



# SMARTSIM

DL SMART-AI

ARTIFICIAL INTELLIGENCE  
COURSE



DE LORENZO

# SMART SIMULATOR FOR LEARNING AI WITH PYTHON

The DL SMART-AI is a software that has been developed to teach artificial intelligence with Python in a unique and effective way.

With this software, students can improve their individual experience on studying artificial intelligence in practice.

Professors can explore this trainer to provide experiments to students with the following topics:

- ✔ **Optimization: Introduction, definition, time and cost problems;**
- ✔ **Classification: Neural networks, signal generation, TensorFlow, predictions and failure predicts;**
- ✔ **Reinforcement Learning: Introduction and comparative to modern control;**
- ✔ **Decision trees: Application on regression.**

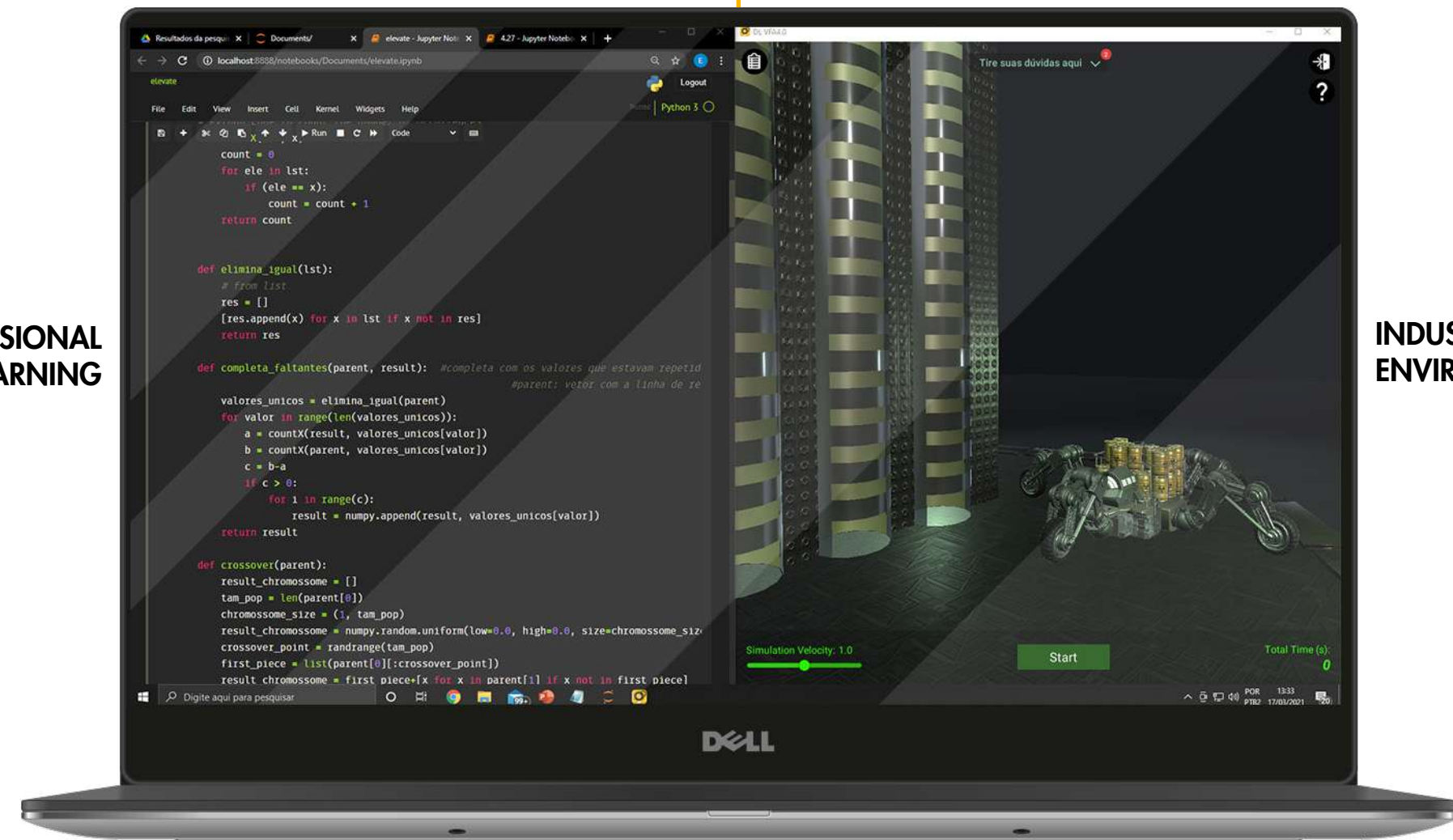
This software works integrated to a Python IDE (not included).

