

Game Object

Game Object - Functions

Game Object - Code Examples

Bufs

Spell Book

Spell Book - Raw Functions

Spell helper

Graphics

Graphics - Functions

Graphics - Notifications

Menu Elements

Input

Geometry

Control Panel

Vectors

Vector 2

Vector 3

Libraries

Spell Prediction

Combat Forecast Library

Health Prediction Library

Unit Helper Library

Target Selector

PvP Helper Library

PvP UI Module Library

Inventory Helper

Dungeons Helper

Custom UI

Custom UI Functions

Barney's Basic Guide (With ...)

# Target Selector

## Overview

The Target Selector module gives you the tools to effectively retrieve the best targets for your logics. Its basic usage is very simple, so we encourage you to use this module in all your damage or healing-related plugins.

## Including the Module

As with all other LUA modules developed by us, you will need to Import the Target Selector module into your project. To do so, you can use the following lines:

```
1 ---@type target_selector
2 local target_selector = require("common/modules/target_selector")
```

⚠ WARNING

To access the module's functions, you must use `:` instead of `.`

For example, this code is not correct:

```
1 ---@type target_selector
2 local target_selector = require("common/modules/target_selector")
3
4 local function get_targets()
5     return target_selector.get_targets()
6 end
```

And this would be the corrected code:

```
1 ---@type target_selector
2 local target_selector = require("common/modules/target_selector")
3
4 local function get_targets()
5     return target_selector:get_targets()
6 end
```

## Functions

`get_targets(limit: integer (optional)) -> table<game_object>`

- Retrieves the table containing the **best targets possible**, according to the current Target Selector settings.
- The max number of targets it returns is **3**, but you can limit it to less by specifying the limit parameter.

`get_targets_heal(limit: integer (optional)) -> table<game_object>`

- Retrieves the table containing the **best targets to heal possible**, according to the current Target Selector settings.
- The max number of targets it returns is **3**, but you can limit it to less by specifying the limit parameter.

## Manually Modifying the Settings

Although the TS config should be good to go by default, you can manually set them inside your plugin. To do so, you have to access the target selector's menu elements and modify them. For example:

```
1 ---@type target_selector
2 local target_selector = require("common/modules/target_selector")
3
4 -- this function is a simple one, not necessarily the best one for mage fires. This is just an example.
5 local is_ts_overriden = false
6 local function override_ts_settings()
7     if is_ts_overriden then
8         return
9     end
10
11     target_selector.menu_elements.damage.weight_multiple_hits:set(true)
12     target_selector.menu_elements.damage.slider_weight_multiple_hits:set(4)
13     target_selector.menu_elements.damage.slider_weight_multiple_hits_radius:set(8)
14     target_selector.menu_elements.settings.max_range_damage:set(40)
15
16     is_ts_overriden = true
17 end
```

In the previous example, we are manually setting the weight to multiple hits, and the max range of the TS. You can do this for all the parameters.

⚠ WARNING

In general, you will never need to modify the settings. If you have to modify them, we suggest that you add a menu element to give the user the power to disable your custom TS. For example:

```
1 ---@type target_selector
2 local target_selector = require("common/modules/target_selector")
3
4 local is_ts_overriden = false
5 local function override_ts_settings()
6     if is_ts_overriden then
7         return
8     end
9
10     -- define this menu element elsewhere in your code, and render it
11     local is_override_allowed = menu_elements.ts_custom_logic_override:get_state()
12     if not is_override_allowed then
13         return
14     end
15
16     target_selector.menu_elements.damage.is_damage_enabled:set(true)
17
18     target_selector.menu_elements.damage.is_damage_enabled:set(true)
19     target_selector.menu_elements.damage.weight_multiple_hits:set(true)
20     target_selector.menu_elements.damage.slider_weight_multiple_hits:set(4)
21     target_selector.menu_elements.damage.slider_weight_multiple_hits_radius:set(8)
22     target_selector.menu_elements.settings.max_range_damage:set(40)
23
24     is_ts_overriden = true
25 end
```