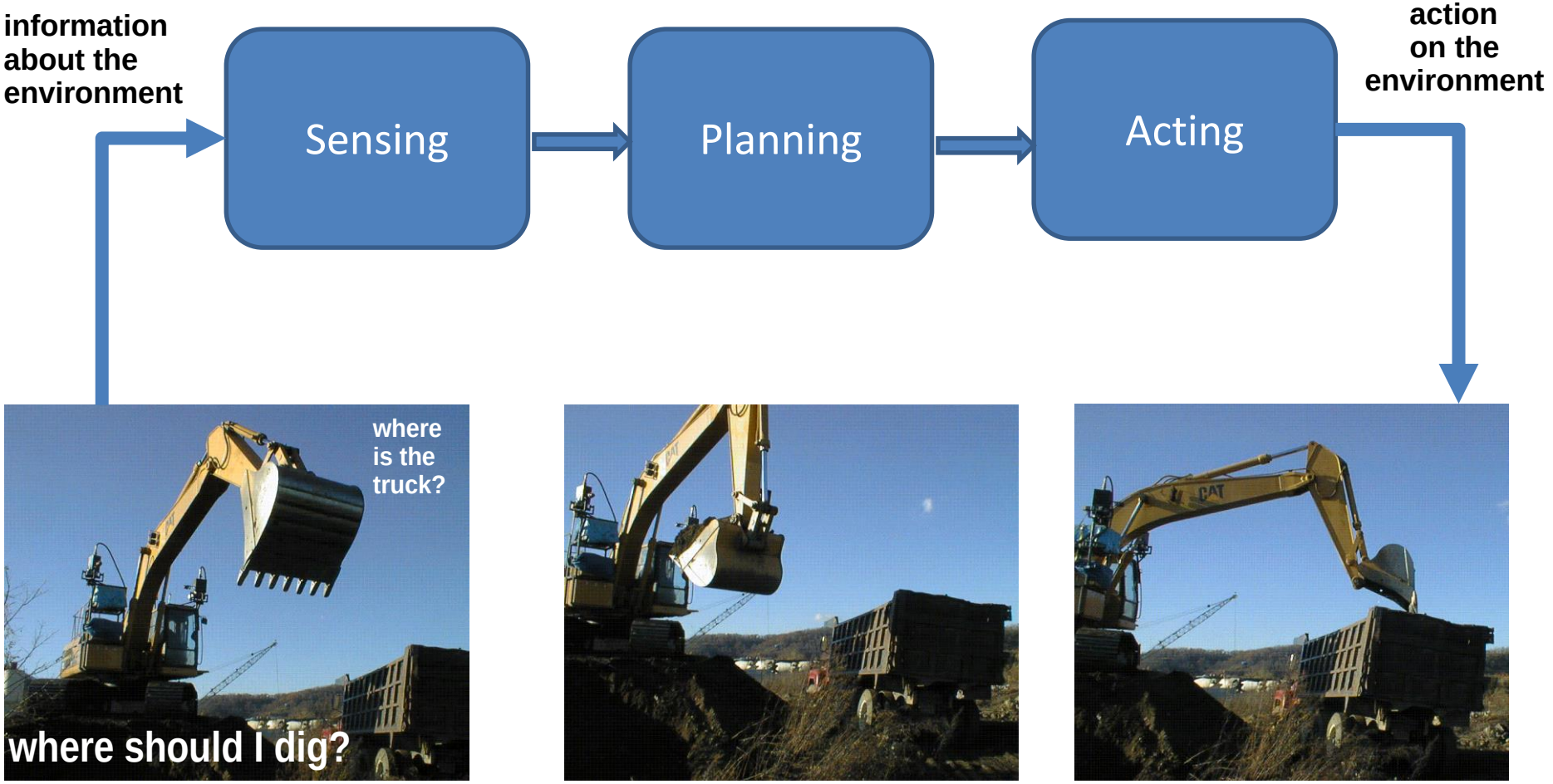
Midterm Project
- Virtual Truck-

10/15

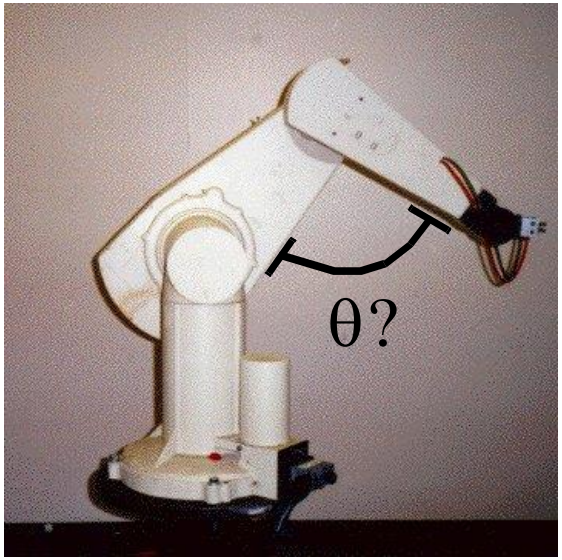
KAIST
NMAIL
Neuro-Machine Augmented Intelligence Lab

What makes a machine a Robot?



Why do robots need sensors?

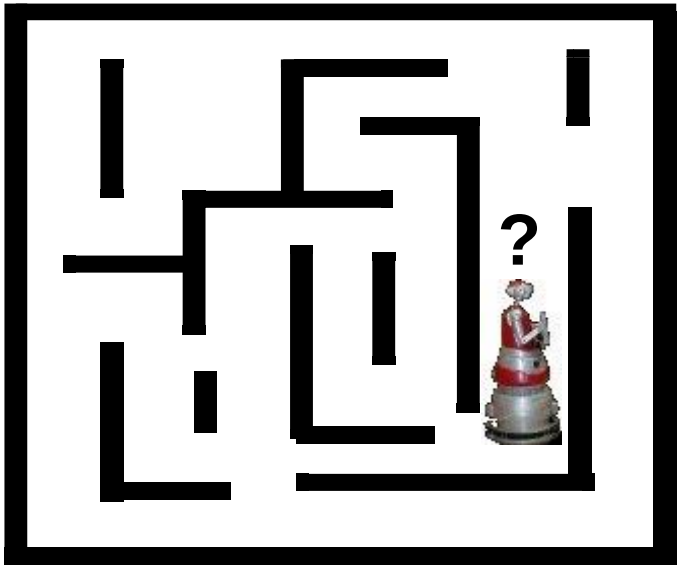
What is the angle of my arm?



internal information

Why do robots need sensors?

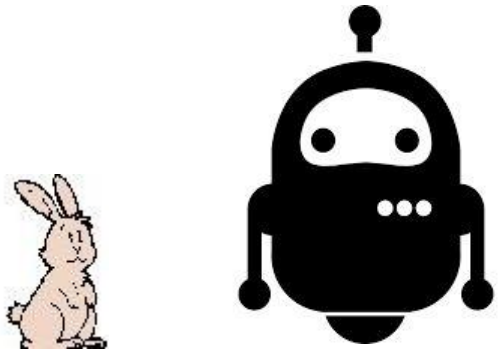
Where am I?



localization

Why do robots need sensors?

Will I hit anything?



obstacle detection

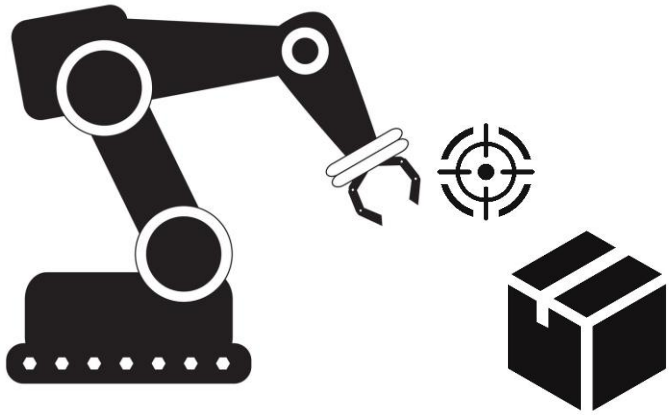
Sensing for specific tasks

Where is the cropline?

Autonomous harvesting

Sensing for specific tasks

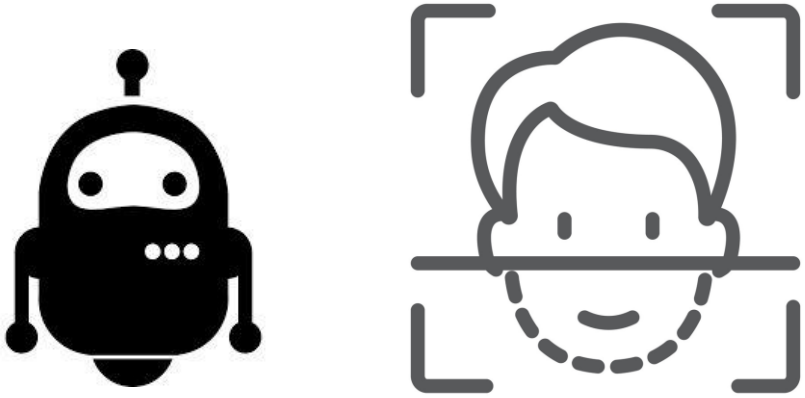
Where is the target object.



Autonomous material handling

Sensing for specific tasks

Where is the face?

