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.**

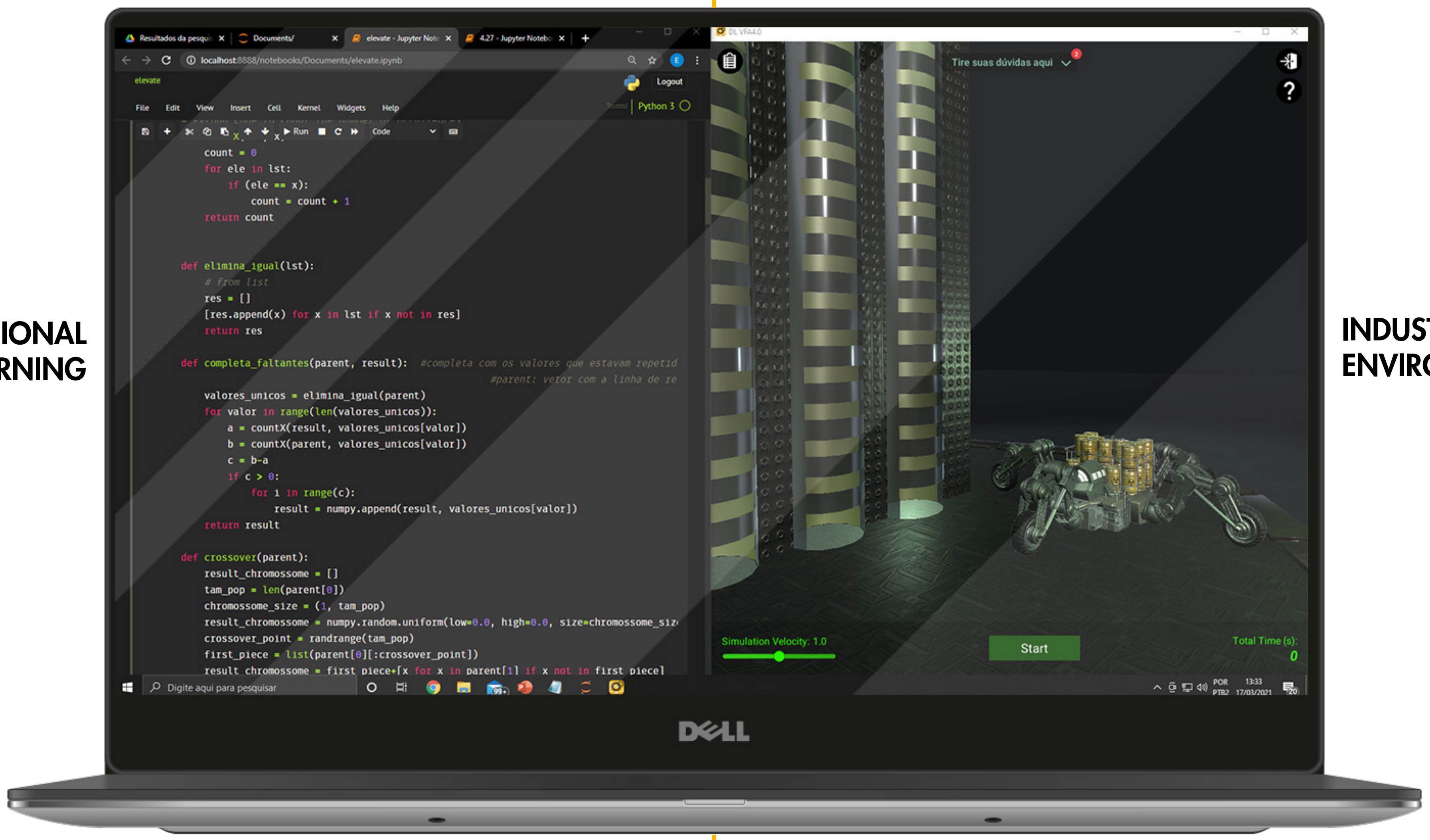
This software works integrated to a Python IDE (not included).

