



# Manual

How to create an animation using the online editor.

---

## Index

Introduction.....	3
First approach.....	3
Login .....	3
Page structure .....	4
File options .....	4
Export XML .....	4
Publish .....	4
Header .....	5
Sprite .....	5
Spawn / Animation / Child .....	6
Key words .....	6
Create your first pet .....	7
What you need .....	7
Upload/Setup your sprite .....	7
Application icon .....	8
Spawn .....	8
ID .....	8
Probability .....	8
Next .....	8
X/Y .....	8
Animations .....	9
ID .....	9
Name .....	9
Start move / End move X,Y .....	9
Offset Y .....	9
Interval .....	9
Opacity.....	9
Sequence .....	10
Repeat count .....	10
Repeat from.....	10
Action .....	10
Next/Probability/Flag .....	10



Border .....	10
Gravity .....	10
Childs .....	11
Animation ID.....	11
Next .....	11
X and Y .....	11
Remarks .....	11



## Introduction

The editor is a part of the eSheep project (<http://esheep.petrucci.ch>) to see again the lovely sheep from Tatsutoshi Nomura on the desktop.

The project is released as Open Source, so you can get the code from github. To allow users to make their own animation and to make it simple, I used an XML to create the sheep and its animations. So you don't need to be able to program! You can simply write a XML (text file) with the animations and the application will parse it and use it to show your sheep!

Everyone can create an XML or an XML editor to create new animations. I wrote an online and simple editor, so you can create your pet and deliver it to everyone. In this document, you will know what you have to do. **YOU DON'T NEED TO BE A DEVELOPER! IT IS EASY LIKE PAINT ;)**

## First approach

In this section you will see what is possible to do. Go to “create your first pet” if you already know what you can do.

### Login

Go to <http://ehseep.petrucci.ch> and press “Open Editor”.

Once you are there, you can create a new account or see the demo pressing “LOGIN” without username or password.





## Page structure

The screenshot shows the eSheep editor interface. On the left, a pink box contains XML code for an animation. In the center, a grey form titled 'Header:' contains fields for Author (Adriano), Version (0.2), Title (Browser Jr.), and Pet name (Browser Jr.). Below this is an 'Info' section with a text area and an 'Application' dropdown. To the right, a blue box titled 'Images:' shows a grid of small sheep icons. Below the form is a green 'Spawn:' table with columns for ID, Probability, Next, x, and y. At the bottom is a yellow 'Animations:' table with columns for ID, Name, and Preview.

ID:	Probability:	Next:	x:	y:
1	100	1	screenW-1	areaH-imageH

ID:	Name:	Preview:
1	drag	

Page is structured like the XML, there is:

- File options (Save, export xml and publish your animation as beta/release)
- Header
- Sprite image, with all frames
- Spawns/Animations/Childs to animate the pet

## File options

The page was wrote to work offline, so it is 100% in javascript. Every change you make is done on your computer and not on the server. So, when you finish, you can press "SAVE WORK" to upload the locally generated XML to the server. You can see the xml in the pink box.

The file (without images) can be no more than 32kb – this limit is set in the database.

## Export XML

Exporting a XML allow you to download your entire XML (with images) to your PC and check if it is correct. To try the animation, drag and drop the file to your sheep on the desktop and it will be opened automatically. You need this function only if:

- You want to test the animation locally
- You want to create an installer with your pet
- You want to use it on your webpage or with Microsoft Edge

## Publish

If you are happy with your work, you can publish it. So everyone can test or use your pet.

There are 2 ways to publish it:

- Create a beta version:
  - o Users and you can try your pet and know that it is not finished yet.
  - o It will stay as beta until you save the work again! Once saved you need to publish it as beta again if you want download it.
- Create a release version:
  - o Users know that it is a finished and working pet, they can rate it and use it.



- A release version is a 1 to 1 copy of your work, this XML will stay there until you publish another release.
- Once released, you can continue your work but the released pet doesn't change anymore until you release a new pet.

## Header

**Header:**

Author: Adriano Title: Browser Jr.

Version: 0.2 Pet name: Browser Jr.

Info: Test animation, with Browser walking on the screen! [br]Image from [link:http://kingasylus91.deviantart.com/]

[br] to insert a line break  
[link:http://...] to insert a link

Application: 1

**Sprite:**

tiles X: 16

tiles Y: 12

Transparency: Magenta

**Icon:**

Durchsuchen...

You can write the main information about your animation in this gray box.

Author: Your name

Version: Give a version to your pet (normally 0.1 to 0.9 in the beta phase)

Title: Title of your animation

Pet name: Name of your pet

Info: Some information about your animation and you, some copyright texts

Application: Application version. In future, if the xml was modified, it is possible that you have to insert 2 as application.