**PYTHON PROGRAMMING TOOLS**

**POWERFUL 3D SIMULATOR**

**PROFESSIONAL LEARNING**

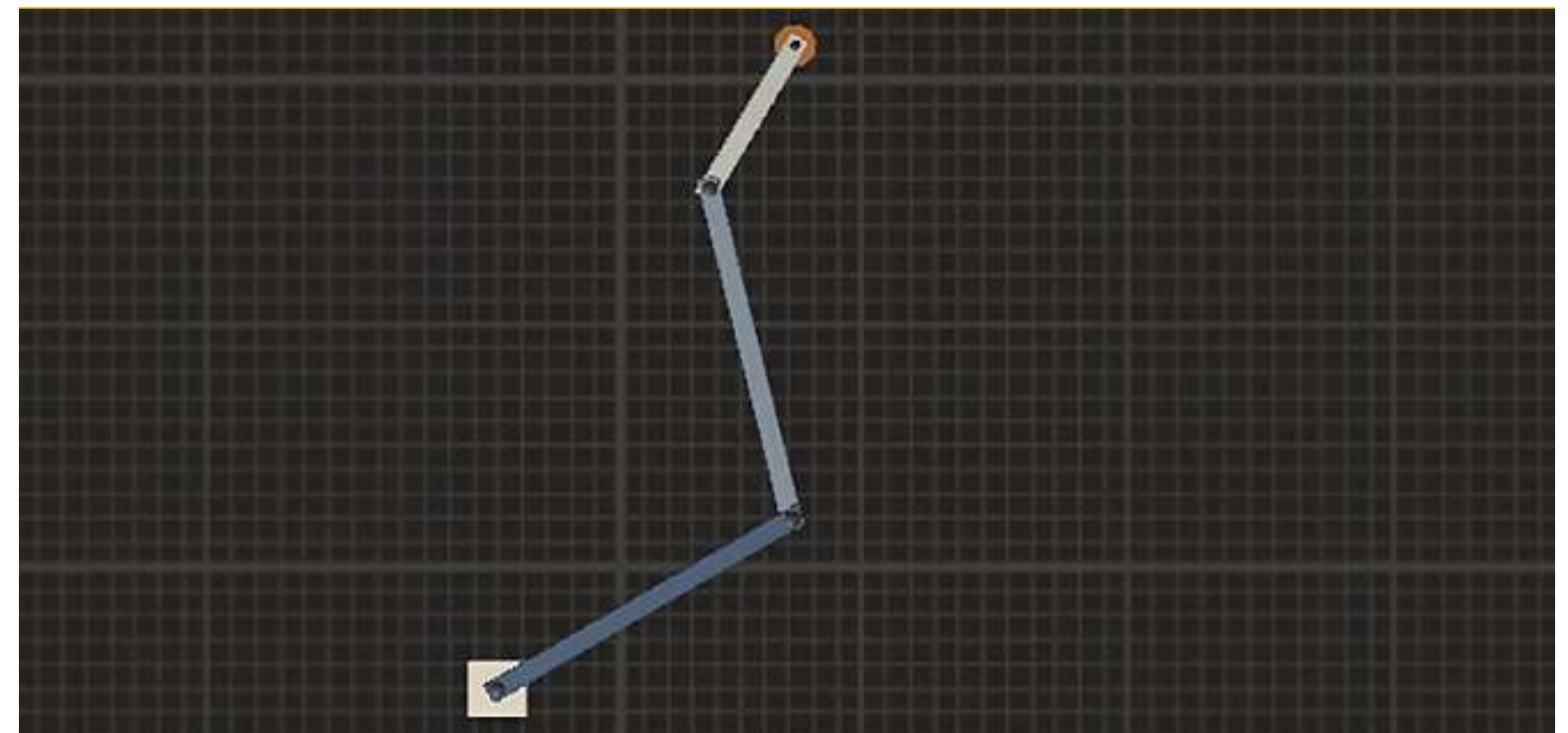
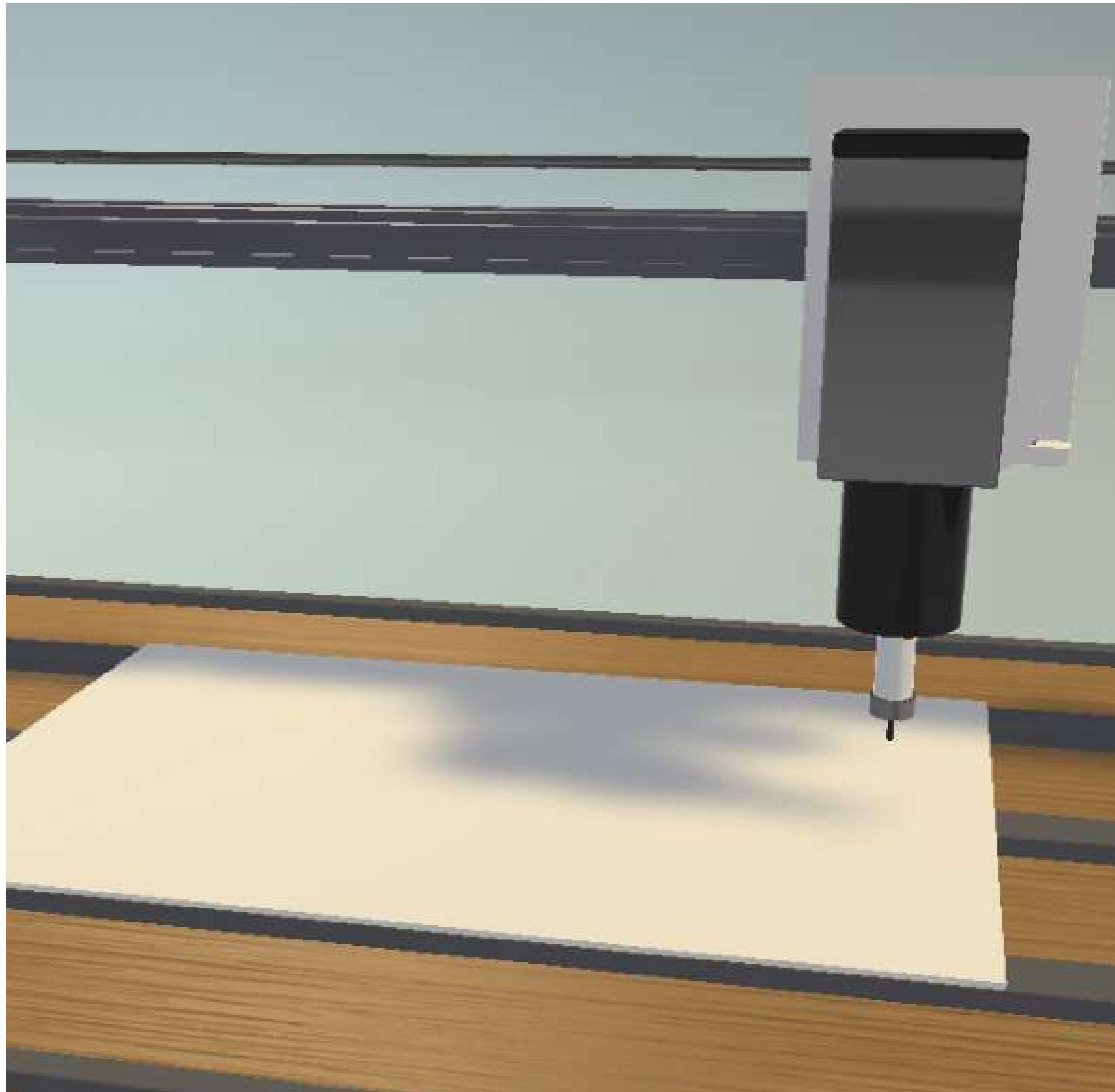
**INDUSTRIAL REALISTIC ENVIRONMENTS**



