
Moonwards Docs Documentation

Moonwards Team

Jul 08, 2019

Contents:

1	What is Moonwards?	1
2	Indices and tables	3
2.1	User Interface	3
2.2	Input Mapping	5
2.3	Gamestate	7
2.4	Options	9
2.5	Namelist	11
2.6	Utils	11
2.7	Log	13
2.8	Debug	14

CHAPTER 1

What is Moonwards?

What if a group of nations was determined to colonize the Moon, and they based their decisions only on technical and economic merit?

We are building that town - . Ok, not a real town, a virtual moon with a virtual town, full of digital models of all the structures and machinery involved. Designs are absolutely technically feasible and sensible. They are based on an extensive, expansive timeline into the future, grounded in solid science and practical engineering.

- `genindex`

2.1 User Interface

2.1.1 UI Manager

Description

The `UIManager` is meant to handle all UIs without knowing what exactly is inside the UI. This manager provides a central point for all the UIs to talk to, so that there are no interfering UIs.

Features include:

- Queuing UI.
- Going back to the previous UI.
- Capturing and releasing the mouse cursor.
- Dismissing UI.
- Loading level.
- Quitting.

Method Descriptions

- void **RegisterBaseUI** (`node` node)

You can select a node in the tree to be the parent of all the UIs. The `UIManager` will handle the adding and removing of the UI. This function should be used in `_ready`.

- void **NextUI** (`string` path)

A new instance of a UI is made and added to the tree. Resource should be a path like "res://UI/menu.tscn" for example.

- **bool CanGoBack ()**

Returns true when there is a previous UI available to which the UIManager can return.

- **void Back ()**

The UIManager will go back to the previous UI if one is available. It will ignore this when the history queue is empty.

- **void DismissUI ()**

The displayed UI will be removed from the tree. If there are any future queued UIs or UIs in the history queue it will go to that.

- **void ClearUI ()**

The current UI will be removed and all queued UIs will be ignored.

- **bool RequestFocus ()**

Returns true if there currently is no UI that has focus. Use `ReleaseFocus` when done.

- **void ReleaseFocus ()**

Once focus is no longer needed it can be released, making character controls work again for example.

Example

Before a UI can be added to the UIManager a base needs to be registered. This base is the parent node of the UI. The UIManager handles the `add_child` and `remove_child`, so there is no need to do this manually.

```
var ui_resource = "res://assets/my_ui.tscn"
var added_ui = false

func _ready():
    UIManager.RegisterBaseUI(self)
```

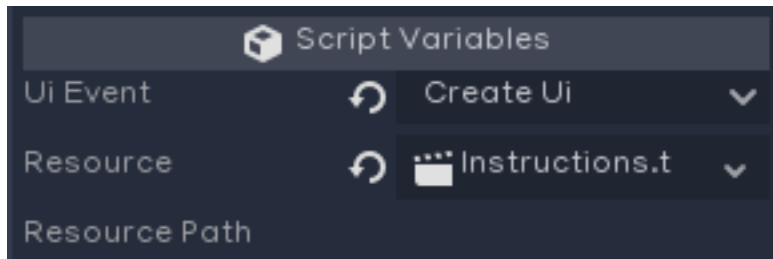
Once the base has been registered, then the UI can be added at will. In this example a menu will be toggled on and off when a key is pressed. The menu script needs to keep track of adding or removing the UI by using a boolean.

```
func _input(event):
    if event.is_action_pressed("ui_button"):
        if added_ui:
            UIManager.ClearUI()
            added_ui = false
        elif UIManager.RequestFocus():
            UIManager.SwitchUI(ui_resource)
            added_ui = true
```

2.1.2 UIElement

Description

To quickly hook up buttons and other UI elements the UIElement script can be used. This script needs to be attached to a node, and then configured using the inspector. The script uses the functions in the UIManager.



UIEvents

- **Back** Go to the previous UI.
- **Create UI** Show a new UI using either the path or the PackedScene variable.
- **Queue UI** Add a new UI to the queue, will be displayed if there is currently no UI or once Dismiss is used on the current UI.
- **Set Setting**
- **Dismiss** Remove the current UI and continue to the next UI if one is available.
- **Load Level** Use the level loader to switch to a new scene.
- **Join Server**
- **Run Locally**
- **Exit** Closes the game, returning to desktop

2.2 Input Mapping

2.2.1 Input Mapping

Methods

void	<i>load_config</i> (bool enable_next=true)
int	<i>save_to_config</i> ()
void	<i>save_all</i> ()

Constants

- `string` CONFIG_FILE

Default value = “user://inputmap.cfg”

Members

- `array` INPUT_ACTIONS

Description

This is a singleton that helps a set of keys to be saved as an input map for later use. This input map will be persistent as it is loaded each time the game is started.

