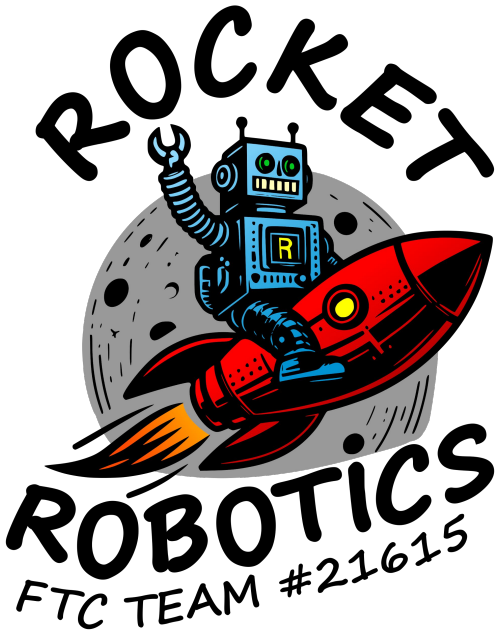


Programming Subsystems with Java

Julien Vanier, Coach
Sofia, Programmer
Rocket Robotics #21615

2026 FTC Kickoff

Problems with a big loop

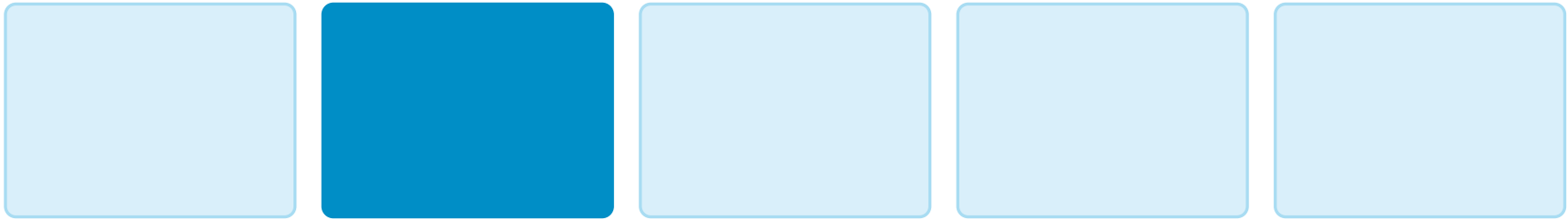
One loop to control every mechanism.

```
while (opModeIsActive()) {
    // drive
    ...
    // intake
    ...
    // lift
    ...
    // shooter
    ...
    // telemetry
    ...
}
```

- **Everything lives in one file**
  Every new mechanism makes it longer
- **Gamepad code mixed with motor code**
  Nothing stops two buttons fighting over one motor
- **The loop gets slow**
  Motors are commanded to the same power every loop
- **Only one person can edit it**
  Your whole programming team works on one file

Organize your code...

with subsystems...



and commands

What is a subsystem

One mechanism. It owns its motors, servos, and sensors. It is the **only** code allowed to touch that hardware.

Drive	Intake	Lift	Bucket	LEDs
4 motors Odometry	1 motor	1 motor Encoder	1 servo	2 channels

No other class reads or writes the intake motor.
If the intake misbehaves, there is exactly one file to open.

What goes inside a subsystem

Hardware

The motors, servos and sensors it owns

```
private final DcMotor motor;
```

State

What it needs to remember between loops

```
private double targetPosition;
```

Reflexes

What it needs to do every loop

```
private void periodic() {}
```

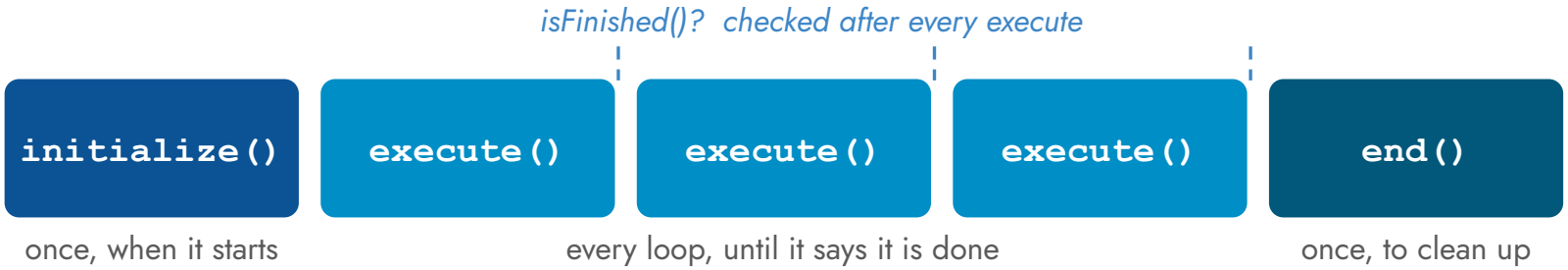
Actions

What the rest of the robot can ask it to do

```
public void goToPosition(double position)
```


What is a command

A command performs a **high-level robot function** using the methods defined by subsystems.
It starts, runs over many loops, then it is done.