PYTHON PROGRAMMING TOOLS

POWERFUL 3D SIMULATOR

PROFESSIONAL LEARNING

INDUSTRIAL REALISTIC ENVIRONMENTS

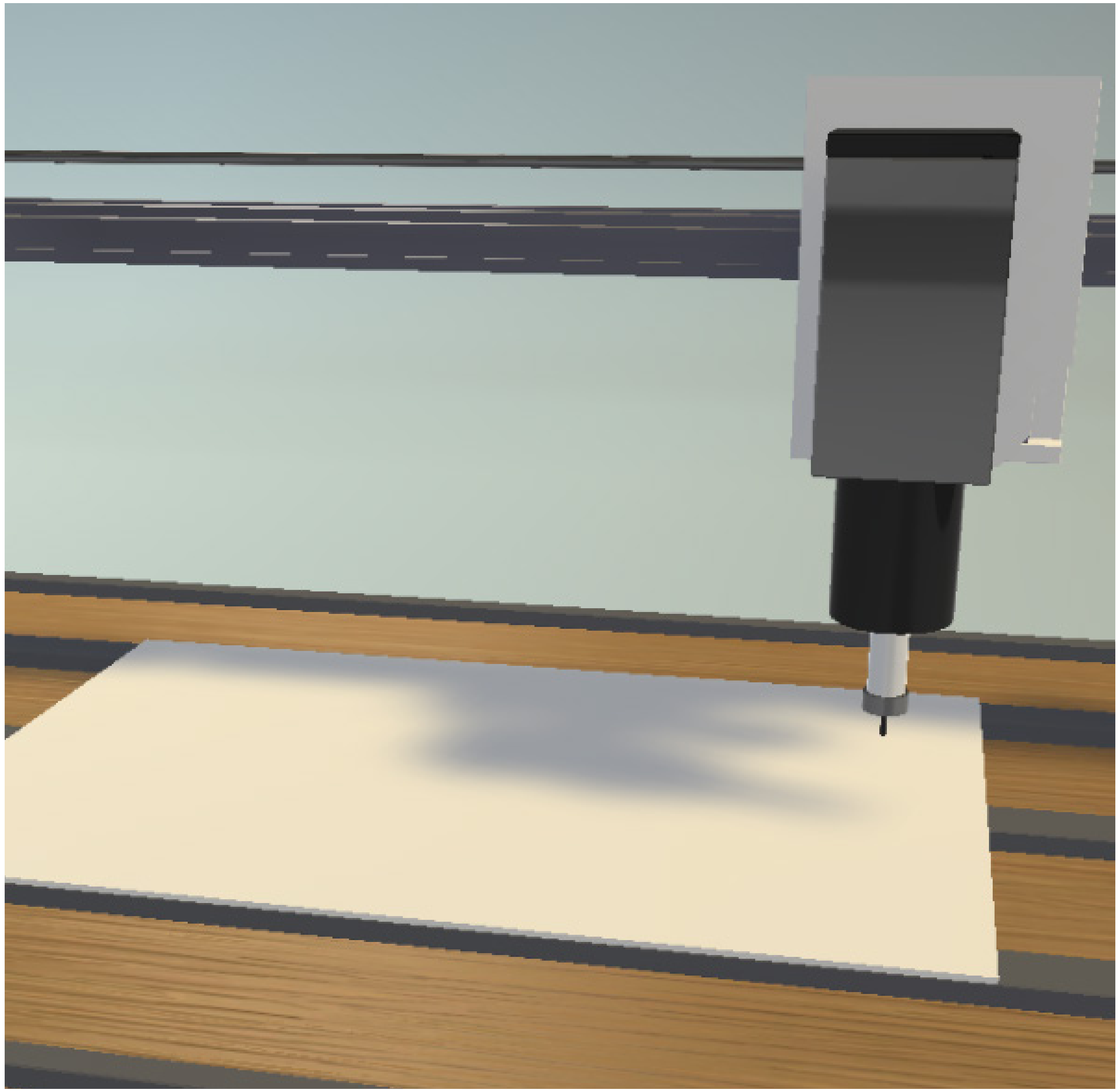


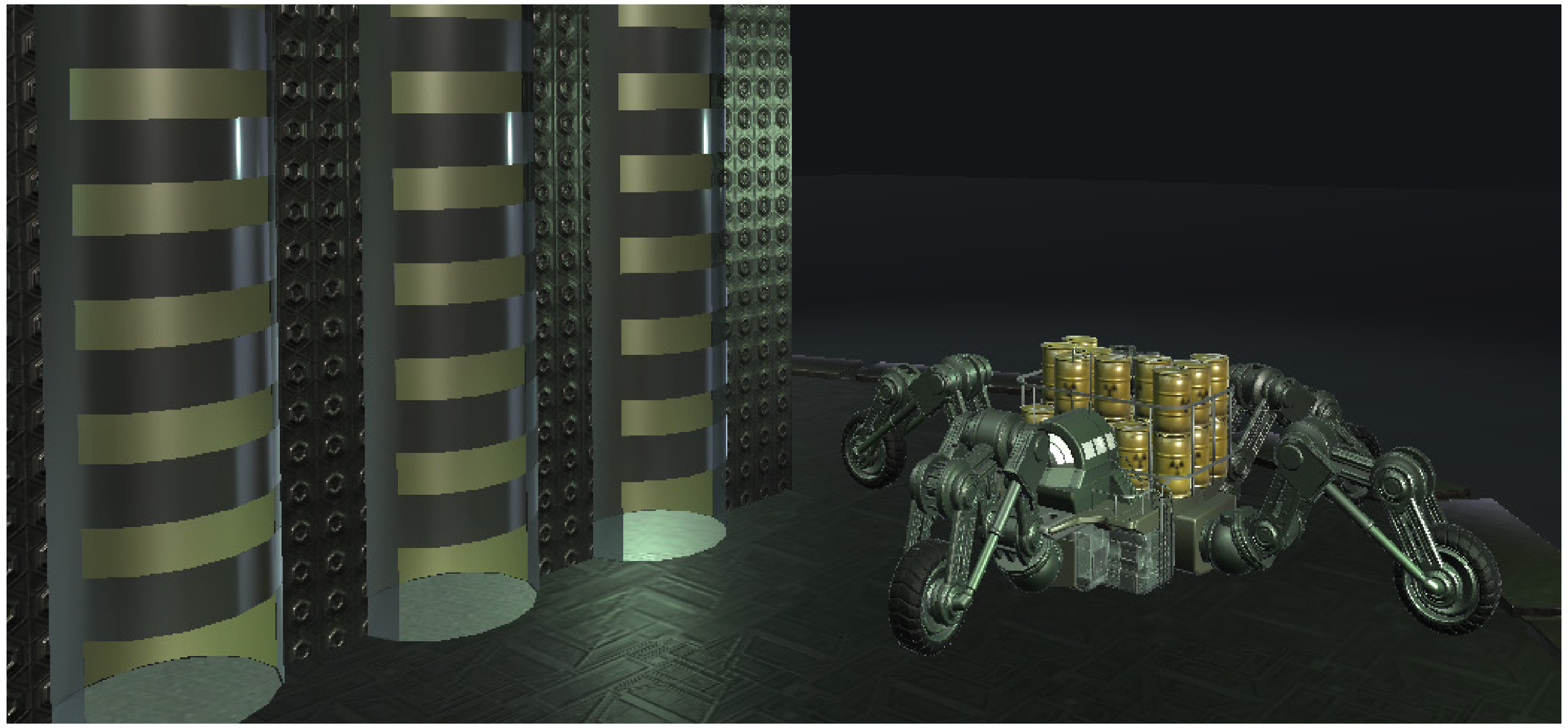
```
    count = 0
    for ele in lst:
        if (ele == x):
            count = count + 1
    return count


def elimina_igual(lst):
    # from list
    res = []
    [res.append(x) for x in lst if x not in res]
    return res

def completa_faltantes(parent, result):  #completa com os valores que estavam repetid
                                         #parent: vetor com a linha de re

    valores_unicos = elimina_igual(parent)
    for valor in range(len(valores_unicos)):
        a = countX(result, valores_unicos[valor])
        b = countX(parent, valores_unicos[valor])
        c = b-a
        if c > 0:
            for i in range(c):
                result = numpy.append(result, valores_unicos[valor])
    return result

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

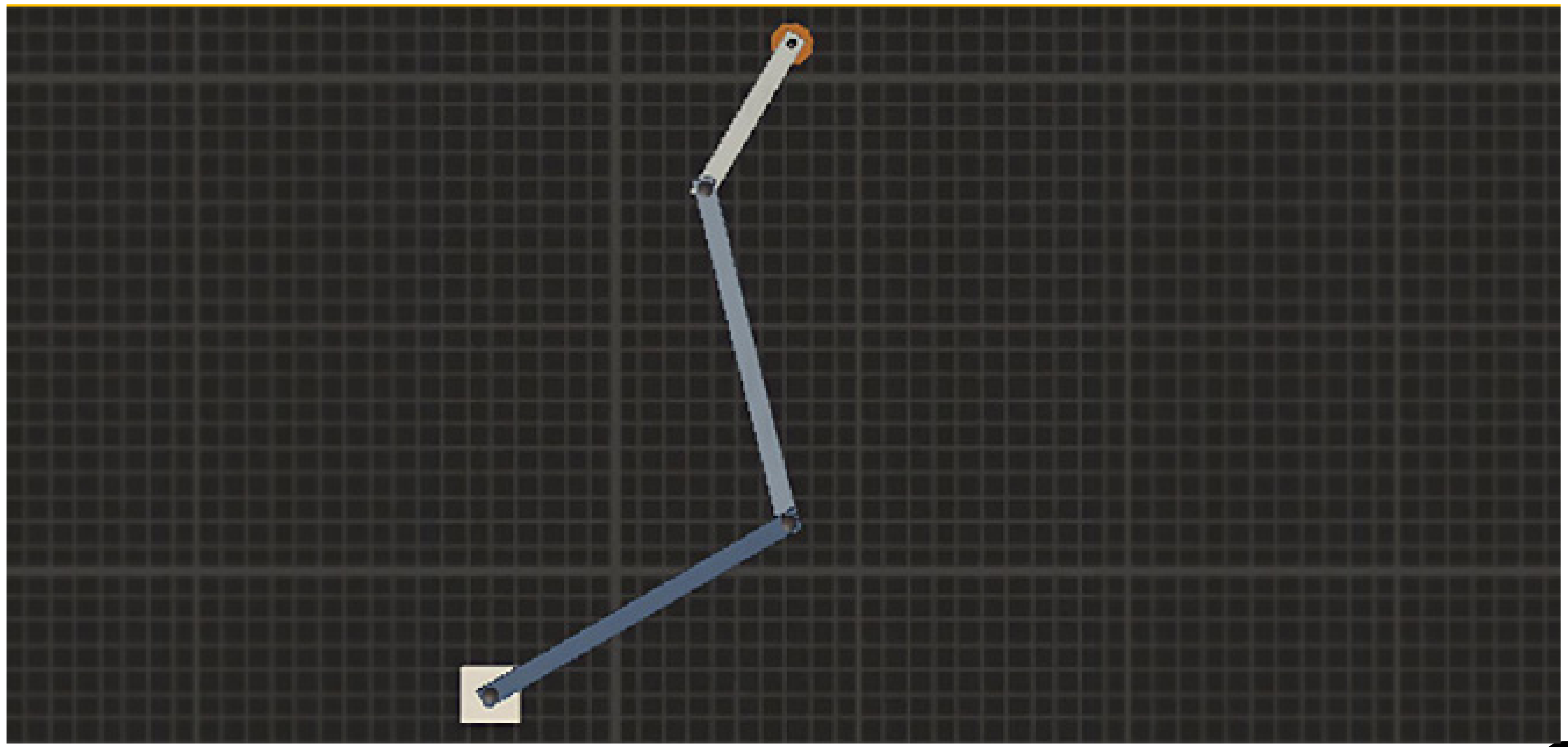
PROFESSIONAL EXPERIENCE

REAL-LIFE SITUATIONS

3D INDUSTRIAL ENVIRONMENTS TO PROVIDE REAL PRACTICAL EXPERIENCE TO STUDENTS

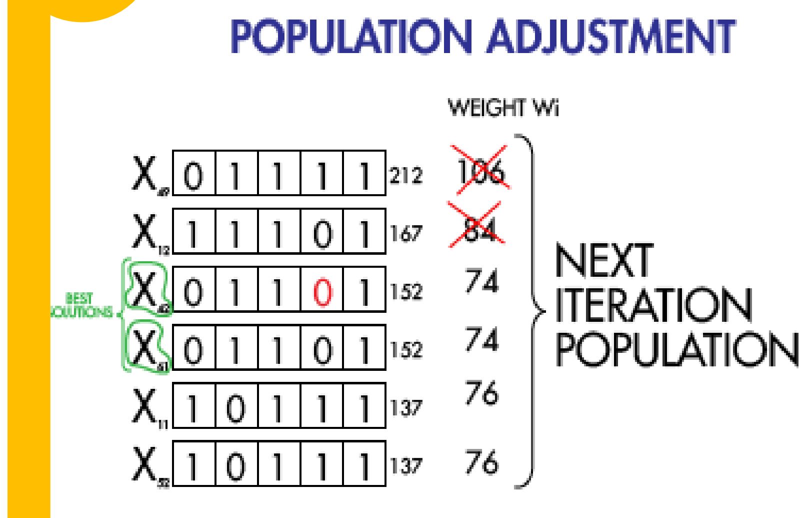






