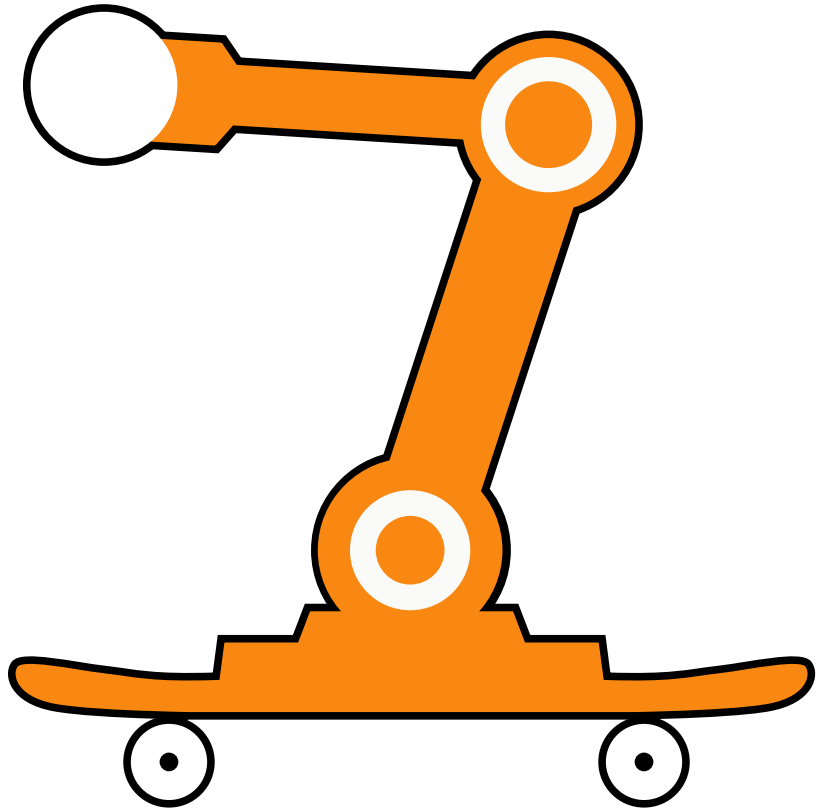
RoboSkate

Kick-Off Project Presentation

Michelle Bettendorf
Gintautas Palinauskas
Meric Sakarya
Batuhan Yumurtacı
Finn Süberkrüb



Cloud-Based Machine Learning in Robotics
Summer Semester 2021

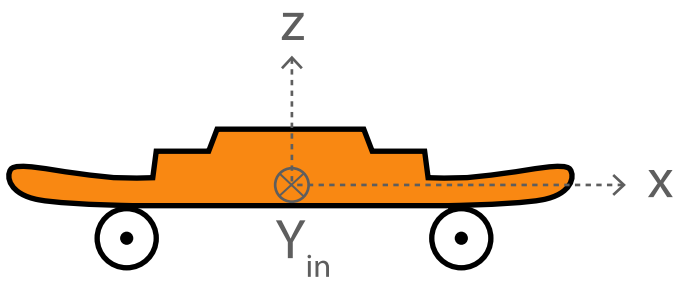
Project Vision and Description

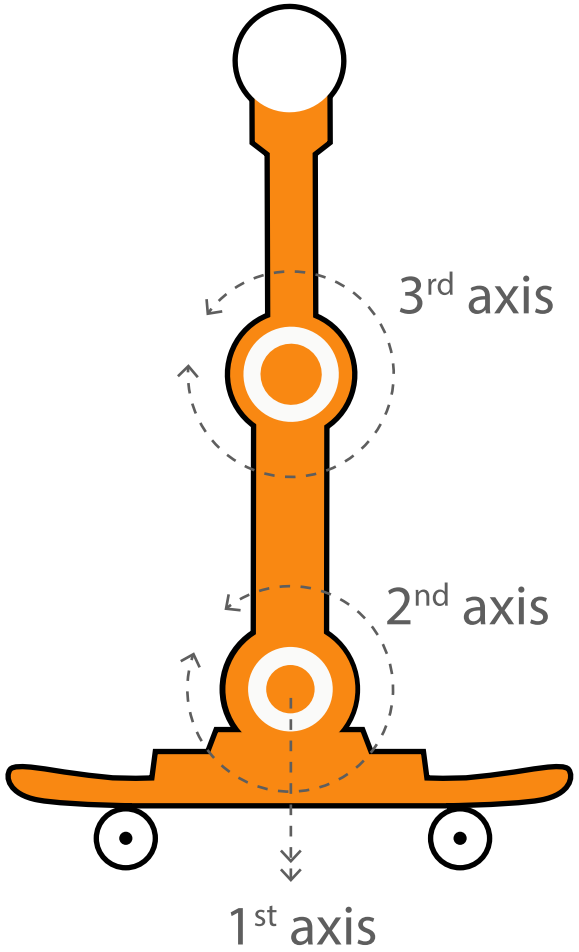
- Topic: Learning human-level skills in gaming using RL
- Scope: Training a RL Agent to play RoboSkate
- Goals:
  - Setting up the simulation
  - Selecting the best suited algorithm
  - Teaching the agent basic movement in simulation
  - Testing the agent on actual game
  - Reaching checkpoints in game
  - Finishing the whole level

