

# Introduction to programming

(Basic)

Stimulate problem solving	2
Stimulate creativity	2
Stimulate critical thinking	2
Stimulate group work	1

Stimulate entrepreneurship	2
Informal learning enviro.	3
Technology use	3

## Practicalities



Preparation: 30 min



Duration: 90 min



Material needs:

- PC- computer with internet access



Group size range: 6 - 12

Ideal sub-group size: 1



Workshop made for: 10 - 12

Easily transferable to workshops for ages between: 10 - 99



Environment FabLab necessary: No



Educational area:

- \* Computer science
- \* Mathematics
- \* Technology
- \* (Visual) Arts

## Precognition

No precognition is necessary to do the step by step assignment.

## Wanted

Participants will probably have their own ideas on how to further develop the game or make something else. To be able to help one day of familiarizing with Scratch is recommended for the Educator.

## Preparation

Go to: <https://scratch.mit.edu/>

Click Join Scratch in upper right corner.

Select a username and a password and provide an email address (to my knowledge there will be no spam mails).

Children will need confirmation from a parent/guardian.

There is also a possibility to sign up for a teacher account which enables you to create and supervise student accounts.

Request a teacher account at the bottom of this page:

<https://scratch.mit.edu/educators>

You can start programming without creating an account, but you won't be able to save your work online.

# Workshop Guidelines

## Phase 1: Getting to know the layout



### Material needs:

*Essential: Laptop with internet access, Scratch account (preferred)*



### Goals:

Skill Goals (**Blue**)

Learn the basic concepts of:

S1 Basics of Block programming

Content Goals (**Green**)

C1 Familiarize with the layout of Scratch



**Background story:** Block programming is an easy way to start programming. Blocks will fit in certain ways and you will get a very visual image of your code. Today there are a few block programming languages and Scratch is a veteran. It's been around for over ten years and it's very straightforward and simple. Scratch is developed by MIT Media Lab to teach children programming. Scratch 2.0 took the platform online 2013. Since then it has grown to a big community where you can look at millions of programs from millions of users.

Goals	Activities	Duration
S1, C1	Get to know the layout of Scratch	10 min

## Phase 2: Make an interactive character



**Goals:**

Skill Goals (**Blue**)

Learn the basic concepts of:

S1 Sprites,

S2 Events

S3 Coordinates

Content Goals (**Green**)

C1 Make an interactive Sprite



**Background story:**

...

Goals	Activities	Duration
S1 C1	Insert a Sprite	2 min
S2, S3 C1	Make the Sprite interactive	6 min

## Phase 3: Adding items and enemies with sound



### Goals:

Skill Goals (**Blue**)

Learn the basic concepts of:

S1 Sprites

S2 Loops

S3 If ... then statements

S4 Sound files

Content Goals (**Green**)

C1 Make and automated Sprite with collision detection

C2 Add sound

Goals	Activities	Duration
S1, S2, S3, S4 C1, C2	Adding a Sprite with collision detection and sound	3 min
S1, S2, S3, S4 C1, C2	Adding a moving Sprite with collision detection and sound	8 min

## Phase 4: Keeping score, Initialization and Graphics



### Goals:

Skill Goals (**Blue**)

Learn the basic concepts of:

S1 Loops

S2 If ... then statements

S3 Variables

S4 Initialization

S5 Messages

S6 Graphics

Content Goals (**Green**)

C1 Make a Score that reacts to collisions

C2 Initialization

C3 Adding a Game Over screen

Goals	Activities	Duration
S3, S2 C1	Add a score	7 min
S4 C2	Set starting position and background	4 min
S2,S3,S5,S6 C3	Adding a GAME OVER screen	10 min

## Phase 5: Fine tuning



**Goals:**

Skill Goals (**Blue**)

S1 Testing and Configuration

Content Goals (**Green**)

C1 Fine Tuning your game

Goals	Activities	Duration
S1 C1	Test the game and change variables, graphics and sounds freely.	15 min



## Pedagogical tips

Make use of the open source code if the students want to develop their own games.  
By clicking *explore* on the startpage you can browse other scratchers games.  
You are able to see their code by clicking *See inside* on the game page.



## How to transfer to non-Fablab environment

No Fablab necessary



## Evaluation of achievements

...



## Content links

Scratch: <https://scratch.mit.edu/>

## Resources

Tutorials: <https://scratch.mit.edu/tips>

Activity cards: <https://scratch.mit.edu/tips> (further down)

Starter projects: <https://scratch.mit.edu/tips> (at the bottom)

Other more advanced block programming:

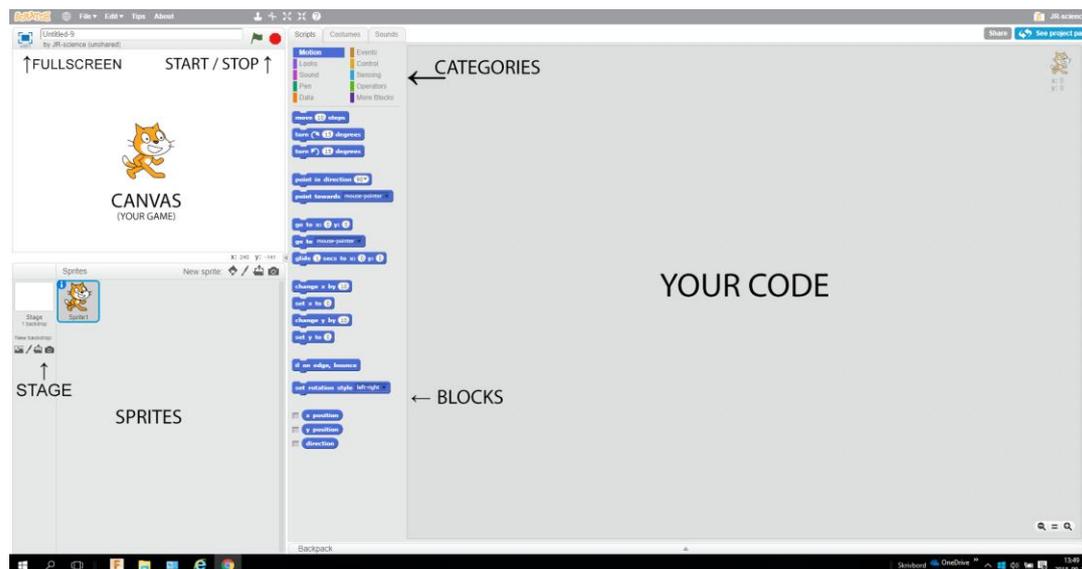
<http://www.stencyl.com/> Create games for appstore and Google Play

