```
using UnityEngine;
using System.IO;
public class JSON : MonoBehaviour
{
    public static JSON instance;
    [HideInInspector]
    public config_data g;
    string path = Application.streamingAssetsPath+"/json.txt";
    void Awake()
    {
        instance = this;
        readJSON();

    }
    public void readJSON()
    {
        string h = File.ReadAllText(path);
        g = JsonUtility.FromJson<config_data>(h);
    }
}

[System.Serializable]
public class config_data
{
    public float timer_json;
    public float blinkrate_json;
    public int blinkCount_json;
    public float eyeBall_MovSpeed_json;
    public string anchor;
    public float rot_time_json;
}
```

```
using UnityEngine;
using UnityEngine.UI;
using DG.Tweening;
using System.Collections;

public class PROJ_MANAGER : MonoBehaviour
{
    public static PROJ_MANAGER instance;

    #region VAR
    public TIMER timer;
    public RawImage parent,image,image2,bg,eyeball;
    public Button btn1,btn2,btn3;
    int count;
    public string anchor;
    RectTransform uitransform;

    #endregion
    void Awake()
    {
        instance = this;
    }
    void Start()
    {
        uitransform = parent.GetComponent<RectTransform>();
        StartCoroutine(JSON.instance.g.anchor);

        timer.starttime = timer.currtime;
        count = image.transform.childCount;

        btn1.gameObject.SetActive(false);
        btn2.gameObject.SetActive(false);
        btn3.gameObject.SetActive(false);
        timer.slider.gameObject.SetActive(false);

        eyeball.gameObject.SetActive(false);
    }



    #region anchorPoints
    //-------------------------------------------------------------
    IEnumerator topMiddle()
    {
        yield return null;

        uitransform.anchorMin = new Vector2(0.5f, 1);
        uitransform.anchorMax = new Vector2(0.5f, 1);
        uitransform.pivot = new Vector2(0.5f, 1);
    }

    //------------Middle-------------------
    IEnumerator middleLeft()
    {
        yield return null;

        uitransform.anchorMin = new Vector2(0, 0.5f);
        uitransform.anchorMax = new Vector2(0, 0.5f);
        uitransform.pivot = new Vector2(0, 0.5f);
    }
```

```
IEnumerator middleRight()
{
    yield return null;

    uitransform.anchorMin = new Vector2(1, 0.5f);
    uitransform.anchorMax = new Vector2(1, 0.5f);
    uitransform.pivot = new Vector2(1, 0.5f);
}

//------------Bottom-------------------
IEnumerator bottomLeft()
{
    yield return null;

    uitransform.anchorMin = new Vector2(0, 0);
    uitransform.anchorMax = new Vector2(0, 0);
    uitransform.anchoredPosition = new Vector3(0,100,0);
    uitransform.pivot = new Vector2(0, 0);
}

IEnumerator bottomMiddle()
{
    yield return null;

    uitransform.anchorMin = new Vector2(0.5f, 0);
    uitransform.anchorMax = new Vector2(0.5f, 0);
    uitransform.pivot = new Vector2(0.5f, 0);
}

IEnumerator bottomRight()
{
    yield return null;

    uitransform.anchorMin = new Vector2(1, 0);
    uitransform.anchorMax = new Vector2(1, 0);
    uitransform.anchoredPosition = new Vector3(0,100,0);
    uitransform.pivot = new Vector2(1, 0);
}
//------------------------------------------------------------
#endregion

#region  UI
// open and close ui on mouse hover - DESCRIPTION
//------------------------------------------------------------
public void onhoverOpen()
{
    //animate canvas panel
    float g = 3;
    image2.rectTransform.DOScale(new Vector3(g,g,g),0.2f);

    // turn on btns
    btn1.gameObject.SetActive(true);
    btn2.gameObject.SetActive(true);
    btn3.gameObject.SetActive(true);
    timer.slider.gameObject.SetActive(true);

    float x = 60;
    float y = 60;
```

```
        switch (JSON.instance.g.anchor)
        {
            case "bottomLeft":
                parent.rectTransform.DOAnchorPos(new
Vector3(x,y,0),0.25f);
                break;
            case "bottomMiddle":
                parent.rectTransform.DOAnchorPos(new
Vector3(0,y,0),0.25f);
                break;
            case "bottomRight":
                parent.rectTransform.DOAnchorPos(new Vector3(-
x,y,0),0.25f);
                break;
            case "middleRight":
                parent.rectTransform.DOAnchorPos(new Vector3(-
x,0,0),0.25f);
                break;
            case "middleLeft":
                parent.rectTransform.DOAnchorPos(new
Vector3(x,0,0),0.25f);
                break;
            case "topLeft":
                parent.rectTransform.DOAnchorPos(new Vector3(x,-
y,0),0.25f);
                break;
            case "topMiddle":
                parent.rectTransform.DOAnchorPos(new Vector3(0,-
y,0),0.25f);
                break;
        }

    }
    public void onhoverClose()
    {
        //animate canvas panel
        float g = 0.5f;
        image2.rectTransform.DOScale(new Vector3(g,g,g),0.1f);

        // turn off btns
        btn1.gameObject.SetActive(false);
        btn2.gameObject.SetActive(false);
