

ARC ENGINE

by Khas Arcthunder

About:

Arc Engine – Platform for RPG Maker VX Ace

Created and developed by Khas Arcthunder

Blog: <http://arcthunder.com>

Forum: <http://rgssx.com>

Twitter: <http://twitter.com/arcthunder>

Youtube: <http://youtube.com/user/darkkhas>

Terms of use:

Check the latest terms of use at <http://arcthunder.com/terms>. Notice that this engine is free only for non-commercial projects. My work is licensed under a Creative Commons License.

Updates and Support:

Get official support at <http://arcthunder.com> or at <http://rgssx.com>. The first address is my blog, and the second one is the RGSSX Project community.

Before using ARC ENGINE:

Backup your Project. The ARC ENGINE developer is not liable for any issue or error in your project. If you find a bug, please post it on ARC ENGINE page at arcthunder.com or at rgssx.com.

Installing ARC ENGINE:

Copy the ARC ENGINE script under the “additional scripts” part. Read carefully all the instructions to configure the script correctly.

How to Use:

Read this manual carefully, and study the demo. Also, read the instructions written on the script.

Main Menu:

The ARC ENGINE's menu will appear when you launch your game in test mode. If you want to hide it, go to the configuration part and set **Disable_Editor** to **false**. If you want to enable it, set **Disable_Editor** to **true** (the Main Menu is enabled by default). At the Main Menu screen, you have some options to select. Read the help window to know which option does what.

Physics Editor:

In order to make a map playable with ARC ENGINE's power, you must load it with the Physics Editor. When the "Select Map" screen appears, type the desired map's ID at the given space, then hit Ok. ARC ENGINE will load and prepare your map to be edited. You can use the following controls:

Mouse Left Button: apply the selected brush as impassable

Mouse Middle Button: apply the selected brush as semi-passable

Mouse Right Button: apply the selected brush as passable

CTRL: Hold to move the Zoom rectangle

Arrows: Move the screen

You can click on the large window to apply brushes, or at the zoom window (if enabled) to apply a single pixel. Also, you can click on the following options:

Save: compress, optimize and save your map

Cancel: go back to the "Select Map" screen

Character Icon: set the physics editor to collision mode

Water Icon: set the physics editor to water mode **(disabled)**

Ice Icon: set the physics editor to ice mode **(disabled)**

Brush: change the current brush

Tile grid: enable the tile grid **(highly recommended)**

Stairs: set the physics editor to stair mode

Your map's data will be saved at the Data folder. ARC ENGINE will recognize this file and load your map automatically.

Script Commands:

The ARC ENGINE implements several new commands for events. You can call an event instance by using `self_event` (for the event itself) or `$game_map.events[id]`. Also, you can call the player instance by using `$game_player`. From this point, consider `char` as `self_event`, `$game_map.events[id]` or `$game_player`.

Each character position is given by two coordinates: `ax` and `ay`. Call:

```
x = char.ax
```

```
y = char.ay
```

Also, you can receive the char's `speed`:

```
speed_x = char.vx
```

```
speed_y = char.vy
```

You can set the char's height, width or weight:

```
char.set_height(new_height)
```

```
char.set_width(new_width)
```

```
char.set_weight(new_weight)
```

You can apply a force in the X or Y axis:

```
char.apply_xforce(fx)
```

```
char.apply_yforce(fy)
```

You can set the character's speed:

```
char.move_x(speed)
```

```
char.move_y(speed)
```

You can move the char at a given speed and a given time (in frames)

```
char.move_x(speed,time)
```

```
char.move_y(speed,time)
```

You can move the char through a distance (pixels) in a given time:

```
char.move_dx(distance,time)
```

```
char.move_dy(distance,time)
```

Comment Commands:

You can use several comment commands with events. Just put a comment with the desired command of the list below:

[width X] -> Set the width to X (in pixels)

[height X] -> Set the height to X (in pixels)

[weight X] -> Set the weight to X (in kilos)

[event_trigger X] -> Make the event sensitive to event X (ID).

[deny_gravity] -> The event will not fall

[pushable] -> The player will be able to push this event

[sticky] -> This event is sticky (useful for platforms)

[apply_friction] -> Apply friction to this event

[turtle] -> This event will move like super Mario turtles

[block_event] -> This event will block other events

[groundpound] -> This event is triggered only by groundpound

[stop_player] -> When this event gets triggered, force the player to stop

[deny_player] -> This event will deny player

[deny_up] -> Deny trigger from above

[deny_down] -> Deny trigger from below

[deny_right] -> Deny trigger from right side

[deny_left] -> Deny trigger from left side