

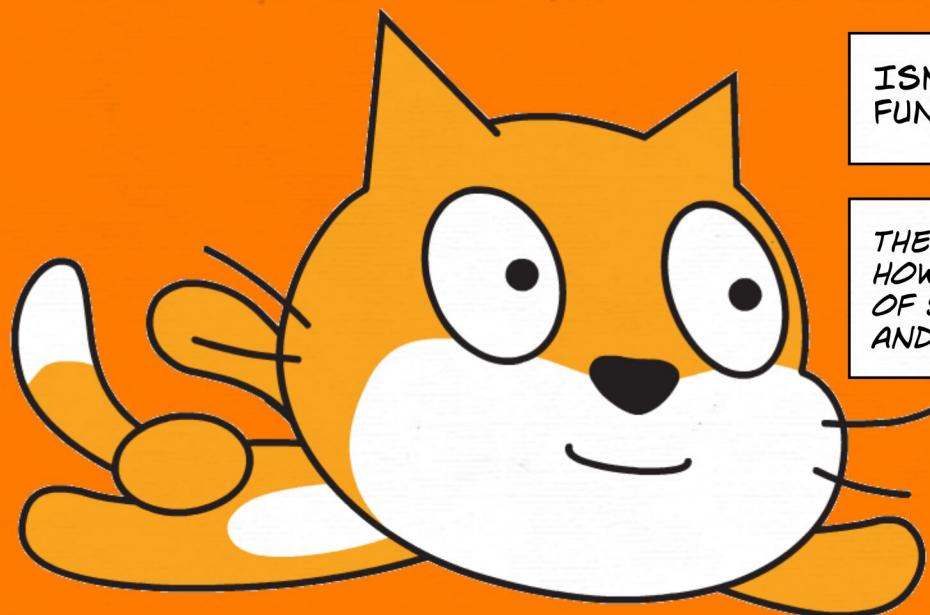
S
C
R
A
T
C
H

P
R
I
M
E
R

MAKERDEMY
LEARN EXPLORE MAKE

For age
8 to
infinity

SCRATCH'N RASPBERRYPI



ISN'T IT CREATING A GAME
FUN AND COOL?

THE INCREDIBLE STORY OF
HOW YOU BECOME THE KING
OF SCRATCH PROGRAMMING
AND RULER OF RPI...

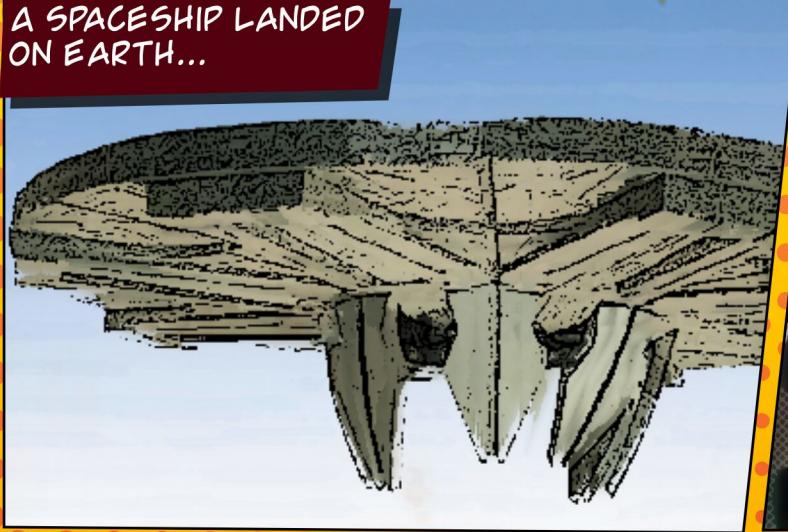
RACHEN RAVICHANDRAN

GAMES

STORIES

AND MORE

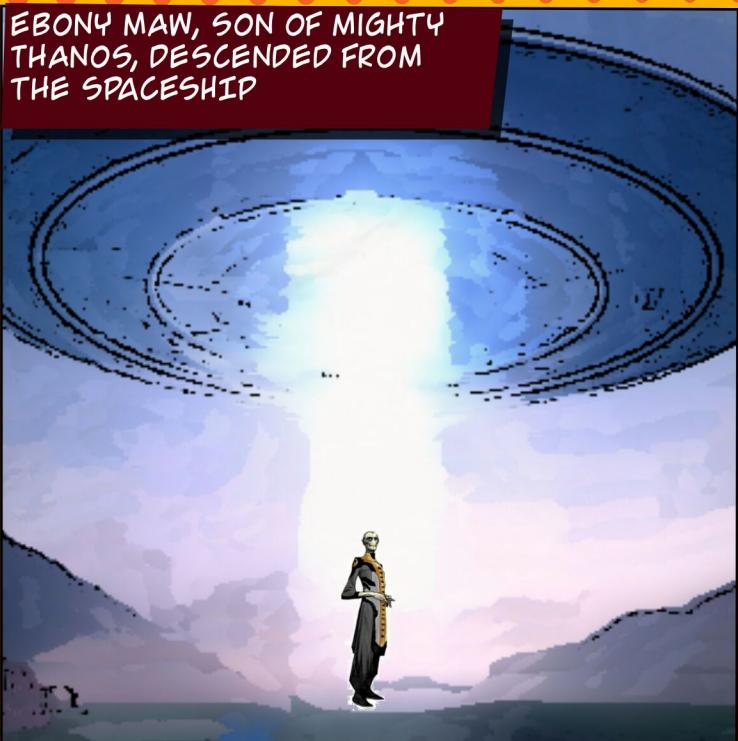
A SPACESHIP LANDED
ON EARTH...



PEOPLE SCURRIED THROUGH
THE CROWDS OF CARS FOR THE
FEAR OF WHAT LAY BEHIND



EBONY MAW, SON OF MIGHTY
THANOS, DESCENDED FROM
THE SPACESHIP



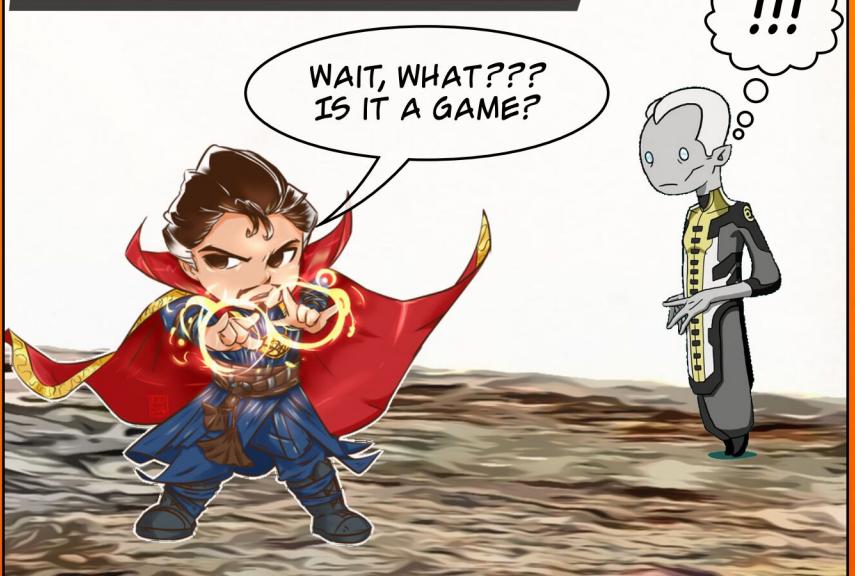
DOCTOR STRANGE, THE MASTER
OF MYSTICAL ARTS, WAS
ALREADY HERE TO WELCOME
EBONY WITH HIS OWN STYLE



GIVE ME THE
TIME STONE,
STONEKEEPER!



