

# Creating Maps in ScrollEd

ScrollEd is a map/level editor that can be used with Scroll.

## Compiling ScrollEd with Scroll

ScrollEd is included in the Scroll header files, so you don't have to make any changes to your Scroll project to use it. However, you do have to make sure it is being compiled.

If you have the following defined in a source or header file, comment or remove it. Otherwise, ScrollEd will not be compiled when Scroll is compiled.

```
#define __NO_SCROLLLED__
```

## Running ScrollEd

Running your game's executable will start the game by default. If you want to start ScrollEd, you need to run the executable with the

```
-editor
```

command line switch.

You must also specify a map file to edit using the

```
--map
```

switch.

If you don't want to specify a map file on the command line every time, you can specify a default map file in Orx config as follows:

```
[Param]
map = ../data/map/Level1.map; <= Setting a default map if none is specified
on the command line
```

Orx objects to be used in ScrollEd need a new property not supported in Orx Used to define "object sets" used in ScrollEd

```
[0-Boy]
Graphic = G-Boy
ScrollEdSet = Players

[0-EnemyBug]
```

```
Graphic = G-EnemyBug  
ScrollEdSet = Enemies
```

After you've defined the ScrollEdSet for each of your Orx objects in config, run ScrollEd again. You will be able to use the mouse wheel to scroll through the object sets. The previous config would create two sets, Players and Enemies, and allow you to switch between them.

The following keyboard and mouse commands are available in ScrollEd. You can find them in data/ScrollEd.ini

```
[Input]  
SetList = ScrollEdInput # ScrollInput  
  
[ScrollEdInput]  
MOUSE_LEFT      = Action  
MOUSE_RIGHT     = Pan # Zoom  
MOUSE_MIDDLE    = Copy # HideSet  
  
MOUSE_WHEEL_UP  = NextSet # NextLayer  
MOUSE_WHEEL_DOWN = PreviousSet # PreviousLayer  
  
KEY_SPACE       = ClearSelection # HideSet  
  
KEY_LALT        = SoftClearSelection  
KEY_RALT        = SoftClearSelection  
  
KEY_F12         = Screenshot  
  
KEY_LCTRL       = Zoom # PreviousLayer # NextLayer  
  
KEY_DELETE      = Delete  
KEY_BACKSPACE   = Delete  
  
KEY_F6          = Save  
KEY_F9          = Reload  
  
KEY_Z           = ToggleMove  
KEY_X           = ToggleScale  
KEY_C           = ToggleRotate  
KEY_S           = ToggleSnap  
  
KEY_G           = ToggleGrid  
  
KEY_D           = ToggleDifferential  
  
KEY_A           = Antialiasing  
KEY_T           = Tiling
```

```
KEY_R      = EditAlpha
KEY_Q      = EditRed
KEY_W      = EditGreen
KEY_E      = EditBlue

KEY_TAB    = ResetZoom

KEY_F3     = ToggleFullScreen
KEY_HOME   = ReloadHistory

KEY_ESCAPE = Quit

KEY_F5     = StartGame

CombineList = @ScrollEdInput.KEY_LCTR
```

Place your objects as you like them and then press F6 to save your map!

## Loading ScrollEd maps in Scroll

When you want to load a new map, you need to tell Scroll to load the new map. You probably also want to pause the game while this is happening. Your game's Update function might be a good place to check and implement this.

Use the following routine to accomplish the map load:

```
// Sets map name
SetMapName("MyMap.map");

// Pauses game
PauseGame(OrxTRUE);

// Stops game
StopGame();

// Loads map
LoadMap();

// Unpauses game
PauseGame(OrxFALSE);

// Starts game
StartGame();

// Clears restart status
mbRestartMap = OrxFALSE;
```

SetMapName, StopGame, LoadMap, PauseGame, and StartGame are all defined in the Scroll classes.

Scroll also includes an OnMapLoad virtual function callback. You can override this if you want to wait until the map is loaded to execute code.

From:  
<https://orx-project.org/wiki/> - **Orx Learning**

Permanent link:  
<https://orx-project.org/wiki/en/tutorials/orxscroll/maps-in-scrolled?rev=1345403609>

Last update: **2017/05/30 00:50 (5 years ago)**