RoboSkate

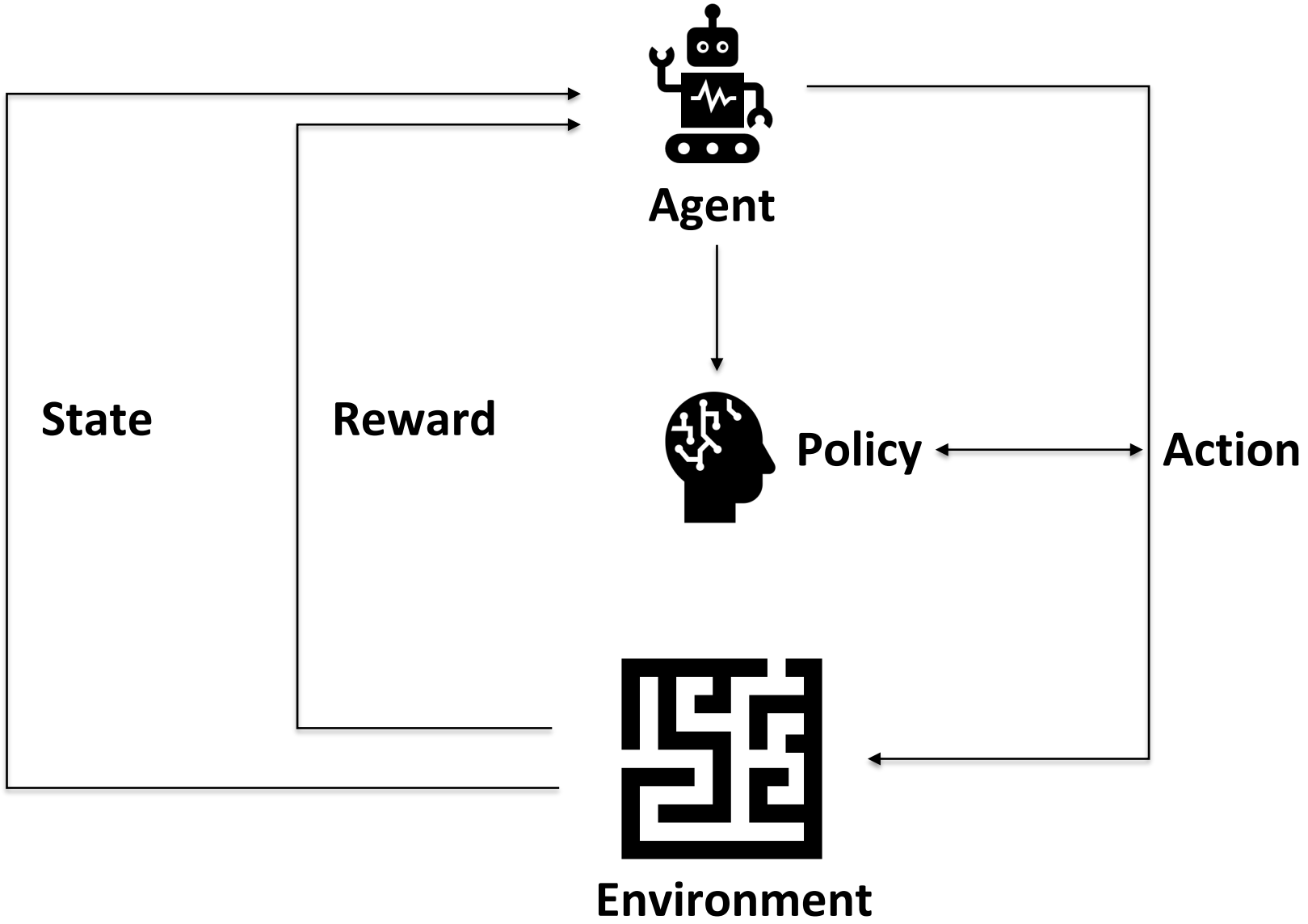
Launch Trailer

Agent

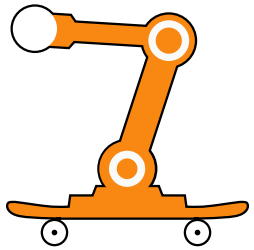




Classic RL Environment



Simulation Environment

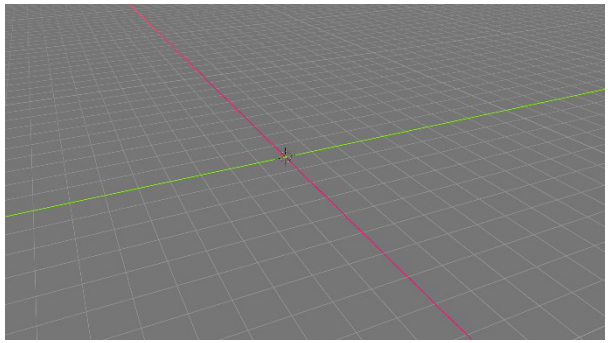


Agent

State
- Joint angles
- Robot coordinates
- Skateboard/Joint velocities
- Image of the environment (?)

