

# **MACHINE LEARNING SYSTEM DEVELOPMENT IN A VIRTUAL ENVIRONMENT FOR INDUSTRIAL ROBOT TRAINING**

Artem Grukhal  
School of Engineering  
Tallinn University of Technology



# **INTRODUCTION TO PROJECT**

# INTRODUCTION TO PROJECT

This project consists of several practical parts which are subject to research as well:

- Finding solutions on manipulating the robot arm with it's digital twin using coordinate systems (as opposed to changing joints' angle values).
- Implementing machine learning in Virtual Environment.
- Transferring machine learning knowledge from digital twin to a real robot.

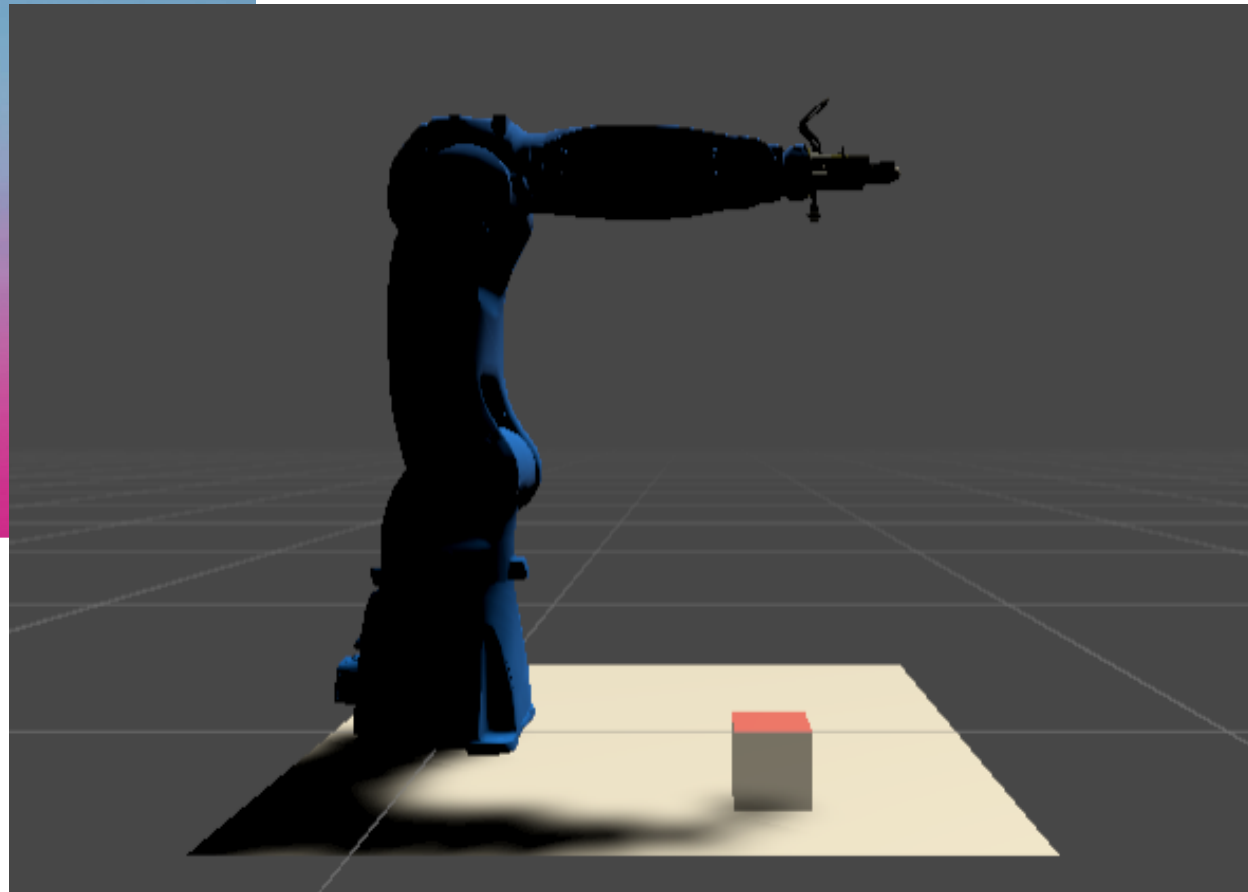
All of them require prior knowledge, practical implementation and posterior analysis in order to consider the project be successful.

In this presentation, I will introduce you to basic features and what can be expected.

