

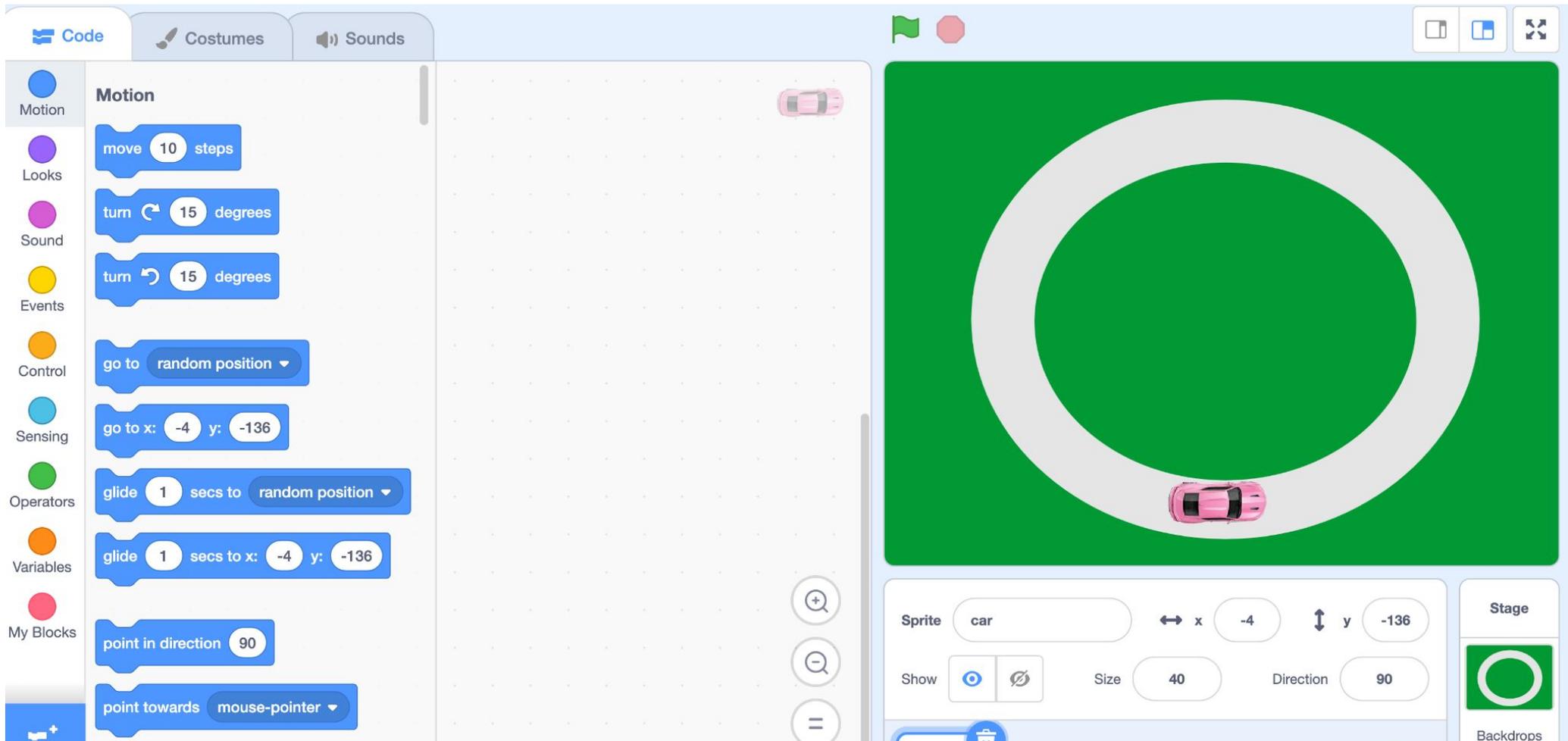
Are you up for a challenge?



→ Go to Scratch: Click [here](#)

The image shows the Scratch programming environment. On the left, the 'Code' tab is active, displaying a list of motion blocks: 'move 10 steps', 'turn 15 degrees' (clockwise and counter-clockwise), 'go to random position', 'go to x: -4 y: -136', 'glide 1 secs to random position', 'glide 1 secs to x: -4 y: -136', 'point in direction 90', and 'point towards mouse-pointer'. The central workspace shows a pink car sprite on a green stage with a white circular track. The car is positioned at the bottom of the track. On the right, the 'Sprite' panel shows the 'car' sprite selected, with its position set to x: -4 and y: -136, and its direction set to 90 degrees. The 'Stage' panel shows a green background with a white circular track.

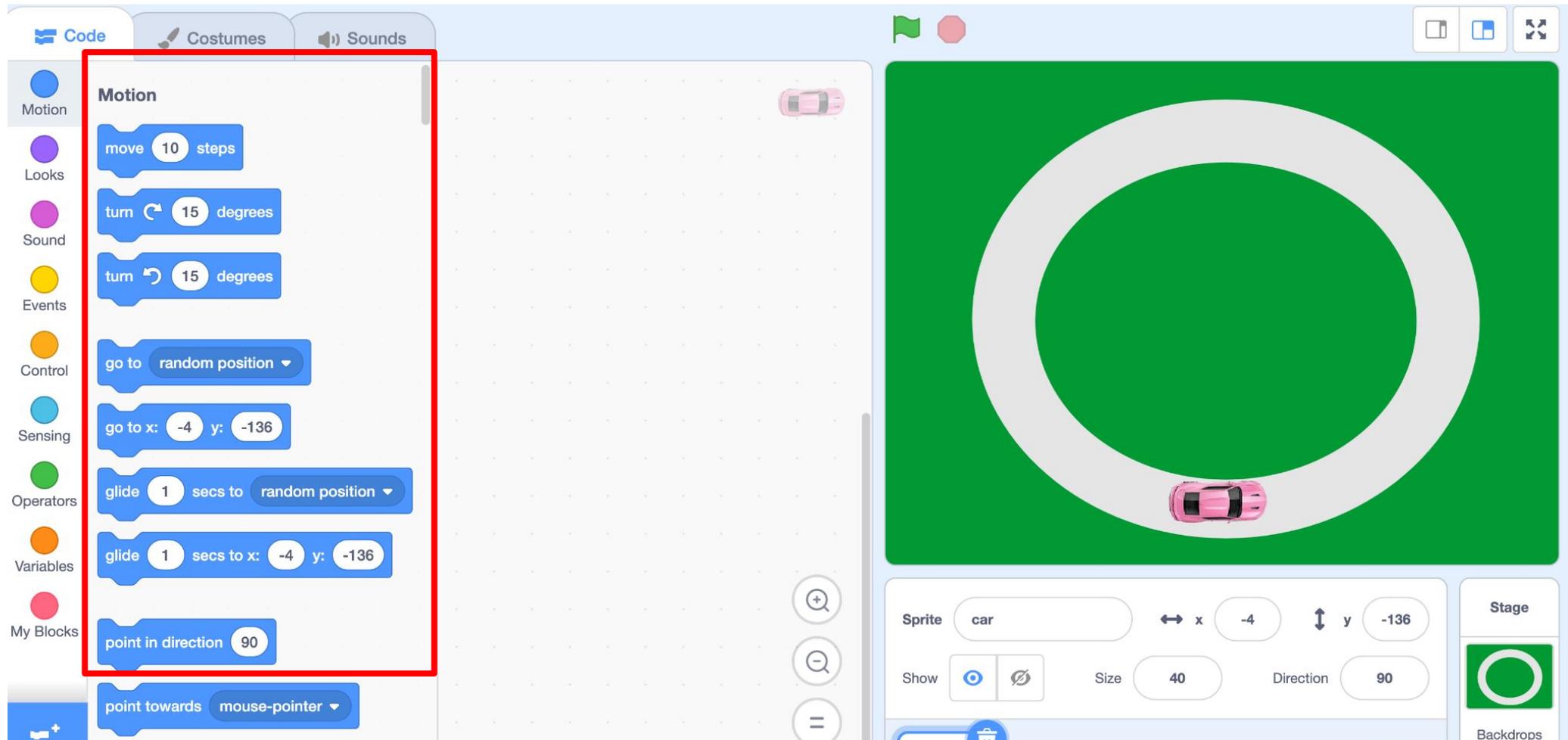
This is Scratch. Scratch is a website which you can use to program things!



Programming is a little like building a recipe for a computer. You program 'code'. And the computer will follow that code step by step.

The image shows the Scratch programming environment. On the left, the 'Code' tab is active, displaying a list of motion blocks: 'move 10 steps', 'turn 15 degrees' (clockwise and counter-clockwise), 'go to random position', 'go to x: -4 y: -136', 'glide 1 secs to random position', 'glide 1 secs to x: -4 y: -136', 'point in direction 90', and 'point towards mouse-pointer'. The main workspace shows a pink car sprite on a green stage with a white circular track. The car is positioned at the bottom of the track. The bottom right panel shows the 'Sprite' settings for the 'car' sprite, with 'x' at -4, 'y' at -136, 'Size' at 40, and 'Direction' at 90. The 'Stage' panel shows a green background with a white circle.

These are examples of 'steps' or 'commands' you can give to the computer. For example: Move 10 steps.



The image shows the Scratch code editor interface. On the left, the 'Code' tab is active, and the 'Motion' category is selected. A red box highlights the following motion blocks:

- move 10 steps
- turn 15 degrees (clockwise)
- turn 15 degrees (counter-clockwise)
- go to random position
- go to x: -4 y: -136
- glide 1 secs to random position
- glide 1 secs to x: -4 y: -136
- point in direction 90

Below the highlighted blocks, the 'point towards mouse-pointer' block is partially visible. On the right, the stage area shows a pink car sprite on a green circular track. The bottom right panel displays the sprite's current position (x: -4, y: -136) and direction (90 degrees).

Let's try moving the car!

The image shows the Scratch code editor interface. On the left, the 'Motion' category is selected in the block palette. The code area contains a single 'go to x: -4 y: -136' block. The stage on the right features a green background with a large white circular track. A pink car sprite is positioned at the bottom of the track. The bottom-right panel shows the 'Sprite' section with 'car' selected, and the 'Stage' section with a green circular backdrop. The 'x' coordinate is set to -4 and the 'y' coordinate is set to -136. The 'Size' is 40 and the 'Direction' is 90.

Code:

```
go to x: -4 y: -136
```

Sprite: car, x: -4, y: -136, Size: 40, Direction: 90

Stage: Green circular backdrop

Click on Events

The image shows the Scratch IDE interface. On the left, the 'Code' tab is active, and the 'Events' category is highlighted with a red box. The 'Motion' category is also visible. The main workspace shows a pink car sprite on a green stage with a white circular path. The bottom right panel shows the 'Sprite' and 'Stage' settings.

Code Panel:

- Motion:** move 10 steps, turn 15 degrees, turn 15 degrees, go to random position, go to x: -4 y: -136, glide 1 secs to random position, glide 1 secs to x: -4 y: -136, point in direction 90, point towards mouse-pointer.

Sprite Panel:

- Sprite: car
- x: -4, y: -136
- Show:
- Size: 40
- Direction: 90

Stage Panel:

- Stage:
- Backdrops:

Now you see this:

The screenshot displays the Scratch IDE interface for a project titled "Self-Driving Car | Les voit..." by user "llc-toronto". The top navigation bar includes the Scratch logo, a globe icon, and menu options: File, Edit, Mode, Tutorials, and a "See Project Page" button. On the right side of the top bar, there are links for "Join Scratch" and "Sign in".

The main workspace is divided into three sections:

- Code Area (Left):** Contains a vertical sidebar with category icons (Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, My Blocks) and a script area with the following blocks:
 - Events: "when clicked" (green flag), "when space key pressed", "when this sprite clicked".
 - Control: "when backdrop switches to backdrop1", "wait 1 seconds", "repeat 10".
 - Operators: "when loudness > 10".
 - Variables: "when I receive message1", "broadcast message1", "broadcast message1 and wait".
- Stage Area (Right):** Shows a green circular track with a pink car sprite positioned at the bottom. The stage background is green.
- Sprite Panel (Bottom Right):** Displays the selected "car" sprite with the following properties:
 - Sprite: car
 - x: -4, y: -136
 - Show: (visible), (invisible)
 - Size: 40
 - Direction: 90

Now you see this:

The screenshot displays the Scratch IDE interface. At the top, the title bar reads "Self-Driving Car | Les voit..." by llc-toronto. The "Code" tab is active, showing a script area with several event blocks. The first block, "when green flag clicked", is highlighted with a red box. Below it are "when space key pressed", "when this sprite clicked", and "when backdrop switches to backdrop1". The "Events" category on the left sidebar is selected. The stage area shows a green backdrop with a white circular track and a pink car sprite. The "Sprite" panel at the bottom right shows the "car" sprite with x: -4 and y: -136. The "Backdrops" panel shows a green backdrop with a white circle and a count of 3.

This command says:
Start running the code
when we click on
the green flag.

Now you see this:

The image shows the Scratch IDE interface. At the top, the title bar reads "Self-Driving Car | Les voit..." by llc-toronto. The left sidebar contains a palette of block categories: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The "Events" category is selected, and a "when clicked" block is highlighted with a red box. A red arrow points from this block to a large, empty workspace area, also outlined in red. Text next to the arrow says "Drag this block to this area". The workspace contains a green circular backdrop with a white ring and a pink car sprite at the bottom. The bottom right of the screen shows the sprite and stage properties panels.

Scratch

File Edit Mode Tutorials Self-Driving Car | Les voit... by llc-toronto See Project Page Join Scratch Sign in

Code Costumes Sounds

Events

when clicked

when space key pressed

when this sprite clicked

when backdrop switches to backdrop1

when loudness > 10

when I receive message1

broadcast message1

broadcast message1 and wait

Control

wait 1 seconds

repeat 10

Sprite car x -4 y -136

Show Size 40 Direction 90

Stage

Backdrops 3

car

Now you see this:

The image shows the Scratch IDE interface for a project titled "Self-Driving Car | Les voit..." by user "llc-toronto". The interface is divided into several sections:

- Top Bar:** Includes the Scratch logo, navigation icons (File, Edit, Mode, Tutorials), the project title, a "See Project Page" button, and "Join Scratch" / "Sign in" links.
- Code Editor:** Features tabs for "Code", "Costumes", and "Sounds". The "Code" tab is active, showing a script area with a single block: "when green flag clicked".
- Block Palette:** Located on the left, it lists various block categories: Motion, Looks, Sound, Events (highlighted), Control, Sensing, Operators, Variables, and My Blocks. The "Events" category is expanded, showing blocks like "when this sprite clicked", "when backdrop switches to", "when loudness > 10", "when I receive message1", "broadcast message1", and "broadcast message1 and wait".
- Stage:** The main workspace on the right, featuring a green backdrop with a white circular track. A pink car sprite is positioned on the track.
- Sprite Inspector:** Located below the stage, it shows the selected "car" sprite with its position (x: -4, y: -136), size (40), and direction (90).
- Backdrops:** A panel on the right side of the stage showing the current backdrop (a green circle) and a total of 3 backdrops.

Click on Motions

The image shows the Scratch IDE interface. At the top, the navigation bar includes the Scratch logo, a globe icon, and menu items: File, Edit, Mode, Tutorials, and a project title "Self-Driving Car | Les voit..." by "llc-toronto". There are also links for "See Project Page", "Join Scratch", and "Sign in".

The left sidebar contains a vertical list of categories, each with a colored circle icon: Motion (blue), Looks (purple), Sound (pink), Events (yellow), Control (orange), Sensing (light blue), Operators (green), Variables (red), and My Blocks (pink). The "Motion" category is highlighted with a red rectangular box.

The main workspace is a grid where a script is being built. It starts with a "when green flag clicked" event block. Below it, there are several "when" event blocks: "when space key pressed", "when this sprite clicked", "when backdrop switches to backdrop1", "when loudness > 10", and "when I receive message1". These are followed by "broadcast message1" and "broadcast message1 and wait" blocks. At the bottom of the script, there is a "wait 1 seconds" block and a "repeat 10" block.

The right side of the interface features a stage with a green background and a white circular track. A pink car sprite is positioned at the bottom of the track. Below the stage, the "Sprite" panel shows the selected "car" sprite with its x and y coordinates (-4, -136), size (40), and direction (90). The "Stage" panel shows the selected backdrop (a green circle) and indicates there are 3 backdrops in total.

Drag this command under your other command

The image shows the Scratch web interface for a project titled "Self-Driving Car | Les voit... by Ilc-toronto". The interface is divided into several sections:

- Top Bar:** Includes the Scratch logo, navigation menus (File, Edit, Mode, Tutorials), the project title, and options to "See Project Page", "Join Scratch", and "Sign in".
- Left Panel:** Contains the "Code" tab and a sidebar with category icons (Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, My Blocks). The "Motion" category is selected, showing a list of motion blocks.
- Code Area:** The workspace where blocks are assembled. A yellow "when green flag clicked" block is already present. A red box highlights a blue "move 10 steps" block in the sidebar, with a red arrow pointing to it being dragged into the workspace.
- Stage:** A green circular track with a pink car sprite positioned at the bottom. The car's current position is x: -4 and y: -136.
- Right Panel:** Contains the "Sprite" panel (showing the "car" sprite with its position and size/direction settings) and the "Stage" panel (showing the current backdrop).

Drag this command under your other command

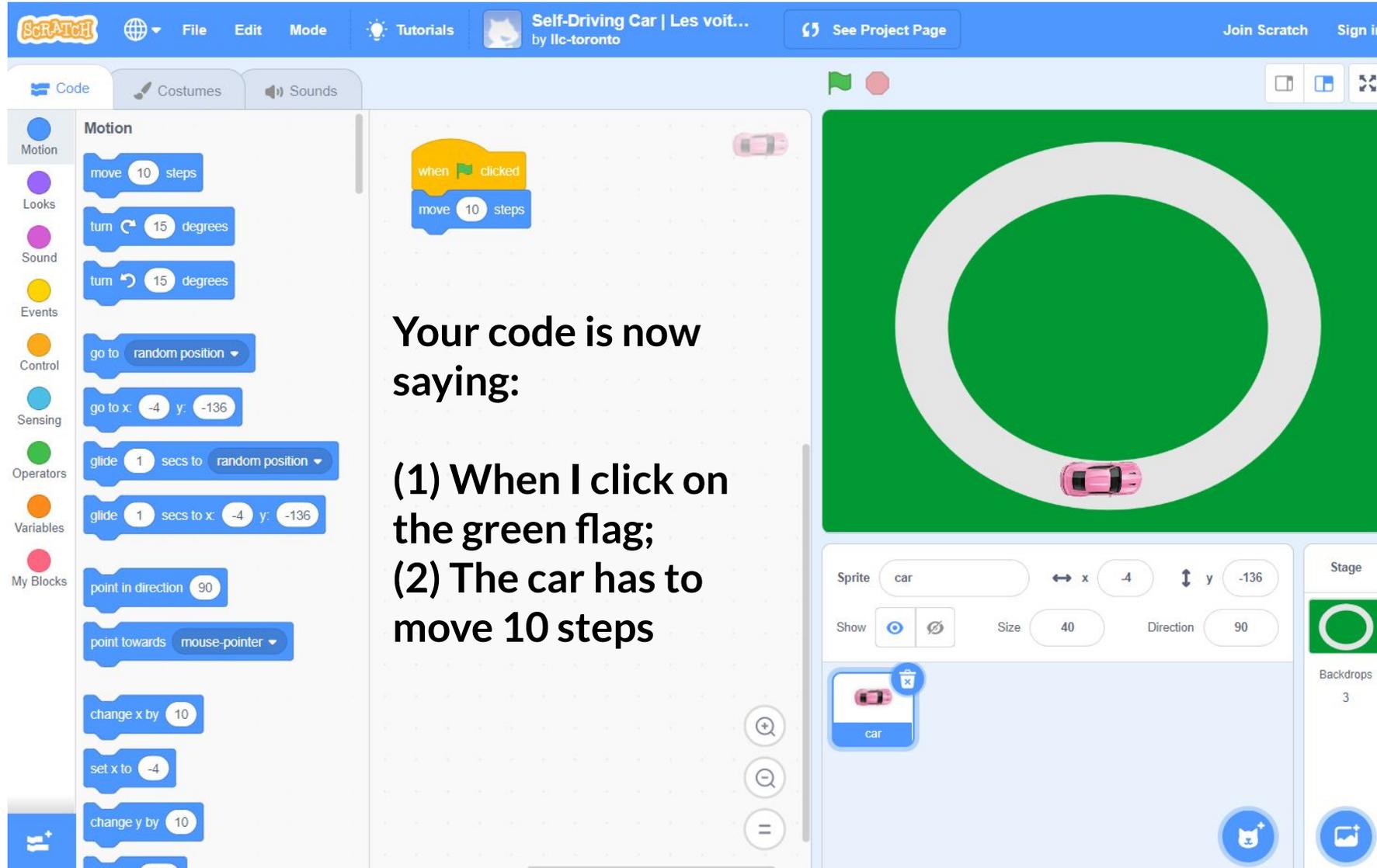
The image shows the Scratch IDE interface. At the top, the title bar reads "Self-Driving Car | Les voit..." by llc-toronto. The main workspace is divided into three sections: the left sidebar with category tabs (Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, My Blocks), the central code editor, and the right stage and sprite panels.

In the code editor, the following blocks are visible:

- when green flag clicked (yellow)
- move 10 steps (blue)

The stage panel shows a pink car sprite on a green circular track. The sprite's properties are: x: -4, y: -136, size: 40, direction: 90. The stage background is a green circle with a white border.

Drag this command under your other command



The image shows the Scratch editor interface. The top bar includes the Scratch logo, navigation menus (File, Edit, Mode, Tutorials), the project title "Self-Driving Car | Les voit..." by "llc-toronto", and links for "See Project Page", "Join Scratch", and "Sign in". The left sidebar shows the "Code" tab selected, with a list of categories: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The "Motion" category is expanded, showing various motion blocks. The main workspace displays a script with two blocks: a yellow "when green flag clicked" block and a blue "move 10 steps" block. A text overlay in the center of the workspace reads: "Your code is now saying: (1) When I click on the green flag; (2) The car has to move 10 steps". The right side of the workspace shows a stage with a green background and a white circular track. A pink car sprite is positioned at the bottom of the track. Below the stage, the "Sprite" panel shows the "car" sprite selected, with its position set to x: -4 and y: -136, and its direction set to 90. The "Stage" panel shows a green circular backdrop. The bottom right corner of the interface includes icons for adding new sprites and backdrops.

Scratch Editor Interface:

- Project Title: Self-Driving Car | Les voit... by llc-toronto
- Code Tab: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, My Blocks
- Script Area:
 - when green flag clicked
 - move 10 steps
- Stage:
 - Sprite: car
 - Position: x: -4, y: -136
 - Direction: 90
 - Backdrop: Green circle

Your code is now saying:

- (1) When I click on the green flag;
- (2) The car has to move 10 steps

Let's try it out! Click on the green flag.

The image shows the Scratch IDE interface. At the top, the project title is "Self-Driving Car | Les voit..." by user "llc-toronto". The top right corner has "Join Scratch" and "Sign in" buttons. Below the top bar, there are tabs for "Code", "Costumes", and "Sounds". The "Code" tab is active, showing a script with two blocks: "when green flag clicked" (yellow) and "move 10 steps" (blue). The stage area shows a green background with a white circular track and a pink car sprite at the bottom. The bottom right panel shows the sprite's properties: "Sprite" is "car", "x" is "-4", "y" is "-136", "Size" is "40", and "Direction" is "90". A red box highlights the green flag icon in the top right corner of the code editor area.

