



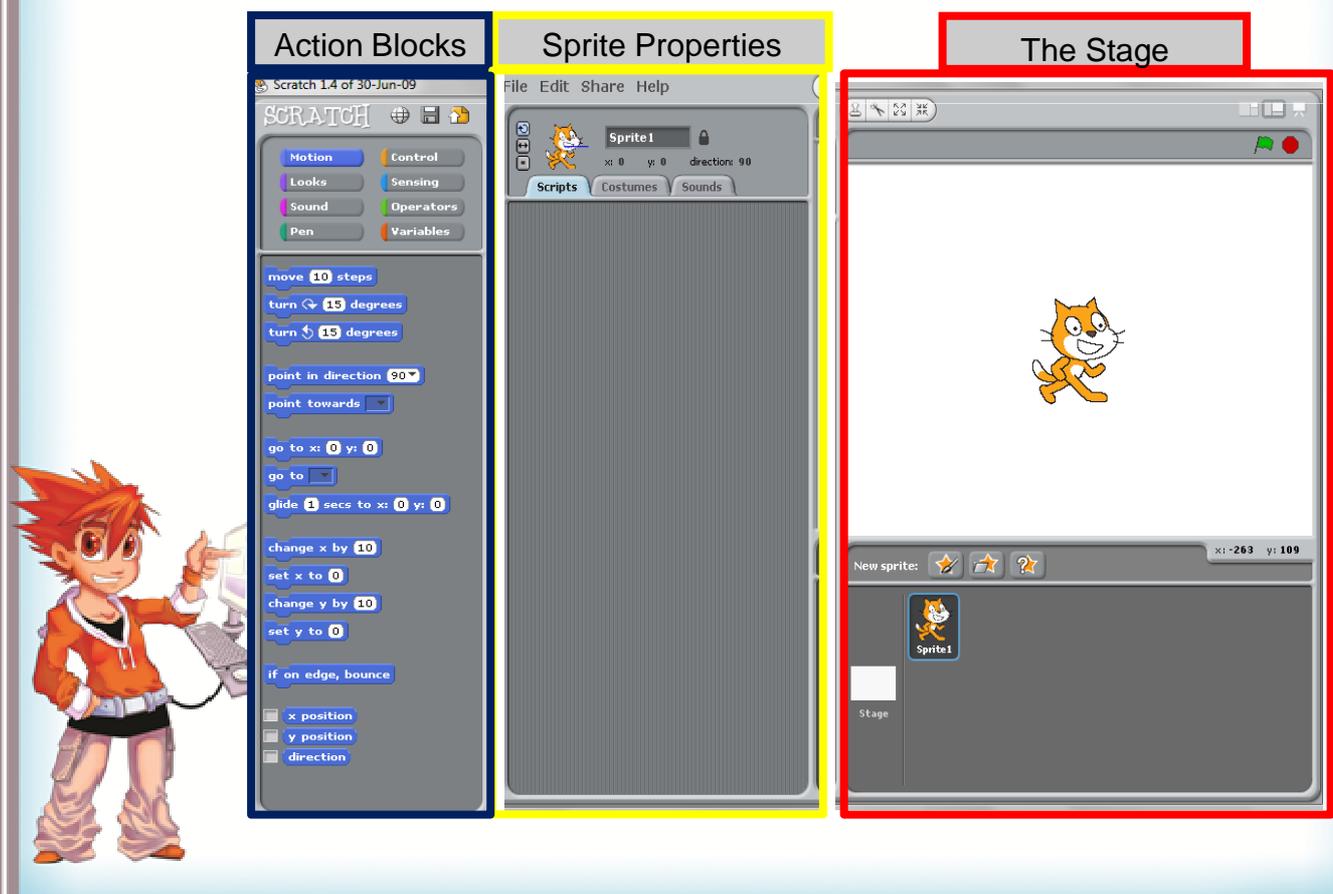
## Valuable Information

The Scratch Interface is divided to three:

1. Stage
2. Sprite/background properties
3. Scratch Action Blocks

Building the game by designing the sprites and the background.

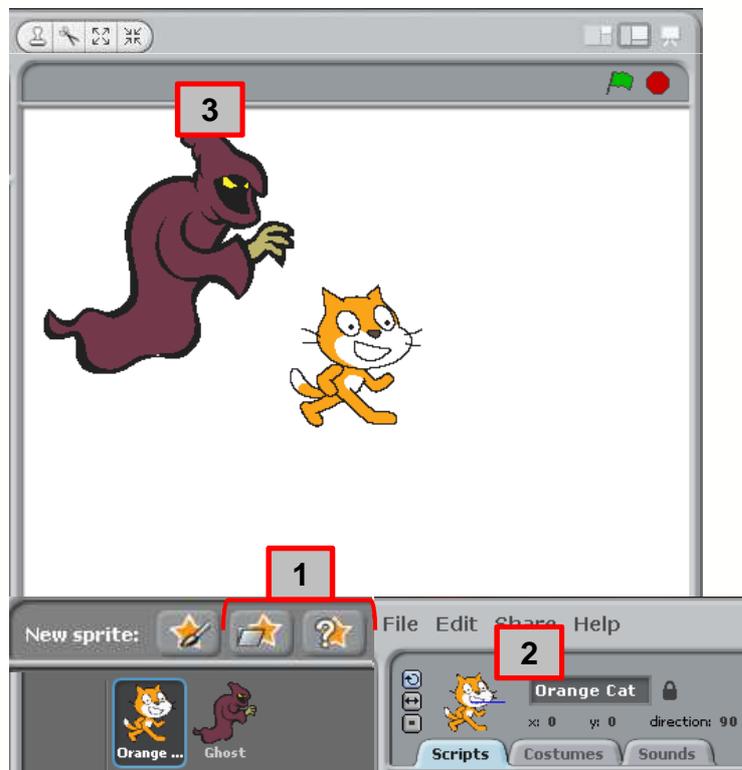
Adding scripts to the sprites and positioning the sprites on stage





### Your Task...

1. Use the buttons under the stage to add a "sprite from file", or "surprise sprite"
2. Give a meaningful name to your sprite in sprite properties (like **Ghost** or **Orange Cat**)
3. Repeat the action with your other sprites