# INTRODUCTION TO PROJECT: TOOL STACK USED

So far following tools are used in this project:

- Core:
  - Unity
  - C# programming language
- Coordinate system manipulation:
  - BIO IK asset
  - CCDIK asset (discussion only)
- Machine Learning injection:
  - ML Agents asset
  - Command Line
  - TensorBoard



Here we can see the Motoman GP8 Robot on a test platform with a test cube side view.



# **COORDINATE SYSTEMS IN DIGITAL ENVIRONMENT**

# COORDINATE SYSTEMS IN DIGITAL ENVIRONMENT

The main problem which had to be solve is to find a way to set coordinates for a robot to reach instead of setting angles only.

This can be achieved with Inverse Kinematics for robot manipulators. The robot has 6 links, including the Gripper, so there is no point in reinventing the wheel and writing our own IK solver. There were two alternatives to consider:

- CCD IK:

This asset works great with robots with simple geometry, however, there is a major drawback in not being able to manipulate the final orientation which the Gripper must have.

- Bio IK:

This asset is rich in functionality, it is often used for humanoid models in virtual environment, however it is a complex system with many settings that must be considered.

## COORDINATE SYSTEMS IN DIGITAL ENVIRONMENT: BIO IK

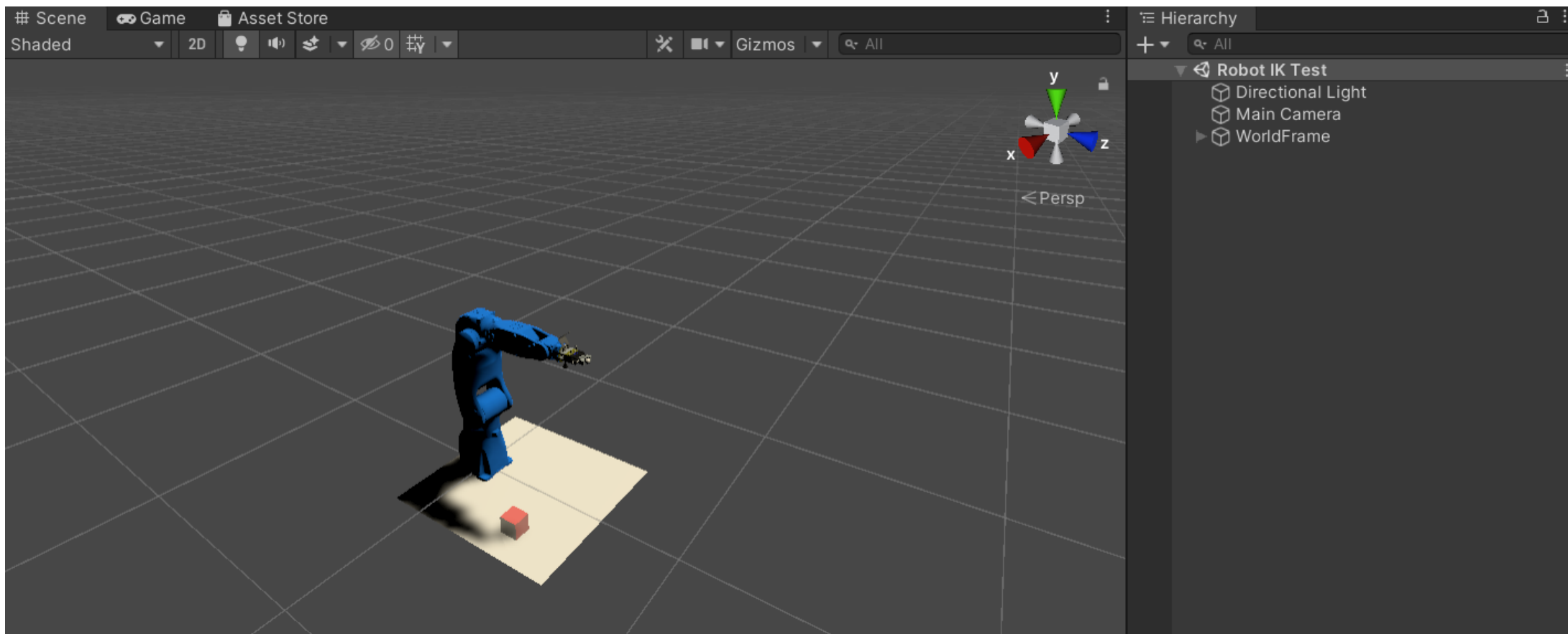
It was decided to choose Bio IK asset, due to functionality and important features like setting realistic speed of manipulator, setting orientation for the robot arm and more precise control over movement of joints.