**PROFESSIONAL EXPERIENCE**

**REAL-LIFE SITUATIONS**

# 3D INDUSTRIAL ENVIRONMENTS TO PROVIDE REAL PRACTICAL EXPERIENCE TO STUDENTS

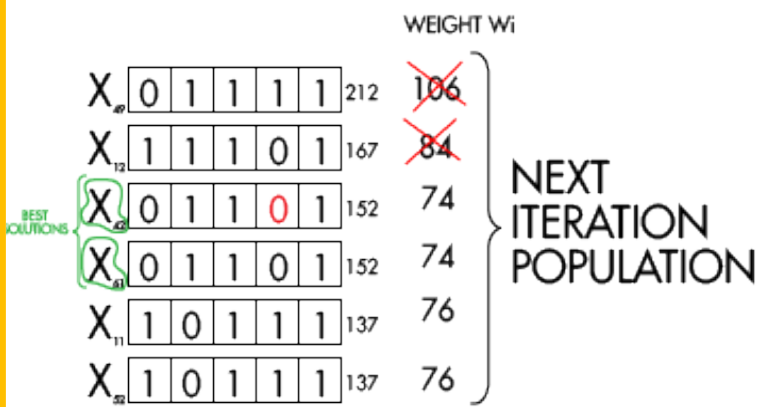




# EFFECTIVE LEARNING WITH GUIDANCE, REAL-LIFE PROJECTS, THEORY AND INSTRUCTIONS FROM BASIC TO ADVANCED

1

## POPULATION ADJUSTMENT

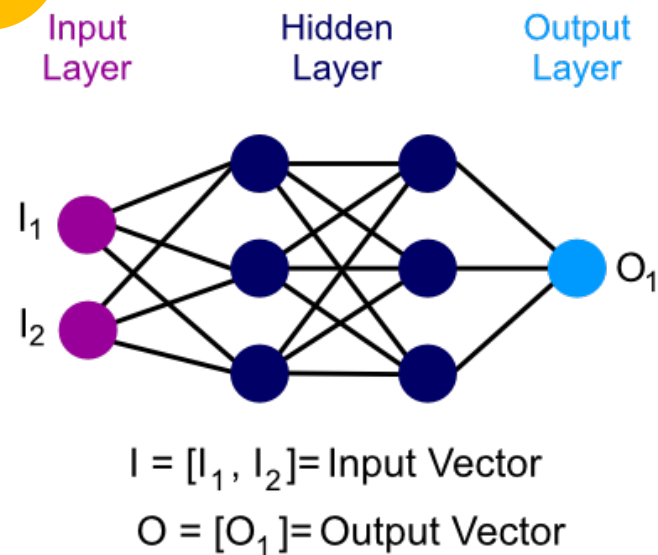


### OPTIMIZATION

**Goal:** Use genetic algorithm to resolve optimization problems, like the time problem or the cost one.

**AI concepts:** Introduction, genetic algorithm.

2

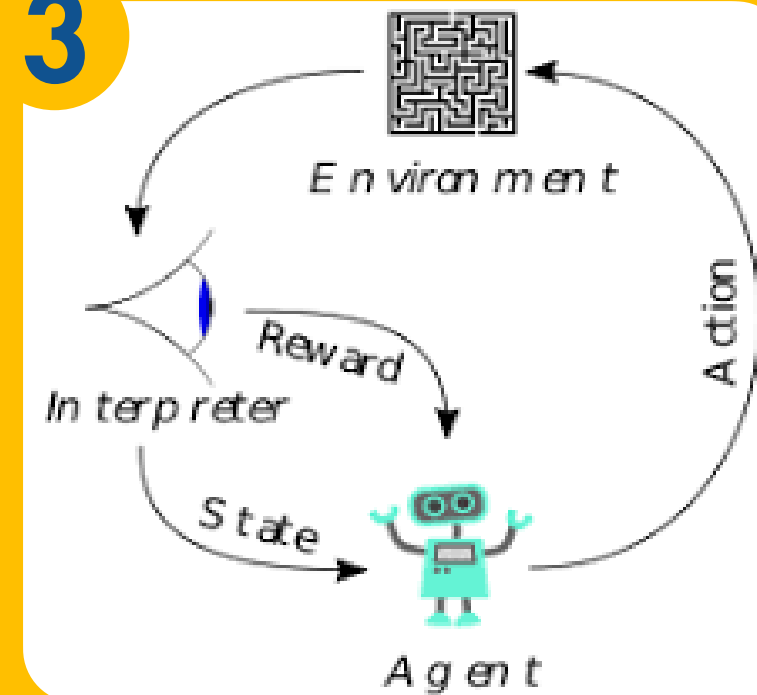


### CLASSIFICATION

**Goal:** Use neural networks to resolve classification problems.

**AI concepts:** Neural networks.

3

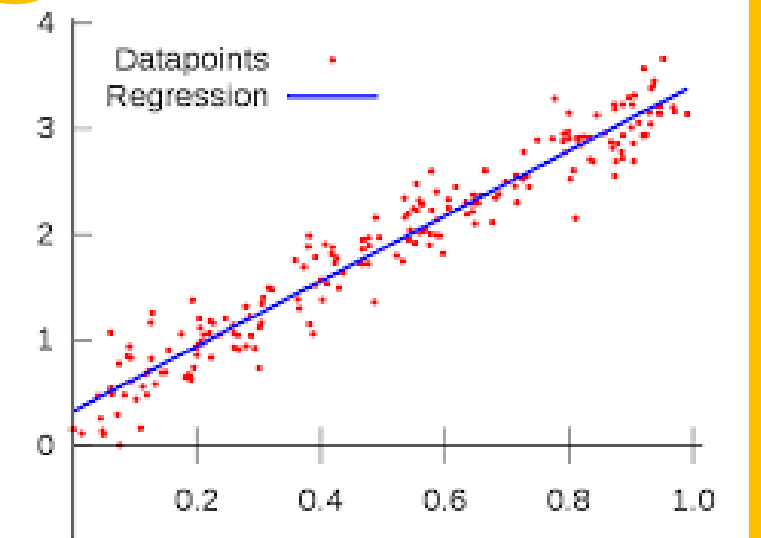


### REINFORCEMENT LEARNING

**Goal:** Use reinforcement learning to train a robot and a lead screw to reach a specific position.

**AI concepts:** Reinforcement learning.

4



### REGRESSION

**Goal:** Compare performances of decision tree and neural network algorithms in system modeling and predictions.

**AI concepts:** Decision trees.

# STUDENT CAN LEARN AND PRACTICE FROM BASIC TO ADVANCED AI TOPICS

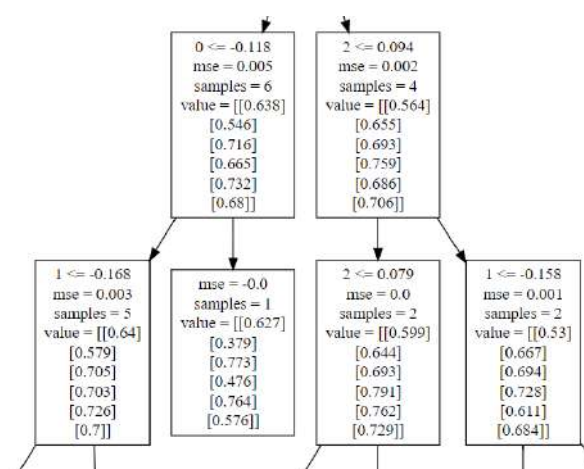
With the industrial 3D environments and also the built-in projects it's possible to develop solutions that evolve optimization, genetic algorithms, regression, neural networks and a lot more.