EFFECTIVE LEARNING WITH GUIDANCE, REAL-LIFE PROJECTS, THEORY AND INSTRUCTIONS FROM BASIC TO ADVANCED

1

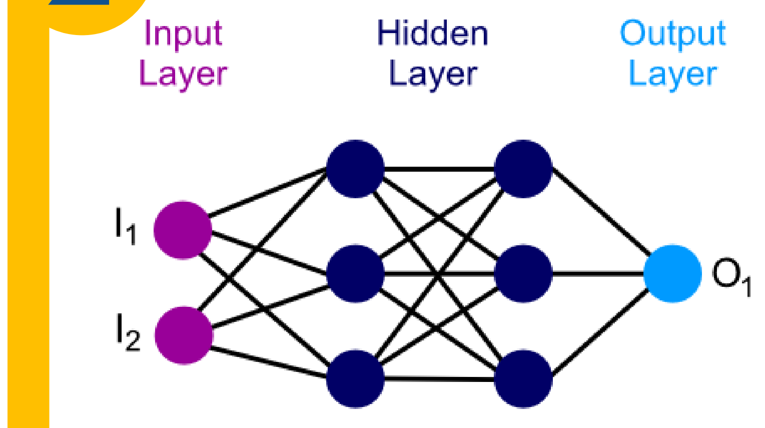


OPTIMIZATION

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

AI concepts: Introduction, genetic algorithm.

2



$I = [I_1, I_2]$= Input Vector

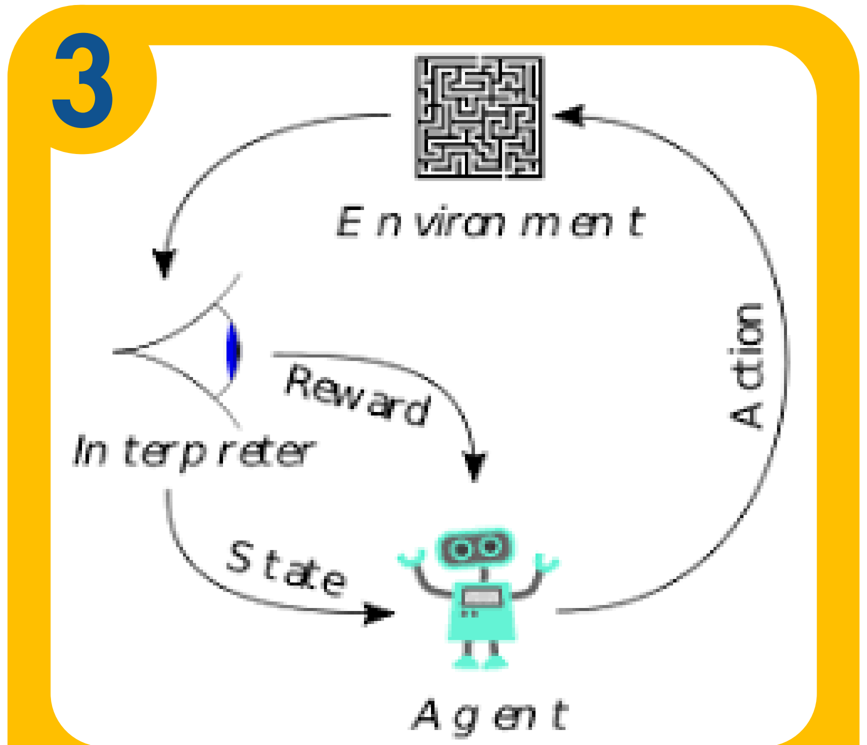
$O = [O_1]$= Output Vector

CLASSIFICATION

Goal: Use neural networks to resolve classification problems.

AI concepts: Neural networks.

3

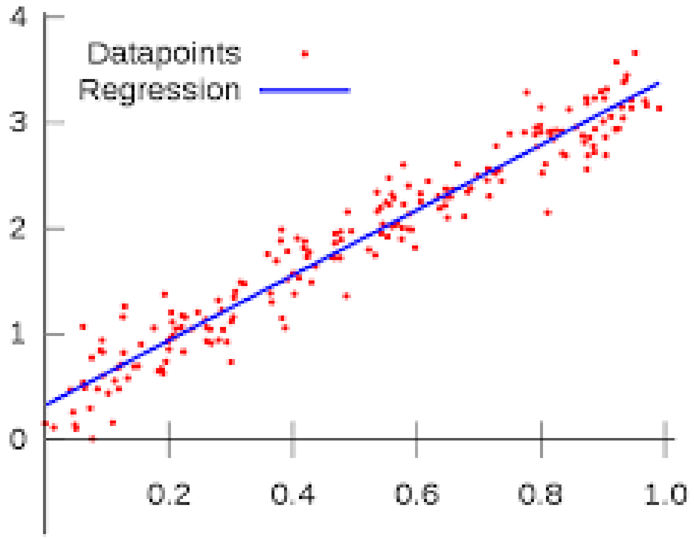


REINFORCEMENT LEARNING

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

AI concepts: Reinforcement learning.

4



REGRESSION

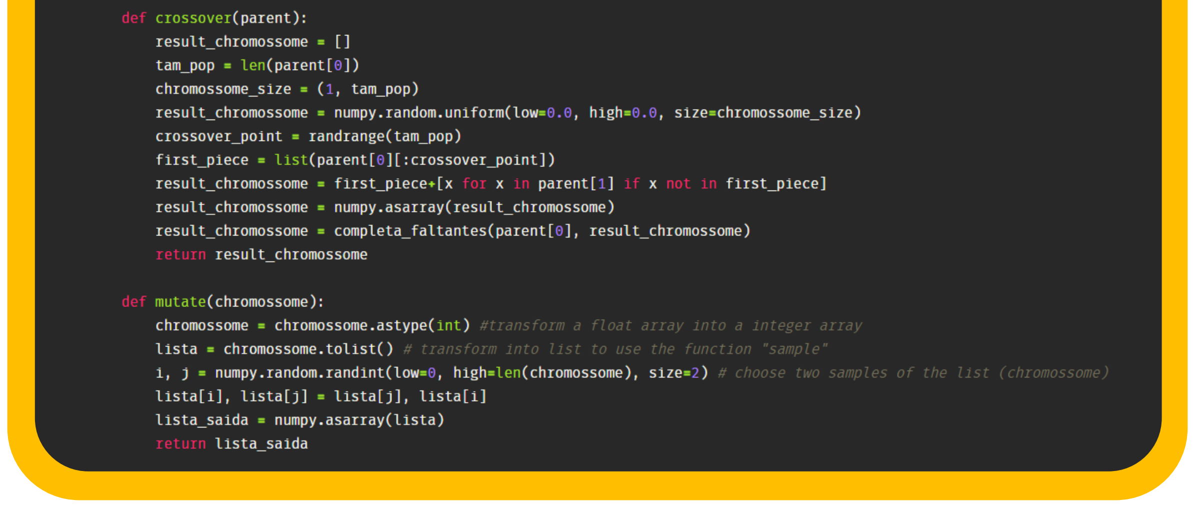
Goal: Compare performances of decision tree and neural network algorithms in system modeling and predictions.

AI concepts: Decision trees.

STUDENT CAN LEARN AND PRACTICE FROM BASIC TO ADVANCED AI TOPICS

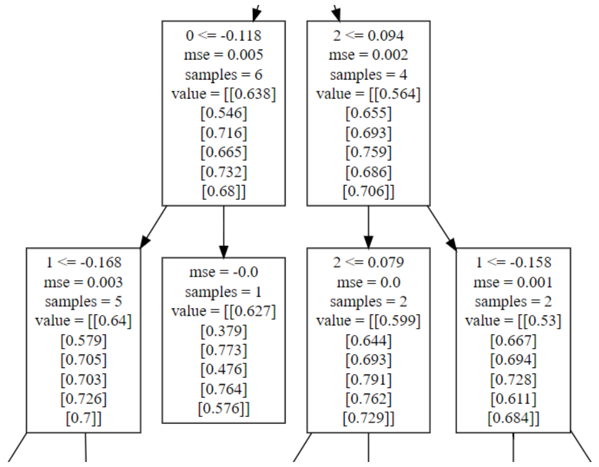
With the industrial 3D environments and also the built-in projects it´s possible to develop solutions that evolve optimization, genetic algorithms, regression, neural networks and a lot more.

PYTHON INSTRUCTIONS

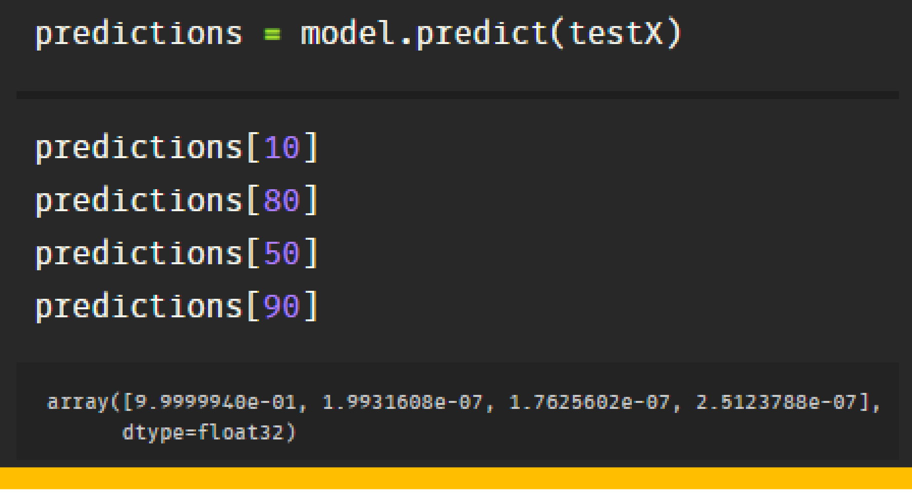