You can access this singleton via the name `Input_Map` followed by one of the methods of the class.

Method Descriptions

- void **load_config** ()

Loads the file specified in `CONFIG_FILE`, if it can not find it, this function will create it.

- int **save_to_config** (string section, int key, int value)

This is a generic function to save specific information to a `ConfigFile`. Can be used for saving any value, related to any key, under any section. For input information, the section is “input”, the key is the action name and the value is the scancode of the action.

- void **save_all** ()

Saves the current Input Map into a `ConfigFile` saved in `CONFIG_FILE` that can be later accessed using `load_config`.

2.2.2 Options Component

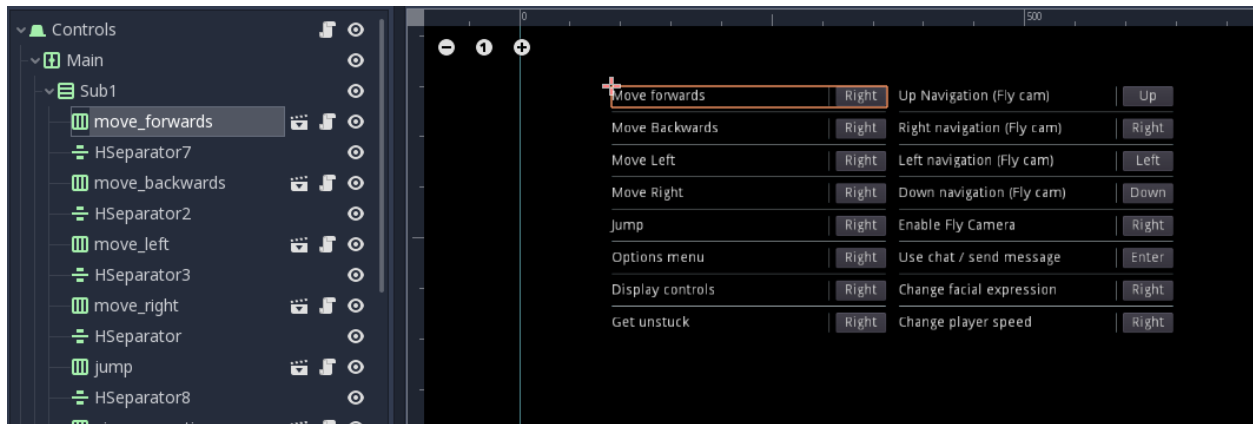
Properties

string	<i>label text</i>
--------	-------------------

Description

This button allows the user to identify the name and the currently assigned key for an action. This is a scene located at “`res://assets/UI/Menu/control_option_component.tscn`” when instantiated in another scene it is important to note the following requirements:

- The name of the node should be the same as the action that it will change, like in the following example:



- To change the text displayed in the button, change the **Label text** field in the script variables.

Properties descriptions

This is the text that will be displayed when changing settings.

2.3 Gamestate

2.3.1 Gamestate

Inherits: `Node < Object`

Category: Singleton

Brief Description

Handles multiplayer gamestate.

Methods

float	<i>get_playback_position ()</i>
AudioStreamPlayback	<i>get_stream_playback ()</i>
void	<i>play (float from_position=0.0)</i>
void	<i>seek (float to_position)</i>
void	<i>stop ()</i>

Signals

- **user_join ()**

Emits when user is fully registered.

- **user_leave ()**

Emits when a registered user leaves.

- **user_name_disconnected (`String` name)**

Emits when user is connected for chat.

- **user_name_connected (`String` name)**

Emits when user is disconnected from chat.

- **user_msg (`int` id, `string` msg)**

Emits message of id user.

- **player_id (`int` id)**

Emits id of player after establishing a connection.

- **server_up ()**

No description provided.

- **server_connected ()**

No description provided.

- **server_select ()**

Shows dialog to connect to a server or create a server.

- **network_error (`string` message)**

No description provided.

- **network_log (`string` message)**

Emit on change in server status, client status - connection, establishing connection, etc.

- **scene_change ()**

No description provided.

- **scene_change_name (name)**

No description provided.

- **scene_change_error (msg)**

No description provided.

- **loading_progress (percentage)**

No description provided.

- **loading_done ()**

- **loading_error (msg)**

No description provided.

- **player_scene ()**

No description provided.

Emit when a scene for players is detected

- **connection_failed ()**

No description provided.

- **connection_succeeded ()**

No description provided.

- **game_ended ()**

No description provided.

- **game_error (what)**

No description provided.

Description

Does something with multiplayer

Tutorials

- Should be done

Method Descriptions

- float **get_playback_position** ()

Returns the position in the AudioStream.