PRESS 'A' TO ATTACK AND 'S' TO DEFEND.



YOU ARE
TRESPASSING IN
THIS CITY AND ON
THIS PLANET.

LEAVE NOW!

WAIT, WHAT???
IS IT A GAME?

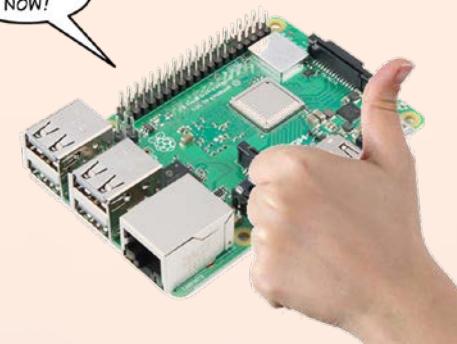
Yes, we are going to create a game and save the time stone from Ebony Maw. Isn't it creating a game mysterious for kids? Isn't it suitable only for grownups with extreme coffee addiction who code hundreds of lines of programs? But, isn't it fun and cool to create one?

Scratch is right here to help us. It is extremely easy for us to create games, stories, Mother's day cards, science reports or even a Kinect (that's what we are going to do today.)

Scratch is created by good people at MIT and is a way of programming where you'll be having fun with dragging blocks and fitting them like a brick building; instead of the nitpicking, syntax based codes which will haunt you with errors. Sounds interesting? Let's get down into the business...



We'll create the game 'Doctor Strange Vs Ebony Maw' (to see how this works, play the video 'game.mp4' or click the above image.) I am assuming that you have a Raspberry Pi 3 (or 2) for creating this game. If not, I strongly recommend you to buy one.

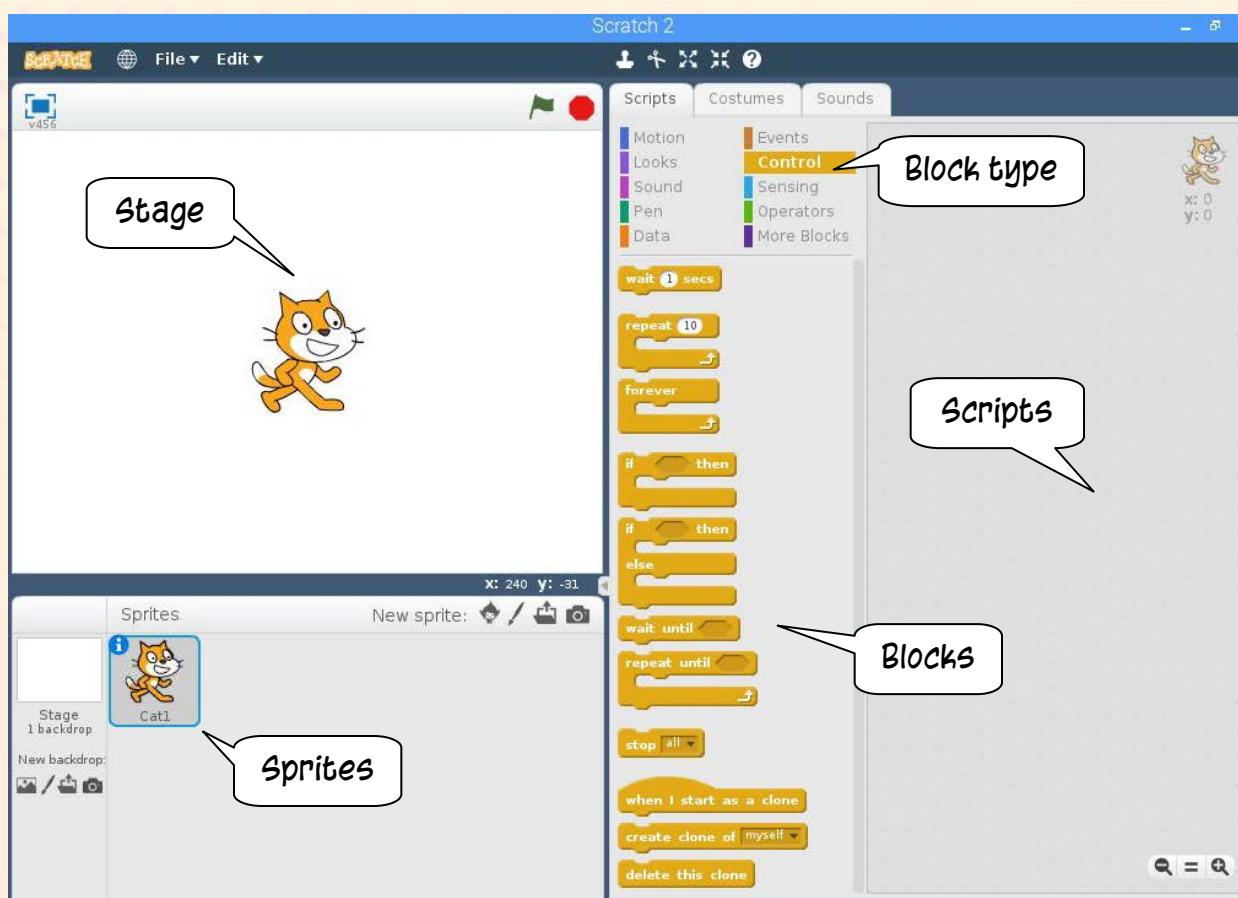


Raspberry Pi (RPI) is an extremely small computer, much like your big old CPU. You can do anything with RPI as much as you can do with a computer and actually more (a bit about that later)

Scratch 2 (a version of Scratch) comes preinstalled with RPI in recent updates. If you don't have an RPI, you can download and install Scratch 2 from this [link](#) to use it in your windows laptop.

No more waiting! Open 'Scratch 2' from Programming Menu in RPI.

