Core functions

iman_add(str, playerID, kb_code, gp_code)

Adds an abstract key, mapps it to a selected player and a keyboard key and a gamepad button. This is how you map arbitrary string keys to physical keyboard and gamepad keys.

```
iman_add("shoot", player_id, vk_space, IMAN_BTN_A);
```

In the above example, shoot is mapped for a particular player (player_id can be one of the four – IMAN_P1, IMAN_P2, IMAN_P3, IMAN_P4 – as only four controllers are supported by GameMaker at this time.) and mapped to the spacebar on the keyboard and the A button on the gamepad. You can also use the gamemaker constant gp_face1 instead of IMAN_BTN_A as well. Check the Global Constants and Variables section below for a list of constants.

iman_pressed(key, playerID)

Returns true if mapped abstract key is pressed, either on the gamepad or the keyboard

```
has_shot = iman_pressed("shoot", player_id);
```

In the above example, has_shot will be true as soon as the player presses the spacebar or the A button on the gamepad. This will be true only once per press.

iman_released(key, playerID)

Returns true if mapped abstract key is released, either on the gamepad or the keyboard

```
has_shot = iman_released("shoot", player_id);
```

Similarly, here has_shot will be true as soon as the player releases the spacebar or the A button on the gamepad. This will be true only once per release.

iman_down(key, playerID)

Returns true if mapped abstract key is held down, either on the gamepad or the keyboard

```
is_shooting = iman_pressed("shoot", player_id);
```

Similar to the above two examples, is_shooting will be set to true as long as the spacebar or button is is pressed down. This will only be false if the user lets go of the button.

Note: If a gamepad is un/plugged, ev_user0 and ev_user1 are called for the InputMan object. One can use the iman_gp_last_lost and iman_gp_last_found vars to see which gamepad was un/plugged last.

Global constants and variables

Player ID constants

Use the player constants when using all iman_* or gp_* functions.

IMAN_P1 – Player 1

IMAN_P2 – Player 2

IMAN_P3 – Player 3

IMAN_P4 – Player 4

Gamepad Key Constants

The following constants are used primarily when you register a key using iman_add or checking for via the gamepad only functions – gp_down, gp_pressed and gp_release functions. You can use these constants for consistency or except for our custom left and right stick custom button states, you can use gamemaker constants that start with gp_* (also listed below).

IMAN_LEFT_STICK_UP – Custom, treats left stick up as a button press

IMAN_LEFT_STICK_DOWN – Custom, treats left stick down as a button press

IMAN_LEFT_STICK_LEFT – Custom, treats left stick left as a button press

IMAN_LEFT_STICK_RIGHT – Custom, treats left stick right as a button press

IMAN_RIGHT_STICK_UP – Custom, treats right stick up as a button press

IMAN_RIGHT_STICK_DOWN – Custom, treats right stick down as a button press

IMAN_RIGHT_STICK_LEFT – Custom, treats right stick left as a button press

IMAN_RIGHT_STICK_RIGHT – Custom, treats right stick right as a button press

IMAN_PAD_UP or gp_padu – Dpad Up

IMAN_PAD_DOWN or gp_padd – Dpad Down

IMAN_PAD_LEFT or gp_padl – Dpad Left

IMAN_PAD_RIGHT or gp_padr – Dpad Right

IMAN_BTN_A or gp_face1 – A button

IMAN_BTN_B or gp_face2 – B button

IMAN_BTN_X or gp_face3 – X Button

IMAN_BTN_Y or gp_face4 – Y Button

IMAN_BTN_SELECT or gp_select – Select Button

IMAN_BTN_START or gp_start – Start Button

IMAN_BTN_STICK_LEFT or gp_stickl – Left Stick Button

IMAN_BTN_STICK_RIGHT or gp_stickr – Right Stick Button

IMAN_BTN_SHOULDER_LEFT or gp_shoulderl – Left Shoulder Button

IMAN_BTN_SHOULDER_RIGHT or gp_shoulderr – Right Shoulder Button

IMAN_BTN_TRIGGER_LEFT or gp_shoulderlb – Left Trigger Button

IMAN_BTN_TRIGGER_RIGHT or gp_shoulderrb – Right Trigger Button

Keyboard Map Variable

iman_kb_map – is a globalvar, an array of ds_maps used to store the key value pairs (string key against, keyboard key constant), which is used to check key states. Please refrain from manipulating unless you know what you are doing.

Gamepad Variables

iman_gp_used – is an array and a globalvar, this checks if a gamepad for a player is in use. Check function gp_used()

iman_gp_map – is a globalvar similar to the one above, an array of ds_maps used to store the Gamepad value pairs (string key against, gamepad button constants), which is used to check button states. Please refrain from manipulating unless you know what you are doing.

iman_gp_supported – is a globalvar, which will be true if gamepads are supported

iman_gp_deadZone – value is 0.5 by default, is a globalvar, set this in the InputMan object directly when you need to change it

iman_gp_last_found – value is -1 by default, is a globalvar, stores the player id / number of the last gamepad plugged in. Use this in conjunction with ev_user1 in the InputMan object to track new gamepad connections.

iman_gp_last_lost – value is -1 by default, is a globalvar, stores the player id / number of the last unplugged gamepad. Use this in conjunction with ev_user0 in the InputMan object to track gamepad disconnections.