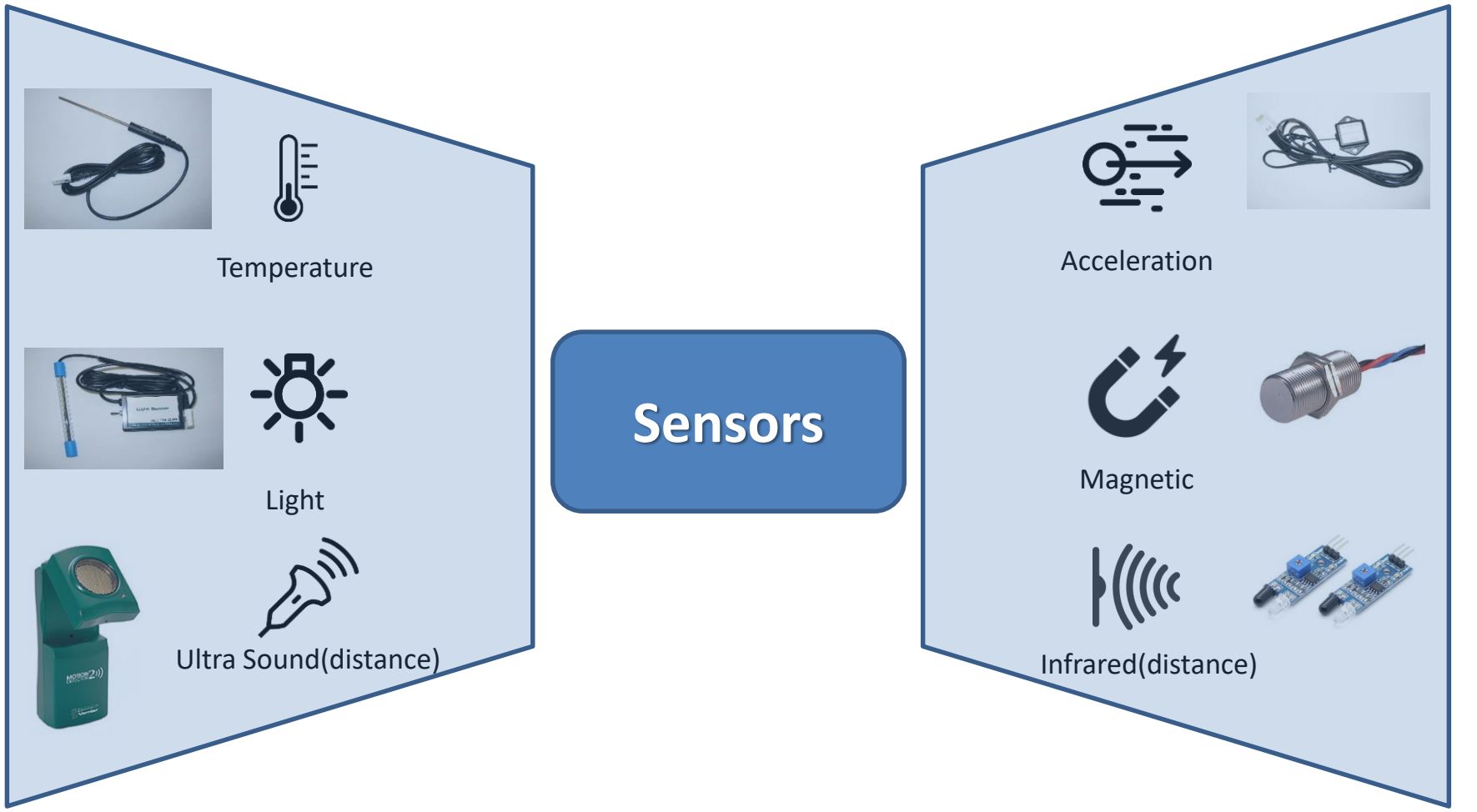


Face detection & tracking

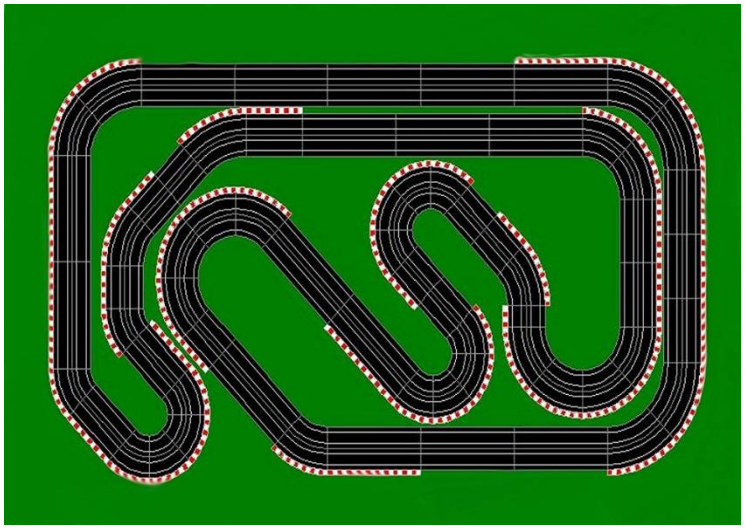
Detectable Phenomenon

Stimulus	Quantity
Acoustic	Wave (amplitude, phase, polarization), Spectrum, Wave Velocity
Biological & Chemical	Fluid Concentrations (Gas or Liquid)
Electric	Charge, Voltage, Current, Electric Field (amplitude, phase, polarization), Conductivity, Permittivity
Magnetic	Magnetic Field (amplitude, phase, polarization), Flux, Permeability
Optical	Refractive Index, Reflectivity, Absorption
Thermal	Temperature, Flux, Specific Heat, Thermal Conductivity
Mechanical	Position, Velocity, Acceleration, Force, Strain, Stress, Pressure, Torque

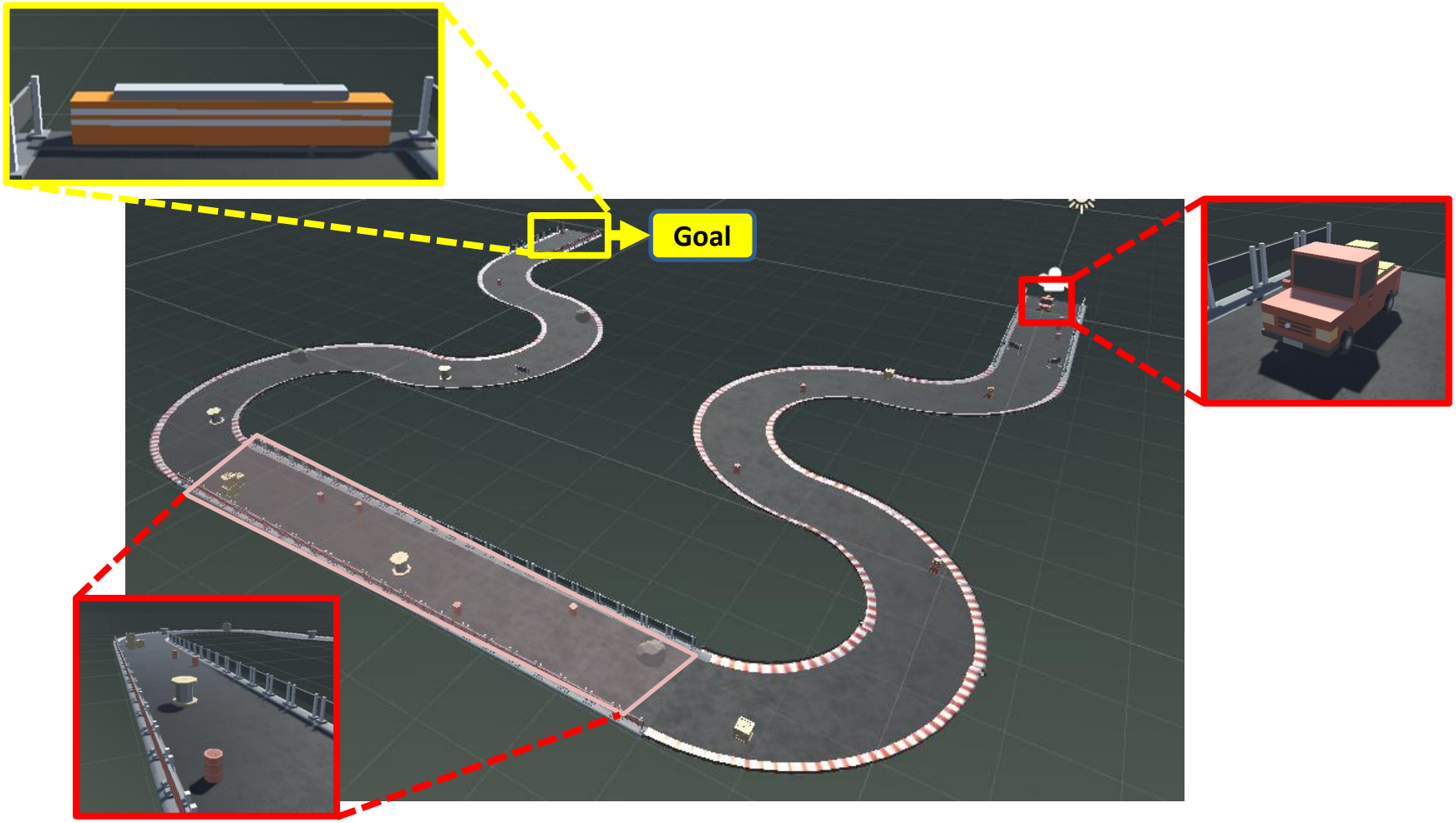
Sensor Types



Introduction of the midterm project



Introduction of the midterm project



Introduction of the midterm project

IR (Infrared) Sensor

Motor torque

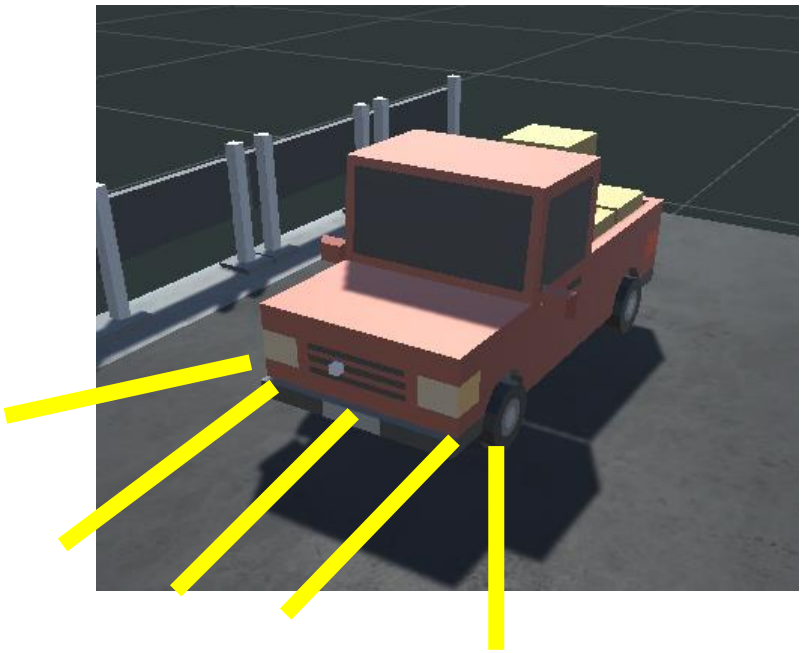
Truck Steering Wheel

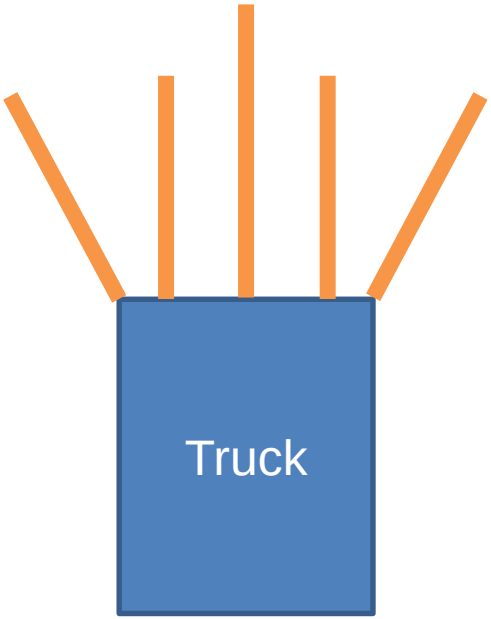
KAIST

NMAIL

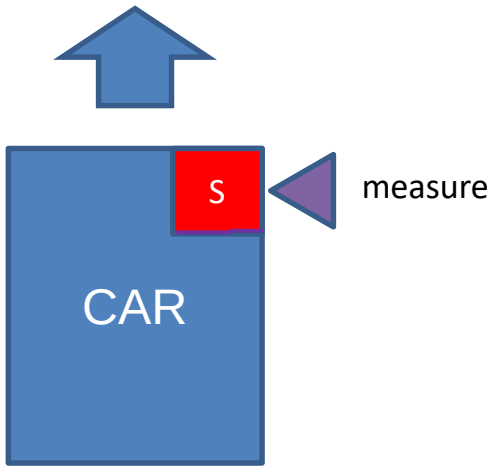
Neuro-Machine Augmented Intelligence Lab