#### Related Cards:

- 1.2-Background Design
- 1.3-Adding costumes
- 2.0-Sprite Motion

#### Prerequisite Cards:

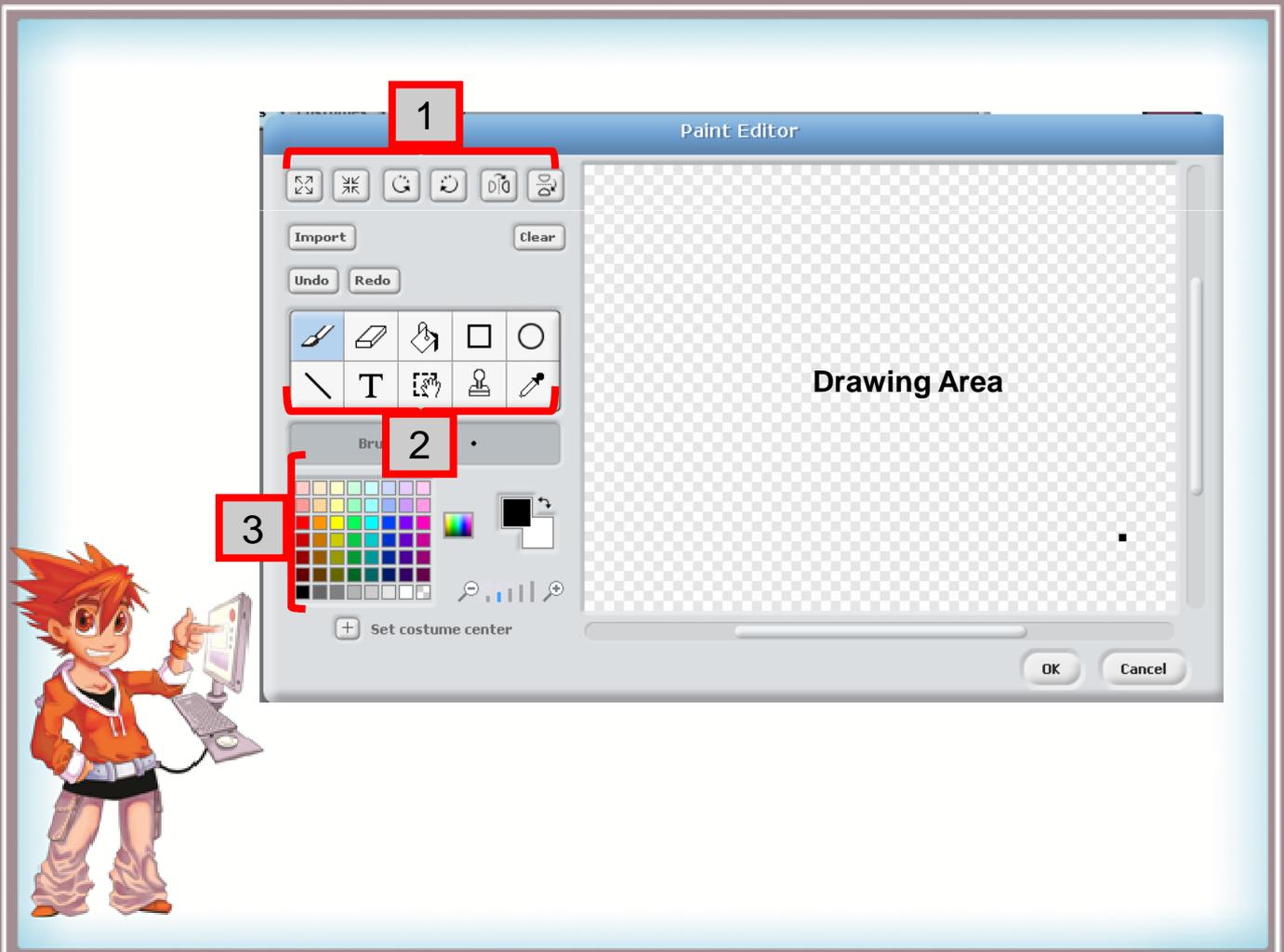




### Valuable Information

The main tools in the Paint Editors are:

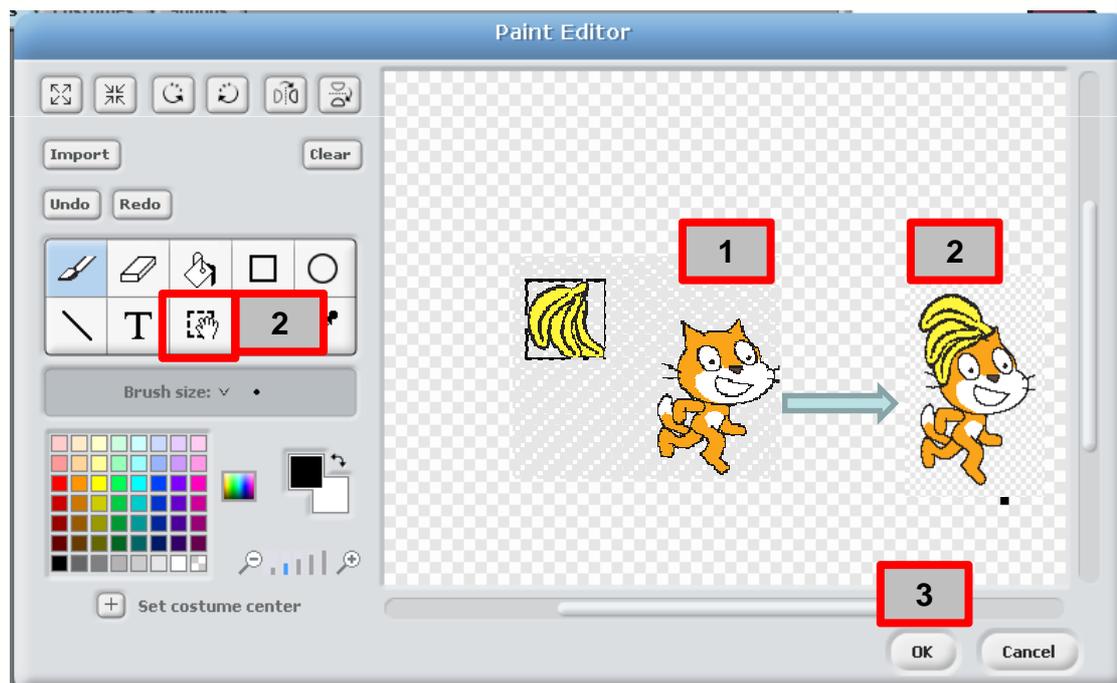
1. Rotate, flip and size tools
2. Sprite editing tools (paintbrush, select, eraser, stamp)
3. Color Palette





## Your Task...Design a new sprite

1. Add two or more costumes to create a new sprite
2. With the Select Tool, attach the costumes together
3. Press OK when finished



**Related Cards:**  
2.0- Move Your Sprite  
1.3- Adding Costumes

**Prerequisite Cards:**



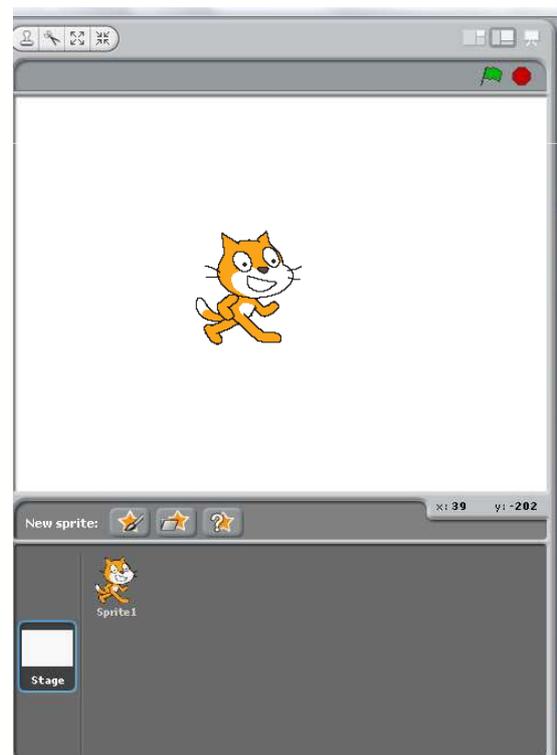
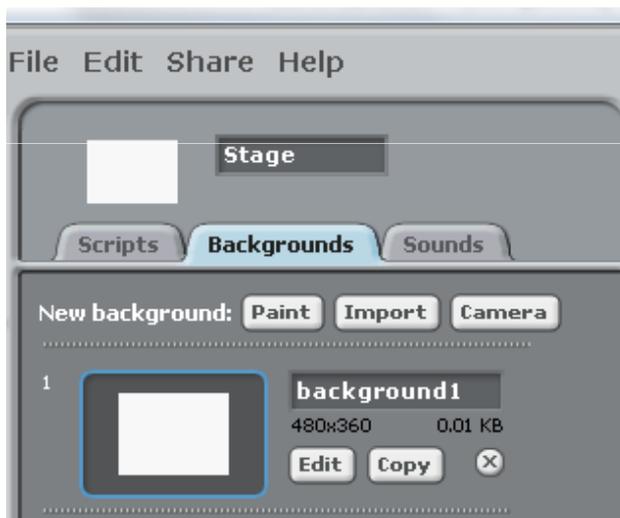


## Valuable Information

Difficulty



- The background is used to design various levels/rooms of a game.
- To add/change background, press the **background** button at the bottom of the stage.
- At the properties area choose **background** tab
- You have three options to add a background:
  1. Paint- in scratch paint editor (card 1.1)
  2. Import- Import an external image from Scratch's library or your own
  3. Camera- use a web cam to capture an image