## PYTHON INSTRUCTIONS

```
def crossover(parent):
    result_chromosome = []
    tam_pop = len(parent[0])
    chromosome_size = (1, tam_pop)
    result_chromosome = numpy.random.uniform(low=0.0, high=0.0, size=chromosome_size)
    crossover_point = randrange(tam_pop)
    first_piece = list(parent[0][:crossover_point])
    result_chromosome = first_piece+[x for x in parent[1] if x not in first_piece]
    result_chromosome = numpy.asarray(result_chromosome)
    result_chromosome = completa_faltantes(parent[0], result_chromosome)
    return result_chromosome

def mutate(chromosome):
    chromosome = chromosome.astype(int) #transform a float array into a integer array
    lista = chromosome.tolist() # transform into list to use the function "sample"
    i, j = numpy.random.randint(low=0, high=len(chromosome), size=2) # choose two samples of the list (chromosome)
    lista[i], lista[j] = lista[j], lista[i]
    lista_saída = numpy.asarray(lista)
    return lista_saída
```

## DECISION TREE



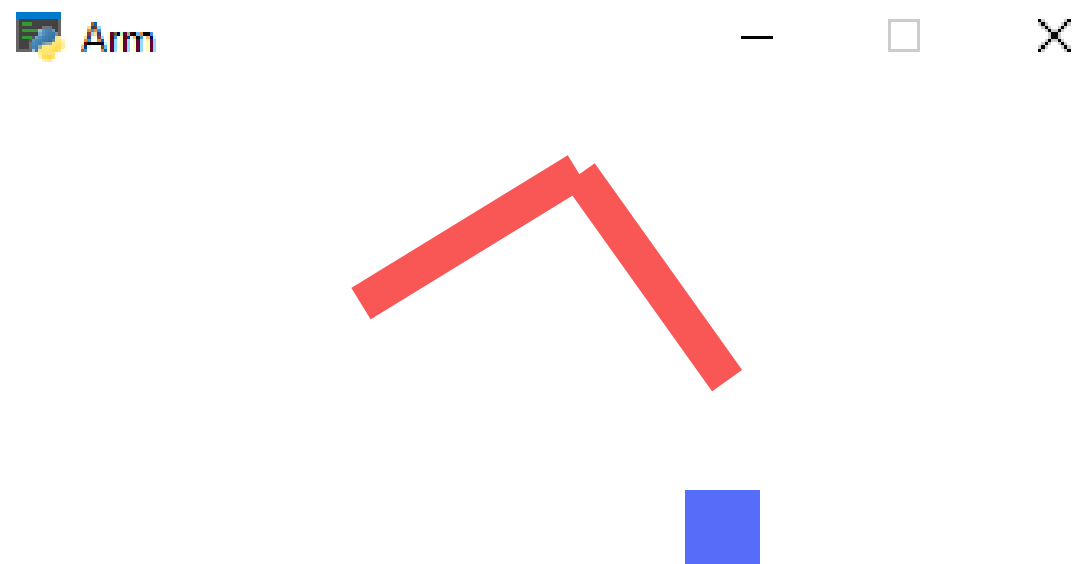
## PREDICTIONS

```
predictions = model.predict(testX)

predictions[10]
predictions[80]
predictions[50]
predictions[90]

array([9.999940e-01, 1.9931608e-07, 1.7625602e-07, 2.5123788e-07],
      dtype=float32)
```