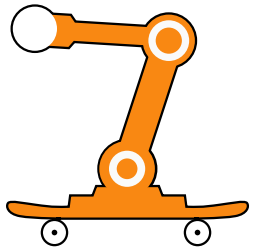
Reward
- Upright position
- Distance travelled
- Speed

Action
- Rotate Robot Arm (+/-)
- Move First Joint (+/-)
- Move Second Joint (+/-)



Environment

Game Environment



Agent

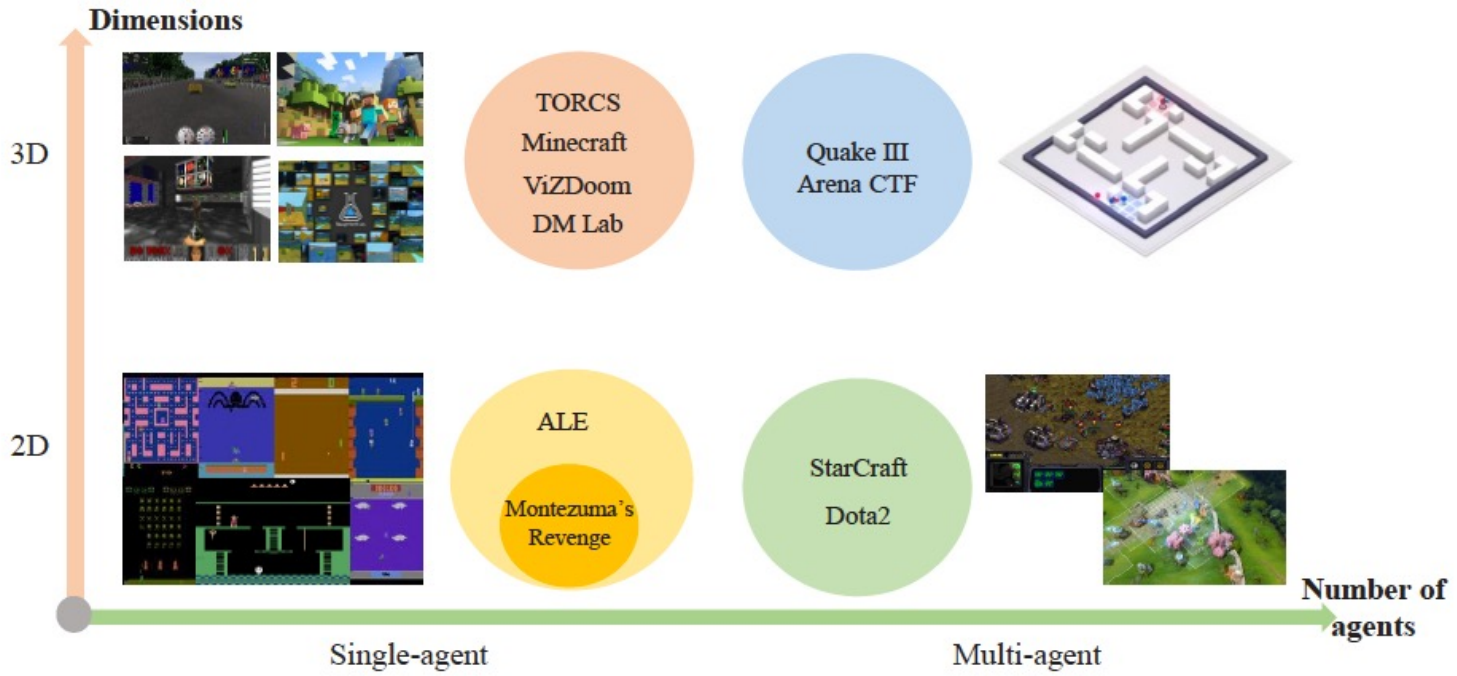
State
- Joint angles
- Robot coordinates
- Skateboard/Joint velocities
- Image of the environment (?)

Reward
- Upright position
- Distance from start
- Distance to goal
- Checkpoints
- Speed

Action
- Rotate Robot Arm (+/-)
- Move First Joint (+/-)
- Move Second Joint (+/-)
- Moving the Camera (?)

Environment

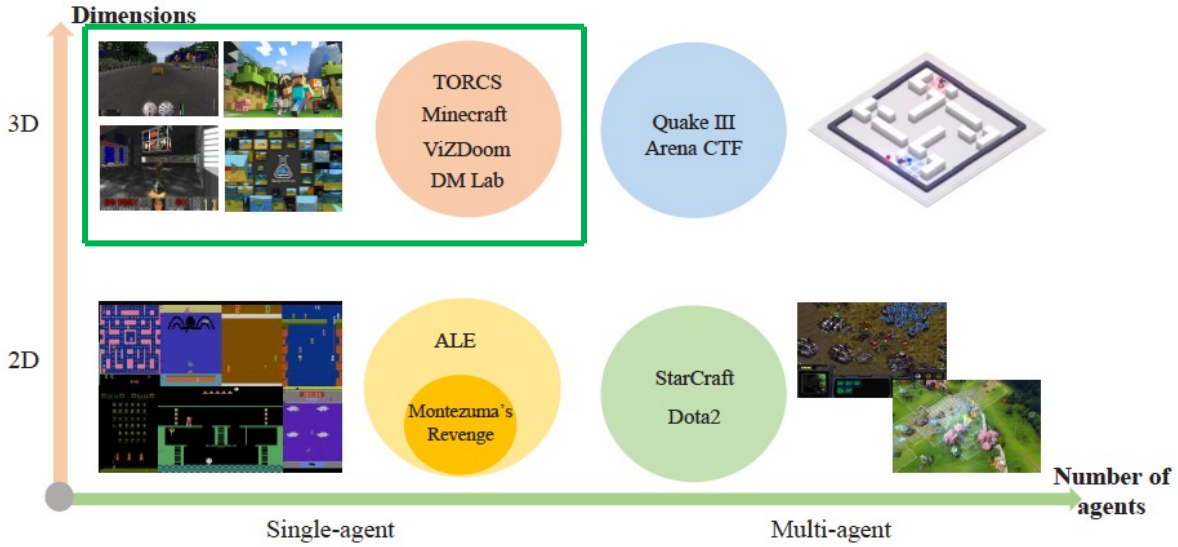
Categories



Source: Shao, Kun, et al. "A survey of deep reinforcement learning in video games." *arXiv preprint arXiv:1912.10944* (2019)