<https://developers.google.com/blockly/> Create Javascript for webpages

<http://appinventor.mit.edu/explore/get-started?> Create apps for android

## Getting to know the layout

Scratch is divided in four main windows.



### CANVAS

This is your game Objects can be dropped on the canvas or spawned by code.

### SPRITES (OBJECTS AND CHARACTERS)

These are your objects, that you can use in your game.

The objects are called Sprites. They are a collection of pixels forming an object and they can each have a specific code connected to them.

### CODE

This is where your code goes. Drag blocks here to make them part of your program.

There is a unique code for each Sprite

### BLOCKS

Each block contain a command and they can be put together forming instructions for your game. The blocks come in different shapes and will only fit in certain ways.

Blocks are divided into ten categories. The categories are colour coded to help you keep track of your code.

### START AND STOP

The green flag starts your code and the red stop sign stops the code. Running code will light up in the Code window.

### FULL SCREEN

Press to view the Canvas in Full screen

### SHRINK AND GROW SPRITE

Press inward or outward arrows and then a sprite to shrink or grow the sprite.

### STAGE

Choose or change the stage. Code can also be put on stage.

## Make an interactive character

### Sprites



First step is to name your character. As default the cat is named "Sprite 1". This is OK for now but as soon as you add some other characters, names will be confusing.

It's good programming practice to always name things what they are. Let's begin by naming the Sprite Kitty.

### Naming

Press **i** next to the cat and write "Kitty" in the text field. Then press the blue back-arrow to go back. Your character's name is now Kitty

Kitty is standing still in the middle of the screen. Let's give it the ability to move around by making some code for Kitty.

### Moving

Make sure Kitty is selected and click Events category in the blocks section.

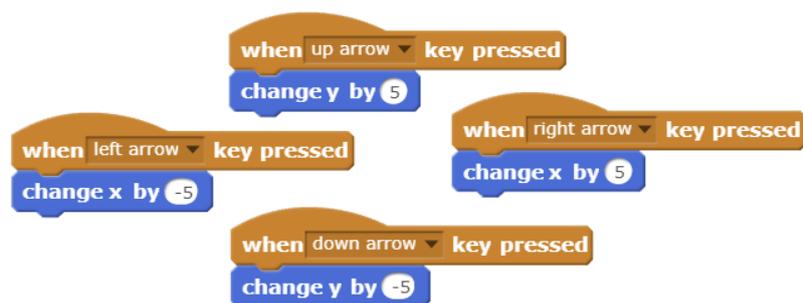
Use the "when \*\* key pressed" block. Drag that block to the code window and drop it there. Change the key that will trigger the event by using the roll-down list of the block. Choose "right arrow"

Blocks connected to the event block will be executed when the event is triggered. Let's connect a block moving Kitty to the right along the X-axis.

Choose Block category Motion and drag the block change x by \*\* click the text field on the block and input 5.

Kitty is now ready to move. Press right arrow to test the program

Continue to make events and move commands for Up (positive Y), down (negative Y) and left (negative X)



## Adding items and enemies with sound

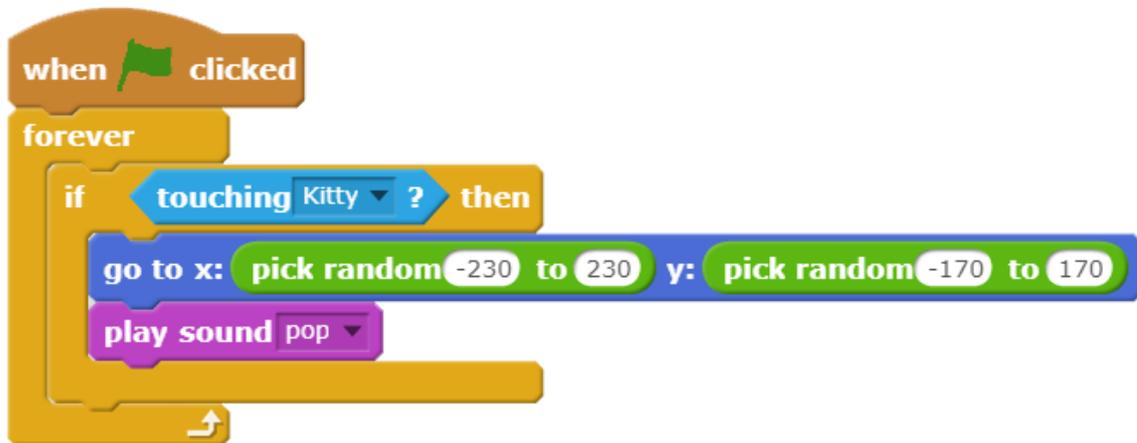
Let us give Kitty something to eat. Add another Sprite by clicking small character right beside New Sprite. Browse for the Cheesy-puffs in the category “things” and insert a new Sprite. Default name is Cheesy-Puffs so that’s already OK. The cheesy puffs are a bit big. Resize them by clicking the shrink icon in the menu bar and then the Cheesy puffs.