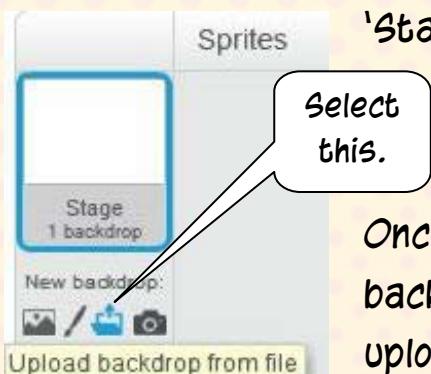
You'll get a window like this:



Right-click the Cat and delete it.

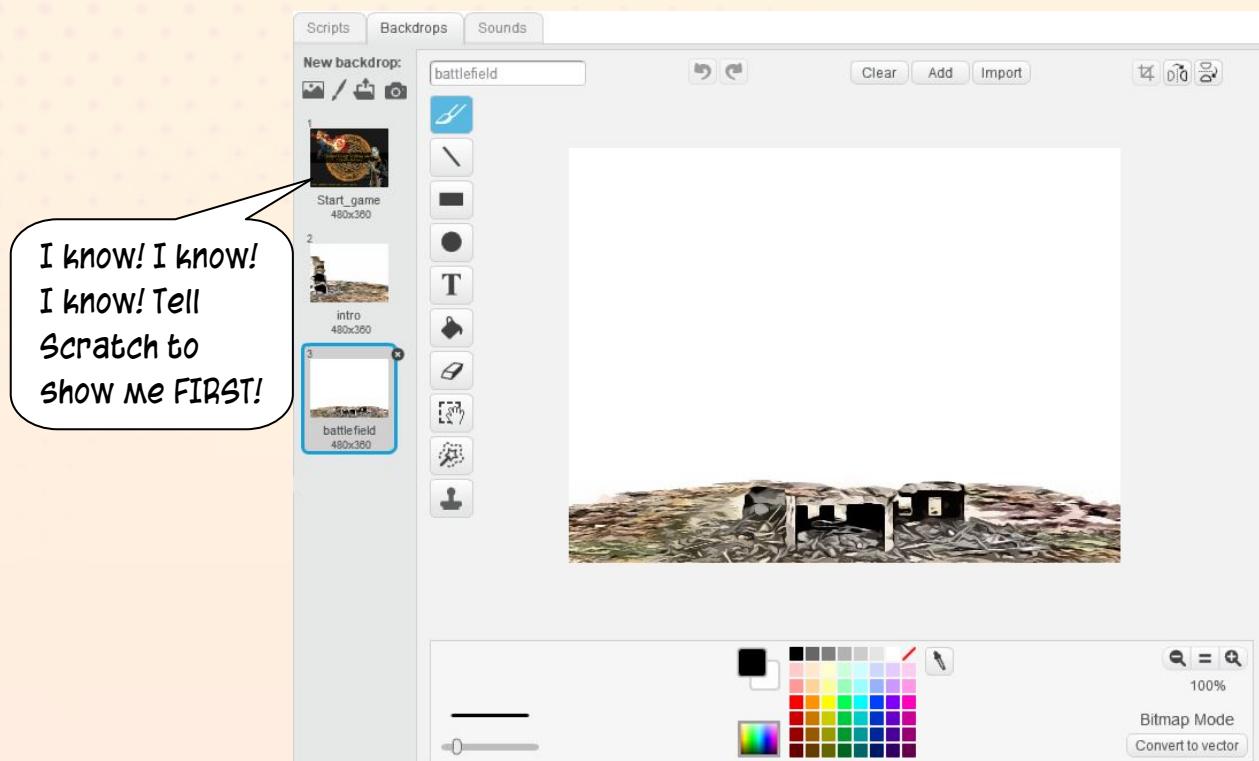
When we start the game, we need an introduction or title page to it. So, upload the image

'Start_game.png', provided in 'images for game' folder, as your backdrop. Delete the white backdrop.



Once the game has started, we need the title backdrop to change to battlefield backdrop. So, upload 'intro.png' and 'battlefield.png' into Scratch. [HINT: Press 'Ctrl' + click to select multiple images]

Now, you'll find the 3 backdrops in the 'Backdrops' tab. But when the game starts, how will Scratch know which backdrop to show first?



That's right! You need to tell Scratch what to do using **SCRIPTS**. Click the 'Scripts' tab. You can create your own scripts by dragging and placing the block you want into the Scripts area.

The block in the right will switch the backdrop to 'Start_game'. But when? When should it switch?

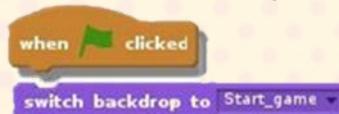
...when we press the flag .

switch backdrop to battlefield

Drag and place
this block from
LOOKS menu

Start_game
intro
battlefield
next backdrop
previous backdrop

To do this, add the block as given below



Press Flag to see
what happens.

But the game should wait till the player presses 'space bar'. So, add the blocks given below to your script.



Insert the KEY
block (SENSING
menu) into the
wait until block

'SPACE' IS PRESSED which is the condition.

A WAIT UNTIL block waits until a condition in the gap is complete. Here, it waits until the KEY

Now we'll add our characters, Ebony Maw and Doctor Strange, into the game. Characters, objects or anything that you'll use with the scripts are called SPRITES.

Sprites

New sprite:

 Upload sprite from file

Select
this



First, let's add Ebony to our game. Add the image 'ebony_intro.png' from the Sprites tab. You'll now find this image in the Sprites list.

Drag and place this
sprite wherever you
need on the stage.

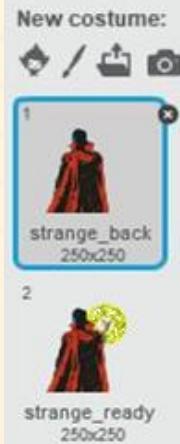
In Scripts tab of ebony_intro sprite, add the blocks below so that it'll SHOW this sprite when the backdrop changes to intro.png...

```
when backdrop switches to intro
show
say Hello Stonekeeper! Give me the time stone! for 3 secs
```

Change the text to 'Hello Stonekeeper! Give me the time stone!' and '2' secs to '3' secs.



This SAY block will show a bubble speech which will disappear after 3 seconds. Now we need Doctor Strange to reply to this after 3 seconds. Before creating a reply script, upload the image 'strange_back.png' in the Sprites tab. Rearrange Doctor Strange in the stage.



Upload another image 'strange_ready.png' in the costume tab for the same sprite. We need two images for Doctor Strange: strange_back and strange_ready.

```
when backdrop switches to intro
switch costume to strange_back
show
wait 3 secs
say You're trespassing in this city and on this planet for 3 secs
switch costume to strange_ready
say Get out now! for 2 secs
wait 2 secs
hide
```

Switches the costume to strange_ready

When Ebony says he wants the time stone, Doctor Strange has to get his shield ready which is shown in the second image. These group of images in a sprite are called costumes. Select the Strange's Sprite and go to Scripts tab...