Did your car move? Great job!

The image shows the Scratch project editor interface for a project titled "Self-Driving Car | Les voit..." by user "llc-toronto". The project is in "Code" mode. The main stage displays a green background with a white circular track. A pink car sprite is positioned at the bottom of the track. The code area contains a single block: "when green flag clicked" followed by "move 10 steps". The "when green flag clicked" block is highlighted with a red box. The right-hand side of the interface shows the "Sprite" panel with the "car" sprite selected, and the "Stage" panel with a green circular backdrop. The "Backdrops" panel shows 3 backdrops.

What happens if you put 100 instead of 10 and press the green flag again?

The screenshot displays the Scratch IDE interface. At the top, the project title is "Self-Driving Car | Les voit..." by "llc-toronto". The "Code" tab is active, showing a script with two blocks: "when green flag clicked" and "move 10 steps". The "Green flag clicked" button in the top right corner is highlighted with a red box. The stage area shows a pink car on a green circular track. The "Sprite" panel shows the car's position at x: -4, y: -136, size: 40, and direction: 90. The "Backdrops" panel shows 3 backdrops.

And -100?

The image shows the Scratch project editor interface for a project titled "Self-Driving Car | Les voit..." by user "llc-toronto". The interface includes a top navigation bar with "File", "Edit", "Mode", "Tutorials", and "See Project Page" buttons. Below this is a toolbar with "Code", "Costumes", and "Sounds" tabs. A red box highlights a green flag icon in the toolbar, which is used to trigger the start of the script.

The code editor on the left contains the following blocks:

- Events:** "when green flag clicked"
- Motion:** "move 10 steps"
- Motion:** "turn 15 degrees" (clockwise)
- Motion:** "turn 15 degrees" (counter-clockwise)
- Control:** "go to random position"
- Sensing:** "go to x: -4 y: -136"
- Operators:** "glide 1 secs to random position"
- Operators:** "glide 1 secs to x: -4 y: -136"
- Motion:** "point in direction 90"
- Motion:** "point towards mouse-pointer"
- Operators:** "change x by 10"
- Operators:** "set x to -4"
- Operators:** "change y by 10"

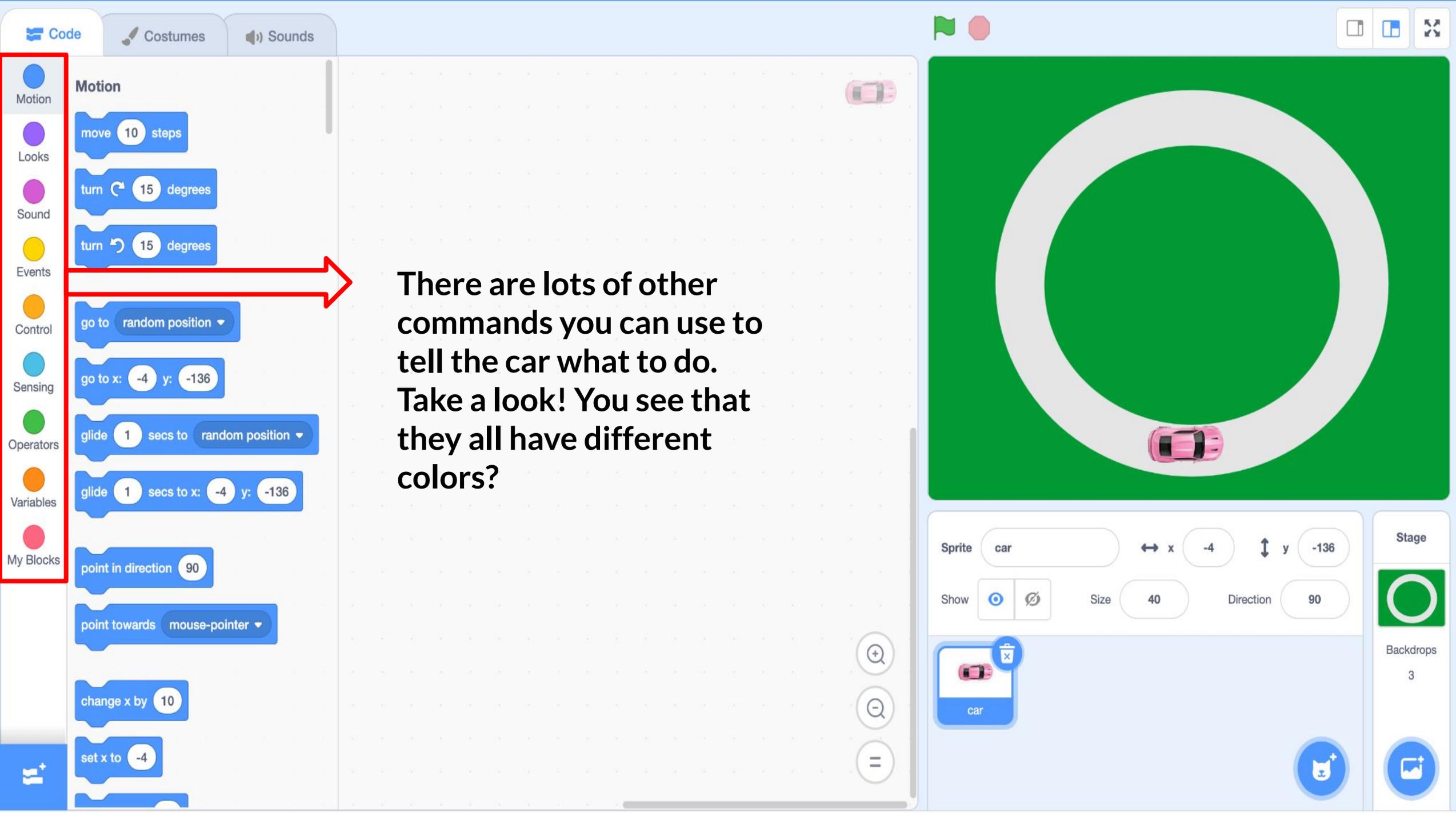
The stage on the right features a green background with a white circular track. A pink car sprite is positioned at the bottom of the track. The sprite's properties are set to "car", x: -4, y: -136, size: 40, and direction: 90. The stage also shows a "Backdrops" section with 3 backdrops.

You can now move the car!

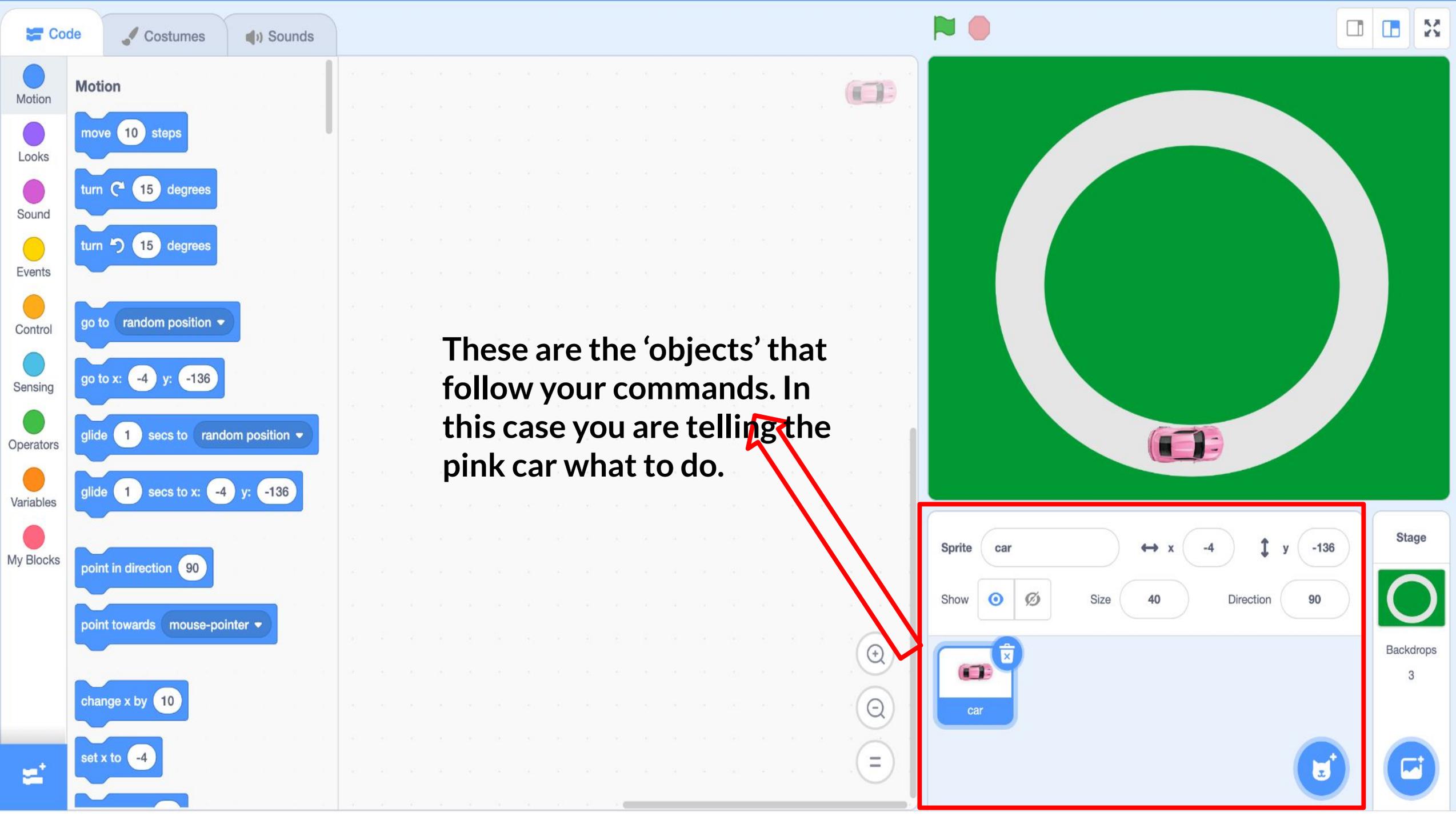
The image shows the Scratch project editor interface for a project titled "Self-Driving Car | Les voit..." by user "llc-toronto". The interface includes a top navigation bar with "File", "Edit", "Mode", "Tutorials", and "See Project Page" buttons. Below this is a toolbar with "Code", "Costumes", and "Sounds" tabs. A red box highlights a green flag icon in the toolbar, which is the "when green flag clicked" event trigger. The code editor shows a script starting with "when green flag clicked" followed by a "move 10 steps" block. The stage area displays a green background with a white circular track and a pink car sprite positioned at the bottom of the track. The bottom right panel shows the "Sprite" panel with the "car" sprite selected, and the "Stage" panel with a green circular backdrop. The "Sprite" panel also shows the car's current position (x: -4, y: -136) and direction (90 degrees).

Let's see what Scratch can do more!

The image shows the Scratch IDE interface for a project titled "Self-Driving Car | Les voit... by Ilc-toronto". The top navigation bar includes the Scratch logo, a globe icon, and menu items for File, Edit, Mode, Tutorials, and a "See Project Page" button. On the right side of the top bar, there are "Join Scratch" and "Sign in" links. Below the top bar, there are tabs for "Code", "Costumes", and "Sounds". The "Code" tab is active, showing a script area with two blocks: a yellow "when green flag clicked" event block and a blue "move 10 steps" block. The stage area displays a green background with a large white circular track. A pink car sprite is positioned at the bottom of the track. The right sidebar shows the "Sprite" panel with the "car" sprite selected, displaying its coordinates (x: -4, y: -136) and direction (90). The "Stage" panel shows a green circular backdrop. The "Backdrops" panel shows 3 backdrops. The bottom right corner of the IDE has a Scratch logo and a "Share" icon.



There are lots of other commands you can use to tell the car what to do. Take a look! You see that they all have different colors?



These are the 'objects' that follow your commands. In this case you are telling the pink car what to do.

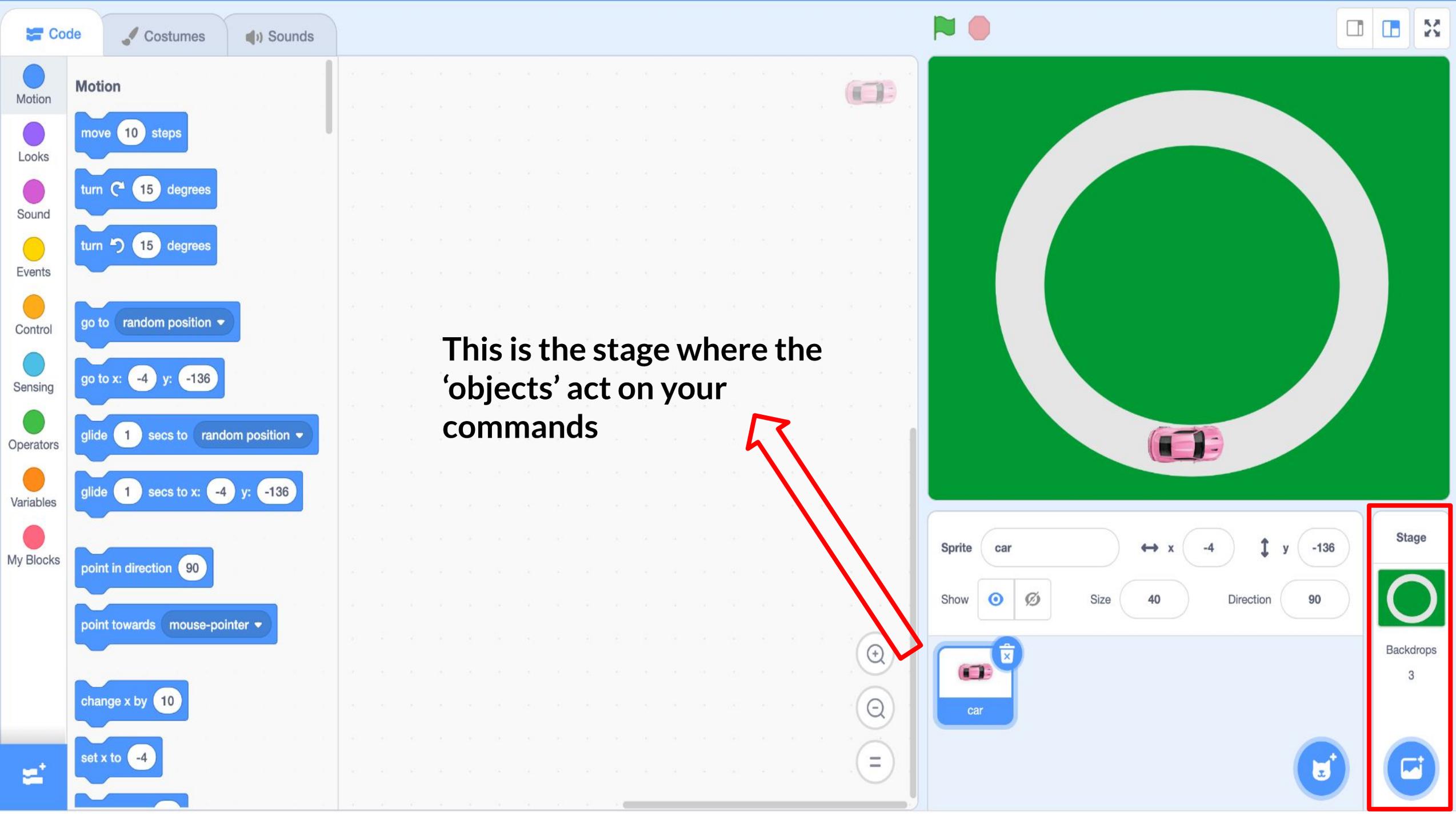
Sprite: car

x: -4 y: -136

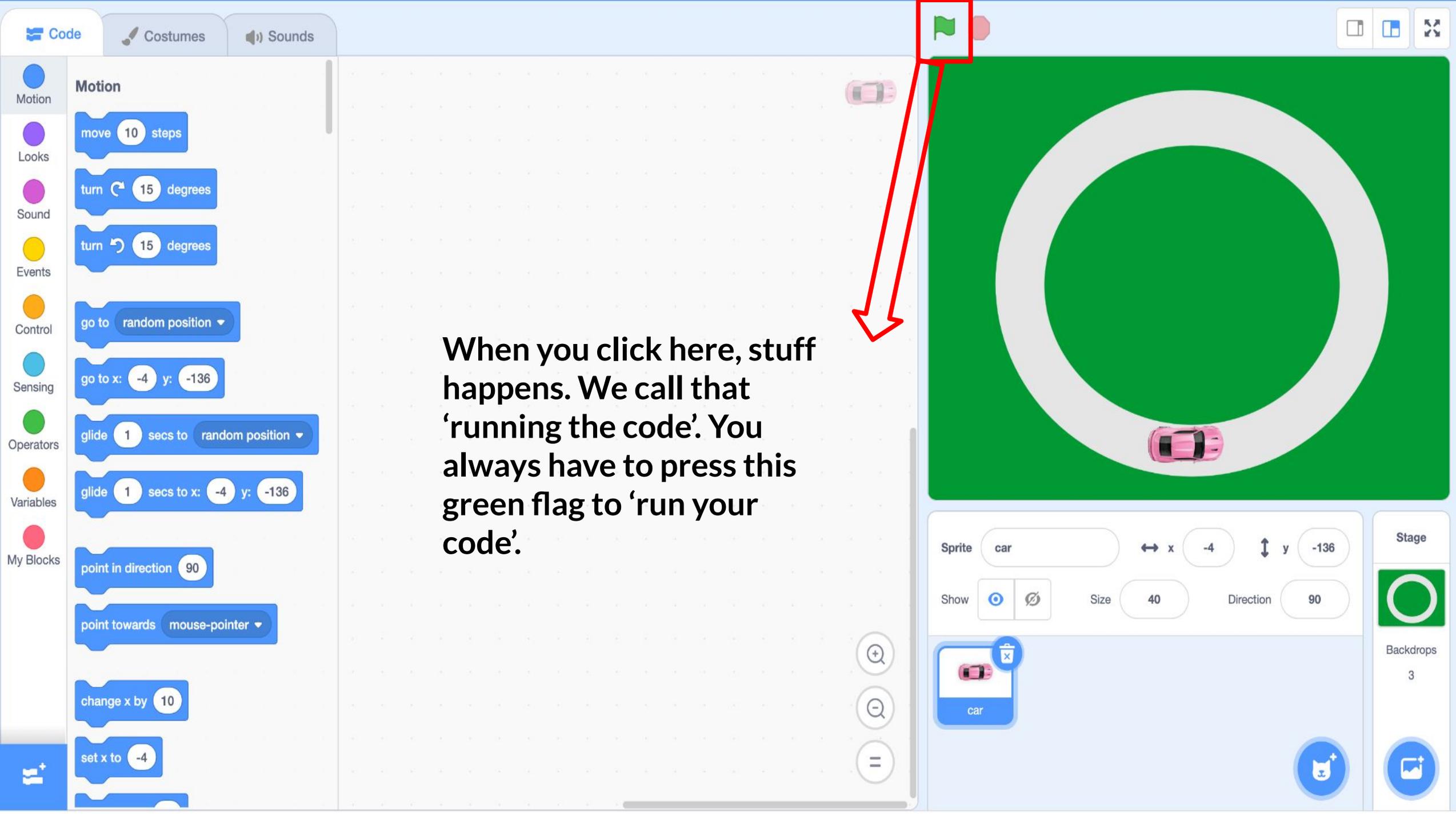
Show:

Size: 40 Direction: 90

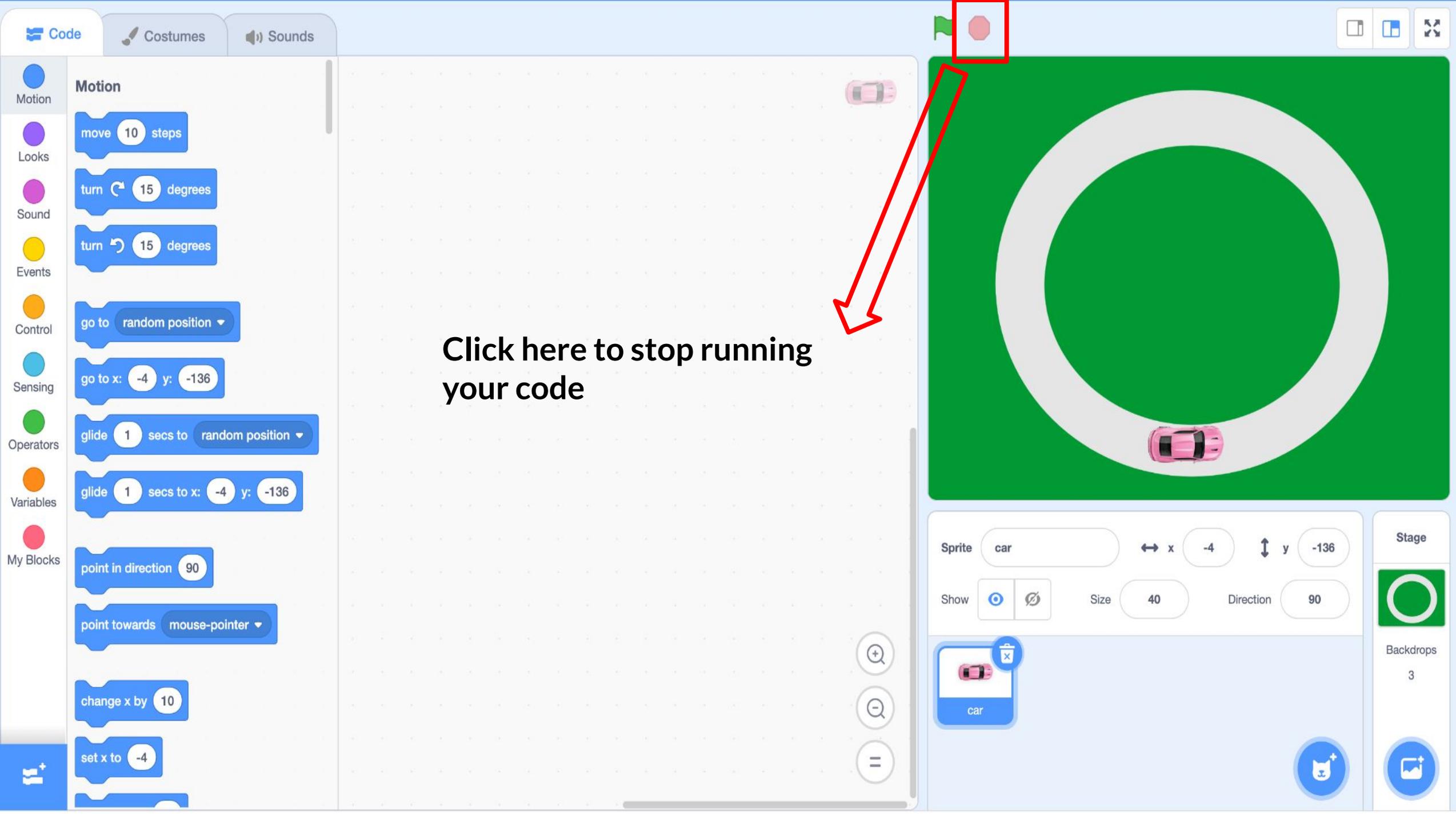
car



This is the stage where the 'objects' act on your commands



When you click here, stuff happens. We call that 'running the code'. You always have to press this green flag to 'run your code'.