Introduction of the midterm project

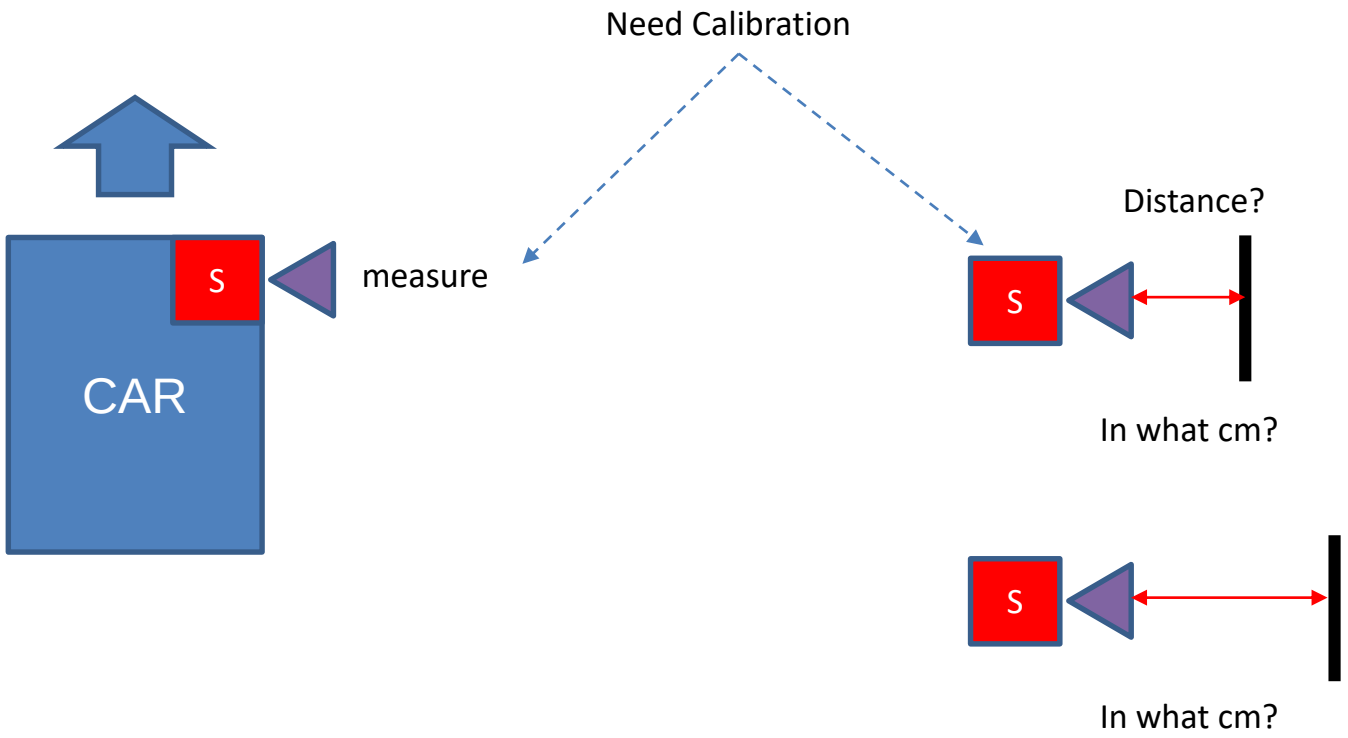




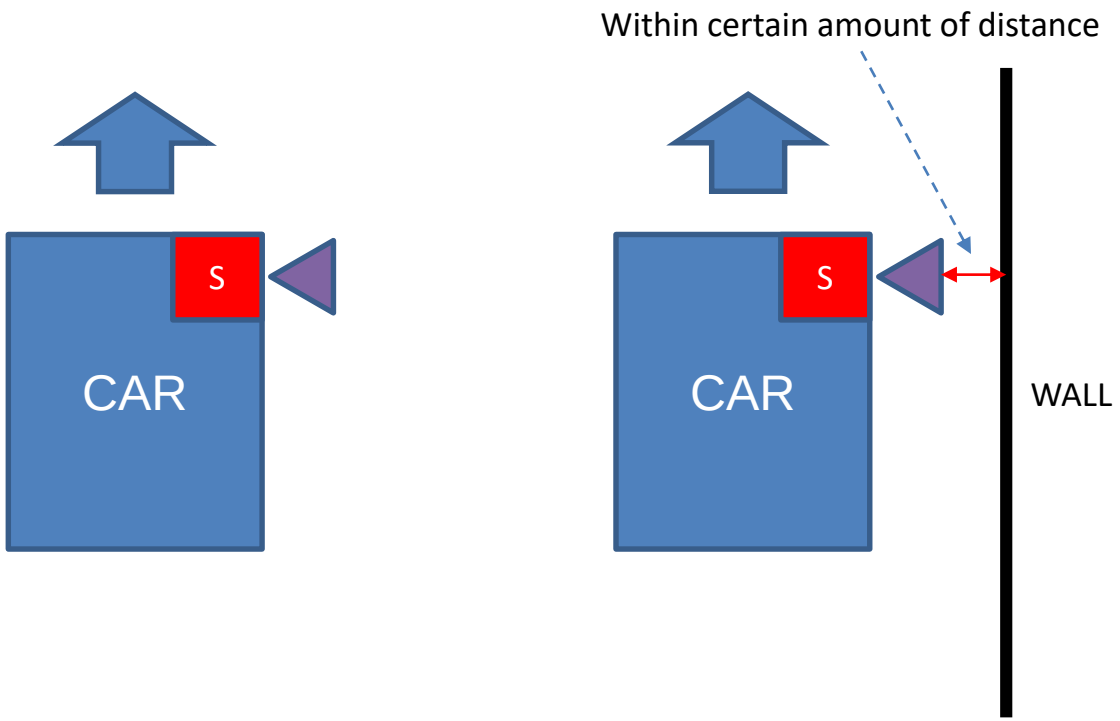
Simple Logic makes a car traverse



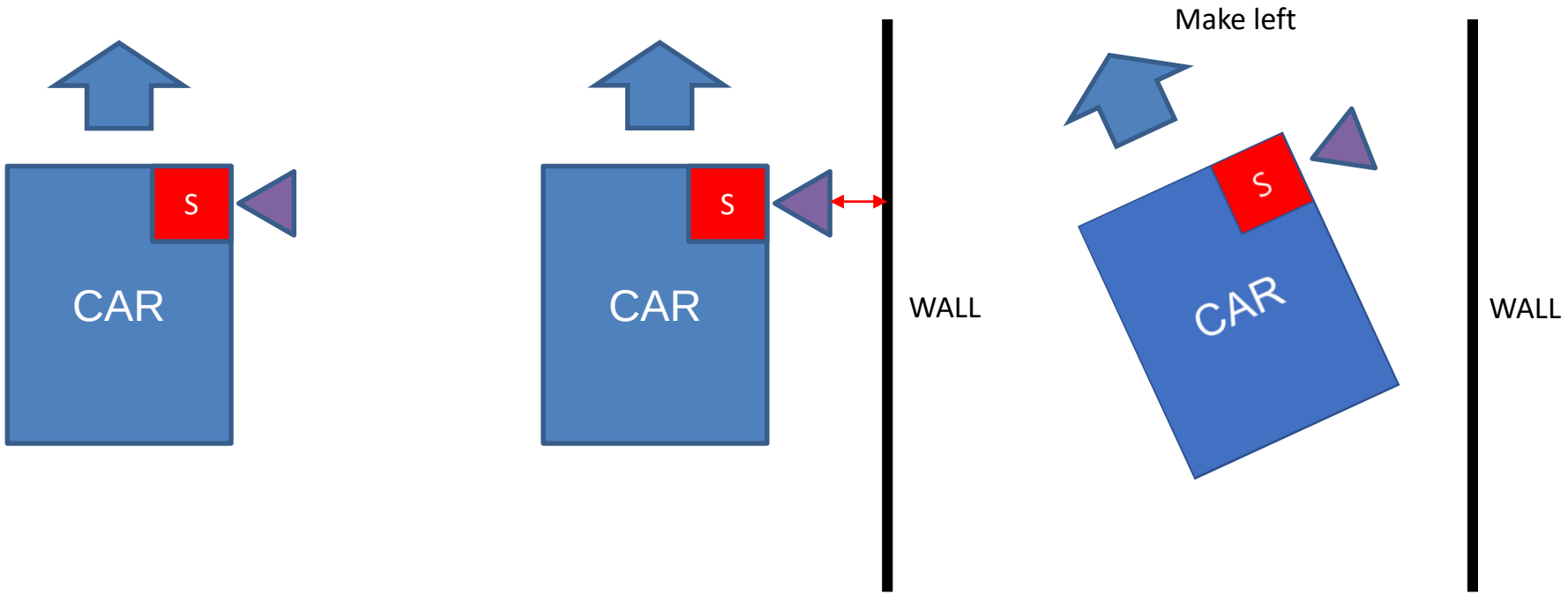
Simple Logic makes a car traverse



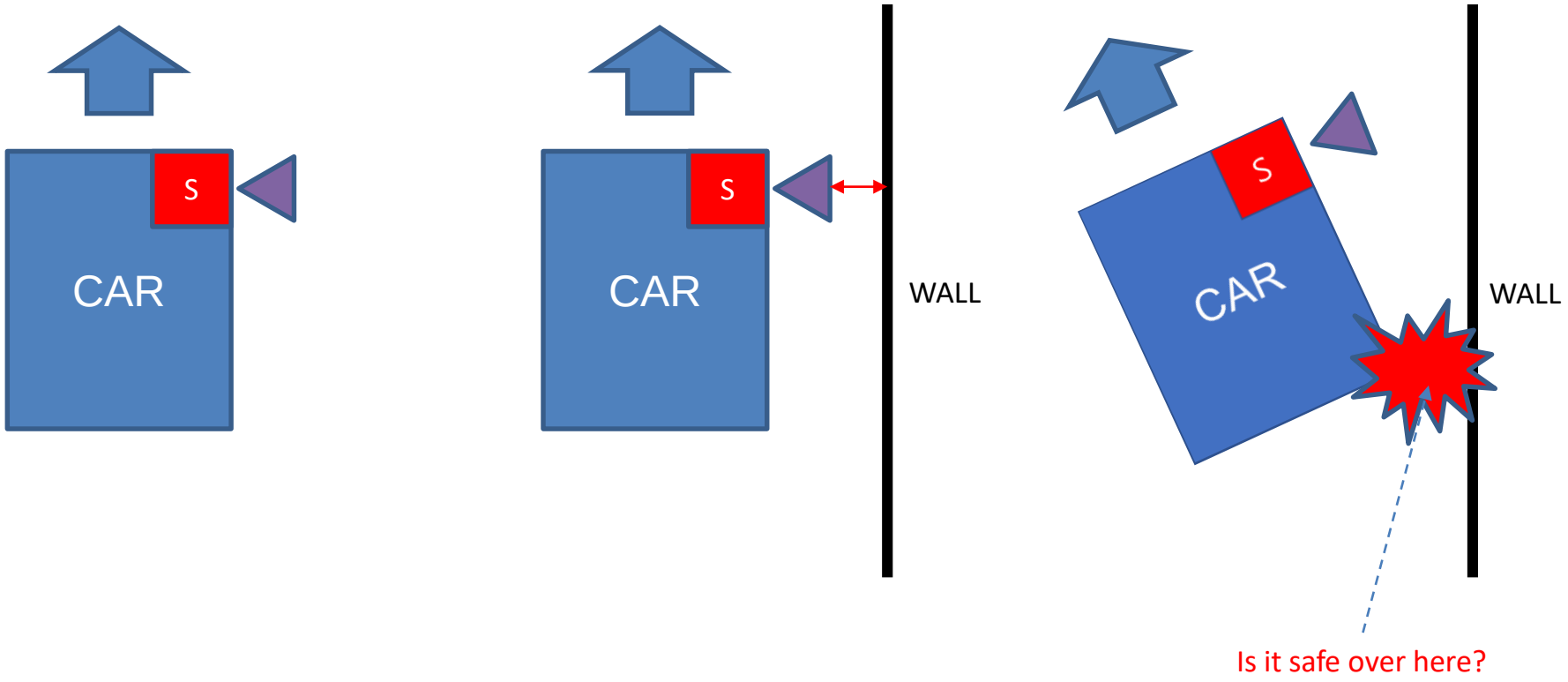
Simple Logic makes a car traverse



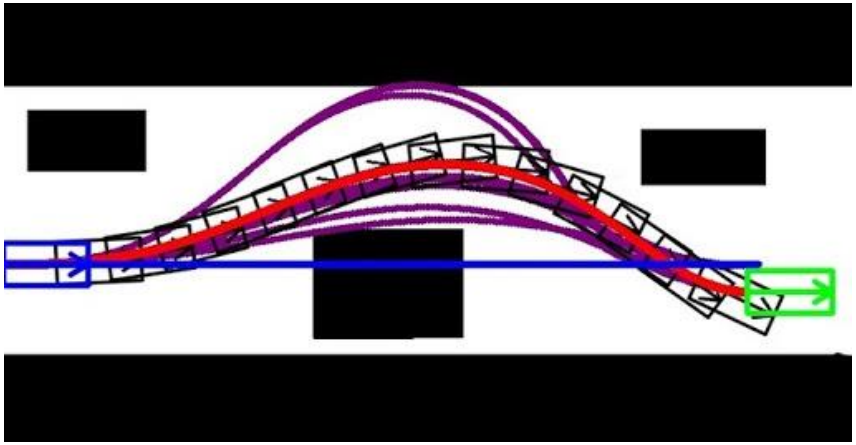
Simple Logic makes a car traverse



Simple Logic makes a car traverse



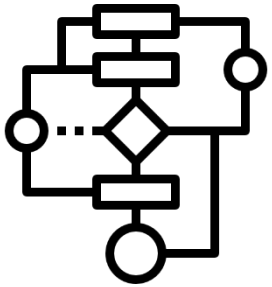
Simple Logic makes a car traverse



Introduction of the midterm project

- **Project Goal**