## VIRTUAL TRAINING ROBOT



## TRAINING

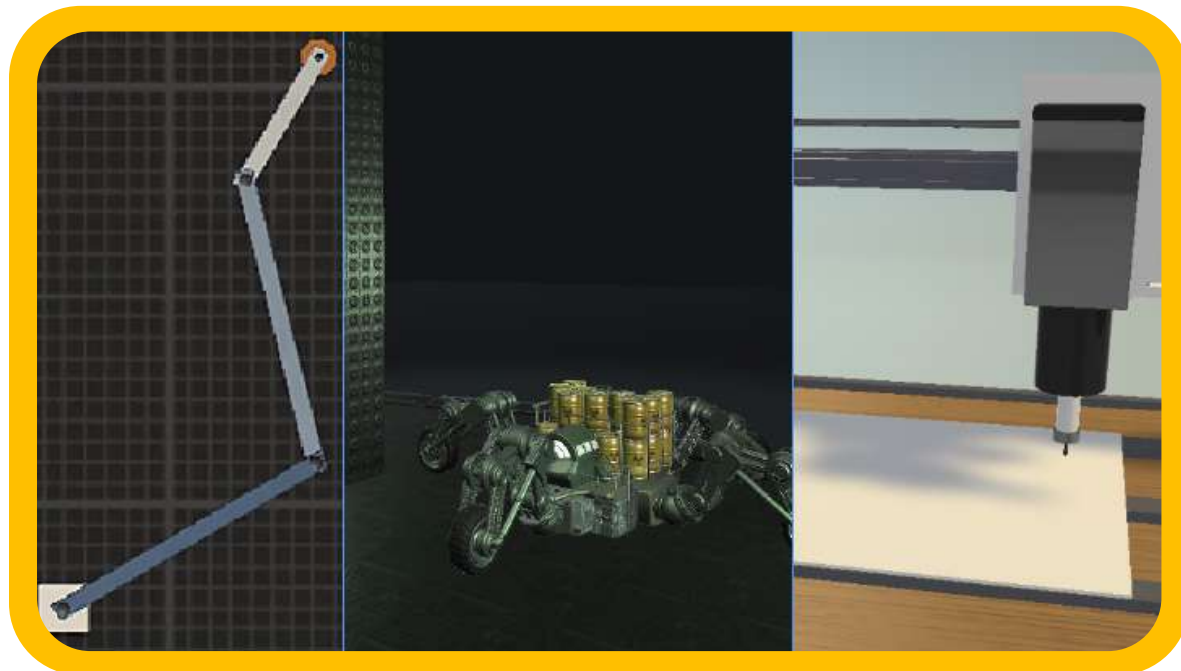
```
Epoch 2245/4000
1280/1280 [=====] - 1s 998us/step - loss: 0.1307 - accuracy: 0.9531 - val_loss: 0.1074 - val_accuracy: 0.9719
Epoch 02245: early stopping
Train: 0.972, Test: 0.972
```





# SUMMARY OF FEATURES

## IT'S A 3D SIMULATOR



## IT HAS BUILT-IN PROJECTS

Artificial Intelligence

### NEURAL NETWORK TRAINING

As we already know how to use a neural network in Tensorflow, we will comment only on the changes that need to be made to solve a regression problem.

Open the program "Neural Network - State Machine"

We will use the same network that we use, but with some modifications, we modify the number of neurons in the input, which must be equal to the number of input variables of our problem, which are 4, and the number of output neurons, for the number of engines that we want to control, which are 2.

We also changed the function of activating the output to sigmoid, as our output must be between 0 and 1. Note that the output will not be exact, but a poorly trained network will give for example 0.4 whenever it is 0 and 0.6 whenever it is 1. Now a better trained network, it will give for example 0.1 whenever it is 0 and 0.9 whenever it is 1. As we already have our truth table with all possible known states, we will not need a test group, because we will never it will be possible to find values different from these.

Another thing we need to change is the metric for Mean Squared Error (MSE) and Mean Absolute Error (MAE), which are good metrics for regression.

It is very important that you identify and understand the changes that we have made! See how our model looks:

## THE PROJECTS INCLUDE GUIDANCE

Change the activation function of the output layer to the linear function. Because we want our output to be an integer that will tell the position of the joint.

Train the network and see the output errors, if it doesn't look good, change the hyperparameters.

Did it look good? Use the predict function to predict some values.

