```python
def when_started1():
    # Prematch: starting location H, direction South, preloaded true, field preload
location 6
    # Started out making every action as fast as possible. Intake has no change in
behavior when speed is changed.
    drivetrain.set_drive_velocity(100, PERCENT)
    drivetrain.set_turn_velocity(100, PERCENT)
    arm_motor.set_velocity(100, PERCENT)
    intake_motor.set_velocity(60, PERCENT)
    # Score the pre-load in the side so we can easily move to grab the ball in the blue
loading zone
    drivetrain.turn_for(RIGHT, 60, DEGREES)
    drivetrain.drive_for(REVERSE, 600, MM)
    drivetrain.turn_for(LEFT, 60, DEGREES)
    # While loop makes sure the ball is successfully grabbed by checking the optical
sensor
    while optical.is_near_object():
        intake_motor.spin(REVERSE)
    arm_motor.spin(FORWARD)
    intake_motor.stop()
    drivetrain.drive_for(REVERSE, 300, MM)
    drivetrain.drive_for(FORWARD, 400, MM)
    drivetrain.turn_for(LEFT, 30, DEGREES)
    while not optical.is_near_object():
        intake_motor.spin(FORWARD)
    wait(0.5, SECONDS)
    arm_motor.spin(REVERSE)
    # moving the arm over to score the ball is safer and easier than turning the robot.
Any bump or drag can mess with the heading of the bot and add randomness to the later
stages of the program
    drivetrain.turn_for(RIGHT, 30, DEGREES)
    drivetrain.drive_for(REVERSE, 100, MM)
    wait(2, SECONDS)
    while optical.is_near_object():
        intake_motor.spin(REVERSE)
    arm_motor.spin(FORWARD)
    intake_motor.stop()
    wait(1, SECONDS)
    drivetrain.drive_for(REVERSE, 400, MM)
    drivetrain.drive_for(FORWARD, 300, MM)
    drivetrain.turn_to_heading(60, DEGREES)
    drivetrain.drive(FORWARD)
    # we use the gps sensor to ensure that we are are line up with the next ball
    while not gps.y_position(MM) < -1500:
        wait(5, MSEC)
    drivetrain.turn_to_heading(90, DEGREES)
    intake_motor.spin(FORWARD)
```

```python
    drivetrain.drive(FORWARD)
    # after moving to grab the ball under the elevation bar, we use the optical sensor
to ensure that we grab it.
    while not optical.is_near_object():
        wait(5, MSEC)
    intake_motor.stop()
    arm_motor.spin(REVERSE)
    drivetrain.drive(REVERSE)
    while not gps.x_position(MM) > 900:
        wait(5, MSEC)
    drivetrain.turn_to_heading(180, DEGREES)
    drivetrain.set_heading(0, DEGREES)
    drivetrain.set_rotation(0, DEGREES)
    drivetrain.drive_for(FORWARD, 600, MM)
    # we take the ball that we have and orient ourselves into the center of the
offensive zone so we can create a looping code to grab the two balls directly behind
us
    drivetrain.turn_to_heading(-45, DEGREES)
    drivetrain.drive(FORWARD)
    while not 0 < gps.y_position(MM):
        wait(5, MSEC)
    drivetrain.turn_to_heading(-90, DEGREES)
    drivetrain.drive(REVERSE)
    while not gps.x_position(MM) > 500:
        wait(5, MSEC)
    drivetrain.stop()
    for repeat_count in range(2):
        while optical.is_near_object():
            intake_motor.spin(REVERSE)
        arm_motor.spin(FORWARD)
        intake_motor.stop()
        wait(1.9, SECONDS)
        drivetrain.drive_for(REVERSE, 375, MM)
        intake_motor.spin(FORWARD)
        drivetrain.drive(FORWARD)
        while not optical.is_near_object():
            wait(5, MSEC)
        intake_motor.stop()
        arm_motor.spin(REVERSE)
