

```

% Parameters
roadWidth = 30; % Width of the track
coneColors = ['r', 'g', 'b']; % Cone colors (red, green, blue)
carSize = 5; % Size of the car dot

% Define racetrack points (closed loop)
trackX = [0, 50, 100, 150, 150, 100, 50, 0, 0];
trackY = [0, 0, 50, 100, 150, 200, 200, 150, 0];

% Interpolate to smooth the track
t = linspace(0, 1, length(trackX));
smoothT = linspace(0, 1, 500);
smoothTrackX = spline(t, trackX, smoothT);
smoothTrackY = spline(t, trackY, smoothT);

% Define track boundaries
theta = atan2(diff([smoothTrackY, smoothTrackY(end)]), diff([smoothTrackX,
smoothTrackX(end)]));
trackLeftX = smoothTrackX + roadWidth / 2 * sin(theta);
trackLeftY = smoothTrackY - roadWidth / 2 * cos(theta);
trackRightX = smoothTrackX - roadWidth / 2 * sin(theta);
trackRightY = smoothTrackY + roadWidth / 2 * cos(theta);

% Generate a random number of cones (between 15 and 30)
numCones = randi([15, 30]); % Random number of cones
coneX = zeros(1, numCones);
coneY = zeros(1, numCones);
coneColorIdx = randi(length(coneColors), 1, numCones); % Random cone colors

% Place cones randomly on the road
for i = 1:numCones
    idx = randi(length(smoothTrackX)); % Random index along the track
    coneX(i) = (trackLeftX(idx) + trackRightX(idx)) / 2 + (rand - 0.5) *
roadWidth / 2;
    coneY(i) = (trackLeftY(idx) + trackRightY(idx)) / 2 + (rand - 0.5) *
roadWidth / 2;
end

% Plot the racetrack
figure;
hold on;
fill([trackLeftX, fliplr(trackRightX)], [trackLeftY, fliplr(trackRightY)],
[0.8, 0.8, 0.8]); % Track area
plot(smoothTrackX, smoothTrackY, 'k--', 'LineWidth', 1); % Centerline
plot(trackLeftX, trackLeftY, 'k', 'LineWidth', 2); % Left boundary
plot(trackRightX, trackRightY, 'k', 'LineWidth', 2); % Right boundary

% Add start/finish line
startX = [trackLeftX(1), trackRightX(1)];
startY = [trackLeftY(1), trackRightY(1)];

```

```

plot(startX, startY, 'r-', 'LineWidth', 3); % Start/finish line
text(mean(startX), mean(startY), 'Start/Finish', 'Color', 'r', 'FontSize',
10, 'HorizontalAlignment', 'center');

% Plot cones with random colors
for i = 1:numCones
    scatter(coneX(i), coneY(i), 30, coneColors(coneColorIdx(i)), 'filled');
% Cone
end

% Simulate car movement
carPos = [smoothTrackX(1); smoothTrackY(1)]; % Initial position
carPathX = carPos(1); % Store car path
carPathY = carPos(2);

for t = 2:length(smoothTrackX)
    % Move the car forward along the centerline
    carPos(1) = smoothTrackX(t);
    carPos(2) = smoothTrackY(t);

    % Check proximity to cones and avoid collisions
    for i = 1:numCones
        distance = sqrt((carPos(1) - coneX(i))^2 + (carPos(2) - coneY(i))^2);
        if distance < 10
            if coneColors(coneColorIdx(i)) == 'r'
                text(coneX(i), coneY(i) + 2, 'Red', 'Color', 'k',
'FontSize', 5, 'HorizontalAlignment', 'center');
            elseif coneColors(coneColorIdx(i)) == 'g'
                text(coneX(i), coneY(i) + 2, 'Green', 'Color', 'k',
'FontSize', 5, 'HorizontalAlignment', 'center');
            elseif coneColors(coneColorIdx(i)) == 'b'
                text(coneX(i), coneY(i) + 2, 'Blue', 'Color', 'k',
'FontSize', 5, 'HorizontalAlignment', 'center');
            end
            if carPos(1) < coneX(i)
                carPos(1) = carPos(1) - 2; % Move left
            else
                carPos(1) = carPos(1) + 2; % Move right
            end
            if carPos(2) < coneY(i)
                carPos(2) = carPos(2) - 2; % Move down
            else
                carPos(2) = carPos(2) + 2; % Move up
            end
        end
    end

    % Stay within track boundaries
    leftDist = sqrt((carPos(1) - trackLeftX(t))^2 + (carPos(2) -
trackLeftY(t))^2);