Sprites



Let us put together this bit of code. This is for the Cheesy puffs so select them in the Sprite window. Your code window should now be empty. Start dragging blocks to the code window. Remember that the block categories are colour coded and look for each block in the right category.



The first block is a start block. This starts to run the connected blocks when the green flag is clicked. It can be found in the category Events

The forever loop, found in Control category, repeats everything inside the bracket over and over forever.

We use this to contentiously look for a collision.

The If ... then block only runs the blocks inside the bracket when the cyan hexagonal statement is true. This is also a Control block

The cyan touching block is a Sensing block and can be either true or false

The blue block is a motion block and instantly moves the Sprite to a position.

The green pick random blocks are Operators

Now Kitty can run around and eat all day. Let's make it a bit harder by adding a vicious dog. Add a dog the same way you just added the Cheesy-Puffs

The code window should now be empty again.

The dog does not move and nothing happens if Kitty touches it.

Let make it move and make an if ... then statement that triggers when it catches Kitty.

Sprites

New sprite: 



Kitty



Cheesy-P...



Dog1

We start off with a start block to activate this bit of code.

The dog is turned in a direction so that it won't just bounce back and forth. Notice that this is done before the loop, so it is only activated once when the flag is clicked.

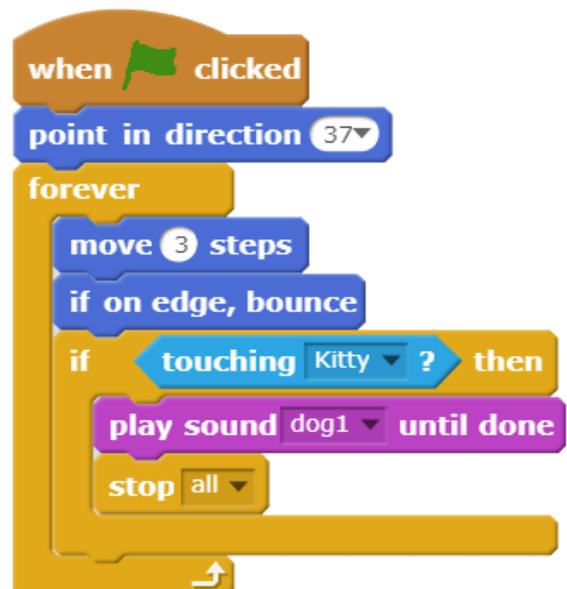
The loop repeats everything inside the bracket.

Move \*\* step is a different kind of move than earlier. Instead of using X and Y axis it moves the Sprite in the direction the Sprite is pointing.

If the Sprite reaches the edge we want it to bounce.

We continuously check if the Dog touches Kitty.

If it does we play a sound (Bark) and stop running the code



## Keeping score, Initialization and Graphics

The game is now playable by pressing the green flag but it still needs some work and tuning.

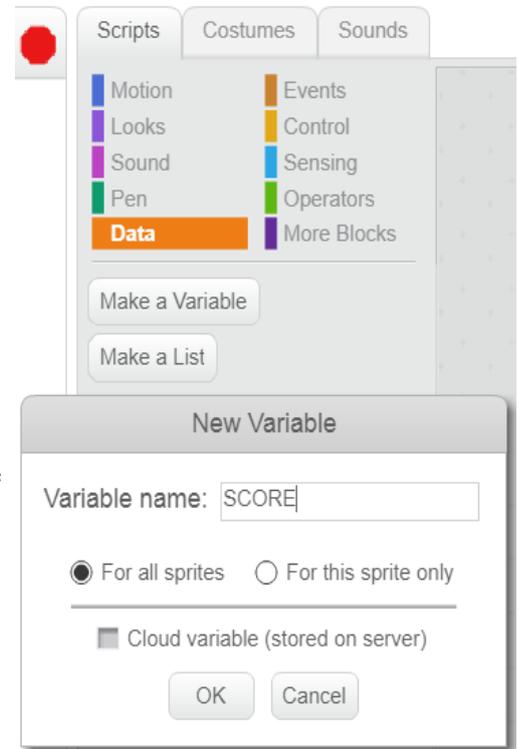
We want a Score to keep track of our progress in the game. To do this we will use a variable. A variable can store many different things. In this case we want it to store a number. We will then be able to perform operations on this number (regardless of what number it is). We can add, subtract, multiply, divide and so on.

To create a variable press the Data Category and choose New variable. The “for this sprite only” should be used when we store something not used by other sprites. We may add other things that can change the score later so we make the variable accessible to all sprites by selecting “for all sprites”

The score should be set to zero every time we start a new game and for every bowl of cheesy puffs Kitty picks up 5 points should be added.

Select the cheesy puffs and compliment the code by adding two blocks.

“Set score to 0” before the loop and “Change score by 5” inside the if ... then bracket.



Have you noticed that sometimes the Sprites starts in positions that makes it impossible to get away from the Dog. Let us do something about that by initializing the X- and Y-positions of each sprite so they start where we want when the green flag is pressed.

Stop the code from running by pressing the stop-button beside the green flag.

Drag Kitty, the Cheesy puffs and the Dog to the positions you would like them to start.



Click Kitty and insert a “When green flag is clicked” block.

For each of the three sprites: Drag the “go to x: \*\* y: \*\*” block found in Motion Category into the code window and place it directly under the “When green flag is clicked” block.

