Health Prediction Library

Overview

The Health Prediction module is a powerful tool that provides developers with various functions to predict incoming damage and make better decisions for defensive and healing logics. This module plays a crucial role in enhancing the adaptability and accuracy of gameplay strategies in both PvP and PvE scenarios. Below, we'll explore its core functions and how to utilize them effectively.

> TIP
> You should check User Health Pred Guide to understand what this module is about in more depth before starting to work with it.

Including the Module

Like with all other LUA modules developed by us, you will need to import the health prediction module into your project. To do so, you can just use the following lines:

```lua
---@type health_prediction
local health_pred = require("common/modules/health_prediction")
```

> WARNING
> To access the module's functions, you must use `:` instead of `.`
>
> For example, this code is not correct:

```lua
---@type health_prediction
local health_pred = require("common/modules/health_prediction")

local function get_incoming_damage_in_3_seconds(player)
    local health_pred_calculated_health = health_pred.get_incoming_damage(player, 3.0)
    return health_pred_calculated_health
end
```

> And this would be the corrected code:

```lua
---@type health_prediction
local health_pred = require("common/modules/health_prediction")

local function get_incoming_damage_in_3_seconds(player)
    local health_pred_calculated_health = health_pred:get_incoming_damage(player, 3.0)
    return health_pred_calculated_health
end
```

Functions

> NOTE
> There is only one relevant function for developers within the health prediction module. That is the `get_incoming_damage` function. You could also use the unit_helper library, which also has a function that will return the health percentage taking into account the incoming damage. This functions is `get_health_percentage_inc`.

> TIP
> Instead of using the health_prediction module, you can just include directly the unit_helper module, which already includes the functionality below.

`get_incoming_damage(target: game_object, deadline_time_in_seconds: number, is_exception?: boolean) -> number`

Retrieves the amount of incoming damage to a specified target within a given timeframe.

Code Example

Below, a complete function example to check if you should cast deffensives or not, according to health prediction or raw health.

> WARNING
> This function is using **previously defined** menu elements. The comments explain them, but you can choose to remove them or create your own menu elements to replace them. This is just a real-life example used in one of our Mythic+ plugins.

```lua
---@type health_prediction
local health_pred = require("common/modules/health_prediction")

local function should_cast_deffensive_spell_on_incoming_damage(local_player)
    -- menu element to check if we are using health pred or not (you can remove this)
    local is_using_inc_dmg_logic = menu_elements.override_min_hp_on_incoming_hp_pct:get_state()

    -- we store player hp and max hp on a local variable to avoid calling the same function multiple times (per
    local local_player_hp = local_player:get_health()
    local local_player_max_hp = local_player:get_max_health()

    -- if this is true, it means that the user decided not to use the health prediction, so we just check for p
    if not is_using_inc_dmg_logic then
        local player_current_hp_pct = local_player_hp / local_player_max_hp
        return player_current_hp_pct <= menu_elements.spell_min_hp_pct:get()
    end

    -- if this code is read, means the previous check was false, so the user decided to use health prediction

    --- get incoming damage in the next 3 seconds
    local inc_dmg_hp = health_pred:get_incoming_damage(local_player, 3.0)
    -- get our hp after all the incoming damage is received
    local hp_minus_inc_dmg = local_player - inc_dmg_hp
    -- check our future health percentage, after all the expected damage is received
    local inc_dmg_hp_pct = hp_minus_inc_dmg / local_player_max_hp

    local min_inc_dmg_hp_pct_slider_value = menu_elements.incoming_hp_pct:get()

    -- compare the previous future hp pct that we calculated to the min hp pct value set by the user
    local should_cast_deffensive = inc_dmg_hp_pct <= min_inc_dmg_hp_pct_slider_value

    if should_cast_deffensive then
        -- change spell_data.name with your spell name!
        core.log("Should Cast " .. spell_data.name .. "On Inc DMG HP PCT - - Inc Dmg: " .. tostring(inc_dmg_hp_
        return true
    end

    return false
end
```

> NOTE
> As stated before, you can swap the health_pred module for the unit_helper module, since the latter also provides the get_inc_damage functionality.

Previous
« Combat Forecast Library

Next
Unit Helper Library »