RoboSkate

- 3D
- Single agent
- Model-Free
- Episodic
- Third-person perspective games
- continuous observations
- optimal policy is deterministic
- Fully observable Markov decision process

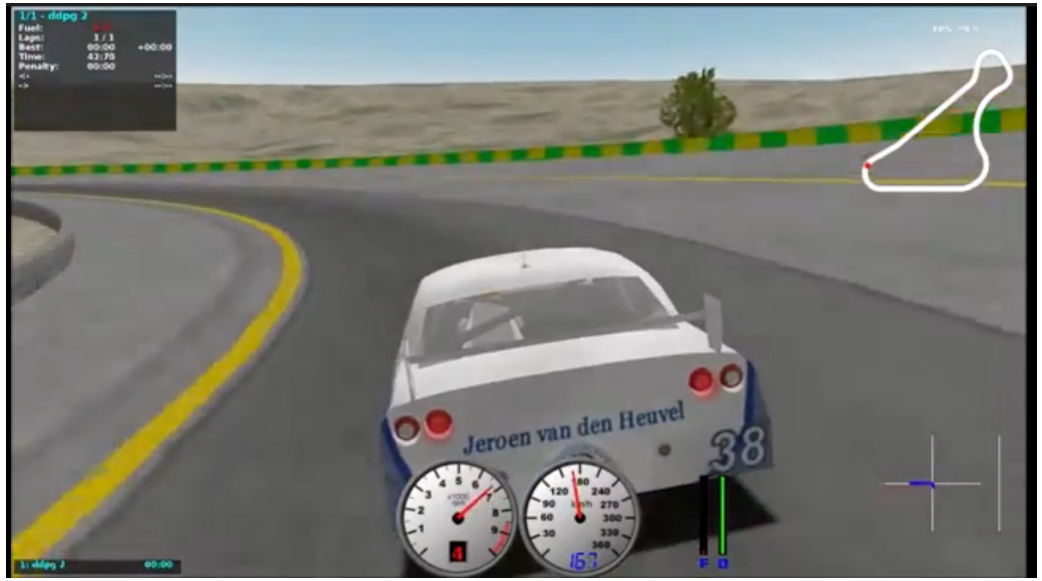


https://store.steampowered.com/app/1404530/RoboSkate/

Similar Approaches

- TORCS , Racing game
- Actions: acceleration, steering, braking

- RL Algorithm: DDPG
  - DQN with deterministic policy gradient
  - Used for continuous control
  - Networks: CNN-LSTM
  - Category: policy gradient, off-policy



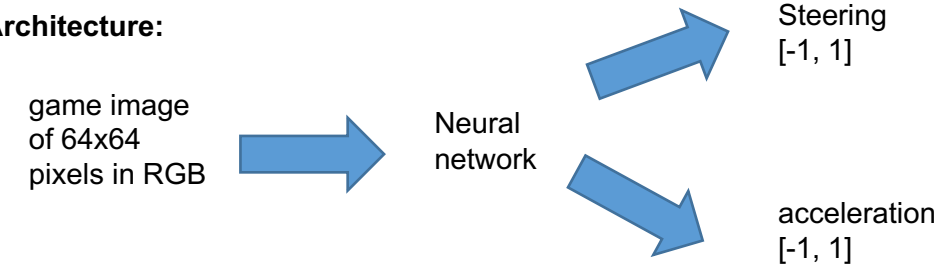
Source: https://www.youtube.com/watch?v=lV5JhxsrSH8

