Custom Objects: Animator's Manual

Contributed by SiamJai

With the release of Knytt Stories 1.2 , custom objects (CO) have been introduced. This manual explains the basic usage principles of COs; how to create, configure and embed them into the game.

Custom Objects overview

The most common use of COs is animation.To create custom animated objects, the following three things are required:
- A spritesheet that contains the individual frames of the animation;
- A few lines of code for configuring various properties of the animation,
- An object bank symbol for placing the animation into the game.

1. The spritesheet

Every animated object consists of frames. In Knytt Stories, these frames are rectangular tiles that are spread out sequentially on a flat PNG, the animation spritesheet. The game reads this spritesheet from left to right; rows top to bottom, and then displays it as an animated object in a process shown below.

Figures 1&2. The static spritesheet (left) and the rendering action (right). In the game, grey areas are invisible to the viewer.

1.1 Anatomy of a spritesheet

The frame tiles must have a specified height and width; these values will be the same for all frames. The overall dimensions of the spritesheet are not important, as long as the frame tiles are sequential and set right next to each other without gaps or overlap. If there are multiple rows of frames, they must be arranged from top to bottom. Place the final PNG spritesheet in Knytt Stories\Worlds\yourlevelname\Custom Objects\. KS supports alpha transparency. To make transparent regions in your animation, save the spritesheet as a PNG32 with no matte. If your graphics program doesn’t support that, then color the transparent regions fully black (#000000, RGB: 000) before export. Extra space on the right and bottom sides of the spritesheet can be used for displaying credits and miscellaneous information.

2. The code at a glance

Users have a handful of options for configuring the animation. These options have to be written in the World.ini configuration file found in the folder of your level: Knytt Stories\Worlds\yourlevelname\.

Open the World.ini file with Notepad or your preferred text editor. Depending on your previous work on the level, some command lines may be written already. Scroll down to add your CO parameters to the end of the file. The figure below illustrates some of the most common code tags. These and other parameters are explained in detail in the following section.

Figure 3. Common code tags and the way they apply to the spritesheet. Note that numbering starts from 0.

2.1 Anatomy of the code

```
[Custom Object n]
Image=pacman3.png
Tile Width=24
Tile Height=24
Init AnimTo=11
Init AnimSpeed=525
Offset Y=0
```

The above code is used to configure the PacMan animation in our previous example. Each line is explained in detail below.

[Custom Object n]

Always start with this bracketed command. It tells the program which object bank symbol the code is referring to. (Read more about object bank symbols in the next section.) The italicized n in the above example must be replaced with the number of the object bank symbol. If this is your first animation for this level, type "1" and use object bank symbol 1 in the Level Editor. (In our example, "Custom Object 4" was used, along with object bank symbol 4 - see Figure 4 later in the chapter.)

Image=pacman3.png

This is where you need to enter the name of your spritesheet. Only the filename has to be specified, since the program will automatically look for it in the Custom Objects folder.

Tile Width=24
Tile Heigth=24

These parameters are used to specify the dimensions of a single frame. If left blank, they default to 24 pixels. If your spritesheet spans more than one row, you must specify the width and height of the individual frames, even if they are 24 pixels, as the example shows.
Init AnimTo=11

This code specifies the last frame tile in your animation spritesheet. In most cases, you can get the value of this code by subtracting 1 from the total number of frame tiles. (The program counts from 0; see Figure 3 .)

Init AnimSpeed=525

Here you can set the speed of the animation. Technically this is not required, but if you find the animation too slow or too fast with the default value, change it here. (Increasing this number makes the animation faster.)

Offset X=0
Offset Y=0

Change the horizontal/vertical position of the animation in the level. By default, the animated object is centered on the object bank symbol. This often leads to incorrect positioning on the 24-pixel grid of the Level Editor. Offset Y and Offset X allow pixel-perfect positioning of these objects, regardless of their center and dimensions. When specifying these values, keep in mind the inverted coordinate system of Knytt Stories. If you want to nudge the object 5 pixels upwards, you must enter -5.

The rest of the parameters go as follows: Init AnimRepeat

Despite having no “loop” in the name of this code, it actually specifies the number of times the animation will loop. If left out, the animation will loop indefinitely. To prevent looping, type 1 instead of 0.

Init AnimLoopback

This is a rarely used parameter. In case you want the animation to loop from a tile other than the first one, specify the loop-starting tile number here.

Init AnimFrom

Another rarely used code: useful only when you want to start the animation from a frame tile other than the first one.

3. Object Bank symbols

Once you prepared the PNG spritesheet, placed it in the correct folder and wrote the appropriate code in the World.ini file, open your level in the Knytt Stories Level Editor. This guide assumes a working knowledge of the Editor’s functions. If you need to brush up on your editing skills, the KS Level Editor Manual covers the use of this software in detail.

Navigate to the screen you would like the custom object to appear in and select a layer (#4 or higher). Open Bank 255 and pick the object symbol that has the same number that you specified with the [Custom Object n] code.

Figure 4. Select the bank and object for the CO symbol, place it in the level and then test the results.

3.1 Testing and Troubleshooting

Once you placed the object symbol on the screen, save and test the level. Check for the following:
- Is the animation positioned correctly? (If not, adjust the Offset values.)
- Does it play with the right speed? (If not, adjust the Init AnimSpeed property.)

- Is it the right size? (If not, adjust the sprite sheet)

If your object is for public release, test it against different gradients. Sometimes an animation looks smooth on dark gradients but badly pixelated against light backgrounds and vice versa.Sometimes the KS preview window may crash. This could be due to incorrect naming and/or parameters in the World.ini file, incorrect positioning, or if the object dimensions are too large.

That's all there is to creating custom animated objects for Knytt Stories. If you have questions, comments about this manual or about animation in general, drop by on the TW forums and let us know! For more custom objects, I recommend visitng Nifflas' Support Forum .