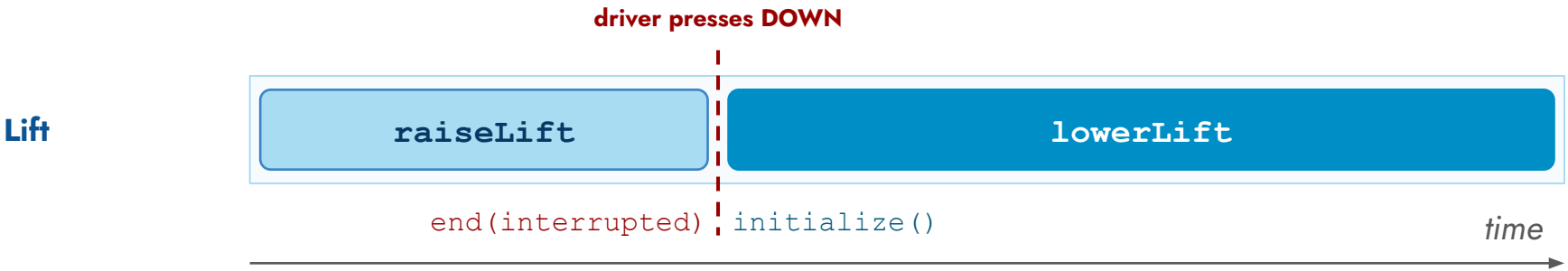


Subsystem = what the hardware can do. Command = when to do it.

One command per subsystem at a time

A command declares which subsystem it needs. **requires: Lift**

Schedule a second command on the same subsystem and the first one is cancelled.



Two commands can never fight over the same motor.

Example: Intake subsystem

Everything the robot knows about the intake lives here.

```
public class Intake extends SubsystemBase {

    private final DcMotor motor;

    public Intake(HardwareMap hardwareMap) {
        motor = hardwareMap.get(DcMotor.class, "intake");
    }

    public void spin(double power) {
        motor.setPower(power);
    }
}
```

The hardware

Declare intake motor.
Nothing outside this class can control this motor.

The one action

One function to spin the intake.
Positive pulls in, negative pushes out, zero stops.

Example: Intake commands

All the high-level functions the robot can do with the intake.

`intakeIn` — requires: Intake

```
public Command intakeIn() {
    return startEnd(
        () -> spin(1.0),
        () -> spin(0));
}
```

`intakeOut` — requires: Intake

```
public Command intakeOut() {
    return startEnd(
        () -> spin(-1.0),
        () -> spin(0));
}
```

How to stop the intake? **Don't schedule any intake command.**

Triggering commands

At robot init, list all the conditions that triggers commands, instead of polling every loop.

Bind to a button

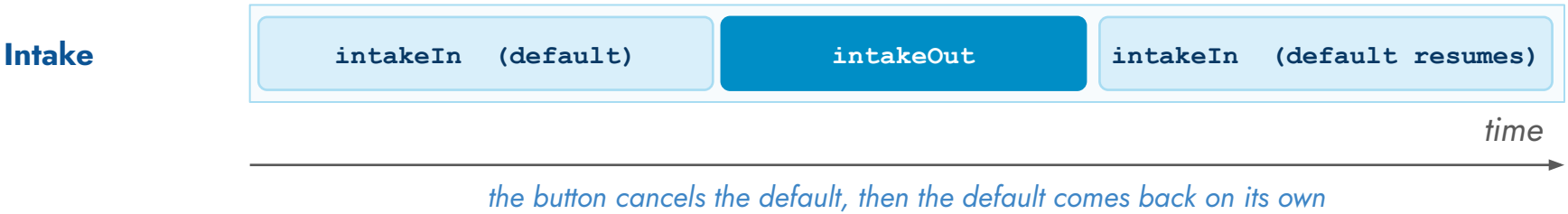
Runs while the driver holds it.

```
gamepad1.getButton(RIGHT_BUMPER)
        .whileHeld(intakeOut());
```

