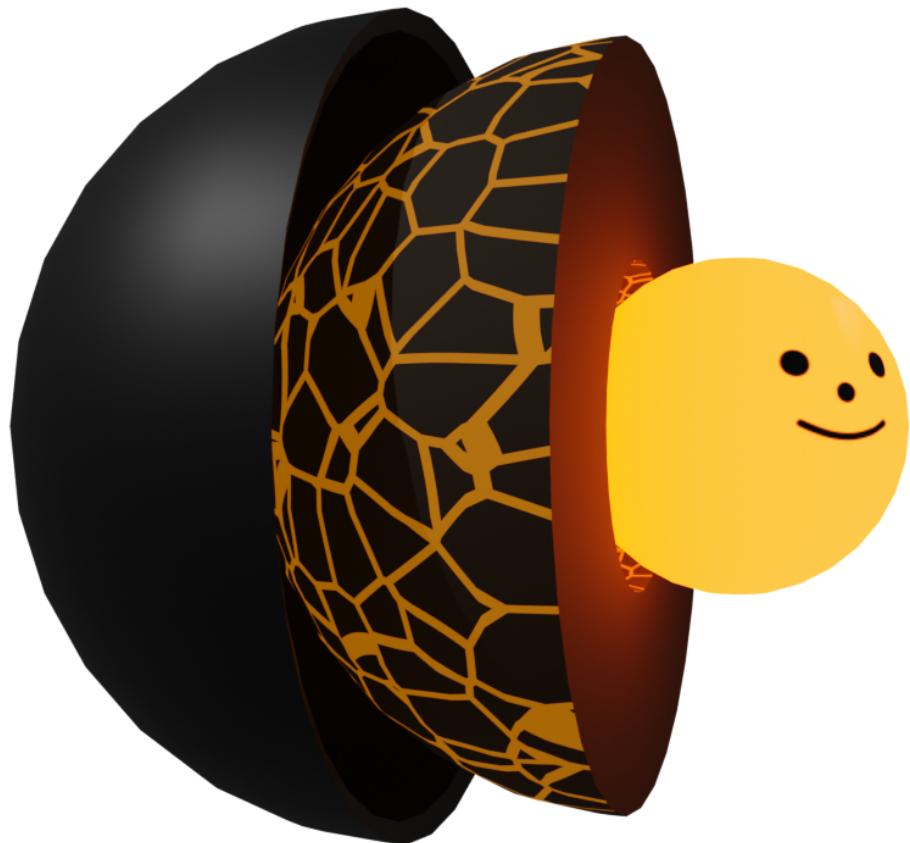


RIT VEXU Software Engineering Notebook

2023-2024

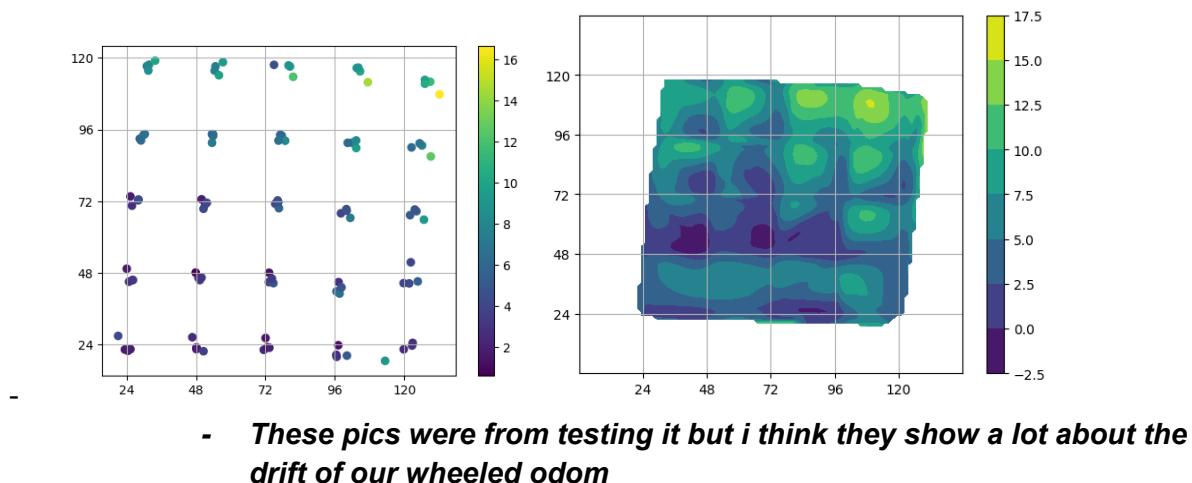
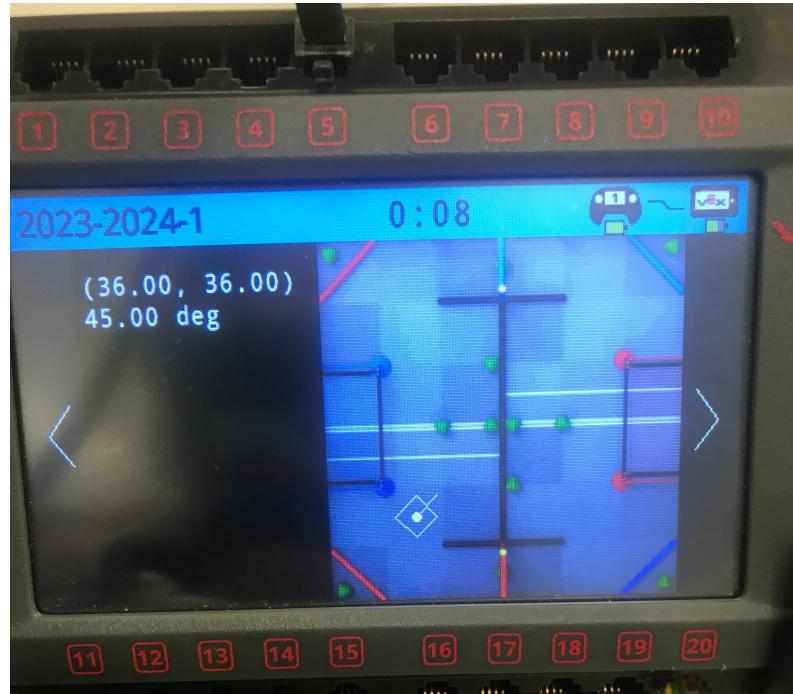