```python
def crossover(parent):
    result_chromossome = []
    tam_pop = len(parent[0])
    chromossome_size = (1, tam_pop)
    result_chromossome = numpy.random.uniform(low=0.0, high=0.0, size=chromossome_size)
    crossover_point = randrange(tam_pop)
    first_piece = list(parent[0][:crossover_point])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

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

DECISION TREE



PREDICTIONS

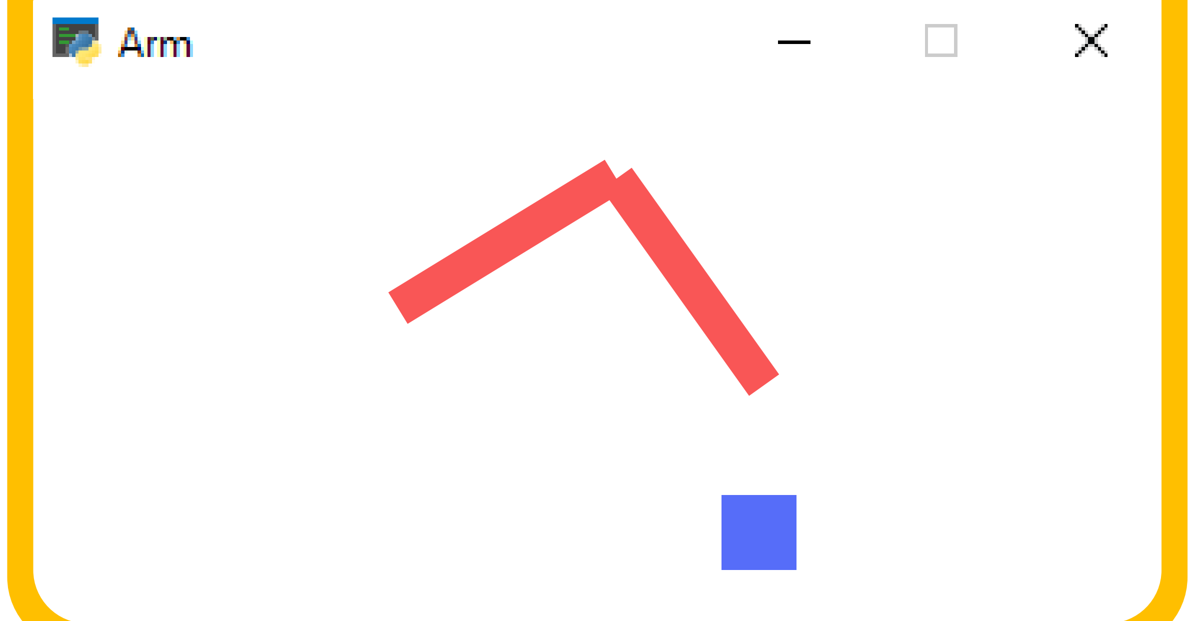


```python
predictions = model.predict(testX)
```

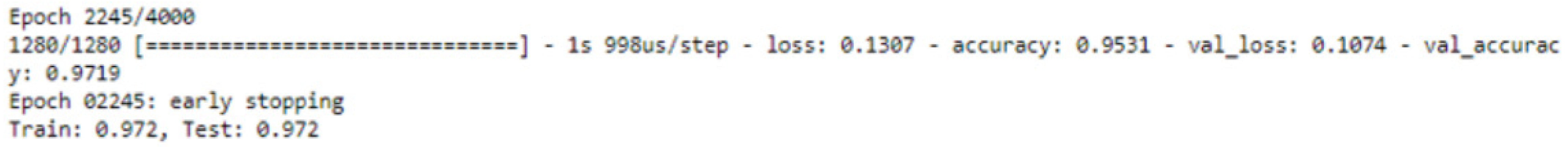
```python
predictions[10]
predictions[80]
predictions[50]
predictions[90]
```

```
array([9.9999940e-01, 1.9931608e-07, 1.7625602e-07, 2.5123788e-07],
      dtype=float32)
```

VIRTUAL TRAINING ROBOT



TRAINING



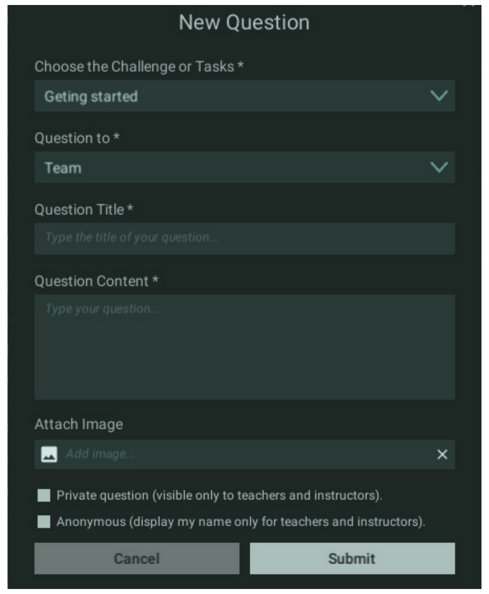
```
Epoch 2245/4000
1280/1280 [==============================] - 1s 998us/step - loss: 0.1307 - accuracy: 0.9531 - val_loss: 0.1074 - val_accurac
y: 0.9719
Epoch 02245: early stopping
Train: 0.972, Test: 0.972
```

WHY IS IT A SMARTSIM?

IT CONNECTS PROFESSOR, STUDENT AND SCHOOL

De Lorenzo´s cloud server receives students activities and provides reports and analytics to professors and institutions. Besides, a student can start working at school and continue at home or vice-versa.

The platform includes a query and answer system that enables professors to support the students counting on a team of monitors. That means better support with less effort of the professors. The students can see questions asked by other colleagues too so that way if more than one student have the same doubt the professors answer will attend them all.



New Question

Choose the Challenge or Tasks *
Geting started

Question to *
Team

Question Title *
Type the title of your question...

Question Content *
Type your question...

Attach Image
Add image...

□ Private question (visible only to teachers and instructors).
□ Anonymous (display my name only for teachers and instructors).

