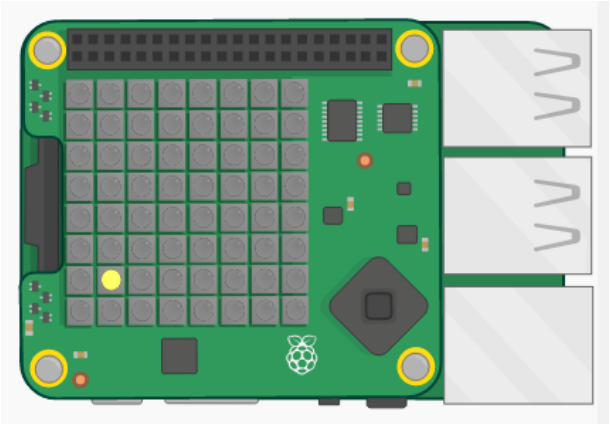
Raspberry Pi Projects

Where's the Treasure?

Introduction:

In this project you will use the joystick and LED Matrix on the Sense HAT to play a memory game. The Sense HAT will show a gold coin and you have to remember where it was and use the joystick to find the hidden treasure.



To play the game press Run and watch to see where the yellow dot appears - this is the treasure! Then use the arrow keys on the keyboard to move the white dot to where you think the treasure is hidden. When you get there, press return. You'll see a green dot if you were correct and a red dot if you got it wrong. You'll get 10 tries and then a score out of 10.

Note that when you are using the Sense HAT emulator you use the arrow keys and return instead of the joystick on the Sense HAT.

Additional information for club leaders

If you need to print this project, please use the Printer friendly version (https://projects.raspberrypi.org/en/projects/wheres-the-treasure/print).

Club leader notes

Introduction:

In this project, children will learn how to use the Sense HAT joystick to create a memory game. The Sense HAT shows a gold coin and you have to remember where it was and use the joystick to find the hidden treasure.

Online Resources

This project uses Python 3. We recommend using Trinket (https://trinket.io/) to write Python online. This project contains the following Trinkets:

- 'Where's the Treasure?' Starter Trinket – jumpto.cc/treasure-go (http://jumpto.cc/treasure-go)

There is also a trinket containing the completed project:

- 'Where's the Treasure' Finished – trinket.io/python/79ac6a377d (https://trinket.io/python/79ac6a377d)

Offline Resources

This project can also be completed offline (https://www.codeclubprojects.org/en-GB/https://projects-static.raspberrypi.org/projects/wheres-the-treasure/e5a6a8e04a0a553f6b412e6b7b6c17962857eb20/en/resources/physical-sense-hat/) on a Raspberry Pi computer with a Sense HAT. You can access the project resources by clicking

the 'Project Materials' link for this project. This link contains a 'Project Resources' section, which includes resources that children will need to complete this project offline. Make sure that each child has access to a copy of these resources. This section includes the following files:

- treasure/treasure.py

You can also find a completed version of this project in the 'Volunteer Resources' section, which contains:

- treasure-finished/treasure.py

(All of the resources above are also downloadable as project and volunteer `.zip` files.)

Learning Objectives

- Sense HAT joystick;
- Boolean logic;

This project covers elements from the following strands of the Raspberry Pi Digital Making Curriculum (http://rpf.io/curriculum):

- Combine programming constructs to solve a problem. (https://www.raspberrypi.org/curriculum/programming/builder)

Challenges

- Customize the game - use different colours or add text messages.
- Make it harder - show the coin for less time or confuse players by filling the display with coins after showing the hidden coin.

Project materials

Project resources

- .zip file containing all project resources (https://projects-static.raspberrypi.org/projects/wheres-the-treasure/e5a6a8e04a0a553f6b412e6b7b6c17962857eb20/en/resources/treasure-project-resources.zip)
- Starter project (http://jumpto.cc/treasure-go)
- Offline starter Python file (https://projects-static.raspberrypi.org/projects/wheres-the-treasure/e5a6a8e04a0a553f6b412e6b7b6c17962857eb20/en/resources/treasure-treasure.py)

Club leader resources

- .zip file containing all completed project resources (https://projects-static.raspberrypi.org/projects/wheres-the-treasure/e5a6a8e04a0a553f6b412e6b7b6c17962857eb20/en/resources/treasure-volunteer-resources.zip)
- Online completed Trinket project (https://trinket.io/python/79ac6a377d)
- treasure-finished/treasure.py (https://projects-static.raspberrypi.org/projects/wheres-the-treasure/e5a6a8e04a0a553f6b412e6b7b6c17962857eb20/en/resources/treasure-finished-treasure.py)

Hiding the treasure

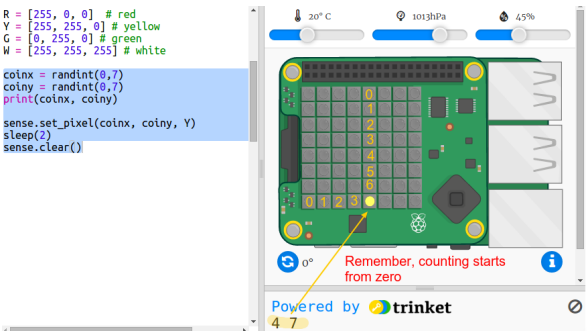
First let's show a yellow coin at a random pixel and then hide it.

