Code release notes version 9, section 3, revision 1

Main code notes:

I. Control scheme:



II. Teleop

1. Roaming mode

Allows for control of the robot though the joystick. User can move around the game field and change the angle of shooter using the analog sticks. During this mode, the shooter wheels cannot spin and the piston cannot fire a frisbee. Intended to be used for driving around the field, loading frisbees, and climbing the pyramid. Can change to acquiring mode by pressing the actuator button

Todo:

1) Add in controls for quicker movement, or just allow it by default

2) Add in the necessary controls to climb if needed.

2. Acquiring mode

Uses vision processing to try and find a target. Runs through the vision code 4 times. If unsuccessful returns to roaming. If it is successful it sets local variables representing the angle that we are at from the target, and the distance we are from the target. It then switches to positioning code. There is no need for user controls in this section of the code, and thus there isn't any except the break button.

Todo:

Nothing

3. Positioning mode

Uses the values that were returned during the acquiring code to position the robot correctly for firing frisbees. Turns to the correct angle using the encoders on the drive wheels. Then it moves the distance specified in the driver station array. It then switches to the firing code. Again there is no user controls except for a break.

Todo:

1) Find correct adjustment constants for the production robot.

2) Possibly add in multiple PID gains for the positioning code

4. Firing mode

With the robot in position, we now need to fire frisbees. Uses the distance reading to set the speed for the shooters. Uses the shooter encoders to keep the speed of the shooter wheels at this value. When the shooter wheels are at the correct speed, we open up the piston controls. The user can use the actuator button to fire frisbees. Each time the piston fires for a set amount of time, then rests for a set amount of time. The shooter speed data is sent to the driver station. The break button returns us to the roaming code.

Todo:

Nothing

III. Autonomous

1. Initial positioning

Moves a set amount of feet forward/backward, then adjusts it's angle a set amount. Currently, the configuration is set up so that the robot will move back two feet then turn about 60° to its right. This is meant to be used in conjunction with an initial robot position that is at the back right corner of the pyramid and facing the pyramid.

Todo:

1) Allow for more initial positioning sequences

2) Allow the positioning codes to be switched from the driver station

2. Teleop code run

Does the exact same thing that the teleop code would do if you held down the actuator button. Will locate a target, position itself to the target, and fire all frisbees in the robot.

Todo:

Nothing

IIII. Driver Station

Has a controllable array for all of the positioning and speed values. These will need to be filled out for the final robot. This array is saved to, and loaded from, a spreadsheet so it is persistance. It displays all of the important information about the shooting speed, and whether the shooter speed is close enough to what it should be. Additionally, there are manual overrides and controls. There is also controls for piston firing times, and simular.

Todo:

1) Fill out the array spreadsheet

Task Lists:

Things that are needed to be done before practice day:

1) Meeting to tune the code slightly (Tuesday)

2) Meeting to explain code to drivers (Friday)

3) Driver practice (Friday + ?)

4) Change the final code to be compatible with the production robot

5) Get a list made of everything that needs to be done on practice day, hardware wise, to the robot

6) Make this list into a more detailed one (Who will be doing what, how, and when)

7) Obtain/check that we have ALL parts needed for this

Things to be done at practice day:

1) Do everything that needs to be done to the robot hardware

2) Find all of the program data needed

Required at practice day:

1) Programming hardware (computers, LAN cables, ect.)

2) Many copies of the list of hardware changes and people that will be doing it

3) Enough time on a realistic environment to determine the positioning constants and shooting distances