1. Make your team's universal algorithm for Truck to reach the goal point of the track.
2. Optimize the algorithm to achieve better time attack record.
3. We will open competition in online with each teams algorithm.(11/5 scheduled)



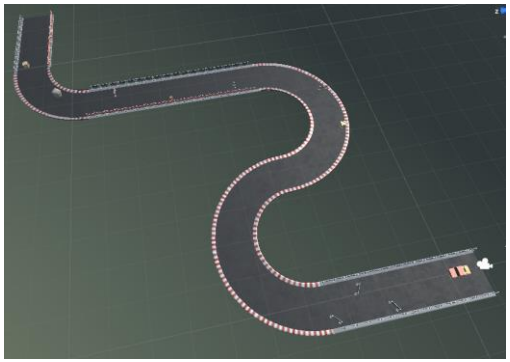
Algorithm

Time attack

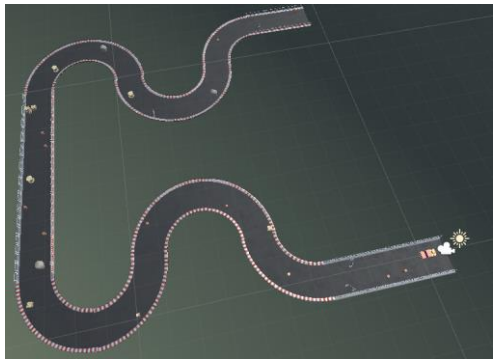
Competition

Introduction of the midterm project

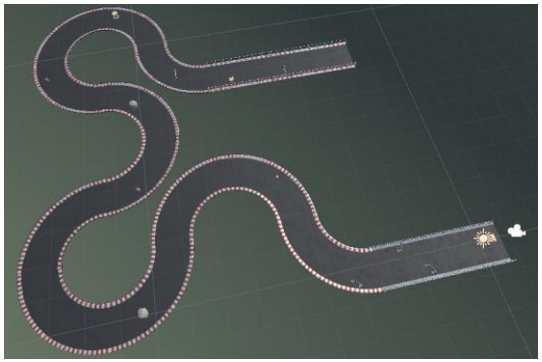
- Three practice maps will be provided to make your truck's obstacle avoidance algorithm



Track 1



Track 2



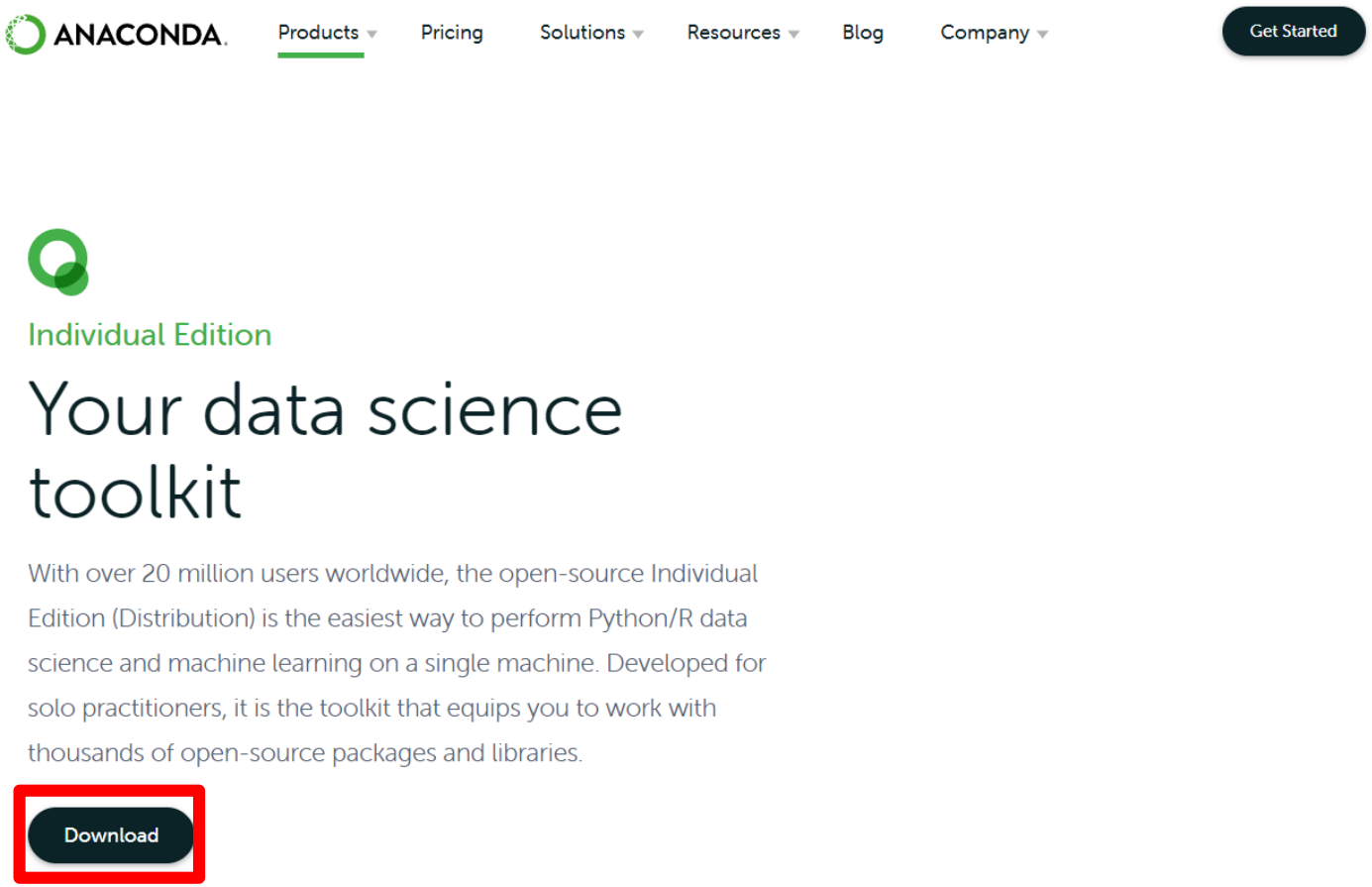
Track 3

- Competition track will be revealed at 11/5.

