

# Make a Scratch 3 Sense HAT game



**Sean McManus**

Author of *Scratch Programming in Easy Steps*, *Cool Scratch Projects in Easy Steps*, and *Mission Python*. Get free chapters at Sean's website.

[sean.co.uk](http://sean.co.uk)

Discover what's new in Scratch 3 as you build this exciting racing game, using your Raspberry Pi as a steering wheel

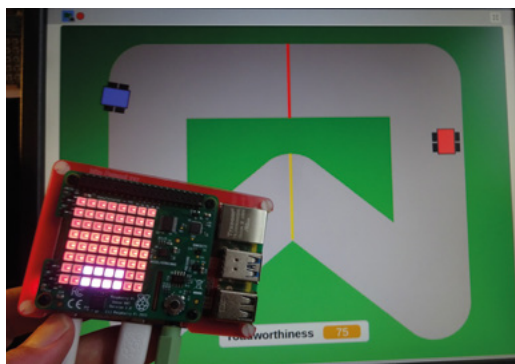
**S**cratch 3 is now on Raspberry Pi, with new blocks for the Sense HAT. In this game, you tilt your Raspberry Pi to turn left and right, and use the HAT's up and down joystick controls as an accelerator. Going off-road or colliding with the computer's racer damages your car. Can you complete three laps before your car is too beaten up to drive? If so, how fast is your qualifying time? This project will give you a tour through what's new in Scratch 3, including computer speech, new sound effects, and the extensions. Rev up!

## 01 Add extensions to Scratch

One of the best features of Scratch 3 is the extensions, which enable you to add new capabilities to Scratch. For example, you can add blocks for simple electronics, controlling the GPIO, video sensing, and using the Makey Makey input device. To simplify Scratch for first-timers, the Pen and Music blocks that used to be in the Blocks Palette have now been put into extensions too. For this project, you need to add the extensions for Raspberry Pi Sense HAT, Pen, and Text to Speech. The menu to add extensions is in the bottom left.

### You'll Need

- ▶ Raspberry Pi 4 (2GB or 4GB memory) [magpi.cc/rpi4](http://magpi.cc/rpi4)
- ▶ Sense HAT [magpi.cc/sense-hat](http://magpi.cc/sense-hat)
- ▶ Scratch 3 Desktop [magpi.cc/rNWKPO](http://magpi.cc/rNWKPO)
- ▶ PiBow or other case compatible with Sense HAT [magpi.cc/bbSsQo](http://magpi.cc/bbSsQo)
- ▶ Speakers to hear the engines roar



▲ The vertical bar chart in the middle of the LED matrix is the speedometer. The tiny joystick is your accelerator. Tilt to steer!

## 02 Draw the road

Hover over the icon in the bottom right to open the backdrop menu. Click the option to paint a backdrop and draw a green box that fills the Stage. Add **Listing 1** to Sprite1 (the cat). You'll need to use the menu in the first block to set up a new message called 'draw road'. Click the script to test it. You should see the road is drawn with two differently coloured checkpoints at the top and bottom.

