

Dev Journal Update

11/10/16

Nate hales

Updated Bubble Change Script

The image shows a configuration window for a script named "Bubble Change (Script)". The window is divided into two main sections: a list of options on the left and a corresponding configuration area on the right.

Left Panel (Options):

- Script
- Soup Bubble
- Fire Bubble
- Glue Bubble
- Oil Bubble
- Fire Bar**
- Oil Bar**
- Glue Bar**
- Fire Bar High**
- Oil Bar High**
- Glue Bar High**
- Energy Bar
- Drain Rate**
- Recharge Rate**
- E Change Min**
- Time Betweencharge**
- Power On
- Unlimited Energy
- Flashing Speed**
- Flames**
- Size
- Element 0**
- Element 1**
- Element 2**
- Element 3**
- Element 4**
- Element 5**
- Element 6**

Right Panel (Configuration):

- BubbleChange
- Transparent
- FlamingBubble
- GlueBubble
- OilBubble
- Fire Bar (Red bar)
- Oil Bar (Black bar)
- Glue Bar (Green bar)
- Fire Bar High (Pink bar)
- Oil Bar High (Grey bar)
- Glue Bar High (Light Green bar)
- None (Slider)
- 5
- 5
- 5
- 10
-
-
- 2
- 7
- FireComplex
- FireComplex (1)
- FireComplex (2)
- FireComplex (3)
- FireComplex (4)
- FireComplex (5)
- FireComplex (6)

Flashing energy bar for draining

```
void FixedUpdate () {  
  
    drainTracker = Time.deltaTime / 3;  
    chargeTracker = Time.deltaTime / 3;  
    flasher = Time.deltaTime / 3;  
  
    // for golden powerup.  
    if(unlimitedEnergy){  
        energyBar.value += drainRate * 10;  
    }  
    if (powerOn) {  
        if(drainTracker > timeBetweencharge){  
            energyBar.value -= drainRate; // for each second.  
            isFlashing = true;  
            drainTracker = 0;  
        }  
    }  
  
    if(isFlashing){  
        if(flasher >= flashingSpeed){  
            energyBar.colors = energyColorhigh;  
            flasher = 0.0f;  
        }else if(flasher < flashingSpeed){  
            energyBar.colors = energyColor;  
        }  
    }  
}
```

Added energyBar and bool implementation

```
if (Input.GetKeyDown (KeyCode.Alpha1) || energyBar.value == 0.0f) {
    powerOn = false;
    isFlashing = false;
    energyBar.colors = ColorBlock.defaultColorBlock;
    GetComponent<Renderer> ().material = SoupBubble;
    GetComponent<Player_Controller> ().soap = true;
    GetComponent<Player_Controller> ().fire = false;
    GetComponent<Player_Controller> ().glue = false;
    GetComponent<Player_Controller> ().oil = false;
    for (int i = 0; i < Flames.Length; i++) {
        Flames [i].SetActive (false);
    }
}

if (Input.GetKeyDown (KeyCode.Alpha2) && energyBar.value > eChangeMin && energyBar.value > eChangeMin) {
    powerOn = true;
    GetComponent<Renderer> ().material = FireBubble;
    GetComponent<Player_Controller> ().soap = false;
    GetComponent<Player_Controller> ().fire = true;
    GetComponent<Player_Controller> ().glue = false;
    GetComponent<Player_Controller> ().oil = false;
}

for (int i = 0; i < Flames.Length; i++) {
    Flames [i].SetActive (true);
}
```

Edited OnTrigger for recharging energy Bar

```
/* Author: Nathan Hales
Changes bubble based on the substance it rolls through.*/
void OnTriggerStay(Collider sub){

    if (sub.tag == "bubbleChanger") {

        #region IndividualBars

        string changeBubble = sub.GetComponent<SubstanceBox> ().substance;

        if(changeBubble == "Oil"){
            if (energyBar.value >= energyBar.maxValue && !powerOn && chargeTracker > timeBetweencharge) {
                energyColor.normalColor = oilBar;
                energyColorhigh.normalColor = oilBarHigh;
                energyBar.colors = energyColor;
                energyBar.value += rechargeRate;
                chargeTracker = 0;
                return;
            }
            return;
        }

        if(changeBubble == "Fire"){
            if (energyBar.value >= energyBar.maxValue && !powerOn && chargeTracker > timeBetweencharge) {
                energyColor.normalColor = fireBar;
                energyColorhigh.normalColor = fireBarHigh;
                energyBar.colors = energyColor;
                energyBar.value += rechargeRate;
                chargeTracker = 0;
                return;
            }return;
        }

        if (changeBubble == "Glue") {
            if (energyBar.value >= energyBar.maxValue && !powerOn && chargeTracker > timeBetweencharge) {
                energyColor.normalColor = glueBar;
                energyColorhigh.normalColor = glueBarHigh;
                energyBar.colors = energyColor;
                energyBar.value += rechargeRate;
                chargeTracker = 0;
                return;
            }return;
        }
    }

    #endregion
}
```