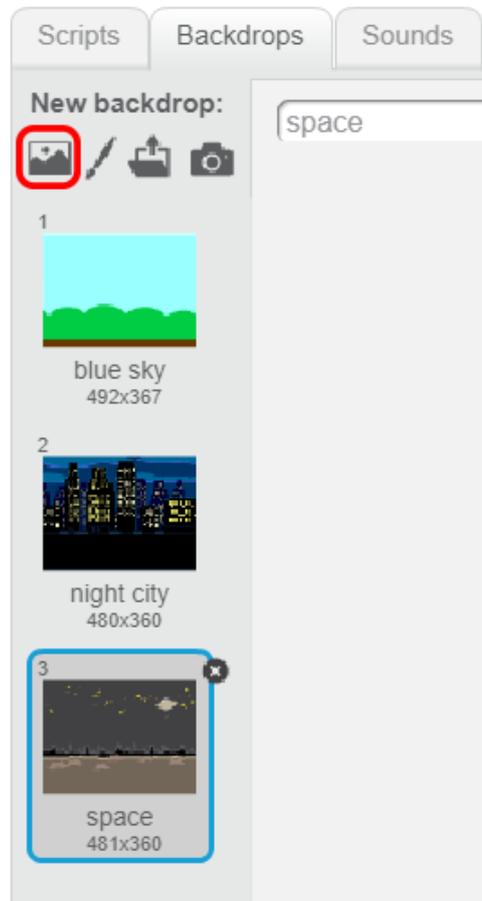
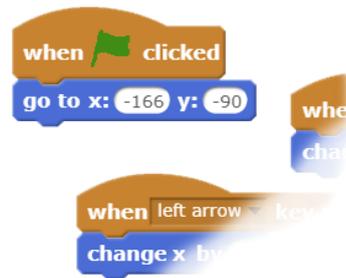
## Dog



## Cheesy-puffs



## Kitty

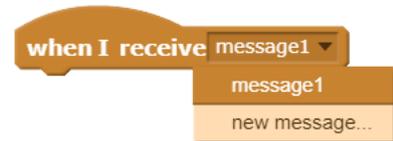


Add stages for Kitty's adventures by clicking the stage in the bottom left corner and the backdrops tab.

By clicking the import button (red rectangle) you can choose different backgrounds.

Import "blue sky", "night city" and "space" and delete the blank Backdrop by clicking the X in it's upper right corner.

We will make a call to bit of code when the Score is changed. Click the Scripts tab for the Stage and use the "When I receive \*\*\*" block from the Events category. Press the drop-down list and choose new message.



Name the message SwapBackground and insert the following code. (remember: blocks are colour coded)

Try to see what the code does.

Broadcast the message from inside the if ... then statement if the Cheesy-puffs sprite.

```
when I receive SwapBackground
  if SCORE < 25 then
    switch backdrop to blue sky
  if 24 < SCORE and SCORE < 50 then
    switch backdrop to night city
  if 49 < SCORE then
    switch backdrop to space
```

```
forever
  if touching Kitty? then
    go to x: pick random (-230 to 230) y: pick random (-170, 170)
    play sound pop
    change SCORE by 5
    broadcast SwapBackground
```

The

background should set to blue sky when the green flag is clicked.

Do you know how to do that?

Choose the stage, insert a “when green flag is clicked” and connect a “switch backdrop to blue sky” to that.

Finally we’ll add a GAME OVER sprite.

Add a blank sprite by clicking the pen next to new sprite in the sprites window.

Name the new sprite GameOver

Select the Costumes Tab and use the drawing tool or Text tool to make a Game Over Sprite. This Sprite will appear when the Kitty is caught by the dog.

Select the Scripts tab and use this code:

```
when green flag clicked
  hide
when I receive GameOver
  show
```

Broadcast GameOver from the Dog.  
Put the broadcast inside the if loop so it will be sent when the dog catches Kitty.

```
if on edge, bounce
  if touching Kitty? then
    play sound dog1 until done
    broadcast GameOver
    stop all
```

### Fine tuning

Test the game. Is it too easy or too hard?  
Make changes to the code.

The finished code can be viewed at:  
<https://scratch.mit.edu/projects/249366504/>



Game name

Game type (Action, Platform, Puzzle, Rollplay ...)

Main character

Enemy / Problem

Side character / item

Bikaraktär / item

Bikaraktär / item

Description of main character and enemy / problem

Description of interaction with side characters / items

Artifex 2019

Storyboard / plot / progress

Picture					
Description					

# Idémall för spel

Ni ska göra ett spel med tema strömavbrott

Om ni inte har en färdig grund idé går det att börja med vilken del som helst i idémallen.

Kanske vill ni starta med ett problem och sedan utarbeta huvkaraktären och därefter fylla på med bikaakterer och handling. Kanske har ni en supercool huvudkaraktär men inte mycket mera klart. Börja med att rita huvudkaraktären i så fall. Så fort ni sätter igång brukar det bli lättare att komplettera med de resterande rollerna.

Ni behöver inte fylla alla rutor, men se till att ha en gemensam grundidé så att alla i gruppen arbetar mot samma mål.

# MAKE A GAME WITH SCRATCH 3.0

Artifex 2019

Scratch is a block based, visual programming language, produced by MIT Media Lab. It has been existing for over 10 years, and is translated into more than 70 languages.

All code is available on <https://scratch.mit.edu/studios/4762521/>  
The game in this folder is called "My first game"

Create an account

Go to [www.scratch.mit.edu/](http://www.scratch.mit.edu/)

Click "join Scratch" in the upper right corner.

Choose a username and password and write a valid mailadress

Children need permission by a guardian to create an account.

There is also the possibility to create a educator account. An educator account has the possibility to create and supervise student accounts

Apply for a educator account at the bottom of this page:

<https://scratch.mit.edu/educators>

If you don't create an account you can still use scratch but you won't be able to save your work online.

When you're ready to start coding click Create in the upper left.

If Scratch is in the wrong language you can change the language settings under the globe

## 1. BLOCKS

Every Block has a specific function which are combined to create your program.

The Blocks are constructed as pieces of puzzle to help you with the structure of the code.