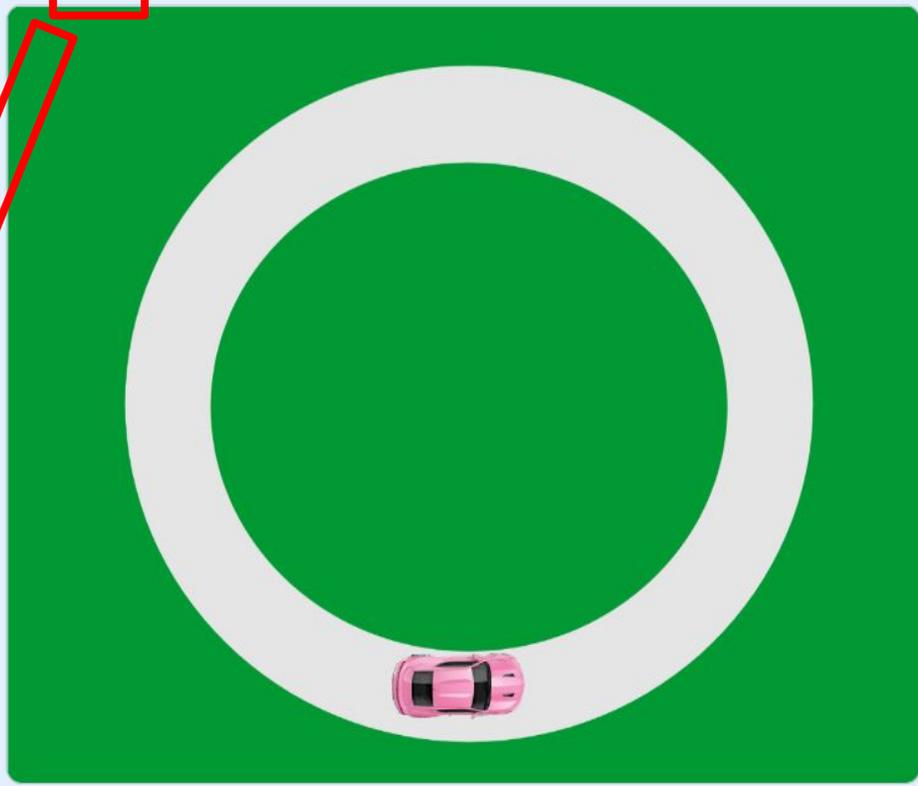


- Motion
- Looks
- Sound
- Events
- Control
- Sensing
- Operators
- Variables
- My Blocks

Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: -4 y: -136
- glide 1 secs to random position
- glide 1 secs to x: -4 y: -136
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to -4

Click here to stop running your code



Sprite: car x: -4 y: -136

Show: Size: 40 Direction: 90

Stage: Backdrops: 3

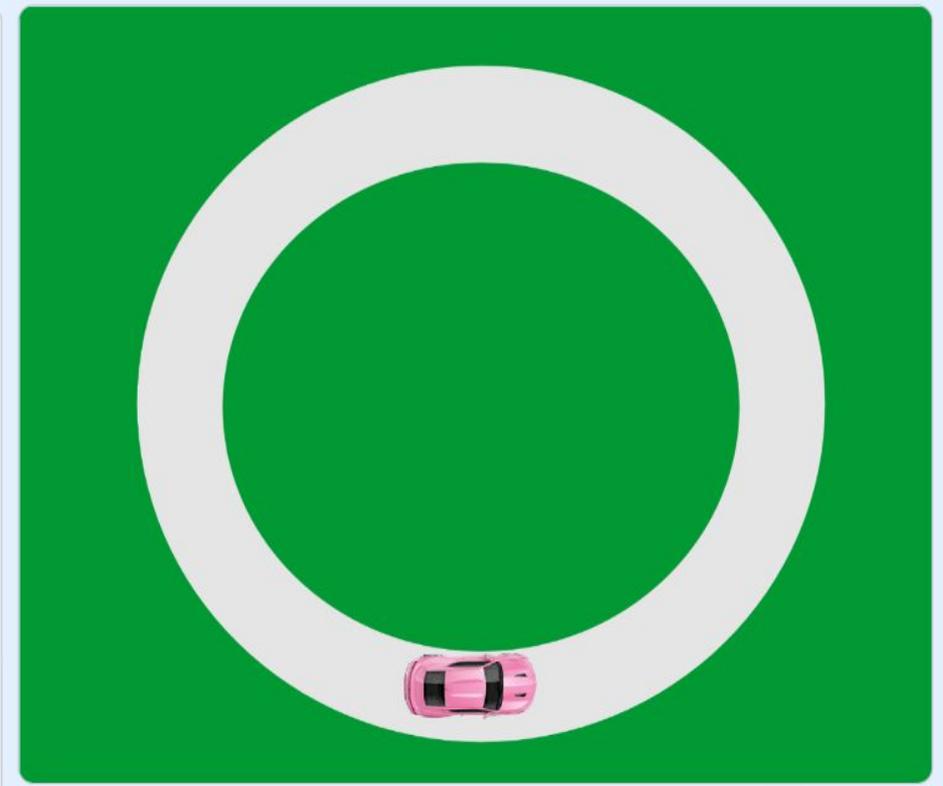
car

- Motion
- Looks
- Sound
- Events
- Control
- Sensing
- Operators
- Variables
- My Blocks

Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: -4 y: -136
- glide 1 secs to random position
- glide 1 secs to x: -4 y: -136
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to -4

Let's try to build something more difficult. Try to copy the following:



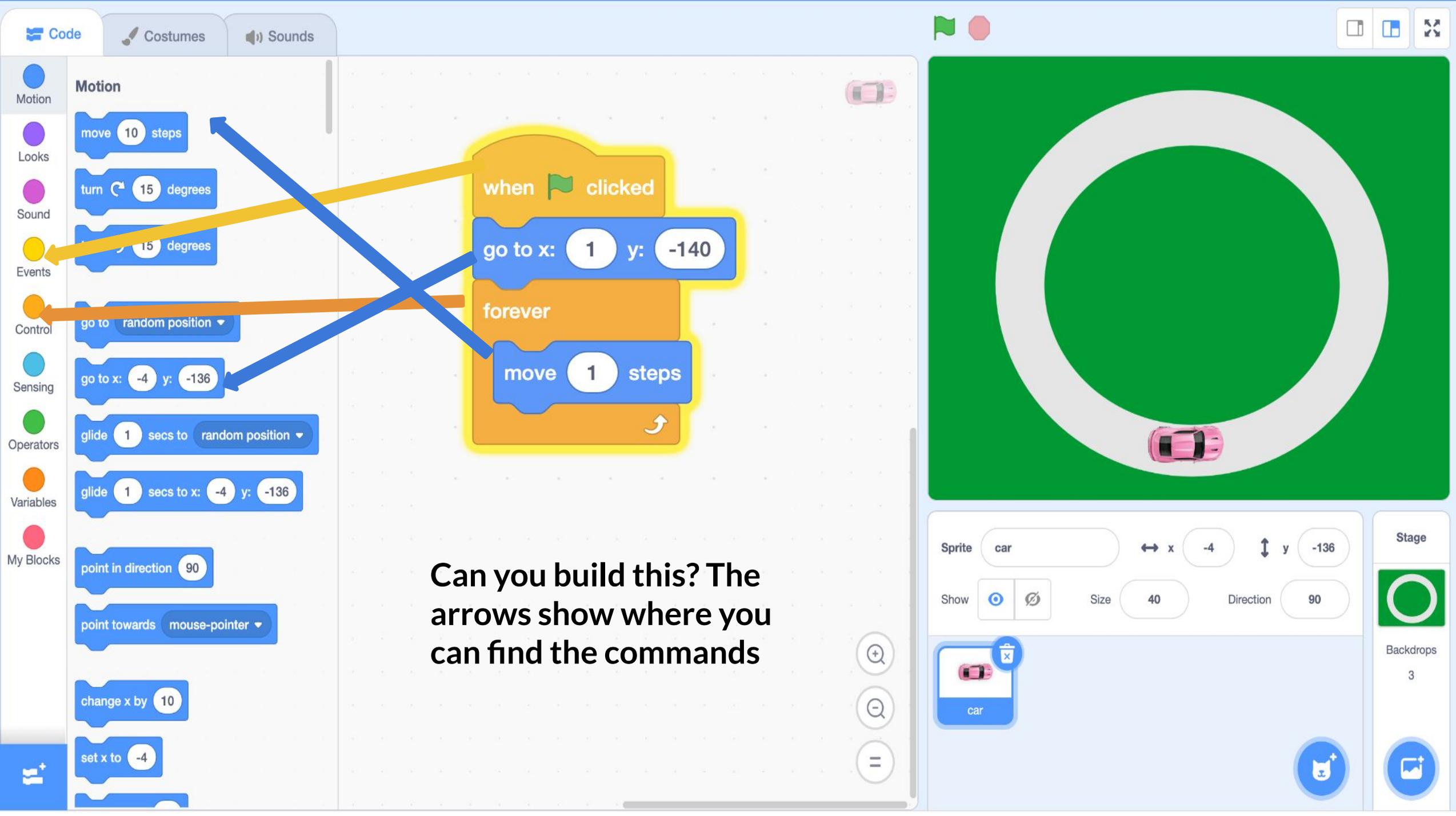
Sprite: car

x: -4 y: -136

Show:

Size: 40 Direction: 90

Backdrops: 3



Can you build this? The arrows show where you can find the commands

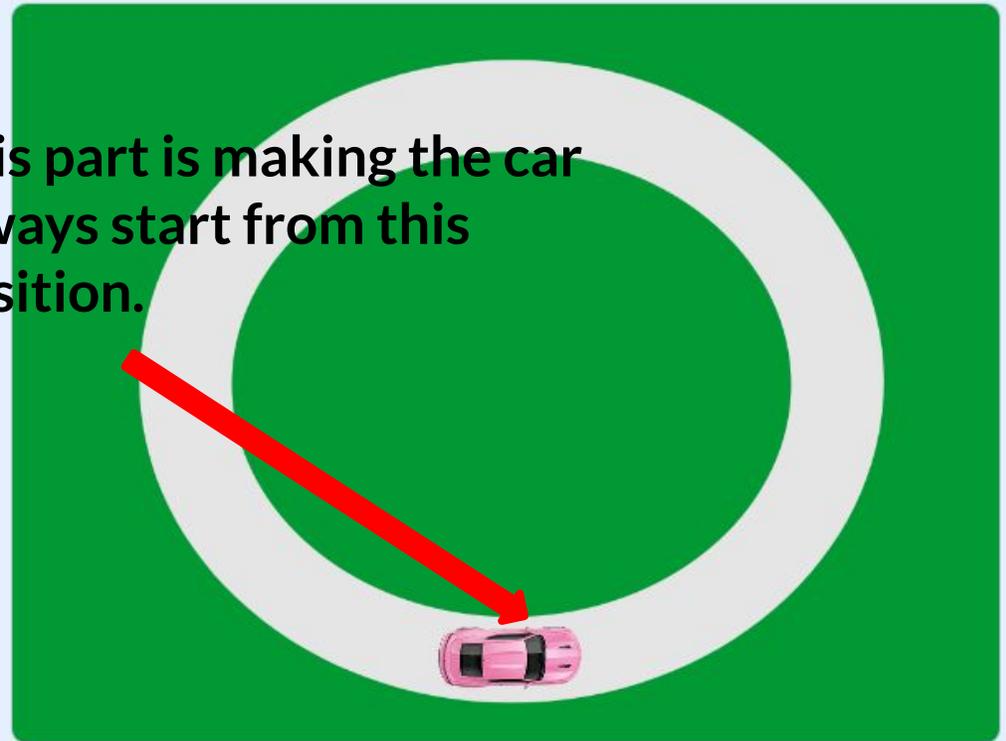
Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 1 y: -140
- glide 1 secs to random position
- glide 1 secs to x: 1 y: -140
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 1
- change y by 10

```
when green flag clicked  
go to x: 1 y: -140
```

```
forever  
move 1 steps
```

This part is making the car always start from this position.



Sprite: car

x: 1 y: -140

Show:

Size: 40 Direction: 90

Stage

Backdrops: 3

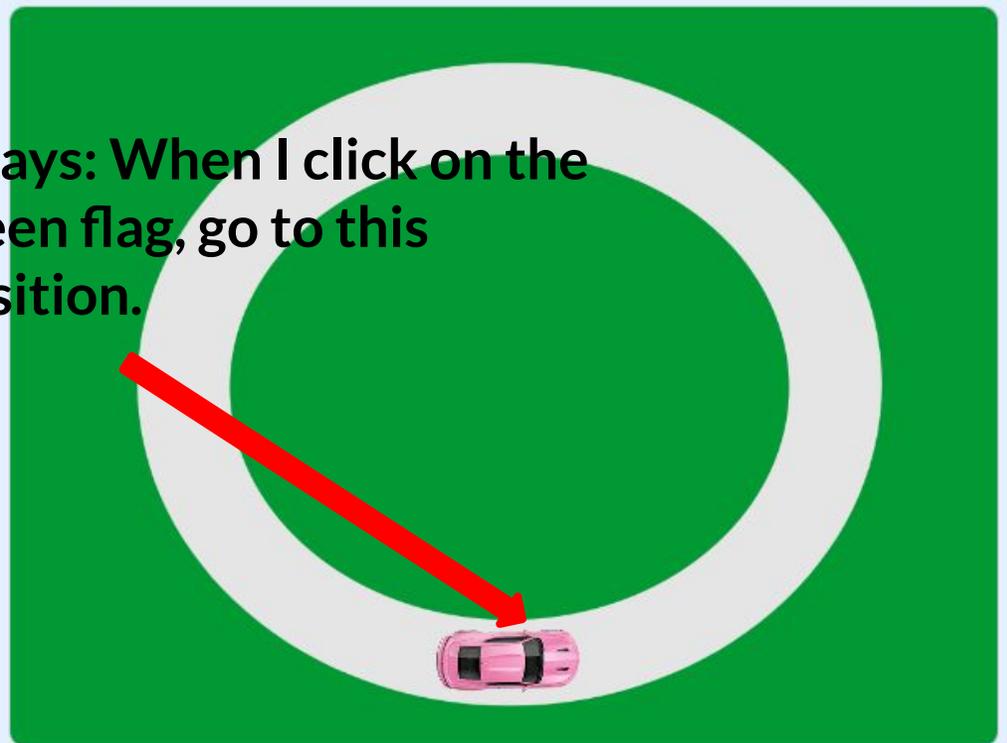
Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 1 y: -140
- glide 1 secs to random position
- glide 1 secs to x: 1 y: -140
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 1
- change y by 10

```
when green flag clicked  
go to x: 1 y: -140
```

```
forever  
move 1 steps
```

It says: When I click on the green flag, go to this position.



Sprite: car x: 1 y: -140

Show: Size: 40 Direction: 90

car

Stage

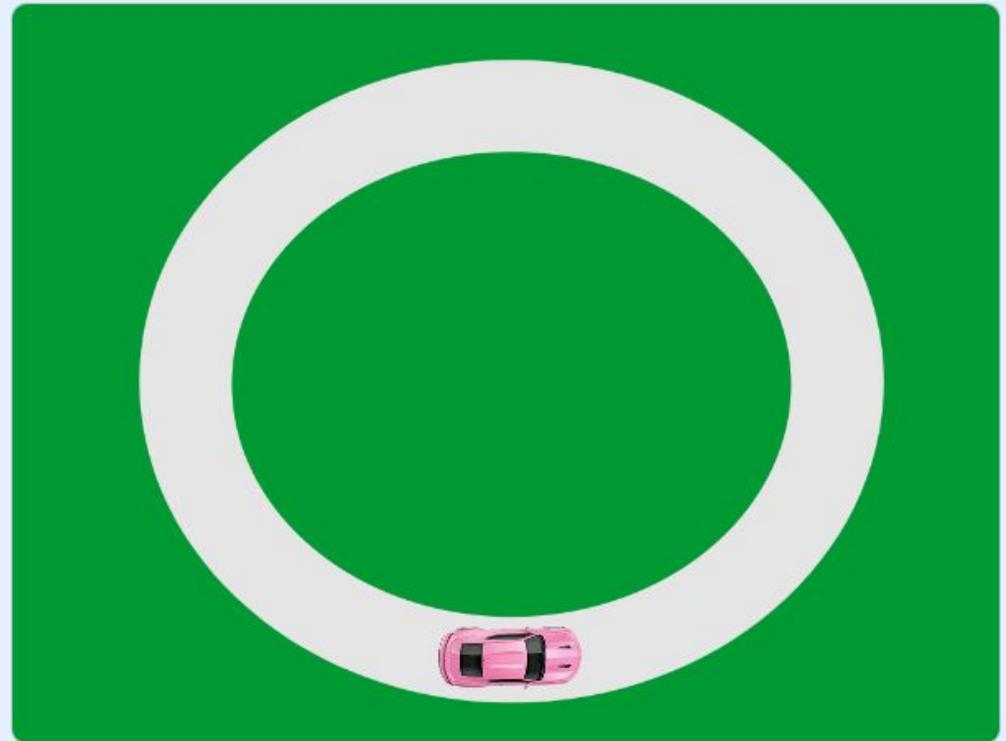
Backdrops: 3

Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 1 y: -140
- glide 1 secs to random position
- glide 1 secs to x: 1 y: -140
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 1
- change y by 10

```
when green flag clicked
  go to x: 1 y: -140
  forever loop
    move 1 steps
```

This part is making the car move forever. Do you see that the 'move' command is inside the forever command. That means that it will forever do the 'move' command.



Sprite: car

x: 1 y: -140

Show:

Size: 40 Direction: 90

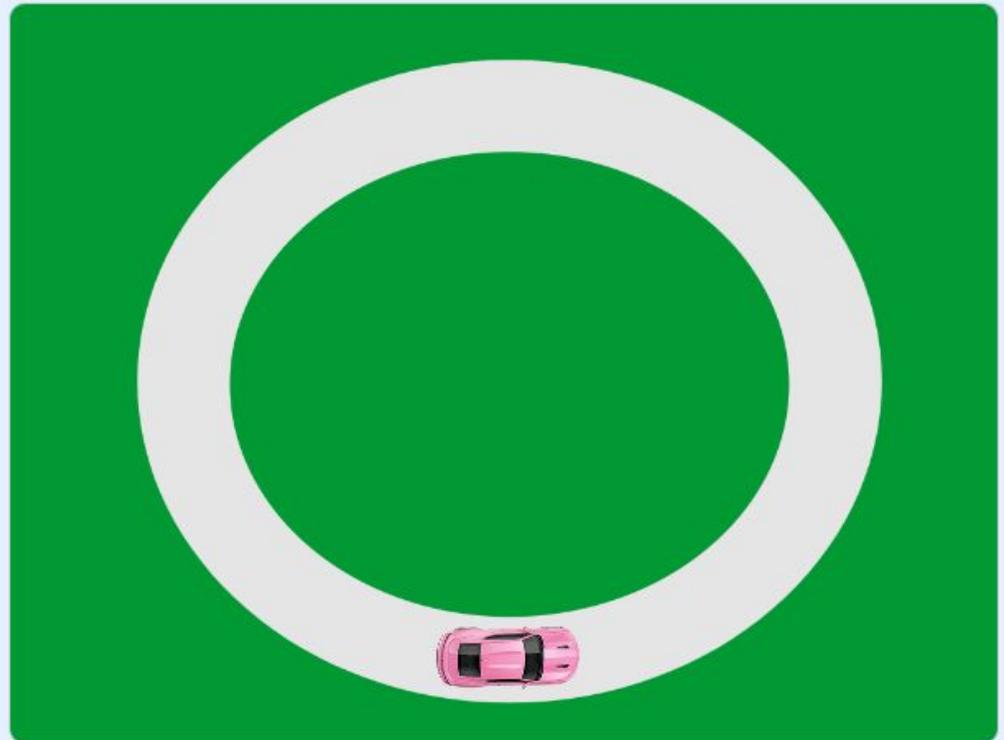
Backdrops: 3

Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 1 y: -140
- glide 1 secs to random position
- glide 1 secs to x: 1 y: -140
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 1
- change y by 10

```
when green flag clicked
  go to x: 1 y: -140
  forever loop
    move 1 steps
```

**This part of the command will make the car move. It says:
Move 1 step, forever.**



Sprite: car

x: 1 y: -140

Show:

Size: 40 Direction: 90

Backdrops: 3

car

Code Costumes Sounds

Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 1 y: -140
- glide 1 secs to random position
- glide 1 secs to x: 1 y: -140
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 1
- change y by 10

Looks

Sound

Events

Control

Sensing

Operators

Variables

My Blocks

```
when green flag clicked
  go to x: 1 y: -140
  forever loop
    move 1 steps
```

Try it out! Run your code, by clicking on the green flag.

Your car probably won't stop driving...

Stage

Sprite: car x: 1 y: -140

Show: Size: 40 Direction: 90

Backdrops: 3

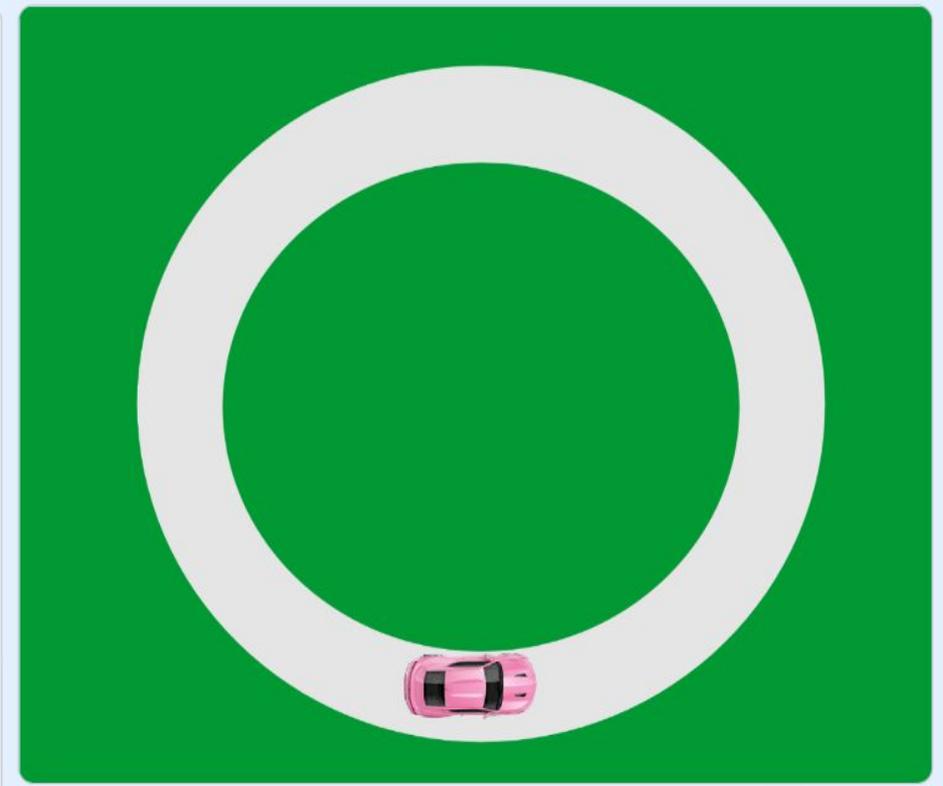
- Motion
- Looks
- Sound
- Events
- Control
- Sensing
- Operators
- Variables
- My Blocks

Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: -4 y: -136
- glide 1 secs to random position
- glide 1 secs to x: -4 y: -136
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to -4

To make it a self-driving car, that can follow the road. We need to add a sensor to it.