Soft actor critic Trackmania example

22 hours of training
15 actions/s

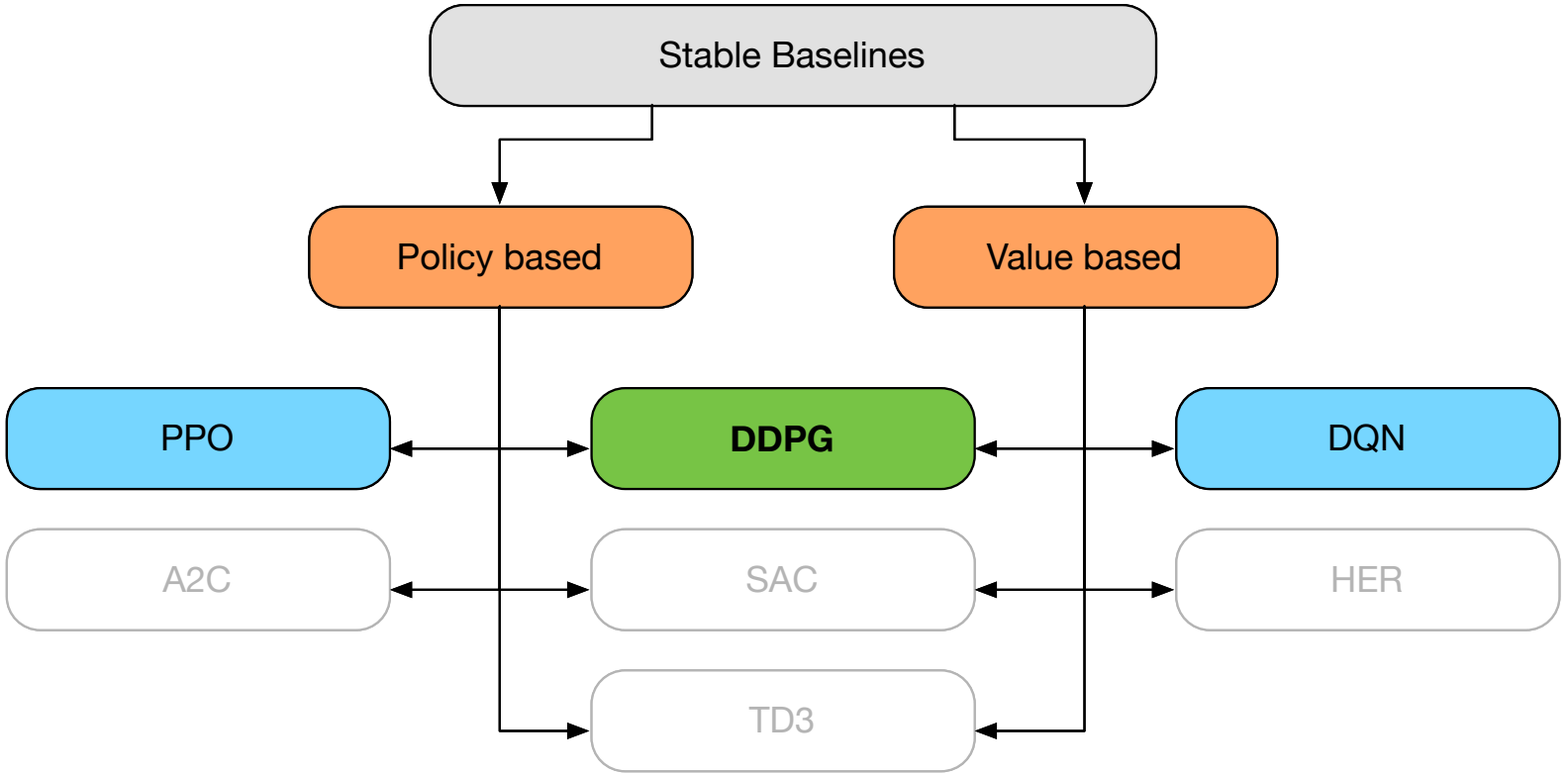
Source: https://www.youtube.com/watch?v=yZFY5ZJtgyM&t=6s&ab_channel=Bluemax666

Architecture:



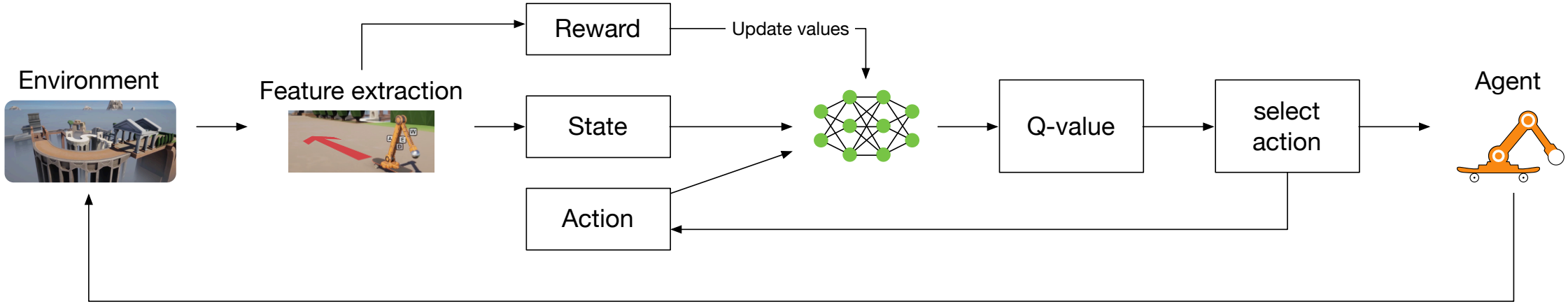
Reward_1 = (speed/150)^2 - 0.15
Reward_2 = 20, if Checkpoint reached

Which RL algorithm? (discrete action space)



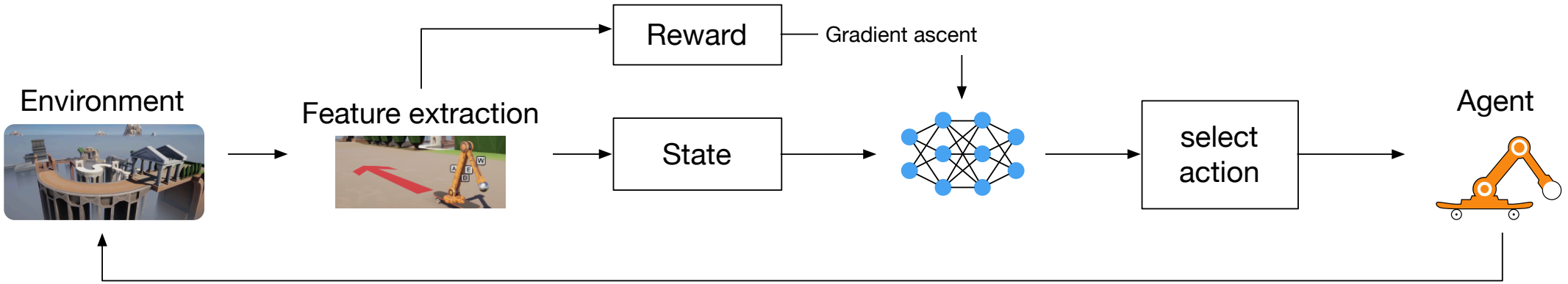
Deep Q Network (DQN)

