



xV : x ways to visualize things.

Unity

*Summary: To build an application with Unity allowing its users to create and animate a virtual industrial environment.*

# Contents

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Introduction</b>	<b>3</b>
<b>III</b>	<b>Objectives</b>	<b>4</b>
<b>IV</b>	<b>General Instructions</b>	<b>5</b>
<b>V</b>	<b>Mandatory part</b>	<b>6</b>
V.1	Presentation . . . . .	6
V.2	Specifications . . . . .	6
V.3	images from the POC . . . . .	8
<b>VI</b>	<b>Partie bonus</b>	<b>10</b>
<b>VII</b>	<b>Submission and peer-evaluation</b>	<b>11</b>

# Chapter I

## Préambule



This topic is brought to you by Daher. Daher is an equipment manufacturer that develops integrated industrial systems for aeronautics and advanced technologies.

Turning to innovation since its creation in 1863, Daher is today one of the major players in the 3rd industrial revolution.

Its fields of activity are numerous: aircraft construction, nuclear services, systems and aerospace, integrated logistics ...

# Chapter II

## Introduction

Have you ever played sims? This game a little dull, without challenge where we observe characters doing everyday actions. An adaptation of this game for an industrial environment would be a real game changer for us. With an industrial building instead of a house. Replace the food in the fridge with rubber parts. Being able to take these pieces, not being able to eat them, but to be able to put them on a table or in a machine. Have objects and men interact. That's what we want.

# Chapter III

## Objectives

We need to show what we do in our factories and warehouses.

The software in the industry today is far too expensive, not functional (you do not see a man making a movement close to reality) and have decors worthy of the worst games of PlayStation 1.

After studying the best tool for our project, we chose real-time 3D technology to create an application that allows us to represent our infrastructure and then integrate some human-machine processes. A proof of concept has already been produced.

It is to have a more robust tool that we turn to you, the coding specialists.

# Chapter IV

## General Instructions

- This is a Unity project it has to be made entirely, exclusively and mandatorily in C#. No Javascript/Unityscript ,Boo or any other horrors.
- In case you are wondering, there is no norm imposed at 42 for C#. You can use the style you like without restriction. But remember that a code that your corrector cannot read is a code that he or she cannot grade.
- You must sort your project's assets in appropriate folders. For every folder correspond one and only one type of asset. For example: "Scripts/","Scenes/","Sprites/","Prefabs" . . .
- The use of the Unity Asset Store is forbidden. You are encouraged to use assets on the Internet, exception made of scripts obviously because you have to create everything you submit. However the use of Unity Standard Assets is authorised.
- This project will be corrected by humans only.

# Chapter V

## Mandatory part

### V.1 Presentation

The perimeter of the application in its use includes:

- The creation of an environment.
- Its settings.
- Actions and animations we want to see.
- A system that allows you to better visualize the whole thing.

In our proof of concept, we do things in 3 steps. On one hand we create the environment, on the other hand, we do the animations. Finally, we launch a replay of the actions, which generates a short movie on the screen in which one can move.

### V.2 Specifications

Thus, the application must have several features:

1. Building an environment from 3D models:
  - Already integrated into the application
  - Import into the application and back it up
2. Being able to modify this environment:
  - Change the name
  - Change the position
  - Change the rotation
  - Destroy the objet
  - Change the color

3. Being able to animate this environnement:

- Allow interaction between two objects
- Allow to run object movements (animation, translation, rotation)
- Modify a sequence
- Display text explaining the current action (the user must be able to change it).
- Being able to parameterize actions and animations (change of speed)
- Create action sequences: a man takes a piece and places it somewhere. The application must be able to generate the task in a loop.

4. Be able to save all content:

- Being able to come back, after exiting the app, exactly where we were.
- Load the backup of another environment.

5. Have multiple view modes:

- Overview
- First person view
- Move freely on the ground for the first person view and in the space for the third person.

6. Have a replay system:

- Allow to replay animated sequences.

7. Export a video of the animated sequences:

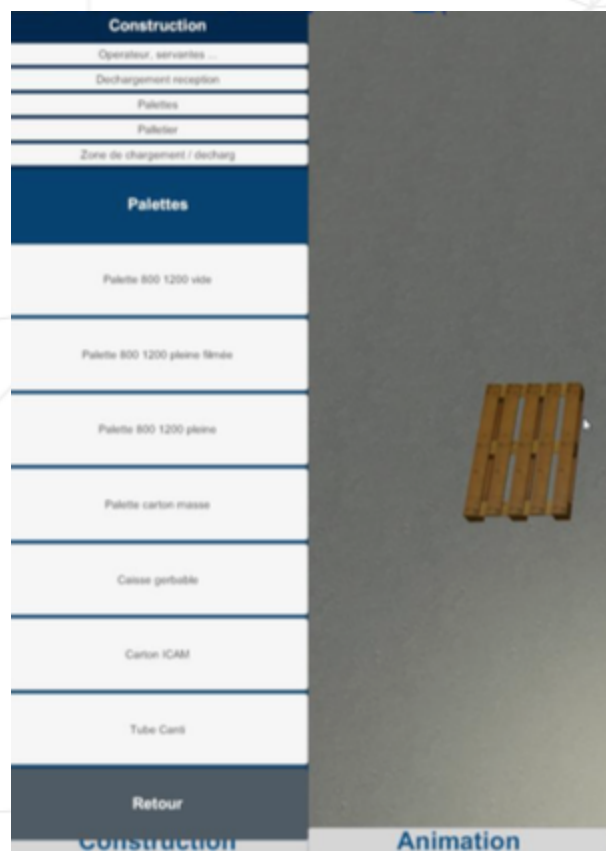
8. Make the application ergonomic:

- Menu
- Buttons

We will provide you with a set of 3D models as well as human animations.



## V.3 mages from the POC





# Chapter VI

## Partie bonus

If you have correctly followed the very precise specifications in the mandatory part, here are some bonuses we propose to you:

- Add a touch of realism to the project.
- Any additional bonus you will find useful, if you think correctors will share your opinion on what's useful.

# Chapter VII

## Submission and peer-evaluation

Submit your work on your GiT repository as usual. Only the work on your repository will be graded.