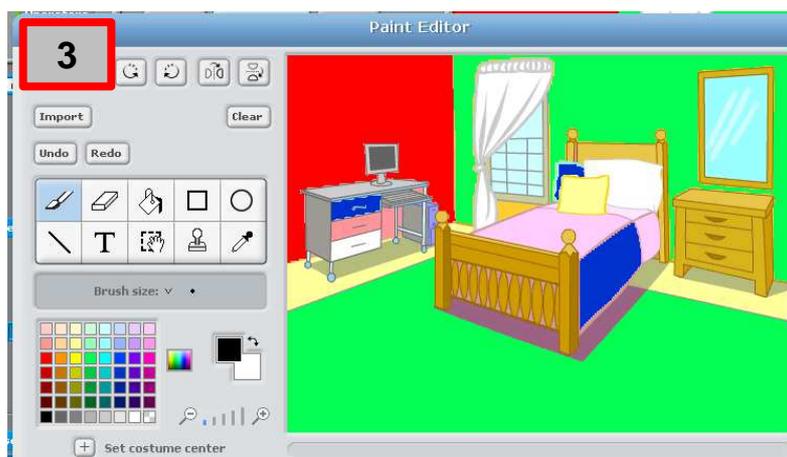
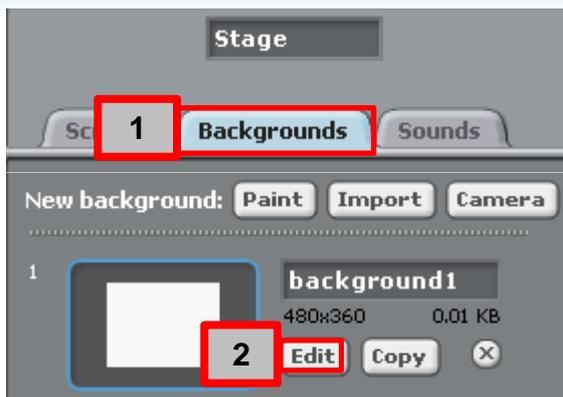


### Your task...import a background

Difficulty



1. Choose **background** tab and then press the **import** button
2. Choose one of Scratch's background image
3. Press **Edit** and recolor your background
4. Press **OK** to finish your editing



**Related Cards:**  
 2.0-Motion With Keys  
 2.9-Collisions-Jump to...

**Prerequisite Cards:**  
 1.1- Paint Editor





## Valuable Information

Difficulty



What are **Scratch Action Blocks**?

**An Action Block** is a graphic block that contains an instruction.

Scratch Action Blocks are divided to groups, that are marked with different colors.

You can add Scratch Action Blocks to any sprite or background.

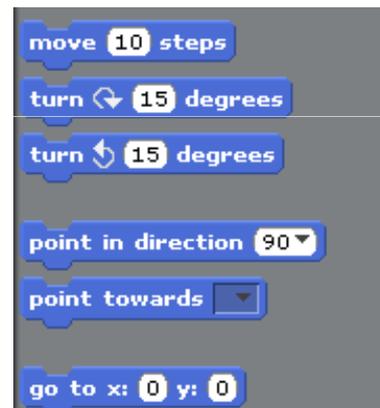
When you put together various Action Blocks, you create a Script.

Scripts are actually programming of the sprites.

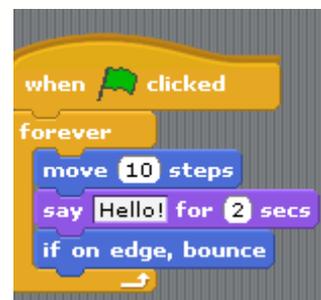
Action Block Palette



Individual Blocks



Blocks In a Script





## Your Task...-Move Your Sprite

Difficulty



In Scratch all the scripts begin with a “hooded” **control Action Block**

Move Your Sprite begin with the “hooded” Action Block

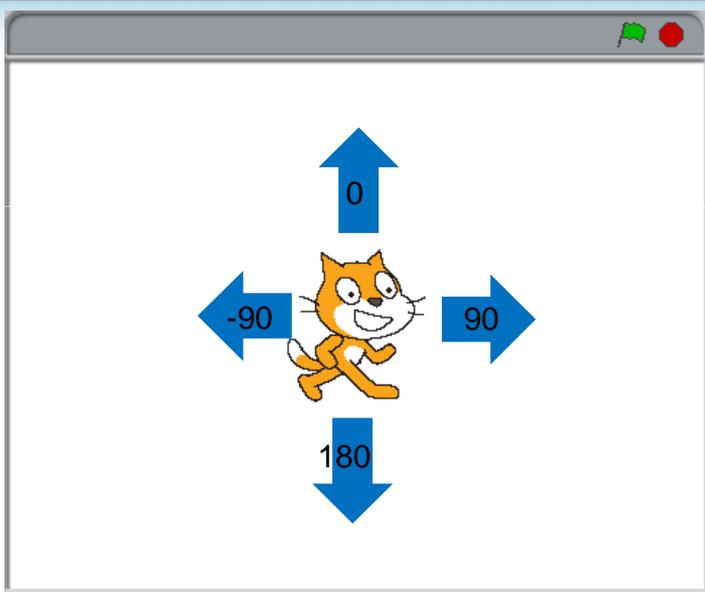
**When (name of the Key) key pressed**

Script order :

When (an arrow key) pressed

Point in direction...(see card 2.1)

Move (set desired number) steps



when **up arrow** key pressed

point in direction **0**

move **10** steps

when **left arrow** key pressed

point in direction **-90**

move **10** steps

when **right arrow** key pressed

point in direction **90**

move **10** steps

when **down arrow** key pressed

point in direction **180**

move **10** steps

### Related Cards:

2.1-Rotation Style  
2.3- Following a sprite  
2.5-Coordination System

### Prerequisite Cards:

1.0- Scratch Interface





## Valuable Information

Difficulty



Sprites in Scratch can rotate in three possible ways:

1. Vertically-(up and down) this is the default setting
2. Horizontally-(Left and right)
3. Fixed – the sprite doesn't change direction





## Your task...where are you going?

Difficulty



Add three sprites. For each sprite, add the following code:

When green flag clicked

1. In a **forever** loop block (card 2.2) add :
  - a. **Move... steps** (set desired steps)
  - b. if on edge, bounce
2. Change the rotation style of each sprite
3. Copy a script by dragging and dropping it to the images under the stage

File Edit Share Help



### Related Cards:

2.2-Motion Animation  
2.3- Following a sprite  
2.4-Motion Path

### Prerequisite Cards:

1.0- Scratch Interface  
2.0-Motion with Keys





### Valuable Information

Difficulty



A loop is an action that repeats itself numerous times instead of repeating

Code writing manually. In Scratch, we have four types of Motion Animation:

1. **Forever**-the script runs as long as the game runs- for example: sprite animation
2. **Repeat..**-the script repeat itself up to the number in the loop-for example- timer
3. **Forever if...** the script runs **only** when the condition is fulfilled (see card 2.7)
4. **Repeat Until** -The script stops when condition fulfilled- for example :  
Number of apples

### Common Motion Animation Patterns

1 when clicked  
forever  
next costume  
wait 0.1 secs

2 when clicked  
repeat 10  
wait 1 secs  
change time by -1

3 when clicked  
forever if time > 0  
next costume  
wait 0.1 secs

4 when clicked  
repeat until time = 0  
wait 1 secs  
change time by -1





Difficulty



## Your Task...-Motion Animation

1. Add a sprite with two costumes- [www.clker.com](http://www.clker.com) has great royalty free drawing
2. In the paint editor, change the sprites a little to imitate movement  
(in this sample the button of the camera was cut to half)
3. Add a Loop identical to the screen capture.
4. To prevent frantic change in animation, add the Action block **wait 0.1 seconds**

The image displays the Scratch interface for creating a motion animation. On the left, the code editor shows a script starting with a 'when green flag clicked' event block, followed by a 'forever' loop containing a 'next costume' block and a 'wait 0.1 secs' block. On the right, the sprite editor shows two costumes for a green camera-like sprite: 'Unpressed' (174x184, 22 KB) and 'pressed' (174x184, 21 KB). Below the editor, two green camera-like sprites are shown with red arrows indicating a transition between them.



