Composable Functions

Overview

Using composable functions essentially means ordering commands and paths into autos that are created as commands. This is, after some consideration, what YETI prefers to use to create autos. It is important to note that, while commands are created and used normally, paths have to be created through PathPlanner or Choreo.

Walkthrough

Here is an example of an auto we used in 2025:

```
public Command right1Pc() {  1 usage   boomermath +1
    Optional<PathPlannerPath> lineF = PathPlannerUtils.loadPathByName("lineF");

    return lineF.isEmpty()
            ? Commands.none()
            : AutoBuilder.followPath(lineF.get())  Command
                    .andThen(reefAlignPPOTF.setBranch(ReefAlignPPOTF.Branch.LEFT))  SequentialCommandGroup
                    .andThen(reefAlignPPOTF.reefAlign())
                    .andThen(
                            coralManipulator
                                    .transitionTo(CoralManipulatorState.L4)
                                    .withTimeout( seconds: 0.5))
                    .andThen(coralManipulator.transitionTo(CoralManipulatorState.SCORE_L4))
                    .andThen(coralManipulator.transitionTo(CoralManipulatorState.STOWED));
}
```

As you can see, the auto- *right1Pc*- is actually a command. The first thing we do is define *lineF*, which is a path from PathPlanner. The next step is to make sure the path exists, and when it is confirmed, start creating the auto. In this case, the first thing we do is follow the *lineF* path, but we can switch it around with other commands in any order that we deem desirable. We use other commands normally.

Note that paths are *not* the only way for the robot to drive around. As you can see, we use our auto-align command, which translates and rotates the robot to point towards our target.

Happy auto-making!!!

Revision #2
Created 2025-07-01 02:12:44 UTC by Vihaan Gaurav
Updated 2025-07-02 14:54:44 UTC by Vihaan Gaurav