- AudioStreamPlayback **get_stream_playback** ()
- void **play** (float from_position=0.0)

Plays the audio from the given position 'from_position', in seconds.

- void **seek** (float to_position)

Sets the position from which audio will be played, in seconds.

- void **stop** ()

Stops the audio.

2.4 Options

2.4.1 options

Inherits: [Node](#)

Category: <FILL THIS SPACE>

Brief Description

<FILL A BRIEF DESCRIPTION HERE>

Methods

	<i>LoadUserSettings</i> ()
	<i>SafeGetColor</i> (var color_name, var default_color)
	<i>SafeGetSetting</i> (var setting_name, var default_value)
	<i>SaveUserSettings</i> ()
	<i>_ready</i> ()
	<i>del_state</i> (prop)
	<i>get</i> (category, prop = null, default=null)
	<i>get_tree_opt</i> (opt)
	<i>get_tree_options</i> (tree)
	<i>has</i> (category, prop = null)
	<i>load</i> ()
	<i>player_opt</i> (type, opt = null)
	<i>printd</i> (s)
	<i>save</i> ()
	<i>set</i> (category, value, prop = null)
	<i>set_defaults</i> ()

Signals

- **user_settings_changed** ()

!<FILL DESCRIPTION HERE>!

Description

!<FILL DESCRIPTION HERE>!

Methods Descriptions

- **LoadUserSettings** ()

!<FILL DESCRIPTION HERE>!

- **SafeGetColor** (var color_name, var default_color)

!<FILL DESCRIPTION HERE>!

- **SafeGetSetting** (var setting_name, var default_value)

!<FILL DESCRIPTION HERE>!

- **SaveUserSettings** ()

!<FILL DESCRIPTION HERE>!

- **_ready** ()

!<FILL DESCRIPTION HERE>!

- **del_state** (prop)

!<FILL DESCRIPTION HERE>!

- **get** (category, prop = null, default=null)

!<FILL DESCRIPTION HERE>!

- **get_tree_opt** (opt)

!<FILL DESCRIPTION HERE>!

- **get_tree_options** (tree)

!<FILL DESCRIPTION HERE>!

- **has** (category, prop = null)

!<FILL DESCRIPTION HERE>!

- **load** ()

!<FILL DESCRIPTION HERE>!

- **player_opt** (type, opt = null)

!<FILL DESCRIPTION HERE>!

- **printd** (s)

!<FILL DESCRIPTION HERE>!

- **save** ()

!<FILL DESCRIPTION HERE>!

- `set ( category, value, prop = null )`

!<FILL DESCRIPTION HERE>!

- `set_defaults ( )`

!<FILL DESCRIPTION HERE>!

2.5 Namelist

2.5.1 namelist

Inherits: `Node`

Category: <FILL THIS SPACE>

Brief Description

<FILL A BRIEF DESCRIPTION HERE>

Methods

	<code>_ready ()</code>
	<code>get_name ()</code>

Description

!<FILL DESCRIPTION HERE>!

Methods Descriptions

- `_ready ( )`

!<FILL DESCRIPTION HERE>!

- `get_name ( )`

!<FILL DESCRIPTION HERE>!

2.6 Utils

2.6.1 utils

Inherits: `Node`

Category: <FILL THIS SPACE>

Brief Description

<FILL A BRIEF DESCRIPTION HERE>

Methods

	<i>CreateScreenshot ()</i>
	<i>_input (var event)</i>
	<i>array_add (a, b)</i>
	<i>feature_check_server ()</i>
	<i>file_mtime (fname)</i>
	<i>get_cs_list (root)</i>
	<i>get_cs_list_cs (root)</i>
	<i>get_node_file (node)</i>
	<i>get_node_root (node)</i>
	<i>get_nodes (root, recurent=false)</i>
	<i>get_nodes_type (root, type, recurent=false)</i>
	<i>get_safe_bool (obj, propetry)</i>
	<i>get_scene ()</i>
	<i>obj_has_groups (obj, groups)</i>
	<i>obj_has_property (obj, pstr)</i>
	<i>printfd (s)</i>

Description

!<FILL DESCRIPTION HERE>!

Methods Descriptions

- **CreateScreenshot ()**

!<FILL DESCRIPTION HERE>!

- **_input (var event)**

!<FILL DESCRIPTION HERE>!

- **array_add (a, b)**

!<FILL DESCRIPTION HERE>!

- **feature_check_server ()**

!<FILL DESCRIPTION HERE>!

- **file_mtime (fname)**

!<FILL DESCRIPTION HERE>!

- **get_cs_list (root)**

!<FILL DESCRIPTION HERE>!

- **get_cs_list_cs (root)**

!<FILL DESCRIPTION HERE>!

- **get_node_file** (node)

!<FILL DESCRIPTION HERE>!