- Value-based (induced policy, possibly complex value function)
- Off-policy (allows experience replay, efficient in terms of training)



Proximal Policy Optimization (PPO)

- On-Policy Learning (always needs new experience)
- Policy based (no need to learn a value function)
- Allows multiple workers



Actions (high level)

Basic actions

- Pushing
  - Front
  - Back
- Turning
- Braking
  - Front
  - Back
  - Side

Situational actions

- Reset
- On spot turning
- Climbing
- Recovery
- Bridging

Pushing

Turning

Braking

Climbing

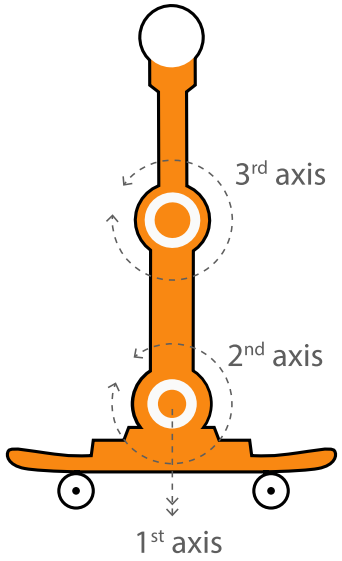
On spot turning

Bridging

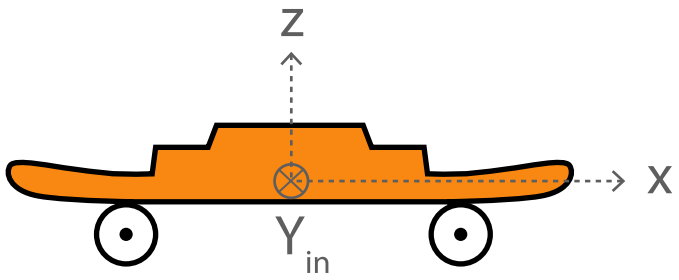
Reseting

State space (1)

- Joint positions

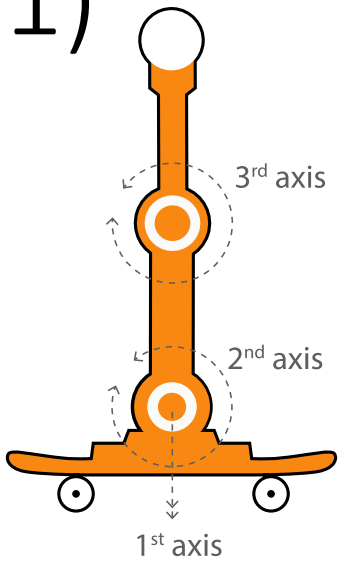


- Robot orientation

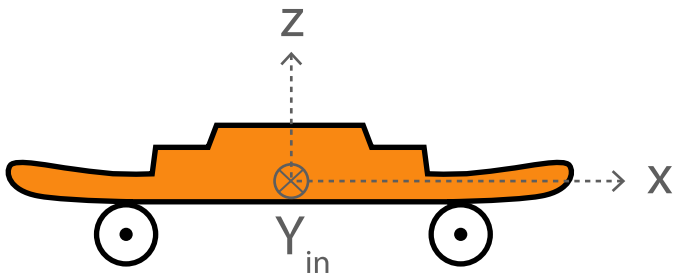


Feature Extraction (1)

