DRAG AND DROP CODE

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;

public class DragDrop : MonoBehaviour, IBeginDragHandler, IEndDragHandler, IDragHandler,
IDropHandler, IPointerDownHandler
{
    [SerializeField] private Canvas canvas;

    private RectTransform rectTransform;
    private CanvasGroup c;
    public bool droppedOnSlot;
    private Vector3 defaultPosition;

    void Start()
    {
        defaultPosition= GetComponent<RectTransform>().localPosition;
    }

    private void Awake()
    {
        rectTransform = GetComponent<RectTransform>();
        c = GetComponent<CanvasGroup>();
    }

    public void OnBeginDrag(PointerEventData eventData)
    {
        Debug.Log("OnBeginDrag");
        c.blocksRaycasts = false;
        droppedOnSlot = false;
    }

    public void OnDrag(PointerEventData eventData)
    {
        Debug.Log("OnDrag");
        rectTransform.anchoredPosition += eventData.delta / canvas.scaleFactor;
    }

    public void OnEndDrag(PointerEventData eventData)
    {
        Debug.Log("OnEndDrag");
```

```
        c.blocksRaycasts = true;

       if (droppedOnSlot)
        {
            // Update the default position value to the new position.
            defaultPosition = this.rectTransform.localPosition;
        }
        else // The item was NOT dropped onto a slot.
        {
            // Set the items position back to starting default value;
            this.rectTransform.localPosition = defaultPosition;
        }
    }

    public void OnPointerDown(PointerEventData eventData)
    {
        Debug.Log("OnPointerDown");
    }

    void IDropHandler.OnDrop(PointerEventData eventData)
    {
        Debug.Log("OnDrop");
    }
}
```

ITEM SLOT CODE

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;
using UnityEngine.GameFoundation;
using UnityEngine.GameFoundation.DefaultLayers;
using UnityEngine.Promise;
using UnityEngine.UI;

public class ItemSlot : MonoBehaviour, IDropHandler {

    public void OnDrop(PointerEventData eventData)
    {
        Debug.Log("OnDrop");
        if (eventData.pointerDrag != null)
        {
```

```
        eventData.pointerDrag.GetComponent<RectTransform>().anchoredPosition =
GetComponent<RectTransform>().anchoredPosition;
            // Set the has dropped on slot bool to true.
        eventData.pointerDrag.GetComponent<DragDrop>().droppedOnSlot = true;

        // Use the key you've used in the previous tutorial.
        const string definitionKey = "coins";

        // Finding a currency takes a non-null string parameter.
        Currency definition = GameFoundationSdk.catalog.Find<Currency>(definitionKey);

        long balance = GameFoundationSdk.wallet.Get(definition);

        // Set replaces the current balance by a new one.
        bool success = GameFoundationSdk.wallet.Set(definition, balance + 1);
      }
   }
}
```

CHANGE SCENE CODE

```
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class ChangeScene : MonoBehaviour
{

        public void ReturnHome()
        {
                SceneManager.LoadScene("HomePage");
        }

    public void OptionPage()
        {
                SceneManager.LoadScene("Settings page");
        }

        public void Home()
        {
                SceneManager.LoadScene("HomePage");
        }

        public void Calendar()
```

```
    {
        SceneManager.LoadScene("Calendar");
    }

    public void SortTrash()
    {
        SceneManager.LoadScene("testscene");
    }

    public void Help()
    {
        SceneManager.LoadScene("Help");
    }

    public void Rewards()
    {
        SceneManager.LoadScene("RewardsPage");
    }

    public void Game()
    {
        SceneManager.LoadScene("Game");
    }

    public void Level2()
    {
        SceneManager.LoadScene("Game2");
    }

    public void Level3()
    {
        SceneManager.LoadScene("Game3");
    }
}
```

COIN SCRIPT CODE

```
using System;
using System.Collections;
using UnityEngine;
using UnityEngine.GameFoundation;
using UnityEngine.GameFoundation.DefaultLayers;
using UnityEngine.Promise;
using UnityEngine.UI;
```

```
public class CoinScript : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        // Use the key you've used in the previous tutorial.
        const string definitionKey = "coins";

        // Finding a currency takes a non-null string parameter.
        Currency definition = GameFoundationSdk.catalog.Find<Currency>(definitionKey);

        // Make sure you retrieved a valid currency.
        if (definition is null)
        {
            Debug.Log($"Definition {definitionKey} not found");
            return;
        }

        Debug.Log($"Definition {definition.key} ({definition.type}) '{definition.displayName}' found.");

        long balance = GameFoundationSdk.wallet.Get(definition);

        Text txtMy = GameObject.Find("Canvas/CoinCount").GetComponent<Text>();
        txtMy.text = balance.ToString();

    }

    // Update is called once per frame
    void Update()
    {
        // Use the key you've used in the previous tutorial.
        const string definitionKey = "coins";

        // Finding a currency takes a non-null string parameter.
        Currency definition = GameFoundationSdk.catalog.Find<Currency>(definitionKey);

        long balance = GameFoundationSdk.wallet.Get(definition);

        Text txtMy = GameObject.Find("Canvas/CoinCount").GetComponent<Text>();
        txtMy.text = balance.ToString();
    }
}
```

NEW BEHAVIOUR SCRIPT

```
using System.Collections;
using System.Collections.Generic;
using System.Diagnostics;
using UnityEngine;

public class NewBehaviourScript : MonoBehaviour
   {
      // Start is called before the first frame update
      void Awake()
      {
         if (MusicInit.music==false)
         {
            DontDestroyOnLoad(transform.gameObject);
            MusicInit.music = true;
         }

      }

   }
```