**Software Project – [AI]T Racing – Phase 1**

**Our Team - [AI]T Racing:**

* Marten Kreis 🡪 Design Neural Network
* Niklas Kugler 🡪 Team Leader, Neural Network Training
* Valerio Müller 🡪 Design Neural Network
* Finn Lehnert 🡪 Training Data
* Jakob Häringer 🡪 Organisation, Scrum-Master, Code-Master
* Daniel Kosc 🡪 Image Processing
* Frederik Poschmann 🡪 Output Optimization

1. **Design of the Neural Network (5. Model Definition) R: Jakob Häringer, Frederick Poschmann, Marten Kreis, Valerio Müller**
   1. Try to find an existing architecture that fits our problem (look for Benchmarks)
      1. Cifar10
      2. Imagenet
   2. Think about two architectures - so we can provide two solutions to the problem in the end

If we use a non-existing architecture:

* 1. How should the CNN model be structured?
     1. Convolutional layers (# of filters, filter size)
     2. Non-linearity layer
     3. Which method should be used for the pooling layers?
     4. Fully connected layer

1. **Training Data (1. Data Reading / 2.Data Loading) R: Finn Lehnert, Marten Kreis**
   1. Is speed/steering available for input?
   2. How to extract data from the game (video or frames)
   3. How to extract steering data (steering, acceleration, braking) and connect to the matching frame
   4. Can we randomize images to generate more training data?
   5. Data Balancing
2. **Image Processing (3. Data Preprocessing) R: Daniel Kosc, Niklas Kugler**
   1. Find out size of image (height, width, # of color channels)
   2. Reduce image information (color, etc.)
   3. Preprocessing of Training Data
      1. Extract grey filtered image
      2. Binary coding of image (Street/Grass)
3. **Training of the Neural Network? (4. Model Training) R: Niklas Kugler**
   1. How to pass Training Data to ANN? Data Loading/Reading
   2. Research
   3. Different functions and implementations in PyTorch
   4. Definition of hyperparameters (learning rate, # of epochs, etc.)
4. **Output Optimization (6. Model Evaluation)**
   1. Input / Output Layers
   2. Evaluation of the NN
   3. Increase stability of the NN
   4. Further Image Preprocessing possible?
5. **Organization R: Jakob Häringer**
   1. Weekly SCRUM meetings to share status of current tasks
   2. Organize Meetings 🡪 R: SCRUM Master: **Jakob Häringer**
   3. Distribute Work 🡪 how can we work concurrently on the code?

Code-Master: **Jakob Häringer**

* + 1. GitLab
    2. Code structure

1. **How to implement a CNN in PyTorch? R : Everybody**
   1. Focus on implementing a basic example
   2. Get to know pytorch methods
   3. Do we need further packages, which are needed?
   4. Present / Share knowledge with others
   5. Final Implementation 🡪 Everybody’s responsibility