The following addresses modelling of the boxes with grapes.

4.1 Boxes with grapes

The boxes with grapes are modelled to represent the geometry of the real boxes: a concave box which can hold a certain number of grape bunches. The centre of gravity of the box with grape bunches depends on the positions of the bunches inside the box and may even change when the position of grape bunches in the box changes during handling. This leads to realistic dynamics of the box while going through the box exchange procedure.

In this regard, it should be noticed that since the grape bunches have a limited flexibility, it may be hard to arrange the grape bunches in a realistic way. As is described in the section 2.4.2 Grapes section, we plan to make the grape bunches rigid with no internal movements at first. A more dynamic grape brunch model could be considered should this be deemed necessary at a later stage of the project. An alternative would be to artificially arrange the grape bunches in the box: they get a pre-set position in the box when they are put in the box. This latter option then can lead to a (too) neatly arranged box.

Note that, on the other hand simulating a box without grapes presents a very straightforward case for physics simulation. We therefore do not foresee challenges with such setup since there are no internal movements for the rigid box without grapes.

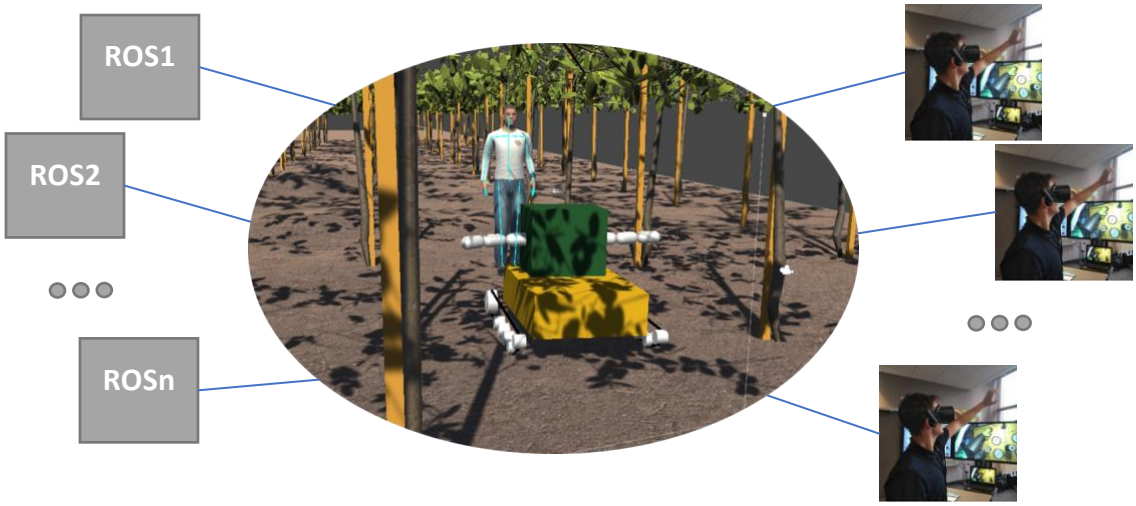
5 VR Environment

The virtual reality solution is based on the PaleBlue VR Platform. It consists of a network of synchronized clients. Each client represents a human user or a simulated robot (mobile base with or without a dual arm) or in the same virtual environment.

The virtual environments of the clients are synchronized between clients in real-time. In that way each client can observe and response to changes in the virtual environment performed by the other clients.

Human users either use a mobile solution (for instance an Oculus Quest) or a PC-based solution (like an HTC Vive + PC). In the case of the mobile solution the network client is running on the so no additional hardware is necessary. The downside of the mobile solution is the lower performance level it can achieve which may be not enough for this application. The decision between the two approaches has not been made yet as the information about the required performance is not complete yet.

Each client for the robot runs on a PC. We aim to use a Linux based PC running Ubuntu 20.04 and



ROS as this provides the best integration with the rest of the ROS packages. The Unity engine we use for the VR environment runs on Linux but some features are not as well supported as on a Windows based environment. So it may be that the Linux platform can be swapped for a Windows environment in the future. These clients use ROS topics to communicate with the controlling ROS packages for controlling the various robot parts and providing necessary sensor information.

References

Id	Title	Version	Date
[D2.1]	Requirements, Specifications and Benchmarks	1.0	2021-04-30
[D2.2]	Specifications and KPIs for the two CANOPIES robot prototypes	1.0	2021-06-30

Table 2 - References.