Go to ebony_intro's script tab and add these blocks...

Switches the backdrop to 'battlefield' when game starts.

```
when backdrop switches to intro
show
say Hello Stonekeeper! Give me the time stone! for 3 secs
wait 5 secs
say Not without the time stone, Stonekeeper. for 2 secs
hide
switch backdrop to battlefield
```

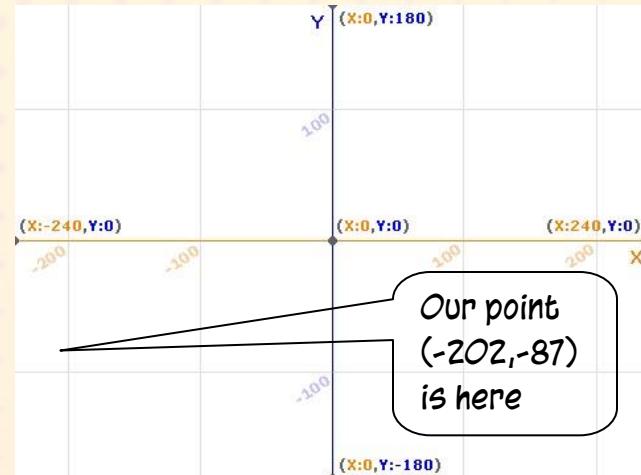
Since we have created an intro conversation, let's start making the game. Import 'doctor_fight.png' image into the Sprite tab and add these blocks in its Scripts:



Add this block from MOTION menu and set x as -202 and y as -87

The GO TO block takes the sprite to that x, y-axis coordinates. The stage is divided into x- and y-axis like a graph sheet. The centre of the stage is (0,0).

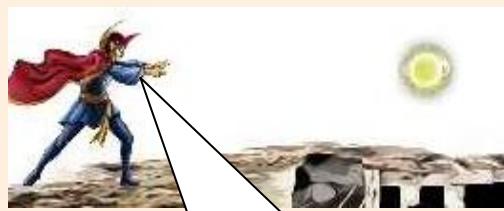
Before going any further, let's take some time to understand how our game works...



These bricks are sent by Ebony Maw to attack Strange.



Doctor protects himself from bricks using Shield by pressing 's' key which causes the bricks to explode.



Doctor Strange and Ebony Maw have three lives each.

Once the fire or shield key is pressed, it can be activated again only after 4 seconds. The boxes (given below) indicate whether the shield and firing are activated or deactivated.

Doctor's Life

3

Shield

ON

Fire

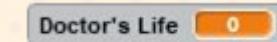
ON

Ebony's life

3

Once Ebony or Strange gets attacked, their life has to decrease by 1. How do we keep track of this?

Using variables... Variables are like containers. You can store numbers, names or anything in them. For example, you can store the life of Doctor in a variable named 'Doctor's Life'. To create a variable, go to DATA menu, click 'Make a Variable' and type the variable name as 'Doctor's Life'.

The value stored in the variable is displayed in this box: 

Similarly, create other variables: Shield, Fire, Ebony's Life.

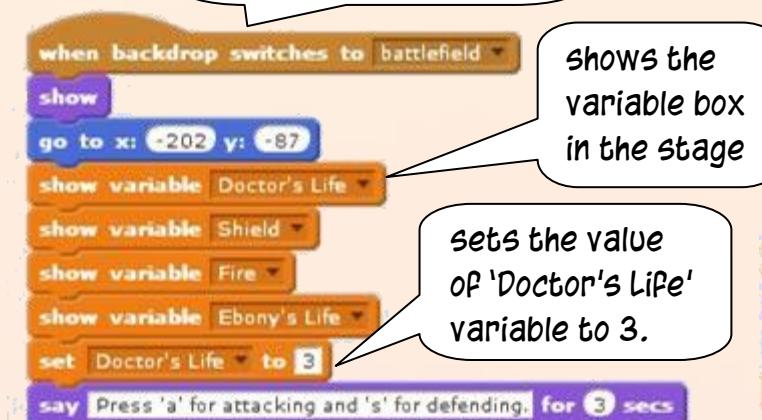
A list of blocks are added to the DATA menu when a variable is created



Arrange the variables in the stage like below:

Then...

...in Scripts tab for doctor_fight sprite...



shows the variable box in the stage

sets the value of 'Doctor's Life' variable to 3.

In Scripts tab for ebony_fight sprite...



```

when backdrop switches to battlefield
  set Shield to [ON v]
  show
  go to x: -137 y: -69
  set Shield to [OFF v]
  wait (2) secs
  hide
  wait (2) secs
  set Shield to [ON v]

```

The shield MUST appear in the screen for 2 seconds and it activates again after 2 seconds.

Click Flag and observe your improvement so far. Import 'shield.png' to Sprite tab and in the Script tab... (Keep the two blocks in the left separately)

But all these blocks are to be executed only IF THE KEY 'S' IS PRESSED. So, add the IF block as given below.

This block will run only once. So how do we keep checking this condition FOREVER?

The IF ... THEN block from CONTROL menu checks this condition for you...

Using Forever block... it repeats the block inside it until the end of the game.

```

if [key S v pressed?]
then
  show
  go to x: -137 y: -69
  set Shield to [OFF v]
  wait (2) secs
  hide
  wait (2) secs
  set Shield to [ON v]

```

```

when backdrop switches to battlefield
  set Shield to [ON v]
  forever
    if [key S v pressed?]
    then
      show
      go to x: -137 y: -69
      set Shield to [OFF v]
      wait (2) secs
      hide
      wait (2) secs
      set Shield to [ON v]

```

Our final Script for shield will look like this...

Similarly, import 'fire.png' to Sprites. In its Scripts tab...

```

when backdrop switches to battlefield
  set Fire to [ON v]
  forever
    if [key A v pressed?]
    then
      show
      go to x: -140 y: -71
      set Fire to [OFF v]
      glide (2) secs to x: 170 y: -71
      hide
      wait (2) secs
      set Fire to [ON v]

```

Using GLIDE block, the fire sprite moves smoothly from the present position to (170, -71) for 2 seconds.

Import brick.png and explode.png into the same sprite in Costumes tab.

In brick sprite, add these three blocks separately.



The bricks appear near Ebony

The brick moves left in negative x-axis for 30 steps. Double-click this block to observe what happens.

move (-30) steps

wait (0.1) secs

But moving the brick by 30 steps is not enough. We need to repeat this again and again. REPEAT UNTIL block will repeat the blocks inside it until a condition becomes true i.e., until...