- Open the Where's the Treasure? Starter Trinket: jumpto.cc/treasure-go (http://jumpto.cc/treasure-go).

- Take a look at the code that has been included for you. This sets up the Sense HAT and the libraries you will be using, and also includes some helper code so that you can get to the interesting stuff more quickly:

```
#!/bin/python3

from sense_hat import *
from time import sleep
from random import randint

sense = SenseHat()
sense.clear()

# Just return the actions we are interested in
def wait_for_move():
  while True:
    e = sense.stick.wait_for_event()
    if e.action != ACTION_RELEASED:
      return e

R = [255, 0, 0]  # red
Y = [255, 255, 0] # yellow
G = [0, 255, 0] # green
W = [255, 255, 255] # white
```

- Let's display a yellow coin at a random location and then hide it. The `coinx` and `coiny` variables are the x and y coordinates of the coin. Go to the bottom of the script and add the following code:



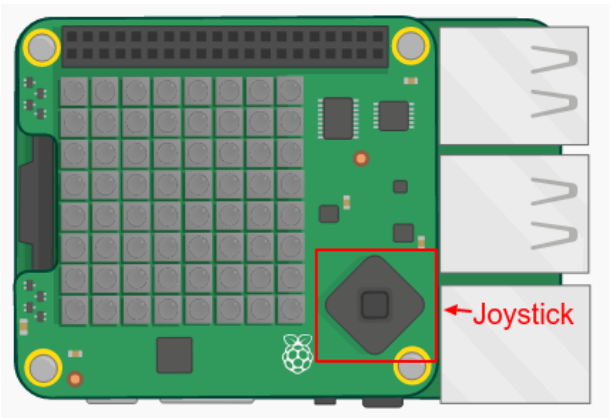
  Make sure you use a capital letter `Y`.

- Run your code a few times to see the coin appear and disappear at random locations.

Finding the treasure

Now let's show the player as a white pixel. You'll need to use the Sense HAT joystick to navigate to where you think the treasure is hidden.

The physical Sense Hat has a mini joystick. You can see a picture of it in the emulator:

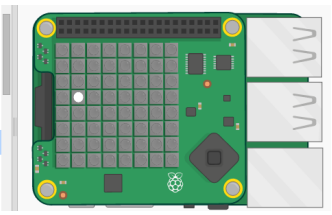


In the emulator you can use the arrow keys for the direction buttons on the joystick and Enter (Return) for pressing the middle button.

Now let's add a pixel that the player can move to where they think the treasure is hidden. The player is a white pixel.

- Now display the player's location using a white pixel:

```
W = [255, 255, 255] # white

coinx = randint(0,7)
coiny = randint(0,7)
print(coinx, coiny)

sense.set_pixel(coinx, coiny, Y)
sleep(2)
sense.clear()

x = randint(0,7)
y = randint(0,7)
sense.set_pixel(x, y, W)
```



x and y are the player's coordinates.

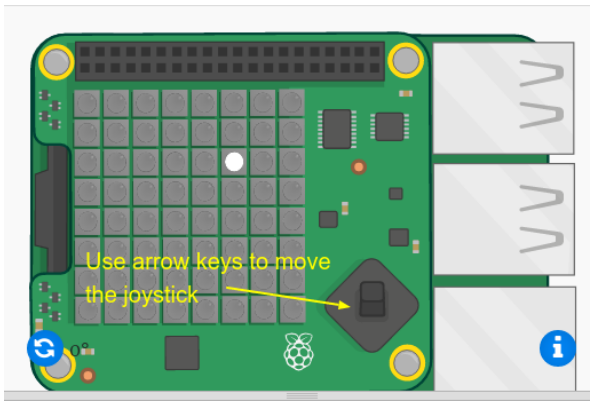
- Let's get the white pixel moving using the joystick. Every time the player presses one of the arrow keys on the joystick we need to clear the current pixel and draw one at the new location. Let's start by allowing the player to move in the y direction (up and down):

```
sense.set_pixel(x, y, W)

while True:
    e = wait_for_move()
    sense.clear()

    if e.direction ==  DIRECTION_UP:
        y = y - 1
    elif e.direction ==  DIRECTION_DOWN:
        y = y + 1
    sense.set_pixel(x, y, W)
```

wait until the joystick is moved

draw player at new location

- Test your code by pressing the up and down arrows on the keyboard.