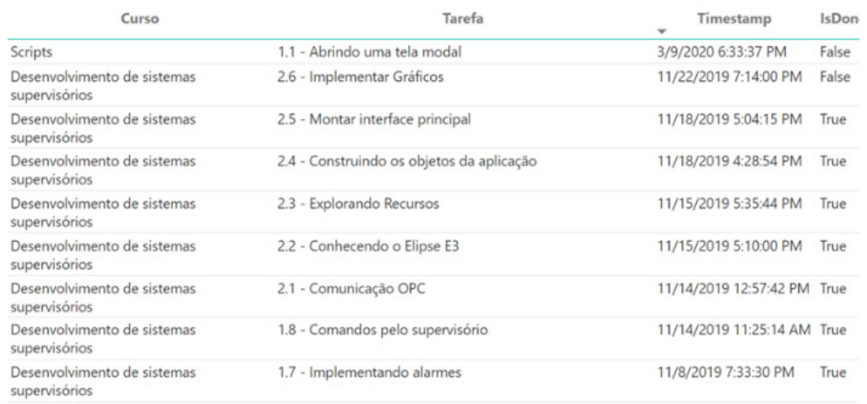
COMPATIBLE WITH THE DL SMART-DASHBOARD (SOLD SEPARETLY)

PROFESSORS CAN FOLLOW STUDENTS PROGRESS

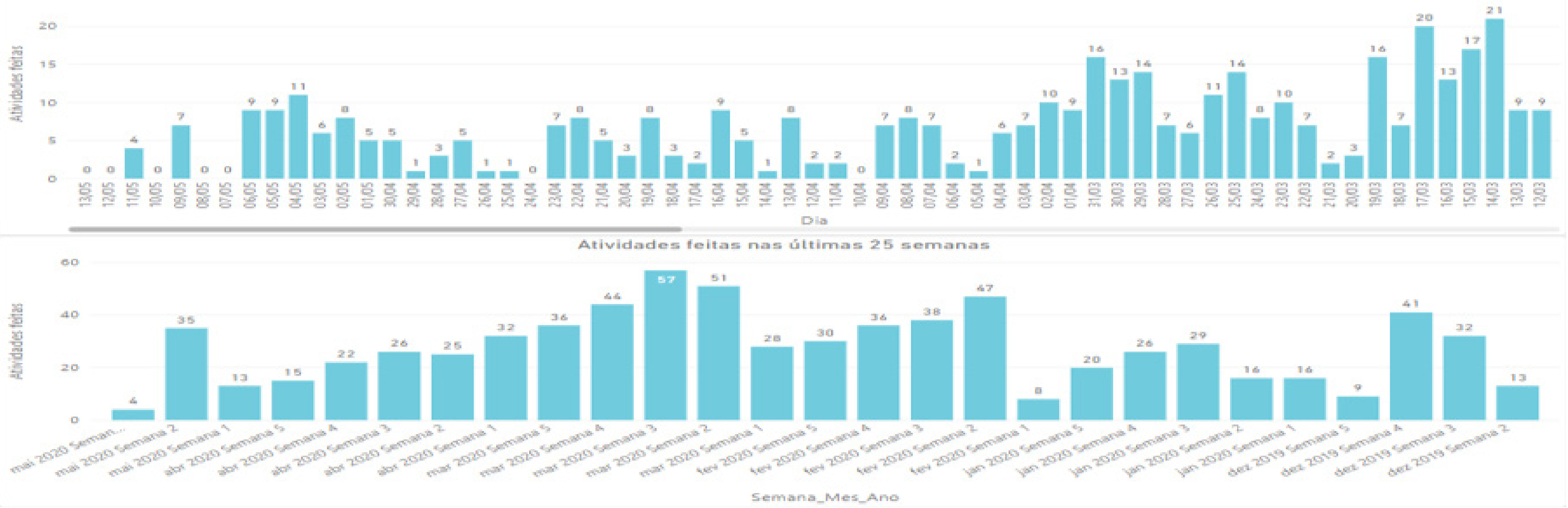
The professor can do and access everything the student can. Besides, he/she can also access the dashboard's portal.
It includes interesting reports and analytics that help the professor to monitor the group in real time, as well as to identify students who are doing very well, as well as those who need help, who are not working at all and who seem to be "cheating".

Tasks report

This is an important tool since it provides evidence of the activities a student worked on. That means the school has evidence of the practical activities the distance learner has done with detailed information about it.

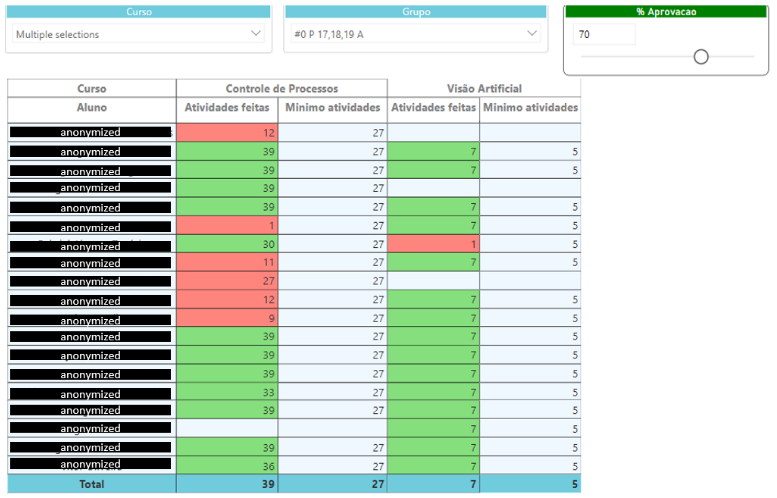


Curso	Tarefa	Timestamp	IsDon
Scripts	1.1 - Abrindo uma tela modal	3/9/2020 6:33:37 PM	False
Desenvolvimento de sistemas supervisórios	2.6 - Implementar Gráficos	11/22/2019 7:14:00 PM	False
Desenvolvimento de sistemas supervisórios	2.5 - Montar interface principal	11/18/2019 5:04:15 PM	True
Desenvolvimento de sistemas supervisórios	2.4 - Construindo os objetos da aplicação	11/18/2019 4:28:54 PM	True
Desenvolvimento de sistemas supervisórios	2.3 - Explorando Recursos	11/15/2019 5:35:44 PM	True
Desenvolvimento de sistemas supervisórios	2.2 - Conhecendo o Elipse E3	11/15/2019 5:10:00 PM	True
Desenvolvimento de sistemas supervisórios	2.1 - Comunicação OPC	11/14/2019 12:57:42 PM	True
Desenvolvimento de sistemas supervisórios	1.8 - Comandos pelo supervisório	11/14/2019 11:25:14 AM	True
Desenvolvimento de sistemas supervisórios	1.7 - Implementando alarmes	11/8/2019 7:33:30 PM	True



PROFESSOR CAN SEE WHICH STUDENTS ARE ON SCHEDULE

With this interface, the professor may choose which groups he/she wants to monitor, to verify who is on schedule, who is pending and so on. It is possible to define the expected progress percentage in relation to the tasks available in the course.



Curso: Multiple selections
Grupo: #0 P 17,18,19 A
% Aprovacao: 70

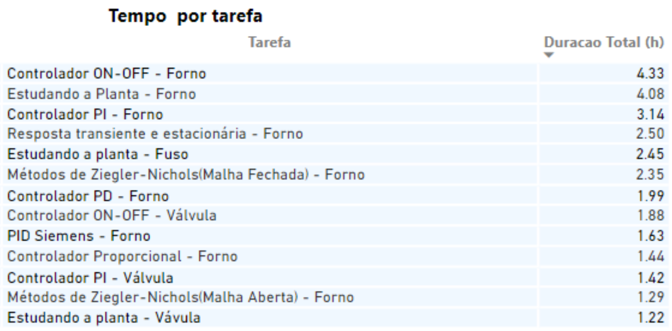
Curso	Controle de Processos		Visão Artificial	
Aluno	Atividades feitas	Minimo atividades	Atividades feitas	Minimo atividades
anonymized	12	27		
anonymized	39	27	7	5
anonymized	39	27	7	5
anonymized	39	27		
anonymized	39	27	7	5
anonymized	1	27	7	5
anonymized	30	27	1	5
anonymized	11	27	7	5
anonymized	27	27		
anonymized	12	27	7	5
anonymized	9	27	7	5
anonymized	39	27	7	5
anonymized	39	27	7	5
anonymized	39	27	7	5
anonymized	33	27	7	5
anonymized	39	27	7	5
anonymized			7	5
anonymized	39	27	7	5
anonymized	36	27	7	5
Total	39	27	7	5

RHYTHM

This other dashboard shows the number of activities the students did daily and weekly. The professor may decide to verify it regarding a whole group/class or a specific student.

