Auras

Auras are temporary or permanent effects on a unit. They can be DoTs, HoTs, buffs, debuffs or absorbs. Every unit has an aura_container scene, which has a Node type root node, and contains Node type nodes named dot_container, hot_container, buff_container, debuff_container, and absorb_container. The dot and hot containers are populated by aura_dot scenes, with hots being negative dots, the buff and debuff containers are populated by aura_buff scenes, with debuffs being negative buffs, and the absorb container is populated by aura_absorb scenes.

aura_dot
The aura_dot scene stores the spell data, source, and target of a dot or hot. In the initialize function, these are stored within the scope of the full script, and the tick timer is added as a child node. In the _ready function, the timer is started. The tick function is connected to the timeout signal of the tick timer, and triggers the combat event by calling Combat.combat_event_entrypoint, and counts the number of ticks that have occurred. If the maximum number of ticks is reached, the aura is removed by calling the remove_aura function, which stops the tick timer and calls Combat.combat_event_aura_entrypoint with remove=true to remove the scene. If the aura is reapplied while it is still active, the reinitialize function is called, which updates the spell data, stops the tick timer, resets the number of occurred ticks to 0, updates the total number of ticks and the tickrate, and restarts the tick timer.

aura_buff
The aura_buff scene stores the spell data, source, and target of a buff or debuff. In the initialize function, these are stored within the scope of the full script, and the expiration timer is added as a child node. In the _ready function, Combat.buff_application is called to apply the buff or debuff to the current stats of the target, and the expiration timer is started. When the expiration timer expires, remove_aura is called, which stops the expiration timer, removes the buff from the target's current stats, and finally remove the buff scene. If a buff is reapplied while it is already active, the reinitialize function is called, which updates the spell data, stops the expiration timer, reapplies the buff, which overwrites old buff values in case they have changed, updates the duration of the expiration timer, and restarts the expiration timer.

aura_absorb
The aura_absorb scene stores the spell data, source, target, and remaining absorb value of an absorb. In the initialize function, the spell data, source and target are stored within the scope of the full script, the total absorb value is calculated by a call to Combat.value_query, and the expiration timer is added as a child node. In the _ready function, the expiration timer is started, and the array of active absorbs on the affected unit are sorted by increasing duration with a call to sort_absorbs. When the expiration timer expires, remove_absorb is called, which sets the remaining absorb value to 0, to avoid any absorbs while the aura is being removed, removes the absorb from the affected unit's array of active absorbs, and finally remove the absorb scene. If an absorb is reapplied while it is already active, the reinitialize function is called, which updates the spell data, stops the expiration timer, calls Combat.value_query to calculate the new total absorb amount, updates the duration of the expiration timer, restarts the expiration timer, and sorts the active absorbs on the

affected target.

Revision #2
Created 2024-03-20 20:42:33 UTC by cheese
Updated 2024-03-20 21:51:46 UTC by cheese