- Joint positions
- Joint velocities

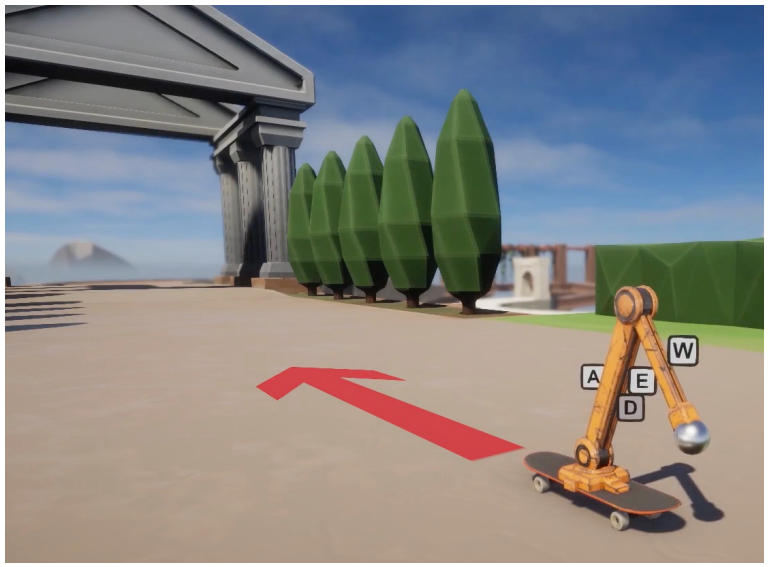


- Robot orientation
- Robot velocities
- Travel direction

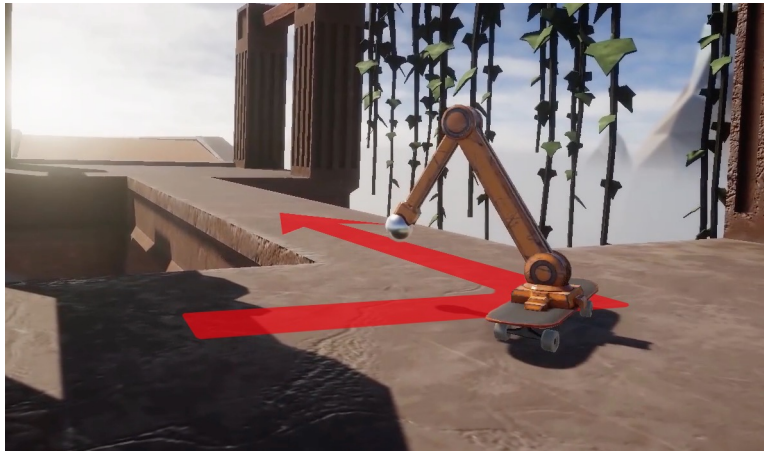


Feature Extraction (2)

- Direction to the next goal

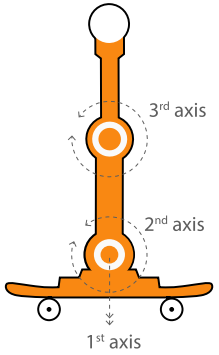


- Obstacle avoidance
- Route planning

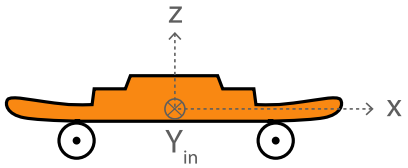


State space (2)

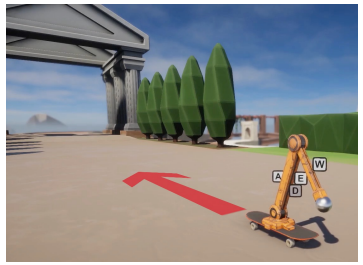
- Joint positions
- Joint velocities



- Robot location
- Robot velocities
- Travel direction



- Goal trajectory



Reward in the simulation

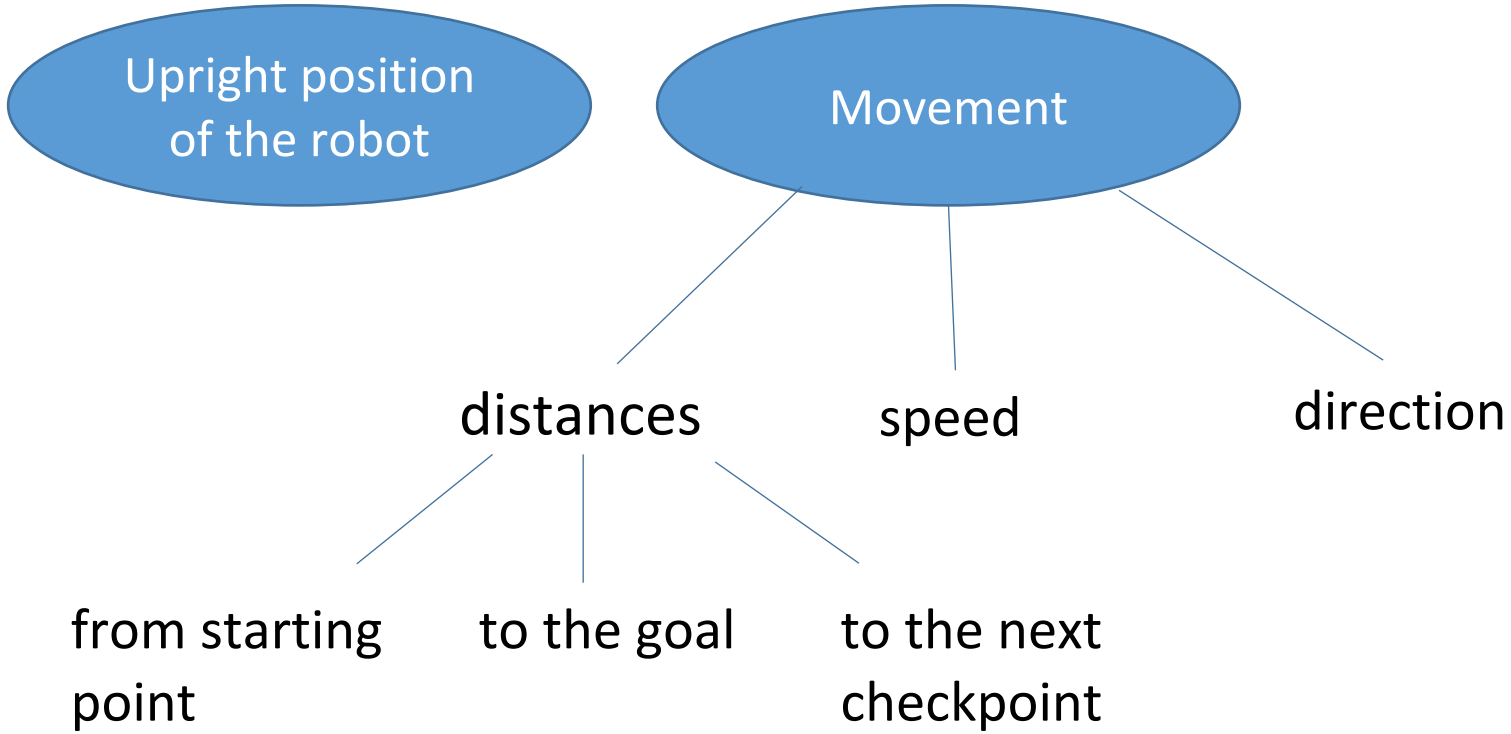
- upright position of the skate board
- distance from starting point
- speed

