With the **package trainer**

you can test the performance of players:

- First start the player program(s), e.g. **Agent_SoccerTeam**, such that it becomes left team (blue).
- Then start the trainer program **Agent_Trainer**, such that it becomes right team (red).

The program will perform the experiments that are defined in classes **Agent_Trainer** and **TrainerThinking** .

It uses the methods from class **TainerCommandExecutor**

- to move the ball
- to move players
- to change game states.

The classes **Agent_Trainer** and **TrainerThinking** work together in a similar way as in package agentSimpleSoccer.

Agent_Trainer implements a state machine that is controlled by the game states. During game state „PlayOn“ it calls the method controlExperiments() from **TrainerThinking.**

TrainerThinking executes the experiments one after each other. It implements a state machine that is controlled by the states of an experiment (variable expState).

The settings of experiments are defined at expState=START by specific methods startExperiment().

An experiment ends when the time limit (defined by maxExpTime) is reached, or after relevant game state changes (e,g. after goals).

For the settings of experiments you can

- move the ball to a certain position and give it a speed into a certain direction by the command

```
beamBall(float x_Position, float y_Position, float z_Position,
                        float x_Velocity, float y_Velocity, float z_Velocity)
```

• use z_Position = 0.042f for positions of ball on the ground

- move a player to a certain position with a certain orientation by:

```
moveRotatePlayer(Team team, int playerNumber,
        float x_Position, float y_Position, float z_Position, float Orientation)
```

• use z_Position = 0.375f for positions of players on the ground,
• orientation is in degrees (0 pointing to y-direction, -90 to x-direction),
• use LEFT or RIGHT to specify a team

See class **TainerCommandExecutor** for more details.

In class **TrainerThinking** you can already find 2 examples:

```
private void startExperiment1() {
        maxExpTime = 40;
         trainer.beamBall(2.3f, 0);
         trainer.moveRotatePlayer
                (TrainerCommandExecutor.Team.LEFT, 2, 2f, 0f, 0.375f, -90f);
}
```

Test for kicking from a fixed position

```
private void startExperiment2() {
         float x_speed; float y_speed;  float z_speed;
         maxExpTime = 20;
         x_speed = -8f;
         y_speed = -2f + ((float) experimentNumber) / 10f;
         z_speed = 0;
        trainer.beamBall(0f, 0f, 0.05f, x_speed, y_speed, z_speed);
        trainer.moveRotatePlayer
                (TrainerCommandExecutor.Team.LEFT, 1, - 4.5f, 0f, 0.375f, -90f);
}
```

Test for goalkeeper with different ball movements

For other experiments you can define your own methods startExperiment() .