Preparation

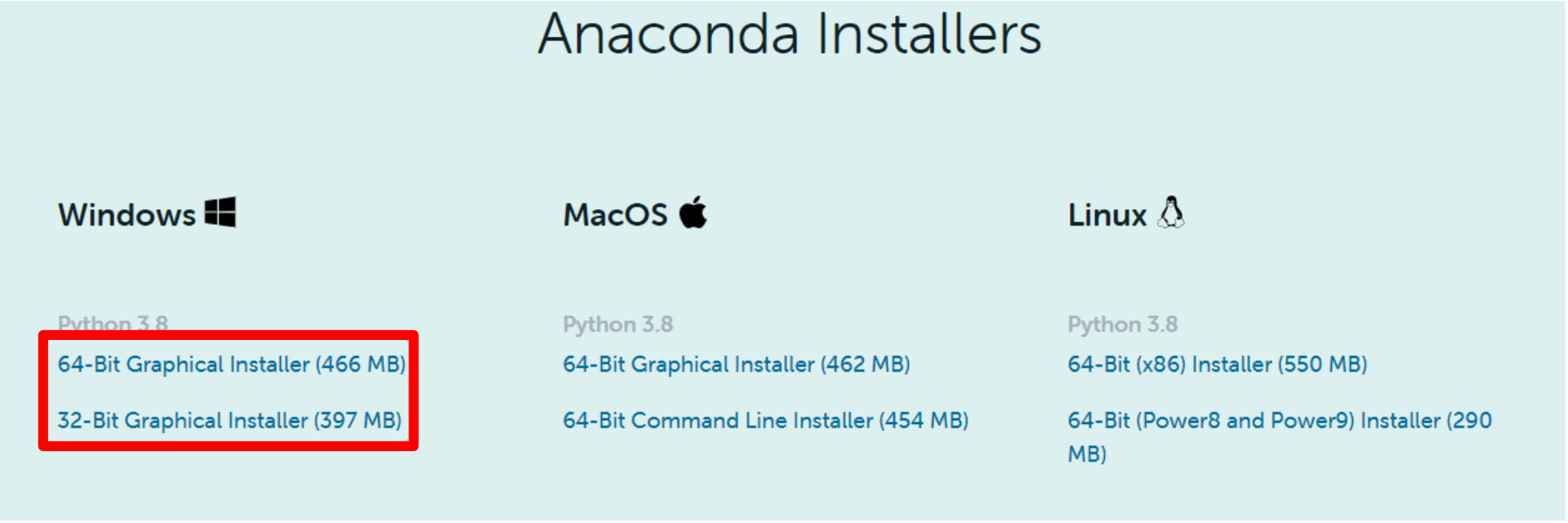
Installation Anaconda (Python)

- https://www.anaconda.com/products/individual



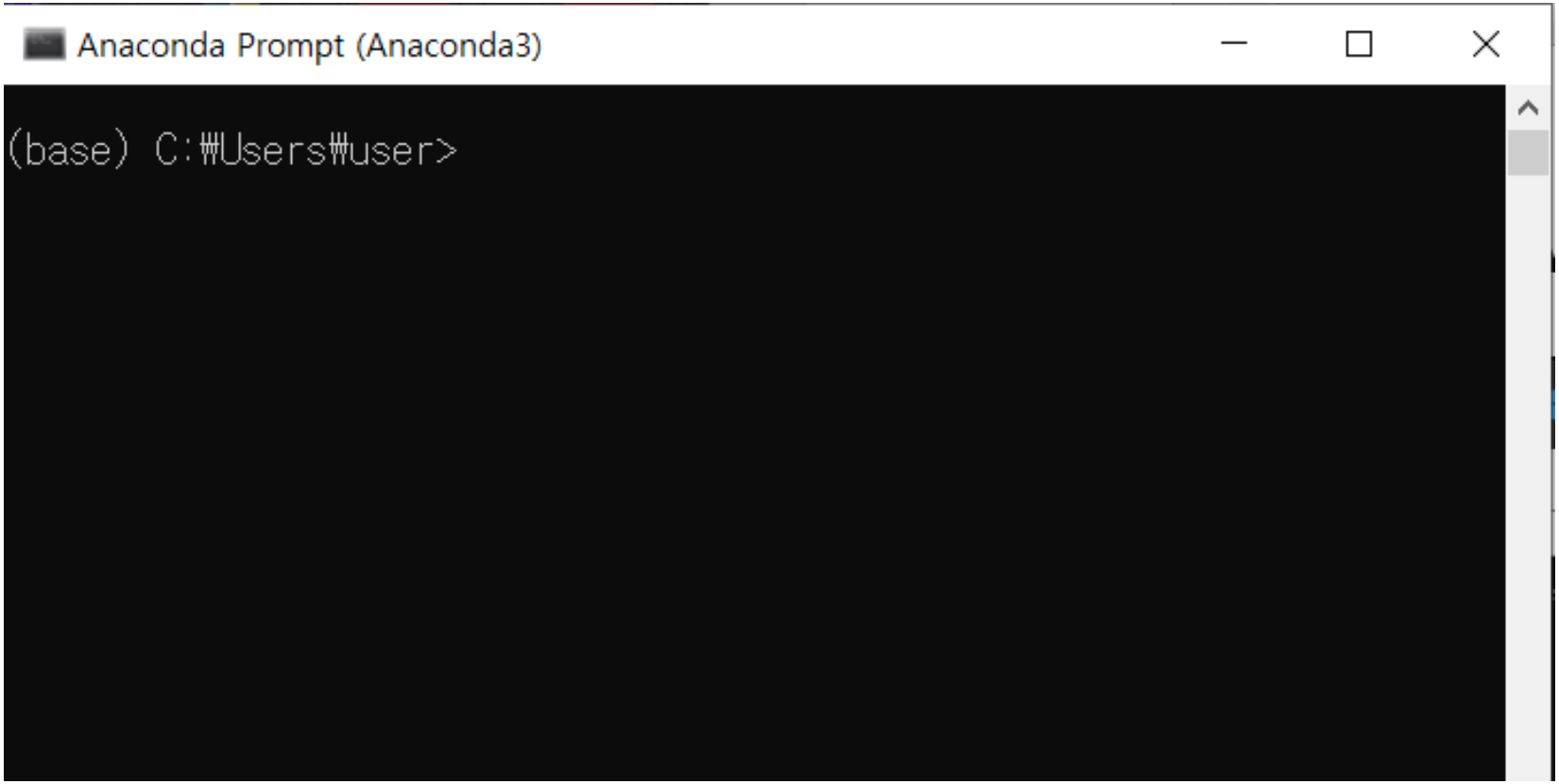
Installation Anaconda (Python)

- Download Python 3.8 window version depend on your computer. (only Window version is available for our project)



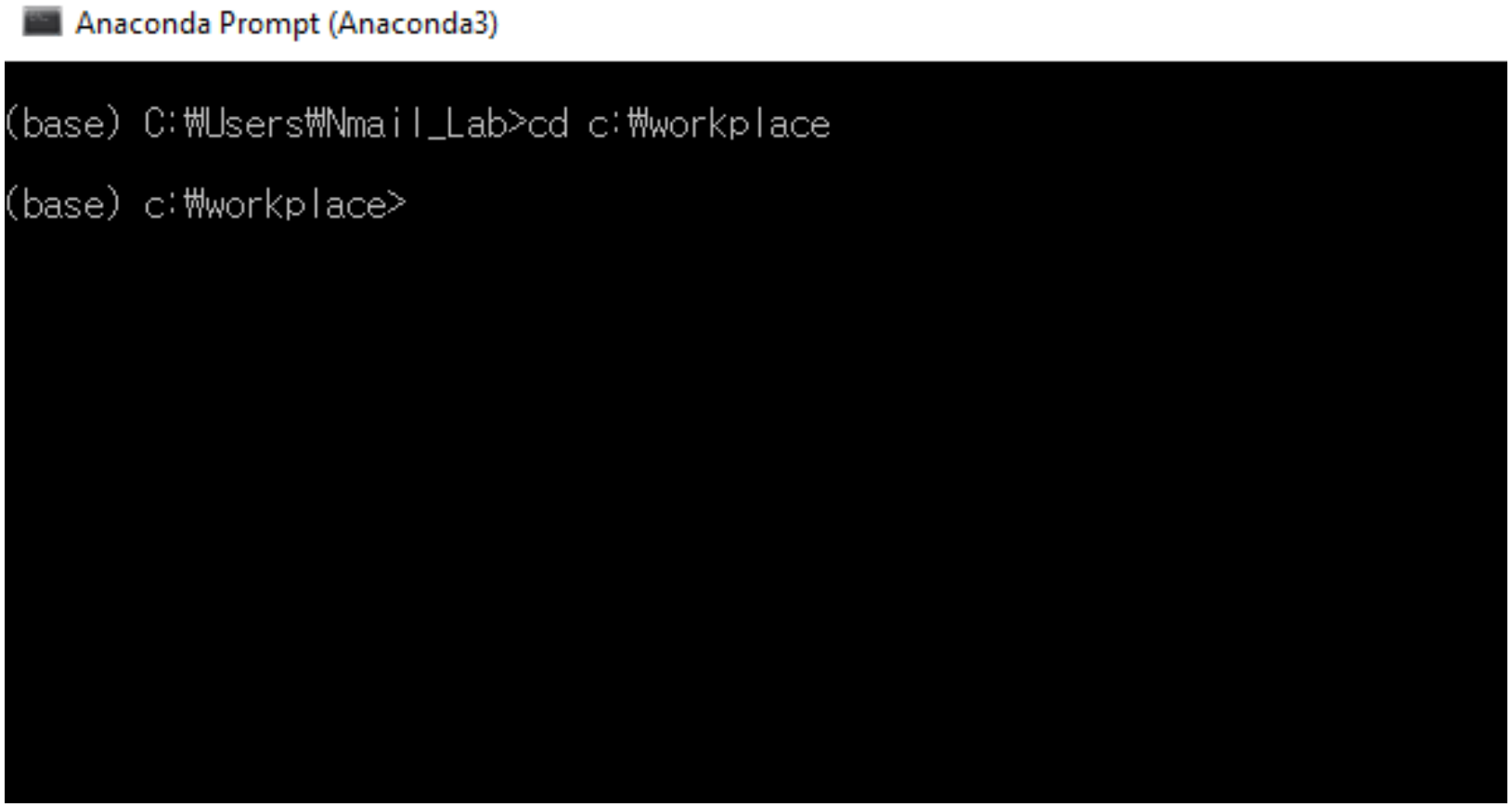
Anaconda Prompt

 → Anaconda3 (64-bit) → Anaconda Prompt (Anaconda 3)