## 2. CATEGORIES

The Blocks are divides into 9 color coded categories.

## 3. CODE

This is where the code for the selected sprite is shown.

Drag and drop blocks here to make them a part of your program.

## 4. SPRITES

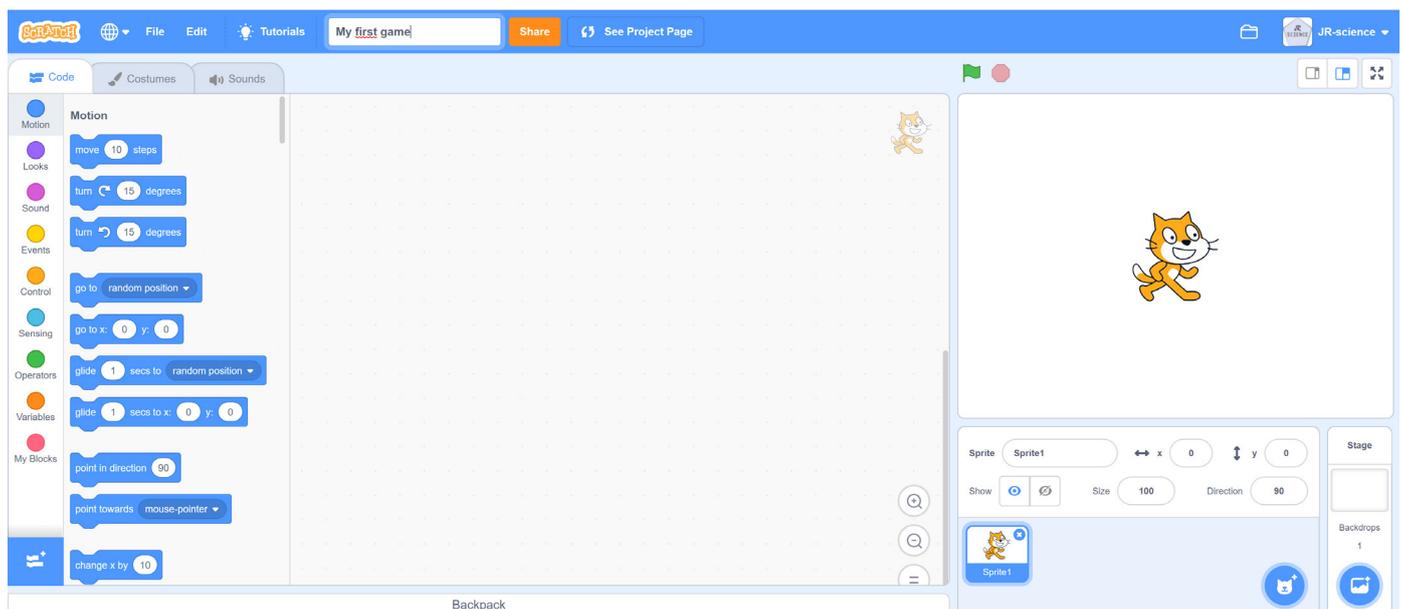
These are the objects you can use in your game.

You can use sprites from the Scratch Library, import or draw your own images.

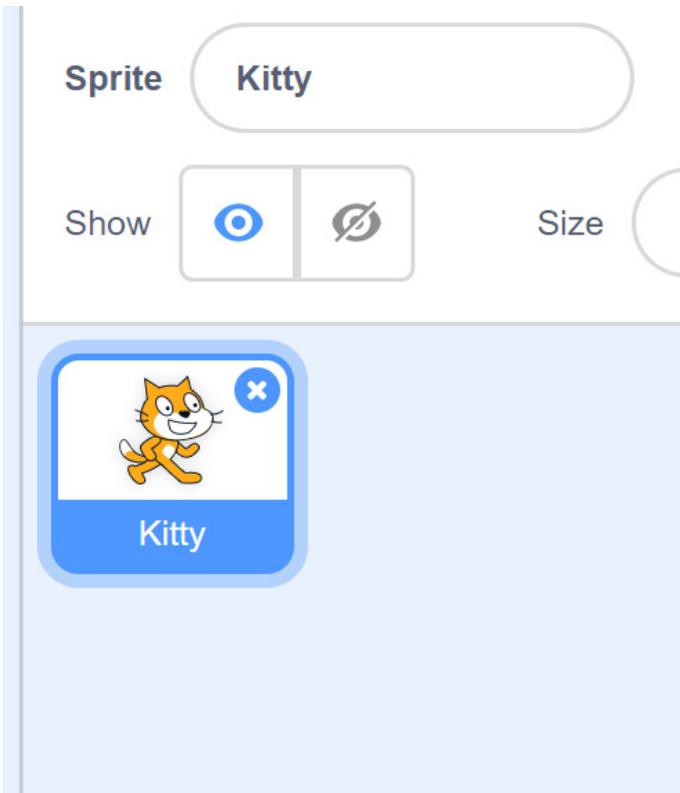
All sprites can have their own bit of code to describe its attributes.

## 5. CANVAS

The canvas is what is shown to the player. You can drag characters or objects to the canvas or generate



## My first game



A well chosen first step is to name your character. From the beginning it is named Sprite 1 which works momentarily, but when you add more objects it soon becomes confusing if they don't have descriptive names.

Tick the box and change the name to Kitty.

So far, Kitty is standing still. Let's give Kitty the ability to move by using a few bits of code.

Click on Kitty to insure that it's actually for Kitty we are modifying the code.

Choose the category xx and find the block xxx. Drag the block to the code window and drop it there. Change which key is supposed to be used by clicking the scrollbar and choose the right arrow.

Blocks placed beneath this block will be activated when the respective key is pressed.

Drag in the change x by 10 block from the motion category and snap it under the other block.

Test the code by pressing the right arrow key.