This sensor will tell the car where the road is.



Sprite: car

x: -4 y: -136

Show:

Size: 40 Direction: 90

Stage:

Backdrops: 3



1

costume1
0 x 0

Costume costume1



Group Ungroup

Forward Backward

Front Back

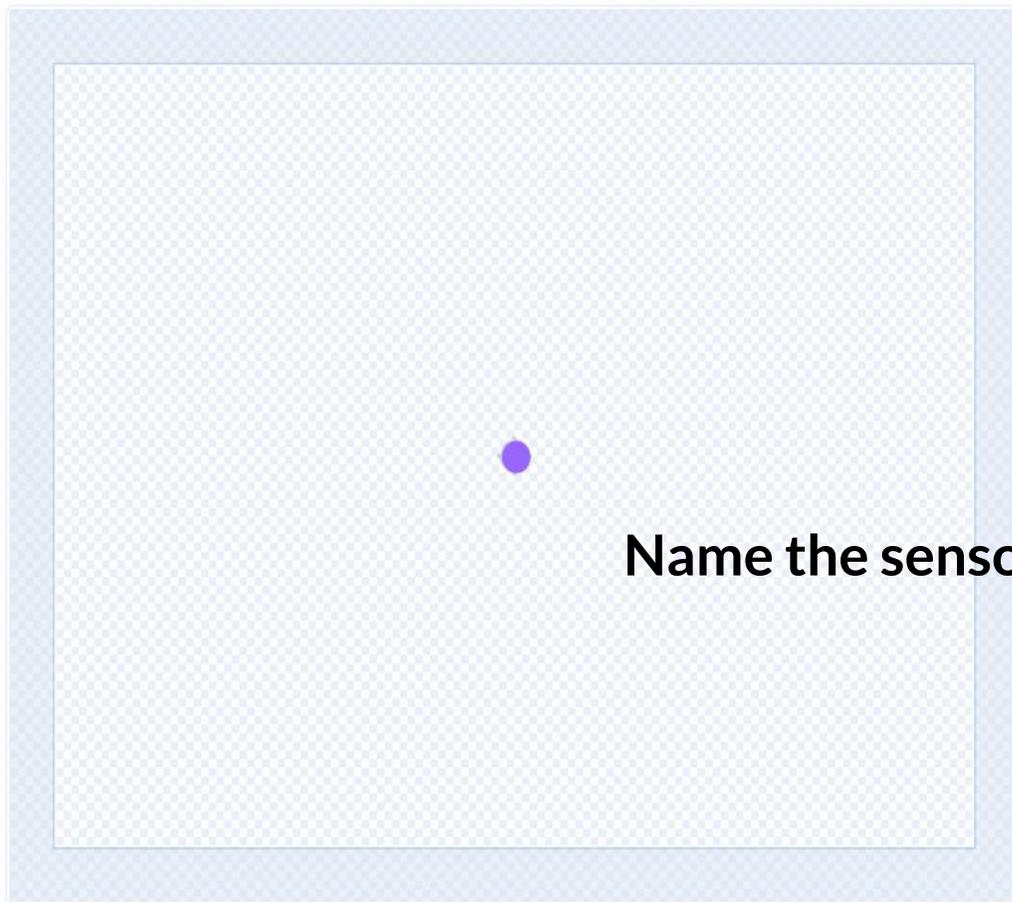
Fill

Outline

4

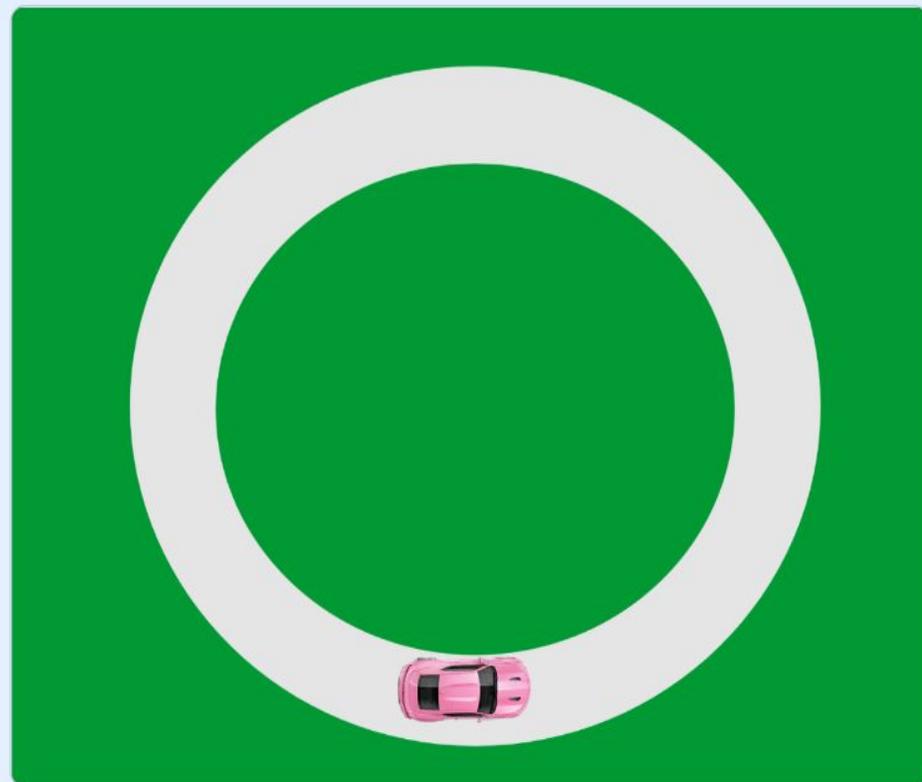


30



Name the sensor

Convert to Bitmap



Sprite Left sensor

x 36

y 28

Show



Size 100

Direction 90



car



Left sensor

Stage



Backdrops 3

3



Click here

1

costume1
0 x 0

Costume costume1



Group Ungroup

Forward Backward

Front Back

Fill 

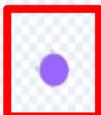
Outline 

4



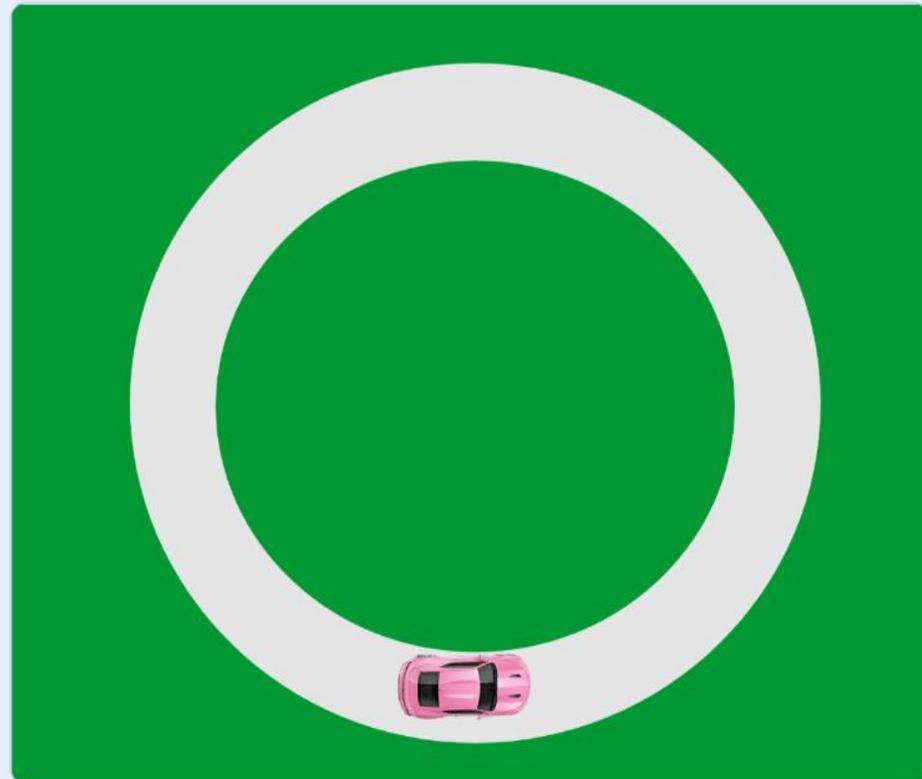
30

Select size 30



Make a dot in the center of the screen

Convert to Bitmap



Sprite Left sensor

x 36

y 28

Show  

Size 100

Direction 90



car

Left sensor

Stage



Backdrops 3

3



Click here

Code

Costumes

Sounds

Go back to Code tab

Events

when  clicked

when space key pressed

when this sprite clicked

when backdrop switches to backdrop1

when loudness > 10

when I receive message1

broadcast message1

broadcast message1 and wait

Control

wait 1 seconds

when  clicked

Drag the sensor on the top-left of the car

Sprite: Left sensor

x: 37 y: -118

Show:

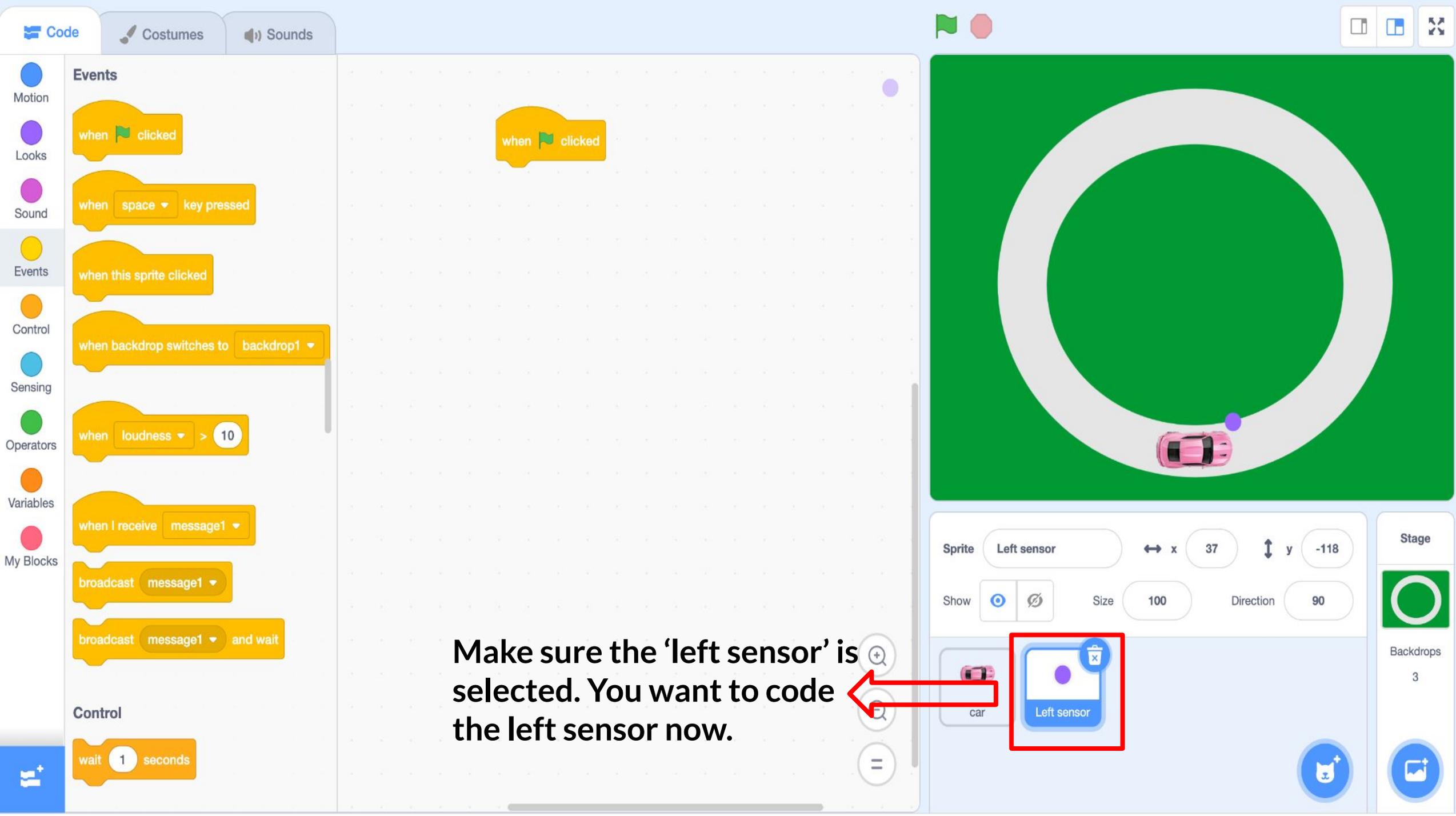
Size: 100 Direction: 90

car

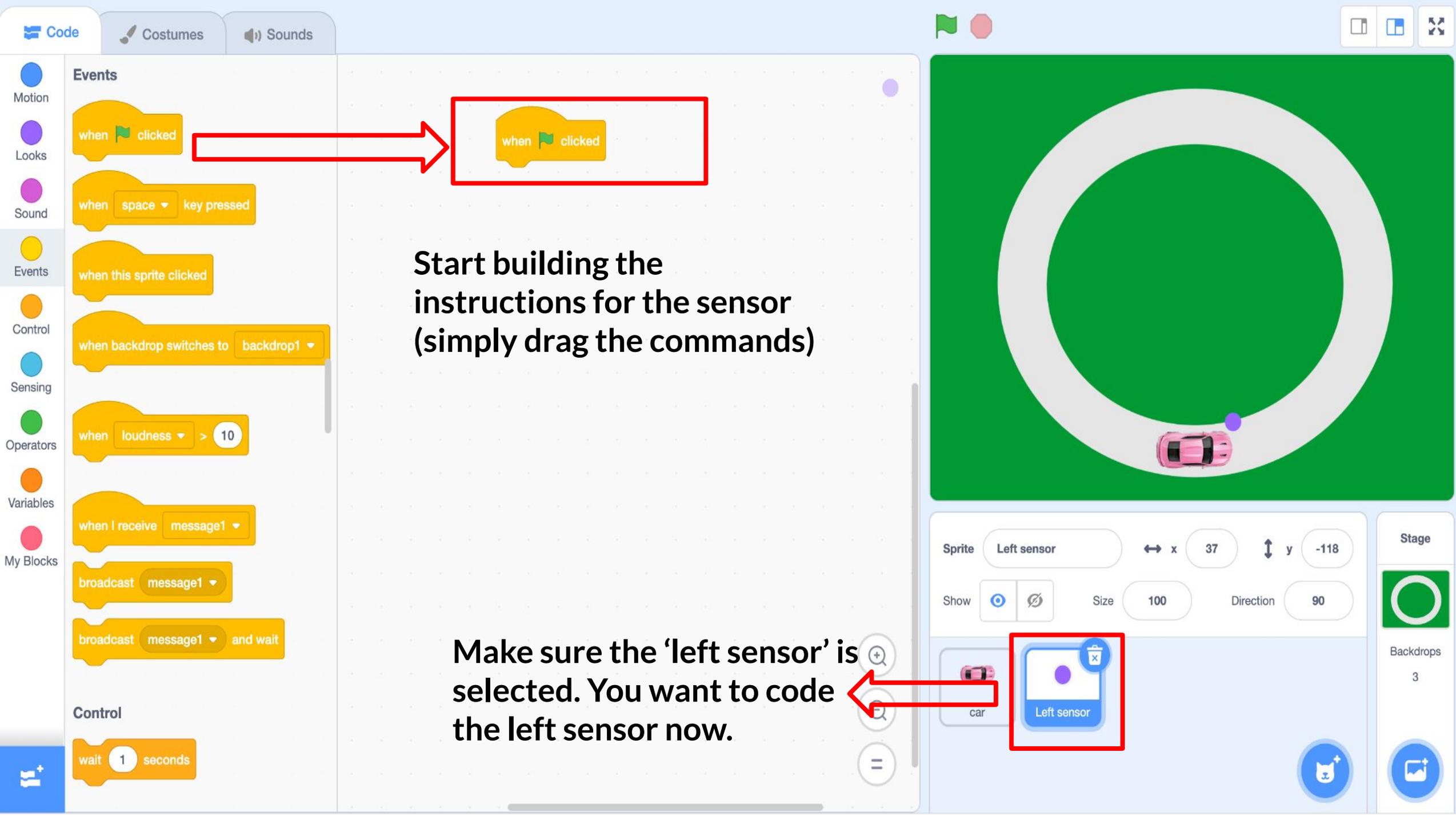
Left sensor

Stage

Backdrops: 3

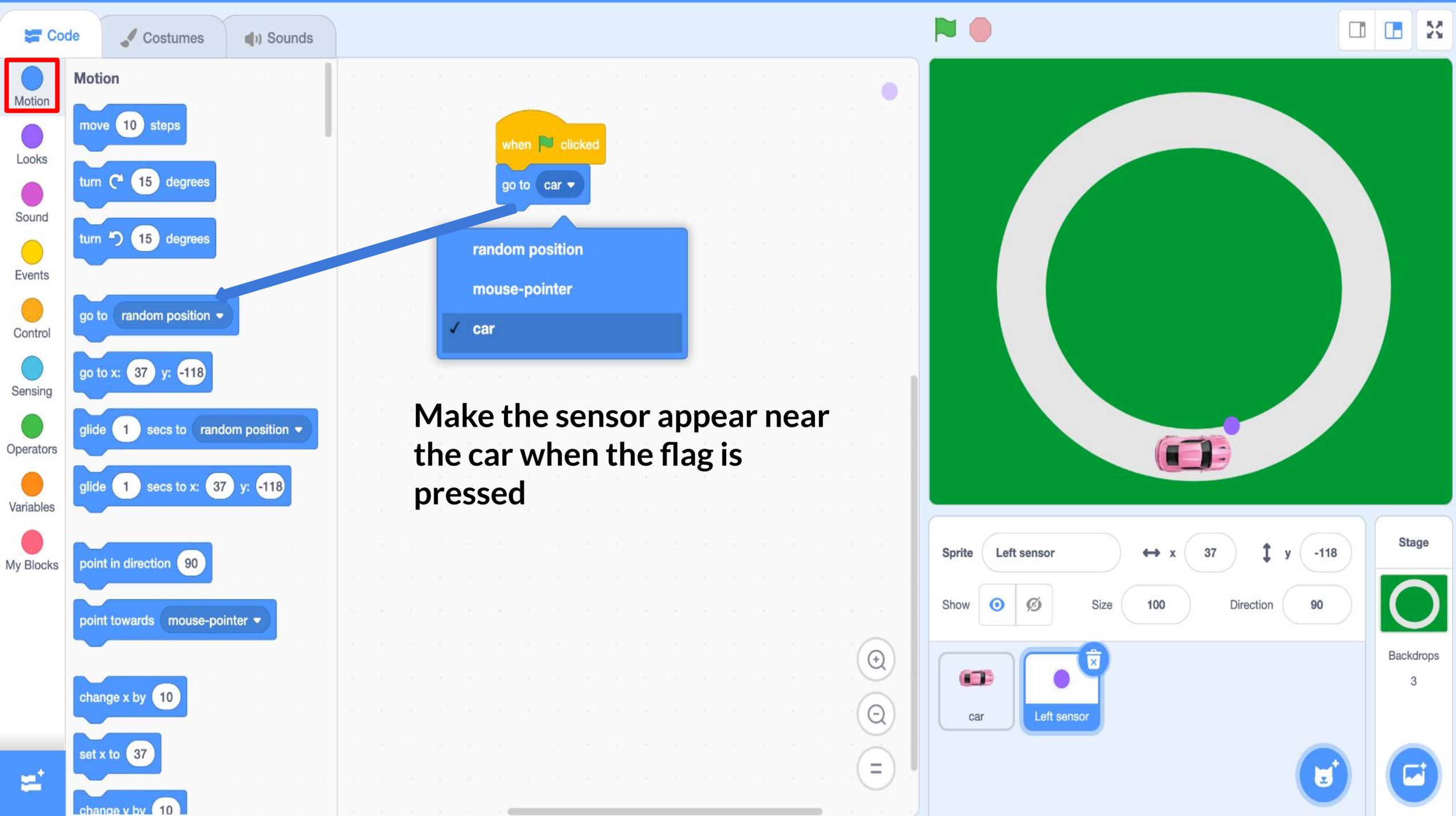


Make sure the 'left sensor' is selected. You want to code the left sensor now.



Start building the instructions for the sensor (simply drag the commands)

Make sure the 'left sensor' is selected. You want to code the left sensor now.