## + CONTENTS AND SUPPORT MATERIALS, SO THEY CAN LEARN BY THEMSELVES

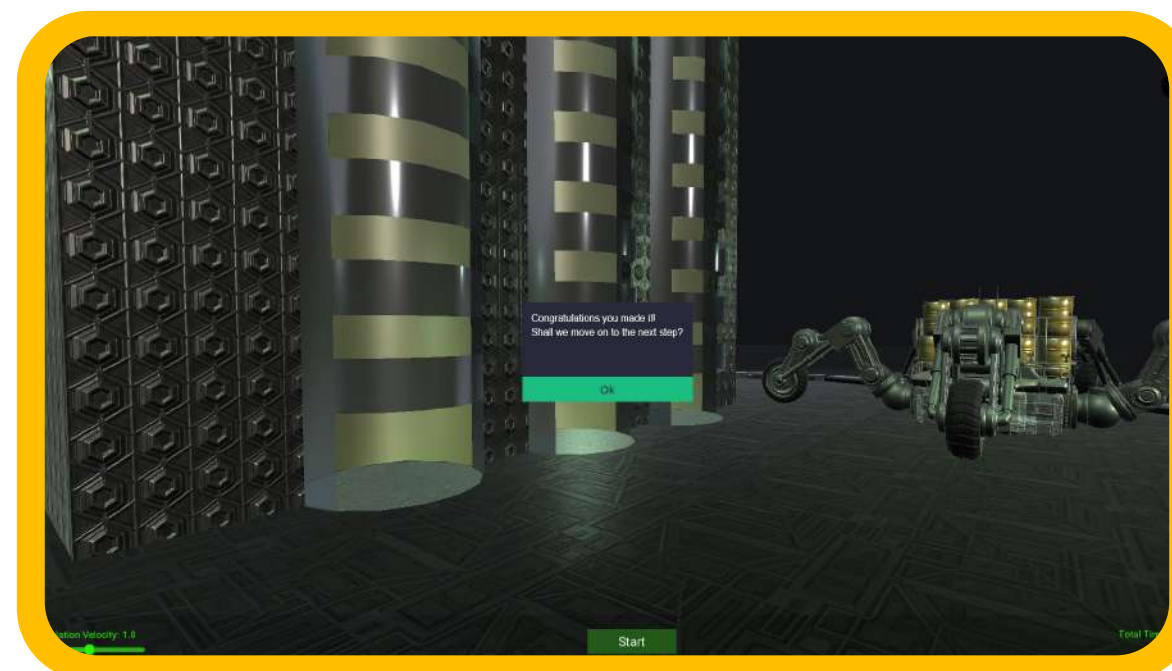
Let's study a little bit about GA before we continue. For that, you can take a look at the video below and, if you want to complement, see also the links:

Video 1: [https://www.youtube.com/watch?v=FYF6IS\\_BHKA](https://www.youtube.com/watch?v=FYF6IS_BHKA)