Reward in the actual game

- upright position of the skate board
- distance from starting point
- speed
- distance to goal
- distance to the next checkpoint
- deviation from the direction in which we want to go

Reward in the actual game

- different subconcepts → different rewards [1]

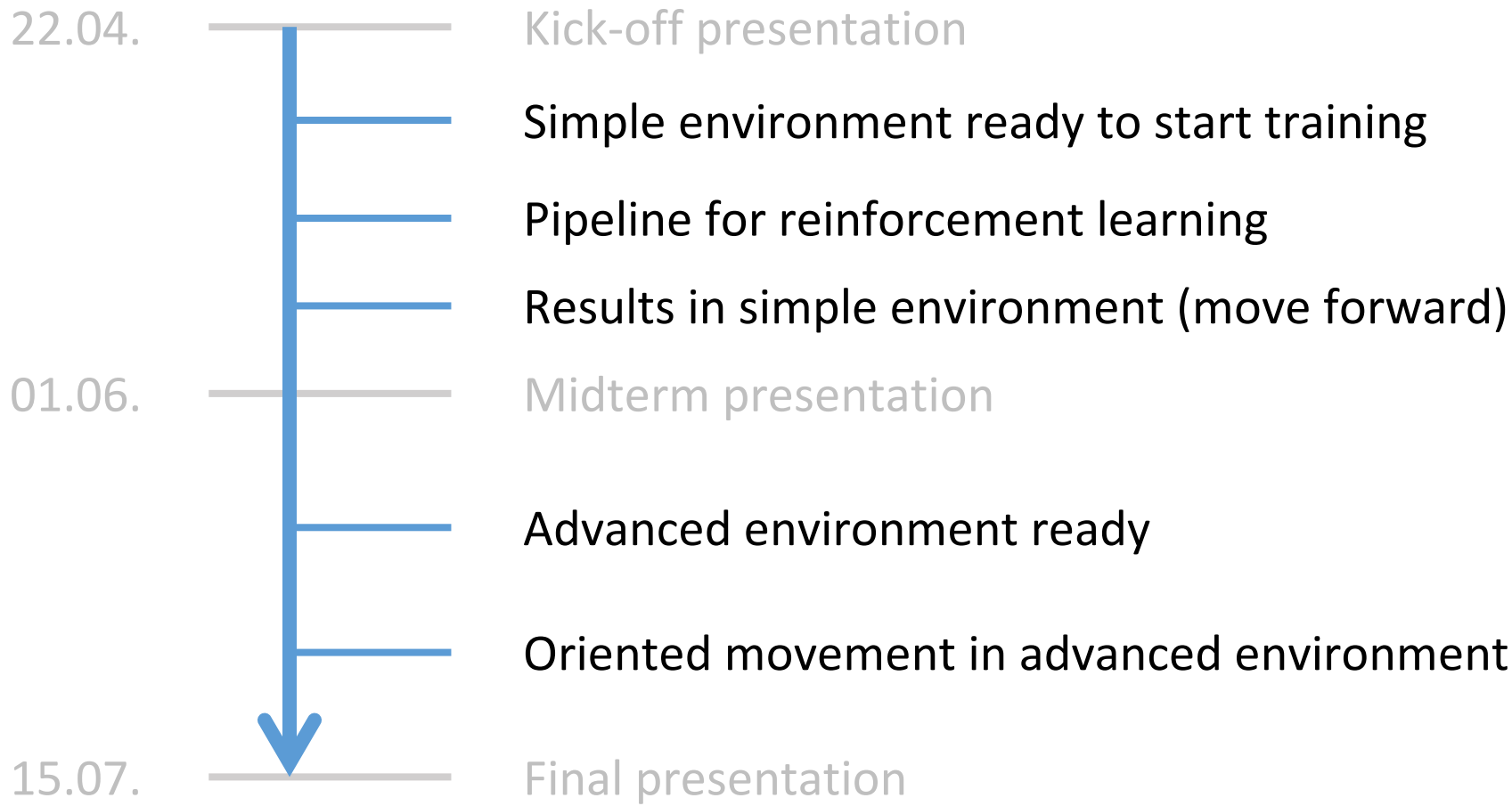


[1] https://medium.com/@BonsaiAI/concept-network-reinforcement-learning-for-flexible-dexterous-manipulation-47bf459b19b9

Responsibilities

	Preliminary Tasks
Michelle Bettendorf	Reward Function
Gintautas Palinauskas	Action Space
Meriç Sakarya	Environment
Batuhan Yumurtacı	RL Algorithm
Finn Süberkrüb	State Space

Schedule


22.04.
Kick-off presentation
Simple environment ready to start training
Pipeline for reinforcement learning
Results in simple environment (move forward)
01.06.
Midterm presentation
Advanced environment ready
Oriented movement in advanced environment
15.07.
Final presentation