...until the brick touches the doctor OR the shield OR the fireball. So, the brick moves left until it touches any one of the below sprites.

touching doctor_fight? or touching shield? or touching fire?

when backdrop switches to battlefield

wait (3) secs

forever

switch costume to brick

show

go to x: 142 y: -75

Join the conditions together

repeat until [touching doctor_fight? or touching shield? or touching fire?]

move (-30) steps

wait (0.1) secs

if [touching shield? or touching fire?] then

switch costume to explode

wait (1) secs

hide

wait [pick random (1) to (5) secs]

If the wait time for throwing the brick is random, it increases the difficulty of the game. Thus, to PICK a RANDOM value, we use this block.

In Ebony Maw Script, we add the blocks as given in the right. Now, if the value of Ebony's

When the fire touches Ebony, he blinks thrice and life reduces by -1.

Life becomes LESS THAN 1, the player has won the game. If the player wins, all the sprites should be informed to stop their execution. Broadcast is a way of notifying all the sprites about something has happened (here, the notification is 'the player has won.') To broadcast any message, use a BROADCAST block from EVENTS menu. Create your message as 'YouWon'.



Now, we need to check if Ebony touches fire and if his life is less than 1 until the end of the game. So, we add the forever block to our script, and our final Ebony and doctor sprites look like this.



In doctor sprite...

In ebony sprite...

Create a broadcast message 'GameOver'



Add the backdrops 'game-over' and 'won' to your Scratch program.

Now, when the message 'GameOver' or 'YouWon' is received, we do the following in Doctor's Script...



Similarly, we need to hide all the sprites when the message 'YouWon' or 'GameOver' is received. Add the two blocks (given in the right) to all these sprites: ebony_fight, shield, fire and bricks.

Now, there is still one piece missing. When the brick touches fire, it explodes but the fireball still continues to go to Ebony without disappearing. How will fire sprite know that the brick has exploded?



...and add this block in the fire sprite

The final scripts for all your sprites are given in the next page.

You can compare your script with the Scratch file 'game.sb2' in case you have any doubts or visit this [site](#) to play the game. Click the Flag button and start playing the fun and exciting game you've created.

```

when backdrop switches to battlefield
set Shield to ON
forever
  if key s pressed? then
    show
    go to x: -137 y: -69
    set Shield to OFF
    wait 2 secs
    hide
    wait 2 secs
    set Shield to ON

```

```

when I receive YouWon
hide

```

```

when I receive GameOver
hide

```

Shield
sprite

```

when backdrop switches to battlefield
wait 3 secs
forever
  switch costume to brick
  show
  go to x: 142 y: -75
  repeat until touching doctor_right ? or touching shield ? or touching fire ?
    move -30 steps
    wait 0.1 secs
  end
  if touching shield ? or touching fire ? then
    broadcast explode
    switch costume to explode
    wait 1 secs
  end
  hide
  wait pick random 1 to 5 secs

```

```

when I receive YouWon
hide

```

```

when I receive GameOver
hide

```

bricks
sprite

```

when backdrop switches to battlefield
set Fire to ON
forever
  if key a pressed? then
    show
    go to x: -140 y: -71
    set Fire to OFF
    glide 2 secs to x: 170 y: -71
    hide
    wait 2 secs
    set Fire to ON

```

```

when I receive explode
hide

```

```

when I receive YouWon
hide

```

```

when I receive GameOver
hide

```

fire
sprite

```

when backdrop switches to battlefield
show
go to x: 170 y: -75
set Ebony's Life to 3
forever
  if touching fire? then
    change Ebony's Life by -1
    repeat (3)
      hide
      wait 0.1 secs
      show
      wait 2 secs
    end
  end
  if Ebony's Life < 1 then
    broadcast YouWon
end

```

```

when I receive YouWon
hide

```

```

when I receive GameOver
hide

```

ebony_fight
sprite

```

when backdrop switches to battlefield
show
go to x: -202 y: -87
show variable Doctor's Life
show variable Shield
show variable Fire
show variable Ebony's Life
set Doctor's Life to 3
say Press 'a' for attacking and 's' for defending. for 3 secs
forever
  if touching brick? then
    change Doctor's Life by -1
    repeat (3)
      hide
      wait 0.1 secs
      show
      wait 2 secs
    end
  end
  if Doctor's Life < 1 then
    broadcast GameOver
end

```

doctor_fight
sprite

```

when I receive YouWon
hide
switch backdrop to won
hide variable Doctor's Life
hide variable Shield
hide variable Fire
hide variable Ebony's Life
stop all

```

```

when I receive GameOver
hide
switch backdrop to game-over
hide variable Doctor's Life
hide variable Shield
hide variable Fire
hide variable Ebony's Life
stop all

```



You're a top Scratcher and you're ready to create your own games.

Well done! You've made it! You've created your first game.

But you may ask...



Is that all you can do with Raspberry Pi? I mean why did you force me to buy one when I can do all the programming with just a normal computer?

NO!!! RPi is something much more fun than your ordinary computers and most of it is from the needle like row of pins on its side called the General Purpose Input Output (GPIO) pins.

These pins are like ON and OFF switch. If a pin says it's ON, it can turn ON a device connected to it and vice versa. It can be used to turn ON/OFF your LEDs, buzzers etc.

Extension Library

Category
All
Hardware

...select this

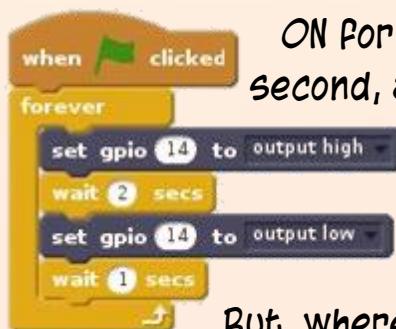


Pi GPIO

Let's start with our first Scratch GPIO program. Open a new Scratch file. Go to scripts tab. Select 'Add an Extension' from MORE BLOCKS menu and...

You'll find some new blocks added in the MORE BLOCKS menu. Now, add the blocks as given below. These blocks will turn your GPIO pin 14

ON for 2 seconds and OFF for 1 second, again and again.