Create Virtual environment

C:\Users\user> cd c:\(your folder)



Create Virtual environment

C:\(your folder)> **conda create –n (env name) python=3.6**
C:\(your folder)> **conda activate (env name)**

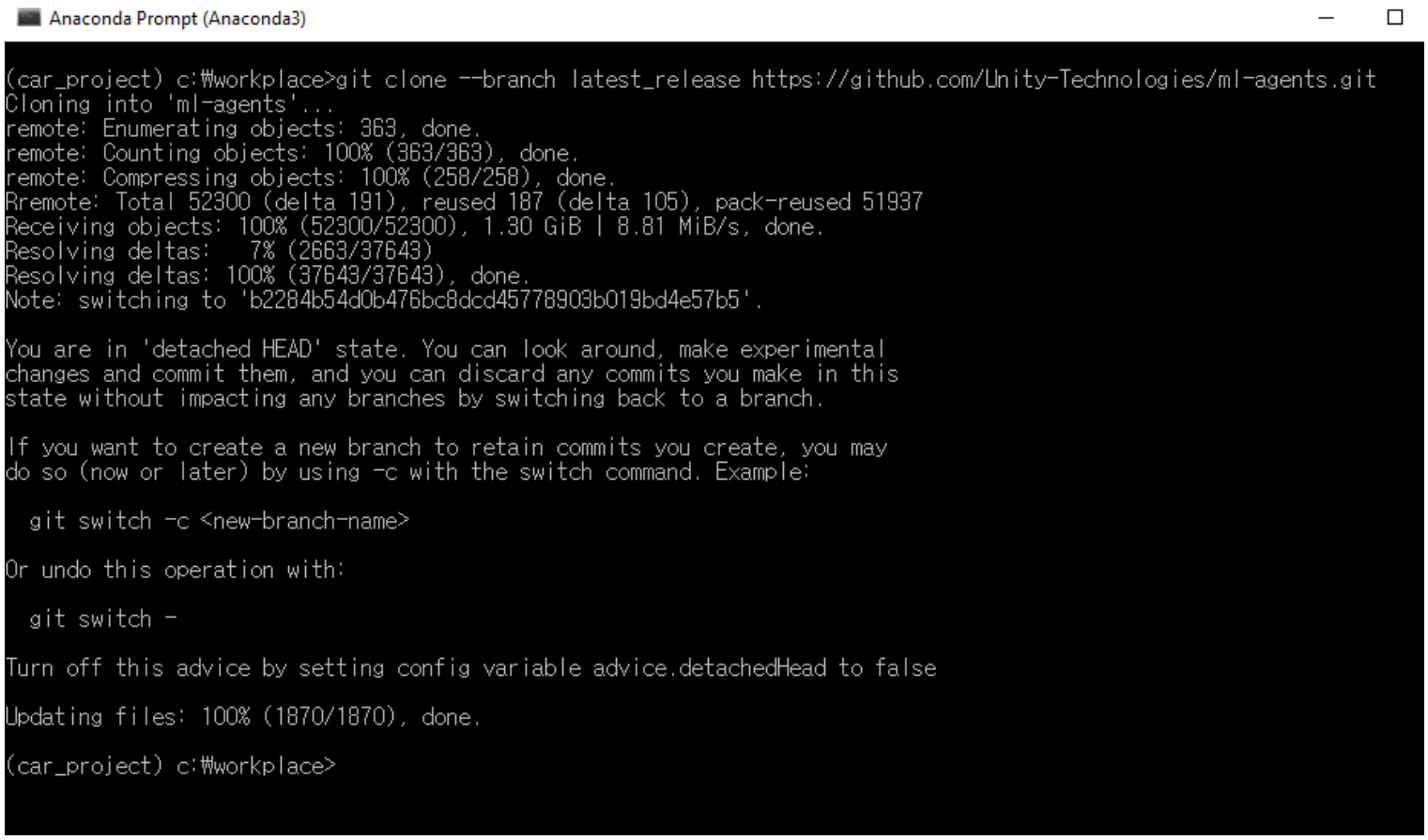
```
# To activate this environment, use
#
#     $ conda activate car_project
#
# To deactivate an active environment, use
#
#     $ conda deactivate


(base) c:₩workplace>conda activate car_project

(car_project) c:₩workplace>
```

Install Ml-Agents library

C:\(your folder)>**git clone --branch latest_release https://github.com/Unity-Technologies/ml-agents.git**



Install Ml-Agents library

C:\(your folder)> **pip install -e ./ml-agents/ml-agents-envs/**

```
(car_project) c:₩workplace>pip install -e ./ml-agents/ml-agents-envs/
Obtaining file:///C:/workplace/ml-agents/ml-agents-envs
Collecting cloudpickle
  Using cached cloudpickle-1.6.0-py3-none-any.whl (23 kB)
Collecting grpcio>=1.11.0
  Using cached grpcio-1.32.0-cp36-cp36m-win_amd64.whl (2.6 MB)
Collecting numpy<1.19.0,>=1.14.1
  Downloading numpy-1.18.5-cp36-cp36m-win_amd64.whl (12.7 MB)
     |████████████████████████████████| 12.7 MB 6.4 MB/s
Collecting Pillow>=4.2.1
  Downloading Pillow-7.2.0-cp36-cp36m-win_amd64.whl (2.0 MB)
     |████████████████████████████████| 2.0 MB 6.4 MB/s
Collecting protobuf>=3.6
  Downloading protobuf-3.13.0-cp36-cp36m-win_amd64.whl (1.1 MB)
     |████████████████████████████████| 1.1 MB 6.4 MB/s
Collecting pyyaml>=3.1.0
  Using cached PyYAML-5.3.1-cp36-cp36m-win_amd64.whl (215 kB)
Collecting six>=1.5.2
  Using cached six-1.15.0-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: setuptools in c:₩users₩nmail_lab₩anaconda3₩envs₩car_project₩lib₩site-packages (from proto
buf>=3.6->mlagents-envs==0.20.0) (50.3.0.post20201006)
Installing collected packages: cloudpickle, six, grpcio, numpy, Pillow, protobuf, pyyaml, mlagents-envs
    Running setup.py develop for mlagents-envs
Successfully installed Pillow-7.2.0 cloudpickle-1.6.0 grpcio-1.32.0 mlagents-envs numpy-1.18.5 protobuf-3.13.0 pyyaml-5
3.1 six-1.15.0

(car_project) c:₩workplace>
```

KAIST NMAIL Neuro-Machine Augmented Intelligence Lab

Install Ml-Agents library

C:\(your folder)> **pip install -e ./ml-agents/ml-agents/**

```
  Using cached importlib_metadata-2.0.0-py2.py3-none-any.whl (31 kB)
Collecting pyasn1>=0.1.3
  Using cached pyasn1-0.4.8-py2.py3-none-any.whl (77 kB)
Collecting oauthlib>=3.0.0
  Using cached oauthlib-3.1.0-py2.py3-none-any.whl (147 kB)
Collecting zipp>=0.5
  Downloading zipp-3.3.0-py3-none-any.whl (5.3 kB)
Installing collected packages: h5py, opt-einsum, google-pasta, absl-py, wrapt, werkzeug, pyasn1, rsa, cachetools, pyasn
-modules, google-auth, idna, chardet, urllib3, requests, oauthlib, requests-oauthlib, google-auth-oauthlib, zipp, impor
lib-metadata, markdown, tensorboard-plugin-wit, tensorboard, tensorflow-estimator, keras-preprocessing, termcolor, astu
parse, gast, tensorflow, attrs, cattrs, pywin32, pypiwin32, mlagents
  Running setup.py develop for mlagents
Successfully installed absl-py-0.10.0 astunparse-1.6.3 attrs-20.2.0 cachetools-4.1.1 cattrs-1.0.0 chardet-3.0.4 gast-0.
.3 google-auth-1.22.1 google-auth-oauthlib-0.4.1 google-pasta-0.2.0 h5py-2.10.0 idna-2.10 importlib-metadata-2.0.0 kera
-preprocessing-1.1.2 markdown-3.3.1 mlagents oauthlib-3.1.0 opt-einsum-3.3.0 pyasn1-0.4.8 pyasn1-modules-0.2.8 pypiwin3
-223 pywin32-228 requests-2.24.0 requests-oauthlib-1.3.0 rsa-4.6 tensorboard-2.3.0 tensorboard-plugin-wit-1.7.0 tensorf
ow-2.3.1 tensorflow-estimator-2.3.0 termcolor-1.1.0 urllib3-1.25.10 werkzeug-1.0.1 wrapt-1.12.1 zipp-3.3.0

(car_project) c:₩workplace>
```

