

# System Design

- Designing Teleop Controls
  - Driver Controls
  - Operator Controls
- Picking a Sensor
  - Example Seeing if a game piece is inside a mechanism

# Designing Teleop Controls

How to design controls in teleop so that they are functional and intuitive

# Driver Controls

## Basic Rules

These aren't absolute rules but you should really think about it when you break them. When in doubt listen to the driver's preference.

1. The driver shouldn't be doing more than driving and aligning the robot. Adding more mental load makes it so that the driver cannot focus as much on avoiding collisions and getting the robot around quickly.
2. The Driver should have a camera that allows for good visibility when picking up game objects (Fisheye cameras are really good here)
3. Make boolean conditions--things like is there a game piece in the robot-- a large box that turns green when it is ready or better yet use leds on the robot.
4. If using tankdrive don't use arcade drive, use curvature drive. Curvature drive behaves much better while turning, arcade drive frequently causes oversteer and overall poor handling.
5. If the driver does need to control something other than the drivetrain make their controls simple.
  1. If the system makes sense to toggle(eg. deploying an intake) have one button that sends it out and then retracts it. when released.
  2. If the system needs precise manual control, rethink the driver doing it.
  3. a command that does something until some goal state is reached should only execute while a button is held. For example, an alignment command should terminate when the robot is aligned or when the driver lets go of the button.
6. If you need more than one camera use a raspberry pi with the WPILIB image and use network tables or equivalent.(it is possible to do this many ways so keep the idea in mind.)

## Not necessarily programming but still good to consider

1. Use alignment indicators on the robot as much as possible. For example 2 pieces of orange tape on an elevator or climber can indicate when the robot is in climbing position.
2. Place the camera in a good spot with clear visibility of the intake of the robot.



# Operator Controls

## General Rules

Once again the operators preferences are key but following these rules first will give a good start.

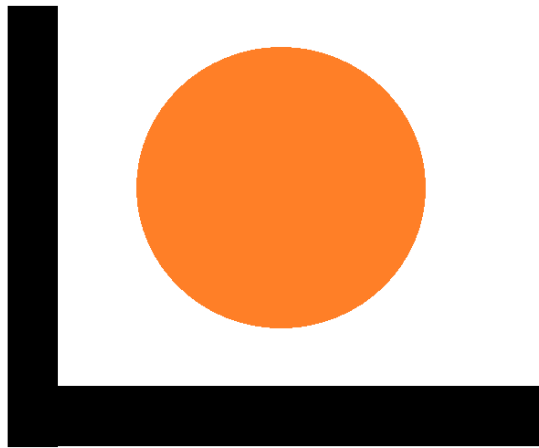
1. Any automated command needs to have a way for a manual backup. Adding a button to the controller that switches everything to manual is a good idea.
2. Automate things as much as possible
  1. Even if a sequence is triggered by a button it is better than fully manual control. For example a command that causes the intake to take in something and then move it inside of the robot is better than 2 or more buttons that do the same thing.
  2. HAVE A MANUAL BACKUP
3. Buttons should automatically execute a task
  1. if you need the operator to control something precisely and can't automate it try to make sure they only need to do one thing at a time, otherwise their performance is going to suffer, For example if the operator is climbing and the arms have to be controlled manually do not make them also have to simultaneously operate a scoring mechanism. Figure out something to reduce the load
  2. If the operator lets go of the button have it stop the command, unexpected things happen on the field and this can prevent robot damage or penalties.
4. If a subsystem has a lot of states don't make a button scroll through them. This can be confusing while operating and you frequently can't see the robot. Instead make each state a different button.
  1. for example a robot arm may need to go from the ground to a variety of scoring positions. It may seem like a good idea to bind these to gamepad up and down but if you cannot see clearly what the robot is doing you may force the robot into the wrong state. Having separate buttons solves this.

# Picking a Sensor

# Example Seeing if a game piece is inside a mechanism

Sometimes you'll need to see if a game piece is inside of the robot. This can make automating a subsystem possible but it is hard to pick a sensor.

For this example we are going to look at sensors that are good at checking if something is there.



for this example we need to check whether the orange ball is inside of the intake or not.

First option photo electric sensor

shopping.webp

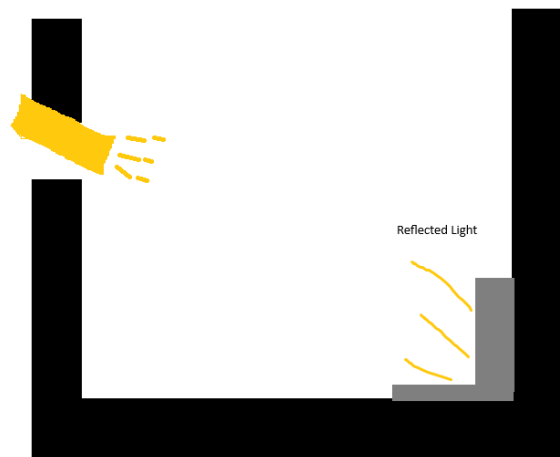
These sensors measure reflected light from a surface and provide a digital output based off of a potentiometer that is tuned by turning a knob.

Pros

- cheap and we have many
- easy to tune
- easy to program for
- reliable mechanically and functionally
- low latency sensor

## Cons

- Can be set off from lighting in a venue to prevent this point them down or at an angle and place a 3d printed focuser on them to point them at where the game piece should be.
- Can be set off from light reflected by something shiny on the robot. To prevent this put nonreflective IR tape on the robot. Below is an example on this problem. Along with the sensor
- 



Another option is to use a Color Sensor