EFFORT/TASK DEDICATED TIME

If the professor selects a student, he/she may verify how much time the student took to develop and deliver each task of the course.

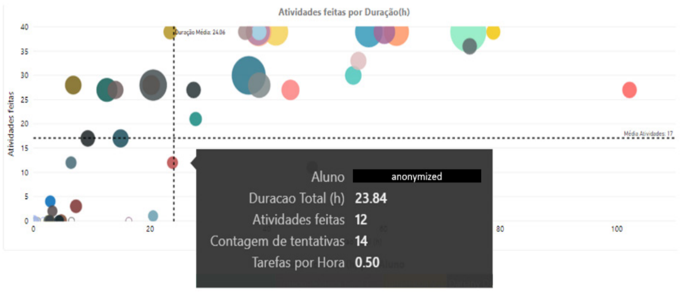


Tempo por tarefa

Tarefa	Duracao Total (h)
Controlador ON-OFF - Forno	4.33
Estudando a Planta - Forno	4.08
Controlador PI - Forno	3.14
Resposta transiente e estacionária - Forno	2.50
Estudando a planta - Fuso	2.45
Métodos de Ziegler-Nichols(Malha Fechada) - Forno	2.35
Controlador PD - Forno	1.99
Controlador ON-OFF - Válvula	1.88
PID Siemens - Forno	1.63
Controlador Proporcional - Forno	1.44
Controlador PI - Válvula	1.42
Métodos de Ziegler-Nichols(Malha Aberta) - Forno	1.29
Estudando a planta - Vávula	1.22

PROGRESS VS TIME TAKEN

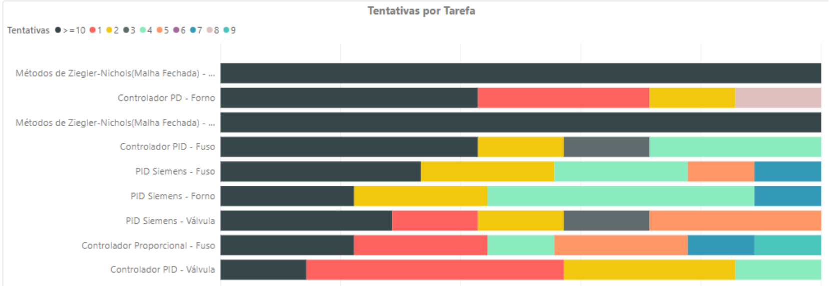
It is also possible to verify the distribution of the dedicated time with relation to the number of tasks done by each student at any period of time. That helps to identify who is doing well, who may need help, who is doing nothing and who is trying to cheat.



Aluno: anonymized
Duracao Total (h): 23.84
Atividades feitas: 12
Contagem de tentativas: 14
Tarefas por Hora: 0.50

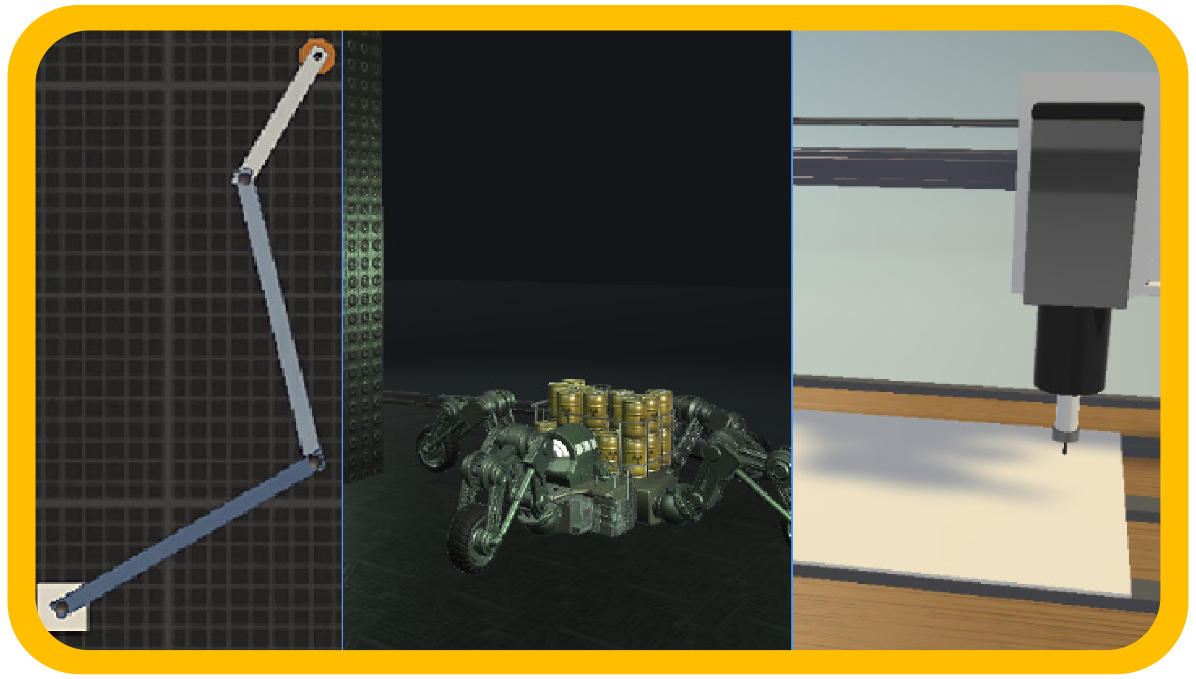
TRIALS PER TASK

This chart helps the teacher to understand which task may be the most difficult and which one may be the easiest in order to adjust the deadlines.

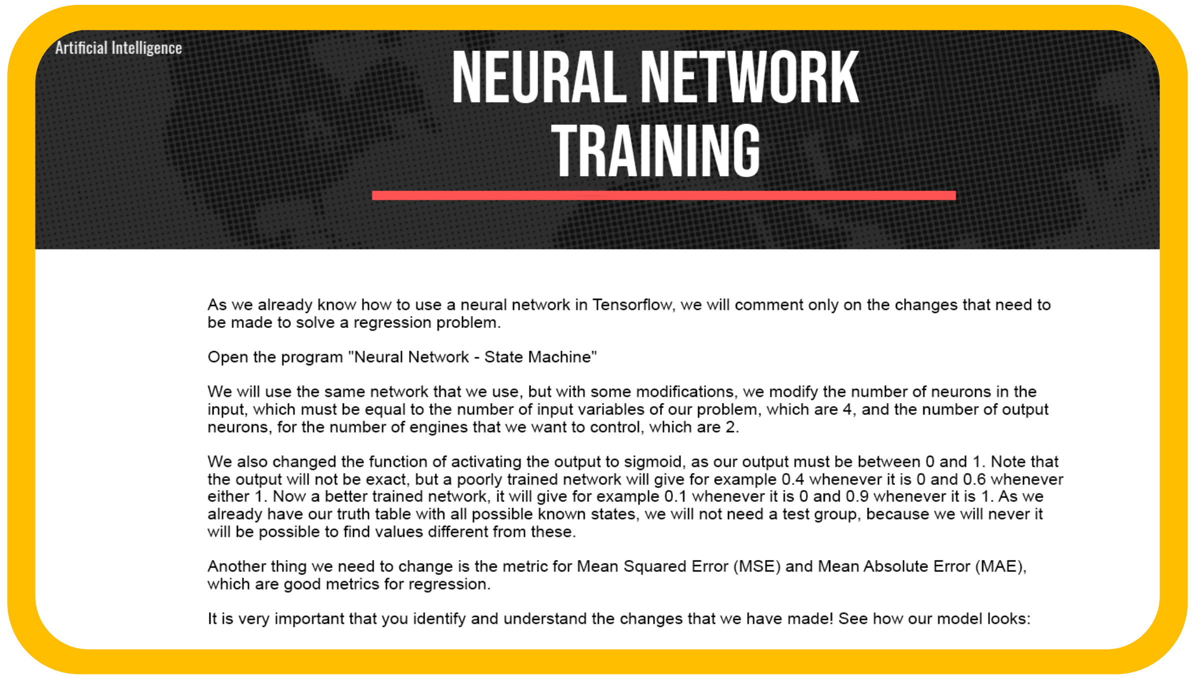


SUMMARY OF FEATURES

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"