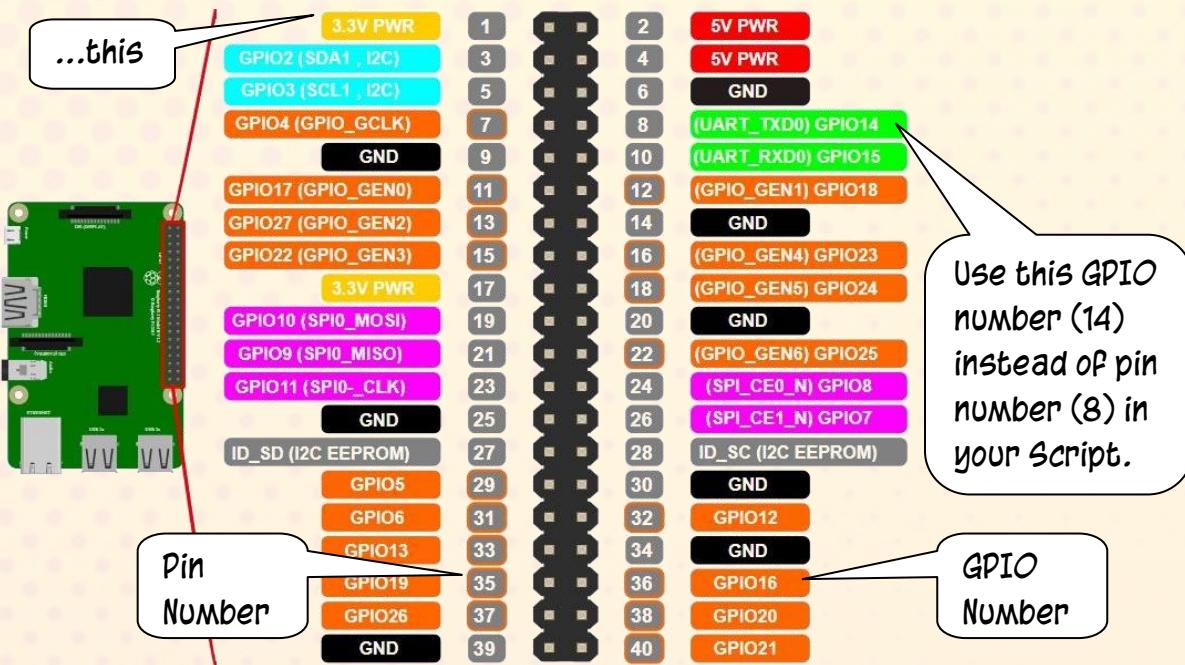


What is there to turn ON/OFF in GPIO 14? Nothing yet. So, we'll connect an LED to this pin to turn it ON/OFF.

But, where is this GPIO pin? Where to connect the LED?



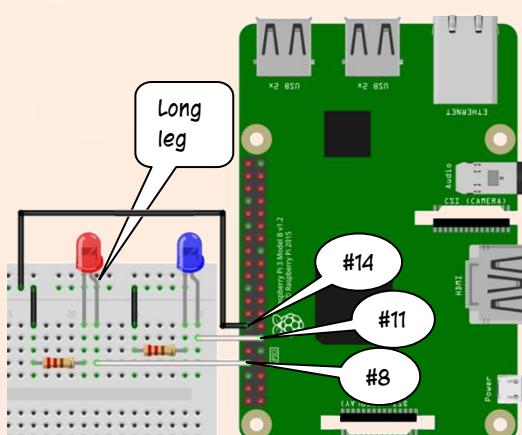
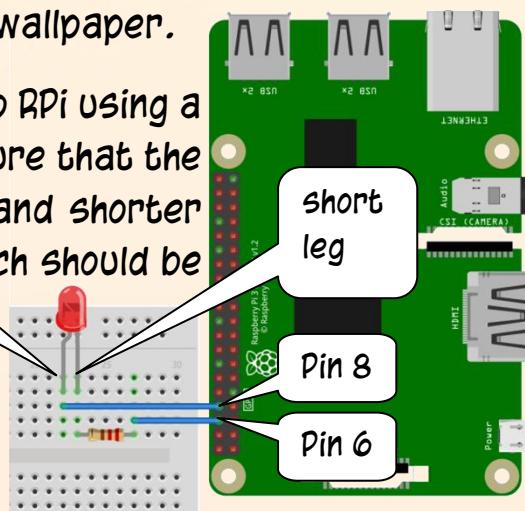
Google 'Raspberry Pi 3 pinout.' You'll find something like...



Stick this on your wall or save it as your wallpaper.

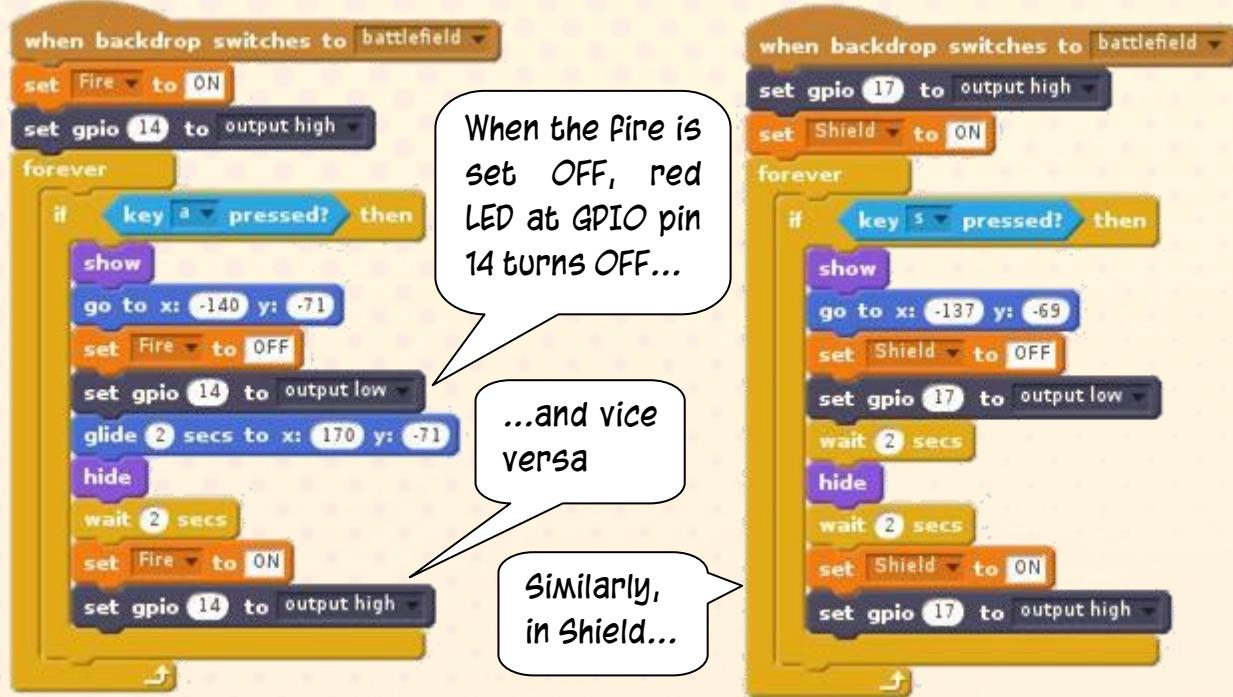
Connect an LED and a 220 ohm resistor to RPi using a breadboard as given in the right. Make sure that the longer leg of LED is connected to pin 8 and shorter leg to the resistor, the other end of which should be connected to pin 6 (GND)

Press Flag. The LED will start blinking.
Amazing! Now, let's see how we can use the LEDs in our game.



We'll use a red LED for the 'Fire' variable and a blue LED for the 'Shield' variable. Connect the LEDs and resistors as given in the left. The numbers given after '#' are the pin numbers. Use 220 ohm resistors for each LED to limit the current through RPi GPIO pin.