What happens when you reach the top edge and press up?

```
sense.set_pixel(x, y, W)
```

Powered by trinket

```
2 3
```

ValueError: Y position must be between 0 and 7 on line 49 in main.py

If the y position goes below 0 or above 7 then you'll get an error when you try and set the pixel colour.

- Let's add a check to make sure the pixel stays on the display:

```
if e.direction ==  DIRECTION_UP and y > 0:
    y = y - 1

elif e.direction ==  DIRECTION_DOWN and y < 7:
    y = y + 1
```

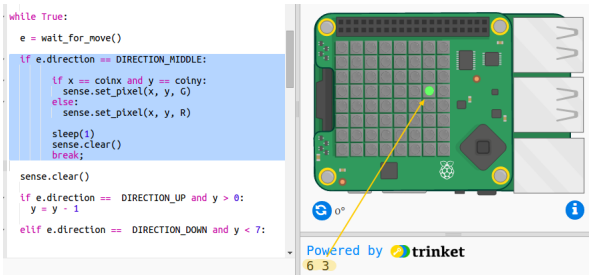
- Now let's add movement in the x direction. Add the highlighted code:

```
    if e.direction ==  DIRECTION_UP and y > 0:
      y = y - 1

    elif e.direction ==  DIRECTION_DOWN and y < 7:
      y = y + 1

    elif e.direction ==  DIRECTION_LEFT and x > 0:
      x = x - 1

    elif e.direction ==  DIRECTION_RIGHT and x < 7:
      x = x + 1

    sense.set_pixel(x, y, W)
```

- Once you have moved to the location where you think the treasure is hidden you need to press the middle button on the joystick. In the emulator you'll need to press Enter (Return) on the keyboard.

  If the player is at the same location as the treasure then they've found it and the pixel goes green for 1 second.

  If the player has picked the wrong location then the pixel goes red for 1 second.



  `break` means we don't need to wait for more events after the player has chosen a location, we can stop repeating the loop.

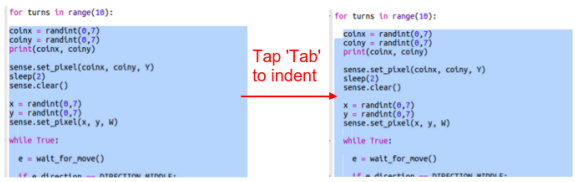
Keeping score

At the moment you only get one go at finding the treasure. Let's give the player 10 turns and keep a score.

- Now you'll need a `for` loop to let the player have 10 goes at finding the treasure:

```
R = [255, 0, 0]  # red
Y = [255, 255, 0] # yellow
G = [0, 255, 0] # green
W = [255, 255, 255] # white

for turns in range(10):

coinx = randint(0,7)
coiny = randint(0,7)
print(coinx, coiny)
```

- In Python, code must be indented to be inside a loop. You don't need to indent the lines one at a time though! Highlight all of the code after the `for` loop then press the 'Tab' on the keyboard and it will all be indented.



  Make sure all of the code after the `for` gets indented, right to the bottom of the code.

- Next add a score variable which starts at zero:

```
R = [255, 0, 0]  # red
Y = [255, 255, 0] # yellow
G = [0, 255, 0] # green
W = [255, 255, 255] # white

score = 0

for turns in range(10):
```

- You'll also need to add one to the score when a player chooses the correct location:

```
    if x == coinx and y == coiny:
      sense.set_pixel(x, y, G)
      score += 1
    else:
      sense.set_pixel(x, y, R)

    sleep(1)
    sense.clear()
```

- And finally, let's display the score at the end.

```
  elif e.direction ==  DIRECTION_RIGHT and x < 7:
    x = x + 1

  sense.set_pixel(x, y, W)

sense.show_message("Score: " + str(score))
```

No indentation

Make sure there's no indentation before this code, it needs to run after the `for` loop has run 10 times and the game is finished.

- Now play the game. Can you score 10 out 10?

Challenge: Customize the game

Try using different colours or showing a different message depending on how well the player did.

Challenge: Make it harder

Are you finding the game too easy? Why not make it harder.

You could show the coin for less time. `sleep(2)` shows the coin for 2 seconds. What about `sleep(0.5)`?

Or how about confusing the player by making all the pixels turn yellow before they get to pick the location? Use `sense.clear(Y)` to fill the screen with yellow coins after showing the player where the coin is hidden, you'll also need `sleep(1)` or however many seconds you want to show the yellow screen for.