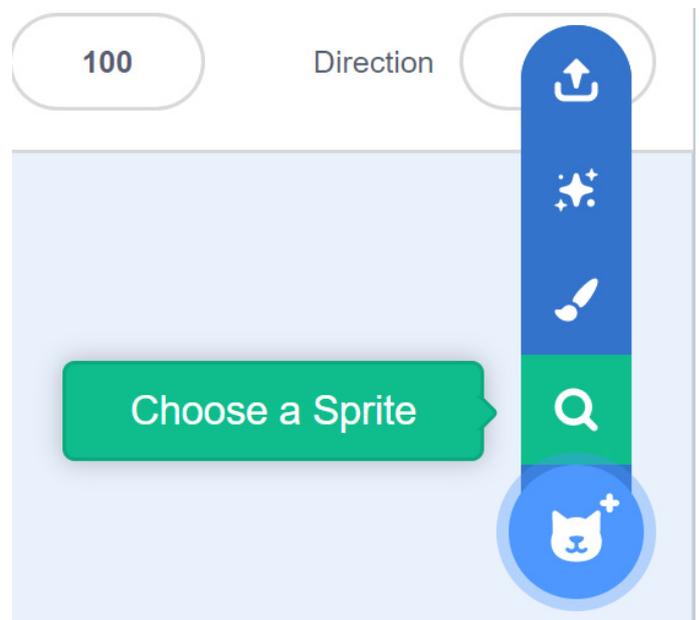
10 steps make Kitty a bit too quick. Change to 5 steps.

Do the same maneuver for the left, up and down arrows.

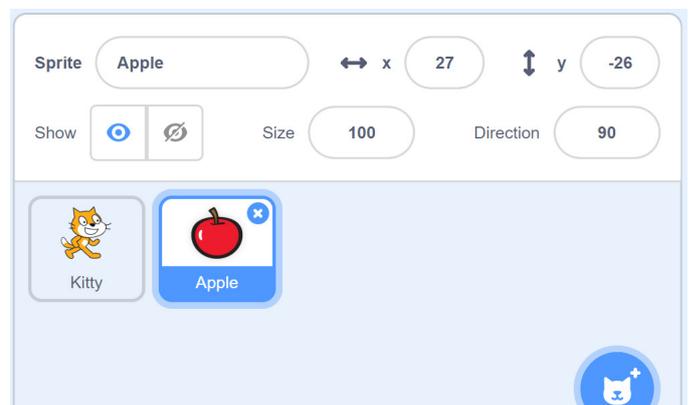
The left arrow key affects the X coordinate negatively. The up arrow key affects the Y coordinate positively. The down arrow key affects the Y coordinate negatively.



Now Kitty needs something to pick up. Create a new Sprite by clicking the cat down in the bottom right. Choose the Apple under the category Food.



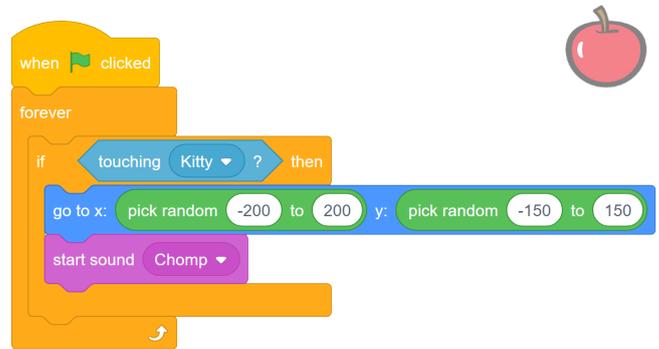
Your program now contains two different figures. Each of the figure has its own bit of code.



Click the apple to choose it. The code window should now be empty and you can insert code for the apple.

The apple is supposed to jump to a new, randomly chosen coordinate when Kitty picks it up.

Let's create a bit of code that continuously controls if Kitty is touching the apple.



You can easily find the blocks in use by using the color coded categories.

The first block is a starting block. It is activated when pressing the green flag.

The Forever block is a loop. Everything placed between the brackets is performed over and over again.

The "If ... then" block is checking if the criteria in the hexagonal box is valid. If it is valid the code within the brackets is performed.

The Touch block is controlling if the sprite is touching the Kitty sprite

The go to block moves the sprite to the specified coordinates

Pick random returns a random integer in the selected interval.

Play sound plays the selected sound

Now our Kitty should be able to move and pick up apples.

Test the program by clicking the green flag.



Let's increase the difficulty by adding an enemy.  
Add a dog in the same way you added the apple.



Click on the dog and make the following code:

Start block - to activate the code.

Forever block - will repeat what's inside the brackets.

Move 3 steps - moves the dog in the direction it is pointing in.

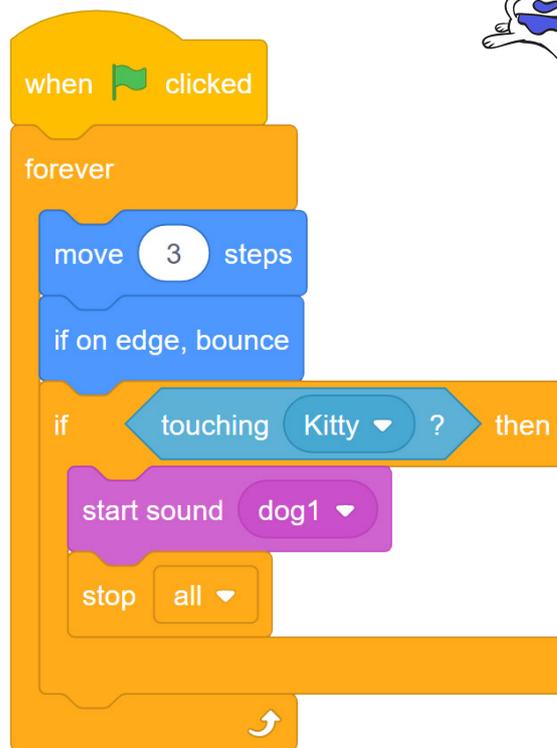
If on edge, bounce - will move the direction of the dog when it hits an edge.