        drivetrain.drive(REVERSE)
        while not gps.x_position(MM) > 500:
            wait(5, MSEC)
        drivetrain.stop()
        wait(1.5, SECONDS)
    # after looping, we augment the existing code to grab two of the other balls north
and south of the two we already acquired
    while optical.is_near_object():
```

```
        intake_motor.spin(REVERSE)
arm_motor.spin(FORWARD)
intake_motor.stop()
wait(2, SECONDS)
drivetrain.drive_for(REVERSE, 350, MM)
drivetrain.drive_for(FORWARD, 350, MM)
drivetrain.turn_to_heading(0, DEGREES)
drivetrain.drive(FORWARD)
while not gps.y_position(MM) > 550:
    wait(5, MSEC)
drivetrain.turn_for(LEFT, 90, DEGREES)
intake_motor.spin(FORWARD)
drivetrain.drive(FORWARD)
while not optical.is_near_object():
    wait(5, MSEC)
intake_motor.stop()
arm_motor.spin(REVERSE)
drivetrain.drive(REVERSE)
while not gps.x_position(MM) > 500:
    wait(5, MSEC)
drivetrain.stop()
drivetrain.turn_to_heading(180, DEGREES)
drivetrain.drive(FORWARD)
while not 0 > gps.y_position(MM):
    wait(5, MSEC)
drivetrain.turn_for(RIGHT, 90, DEGREES)
while optical.is_near_object():
    intake_motor.spin(REVERSE)
arm_motor.spin(FORWARD)
intake_motor.stop()
wait(2, SECONDS)
drivetrain.drive_for(REVERSE, 350, MM)
drivetrain.drive_for(FORWARD, 350, MM)
drivetrain.turn_to_heading(180, DEGREES)
drivetrain.drive(FORWARD)
while not -550 > gps.y_position(MM):
    wait(5, MSEC)
drivetrain.turn_for(RIGHT, 90, DEGREES)
intake_motor.spin(FORWARD)
drivetrain.drive(FORWARD)
while not optical.is_near_object():
    wait(5, MSEC)
intake_motor.stop()
arm_motor.spin(REVERSE)
drivetrain.drive(REVERSE)
while not gps.x_position(MM) > 500:
    wait(5, MSEC)
```

```python
    drivetrain.stop()
    drivetrain.turn_to_heading(0, DEGREES)
    drivetrain.drive(FORWARD)
    while not gps.y_position(MM) > 0:
        wait(5, MSEC)
    drivetrain.turn_for(LEFT, 90, DEGREES)
    while optical.is_near_object():
        intake_motor.spin(REVERSE)
    arm_motor.spin(FORWARD)
    intake_motor.stop()
    wait(0.5, SECONDS)
    # we then move to grab the ball under the north elevation bar
    drivetrain.drive_for(REVERSE, 400, MM)
    drivetrain.drive_for(FORWARD, 150, MM)
    drivetrain.turn_for(LEFT, 85, DEGREES)
    drivetrain.drive(REVERSE)
    while not 1450 < gps.y_position(MM):
        wait(5, MSEC)
    drivetrain.stop()
    drivetrain.turn_to_heading(280, DEGREES)
    drivetrain.drive(FORWARD)
    intake_motor.spin(FORWARD)
    while not optical.is_near_object():
        wait(5, MSEC)
    arm_motor.spin(REVERSE)
    intake_motor.stop()
    drivetrain.stop()
    drivetrain.turn_to_heading(270, DEGREES)
    drivetrain.drive(REVERSE)
    while not 850 < gps.x_position(MM):
        wait(5, MSEC)
    drivetrain.turn_to_heading(190, DEGREES)
    drivetrain.drive(FORWARD)
    # we score this ball in a position that allows us to easily score the other preload
on the opposite goal
    while not gps.y_position(MM) < -250:
        wait(5, MSEC)
    drivetrain.turn_to_heading(270, DEGREES)
    while optical.is_near_object():
        intake_motor.spin(REVERSE)
    intake_motor.stop()
    drivetrain.drive_for(REVERSE, 200, MM)
    # scoring the final ball is simple because we already aligned ourselves when we
scored the previous ball
    drivetrain.drive_for(FORWARD, 1900, MM)
    drivetrain.drive_for(REVERSE, 200, MM)
```

```
vr_thread(when_started1)
```