```

```

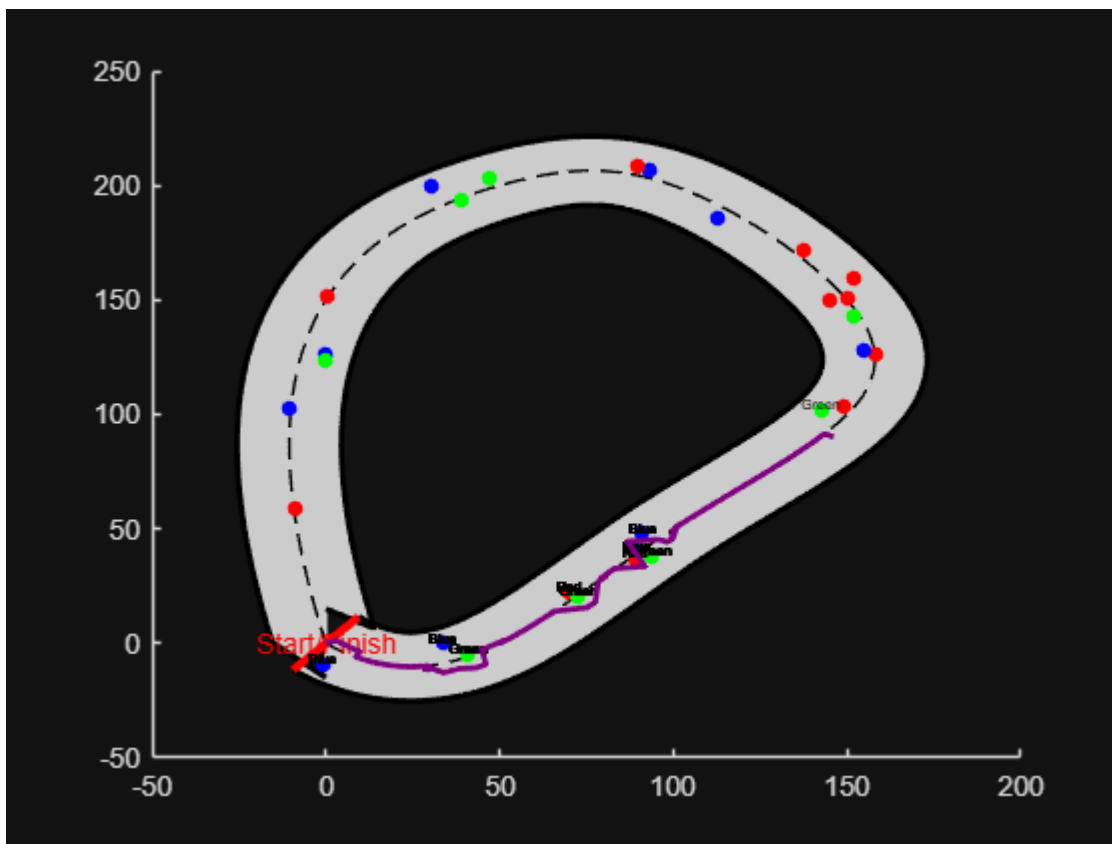
rightDist = sqrt((carPos(1) - trackRightX(t))^2 + (carPos(2) -
trackRightY(t))^2);

if leftDist < 5
    carPos(1) = carPos(1) + 3;
    carPos(2) = carPos(2) + 3;
elseif rightDist < 5
    carPos(1) = carPos(1) - 3;
    carPos(2) = carPos(2) - 3;
end

% Update car path
carPathX = [carPathX, carPos(1)];
carPathY = [carPathY, carPos(2)];

% Plot the car path
plot(carPathX, carPathY, 'Color', [0.5, 0, 0.5], 'LineWidth', 2); %
Purpl car path
pause(0.1); % Slow car movement
end

```



```

hold off;
title('F1 Race Track with Random Cones and Real-Time Car Navigation');

```