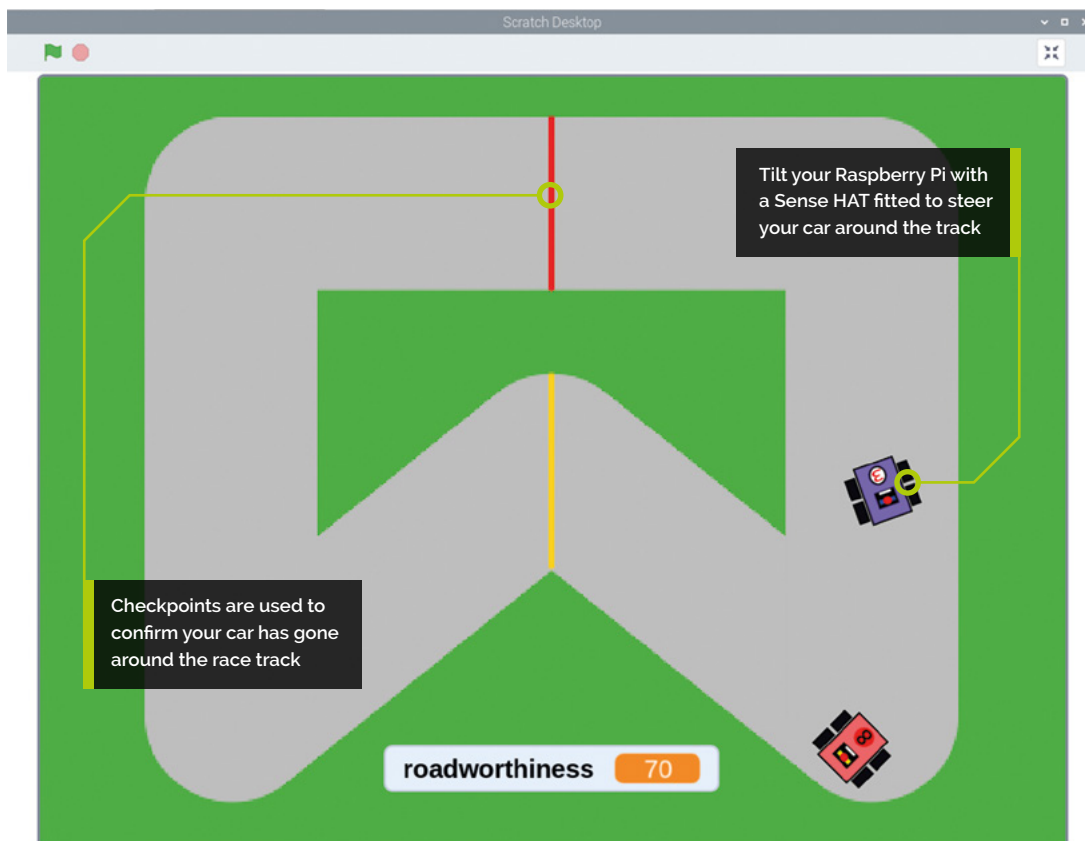
“ Tilt your Raspberry Pi to steer, and use the Sense HAT's up and down joystick controls as an accelerator ”

## 03 Draw the cars

Hover over the icon at the bottom of the Sprite List to add a sprite and choose Paint. The vector editor is now the default. Draw an overhead view of a race car, driving right. Centre your car on the cross-hairs target on the canvas, to make sure it turns correctly. Right-click your car in the Sprite List and duplicate it. With vector images, you can edit the shapes and colours afterwards, so you can easily make the second car look different. Click each car in the Sprite List and use the boxes above them to rename them to 'player car' and 'rival car'.

## 04 Start the rival car moving

The rival car is just an obstacle to avoid. Add the scripts in **Listing 2** to it. You'll need to create a new broadcast message for 'start race'. Click the green flag to position the car, then click the longer script. You should see it looping around the road, without veering off. If needed, adjust the size of the car in **Listing 2** (and **Listing 3** later) to fit both cars comfortably on the road. You might need to tweak the positions in the **glide** blocks in **Listing 2** too.



## 05 Create the variables

In Scratch 2, variables were created in the Data section of the Blocks Palette. Now, it's called Variables again, as it was in Scratch 1.4. Every project now starts with a variable created called **my variable** to help newcomers experiment. You need to make these variables: **half lap count**, **next checkpoint**, **qualifying time**, **roadworthiness**, **row to light**, **row to turn off**, and **speed**. They can be 'for all sprites'. Untick your variables in the Blocks Palette, except for **roadworthiness**, which we want to show on the Stage. Drag its box to the bottom middle of the Stage.

## 06 Set up the player's car

Click the player's car in the Sprite List and add **Listing 3** to it. Sense HAT and Text to Speech are both new in Scratch 3. The **display** block enables you to set a pixel pattern to show on the LED matrix, using the light colours you've set up with **set background** and **set colour** blocks. At the start, the game shows a chequered flag on the Sense HAT and the computer says, "Ready, Get Set, Go!"

## 07 Add player movement controls

You'll steer by holding your Raspberry Pi up, with the Sense HAT facing you, and tilting it left and right. You detect this axis of movement using the **pitch** block in the Sense HAT extension. Tilting left (up to 90 degrees) gives values of 0 to 90. Tilting right (up to 90 degrees) gives values of 360 to 270. **Listing 4** goes on your player's car. It turns your car if your Raspberry Pi is tilted left or right by more than 10 degrees, and uses the Sense HAT joystick to trigger the blocks to change your speed. To make the pink blocks, click My Blocks and use the button to make your own blocks, with the names you see in the listing. For the **touching color** blocks, click the colour box in the block and use the new, improved pipette to copy the checkpoint colours from the Stage.