The orientation is a very important factor to consider, because this would define the way objects will be fixed with Gripper and further performance of such system in general.

Realistic speed is a better way to simulate behavior of a real machine which will also be used for machine learning observations.

Finally, it gives possibility to read the values of angles, which will ideally be sent to a real machine for giving instructions from digital environment straight away.





Here we can see the Motoman GP8 Robot on a test platform with a test cube. Under the WorldFrame directory there are ReferencePoint, Robot model and other objects. It is worth noting, that in this project ReferencePoint is used as a target for the robot's Gripper tip and robot is manipulated by changing the coordinates of a ReferencePoint object.



# **MACHINE LEARNING IN DIGITAL ENVIRONMENT**

# MACHINE LEARNING IN DIGITAL ENVIRONMENT

Main ideas present for this part of the project are more theoretical, yet practical nuances also present.

The idea is to make a virtual robot learn and transfer the information to the real one.

There are several machine learning tasks:

- Choose best possible route for robot to reach the final point
- Sort objects based on color

Both rely on ML Agents asset, which will be described on the next slide.

## MACHINE LEARNING IN DIGITAL ENVIRONMENT: ML AGENTS

ML Agents is an asset for Unity Engine to provide AI capabilities to objects. It is still in beta, however, it can already be used for projects.

It includes several algorithms such as Proximal Policy Optimization and Soft Actor Critic. It is also possible to use Imitation Learning for making objects behave as if a person was using them.

This tool is a mix between Unity specific scripts and Python scripts, and training is organized outside of Unity Environment. For a training it would be needed to Build a Unity executable and run a Python script on it, including the configuration file.

There is also a possibility to track the results of the training live through TensorBoard.

## SECTION HEADER ON TWO LINES IF NECESSARY

This is a depiction of how the model is trained. At the bottom of the picture there is a description with the current performance of the model.



```
2020-04-08 16:45:33 INFO [trainer_controller.py:167] Hyperparameters for the PPOTrainer of brain TestSetup:
  trainer:      ppo
  batch_size:   100
  beta:        0.005
  buffer_size:  10280
  epsilon:      0.2
  hidden_units: 128
  lambda:       0.95
  learning_rate: 0.0003
  learning_rate_schedule: linear
  max_steps:    5.0e5
  memory_size:  128
  normalize:    False
  num_epoch:    3
  num_layers:   2
  time_horizon: 64
  sequence_length: 64
  summary_freq: 10000
  use_recurrent: False
  vis_encode_type: simple
  reward_signals:
    extrinsic:
      strength: 1.0
      gamma: 0.99
  summary_path: robot-arm-test_TestSetup
  model_path:   ./models/robot-arm-test/TestSetup
  keep_checkpoints: 5
2020-04-08 16:45:33.424647: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
2020-04-08 16:46:22 INFO [trainer.py:213] robot-arm-test: TestSetup: Step: 10000. Time Elapsed: 48.722 s Mean Reward: -0.629. Std of Reward: 0.721. Training.
2020-04-08 16:47:10 INFO [trainer.py:213] robot-arm-test: TestSetup: Step: 20000. Time Elapsed: 97.027 s Mean Reward: -0.591. Std of Reward: 0.837. Training.
```



## **PURPOSE OF THE PROJECT**

## PURPOSE OF THE PROJECT

This project is aimed to improve the already existing digital twin, with the possibility to open new fields for research or work. Another aim is to assess the credibility of such system, compare with already existing solutions and determine possible use cases.

One of the most important considerations is to provide ability to work with machinery remotely, which will be crucial for development of manufacture. As we can see from the situation with COVID-19, such systems would help manufacture sites a lot, by enabling people to work remotely from home with zero to no actual presence on site.

This consideration gives additional security when working with dangerous machinery, better control of different machines (i.e. to process different products from different machines using the same digital twin) and potential profits to companies, which embed this technology into their development process.

The logo consists of the words "TAL" and "TECH" stacked vertically in a bold, white, sans-serif font. The background is a horizontal gradient transitioning from light blue on the left to magenta on the right. A vertical magenta bar is on the far left, and a white stepped shape is on the left side of the magenta area.

**TAL  
TECH**

**TALLINN UNIVERSITY OF TECHNOLOGY**

**TALTECH.EE/EN**