Software Development Process

- Subrepo - why we use it, why tried to switch to submdules, and why we switched back
- Auto Documentation Creator
- Auto Notebook Creator
- Clang-tidy

Core: Updates This Season

- Biggerer and betterer screen system
 - Buttons and sliders enabling hardware to test speeds without a coder redeploying every 5 seconds
 - PID, PIDFF tuners
 - Odometry Map
- Motor stats
- Auto chooser
- Cata State Representation for debugging, tuning
- GPS testing (with all the pretty plots (hell yeah))
- Using GPS to tune odometry



- *Also reasonable to point out that the middle was more accurate than the edges. Especially looking at the 24 row (near the wall but not too far off from drift) to 48 row (farther from the wall but still not too far from drift)*
- GPS odom
 - With our fun filter
 - How it fits with our other odometry components
- Cata system state machine
 - Message passing parallelism 😎
- Brake mode stuff
 - Velocity brake
 - Smart brake
 - Motivation. Why pure position brake is bad
- Pure pursuit
 - Idrk ask mcgee
- Autocommand upgrades
 - What it was at start of season

```

● ● ●
1 void do_auto(){
2     CommandController cc;
3
4     AutoCommand *go_way_1 = new InOrder{
5         DriveToPoint(0, 0),
6         Shoot(),
7     };
8
9     AutoCommand *go_way_1 = new InOrder{
10        DriveToPoint(100, 0),
11        Shoot(),
12    };
13
14
15     cc.add(new InOrder{
16        new SetOdom({0, 0, 90}),
17        new DriveForward(10),
18        new TurnDegrees(90),
19
20        // Do two things at once
21        new AllFinish({
22            DriveForward(10),
23            InOrder{
24                Delay(100),
25                IntakeUp(),
26            },
27        }),
28
29
30        // Choose path wisely
31        new Branch(igo_way_1, go_way_2, [](){
32            return random()%2;
33        }),
34
35        // go back to same path
36        new DriveToPoint(50, 0),
37        new DoSomething(),
38
39    });
40
41 }
42
43

```

- Memory safety

- Explicit duplication to avoid incorrect state copying and why that matters
- Syntax niceties

```

1
2 CommandController cmd{
3     odom.SetPositionCmd({.x = 16.0, .y = 16.0, .rot = 225}),
4     printOdom,
5
6     // 1 - Turn and shoot preload
7     {
8         cata_sys.Fire(),
9         drive_sys.DriveForwardCmd(dist, REV),
10        DelayCommand(300),
11        cata_sys.StopFiring(),
12        cata_sys.IntakeFully(),
13    },
14    // 2 - Turn to matchload zone & begin matchloading
15    drive_sys.DriveForwardCmd(dist + 2, vex::fwd, 0.5).with_timeout(1.5),
16
17    // Matchloading phase
18    Repeat{
19        odom.SetPositionCmd({.x = 16.0, .y = 16.0, .rot = 225}),
20
21        intakeToCata.with_timeout(1.75),
22        cata_sys.Fire(),
23        drive_sys.DriveForwardCmd(dist, REV, 0.5),
24        cata_sys.StopFiring(),
25
26        cata_sys.IntakeFully(),
27        drive_sys.TurnToHeadingCmd(load_angle, 0.5),
28
29        drive_sys.DriveForwardCmd(dist + 2, FWD, 0.2).with_timeout(1.7),
30    }
31    .until(TimeSinceStartExceeds(30)),
32 };

```

- Failed experiments in IMU fusion odom?
- Motion controller coolerness + how it interacts with pure pursuit
-

Core: Ongoing Projects

- N pod odom
- Core-rs - big things cooking
 - Bridge layer if theres anything interesting to say about it
- Vex debug board if we get it working
- vexsim?

Core: The Funny

Only maybe, this is not professional or something

- 3d renderer
- Video player
- Nes Emulator (did outreach with this cuz a middle schooler was a big fan)