Início rápido da programação da FIRST Tech Challenge

> Este documento foi desenvolvido em parceria com a equipe JUSTICE FTC TEAM #21036

Exemplo básico para FTC: Exemplo FTC

Na FTC, pode-se utilizar o Android Studio, OnBot Java e Blockly para interagir com o seu Limelight.

A maioria das aplicações requer menos de 10 linhas de código. Aqui está uma rápida visão geral do processo.

Uso básico

1. Inicialize sua Limelight3A usando o mesmo nome que você usou durante a etapa de configuração do Control Hub.
2. Chame *pipelineSwitch()* para selecionar um dos 10 pipelines que você configurou usando a interface da Web
3. Chame *start()* para iniciar a sondagem de resultados em segundo plano a 100 sondagens por segundo.

```
                                                public class Teleop extends LinearOpMode {


private Limelight3A limelight;


@Override
public void runOpMode() throws InterruptedException
{
    limelight = hardwareMap.get(Limelight3A.class, "limelight");
```

```
telemetry.setMsTransmissionInterval(11);

limelight.pipelineSwitch(0);

/*
 * Starts polling for data.
 */
limelight.start();
.
.
```

4. Chame *getLatestResult()* nos seus loops autónomos e de teleoperação para obter o último objeto LLResult
5. Utilize as funções *getTx()*, *getTy()* e *getBotpose()* do *LLResult* para guiar o seu robô.

```
while (opModeIsActive()) {
  LLResult result = limelight.getLatestResult();
  if (result != null) {
     if (result.isValid()) {
        Pose3D botpose = result.getBotpose();
        telemetry.addData("tx", result.getTx());
        telemetry.addData("ty", result.getTy());
        telemetry.addData("Botpose", botpose.toString());
        .
        .
```

Utilização avançada

1. Em casos de uso avançados podem exigir o uso das funções do *LLResult's getColorResults()*, *getFiducialResults()*, etc.

```
   // print some data for each detected target
 if (result.isValid()) {
    // Access fiducial results
    List<LLResultTypes.FiducialResult> fiducialResults = result.getFiducialResults();
    for (LLResultTypes.FiducialResult fr : fiducialResults) {
```

```
        telemetry.addData("Fiducial", "ID: %d, Family: %s, X: %.2f, Y: %.2f", fr.getFiducialId(),
 fr.getFamily(),fr.getTargetXDegrees(), fr.getTargetYDegrees());
    }

    // Access color results
    List<LLResultTypes.ColorResult> colorResults = result.getColorResults();
    for (LLResultTypes.ColorResult cr : colorResults) {
        telemetry.addData("Color", "X: %.2f, Y: %.2f", cr.getTargetXDegrees(), cr.getTargetYDegrees());
    }
}
```

2. Para obter a máxima precisão de localização 3D, chame *updateRobotOrientation()* e utilize *getBotPose_MT2().* O MegaTag2 é um localizador de robôs fundido com IMU que utiliza a imu para resolver o problema de ambiguidade que é fundamental para todos os alvos planares, como os AprilTags.

```
      while (opModeIsActive()) {
    YawPitchRollAngles orientation = imu.getRobotYawPitchRollAngles();


    telemetry.addData("Yaw (Z)", "%.2f Deg. (Heading)", orientation.getYaw(AngleUnit.DEGREES));


    limelight.updateRobotOrientation(orientation.getYaw(AngleUnit.DEGREES));
    LLResult result = limelight.getLatestResult();
    if (result != null) {
        if (result.isValid()) {
            Pose3D botpose = result.getBotpose_MT2();
            .
            .
```

Para mais informações, consultar a página de programação FTC

Revisão #8
Criado 23 setembro 2024 16:55:57 por Team stemOS
Atualizado 16 maio 2025 14:36:01 por João Vitor Loeblein