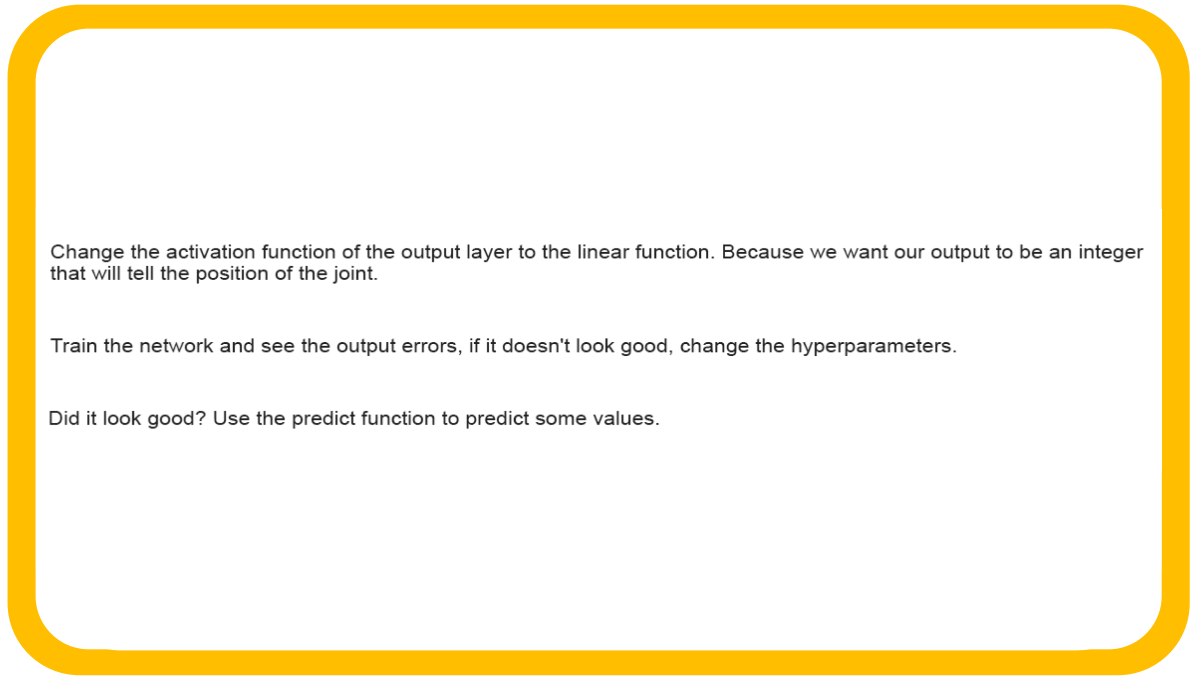
We will use the same network that we use, but with some modifications, we modify the number of neurons in the input, which must be equal to the number of input variables of our problem, which are 4, and the number of output neurons, for the number of engines that we want to control, which are 2.

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 either 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

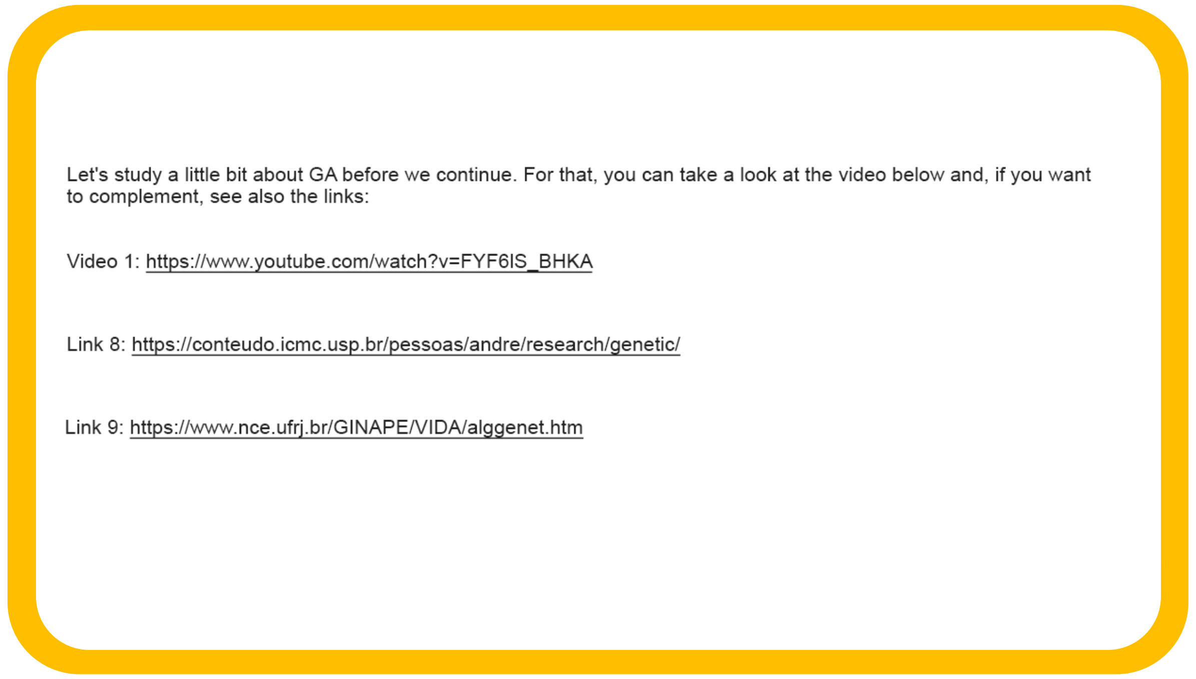


Change the activation function of the output layer to the linear function. Because we want our output to be an integer that will tell the position of the joint.

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



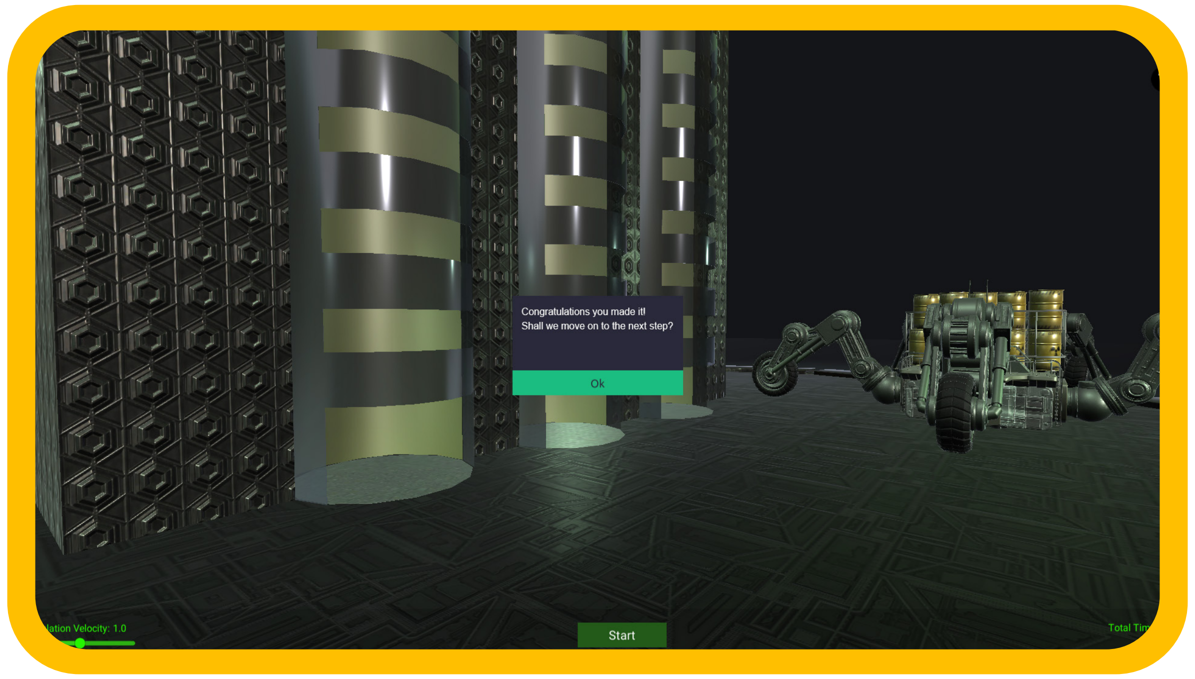
Let's study a little bit about GA before we continue. For that, you can take a look at the video below and, if you want to complement, see also the links:

