**Script Documentation**

**Name:** *“DragBall.cs"*

**Creation Date**: *12/03/18*

**Last Alteration**: *12/03/18*

**Edited by**: *Douglas W.*

**Number of Lines:** *104*

**Description**:

This script is responsible for dragging the player object and moving it around according to the current mouse/finger position, respecting the defined limits (screen borders).

**Variables**

***Number of variables***: 9

***Current variables:***

*GameObject (1) - player; [public]*

*Vector3 (2) - mouseOrig, playerOrig; [private]*

*float (4) - leftLimit, rightLimit, topLimit, bottomLimit, playerLimit; [private]*

*bool (1) - isInsideLimit; [private]*

***Description:***

* + **[GameObject] player** - Set reference in the editor mode, the script is used in an empty gameobject (Touch) and the variable keep information of what object should be dragged along with the mouse.
  + **[Vector3]** **mouseOrig** - Store the mouse/finger position [pixel coordinates] in world coordinates through the ScreenToWorldPoint method when an touch happens. Stores the first touch position.
  + **[Vector3]** **playerOrig** - Store the player gameobject position when the first touch happens before dragging.
  + **[float] left,right,top,bottom,playerLimit** - Variables dedicated to store the screen borders position. Use the ViewportToWorldPoint method to convert a normalized space coordinate (0.0 to 1) into world coordinates. The values are added or subtracted with the best value for the player be accommodated in the screen (playerLimit).
  + **[bool] isInsideLimit** - True when the player is inside the screen limit values, false otherwise. Used to trigger what movement function the script will use.

**Methods**

***Number of methods***: 4

***Current methods***:

*[private void] - OnEnable(),*

*OnMouseDown(),*

*Update(),*

*OnMouseDrag();*

***Description:***

* + **OnEnable()** - Set soon as possible the [limit](#Limit) variables to the screen border coordinates.
  + **OnMouseDown()** - Set [mouseOrig](#mouseOrig) and [playerOrig](#playerOrig) position when a touch/click is registered.
  + **Update()** - Constantly check if the player gameobject position is outside the limits and if true, set it to the most approximate allowed position from where it leaved.
  + ***OnMouseDrag()*** *- During mouse drag moves the player to the mouse/touch position and it keeps constantly checking for each defined boundary in case the player has reach one of then, if it’s true, movement stop and can only keep going along the border.*

*If “*[*isInsideLimit*](#[bool] isInsideLimit) *= true”, then player follow finger.*

*Else, check which border it has reached.*

*Set “isInsideLimit = false”.*

*If it’s in horizontal border, keep updating y position and set x position to defined limit. Set* [*mouseOrig*](#mouseOrig) *and* [*playerOrig*](#playerOrig) *again.*

*If it’s in vertical border, keep updating x position and set y position to defined limit. Set mouseOrig and playerOrig again.*

*If it’s inside limits again, set “isInsideLimit = true”.*