

1 class FastDodge

A class to take the current situation, in which we want to go fast, and output a reasonable dodge to accelerate quickly. Lots of room for improvement here. Careful of changes - `DefaultKickoffs.py` uses this mechanic in its current form.

1.1 self.current_state

Carstate. Typically fed in as `game_info.me`

1.2 self.old_state

Carstate. This is just `current_state` from the previous frame.

1.3 self.goal_state

Carstate. The state we would like our car to be in at the end. Usually just position and orientation updates to the current state using `copy_state()`.

1.4 self.direction

+1 or -1. +1 for right, -1 for left. Chosen by `Miscellaneous.py` based on `current_state` and `goal_state`.

1.5 self.oversteer

Boolean. True if we want to turn past the target, then flip back towards it. This is needed for some existing code, but I'm probably going to stop using it. Especially since I believe that with a cancelled dodge it just isn't faster to oversteer.

1.6 self.boost_threshold

Float. The speed at which the bot will stop boosting on the ground and jump to begin the dodge. If not provided the class calculates a default value that is good enough for the precision currently used in the rest of the process.

1.7 self.input()

Returns the controller input for the maneuver.

2 class GroundTurn

Call this maneuver to turn towards a specified target state while on the ground. I will improve and optimize this over time, currently it's fairly naive, but still good enough. Eventually this may or may not include wall driving, we'll see about that once we get there.

2.1 self.current_state

CarState. The current state of the bot, typically from `game_info.me`.

2.2 self.target_state

CarState. The target state of the maneuver, typically from setting position in `copy_state()`.

2.3 self.can_reverse

Boolean. Set to True if the bot should try to go backwards. I'm still not sure how this should be implemented, but some sort of backwards driving is necessary.

2.4 self.input()

Returns the controller input for the maneuver.

3 class NavigateTo

A maneuver to drive to and stop on a target, then readjust to face a direction.

3.1 self.current_state

CarState. The current state of the bot, typically from `game_info.me`.

3.2 self.goal_state

CarState. The target state of the maneuver, typically from setting position and Orientation in `copy_state()`.

3.3 self.input()

Returns the controller input for the maneuver.