**Related Cards:**  
 2.9-Levels  
 2.4-Motion with Mouse

**Prerequisite Cards:**  
 1.0- Scratch Interface  
 1.2-Paint Editor  
 2.1-Rotation Style





### Valuable Information

Difficulty



Following a sprite gives the programmer the ability to create sprites that automatically follows another sprite.

Chasers, bullets and ball games are some of the options.

Automatically following the mouse cursor, enables a precise control of the movement of the player's sprite.

```
when clicked
  forever
    point towards Basketball
    move 2 steps
```

```
when clicked
  forever
    point towards mouse-pointer
    move 2 steps
```



### Your task... the chase

Difficulty



1. Program sprite One to move with the keyboard (card 2.0)
2. Add another a chaser sprite (no. 2) and program it to follow sprite One
3. Add a third sprite- Finish Line to which your sprite needs to arrive without being captured.

1

```

when up arrow key pressed
  point in direction 0
  move 10 steps

when left arrow key pressed
  point in direction -90
  move 10 steps

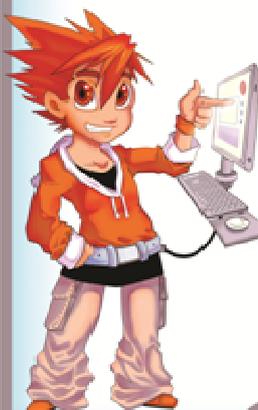
when right arrow key pressed
  point in direction 90
  move 10 steps

when down arrow key pressed
  point in direction 180
  move 10 steps
    
```

2

```

when green flag clicked
  forever loop
    point towards Cat
    move 2 steps
    
```



**Related Cards:**  
 2.0-Motion with keys  
 2.1-Rotation Style  
 2.2-Motion Animation

**Prerequisite Cards:**  
 1.2-Paint Editor  
 2.7-Collisions



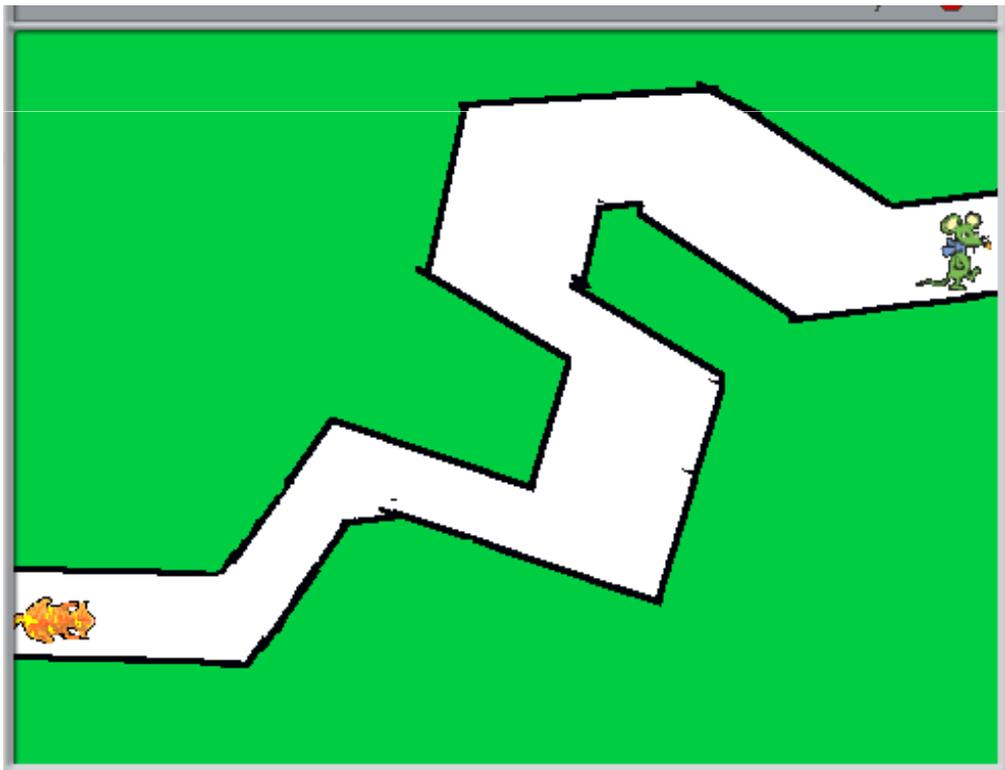


### Valuable Information

So far you've learnt to:

1. Add sprites and backgrounds and edit them graphically
2. The ways to move your sprite and three rotation styles

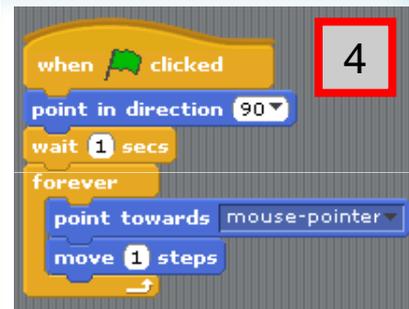
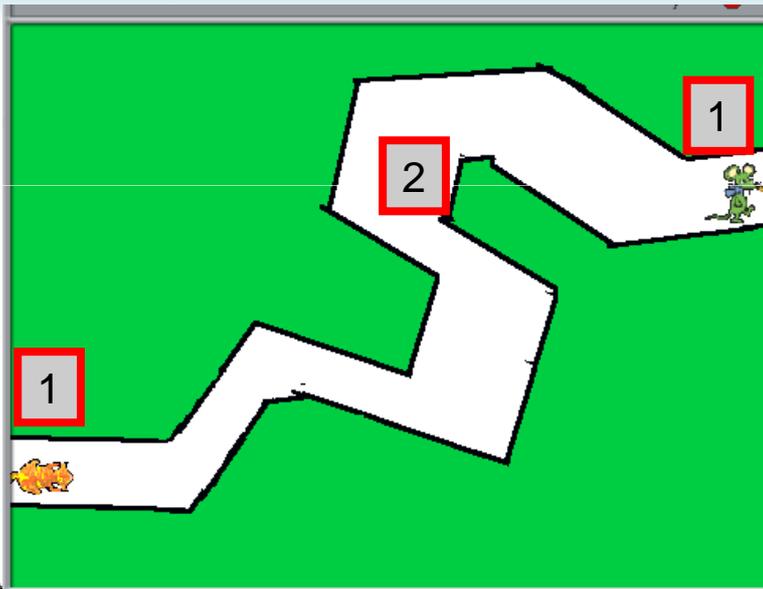
In this card you will practice all these topics combined.





### Your task... Motion Path

1. Add two sprites (card 1.0)
2. Paint a path as a background in the paint editor (card 1.1)
3. Option A: program one sprite to move with the keyboard (card 2.0)
4. Option B: Program the sprite to follow the mouse cursor (card 2.3)
  - add to the script two blocks- **point in direction 90**, **wait 1 sec** (see screen shot)
5. The sprites don't stop at the walls of the path, this can be changed by referring to card 2.7-collisions



