

Scroll_snap

© 2015 Chris Sanyk

Current version: 1.1.1

Important links

- scroll_snap GameMaker Marketplace page:
<https://marketplace.yoyogames.com/assets/1865/scrollsnap>
- current version of this document:
https://docs.google.com/document/d/1tr2hHGlrQXLR3IFjLC7oEOQrw3UXbd3b68-iv_KAUQ/edit?usp=sharing
- author twitter: [@csanyk](https://twitter.com/csanyk)
- author's main site: <https://csanyk.com>

License

scroll_snap is subject to the GameMaker Marketplace EULA.

- <https://marketplace.yoyogames.com/eula>

Scroll_snap Overview

scroll_snap is a simple extension consisting of two functions: `scroll_snap_init()`, and `scroll_snap()`.

These two functions effectively create a virtual “grid” of “screens” that the main view will snap to as it follows a designated instance about the room. This allows the developer to use a single, large room to simulate a number of small rooms that are all concurrently active, without the need to deal with persistent rooms and objects, or dealing with state, and best of all the “rooms” continue to run when the player moves from one screen to the next.

When called during an **Outside View Event**, the `scroll_snap()` function will work whenever the instance moves outside the View, whether by moving (using speed), “position jumping” (assigning the x,y variables new values), or by speed and direction. You do *not* need to set the Follow Object in the room editor to the object; the `scroll_snap()` function handles this instead.

In the very early days of video games, computer hardware wasn't powerful enough to allow developers to create games that had large, scrolling environments. Before sufficiently powerful hardware came about to allow scrolling, videogames that took place in a larger world than would fit on one screen worked by loading a new screen when the player exited the edge of the screen. Atari 2600 games such as *Pitfall!*, *Adventure*, *Superman*, and *Berzerk* worked in this way.

Scroll_snap allows this style of game to be created in a single room, simplifying the management of instances by avoiding room changes, and allowing action to continue happening off-screen in different areas of a large GameMaker room.

One great advantage of `scroll_snap()` when used this way, it allows you to create a collection of “pseudo-rooms” all contained within one large room. This enables you to keep all of these pseudo-rooms concurrently active, which is impossible in GameMaker if you were to split up the game into distinct GML Rooms.

If you wish, you can deactivate instances outside the current view whenever the view is updated by the `scroll_snap()` function..

Demo

To get a feel for what scrollsnap is, create a new project, add the extension, and import everything. Then run the demo room.

The player object (oScroll_snap_demo) can move about using the Arrow keys, or teleport using the Enter key. This is to show that, whether you cross a room border, or appear in any random location in the room, the scroll_snap() function will work correctly.

The player can walk outside the edge of the room and appear on the other side, too. When you move off the edge of the View, the scroll_snap effect will update the View, positioning it at the next “screen” that the player instance has entered. Rather than scrolling smoothly to follow the player, the player appears to walk off the edge of the screen, and appear in a new screen. All of the “screens” are contained within a single GameMaker room. Even when teleporting to a random point in the next “screen”, the scroll_snap() function handles things properly.

There is an instance in the “1” screen of the demo room, oDeactivation_demo, which emits a beep every 2 seconds. If the disable_outside_view option is set to true, the beeping object will be disabled when the player moves outside of the “1” screen (and thus stop beeping), and re-activate when the player returns to the “1” screen. Press the “D” key to toggle whether the object is disabled when outside the view or not.

Usage

To use the extension, add it to your project. The extension provides the following two functions:

scroll_snap_init(which_view, xview, yview, wview, hview)

This init function sets up your snap scrolling system. It accepts the following parameters:

argument	type	acceptable values	description
which_view	real	0-7	designates which of the 8 built-in GameMaker Views is to be used for the scroll_snap system.
xview	real	0-room_width	the x-coordinate for the top-left corner of the view.
yview	real	0-room_height	the y-coordinate for the top-left corner of the view.
wview	real	0-room_width	the width of the view in pixels. For best results, use a number that divides evenly into room_width.
hview	real	0-room_height	the height of the view in pixels. For best results, use a number that divides evenly into room_height.

The function enables the view and makes it visible.

You can use other GML view functions to manipulate the snap_scroll view, however doing so may not work with the snap_scroll system, and cause it break.

scroll_snap(which_view, disable_outside_view)

argument	type	acceptable values	description
which_view	real	0-7	designates which of the 8 built-in GameMaker Views is to be updated by the scroll_snap() function.
disable_outside_view	boolean	true, false	determines whether instances outside the box defined by the new view position are disabled (if true) or left enabled (if false).

Call this event in the Outside View Event of the object you want the scroll_snap system to follow. When the instance triggers the Outside View Event, the View designated by which_view will update its position, snapping to follow the instance.

To disable instances residing outside of the box defined by this updated view position, set disable_outside_view to true. If you want instances residing outside the view to remain active, set disable_outside_view to false.

If you have certain instances that you want to leave enabled when outside view, you'll probably want to use disable_outside_view == false, and manage instance deactivation in your own way.

History

version	notes
1.0	initial release
1.1	- Added second parameter to scroll_snap function to control activation/deactivation of instances residing outside, reactivate anything inside the new view. - Demo improvements: - Coordinates of the scroll_snap View are now drawn to screen to make it more apparent what's going on when you move outside of View. - Added a new demo object to demonstrate instance deactivation feature.

	c. Simplified implementation of the player demo object.
1.1.1	Added second demo room with split screen view to better show what's going on in the room.