Motion

move 10 steps

turn 15 degrees

turn 15 degrees

go to random position

go to x: 37 y: -118

glide 1 secs to random position

glide 1 secs to x: 37 y: -118

point in direction 90

point towards mouse-pointer

change x by 10

set x to 37

change y by 10

when green flag clicked

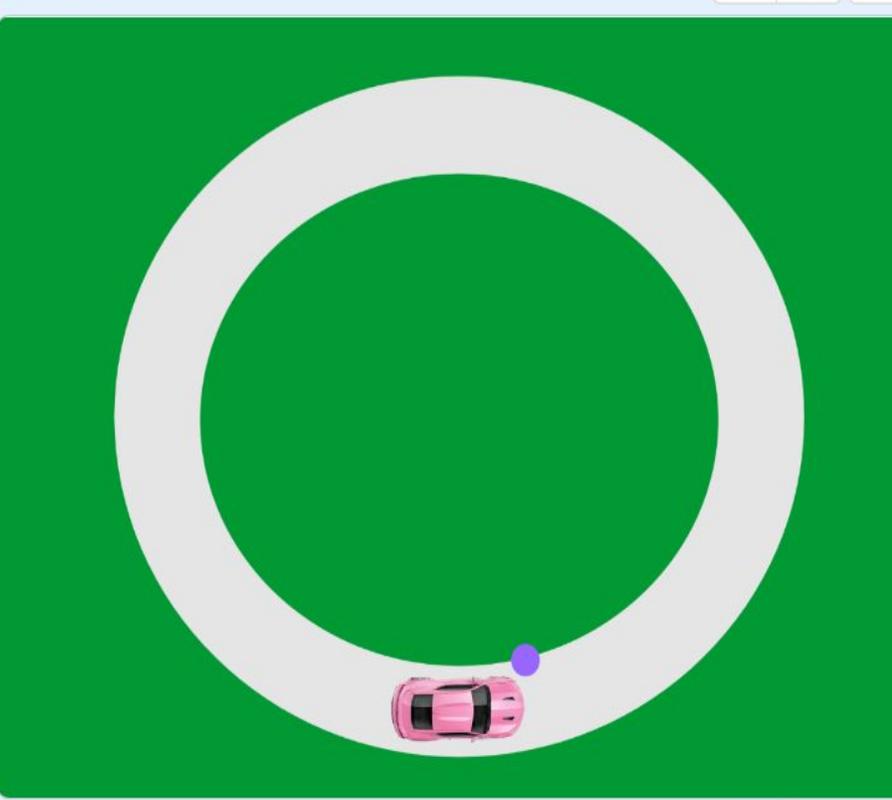
go to car

random position

mouse-pointer

car

Make the sensor appear near the car when the flag is pressed



Sprite: Left sensor

x: 37 y: -118

Show:

Size: 100 Direction: 90

car

Left sensor

Stage

Backdrops: 3



Motion

Looks

Sound

Events

Control

Sensing

Operators

Variables

My Blocks

Motion

move 10 steps

turn 15 degrees

turn 15 degrees

go to random position

go to x: 37 y: -118

glide 1 secs to random position

glide 1 secs to x: 37 y: -118

point in direction 90

point towards mouse-pointer

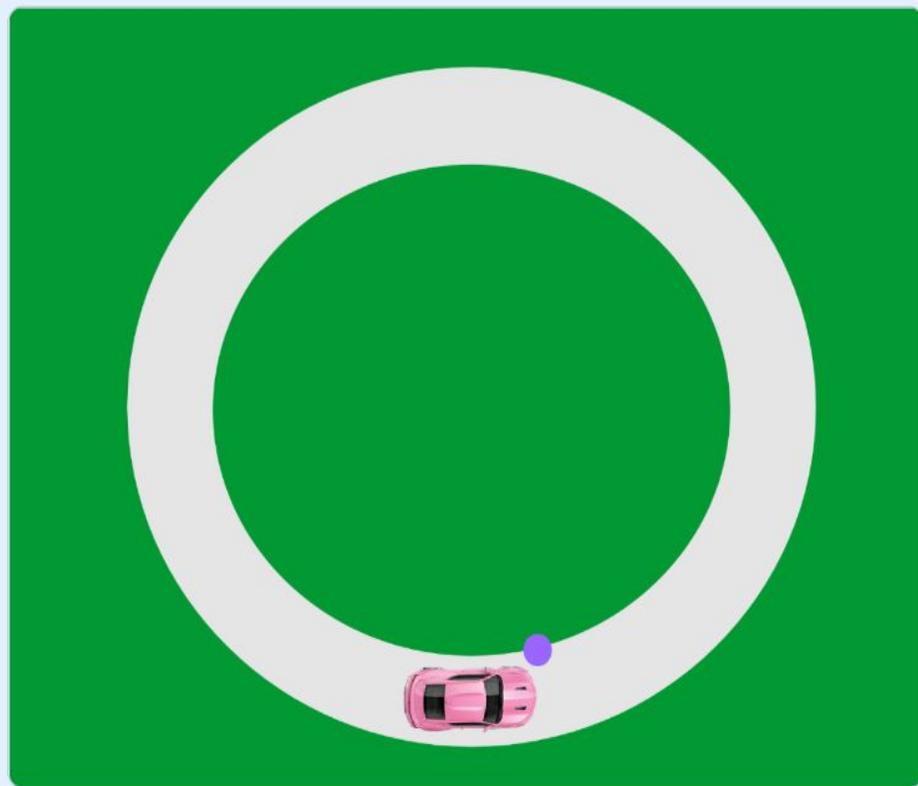
change x by 10

set x to 37

change y by 10

when green flag clicked
go to car
point in direction 90

Start making the sensor follow the car



Sprite: Left sensor

x: 37 y: -118

Show:

Size: 100 Direction: 90

Stage:

Backdrops: 3

Sprite list: car, Left sensor

Code Costumes Sounds

Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks

mouse down?
mouse x
mouse y
set drag mode: draggable
loudness
timer
reset timer
backdrop # of Stage
current year
days since 2000
username

when green flag clicked
go to: car
point in direction: backdrop # of Stage

backdrop # of Stage

Sprite: Left sensor
x: 37
y: -118
Show:
Size: 100
Direction: 90

Stage
Backdrops: 3

car
Left sensor

- Motion
- Looks
- Sound
- Events
- Control
- Sensing
- Operators
- Variables
- My Blocks

Code blocks palette:

- Motion:** mouse down?, mouse x, mouse y
- Events:** set drag mode (draggable)
- Control:** loudness, timer, reset timer
- Sensing:** backdrop # of Stage, current year, days since 2000, username
- Operators:** +

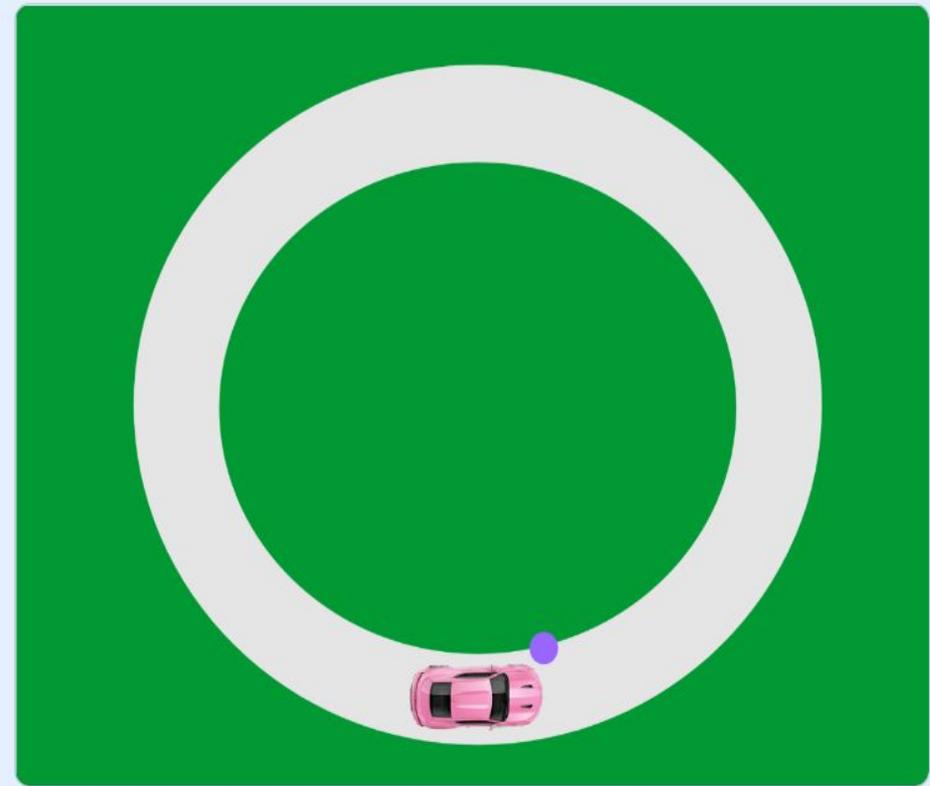
Scratch script area:

```

when green flag clicked
  go to car
  point in direction x position of car
  
```

Stage dropdown menu:

- car (checked)



Properties and Sprites panels:

- Sprite:** Left sensor, x: 37, y: -118, Show: [on], Size: 100, Direction: 90
- Sprites list:** car, Left sensor
- Stage:** [Stage icon]
- Backdrops:** 3

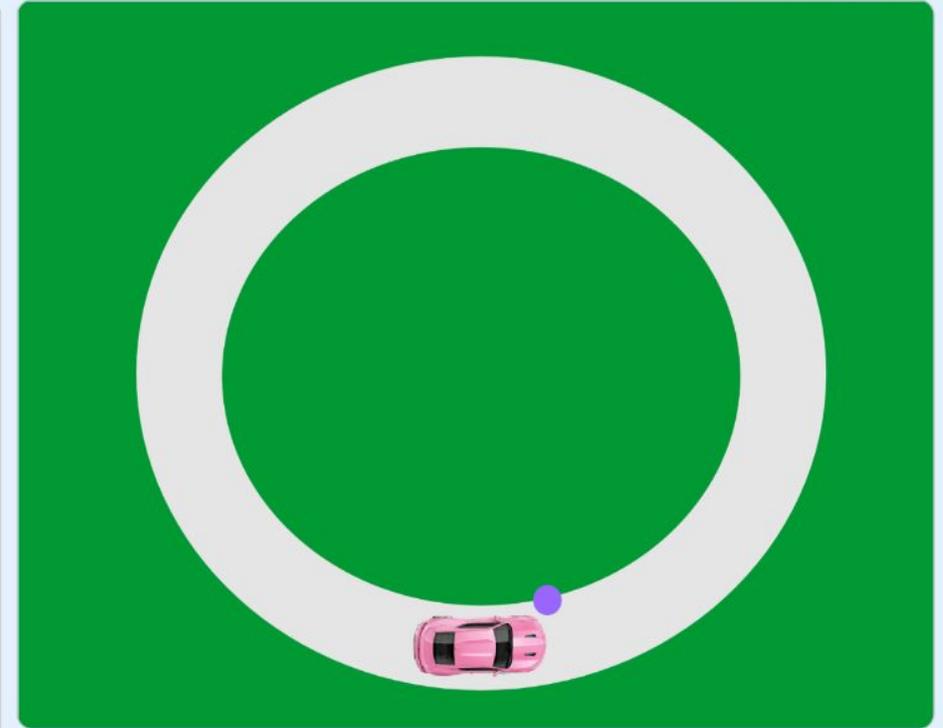
- Motion
- Looks
- Sound
- Events
- Control
- Sensing
- Operators
- Variables
- My Blocks

```
when green flag clicked  
go to car  
point in direction direction of car  
set drag mode draggable  
loudness  
timer  
reset timer  
backdrop # of Stage  
current year  
days since 2000  
username
```

```
when green flag clicked  
go to car  
point in direction direction of car
```

- x position
- y position
- direction**
- costume #
- costume name
- size
- volume

The sensor will now follow the DIRECTION of the CAR



Sprite: Left sensor | x: 37 | y: -118 | Size: 100 | Direction: 90

car | Left sensor

- Motion
- Looks
- Sound
- Events
- Control
- Sensing
- Operators
- Variables
- My Blocks

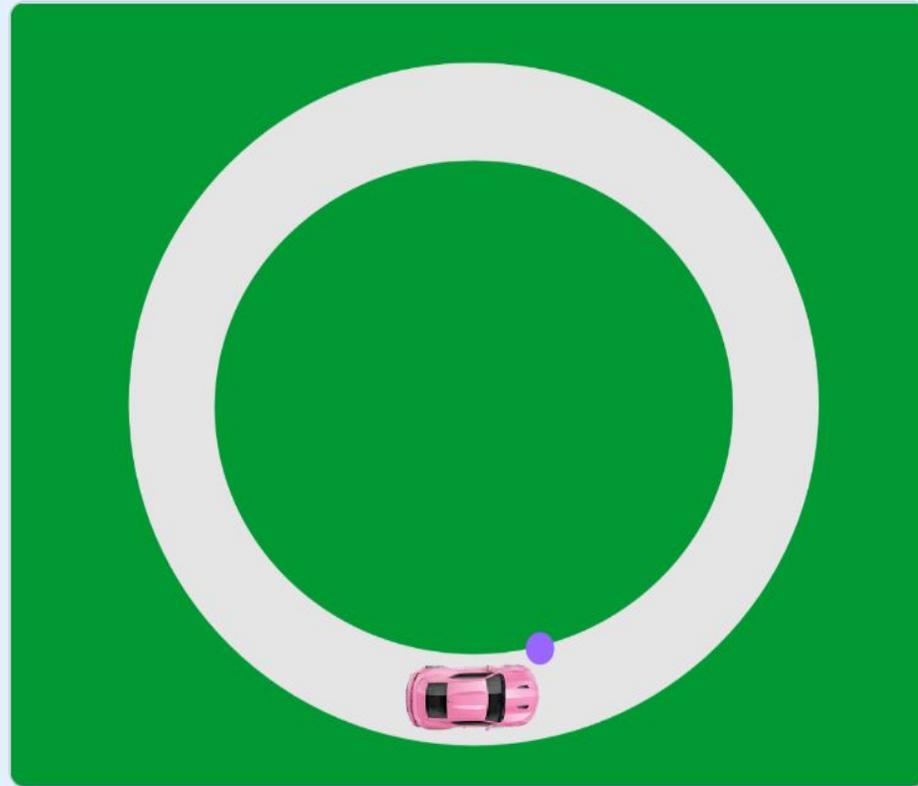
Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: 37 y: -118
- glide 1 secs to random position
- glide 1 secs to x: 37 y: -118
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to 37
- change y by 10

```
when green flag clicked  
go to car  
point in direction direction of car  
turn 25 degrees
```



Place the sensor of the left side of the car



Sprite: Left sensor

x: 37 y: -118

Show:

Size: 100 Direction: 90

Stage

Backdrops: 3

car Left sensor

Code Costumes Sounds

Motion

move 10 steps

turn 15 degrees

turn 15 degrees

go to random position

go to x: 37 y: -118

glide 1 secs to random position

glide 1 secs to x: 37 y: -118

point in direction 90

point towards mouse-pointer

change x by 10

set x to 37

change y by 10

when green flag clicked

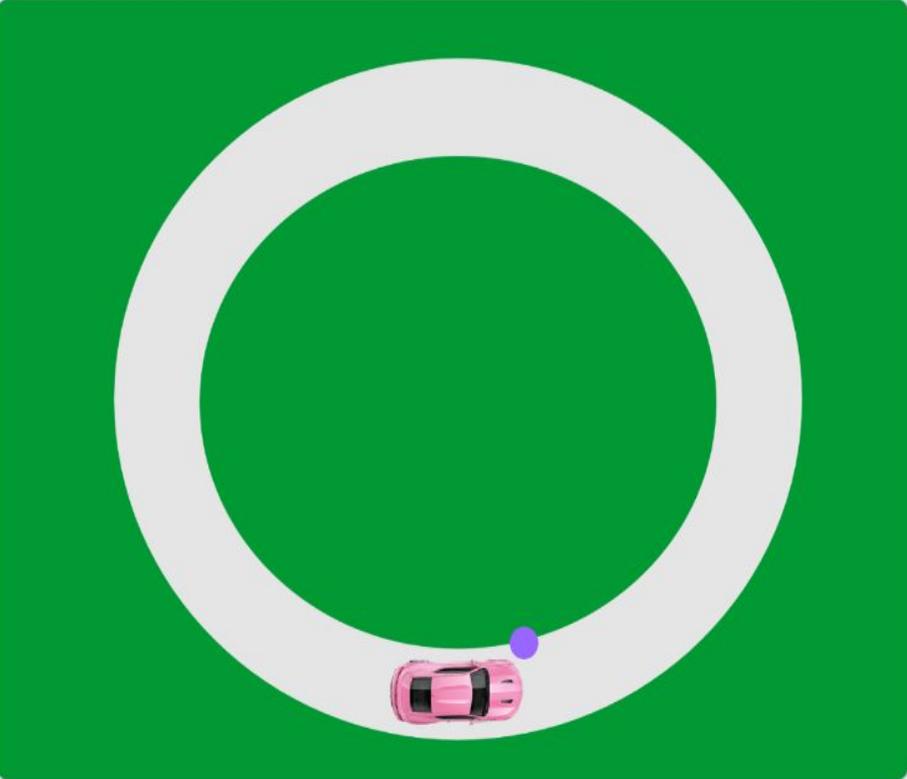
go to car

point in direction direction of car

turn 25 degrees

move 40 steps

Place the sensor in the front of the car



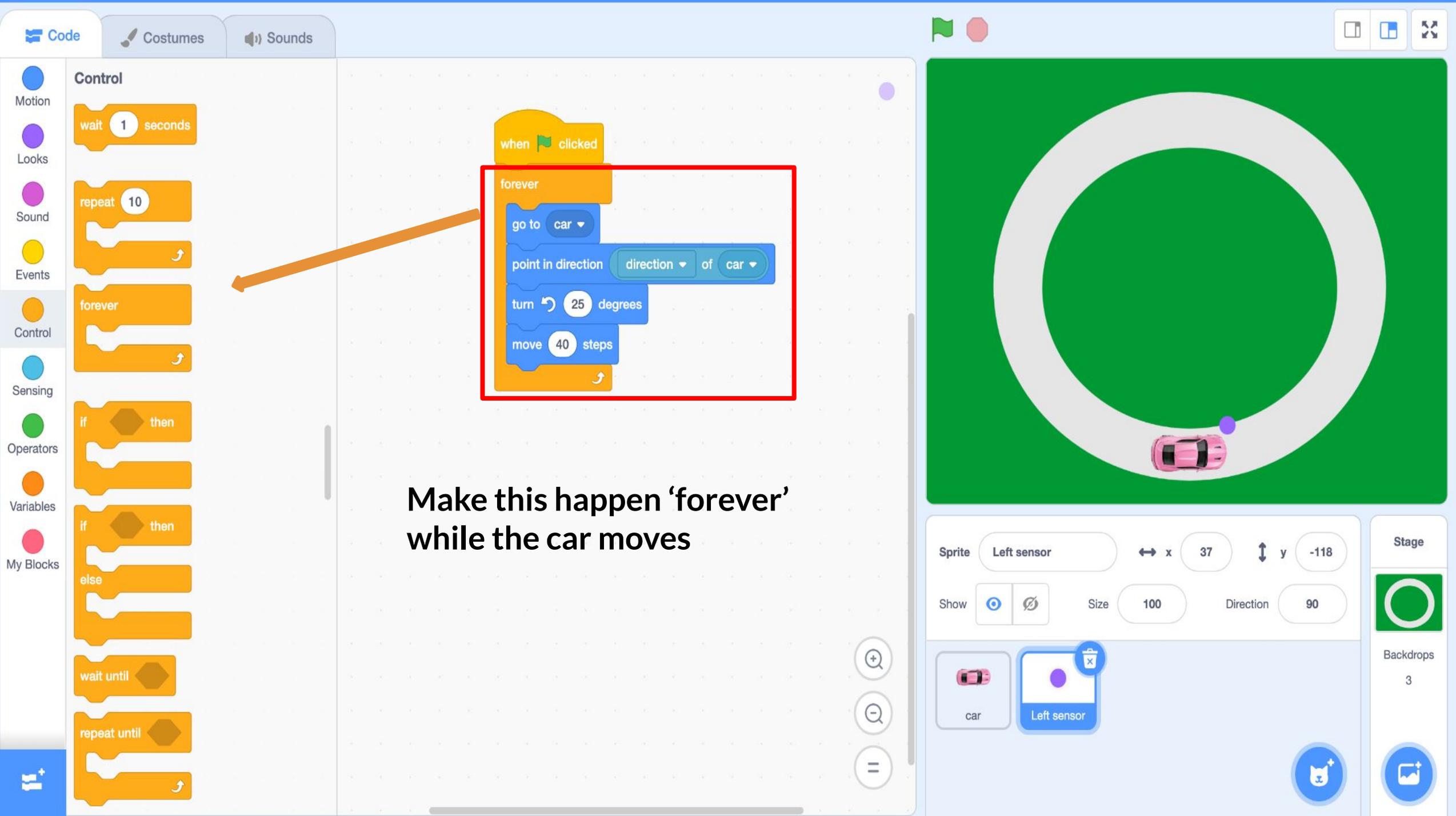
Sprite Left sensor x 37 y -118

Show Size 100 Direction 90

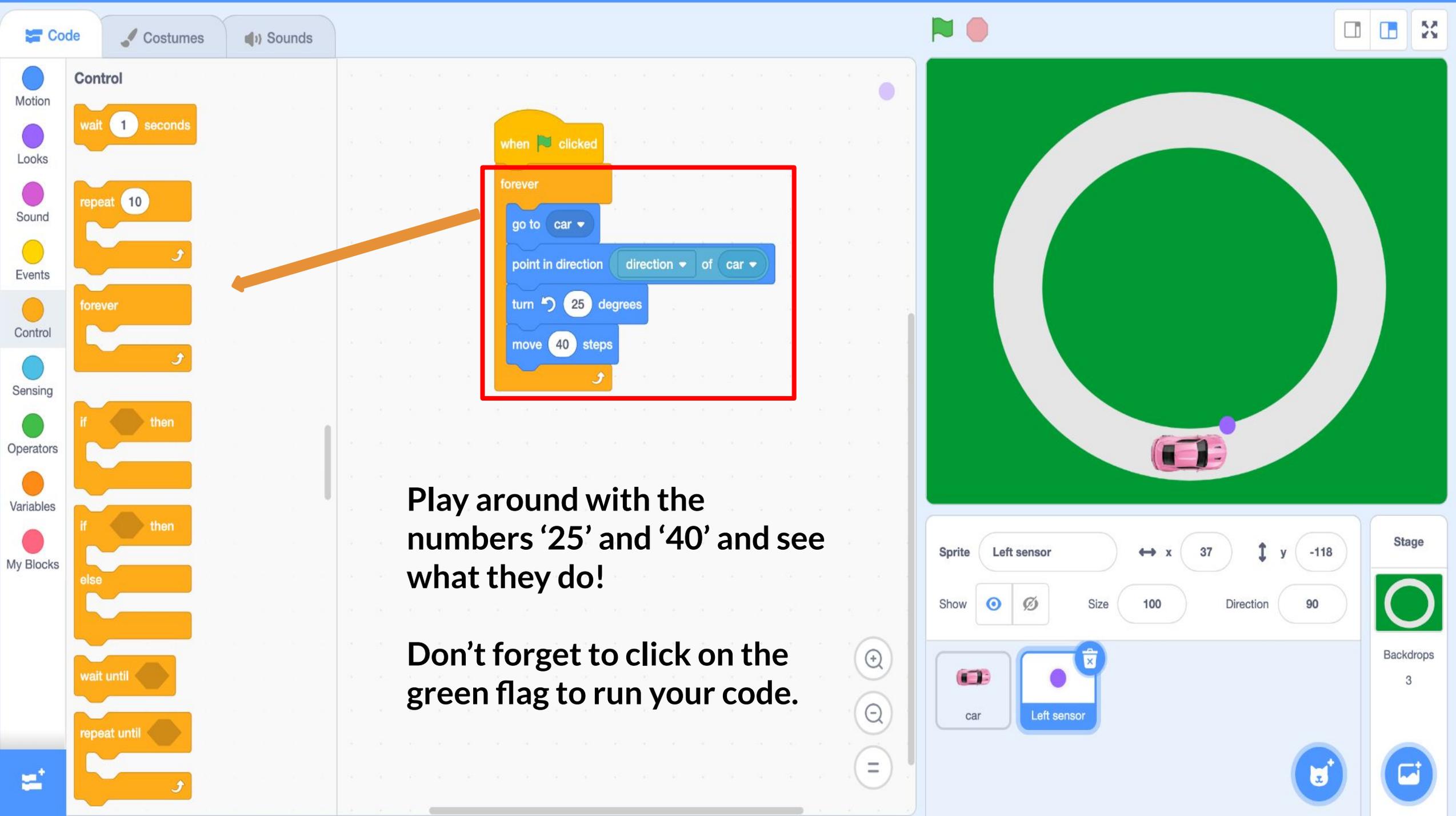
car Left sensor

Stage Backdrops 3

Detailed description: This image shows the Scratch development environment. On the left is the 'Code' area with a 'Motion' category selected. A script is being built on a grid background. The script starts with a 'when green flag clicked' event block, followed by 'go to car', 'point in direction' (set to 'direction' of 'car'), 'turn 25 degrees', and 'move 40 steps'. A red box highlights the 'move 40 steps' block, and a blue arrow points from it to the 'move 10 steps' block in the 'Motion' category. Below the script, a text instruction reads 'Place the sensor in the front of the car'. On the right is the 'Stage' area, which shows a green circular track with a pink car at the bottom. A purple dot representing a 'Left sensor' is placed on the track directly in front of the car. Below the stage is the 'Sprite' area, showing the 'Left sensor' sprite selected with its coordinates (x: 37, y: -118) and a size of 100. The 'car' sprite is also visible in the sprite area. The 'Stage' area shows a backdrop of a green circle on a green background.