#### Related Cards:

2.6-Sensing  
2.7-Collisions  
3.0-Variables

#### Prerequisite Cards:

2.0-Motion with Keys  
2.1-Rotation Style  
2.3 Motion with Mouse





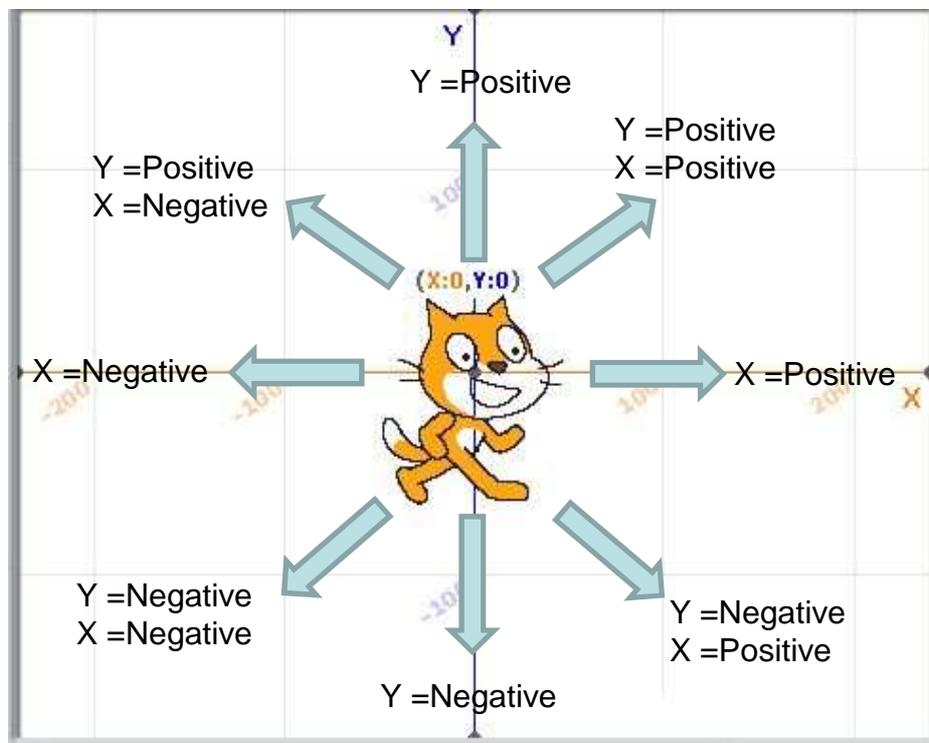
### Valuable Information

Difficulty



With Scratch coordinate system, there's accurate control of sprites' motion.

1. The Stage is divided by two hidden lines (axis)
2. A horizontal line (named X), a Vertical line (named Y)
3. Each point on the axis has a value
4. At the center of the screen  $X=0$   $Y=0$
5. When a sprite moves, it's X & Y values changes according to movement
6. The sprite's X & Y values can be found in the sprite properties





## Your Task...Diagonal motion

Difficulty



Program a sprite that will move diagonally when the up and down arrows are pressed .

1. When moving right the X value increase When moving Left X value decreases
2. When moving up Y value increases. When moving down Y value decreases
3. Diagonal motion is changing values of X and Y simultaneously

In the game below the cat moves diagonally to catch the mouse.

Add the script below to move your cat diagonally

The image shows four Scratch code blocks for key events and a game scene. The code blocks are:

- when up arrow key pressed:** point in direction 0, change x by 3, change y by 3
- when right arrow key pressed:** point in direction 90, change x by 3
- when down arrow key pressed:** point in direction 180, change x by 3, change y by -3
- when left arrow key pressed:** point in direction -90, change x by -3

The game scene below shows a green field with a white path. A cat is at the bottom left, and a mouse is at the top right. The path is a zig-zag shape.



### Related Cards:

2.6-Sensing  
3.0-Variables  
3.1-Counter Variable

### Prerequisite Cards:

2.0- Keys Motion Programming  
2.3- Following a sprite





## Valuable Information

Difficulty



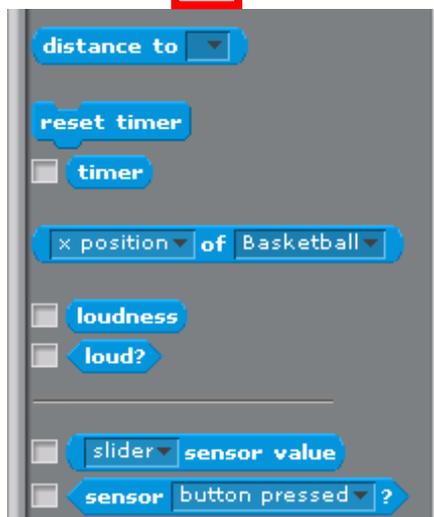
Sensors are action blocks that checks on various conditions and data.

A sensor block is not a standalone block. It is inserted into other blocks.

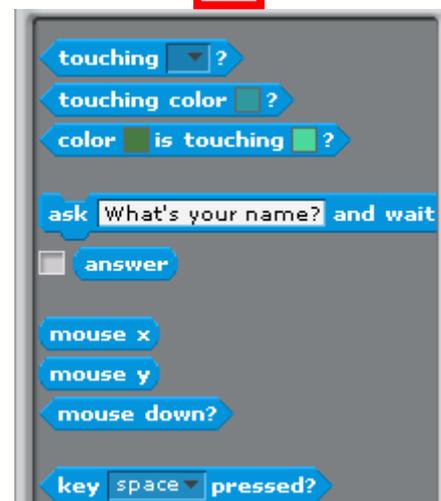
There are two types of sensor blocks:

1. **Value sensor**- these blocks test a numerical value- X&Y value of a sprite, distances, volume level etc.
2. **Test sensor**- These blocks test if various conditions happen- mouse button pressed, color touches another color, sprite touches another sprite etc.

1



2



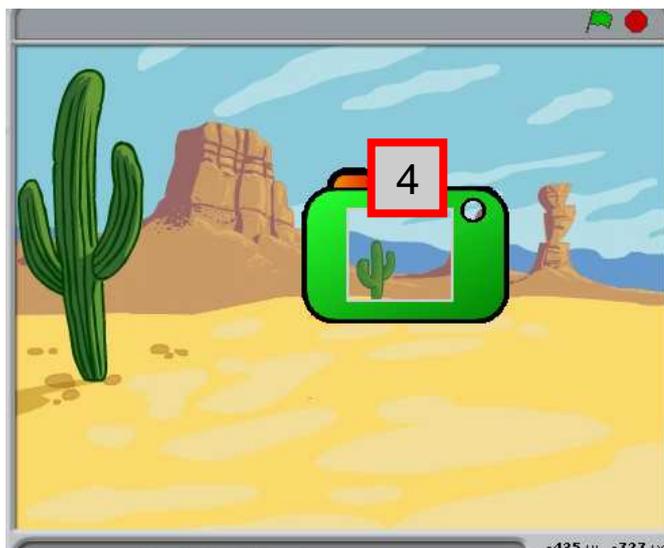


### Your task... photography tour

Difficulty



1. Add a sprite of a camera (royalty free site for sprites [www.clker.com](http://www.clker.com))
2. Add the script that is shown in the screen shot
3. Instead of the original value, add a test blocks- **Mouse X & Mouse Y**
4. Repaint part to transparent color
5. Now the camera will follow the mouse cursor



**Related Cards:**

- 3.0-Variables
- 3.1-Counter Variable

**Prerequisite Cards:**

- 1.2-Paint Editor
- 2.3- Following a sprite
- 2.5-Coordinate System





### Valuable Information

Difficulty



Collisions are contact between sprites or between a sprite and certain color

You check collisions with conditional sentences (If **Car** touch **Tank...**)

Conditional sentences (also known as IF sentences) are under **control tab**.

Collision are used for:

1. Points
2. Collecting and hiding objects
3. Changing levels and setting location of sprites

Collision Script Structure

```

when clicked
  forever
    if touching ?
  