KAIST NMAIL Neuro-Machine Augmented Intelligence Lab

Install Ml-Agents library

C:\(your folder)> pip install -e ./ml-agents/gym-unity/

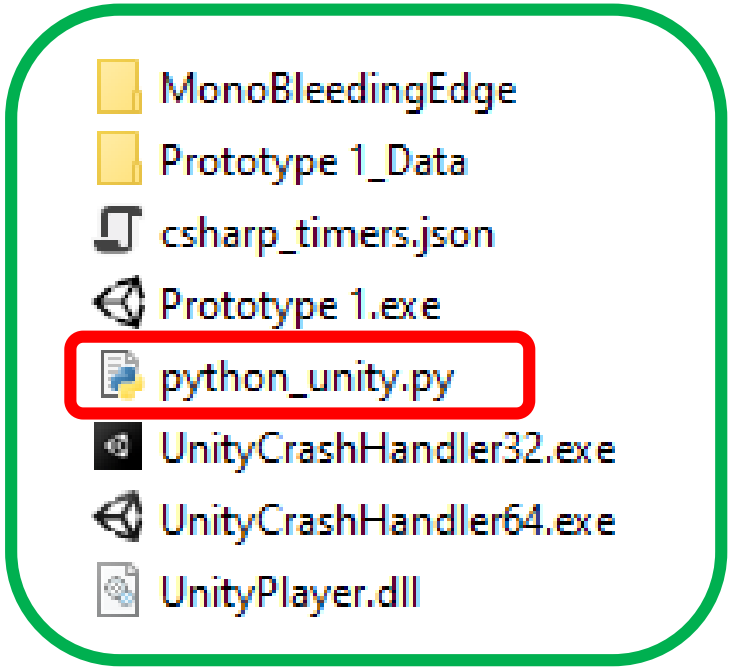
```
Building wheels for collected packages: gym
  Building wheel for gym (setup.py) ... done
  Created wheel for gym: filename=gym-0.17.3-py3-none-any.whl size=1654657 sha256=55385b65bc9996cd86dbd316f
efaf1d787a52b3050e2f58f1c7
  Stored in directory: c:₩users₩nmail_lab₩appdata₩local₩pip₩cache₩wheels₩95₩b0₩62₩af38051b97354eab5b2ff9a5f
Successfully built gym
Installing collected packages: scipy, future, pyglet, gym, gym-unity
  Running setup.py develop for gym-unity
Successfully installed future-0.18.2 gym-0.17.3 gym-unity pyglet-1.5.0 scipy-1.5.2

(car_project) c:₩workplace>
```

KAIST NMAIL Neuro-Machine Augmented Intelligence Lab

Execute program

- There are two parts of template files
  - Unity background part
    - It contains all 3d components to build truck and road track
  - Python code part
    - The main code file which your team has to build the code



Execute program

- When you open the python_unity.py file
  - you should set the unity exe file path(you can write the file name without ‘exe’)in the UnityEnvironment function in line 5.
  - This function makes connection between python and unity.

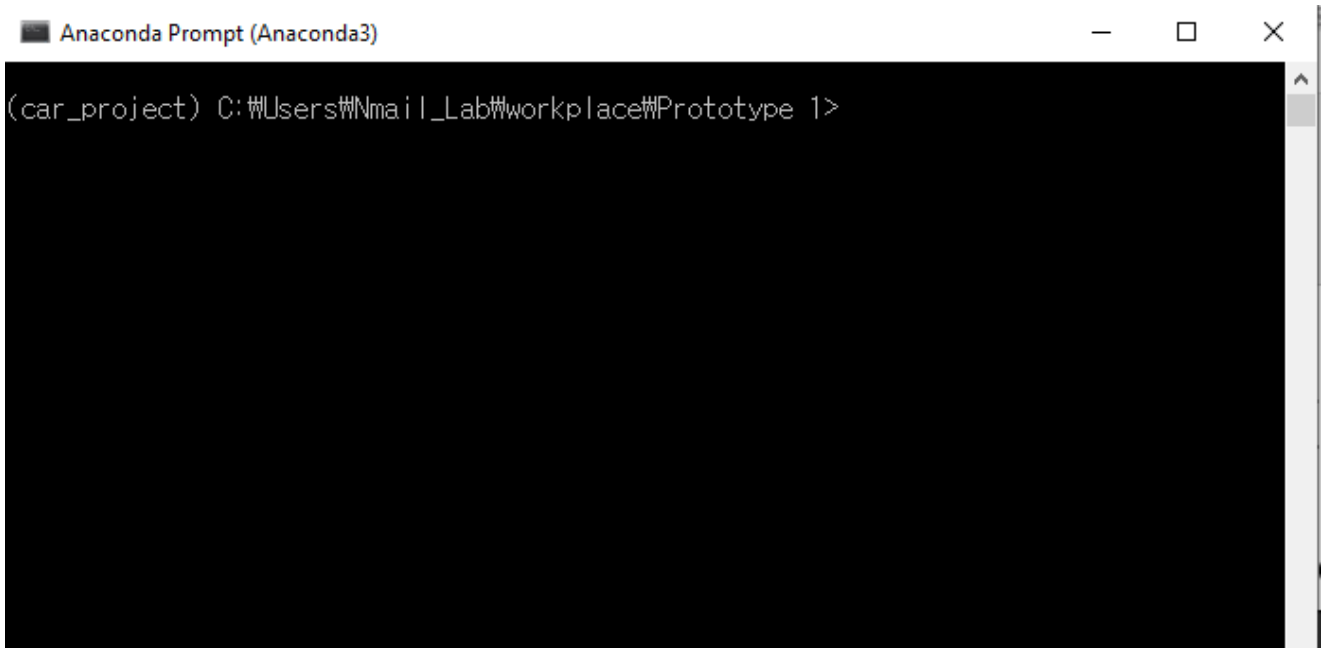
```
1   from mlagents_envs.environment import UnityEnvironment
2
3   import numpy as np
4   print("before connecting ")
5   env = UnityEnvironment(file_name = 'Prototype 1')
6   print("what is going on ?")
7   env.reset()
8   behavior_name = list(env.behavior_specs)[0]
9   print(f"Name of the behavior : {behavior_name}")
10  spec = env.behavior_specs[behavior_name]
```

Execute program

- Open the anaconda prompt

  → Anaconda3 (64-bit) → Anaconda Prompt (Anaconda 3)

- Go to the folder you unzip the template files with ‘cd’ command.



Execute program

- Type ‘ **python python_unity.py** ‘ and enter
- Then program will run.

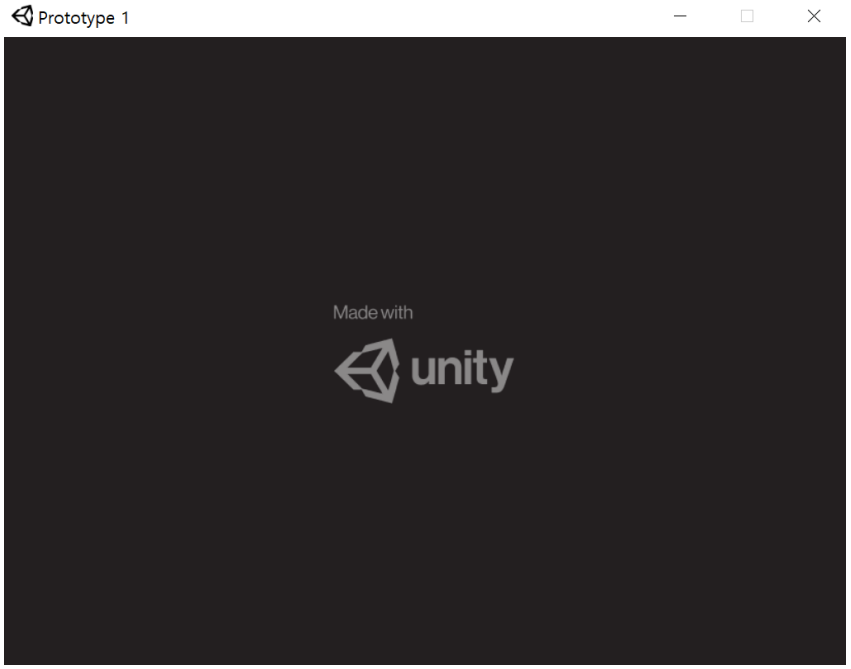
Execute program

- Type ‘ **python python_unity.py** ‘ and enter
- Then program will run.

How to code the python file

- Load Unity Environment
  - Import UnityEnvironment from mlagent library
  - Write file name of the map that you want to try to load the environment

```
from mlagents_envs.environment import UnityEnvironment
env = UnityEnvironment(file_name = 'Road3/Prototype 1')
```



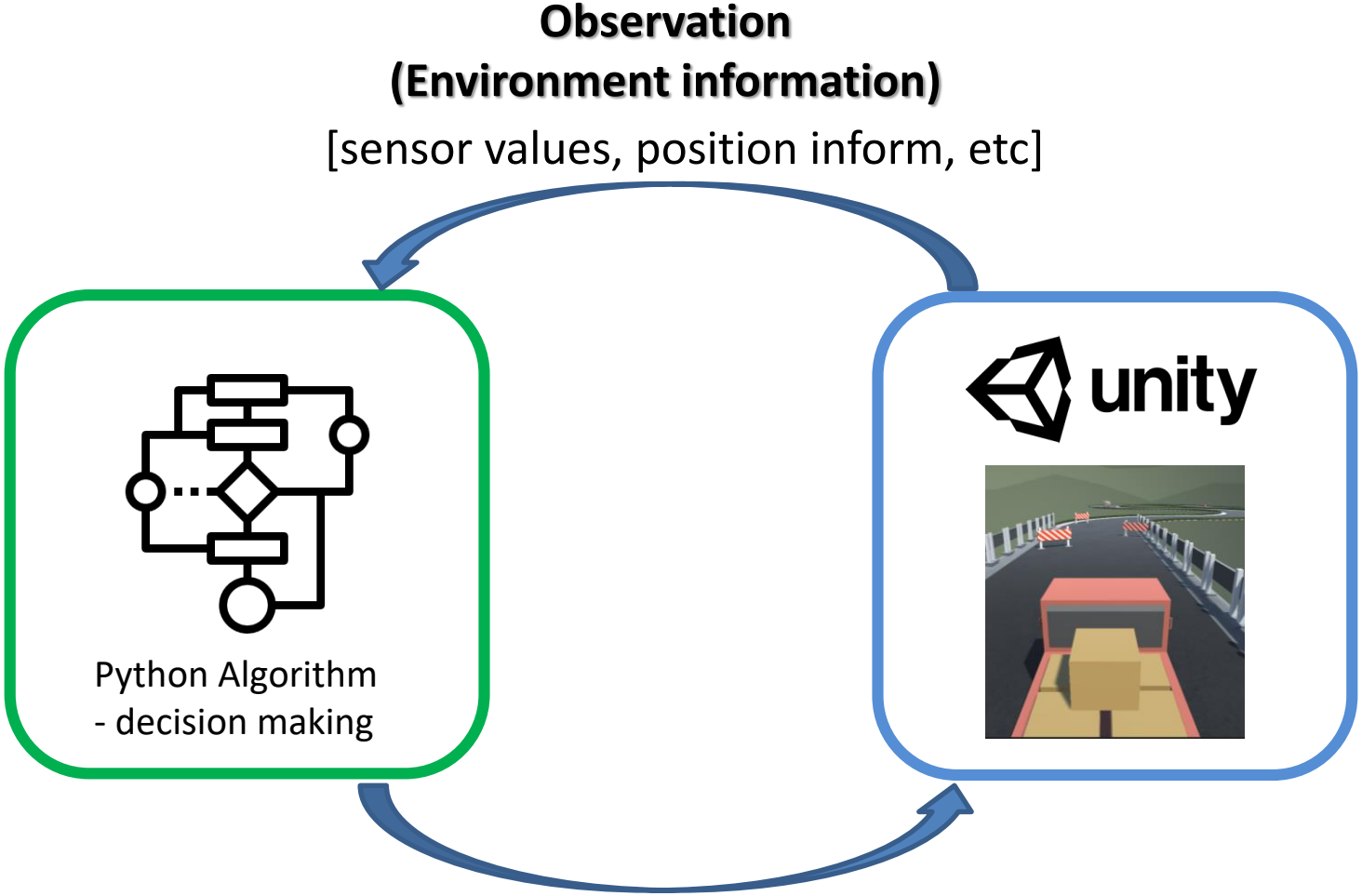
How to code the python file

- Initialize the environment
  - Reset command will initialize unity environment
  - This method takes no argument and returns nothing but will send a signal to the simulation to reset.

```
env.reset()
```

