**January 1st, 2018**

I am trying to implement a total overhaul of the current timing system I have created due to continuum issues. The new system will not have an Hour mark and will be similar to that of Warframes. The current code of the current time systems is as follows.

using System.Collections;

using System.Collections.Generic;

using UnityEngine.UI;

using UnityEngine;

public class TimeClass : MonoBehaviour {

private int hour;

private int minute;

private float seconds;

private int time;

public Text timer;

public float timeFactor = 1;

void Awake()

{

hour = 0;

minute = 0;

seconds = 0;

}

void LateUpdate()

{

if(GameInstructions.GAME\_MODE == Literals.GAME\_MODES.NORNAL\_GAME\_MODE)

{

UpdateCountUpTime();

}

else if(GameInstructions.GAME\_MODE == Literals.GAME\_MODES.SURVIVAL\_GAME\_MODE)

{

UpdateCountDownTime();

}

}

void Update()

{

timer.text = getTime();

}

void UpdateCountUpTime()

{

seconds += Time.deltaTime \* timeFactor;

BalanceCountUp();

}

void UpdateCountDownTime()

{

seconds -= (!GameInstructions.TIME\_UP) ? Time.deltaTime \* timeFactor : 0;

BalanceCountDown();

}

void BalanceCountUp()

{

if(seconds >= 60f)

{

minute++;

seconds = 0;

}

else if(minute >= 60)

{

hour++;

minute = 0;

}

}

void BalanceCountDown()

{

if(seconds <= 0)

{

if(minute != 0)

{

minute--;

seconds = 59f;

}

else if(seconds <= 0)

{

GameInstructions.TIME\_UP = true;

}

}

else if(minute <= 0)

{

if(hour != 0)

{

hour--;

minute = 59;

}

}

}

public int Hour

{

set { hour = value; }

get { return hour; }

}

public int Minute

{

set { minute = value; }

get { return minute; }

}

public float Seconds

{

set { seconds = value; }

get { return seconds; }

}

private string Standadize(string value)

{

int actualValue = int.Parse(value);

if(actualValue <= 9)

{

return "0" + value.ToString();

}

return value.ToString();

}

string getTime()

{

return Standadize(Minute.ToString()) + " : " + Standadize(((int)Seconds).ToString());

}

}

A total overhaul is required for extra time to be added when an enemy is killed by the player as bonus time in the survival game mode. This is rather difficult to pull off because of how this time system is designed. It is not the best practice of creating a unilateral Count up or Count down system but it tends to be the best for a linear timer.

**Other Alternative**

* The first idea I had as a solution to the problem is to get the current time in the survival game mode, process it and return it back to the time system for further processing.
* This idea would require me to remove the hour variant to avoid processing 3 digits. Removing the hour variable will not make much of a difference to will be handy incase the player goes way overboard and can’t see the hour mark.
* Now I have to decide – Overhaul? Or first idea