- **get_node_root** (node)

!<FILL DESCRIPTION HERE>!

- **get_nodes** (root, recurent=false)

!<FILL DESCRIPTION HERE>!

- **get_nodes_type** (root, type, recurent=false)

!<FILL DESCRIPTION HERE>!

- **get_safe_bool** (obj, propetry)

!<FILL DESCRIPTION HERE>!

- **get_scene** ()

!<FILL DESCRIPTION HERE>!

- **obj_has_groups** (obj, groups)

!<FILL DESCRIPTION HERE>!

- **obj_has_property** (obj, pstr)

!<FILL DESCRIPTION HERE>!

- **printd** (s)

!<FILL DESCRIPTION HERE>!

2.7 Log

2.7.1 log

Inherits: [Node](#)

Category: <FILL THIS SPACE>

Brief Description

<FILL A BRIEF DESCRIPTION HERE>

Methods

	<i>print_fd</i> (id, s)
	<i>test_fd</i> (id, s)

Description

!<FILL DESCRIPTION HERE>!

Methods Descriptions

- `print_fd ( id, s )`

!<FILL DESCRIPTION HERE>!

- `test_fd ( id, s )`

!<FILL DESCRIPTION HERE>!

2.8 Debug

2.8.1 debug

Inherits: `Node`

Category: <FILL THIS SPACE>

Brief Description

<FILL A BRIEF DESCRIPTION HERE>

Methods

<code>_input (event)</code>
<code>_ready ()</code>
<code>camera_ready (force=false)</code>
<code>debug_apply_options ()</code>
<code>dir_contents (path="res://")</code>
<code>e_area_lod (enable=true)</code>
<code>e_collision_shapes (enable=true)</code>
<code>features_list (enabled_only=true)</code>
<code>hide_nodes_random (probability=null)</code>
<code>hide_obj_check (root, path)</code>
<code>mouse_toggle ()</code>
<code>on_node_added (node)</code>
<code>on_node_removed (node)</code>
<code>on_scene_change ()</code>
<code>on_tree_change ()</code>
<code>print_active_cameras ()</code>
<code>print_current_players ()</code>
<code>print_groups ()</code>
<code>printd (s)</code>
<code>set_3fps (enable, value = 3)</code>
<code>set_active_camera ()</code>
<code>set_lod_manager (enable)</code>
<code>show_performance_monitor (enable)</code>
<code>tree_idle_frame ()</code>
<code>user_scene_changed ()</code>

Remote Methods

<code>test_remote_call()</code>

Description

!<FILL DESCRIPTION HERE>!

Methods Descriptions

- **_input** (event)

!<FILL DESCRIPTION HERE>!

- **_ready** ()

!<FILL DESCRIPTION HERE>!

- **camera_ready** (force=false)

!<FILL DESCRIPTION HERE>!

- **debug_apply_options** ()

!<FILL DESCRIPTION HERE>!

- **dir_contents** (path="res://")

!<FILL DESCRIPTION HERE>!

- **e_area_lod** (enable=true)

!<FILL DESCRIPTION HERE>!

- **e_collision_shapes** (enable=true)

!<FILL DESCRIPTION HERE>!

- **features_list** (enabled_only=true)

!<FILL DESCRIPTION HERE>!

- **hide_nodes_random** (probability=null)

!<FILL DESCRIPTION HERE>!

- **hide_obj_check** (root, path)

!<FILL DESCRIPTION HERE>!

- **mouse_toggle** ()

!<FILL DESCRIPTION HERE>!

- **on_node_added** (node)

!<FILL DESCRIPTION HERE>!

- **on_node_removed** (node)

!<FILL DESCRIPTION HERE>!

- **on_scene_change** ()

!<FILL DESCRIPTION HERE>!

- **on_tree_change ()**

!<FILL DESCRIPTION HERE>!

- **print_active_cameras ()**

!<FILL DESCRIPTION HERE>!

- **print_current_players ()**

!<FILL DESCRIPTION HERE>!

- **print_groups ()**

!<FILL DESCRIPTION HERE>!

- **printd (s)**

!<FILL DESCRIPTION HERE>!

- **set_3fps (enable, value = 3)**

!<FILL DESCRIPTION HERE>!

- **set_active_camera ()**

!<FILL DESCRIPTION HERE>!

- **set_lod_manager (enable)**

!<FILL DESCRIPTION HERE>!

- **show_performance_monitor (enable)**

!<FILL DESCRIPTION HERE>!

- **tree_idle_frame ()**

!<FILL DESCRIPTION HERE>!

- **user_scene_changed ()**

!<FILL DESCRIPTION HERE>!

Remote Methods Descriptions

- **test_remote_call ()**

!<FILL DESCRIPTION HERE>!