## 08 Add sounds

There are fun new sounds to explore. For this game, we need the Cheer, Coin, Car Horn, Engine, and Skid effects. Click the Sounds tab on your player's car, and use the button in the bottom left to add them. You can now search for them by name.

## Top Tip

### Random gliding

We're not using it here, but there's a new block to glide to a random position, which you might find handy in your games.

### 09 Add speed controls

Find the pink hat blocks for

**define increase speed** and **define decrease speed**.

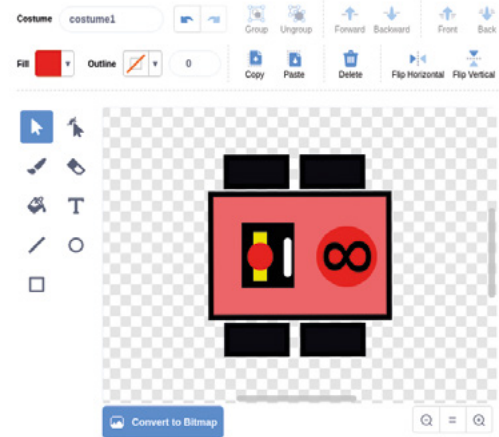
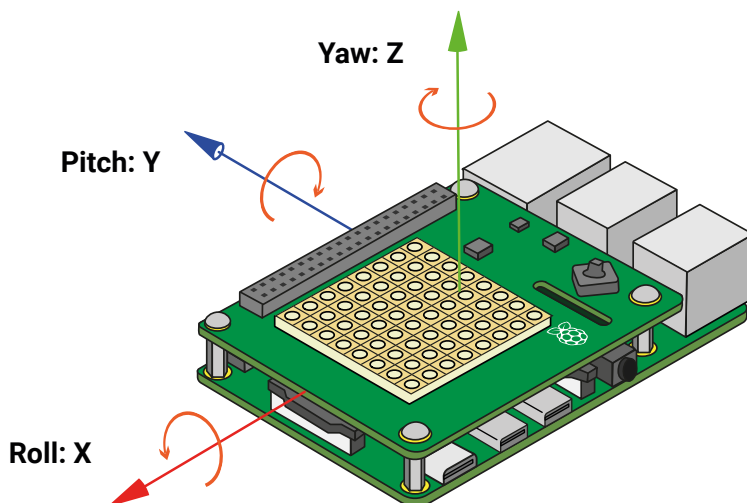
They were created for you in the Code Area when you made those blocks in Step 7. Use them to build **Listing 5**. You'll need to create new blocks (using My Blocks again) for 'speed lights on' and 'speed lights off'. The **set pixel** Sense HAT block enables you to light an individual LED using its x and y position (both numbered from 0 to 7, from the top left). We're using it to turn the LED matrix into a speedometer, with a bar graph that grows with your speed, which can range from 0 to 8. Scratch 3 adds new sound effects blocks. We're making the engine sounds higher when you speed up. Don't confuse the sound pitch (how low or high it is) with the Sense HAT pitch (the angle of the device). Your colours in the 'speed lights off' script should match the Sense HAT background colour in **Listing 3**. Click the colour box in the block to check the numbers.

“ We're turning the LED matrix into a speedometer, with a bar graph that grows with your speed ”

### 10 Add the lap counter

To confirm the car has done a lap of the track, we're using checkpoints. The car has to visit the red one, then the yellow one, and repeat until it's gone around three times. The next checkpoint variable remembers which checkpoint the car needs to visit next. The **half lap count** variable stores how many checkpoints have been

▼ When you lift your Raspberry Pi up to face you, it's still the pitch measurement you need if you move it like a see-saw above the desk



▲ The vector art editor is now the default. It makes it easy to modify the race cars to make two distinct designs

successfully crossed. Add **Listing 6** to your existing **define lap counter** block in the Code Area. Copy the red and yellow colours in the **touching color** blocks from the Stage.

### 11 Detect car damage

Your **roadworthiness** variable is like a health score. It goes down when you hit the grass or the other car. The two scripts in **Listing 7** continuously check whether you've hit something. If you have, they reduce your score and then pause briefly to stop your health sapping too fast. Use the pipette to copy the exact grass colour and the main body colour of your rival car (pink in our case) into your scripts. **Listing 7** goes on the player's car.