Gamepad – generic state and value functions

gp_used(player_id)

Checks if a gamepad was flagged as used during this step

gp_vibration_start(player_id, time - steps, vibAmount_left - 0 to 1, vibAmount_left 0 to 1)

Adds a vibration object to the room and vibrates the current player_id controller for number of steps with given value for left and right motors

gp_vibration_stop(player_id)

Stops any vibration for given player_id

gp_down(gp_code, player_id)

Returns true if a particular gamepad button mapped to the given code is held down.

gp_pressed(gp_code, player_id)

Returns true if a particular gamepad button mapped to the given code is pressed.

gp_released(gp_code, player_id)

Returns true if a particular gamepad button mapped to the given code is released.

Gamepad – left stick state and value functions

gp_left_stick_direction(player_id)

return left sick direction value between 0 and 360

gp_left_stick_distance(player_id)

return left stick distance between 0, 0 and x, y

gp_left_stick_moved(player_id)

returns true if left stick was moved

gp_left_stick_down(player_id)

returns true if there is a change in the left stick axis vertically down

gp_left_stick_left(player_id)

returns true if there is a change in the left stick axis towards left

gp_left_stick_right(player_id)

returns true if there is a change in the left stick axis towards right

gp_left_stick_up(player_id)

returns true if there is a change in the left stick axis vertically up

gp_left_stick_x(player_id)

returns the value in the x axis from the left stick

gp_left_stick_y(player_id)

returns the value in the y axis from the left stick

gp_button_stick_left(player_id)

returns true if left stick button was held down

gp_button_stick_left_pressed(player_id)

returns true if left stick button was pressed

gp_button_stick_left_released(player_id)

returns true if left stick button was released

Gamepad - right stick state and value functions

gp_right_stick_direction(player_id)

return right stick direction value between 0 and 360

gp_right_stick_distance(player_id)

return right stick distance between 0, 0 and x, y

gp_right_stick_moved(player_id)

returns true if right stick was moved

gp_right_stick_up(player_id)

returns true if there is a change in the right stick axis vertically up

gp_right_stick_down(player_id)

returns true if there is a change in the right stick axis vertically down

gp_right_stick_left(player_id)

returns true if there is a change in the right stick axis towards left

gp_right_stick_right(player_id)

returns true if there is a change in the right stick axis towards right

gp_right_stick_x(player_id)

returns the value in the x axis from the right stick

gp_right_stick_y(player_id)

returns the value in the y axis from the right stick

gp_button_stick_right(player_id)

returns true if right stick was held down

gp_button_stick_right_pressed(player_id)

returns true if right stick button was pressed

gp_button_stick_right_released(player_id)

returns true if right stick button was released

Gamepad – dpad state functions

gp_pad_up(player_id)

returns true if dpad up was held down

gp_pad_up_pressed(player_id)

returns true if dpad up was pressed

gp_pad_up_released(player_id)

returns true if dpad up was released

gp_pad_down(player_id)

returns true if dpad down is held down

gp_pad_down_pressed(player_id)

returns true if dpad down was pressed

gp_pad_down_released(player_id)

returns true if dpad down was released

gp_pad_right(player_id)

returns true if dpad right was held down

gp_pad_right_pressed(player_id)

returns true if dpad right was pressed

gp_pad_right_released(player_id)

returns true if dpad right was released

gp_pad_left(player_id)

returns true if dpad left was held down

gp_pad_left_pressed(player_id)

returns true if dpad left was pressed

gp_pad_left_released(player_id)

returns true if dpad left was released

Gamepad – face buttons

gp_button_a(player_id)

returns true if a button is held down

gp_button_a_pressed(player_id)

returns true if a button is pressed

gp_button_a_released(player_id)

returns true if a button was released

gp_button_b(player_id)

returns true if b button is held down

gp_button_b_pressed(player_id)

returns true if b button is pressed

gp_button_b_released(player_id)

returns true if b button is released

gp_button_x(player_id)

returns true if x was held down

gp_button_x_pressed(player_id)

returns true if x was pressed

gp_button_x_released(player_id)

returns true if x was released

gp_button_y(player_id)

returns true if y was held down

gp_button_y_pressed(player_id)

returns true if y was pressed

gp_button_y_released(player_id)

returns true if y was released

Gamepad – start and select buttons

gp_button_select(player_id)

returns true if select button is held down

gp_button_select_pressed(player_id)

returns true if select button is pressed

gp_button_select_released(player_id)

returns true if select button is released

gp_button_start(player_id)

returns true if start is held down

gp_button_start_pressed(player_id)

returns true if start is pressed

gp_button_start_released(player_id)

returns true if start was released

Gamepad – shoulder buttons

gp_button_shoulder_left(player_id)

returns true if left shoulder button held down

gp_button_shoulder_left_pressed(player_id)

returns true if left shoulder button is pressed

gp_button_shoulder_left_released(player_id)

returns true if left shoulder button is released

gp_button_shoulder_right(player_id)

returns true if right shoulder button is held down

gp_button_shoulder_right_pressed(player_id)

returns true if right shoulder button is pressed

gp_button_shoulder_right_released(player_id)

returns true if right shoulder button is released

Gamepad – trigger buttons

gp_trigger_left_value(player_id)

returns the value from the left analog trigger button

gp_trigger_right_value(player_id)

returns the value from the right analog trigger button

gp_button_trigger_left(player_id)

returns true if left trigger was held down

gp_button_trigger_left_pressed(player_id)

returns true if left trigger was pressed

gp_button_trigger_left_released(player_id)

returns true if left trigger was released

gp_button_trigger_right(player_id)

returns true if right trigger was held down

gp_button_trigger_right_pressed(player_id)

returns true if right trigger was pressed

gp_button_trigger_right_released(player_id)

returns true if right trigger was released