Daniel Schäfer (2549458)
Rafael Dewes (2548365)
Kevin Müller (2550062)

# Important Notes

## The GUI and how to install/use it

https://github.com/kvnmlr/simulink-advanced-visualization/releases to download our GUI (Created using Unity)

We decided to use a GUI built in unity, because it made it significantly easier to observe the game and the sending of data using TCP Blocks did not impact performance significantly.

The GUI has to be started before starting the simulation (using `model.slx`)

# Our Submission Scenarios

All scenarios can be observed in the first $t = 10s$ of the default configuration! Our collector will push the opponent collector multiple times, causing him to drive out of bounds because he did not receive a referee update in time, to update his expected position with his real position (as his position was changed).

To play indefinatly just increase the $t$ value in simulink to whatever you like.

# Collision

For performance reasons only the collisions between the two collectors are currently enabled.

# Push conditions to avoid scout

The condition is implemented (and working), but because our Collision is not completely implemented yet, the impossible case of the hitboxes of collector and scout overlapping will cause our collector to not detect the scout of our team correctly.

The detection works as shown below:

Daniel Schäfer (2549458)
Rafael Dewes (2548365)
Kevin Müller (2550062)

**Embedded Systems Milestone 2**