Set as the default

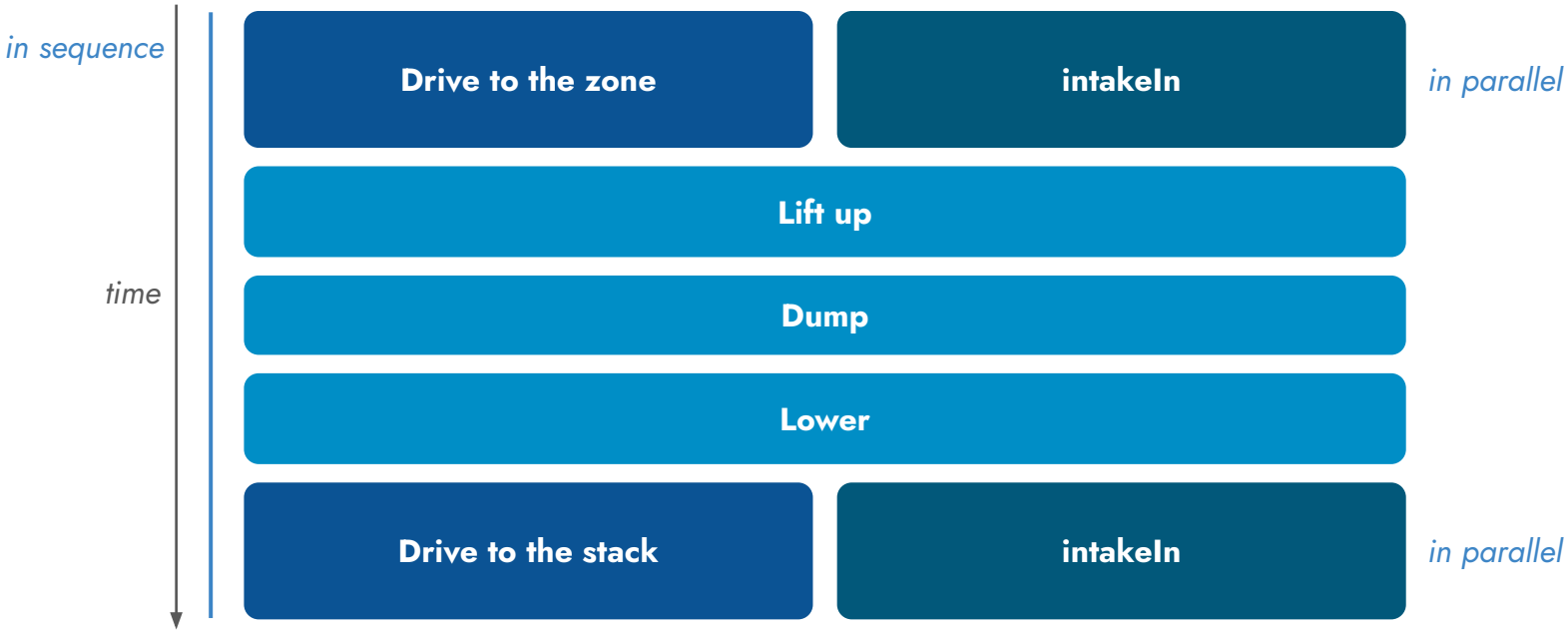
Runs whenever nothing else wants the subsystem.

```
intake.setDefaultCommand(
        intakeIn());
```



Autonomous is one big command

Stack commands in sequence, or run them side by side. A group of commands is itself a command.



Auton is made of the same building blocks as teleop

Which library to use?

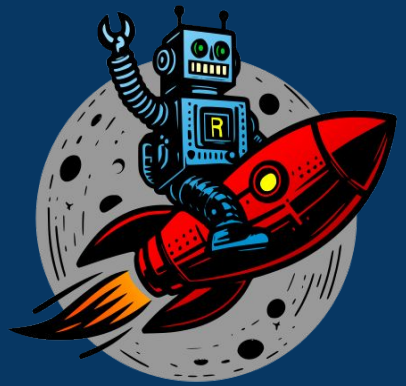
Many libraries for subsystems and commands. We recommend

SolversLib

Same syntax as WPILib used in FRC now, and FTC in 2027.

Alternative libraries: NextFTC, Pedro Pathing Ivy

Learn these concepts now, and use them for many years



Good Luck!

Questions? pccsrocketrobotics@gmail.com

- **This presentation** and past kickoff decks
- **docs.seattlesolvers.com** SolversLib command framework
- **gm0.org** Game Manual 0, start here for anything FTC
- **pccsrocketrobotics.com** our code, CAD and portfolio

Julien Vanier and Sofia, Rocket Robotics 21615, 2026 FTC Kickoff