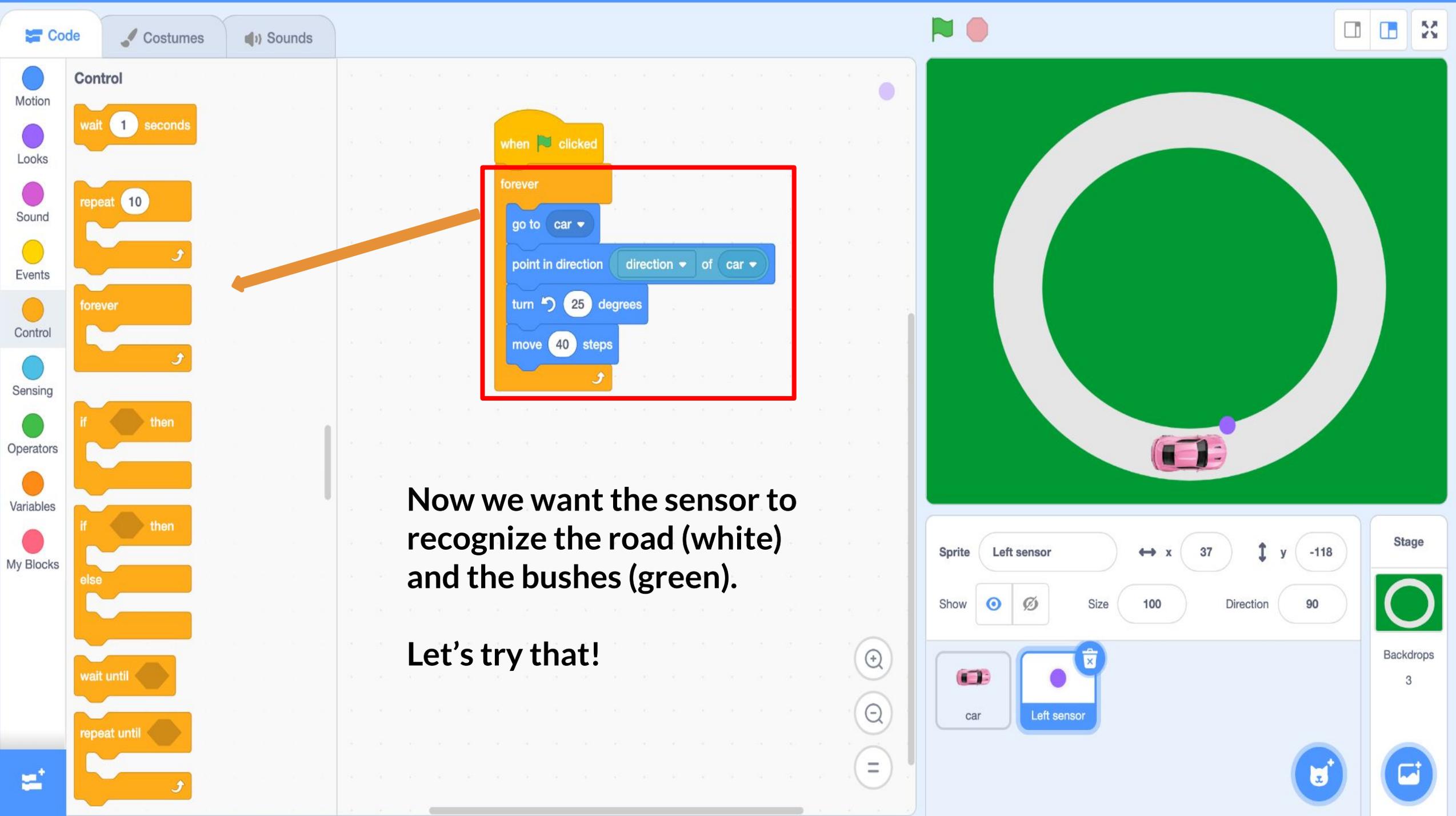
Make this happen 'forever' while the car moves



```
when green flag clicked
  forever loop
    go to car
    point in direction direction of car
    turn 25 degrees
    move 40 steps
```

Play around with the numbers '25' and '40' and see what they do!

Don't forget to click on the green flag to run your code.



Control

wait 1 seconds

repeat 10

forever

if then

if then

else

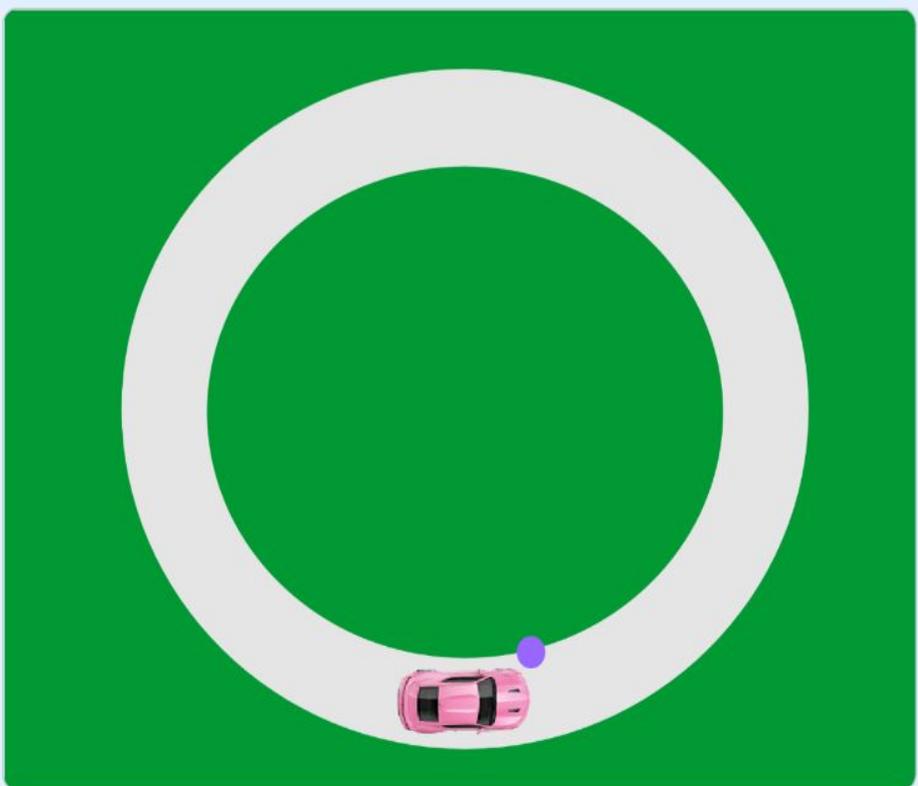
wait until

repeat until

```
when green flag clicked
forever
  go to car
  point in direction direction of car
  turn 25 degrees
  move 40 steps
```

Now we want the sensor to recognize the road (white) and the bushes (green).

Let's try that!



Sprite: Left sensor

x: 37 y: -118

Show:

Size: 100 Direction: 90

Stage:

Backdrops: 3

car Left sensor

Sensing

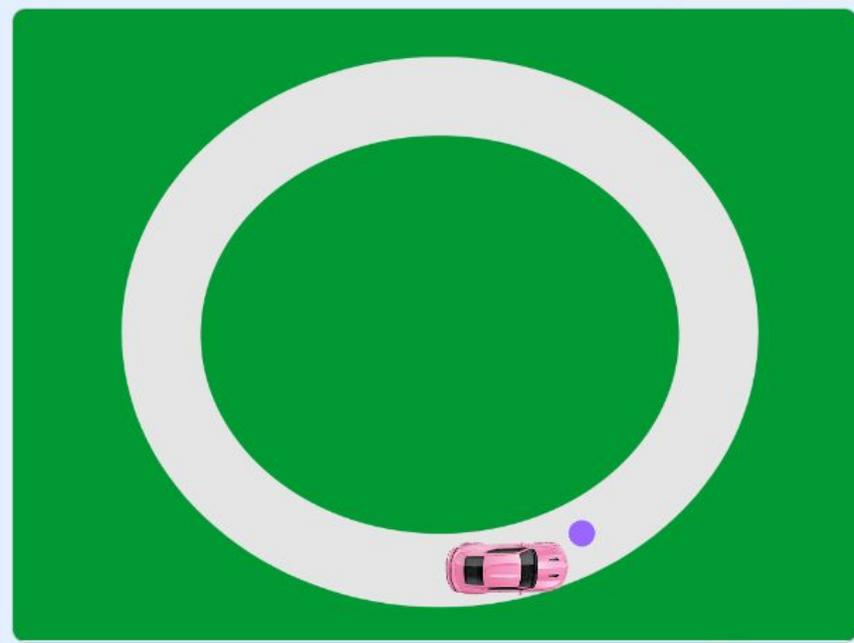
- touching mouse-pointer ?
- touching color ?
- color is touching ?
- distance to mouse-pointer
- asks your name? and wait
- answer
- key space pressed?
- mouse down?
- mouse x
- mouse y
- set drag mode draggable
- loudness
- timer
- reset timer

```
when clicked
  forever
    go to car
    set direction direction of car
    turn 25 degrees
    move 50 steps
```

Build another block of commands that runs 'forever'

```
when clicked
  forever
    if then
```

Add an 'if else' for when the sensor is touching green (goes outside the road)



Sprite Left Sensor x: 84 y: -119

Show Size: 100 Direction: 65

car **Left Sensor**

Stage

Backdrops 3

Sensing

- touching mouse-pointer ?
- touching color ?**
- color is touching ?
- distance to mouse-pointer

Control

- ask "What's your name?" and wait
- answer

Operators

- key space pressed?
- mouse down?
- mouse x
- mouse y

My Blocks

- set drag mode: draggable
- loudness
- timer
- reset timer

```
when green flag clicked
  forever loop
    go to car
    point in direction direction of car
    turn 25 degrees
    move 50 steps

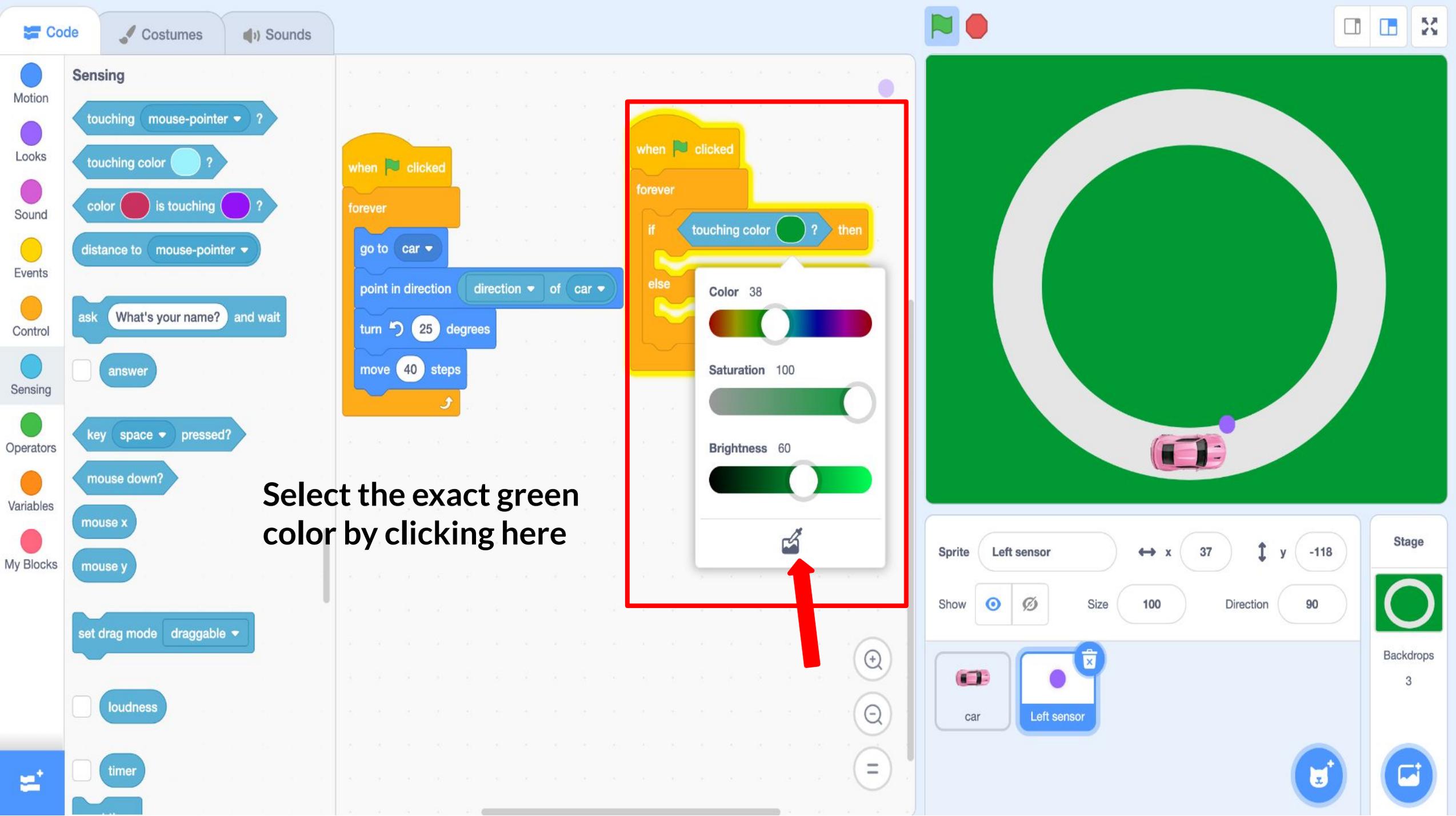
when green flag clicked
  forever loop
    touching color ? then
```

Here we are making the sensor sense a color.

Stage properties: Sprite: Left Sensor, x: 84, y: -119, Size: 100, Direction: 65.

Stage Backdrops: 3

Stage Sprites: car, Left Sensor



Select the exact green color by clicking here

when green flag clicked

forever

if touching color green? then

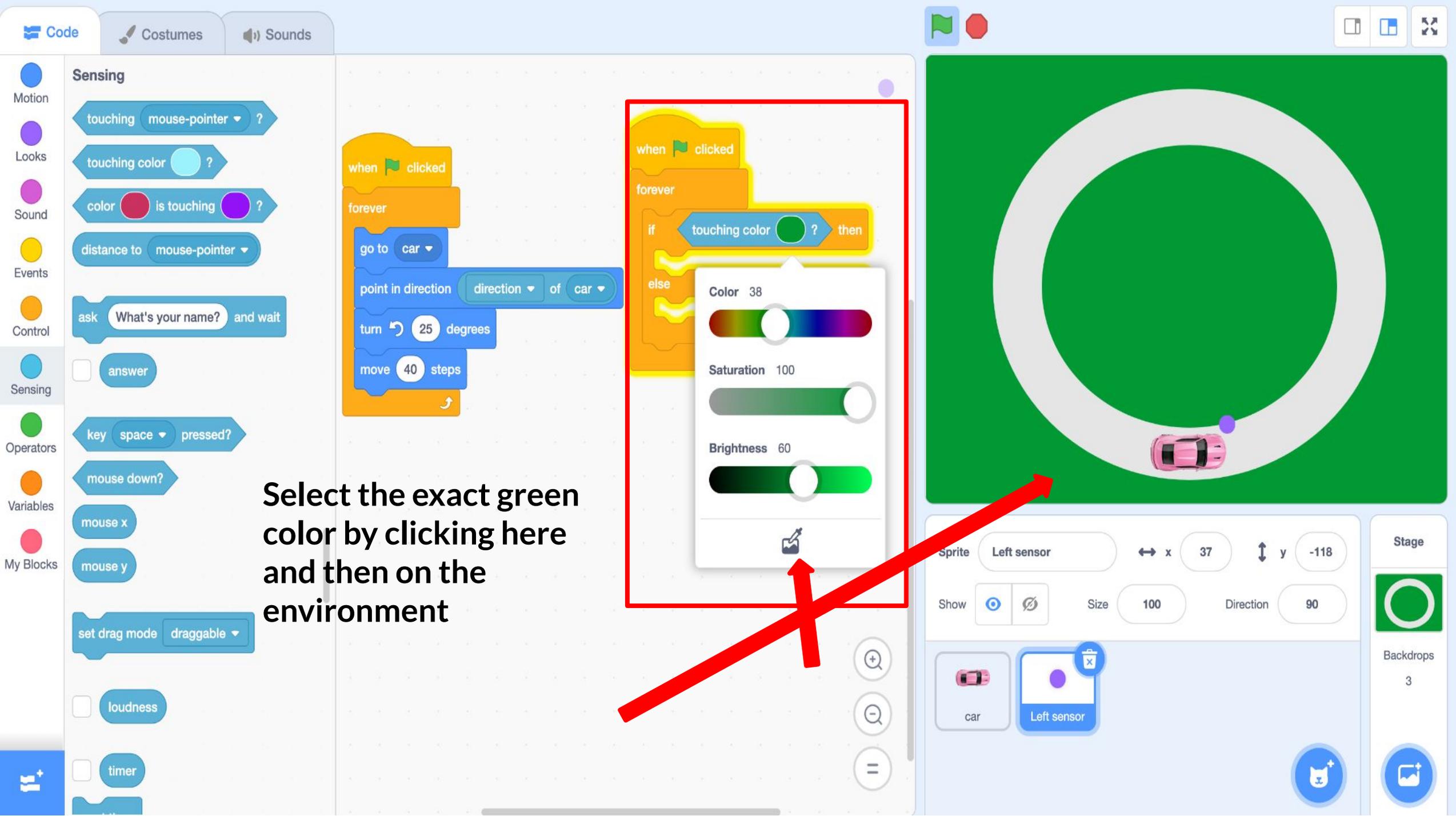
else

Color 38

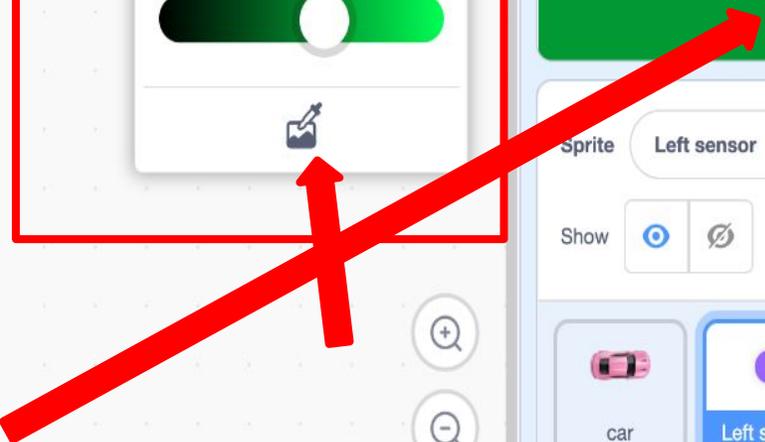
Saturation 100

Brightness 60





Select the exact green color by clicking here and then on the environment



- Motion
- Looks
- Sound
- Events
- Control
- Sensing
- Operators
- Variables**
- My Blocks

Variables

Make a Variable

Make a List

My Blocks

Make a Block

Make 2 new variables ('turn right' and 'turn left')

```

when green flag clicked
  forever loop
    go to car
    point in direction direction
    turn 25 degrees
    move 40 steps
  
```

```

when green flag clicked
  forever loop
    if touching color green? then
  
```

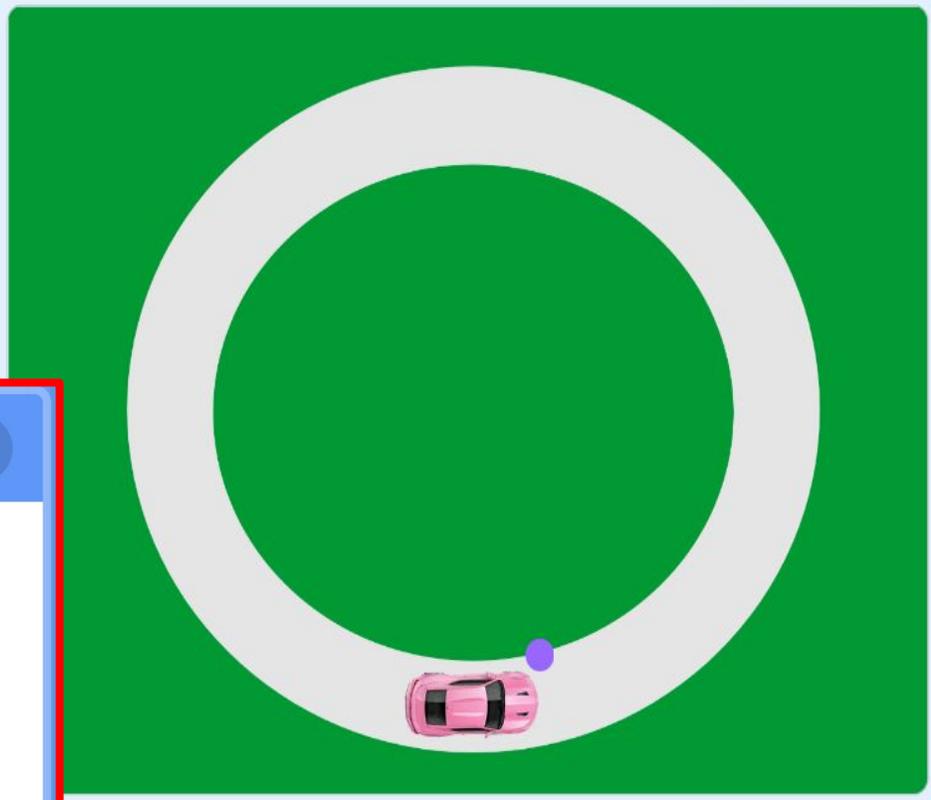
New Variable

New variable name:

turn right

For all sprites For this sprite only

Cancel OK



Sprite: Left sensor

x: 37 y: -118

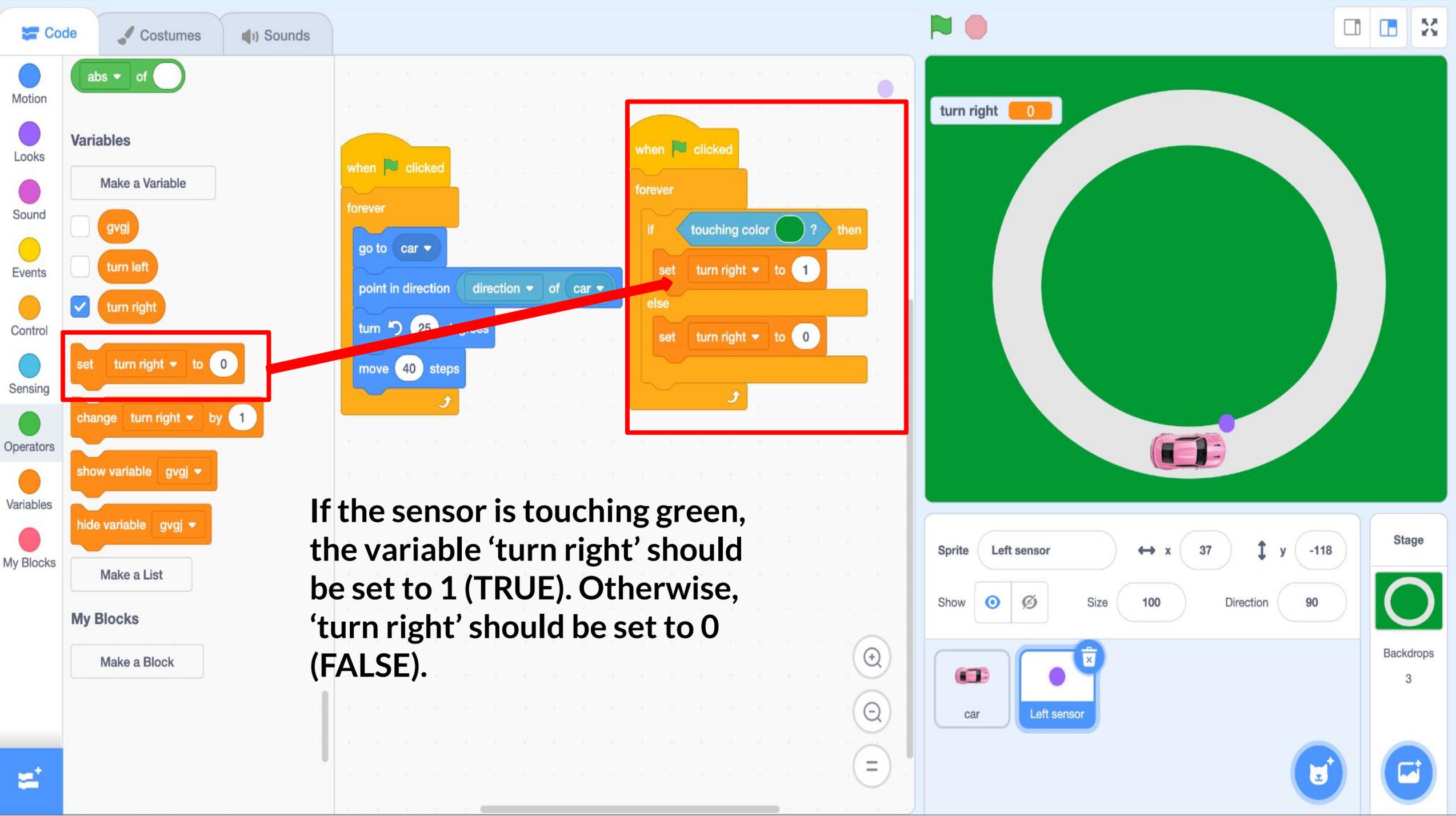
Size: 100 Direction: 90

car

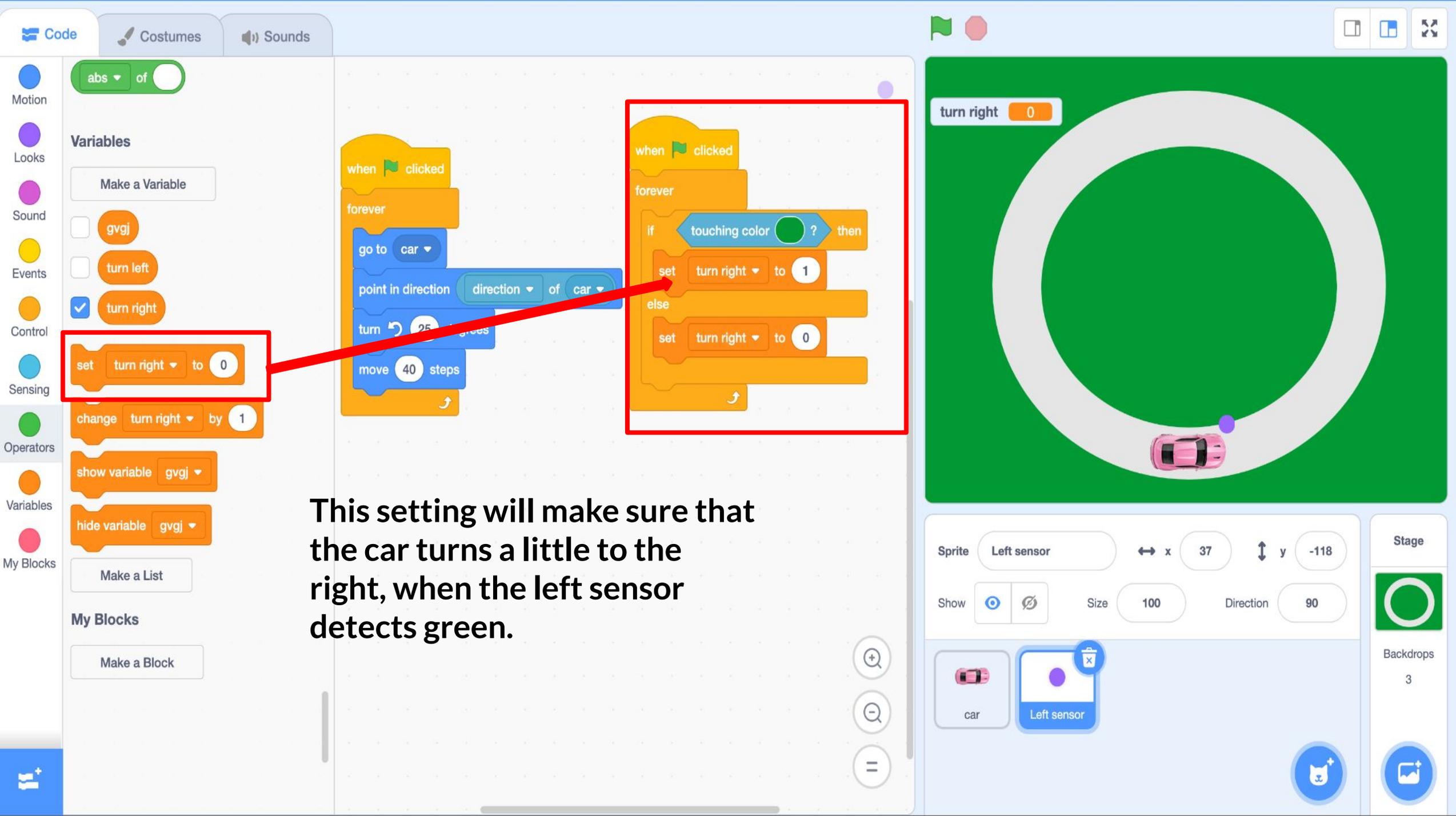
Left sensor

Stage

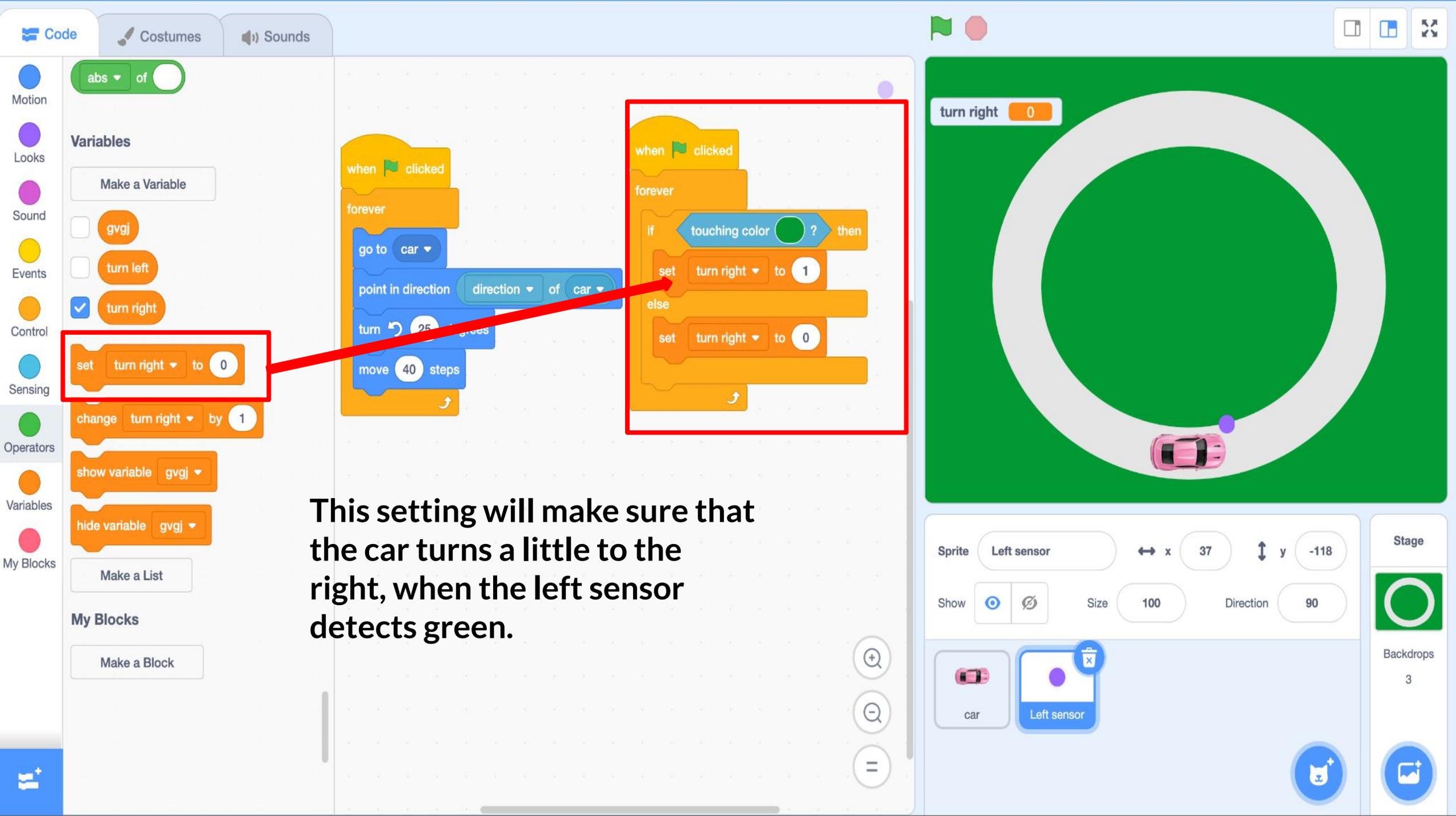
Backdrops: 3



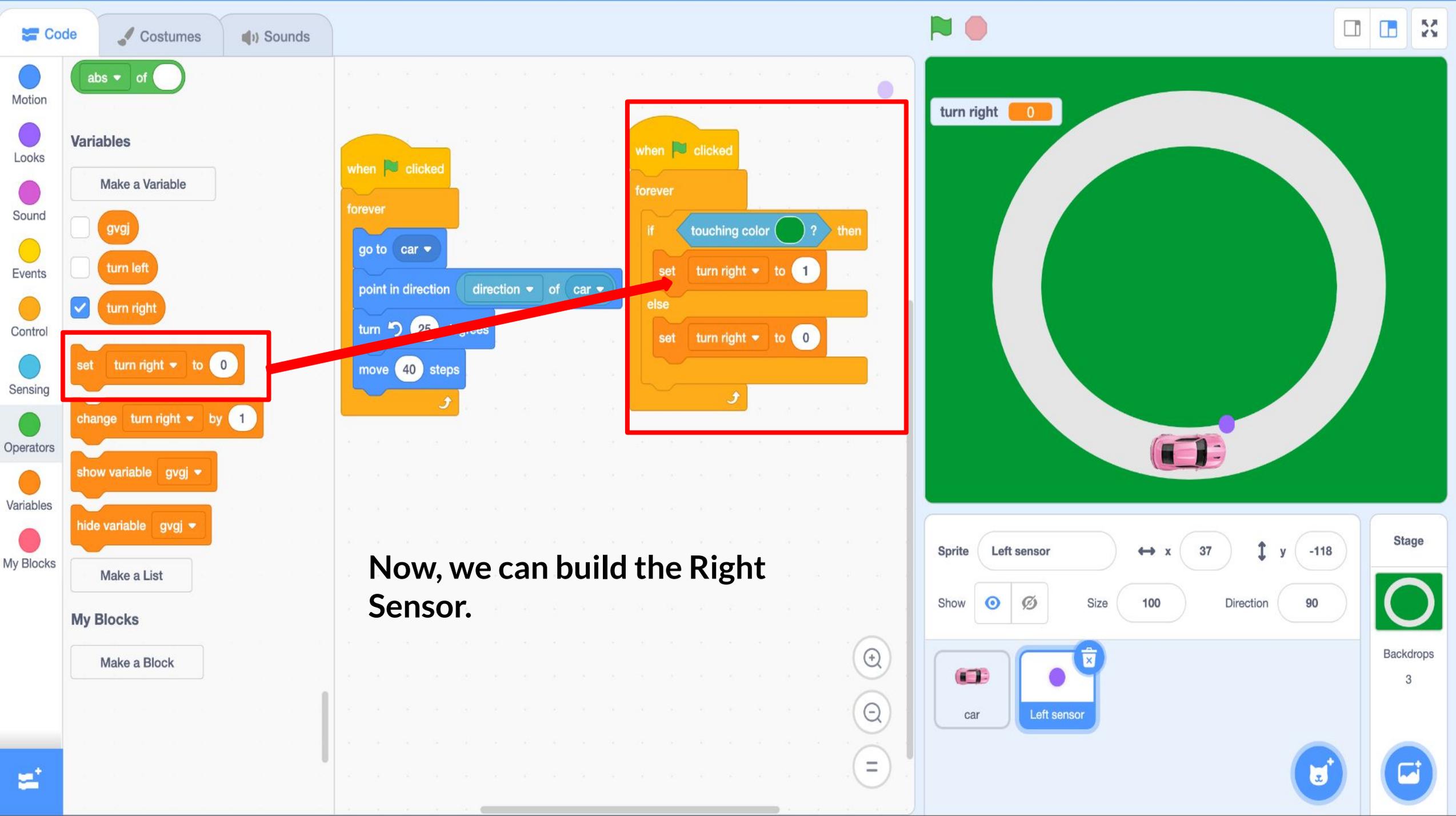
If the sensor is touching green, the variable 'turn right' should be set to 1 (TRUE). Otherwise, 'turn right' should be set to 0 (FALSE).



This setting will make sure that the car turns a little to the right, when the left sensor detects green.



This setting will make sure that the car turns a little to the right, when the left sensor detects green.



Now, we can build the Right Sensor.

Code Costumes Sounds

Motion

Looks

Sound

Events

Control

Sensing

Operators

Variables

My Blocks

abs of

Variables

Make a Variable

gvgj

turn left

turn right

set gvgj to 0

change gvgj by 1

show variable gvgj

hide variable gvgj

Make a List

Make a Block

First, duplicate the left sensor.
Then, make all the necessary changes (See red squares in this picture)

when clicked

forever

go to car

point in direction direction of car

turn -25 degrees

move 40 steps

when clicked

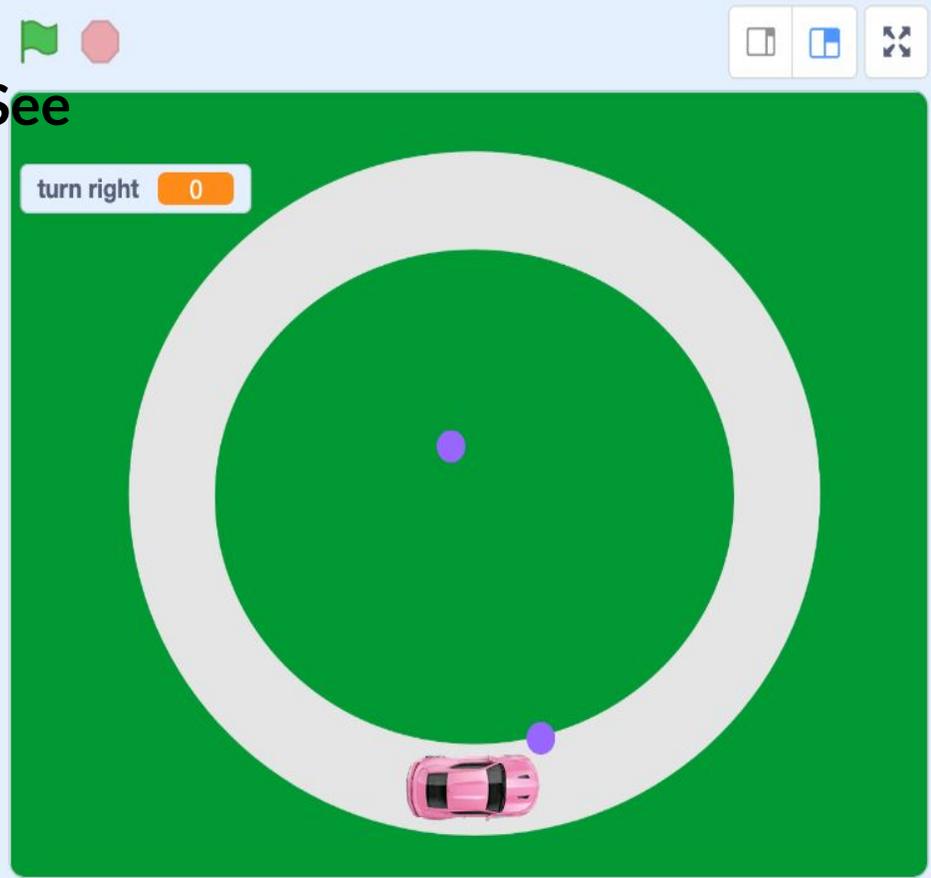
forever

if touching color ? then

set turn left to 1

else

set turn left to 0



duplicate

export

delete

Left sensor

Sprite Right senso

x -10 y 16

Size 100 Direction 90

car Left sensor Right senso

Stage

Backdrops 3

Right click and choose duplicate

Code

Costumes

Sounds

Motion

move 10 steps

turn 15 degrees

turn 15 degrees

go to random position

go to x: -120 y: 76

glide 1 secs to random position

glide 1 secs to x: -120 y: 76

point in direction 90

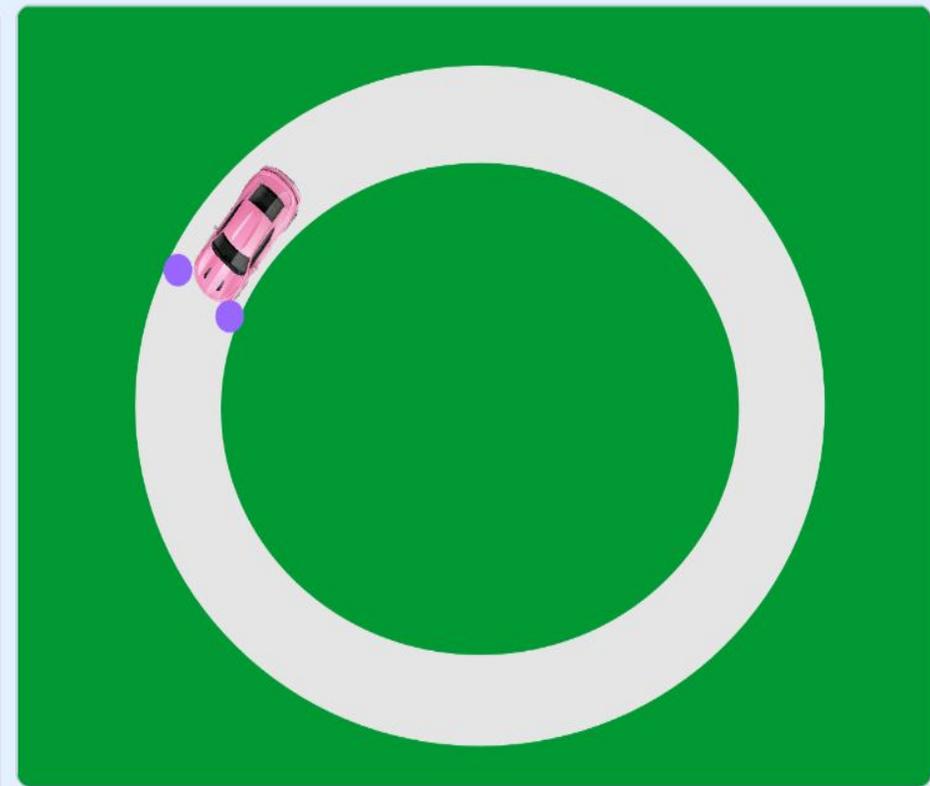
point towards mouse-pointer

change x by 10

set x to -120

change y by 10

```
when green flag clicked
  go to x: 1 y: -140
  forever loop
    move 1 steps
    if turn left = 1 then
      turn 5 degrees
    if turn right = 1 then
      turn 5 degrees
```



Sprite: car

x: -120 y: 76

Show:

Size: 40

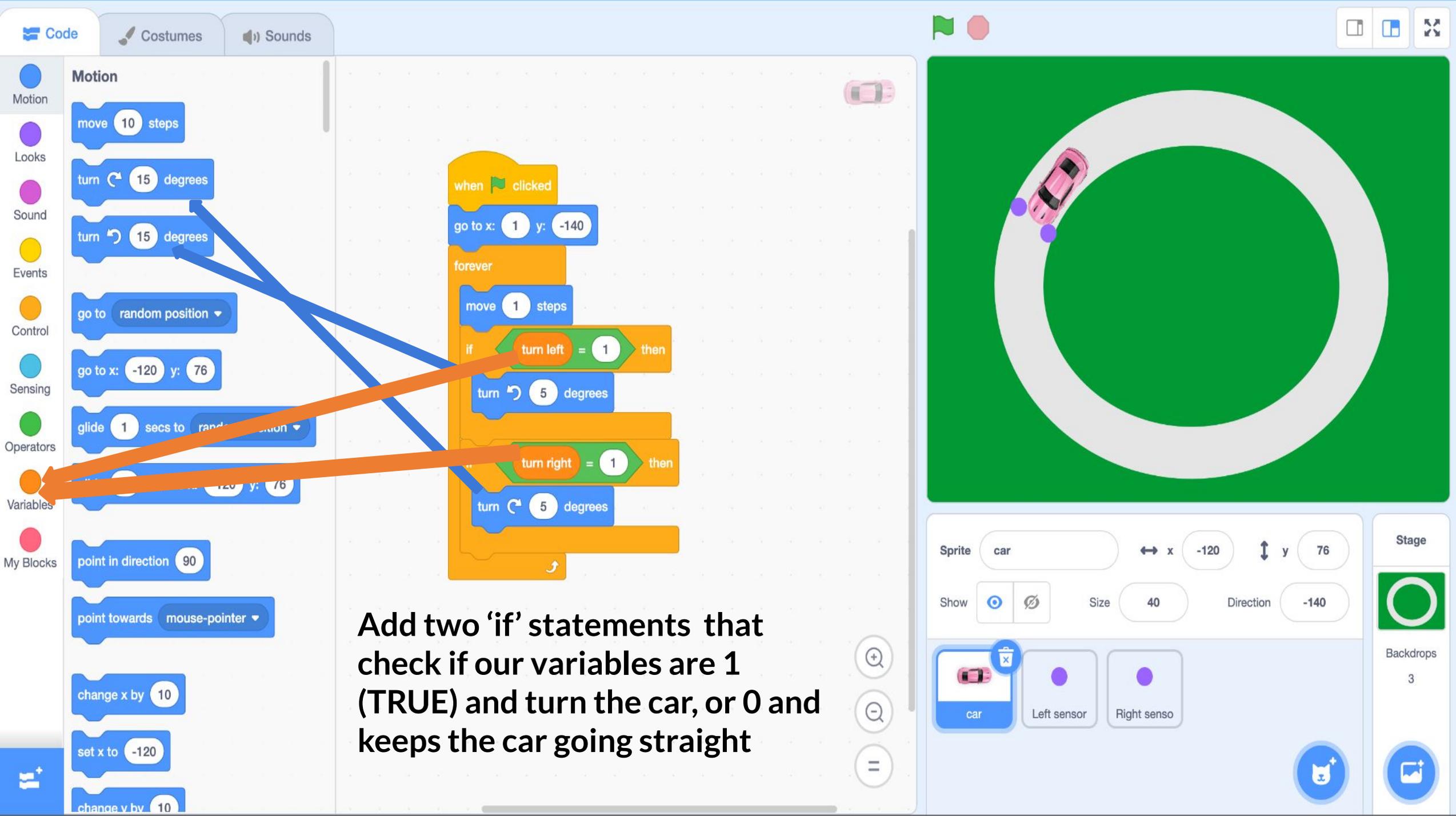
Direction: -140

Stage

Backdrops: 3

Left sensor Right sensor

Click on your car and go to 'Code'