Icon: This icon will be used as tray icon and shortcut icon in the application

Sprite: Say how many tiles you have in X and Y (sprite divisions) and give a colour for the transparency – this is used for the C# application, because there is no transparent colour.

## Sprite

Upload a sprite sheet with all your frames, remember that every frame has the same dimensions.



## Spawn / Animation / Child

The entire animation is described here.

At beginning the pet will be spawned:

- Pet will be positioned on the screen
- Next animation will be defined
- Probability to use this spawn

Once the pet is on the screen there are different animations, each animation defines:

- How many pixels to move in x and y
- Interval to the next move
- Which frame and sequence to use from the sprite sheet
- Actions if the pet reach a border, detect gravity or the sequence is over

Every animation can create another pet (child), this is used for advanced functions:

- Set the parent animation that generate this child
- Set the position of the child frame
- Set the next animation to use for this child

## Key words

All coordinates need a number (for example to set a position or move). Some time you don't want to set the pet at 300px but set it dynamically on the centre of the screen. To allow this, you can use some predefined key values, they will be parsed by the program.

screenW	Width of the screen
screenH	Height of the screen
areaW	Width of the working area (screen without taskbar)
areaH	Height of the working area (screen without taskbar)
imageW	Width of the image (single frame of the sprite)
imageH	Height of the image
random	A random value from 0 to 99
randS	A random value (like random, but this number is defined when the application starts)

So if you want to set your pet at the centre of the screen, you have to write:

$screenW/2 - imageW/2$       or       $(screenW - imageW)/2$

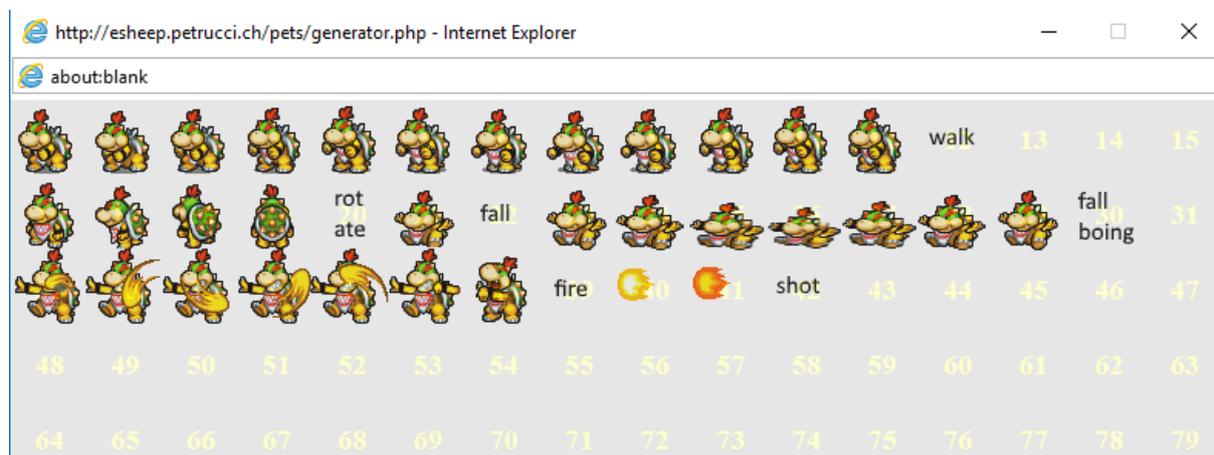


## Create your first pet

### What you need

You need “only” a sheet with all the possible positions of your pet! If you have it, 80% of your work is already done! The XML doesn’t do anything other than animate your pet using this sheet.

Something like this:



**Note: every frame has the same size! You can’t create a bigger and a smaller frame!**

To create your own sprite, you can use editors like <http://www.piskelapp.com/>

There are many sprites on the web:

- <http://www.deviantart.com>
- <http://www.pinterest.com>
- <http://www.spritedatabase.net>
- <http://images.google.com>

Use the key word “sprite sheet” to find what you search ;)

Once you found your sprite sheet, be sure that if you divide the image, every frame has the same size. You can edit the image with applications like [www.getpaint.net](http://www.getpaint.net) drawing a table on a second layer. I can’t explain how to edit, this should be written on another tutorial :P More info about create a sprite sheet can be found here: [http://spritedatabase.net/tutorial\\_sheet.php](http://spritedatabase.net/tutorial_sheet.php)

### Upload/Setup your sprite

Once you have a sprite with more than 2 frames, you can upload it on the page.

