```
                // Find angle to aim
                relativeAngle = getRelativeAngle();
                // Aim at that angle
                turnRelative(relativeAngle,-1);
                // Once we're aimed, go to step 1
                if (abs(relativeAngle) < 1)
                    fireState = 1;
                break;
            default:
                break;
        }

        // Manual button to move catapult up
        if (controller.get_digital(BTN_CAT_UP)) {
            // Clear auto-flags and set speed
            catSpeed = -127;
            fireState = -1;
            catSeek = -1;
        }
        // Manual button to move catapult down
        if (controller.get_digital(BTN_CAT_DOWN)) {
            // Clear auto-flags and set speed
            catSpeed = 127;
            fireState = -1;
            catSeek = -1;
        }

        // Button to fire w/o aiming
        if (controller.get_digital(BTN_FIRE)) {
            // Set flags, go to fireState 1
            aimFire = false;
            fireState = 1;
        }
        // Button to fire w/ aiming
        if (controller.get_digital(BTN_FIRE_AIM)) {
            // Set flags, go to fireState 20
            aimFire = true;
            fireState = 20;
        }

        // Button to abort auto-funtion
        if (controller.get_digital(BTN_ABORT)) {
            // Clear auto-flags
            fireState = -1;
            catSeek = -1;
        }

        // Button to draw catapult w/o firing
        if (controller.get_digital(BTN_TOGGLE)) {
            // Go to fireState 10
            fireState = 10;
        }


        // If we want to seek a position, run motors proportional to distance
        if (catSeek >= 0) {
            catSpeed = (catSeek - catPos);
        }


        // Set motors on catapult
        cat_1.move_voltage(catSpeed * 12000 / 127);
        cat_2.move_voltage(catSpeed * 12000 / 127);


        pros::delay(20);   // don't hog cpu
    }
}
```