Do the following changes in 'fire' and 'shield' sprite.



We can turn OFF the LEDs when the player wins or when the game is over. So, modify 'doctor_fight' sprite as given in the right.

Press flag and you'll see the LEDs glow brightly whenever shield or fire activates...

As a final touch, we'll change the game so that we can play it without even touching the keyboard (like a Kinect.) How do we do that?

PIR sensor... Passive Infra Red sensor...

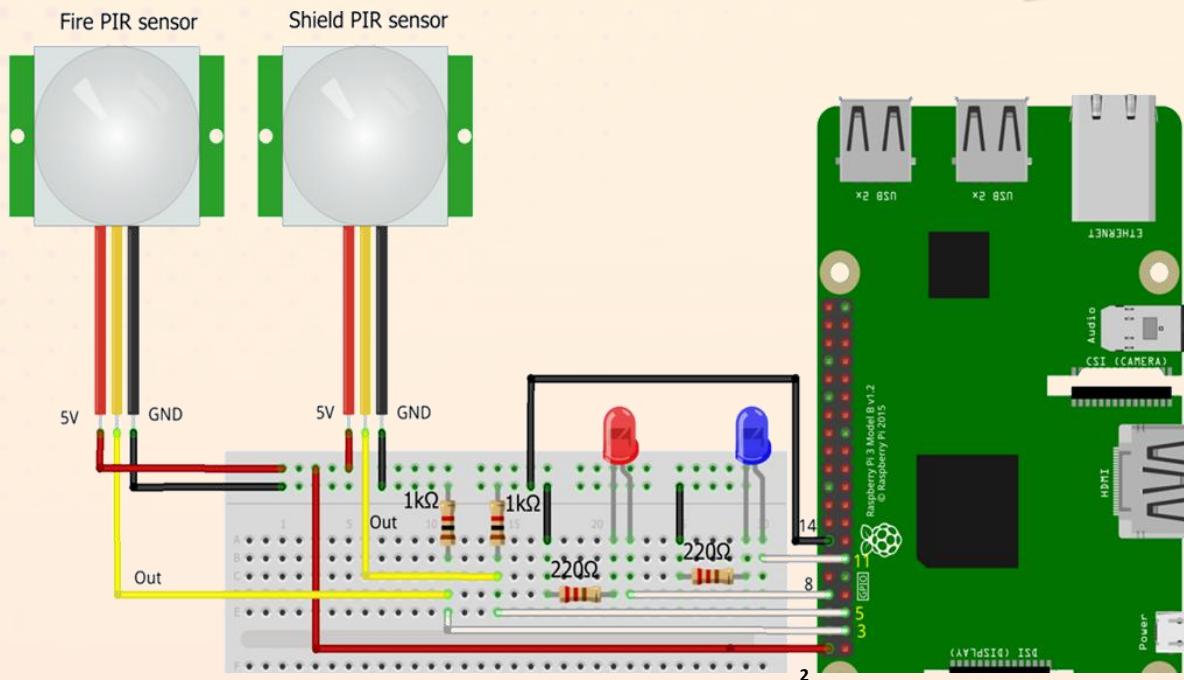




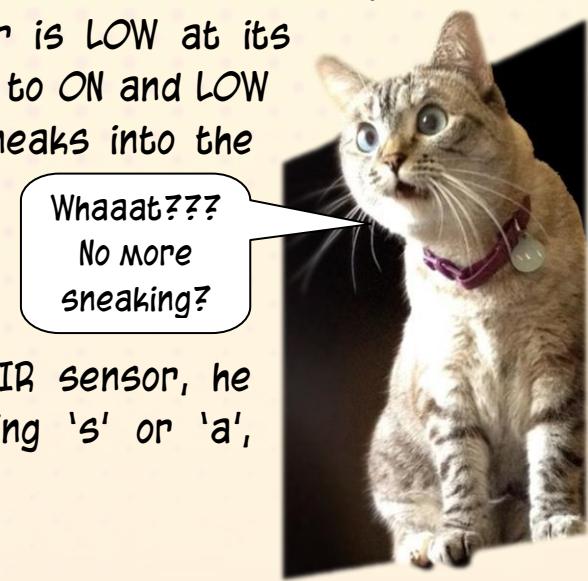
PIR sensors can detect any motion. If an animal, object or anything moves near the sensor, it produces a HIGH output which indicates that motion is detected. If there is no motion, the output of PIR sensor is LOW at its output/OUT pin. HIGH is similar to ON and LOW is to OFF. So, the next time your pet sneaks into the kitchen, you'll know it.

Our game-play is that when the player moves his hands over one PIR sensor, he activates the shield and over another PIR sensor, he activates the fireball; similar to pressing 's' or 'a', respectively on the keyboard.

Connect the sensors like below...



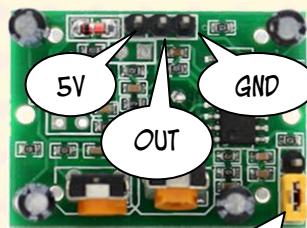
Here, outputs of the sensors are connected to GPIO pins. RPi can read whether its pins are HIGH/LOW i.e., GPIO can act as an input pin. So when PIR senses motion and its output goes HIGH, GPIO can read this as HIGH. Before adding these conditions as blocks in our game...





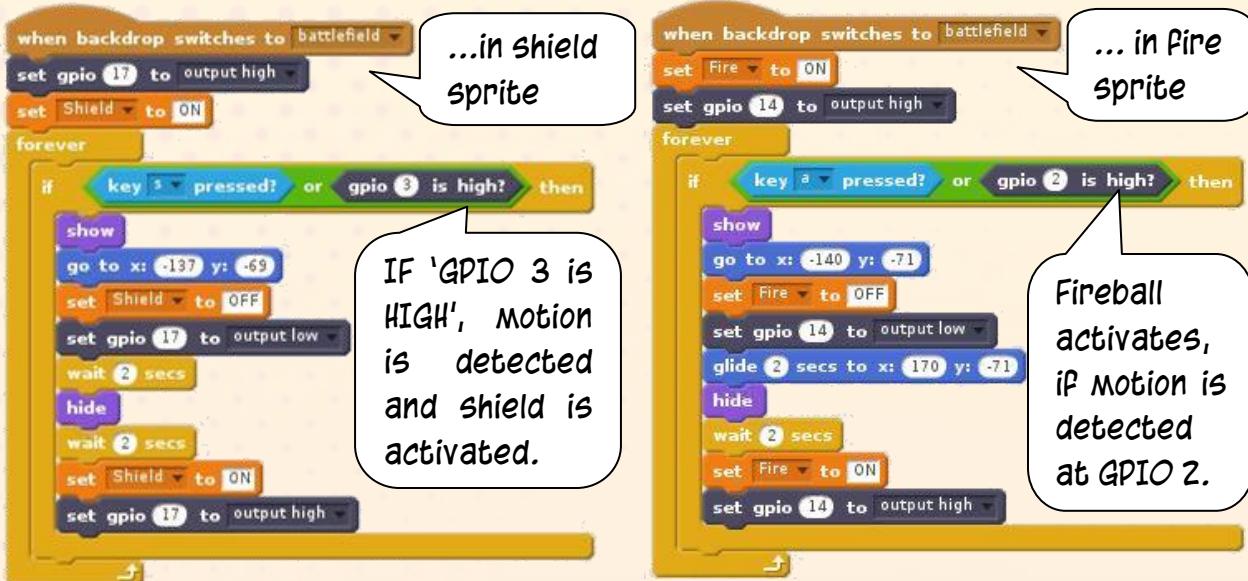
Turn these two in
anti-clockwise
(completely)