You need to set this in the Header (gray box):

Tiles X: divisions in the X axis (for browser Jr. it was 16 => 16 frames for each line)

Tiles Y: divisions in the Y axis (for Browser Jr. it was 12 => 12 frames each column)

In this way, Browser Jr. can have  $16 \cdot 12 = 192$  different positions (frames).

This is all, SAVE your work and click on the sprite image to see a grid over your sprite and get the frame ID of each single position.

Now you can give a pet name, insert your name and a version.



## Application icon

The application change icon as soon a new pet was selected/loaded. To load an icon, you need to convert a png. You can do it with this online tool:

<http://esheep.petrucci.ch/?pagina=utilities> → PNG to ICO converter

Select a single frame of your sprite and save it as PNG on your computer.

Open the tool and convert the image to icon. Save the generated icon to your PC and upload it on the editor to use it.

**Note: you can use the generated icon also for your home page and for Windows.**

## Spawn

Spawn mean “born/positioning”, see [https://en.wikipedia.org/wiki/Spawning\\_\(video\\_gaming\)](https://en.wikipedia.org/wiki/Spawning_(video_gaming))

Spawn:				
ID:	Probability:	Next:	x:	y:
1	100	1	screenW-1	areaH-imageH

### ID

Each Spawn has a unique ID, this is not used at the moment.

### Probability

If you have 2 or more Spawns, you can insert a probability. It is up to you to have a total of 100 if you are calculating in %. If you have 2 spawns with 80 each, every spawn has 50% of probability to be taken at beginning.

### Next

ID of the next animation. Once the pet was spawn, this animation will be executed.

### X/Y

Start position of the pet, you can insert an absolute pixel value or some defined Key words.



## Animations

You can define a lot of different animations (walking, running, dragging, falling, etc...):

ID:	Name:	Preview:
2	walk	
Start move:	x: -2 y: 0 offset Y: 3 interval: 100 opacity: 1.0	
End move:	x: -2 y: 0 offset Y: 3 interval: 100 opacity: 1.0	
Sequence:	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 repeats counts: 5 repeats from: 0 action: -	
	Next: 2 Prob: 98 Flag: none Next: 7 Prob: 30 Flag: none Next:   Prob:   Flag: none Next:   Prob:   Flag: none	
Border:	Next: 3 Prob: 100 Flag: none Next:   Prob:   Flag: none Next:   Prob:   Flag: none Next:   Prob:   Flag: none Next:   Prob:   Flag: none	
Gravity:	Next: 5 Prob: 100 Flag: window Next:   Prob:   Flag: none Next:   Prob:   Flag: none Next:   Prob:   Flag: none Next:   Prob:   Flag: none	

### ID

Unique ID, each animation has another number and it is used to set the next animation to play.

### Name

Used for debugging the animation. 4 animations can have a defined name, to interact with the application. So if your animation has one of the following names, the animation will be set automatically:

- "drag": if the pet is picked up with the mouse
- "kill": if the application was closed -> allow pet to show an animation for 1 second
- "sync": if the user press "cancel" in the about box -> every open pet will be synchronized

### Start move / End move X,Y

How many pixels the pet should be moved on each step in x and y. If your sequence has 10 frames, you can set the start values and end values for the movement. So, all 10 steps will be interpolated between "Start move" and the "End move" values.

### Offset Y

Sometimes you want to see your pet over the taskbar or over the window, even if the detection is on the border. Setting an offset, the border detection is still on the edge of taskbar/window but it is drawn Y pixels over/under the taskbar/window.

### Interval

Times to show the pet with this frame. Don't use a short interval, users don't like a fast movement for a background animation. So use times between 50 and 200 to give a best user experience.

### Opacity

Default: 1.0. You can set a transparency to your pet. 0.0 for 100% transparency, 1.0 for 100% opaque.





## Chlds

Chlds:			
Animation ID:	Next:	x:	y:
8	9	imageX	imageY

Chlds are “cloned” pets. For some animations you want another sprite. For example, a flower so that the sheep can eat the flower. Or a fireball for Browser Jr. This child should not create another pet (but it can). This is like a spawn but for chlds.

### Animation ID

Set the parent animation. Once this animation ID is played, a new child will be automatically created. So you know that child and parent will be synchronized on that animation.

### Next

The animation to play for the child, use a dedicated animation for the child (like flower, fireball, bath and so on).

### X and Y

Start position (like spawn) for the child. Use imageX and imageY to get the position of the parent.

### Remarks

It is possible to create another sheep or pet with this child, but this is not the idea and probably the user will not be happy to see always more pets. This child will be automatically removed if the sprite goes outside the screen or if there is no “next” in the sequence section.