If .. Then - the block will check if the criterias in the hexagonal box is true. If it is true it will execute the code that is written inside the brackets.

Touching - checks if the selected sprite touches the object that's been selected in the drop-down box.

Start sound block - plays the selected sound from the drop-down box.

Stop block - Stops the selected code.



The game is now playable but it still needs a few more features and some fine tuning.

We want a score system to keep score of our progress in the game. To do this we will use a variable.

A variable can store wide variety of things. In this case we want it to store a number. We will then be able to perform operations on this number (regardless which number it is). We can add, subtract, multiply, divide and so on.

## Score

Choose the category 'Variables' and then 'Create a variable'.

Name the variable 'Score' and then apply it to all sprites.



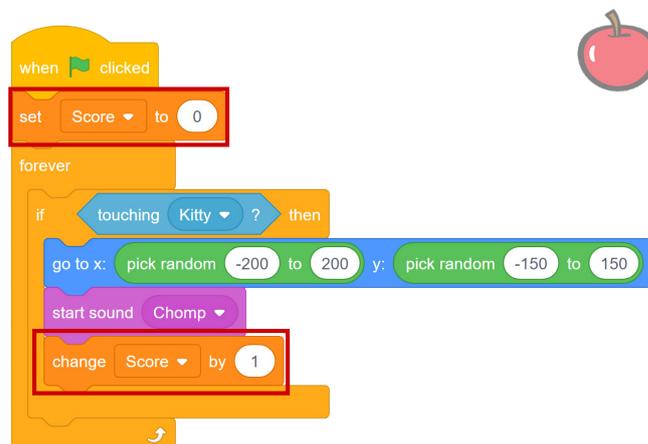
Now we have some new blocks in the category Variables.

Add a set Score to 0 directly beneath the starting block in the apple's code.

This block sets the points to 0 (zero) every time the game starts.

Add a Change Score with 1 at the bottom inside the if ... then brackets.

This block will increase the score by 1 (one) every time Kitty picks up an apple.



## Initiation

Lets make it the game more fair for different players by setting the start coordinates for all sprites.

Let Kitty start in the lower left corner.  
Click Kitty and add blocks for initiation.  
Kitty already has code for steering so just add the new blocks beside that code.



```
when green flag clicked
  go to x: -170 y: -110
```

Make the dog start in the upper right corner.  
Klick on the dog and add the following blocks.



```
when green flag clicked
  go to x: 200 y: 80
  point in direction 37
  forever
    move 3 steps
    if on edge, bounce
    if touching Kitty ? then
      start sound dog1
      stop all
```

To make the apple always start in the lower right corner: Add these blocks to the apple's code.

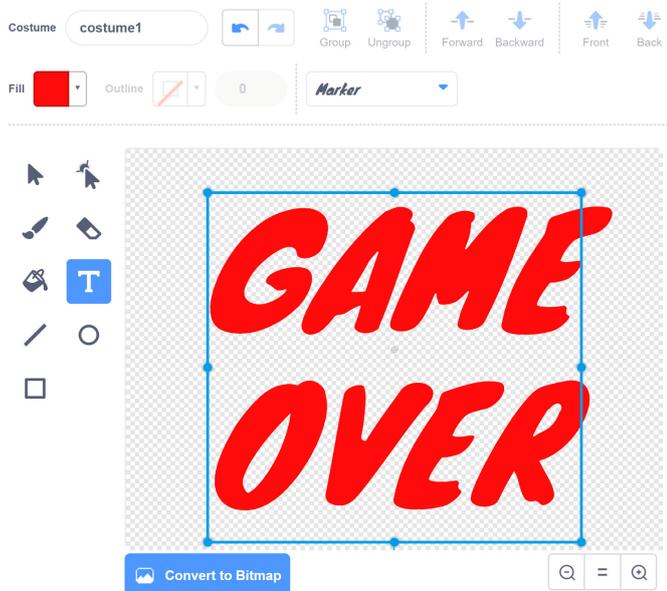
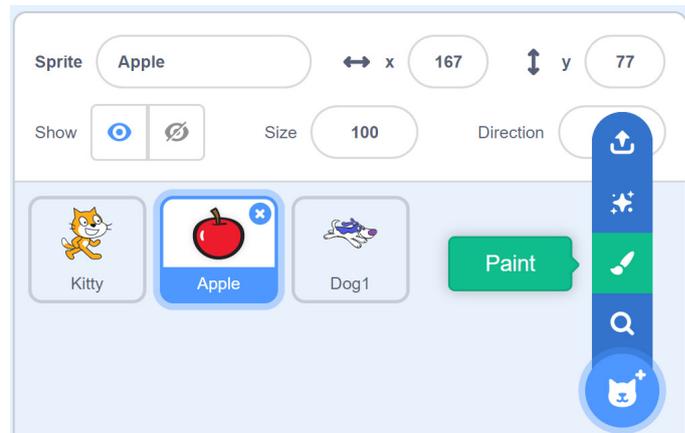


```
when green flag clicked
  go to x: 200 y: -120
  set Score to 0
  forever loop
    if touching Kitty ? then
      go to x: pick random -200 to 200 y: pick random -150 to 150
      start sound Chomp
      change Score by 1
```

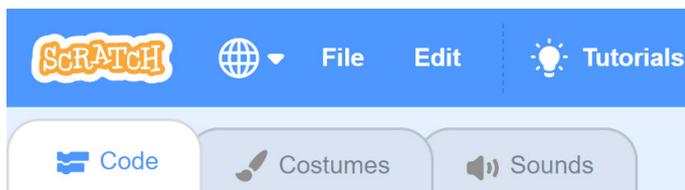
## GAME OVER

Right now everything stops when the dog catches Kitty. Let's add a GAME OVER text that is activated when Kitty is caught.