...you might need to do some
adjustments in the two yellow
potentiometers of the sensors
using a screwdriver - to
change the distance sensitivity (the
range of detection) and time delay
(wait time of sensor after detection)



Fix this
jumper
here

Now, make the following changes...



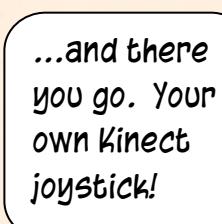
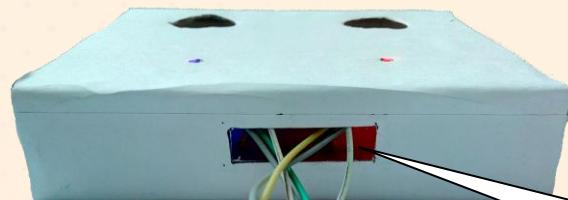
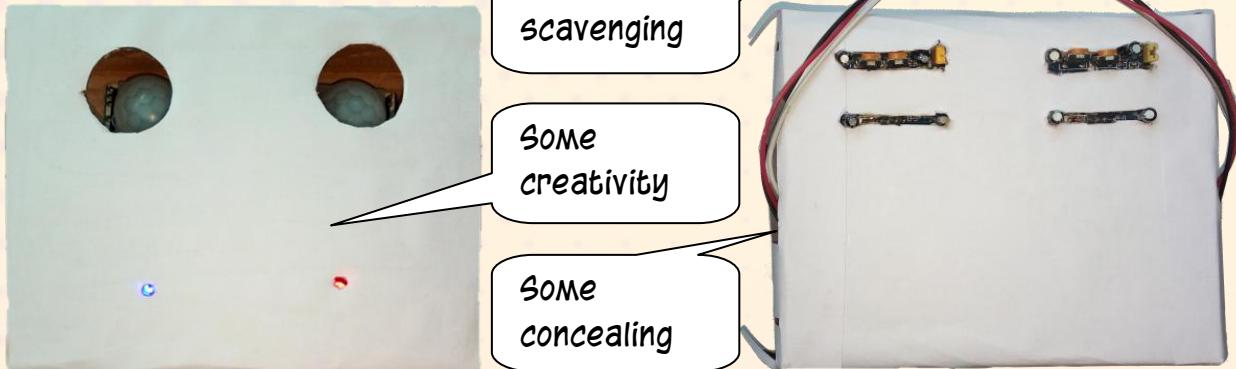
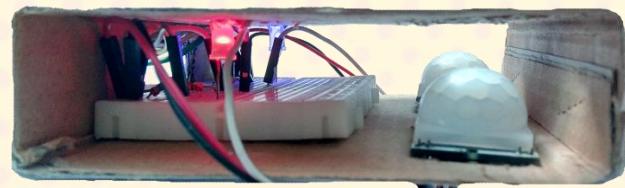
At the beginning of the game, instead of pressing the space bar, we can start the game using shield sensor too. Change the 'backdrop' script as given in the left. Make sure you keep the two PIR sensors as far away as possible else both will be activated if you move your hands. You can cross check your scripts with the 'game_gpio.sb2' file provided. Now, press Flag button. Here we go...

1...2...3...4...5...

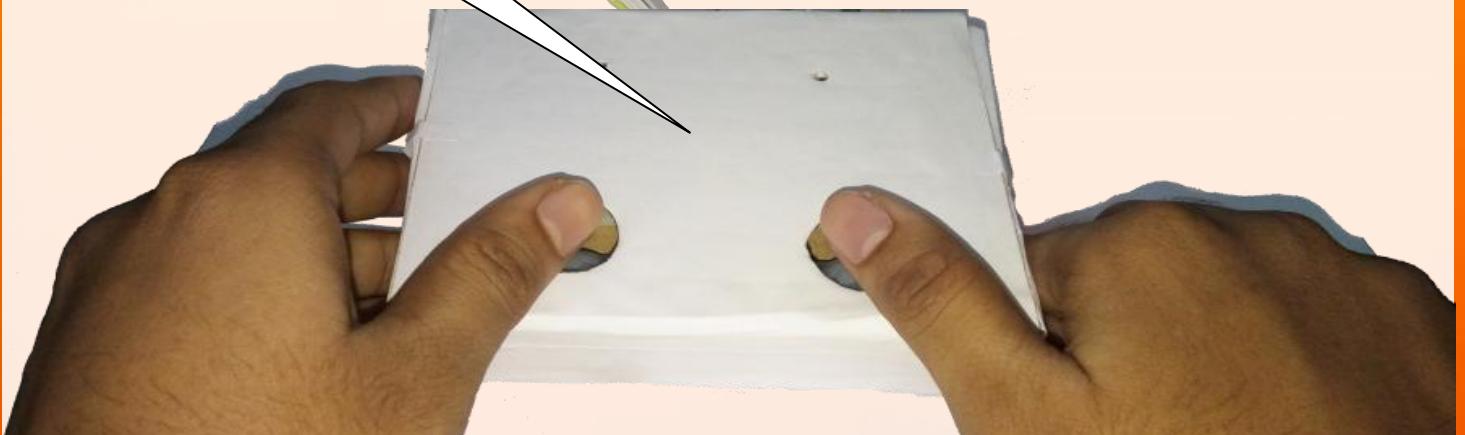
Awesome!!! You did it! All praise the king of Scratch and ruler of RPi GPIO!



If you want to make this game as a joystick, you can do one (like I did ☺) Keep the PIR sensors separated, hide the connections in a box, use your creativity and take it to the next level. A demonstration of the game is in the video file 'game.mp4'



Oh yeah! Some shabbiness too!





THANK YOU! THE TIME STONE IS SAFE. SO, WHAT ARE YOU WAITING FOR? GO AHEAD AND CREATE YOUR OWN GAMES. DO ANYTHING YOU WANT WITH SCRATCH AND RPI. SAVE MORE STONES. ROCK THE WORLD!