        btn3.gameObject.SetActive(false);
        timer.slider.gameObject.SetActive(false);

        switch (JSON.instance.g.anchor)
        {
            case "bottomLeft":
                parent.rectTransform.DOAnchorPos(new
Vector3(0,100,0),0.25f);
                break;
            case "bottomMiddle":
                parent.rectTransform.DOAnchorPos(new
Vector3(0,0,0),0.25f);
                break;
            case "bottomRight":
                parent.rectTransform.DOAnchorPos(new
Vector3(0,100,0),0.25f);
                break;
            case "middleRight":
```

```
                parent.rectTransform.DOAnchorPos(new
Vector3(0,0,0),0.25f);
                break;
            case "middleLeft":
                parent.rectTransform.DOAnchorPos(new
Vector3(0,0,0),0.25f);
                break;
            case "topLeft":
                parent.rectTransform.DOAnchorPos(new
Vector3(0,0,0),0.25f);
                break;
            case "topMiddle":
                parent.rectTransform.DOAnchorPos(new
Vector3(0,0,0),0.25f);
                break;

        }
    }
    //------------------------------------------------------------
    #endregion

    #region  clock function
    // resume | restart | pause  timer - DESCRIPTION
    //------------------------------------------------------------
    public void timerPause()
    {
        timer.GetComponent<TIMER>().enabled = false;
    }
    public void timerResume()
    {
        timer.GetComponent<TIMER>().enabled = true;
    }
    public void timerRestart()
    {
        timer.currtime = timer.starttime;
        timer.GetComponent<TIMER>().enabled = true;
    }
    //------------------------------------------------------------
    #endregion


}
```

```
using UnityEngine;
using UnityEngine.UI;
using DG.Tweening;
using UnityEngine.SceneManagement;
public class TIMER : MonoBehaviour
{
    #region VAR
    [Header("USER_DEFINED_VAR")]
    public GameObject path;
    public float currtime;
    //goes from transparent to full black - (seconds)
    //number of times the transition takes place -(must be a whole number
and must be even)
    public int transitionCount;
    [Space(10)]
    [Header("UI")]
    public Text displaytext;
    public Slider slider;
    bool a ;

    [HideInInspector]
    public  float starttime;
    [HideInInspector]
    public  bool has_timer_reached_zero;
    #endregion

    void Start()
    {
        currtime = JSON.instance.g.timer_json;
        starttime = currtime;
        has_timer_reached_zero = false;
        slider.maxValue = currtime;
        slider.value = currtime;
        a = false;
    }

    void Update()
    {
        if(currtime>0)
        {
            currtime-=Time.deltaTime;
            updatetimer(currtime);
        }
        else if(currtime <= 0)
        {
            has_timer_reached_zero = true;
            if(a==false)
                blinkFunction();
            // return;
        }
        //Debug.Log("finished");
    }

    #region blinker
    //-----------------------------------------------------------------
-
    void blinkFunction()
    {
        RawImage r =
PROJ_MANAGER.instance.bg.gameObject.GetComponent<RawImage>();
```

```
        Tween t = r.DOFade(1,JSON.instance.g.blinkrate_json);
        t.OnComplete(()=> r.DOFade(0,JSON.instance.g.blinkrate_json));

        Sequence s = DOTween.Sequence();
        s.Append(t);
        s.SetLoops(JSON.instance.g.blinkCount_json*2,LoopType.Yoyo);
        Tween h =  s.OnComplete(()=>eyeball_rot_func_turn_on());
        a=true;
    }
    //-----------------------------------------------------------------------
-
    #endregion

    #region countdown timer
    //-----------------------------------------------------------------------
-
    void updatetimer(float currtime)
    {
        currtime += 1;
        float min = Mathf.FloorToInt(currtime/60);
        float sec = Mathf.FloorToInt(currtime%60);
        slider.value = currtime;
        displaytext.text = string.Format("{0:00}:{1:00}",min,sec);
    }
    //-----------------------------------------------------------------------
-
    #endregion

    void eyeball_rot_func_turn_on()
    {
        PROJ_MANAGER.instance.eyeball.gameObject.SetActive(true);
        path.gameObject.transform.DOLocalRotate(new
Vector3(10,10,360),JSON.instance.g.rot_time_json,RotateMode.FastBeyond360
);

PROJ_MANAGER.instance.eyeball.gameObject.GetComponent<PathCreation.Exampl
es.PathFollower>().enabled = true;

Invoke("eyeball_rot_func_turn_off",JSON.instance.g.eyeBall_MovSpeed_json)
;
    }
    void eyeball_rot_func_turn_off()
    {

PROJ_MANAGER.instance.eyeball.gameObject.GetComponent<PathCreation.Exampl
es.PathFollower>().enabled = false;
        PROJ_MANAGER.instance.eyeball.gameObject.SetActive(false);
    }
    public void Reset()
    {
        DOTween.Clear();
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }
}
```