Create a new sprite, but choose the brush to paint your own sprite.



Use the text tool to write GAME OVER with a fitting font and drag a corner of the text to scale it.

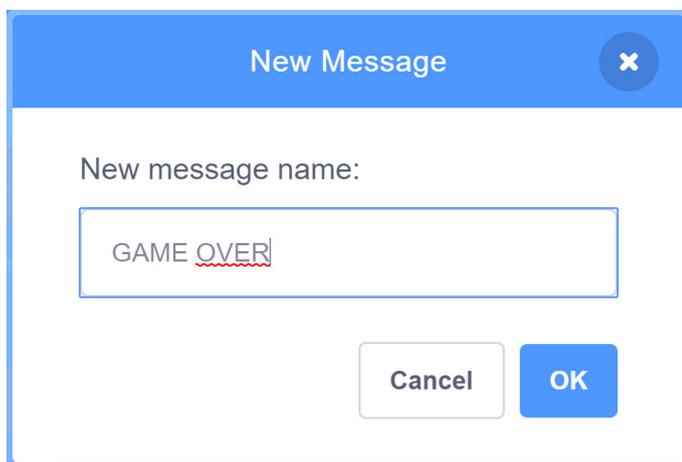
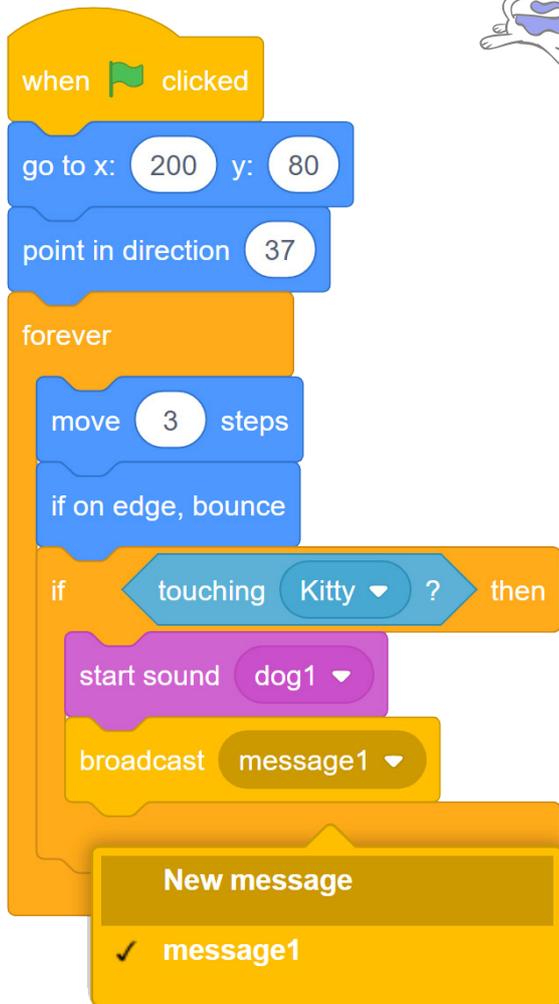


Now that the graphics are done, go back to view the code by selecting the Code tab in the upper left corner of the screen.



The GAME OVER sprite shall be shown when the dog catches Kitty. We need to send a message from the dog to the GAME OVER sprite.

Select the dog and swap the Stop all block to a broadcast block. Click the small arrow and choose new message from the dropdown list.

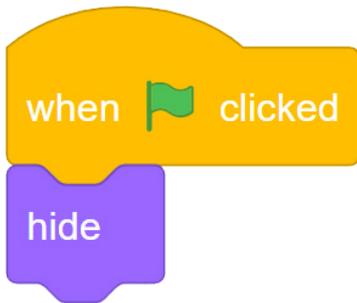


Name your message GAME OVER and press OK.



*GAME OVER*

Change to the GAME OVER sprite and set what should happen when the message is received.

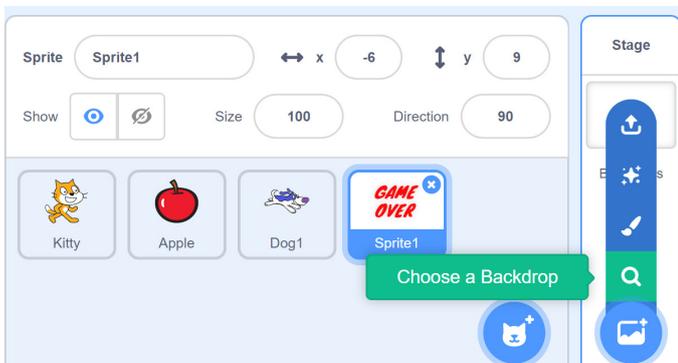


*GAME OVER*

When you start a new game the game over sprite should not be shown.

Add this code to the game over sprite

## Bakgrund



Let's add a backdrop to the game.

Tick Choose a Backdrop and select a backdrop you want.

## Your own games

Now that you have tried the basics in Scratch you are ready to design your own game. If you run into difficulties: Look at what others have done. All code is viewable and someone has probably made something similar.

At the bottom of the category window there is a button to add more categories. Blocks to make music, pen to draw the path of a sprite or speech synthesis are a few examples.

Remember to test your code often.

In Karlstad Makers Scratch-studio finns några exempel på hur man kan göra gravitation, flera rum att gå mellan, inventory, healthbar m.m.

You can find a few examples in Karlstad makers Scratch-studio. Gravity, multiple rooms, inventory and more.  
<https://scratch.mit.edu/studios/4762521/>

Good luck!