```

Collision Script - Points

```

when clicked
  forever
    if touching Tank ?
      change Fuel by 30
  
```

Collision Script - Collect & Hide

```

when clicked
  forever
    if touching car ?
      hide
  
```

Collision Script - Location

```

when clicked
  forever
    if touching edge ?
      go to x: -240 y: 80
  
```





## Your Task...Refill

Difficulty



1. Add two sprites- one of a car and one of a fuel tank (a royalty free site for costumes (www.clker.com)
2. Program the car to follow the mouse pointer or to move with keys
3. Add the script under the title "fuel tank" to the appropriate sprite
4. Add the script under the title "car" to the appropriate sprite

### Car

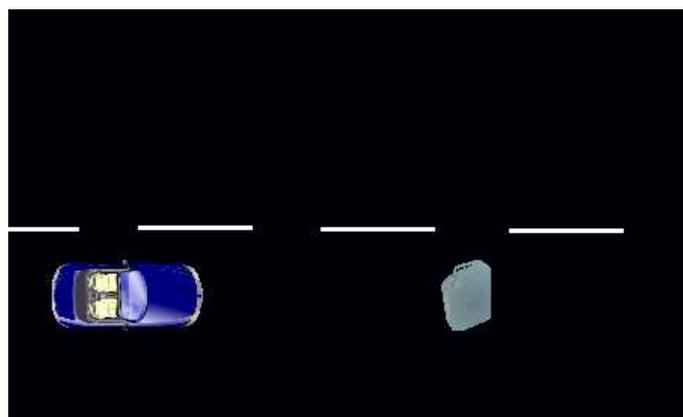
```

when clicked
  forever
    if touching edge?
      go to x: -240 y: 80
  
```

### Fuel Tank

```

when clicked
  forever
    if touching car?
      hide
      wait 1 secs
      show
  
```



**Related Cards:**  
 2.9-Location setting  
 3.0-Variables

**Prerequisite Cards:**  
 2.2- Motion Animation  
 2.3- Following a sprite  
 2.5- Coordination System





Difficulty



### Valuable Information

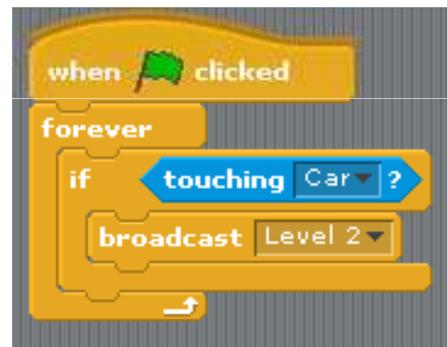


- Broadcast is an order that is sent from one sprite to another (others) sprite.
- Using broadcast controls many events at one time – repositioning sprites, hide/show sprites etc.
- The programmer names the broadcast. There is no limit to the number of broadcasts.
- The block **broadcast** is part of a script.
- The block **when I receive...** is an starting block.

#### Send and receive broadcast blocks



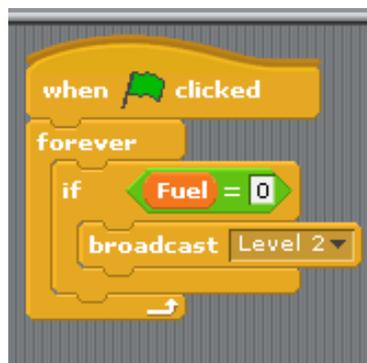
#### Broadcast by collision script



#### Broadcast by click on sprite script



#### Broadcast by value check script



#### Receiving broadcast scripts





### Your task...RC car

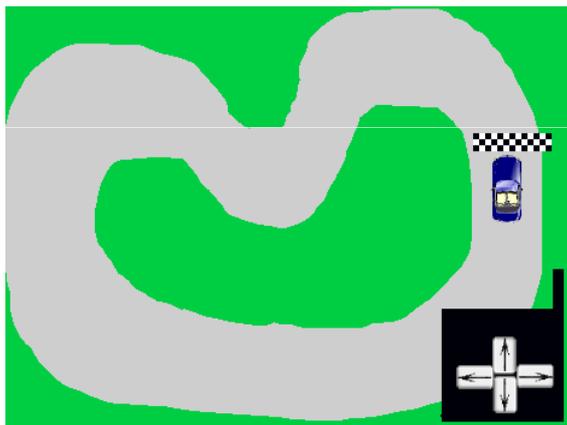
Difficulty



In this task, you will program and drive an RC car.

1. Create four sprites of arrows from the **Things** folder
2. Add a sprite of a car
3. Add the racing track background (You can use **Marble Racer** background)
4. Program the arrows and the car according to the screen shots

#### Background and Sprites



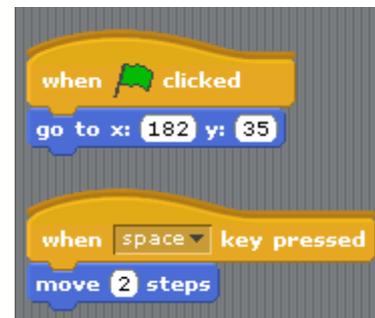
#### Remote Control Scripts



#### Car -Directions Scripts



#### Car -Motion and Initiation Scripts



#### Related Cards:

- 2.9-Location setting
- 3.0-Variables

#### Prerequisite Cards:

- 2.2- Motion Animation
- 2.3- Following a sprite
- 2.5- Coordination System





### Valuable Information

Difficulty



- 
- 
- 
- 
- 
- 
- 

- A lot of sprites change form and location during the course of a game.
- Reset Scripts are used to reset the sprites to their initial status.
- Reset scripts set the hide/show status, point direction, X & Y location and setting points and lives to initial values.
- During a game, the location of a sprite is usually set in a collision script with another sprite or the edge.

#### New Level Position and show reset scripts

```

when I receive Level2
  go to x: 52 y: 87
  show
    
```

#### New Game Reset Scripts

```

when clicked
  switch to costume Square1
    
```

```

when clicked
  hide
  point in direction 90
    
```

#### Jump to after a collision Scripts

```

when clicked
  forever
    if touching edge ?
      go to x: x position of block7 y: y position of block7
    
```





## Your Task...Rainfall Recycle

Difficulty



In this task, your job is to collect the water drops from the raincloud.

1. Add three sprites- a cloud, a drop and a bucket
2. Program the bucket to move left and right with the arrows
3. To program the cloud, copy the code from the screen shot
4. In the programming of the rain drop, add the block x position / Y position instead of the numbers in the **jump to** block.
5. Add the **broadcast** and **when I receive broadcast** blocks to the cloud and the drop

### Cloud Scripts

```

when clicked
broadcast Rain
forever
  move 3 steps
  if on edge, bounce
  
```

### Drop Scripts

```

when clicked
go to x: x position of Cloud y: y position of Cloud

when I receive Rain
repeat until touching bucket
  move 3 steps
  show
go to x: x position of Cloud y: y position of Cloud
broadcast Rain

when clicked
forever
  if touching edge
    go to x: x position of Cloud y: y position of Cloud
  
```



**Related Cards:**  
 2.5- Coordination System  
 3.0-Variables

**Prerequisite Cards:**  
 2.2- Motion Animation  
 2.8-Collisions-Broadcasting





### Valuable Information

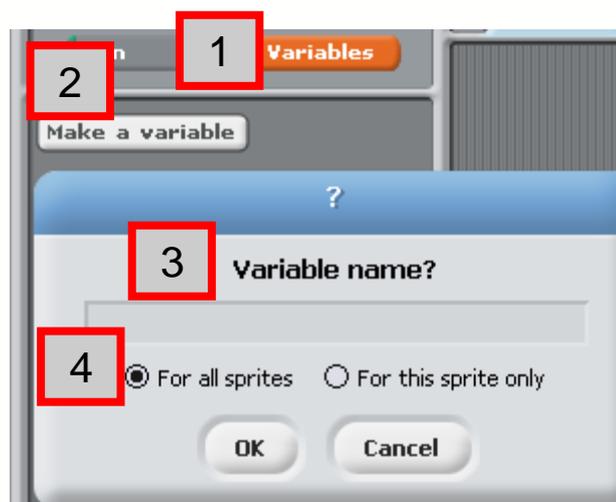
Difficulty



- A variable is like a container whose content constantly change.  
The variable is given a name by the programmer when created.  
Variables are used for counting points, or lives of a player, or to remember an action (Counter Variable). There are two types of variables:
1. **Global Variable-** a variable that all sprites can affect
  2. **Local Variable-** a variable that is affected only by sprite it was created in

### How to Add a Variable to Scratch

1. Choose Variables Tab
2. Press on **Make a variable** button
3. Give a meaningful name to your Variable (speed, points, lives etc.)
4. Choose between Global (**For all sprites**) or local variable (**For this sprite only**)





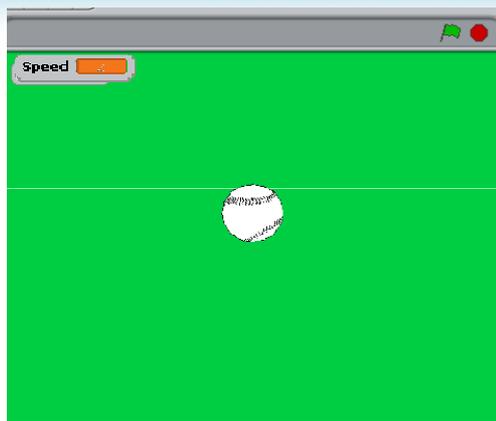
Difficulty



### Your Task...-Touch Ball

In this game you have one sprite- a ball that each time the player touch  
The ball, its motion speed increases.