Link 8: <https://conteudo.icmc.usp.br/pessoas/andre/research/genetic/>

Link 9: <https://www.nce.ufrj.br/GINAPE/VIDA/alggenet.htm>

## IT AUTOMATICALLY CHECKS STUDENT ACTIVITIES TO LET THEM MOVE ON, LIKE IN GAME



## PROFESSORS CAN MONITOR STUDENTS, AND VERIFY WHICH POINT THEY NEED HELP (Option available with Dashboard)

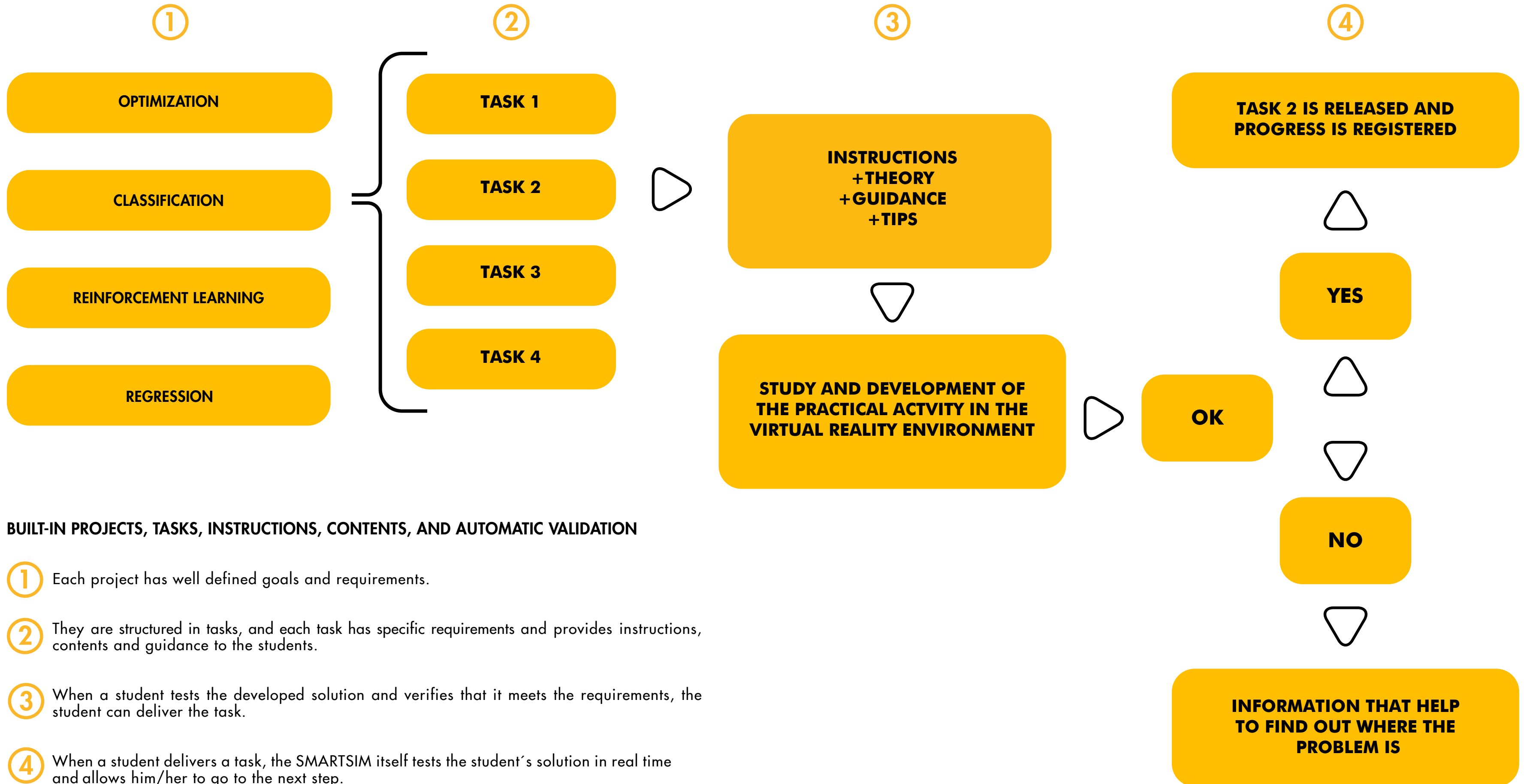
Student Progress

INSTITUTION NAME

Group:  Course:  User:

User Progress (POLI)		User Activities	
Student	Progress	Timestamp	Task -> Task-Description
Student 1	100%	Aug 26, 2019	1.1 - Breaking the inertia
Student 2	100%	Aug 26, 2019	1.2 - Interlocking with endswitches
Student 3	100%	Aug 26, 2019	1.3 - Retentive command
Student 4	100%	Aug 26, 2019	1.4 - Adding other interlocks
Student 5	100%	Aug 26, 2019	1.5 - Using the remote button
Student 6	100%	Aug 26, 2019	2.1 - Manual operation
Student 7	100%	Aug 27, 2019	2.2 - Simultaneous commands
Student 8	100%	Aug 27, 2019	2.3 - Adding water
Student 9	100%	Aug 27, 2019	2.4 - Adjusting the conveyors
Student 10	100%	Aug 30, 2019	3.1 - Dosing station
		Aug 30, 2019	3.2 - Mixing station
		Sep 3, 2019	3.3 - Filling the recipient

# HOW ARE BUILT-IN PROJECTS STRUCTURED?



## BUILT-IN PROJECTS, TASKS, INSTRUCTIONS, CONTENTS, AND AUTOMATIC VALIDATION

- ① Each project has well defined goals and requirements.
- ② They are structured in tasks, and each task has specific requirements and provides instructions, contents and guidance to the students.
- ③ When a student tests the developed solution and verifies that it meets the requirements, the student can deliver the task.
- ④ When a student delivers a task, the SMARTSIM itself tests the student's solution in real time and allows him/her to go to the next step.



# SYSTEM REQUIREMENTS

## ORDER CODES

### DL SMART-AI

ARTIFICIAL INTELLIGENCE COURSE

### DL SMART-DASHBOARD

CLASSROOM MANAGEMENT DASHBOARD FOR SMARTSIMS

### IMPORTANT NOTE:

THIS PRODUCTS DOES NOT INCLUDE ANY THIRD PARTY SOFTWARES. TO OUR KNOWLEDGE, ANACONDA PYTHON CAN BE FREE DOWNLOADED AT ANACONDA WEBSITE.

## MINIMUM REQUIREMENTS

### OPERATIONAL SYSTEM

64-BIT WINDOWNS 10

### DIRECTX VERSION

DIRECTX 11

### PROCESSOR

INTEL i5 9400F OR AMD RYZEN 5 3600

### MEMORY

8GB

### GHRAPHIC CARD

### STORAGE

HDD (1GB)

## RECOMMENDED REQUIREMENTS

### OPERATIONAL SYSTEM

64-BIT WINDOWNS 10 PRO

### DIRECTX VERSION

DIRECTX 12

### PROCESSOR

INTEL i7 9700 OR AMD RYZEN 7 3700X

### MEMORY

16 GB

### GHRAPHIC CARD

NVIDIA GTX 1050 TI 4GB OR RX 550 4GB

### STORAGE

HDD (1GB)