Code Costumes Sounds

Motion

Looks

Sound

Events

Control

Sensing

Operators

Variables

My Blocks

when green flag clicked

go to x: 1 y: -140

forever

move 1 steps

if turn left = 1 then

turn 5 degrees

if turn right = 1 then

turn 5 degrees

Sprite car x: -120 y: 76

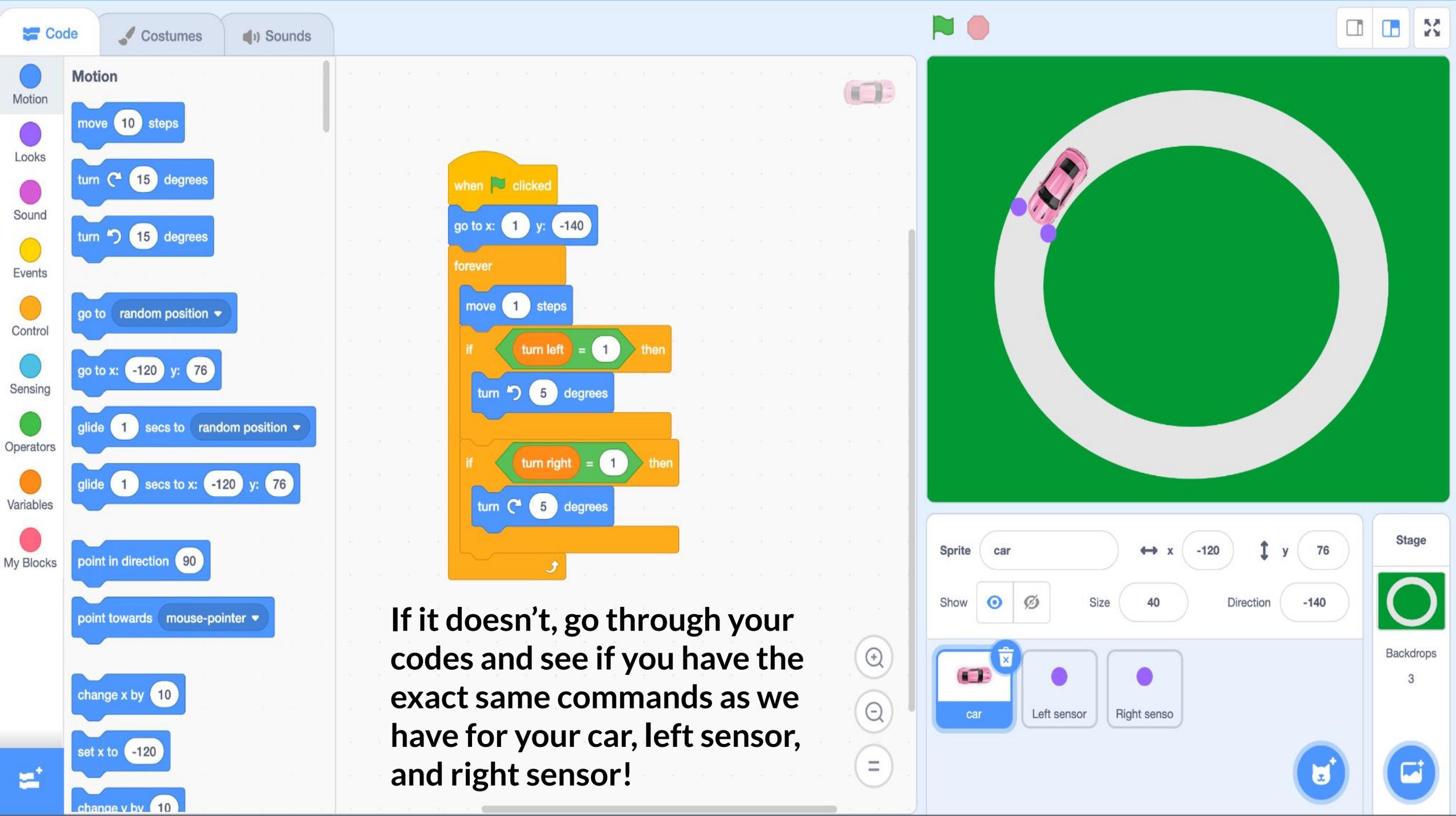
Show Size 40 Direction -140

Stage

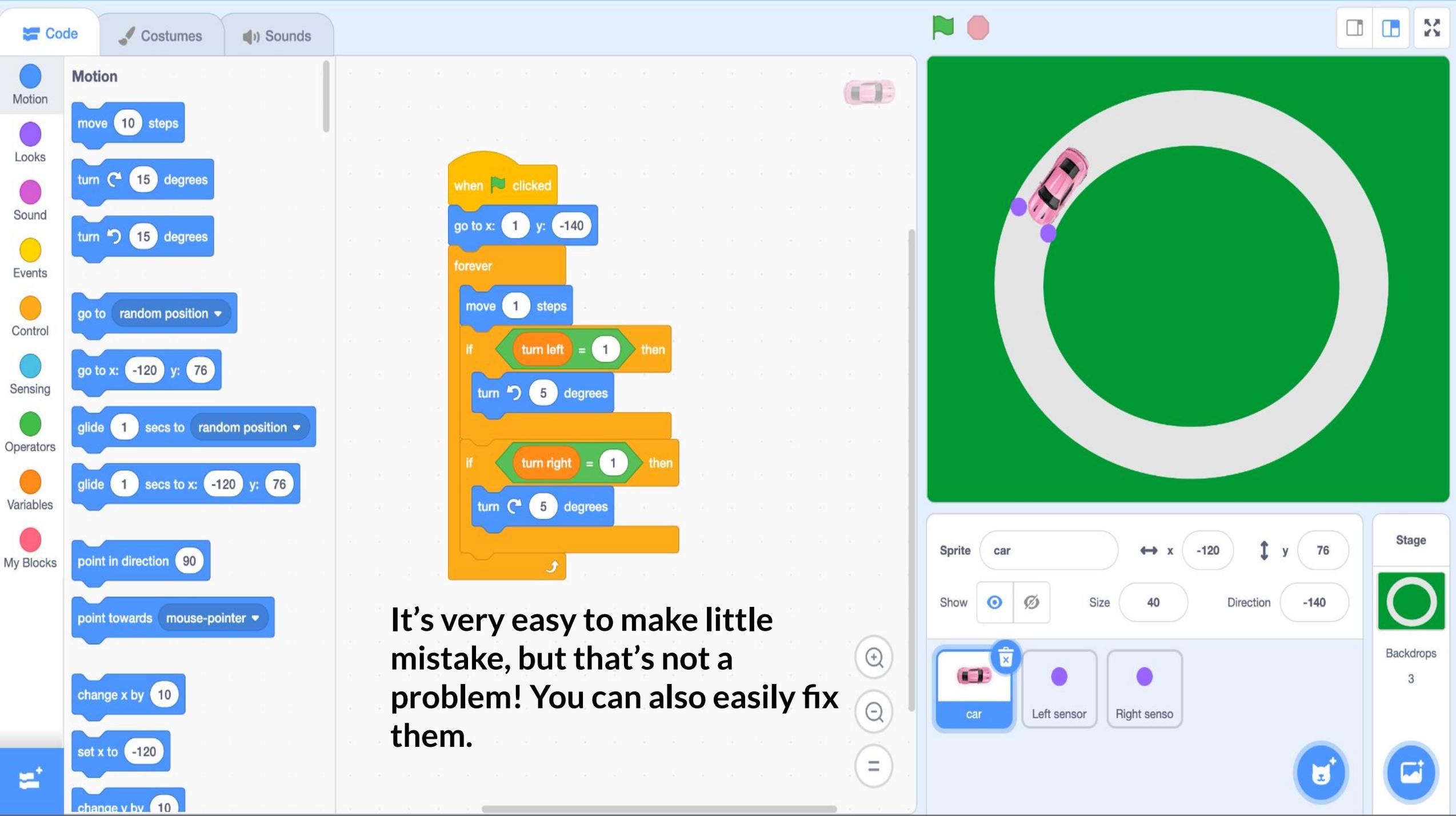
Backdrops 3

car Left sensor Right sensor

Now we are done! Check if your car follows the lane correctly



If it doesn't, go through your codes and see if you have the exact same commands as we have for your car, left sensor, and right sensor!



It's very easy to make little mistake, but that's not a problem! You can also easily fix them.

Code Costumes Sounds

Motion

Motion

Looks

Sound

Events

Control

Sensing

Operators

Variables

My Blocks

move 10 steps

turn 15 degrees

turn 15 degrees

go to random position

go to x: -120 y: 76

glide 1 secs to random position

glide 1 secs to x: -120 y: 76

point in direction 90

point towards mouse-pointer

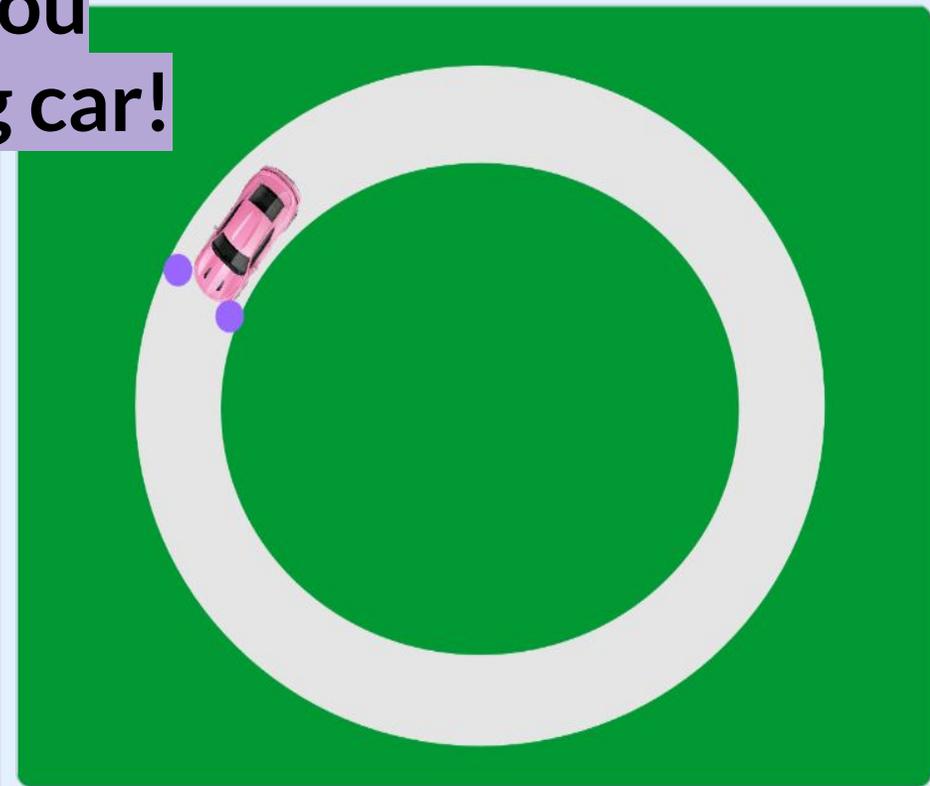
change x by 10

set x to -120

change y by 10

Does it work? Good job! You built your first self-driving car!

```
when clicked
  go to x: 1 y: -140
  forever
    move 1 steps
    if turn left = 1 then
      turn 5 degrees
    if turn right = 1 then
      turn 5 degrees
```



Sprite car

x: -120 y: 76

Size 40

Direction -140

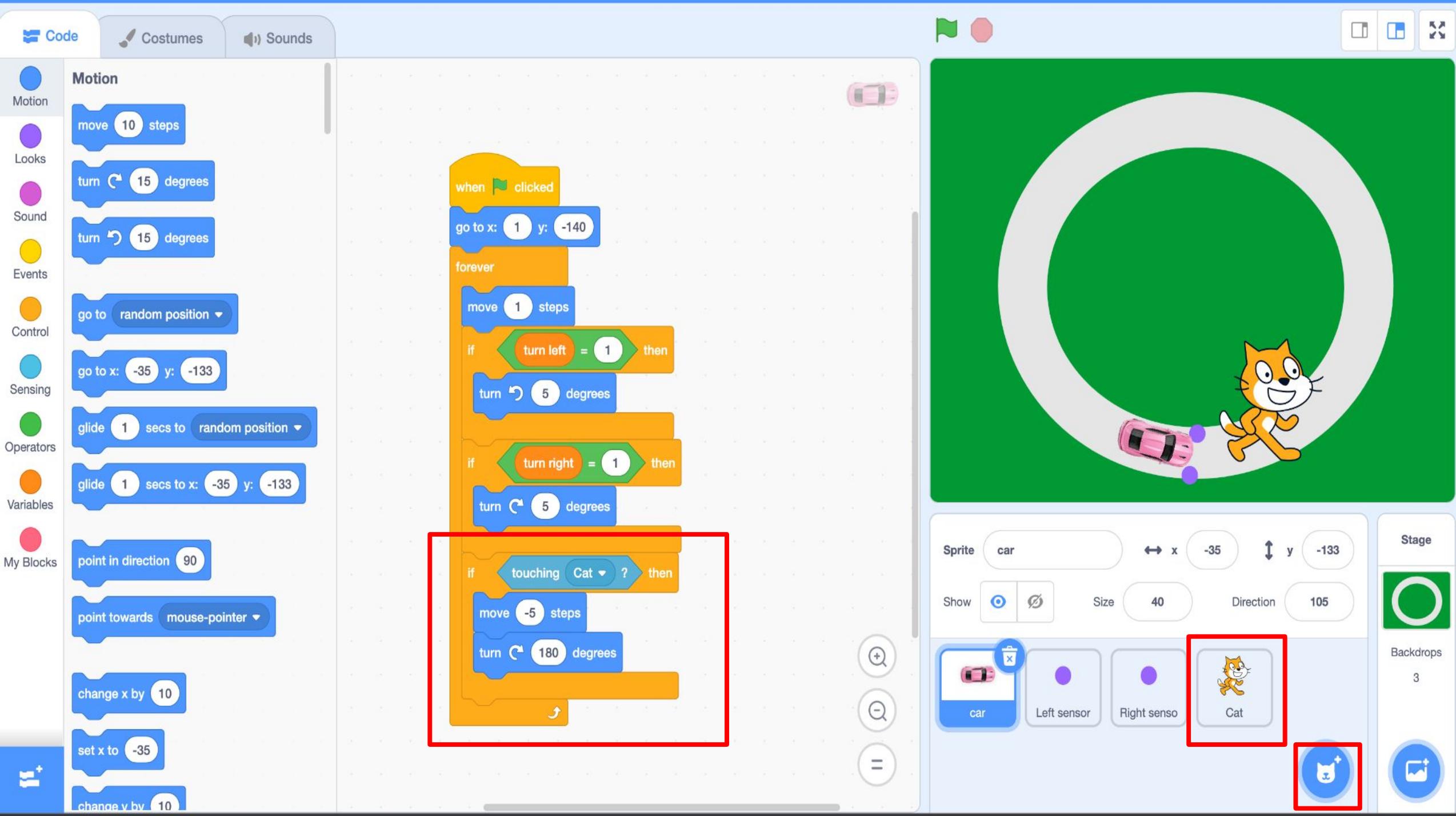
Left sensor

Right senso

Stage

Backdrops 3

Can you implement an ethical choice now? For example: turn around if you encounter a cat?

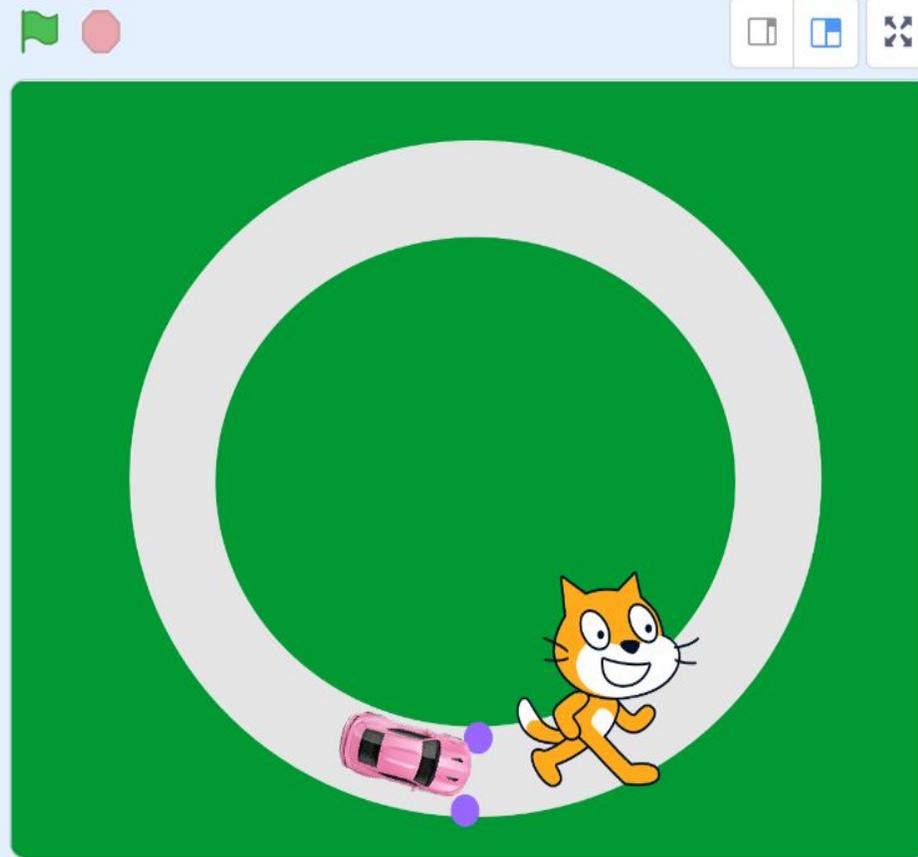


- Motion
- Looks
- Sound
- Events
- Control
- Sensing
- Operators
- Variables
- My Blocks

Motion

- move 10 steps
- turn 15 degrees
- turn 15 degrees
- go to random position
- go to x: -35 y: -133
- glide 1 secs to random position
- glide 1 secs to x: -35 y: -133
- point in direction 90
- point towards mouse-pointer
- change x by 10
- set x to -35
- change y by 10

```
when green flag clicked
  go to x: 1 y: -140
  forever
    move 1 steps
    if turn left = 1 then
      turn 5 degrees
    if turn right = 1 then
      turn 5 degrees
    if touching Cat? then
      move -5 steps
      turn 180 degrees
```



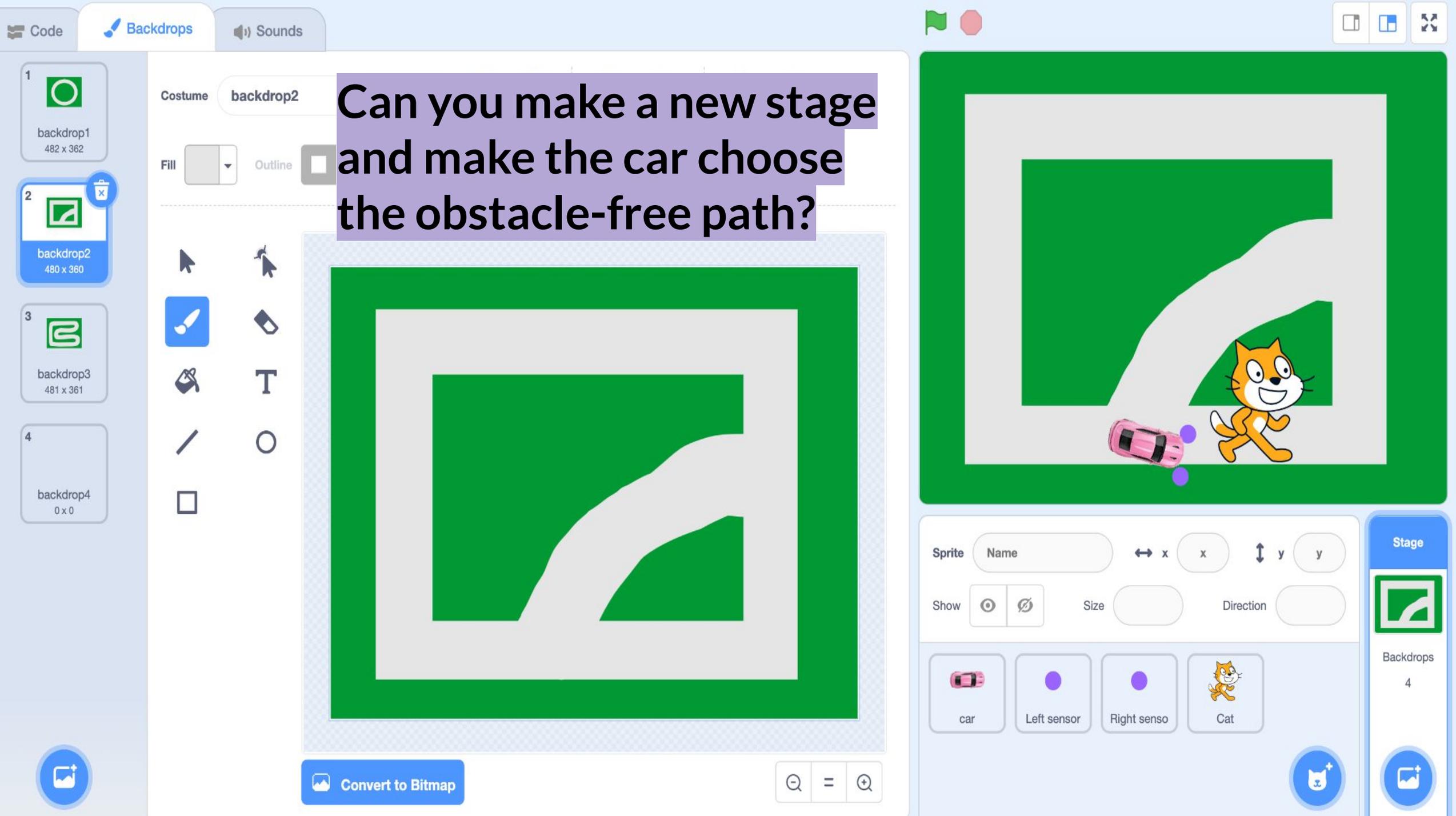
Sprite: car x: -35 y: -133

Show: Size: 40 Direction: 105

car Left sensor Right senso **Cat**

Stage

Backdrops: 3

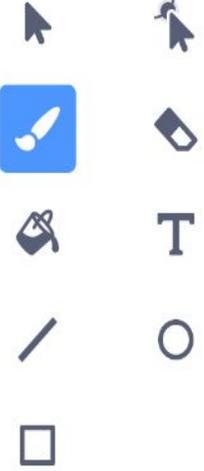


Can you make a new stage and make the car choose the obstacle-free path?

Costume backdrop2

Fill

Outline



Convert to Bitmap

Sprite Name x y

Show Size Direction



Stage



Backdrops

4