### 12 Add game-over sequences

There are two ways for the game to end. Either your roadworthiness gets to zero, or you successfully complete three laps. The **display text** block scrolls an appropriate message across the Sense HAT's LEDs. Attach **Listing 8** to the end of **Listing 4**, the script with movement controls on the player's car. Now you're ready to play! 🎮

## Top Tip

### Simplifying strings

There's a new Operators block called **apple contains a?** Use it to search a piece of text more easily.

# throttleandtilt.sb3

> Language: **Scratch**

**DOWNLOAD THE FULL CODE:**

[magpi.cc/github86](https://magpi.cc/github86)

Listing 1

```

when I receive draw road
  set pen color to black
  set pen size to 80
  go to x: -150 y: 120
  erase all
  pen down
  go to x: -150 y: -120
  go to x: 0 y: 0
  go to x: 150 y: -120
  go to x: 150 y: 120
  go to x: -150 y: 120
  pen up
  set pen size to 2
  set pen color to red
  go to x: 0 y: 80
  pen down
  go to x: 0 y: 180
  pen up
  set pen size to 2
  set pen color to yellow
  go to x: 0 y: -80
  pen down
  go to x: 0 y: -180
  pen up
  hide
  
```

Listing 2

```

when clicked
  go to x: 150 y: 100
  point in direction -90
  set size to 40 %

when I receive start race
  forever
    point in direction -90
    glide 6 secs to x: -140 y: 100
    point in direction -135
    glide 0.5 secs to x: -170 y: 80
    point in direction 180
    glide 4 secs to x: -170 y: -130
    point in direction -45
    glide 4 secs to x: 0 y: -230
    point in direction 135
    glide 4 secs to x: 140 y: -135
    point in direction -45
    glide 0.5 secs to x: 170 y: -120
    point in direction 0
    glide 4 secs to x: 170 y: 80
    point in direction -45
    glide 0.5 secs to x: 150 y: 100
  
```

Listing 3

```

when clicked
  set background to black
  set colour to white
  display
  set roadworthiness to 100
  show variable roadworthiness
  hide variable qualifying time
  set next checkpoint to red
  set half lap count to 0
  set speed to 0
  broadcast draw road and wait
  clear sound effects
  set size to 40 %
  go to x: 150 y: 140
  point in direction -90
  speak Ready
  speak Get set
  speak Go
  clear display
  set background to red
  broadcast start race
  
```

Listing 4

```

when I receive start race
  reset timer
  repeat until half lap count = 7 or roadworthiness < 1
  move speed + 2 steps
  if pitch < 90 and pitch > 10 then
    turn 10 degrees
  if pitch < 350 and pitch > 270 then
    turn 10 degrees
  if joystick pushed up = 7 then
    increase speed
  if joystick pushed down = 7 then
    decrease speed
  if touching color = yellow or touching color = red then
    lap counter
  
```

Listing 5

Listing 5

```

define increase speed
  change speed by 1
  set row to light to 5 - speed
  if speed > 8 then
    set speed to 8
  speed lights on
  set pitch effect to speed * 10
  start sound Engine

define decrease speed
  set row to turn off to 5 - speed
  speed lights off
  change speed by -1
  if speed < 0 then
    set speed to 0
  set pitch effect to speed * 10
  start sound Engine

define speed lights off
  set pixel x: 2 y: row to turn off to red
  set pixel x: 3 y: row to turn off to red
  set pixel x: 4 y: row to turn off to red
  set pixel x: 5 y: row to turn off to red

define speed lights on
  set pixel x: 2 y: row to light to green
  set pixel x: 3 y: row to light to green
  set pixel x: 4 y: row to light to green
  set pixel x: 5 y: row to light to green
  
```

Listing 7

```

when I receive start race
  forever
    if touching color = green and roadworthiness > 0 then
      change roadworthiness by -5
      start sound Sire
      wait 0.5 seconds

when I receive start race
  forever
    if touching color = red and roadworthiness > 0 then
      change roadworthiness by -10
      start sound Car Horn
      wait 0.5 seconds
  
```

Listing 8

```

if roadworthiness < 1 then
  speak Failed to qualify
  display text Game Over
else
  set qualifying time to timer
  show variable qualifying time
  clear sound effects
  start sound Cheer
  speak Congratulations! You qualified
  display text Qualified
stop all
  
```