How to code the python file

- Working Flow



How to code the python file

- Obtain Observation
  - Decision_steps.obs is a tuple containing all of the observation for all of the controllable object (truck) with the provided behavior name
  - You can get current observation by using below lines of codes

```
behavior_name = list(env.behavior_specs)[0]
```

```
decision_steps, _ = env.get_steps(behavior_name)
cur_obs = decision_steps.obs[0][0,:]
```

How to code the python file

- Observation

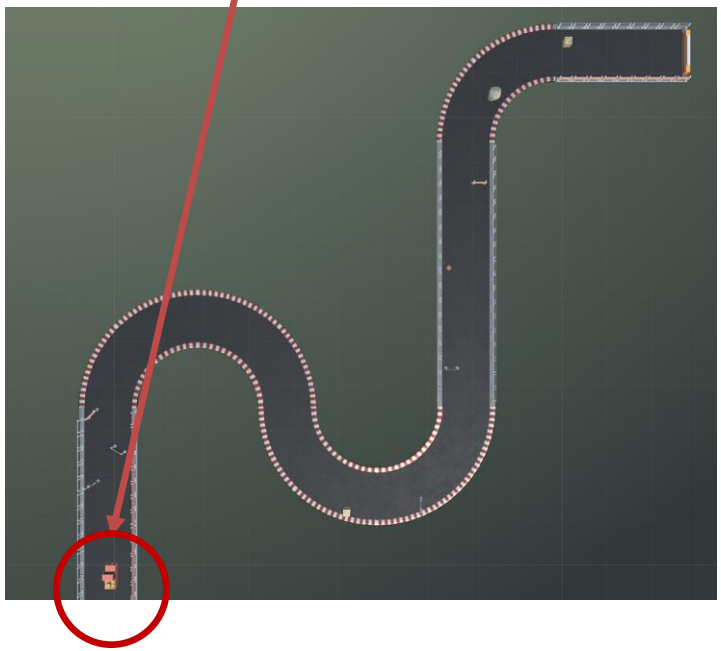
```
decision_steps, _  = env.get_steps(behavior_name)
cur_obs = decision_steps.obs[0][0,:]
```

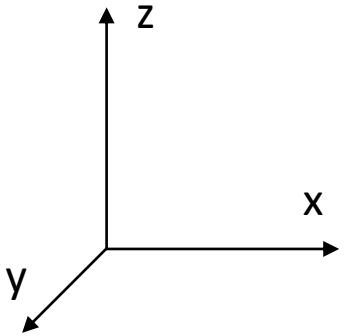
[x, y, z, x', y', z', s1, s2, s3, s4, s5]

How to code the python file

- Observation:
  - first three elements in the tuple are 3d coordinates of our truck

[x, y, z, x', y', z', s1, s2, s3, s4, s5]

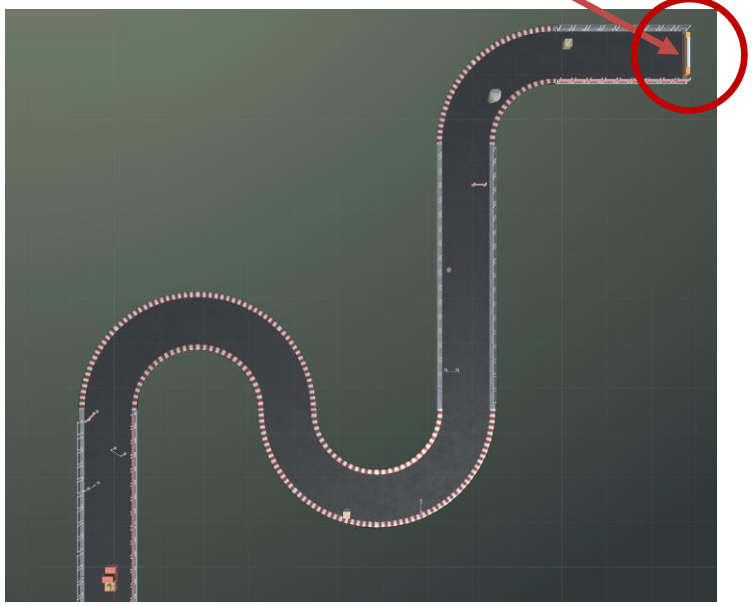


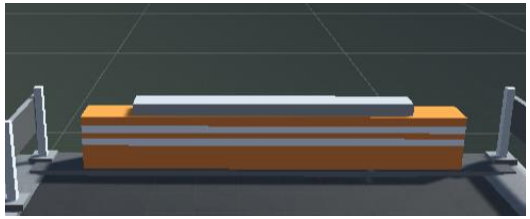


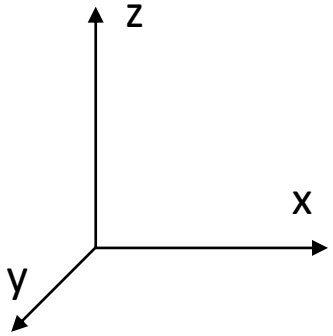
How to code the python file

- Observation:
  - The next three elements in the tuple are 3d coordinates of the final goal
  - Your goal is to **collide** with this **final goal object** as fast as you can

[x, y, z, x', y', z', s1, s2, s3, s4, s5]

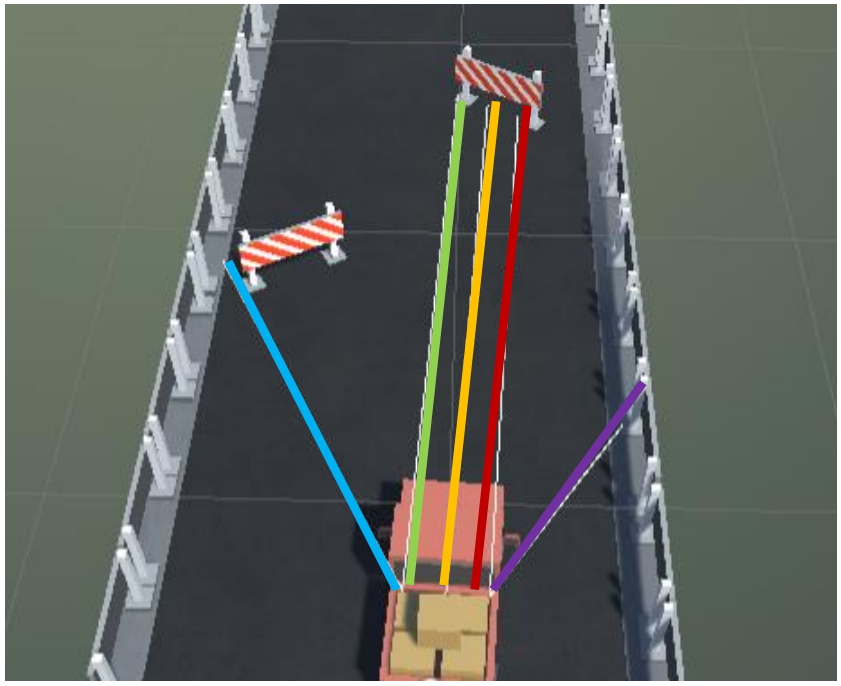






How to code the python file

- Observation:
  - The last five elements are the sensor values that represent the distance between sensor and the detected object
  - Maximum range of detection is 20, and if there is no obstacle in sensor direction, the corresponding sensor value would be 20



[x, y, z, x', y', z',
s1, s2, s3, s4, s5]

How to code the python file

- Make Action
  - You need to put three different values to make any actions
  - First use set_actions function to assign action values
  - Then use step function to move the simulation forward

```
# Set the actions
env.set_actions(behavior_name, np.array([[0,150,150]]))
# Move the simulation forward
env.step()
```

How to code the python file

- Make Action

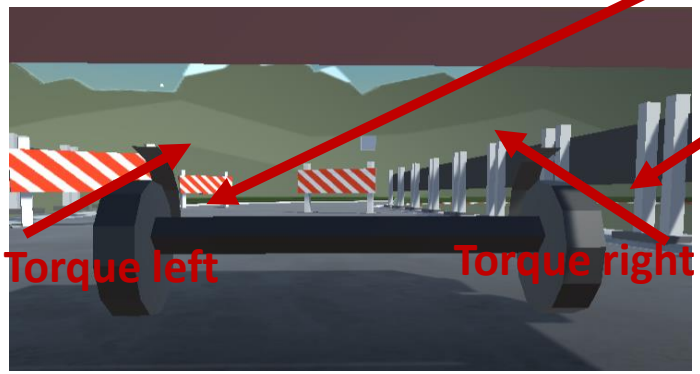
```
# Set the actions
env.set_actions(behavior_name, np.array([[0,150,150]]))
# Move the simulation forward
env.step()
```

- The first action value determines **steering angle** of two front tires
- The value is in between [-1,1], where 1 represents 45 degree

How to code the python file

- Make Action

```
# Set the actions
env.set_actions(behavior_name, np.array([[0,150,150]]))
# Move the simulation forward
env.step()
```



- The following values determine **the amount of torque** applied to left and right tire.
- The range of value is [-150,150]

How to submit the result.

- When you finish your project, you have to submit your python file which contains your algorithm to the KLMS website.
  - Due date : 11/3 23:59pm
- We will run each files in the competition day in real time.
  - Competition day : 11/5 4:00pm

- Help desk will open to help your project and we will announce the help desk schedule on Notice board in KLMS

How to submit the result.

- If you have any questions, contact Email to TA with the following address.
  - munjw777@kaist.ac.kr
  - m-kim@kaist.ac.kr

Thank you