Video 1: https://www.youtube.com/watch?v=FYF6IS_BHKA

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

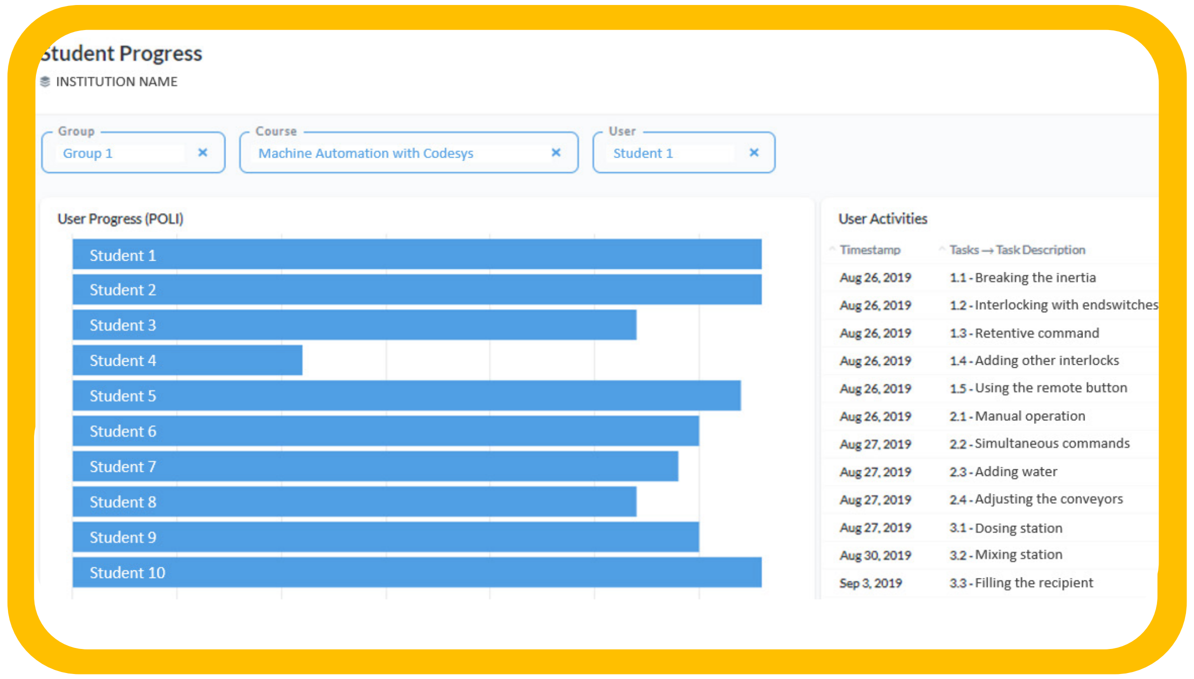
Link 9: https://www.nce.ufrj.br/GINAPE/VIDA/alggenet.htm

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



Congratulations you made it!
Shall we move on to the next step?

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



tudent Progress

INSTITUTION NAME

Group
Group 1 ×

Course
Machine Automation with Codesys ×

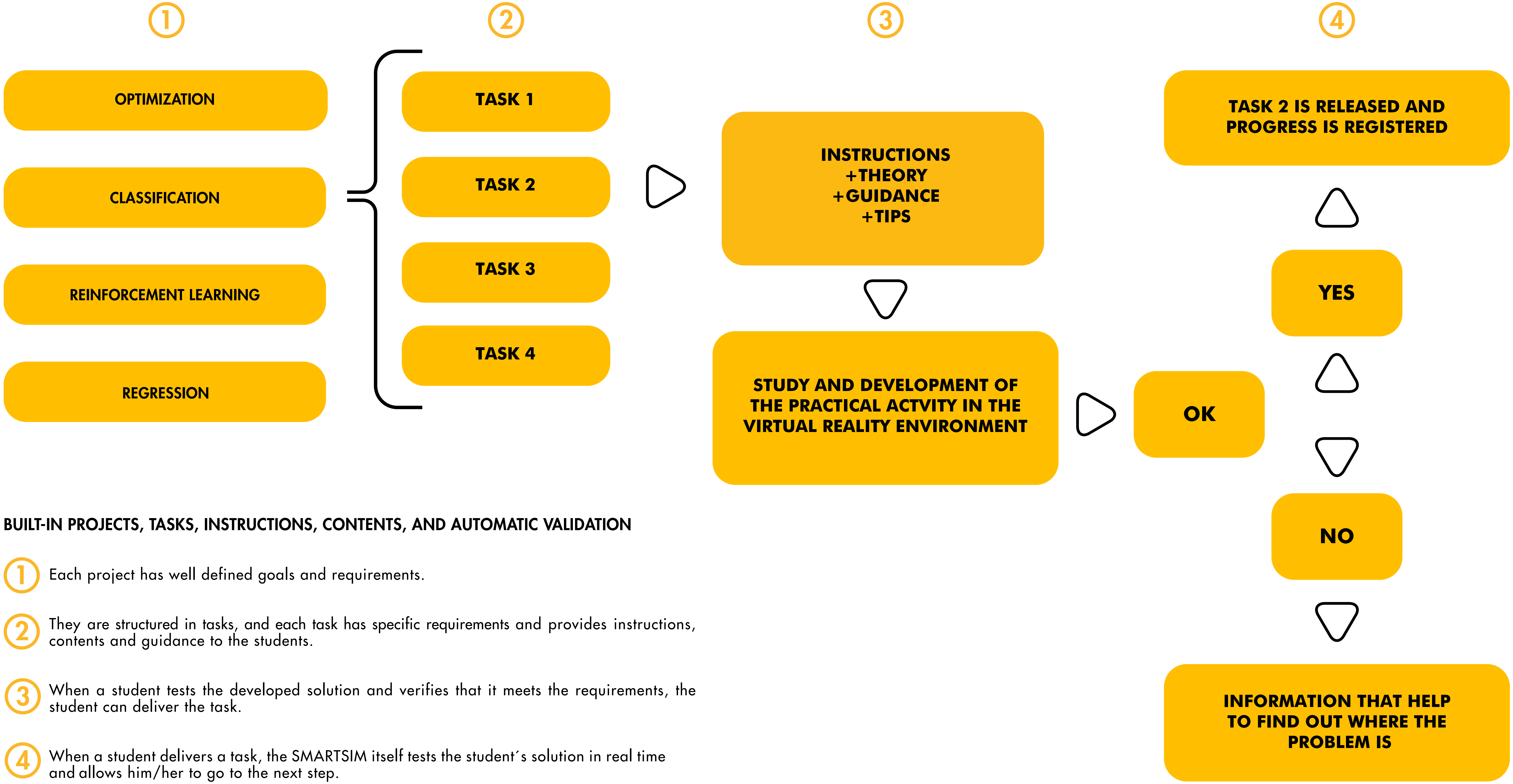
User
Student 1 ×

User Progress (POLI)

User Activities

Timestamp	Tasks → Task Description
Aug 26, 2019	1.1 - Breaking the inertia
Aug 26, 2019	1.2 - Interlocking with endswitches
Aug 26, 2019	1.3 - Retentive command
Aug 26, 2019	1.4 - Adding other interlocks
Aug 26, 2019	1.5 - Using the remote button
Aug 26, 2019	2.1 - Manual operation
Aug 27, 2019	2.2 - Simultaneous commands
Aug 27, 2019	2.3 - Adding water
Aug 27, 2019	2.4 - Adjusting the conveyors
Aug 27, 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?


1
OPTIMIZATION
CLASSIFICATION
REINFORCEMENT LEARNING
REGRESSION
2
TASK 1
TASK 2
TASK 3
TASK 4
3
INSTRUCTIONS +THEORY +GUIDANCE +TIPS
STUDY AND DEVELOPMENT OF THE PRACTICAL ACTVITY IN THE VIRTUAL REALITY ENVIRONMENT
OK
4
TASK 2 IS RELEASED AND PROGRESS IS REGISTERED
YES
NO
INFORMATION THAT HELP TO FIND OUT WHERE THE PROBLEM IS

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

	MINIMUM REQUIREMENTS	RECOMMENDED REQUIREMENTS
OPERATIONAL SYSTEM	64-BIT WINDOWS 10	64-BIT WINDOWS 10 PRO
DIRECTX VERSION	DIRECTX 11	DIRECTX 12
PROCESSOR	INTEL i5 9400F OR AMD RYZEN 5 3600	INTEL i7 9700 OR AMD RYZEN 7 3700X
MEMORY	8GB	16 GB
GHRAPHIC CARD		NVIDIA GTX 1050 TI 4GB OR RX 550 4GB
STORAGE	HDD (1GB)	HDD (1GB)