1. Create a new sprite- give it the name "ball"
2. Create a variable , set it as global (**For all sprites**)
3. Add the scripts in the picture below to your ball



when Ball clicked  
change Speed by 1

when clicked  
set Speed to 1

when clicked  
forever  
move Speed steps  
if on edge, bounce

when clicked  
forever  
turn pick random 1 to 90 degrees  
wait pick random 1 to 3 secs



**Related Cards:**

- 2.5- Coordination System
- 3.1- Counter Variable

**Prerequisite Cards:**

- 2.2- Motion Animation



### Valuable information

Difficulty



Counters are variables that are used for events such as- points change, shooting and timer. There are two types of counters:

1. **Automatic Counter** -this counter value change in a loop (like the variable **time** that counts seconds)
2. **Event Counter**-the loop of this counter is triggered after a collision or a keyboard / mouse input

In counter scripts before the loop insert **Set... to...** block to reset the variable.

In **event counter** script use **wait until** block to repeat the loop one at a time.

#### Automatic Counter

```

when clicked
set time to 30
repeat 30
wait 1 secs
change time by -1
  
```

#### Key/Mouse pressed Event Counter

```

when clicked
set Bullets to 30
forever
wait until mouse down?
change Bullets by -1
wait 0.3 secs
  
```

#### Collision Event Counter

```

when clicked
set Balloons to 0
forever
wait until touching GunSight? and mouse down?
switch to costume Blown_Balloon
wait 0.3 secs
hide
change Balloons by 1
  
```



### Your task...Shoot the Balloons

Difficulty



In this game shoot as many balloons in 30 seconds.

1. Paint / import two sprites- Balloon and gun sight
  2. Program the gun sight to follow the mouse cursor
  3. Program the balloons to move bottom-up and, if not shot, to disappear when touching the edge
  4. Add the variables-**bullets, balloons and time**, use screen shots for scripts
- \*In the gun balloon collision, there are two conditions- key press and collision to get the points for shooting the balloon.

**Time variable-  
insert in Background**

```

when clicked
  set time to 30
  repeat 30
    wait 1 secs
    change time by -1
    
```

**Bullets variables-  
Insert in Gun Sight sprite**

```

when clicked
  set Bullets to 30
  forever
    wait until mouse down?
    change Bullets by -1
    wait 0.3 secs
    
```

**Balloons variable-  
Insert in Balloon sprite**

```

when clicked
  set Balloons to 0
  forever
    wait until touching GunSight? and mouse down?
    switch to costume Blown_Balloon
    wait 0.3 secs
    hide
    change Balloons by 1
    
```

**Balloon sprite-  
Collision with Edge script**

```

when clicked
  forever
    wait until touching edge?
    hide
    go to x: pick random -150 to 150 y: -50
    
```

**Gun sight sprite-  
Reset and motion script**

```

when clicked
  go to front
  forever
    go to x: mouse x y: mouse y
    
```



**Related Cards:**

- 2.5- Coordination System
- 3.0- Variables

**Prerequisite Cards:**

- 2.1- Rotation Style
- 2.2- Motion animation
- 2.6-Sensing





### Valuable information

Difficulty

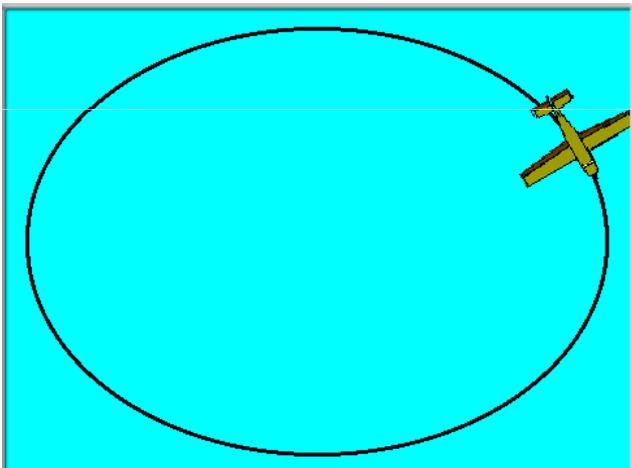


Math operators are used for various calculations.

There are five types of operators:

1. Basic Calculations-round block- multiply, division, add, subtract
2. Comparisons- bigger then, smaller then, equal
3. Advanced calculations
4. Additional condition blocks
5. Random numbers (one block- range of random numbers like the lottery)

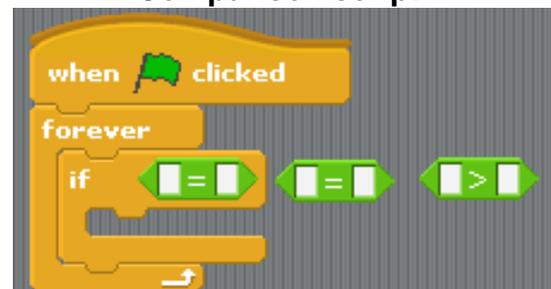
#### Your task-Fly the plane See other side for details



