Draft Proposal for “Trajectory Planning with Moving Obstacles” (#4)

- Come up with a state representation for dynamic environments.
    - Ego drone / collision objects trajectories
        - Position
            - Acquired from environment representation
        - Velocity
            - Calculated from position difference
            - Indirectly by inputting environment representation recursively in the encoder [3] to represent in latent space
    - Environment representation possibilities:
        - Pointcloud (not preferred) [3]-> Voxels(Downsized Pointcloud)
        - BPS [1]
    - States can be given to the encoder network at each time step for representation of the dynamic environment
- Set up a simple 2D (and later 3D) environment in which an agent can navigate through moving obstacles. [2]
    - Simple Test Environment as a start:
        - 1 square robot, 1 dynamic obstacle, no rotation, grid representation
    - Number of parameters according to dimension to expand later (assuming the drone can move omnidirectionally)
        - 2D: 6 dim. state for each agent (3-pose, 3-velocity)
        - 3D: 12 dim. state for each agent (6-pose, 6-velocity)
    - Use RL to plan optimal trajectories in this environment
        - NN structure: MPNet (prioritised) [3], GA3C [4]
        - RL Method (choose a fitting method type from lecture, try different algorithms)
            - Value iteration methods (e.g. DQNN)
            - Policy gradient methods
            - Actor-critic methods (e.g. SAC)
- Optional: Extend the method to work with uncertainties in the motion prediction of the collision objects
    - Noise of collision objects states
        - Modelling it with a Gaussian noise representation for point cloud data to avoid potential collisions
    - Influence of unseen obstacles
        - For example, people behind a truck

References:

[1] Prokudin S, Lassner C, Romero J. Efficient learning on point clouds with basis point sets. InProceedings of the IEEE/CVF international conference on computer vision 2019 (pp. 4332-4341).

[2] Everett, M., Chen, Y.F. and How, J.P., 2018, October. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3052-3059). IEEE

[3]Qureshi AH, Miao Y, Simeonov A, Yip MC. Motion planning networks: Bridging the gap between learning-based and classical motion planners. IEEE Transactions on Robotics. 2020 Aug 3;37(1):48-66.

[4]Surmann H, Jestel C, Marchel R, Musberg F, Elhadj H, Ardani M. Deep reinforcement learning for real autonomous mobile robot navigation in indoor environments. arXiv preprint arXiv:2005.13857. 2020 May 28.

Alternative Topic Choices in Order:

Solving Complex Sparse Reinforcement Learning Tasks (#2)

Non-sequential Reinforcement Learning for Hard Exploration Problems (#3)

Mo Chen (03739983)
Ragip Volkan Tatlikazan (03744935)