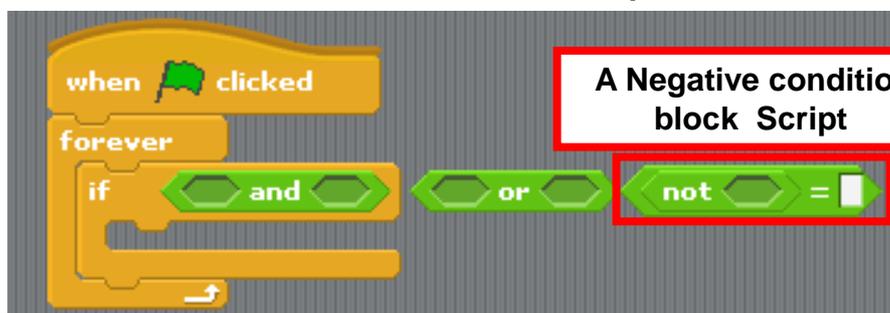
#### Random number script



#### Comparison script



#### Additional condition blocks script



A Negative condition block Script



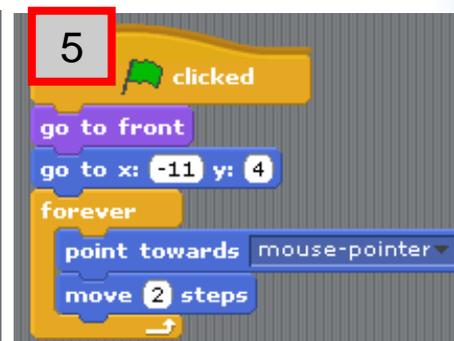
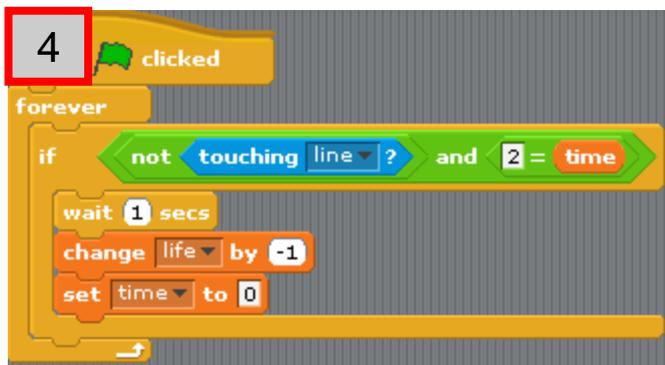
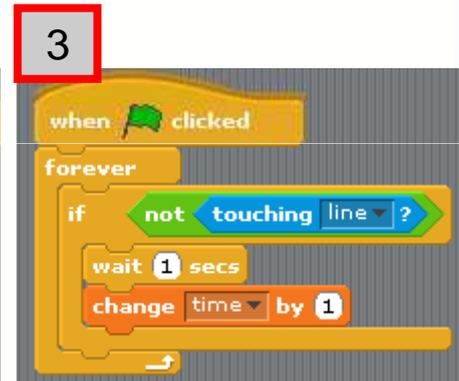
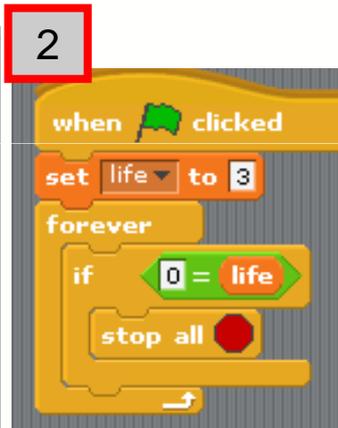
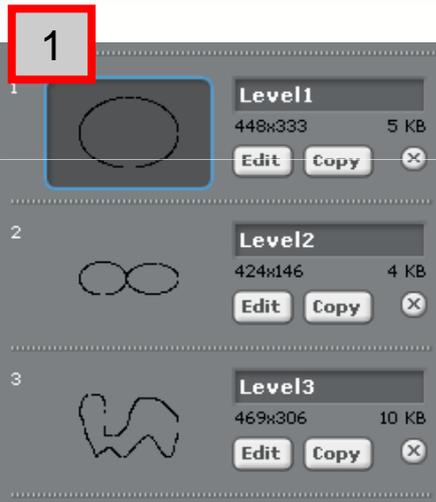


### Your task...Fly the Plane

Difficulty



1. Make a sprite that has various shapes on it (See Screenshot on the other side)
  2. **Condition "if line not touched"**-starts the timer when the line isn't touched.
  3. **Condition "if life = 0"**- stops all sprites- end the game
  4. **Doubled Condition** -this controls the attempts after two seconds out of line the player lose one life
  5. The block **Go to front** is used to make sure the plane is above the line
- Add a script that gets you to the next level



**Related Cards:**  
 2.5- Coordination System  
 2.8-Collisions- Broadcasting

**Prerequisite Cards:**  
 2.2- Motion animation  
 2.6-Sensing  
 3.0- Variables





### Valuable information

Difficulty



Scrolling is a method to create a genre of games known as platform games such as Mario bro.

In these games the background is constantly changing and creates an illusion of movement. Unlike other games, in this game, the background is actually a series of sprites and most of the work is in designing them to look like continuous background. For our purposes, we use the file called **scrollDemo** in the **Games** folder in Scratch.

#### Player scripts- Motion changes value of variable

```

when clicked
  set scrollX to 0

when clicked
  forever if key left arrow pressed?
    point in direction -90
    change scrollX by 5

when clicked
  forever if key right arrow pressed?
    point in direction 90
    change scrollX by -5
    
```

#### Background sprites scripts- Position set by changing values in The equation

```

when clicked
  forever
    set x to scrollX + 480 * 0
    
```

```

when clicked
  forever
    set x to scrollX + 480 * 1
    
```





## Your task...Super Mario

Difficulty

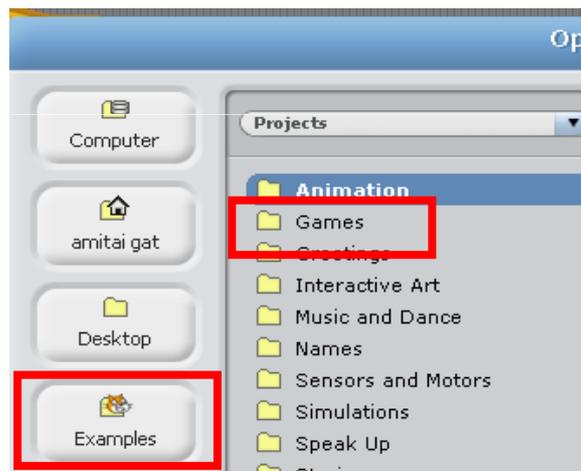


1. Open ScrollDemo from Examples>Games
2. Change the Sprite **Player** to your own sprite
3. Change the sprites named **TerrainX** to another name, keep the numbers
4. Create new background sprites by duplicating the existing sprites
5. In all the existing Sprites the location formula is Scrollx+480\*1,2,3
6. Change the single digit in your new sprite to a negative number, the Sprite is added to the left of the screen

### The New Sprite



### File > Open > Examples > Games



### A negative add sprite left of screen



### Add sprite Right of screen



### Related Cards:

- 1.1- Paint Editor
- 2.7-Collisions
- 3.1- Counters

### Prerequisite Cards:

- 2.5-Coordination System
- 3.0-Variables
- 3.2-Math Operators





## Valuable information

Difficulty



- An array is a variable that stores a large number of values
- The values in an array are called items. An Array can hold numerical or textual items.
- During a game more items can be added to an array and used later.
- In scratch Arrays are called lists. To add a list:
  1. In the **Variables** tab choose **Make a list**
  2. Add the items to the new list
  3. Use the items in various scripts

### 1 List Blocks

Questions  
 add thing to Questions  
 delete 1 of Questions  
 insert thing at 1 of Questions  
 replace item 1 of Questions  
 item 1 of Questions  
 length of Questions  
 Questions contains thing

### 2 A list full of items

Questions  
 1 God helps people who helps themselves  
 2 Only in the Dictionary success come before work  
 3 Think good and everything will be good  
 4 People deserve your trust, give it free of charge  
 5 Trust is better than money  
 6 Great things are waiting for you,  
 7 Patience is a virtue  
 8 When you have to shoot, shoot, Don't talk!  
 9 Money is going to fall from the sky  
 length: 9



### 3 Samples of list usage

move item 1 of Questions steps  
 say item 1 of Questions for 2 secs





### Your task...The Fortune Teller

Difficulty



1. Add two sprites- a fortune teller and a crystal ball
2. Design three costumes for the crystal ball sprite (a ball in 3 different colors)
3. Add a list and fill it with general statements as items

For programming use two blocks:

1. **Ask...and wait** this block is in the sensing tab and receive the user's input
2. **Say item (random number) of...** these blocks use random items from the list, and create an illusion of a conversation

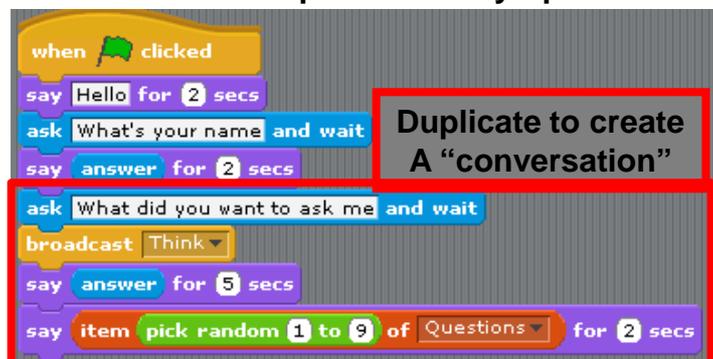
#### Add Crystal Ball and Fairy Sprites



#### Add Costumes and Scripts To the Crystal Ball Sprite



#### Add this Script to the Fairy Sprite



#### Related Cards:

- 1.1- Paint Editor
- 2.2-Motion Animation

#### Prerequisite Cards:

- 2.5-Coordination System
- 3.0-Variables
- 3.2-Math Operators

