using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class DjShow : MonoBehaviour {

[SerializeField]

Pic \_itemPrefab;

[SerializeField]

Transform \_canvas;

[SerializeField]

List<Button> \_djl;

/// <summary>

/// 技能点点亮图片

/// </summary>

public Sprite s;

/// <summary>

/// 道具解锁图片

/// </summary>

public Sprite s1;

//技能点个数图片

public Text \_skillpoint;

//定义列数

public static int colunm\_count = 2;

// 道具图片列表

List<Pic> \_itemPic;

static float x;

void Start()

{

KeepData saveData = SaveData.\_Sav.GetSaveData ();

Itemsave s = GameObject.FindObjectOfType<Itemsave> ();

//背包道具栏是否显示解锁

for (int q = 0; q < saveData.\_IsUclockItem.Count; q++)

{if (saveData.\_IsUclockItem[q])

{

\_djl [q].image.sprite = s1;

}

}

//道具栏是否显示已使用道具

for (int q = 0; q < saveData.\_ItemId.Count; q++) {

if (saveData.\_HaveItemOn [q] == saveData.\_ItemId [q]) {

\_djl [q].image.sprite = s.\_item [saveData.\_ItemId [q]];

}

}

\_itemPic = new List<Pic> (saveData.\_HaveItemId.Count);

for (int i = 0; i < saveData.\_HaveItemId.Count; i++) {

Pic item = GameObject.Instantiate (\_itemPrefab);

item.\_Id = saveData.\_HaveItemId [i];

//更新技能点

Gx (item);

item.GetComponentInChildren<Text> ().text = saveData.\_HaveItemName [saveData.\_HaveItemId [i]];

item.transform.SetParent (\_canvas);

x = -196 \* (Mathf.Pow (-1, (i % colunm\_count)));

saveData.\_Slidery = 88 - 250 \* (i / colunm\_count);

SaveData.\_Sav.SaveGameData ();

item.transform.GetComponent<RectTransform> ().anchoredPosition = new Vector2 (x, saveData.\_Slidery);

//已解锁道具个数大于可装备道具个数

item.\_Use.onClick.AddListener (delegate() {

//道具装备个数小于道具解锁个数

if (saveData.\_Djnum<saveData.\_IsUclockItemNum){

if(item.\_Id<10){

Click (item.\_Id, item);

saveData.\_Djnum++;

SaveData.\_Sav.SaveGameData();

}

}

});

// 把道具图片加入列表

\_itemPic.Add (item);

for (int t = 0; t < \_itemPic.Count; t++)

{

Debug.Log (\_itemPic.Count);

}

for(int t=0;t<item.\_Skill.Count;t++)

{

SkillJs (t, item,item.\_Id);

}

}

for(int i = 0; i < saveData.\_HaveItemOn.Count; i++){

int index = i;

\_djl [i].onClick.AddListener(delegate() {

DropItem(index);

saveData.\_Djnum--;

SaveData.\_Sav.SaveGameData();

});

// 装备栏为空处理

if (saveData.\_HaveItemOn [i] == -1) {

\_djl[i].image.sprite = null;

}

// 有装备处理

for (int j = 0; j < 10; j++) {

if (saveData.\_HaveItemOn [i] == j) {

\_djl [i].image.sprite = s.\_item [j];

\_itemPic [saveData.\_HaveItemOn [i]].gameObject.SetActive (false);

}

}

}

\_skillpoint.text = saveData.\_Skill.ToString ();

}

/// <summary>

/// 动态更新道具列表

/// </summary>

/// <param name="item">Item.</param>

/\*void UpdateShowItem(Pic item)

{ KeepData saveData = SaveData.\_Sav.GetSaveData ();

for ( int k = 0; k <saveData.\_HaveItemId.Count; k++)

{

Gx (item);

item.transform.SetParent (\_canvas);

\_itemPic.Add (item);

for(int t=0;t<item.\_Skill.Count;t++)

{

SkillJs (t, item,item.\_Id);

}

}

SaveData.\_Sav.SaveGameData ();

}\*/

/// <summary>

/// 使用道具

/// </summary>

/// <param name="i">The index.</param>

/// <param name="item">Item.</param>

void Click(int i,Pic item)

{

KeepData saveData = SaveData.\_Sav.GetSaveData ();

Itemsave s = GameObject.FindObjectOfType<Itemsave> ();

//saveData.\_HaveItemId.Remove(i);

saveData.\_ItemId.Add (i);

item.gameObject.SetActive (false);

//道具栏显示道具

if (i < 10)

{

saveData.\_HaveItemOn [saveData.\_Djnum] = i;

SaveData.\_Sav.SaveGameData ();

if (saveData.\_HaveItemOn [saveData.\_Djnum] == i) {

\_djl [saveData.\_Djnum].image.sprite = s.\_item [i];

}

}

}

/// <summary>

/// 技能加点

/// </summary>

/// <param name="t">T.</param>

/// <param name="item">Item.</param>

/// <param name="id">Identifier.</param>

void SkillJs(int t,Pic item,int id)

{

KeepData saveData = SaveData.\_Sav.GetSaveData ();

item.\_Skill [t].onClick.AddListener (delegate {

if(saveData.\_IsUclockSkill[id].\_a[t]==false){

if(saveData.\_Skill>=1){

saveData.\_Skill--;

SaveData.\_Sav.SaveGameData();

\_skillpoint.text = saveData.\_Skill.ToString ();

// 遍历该道具所有等级，解锁一个等级

for(sbyte i = 0; i < saveData.\_IsUclockSkill[id].\_a.Count; i++){

// 已经解锁，继续循环

if(saveData.\_IsUclockSkill[id].\_a[i])

continue;

// 未解锁则解锁并退出

saveData.\_IsUclockSkill[id].\_a[i]=true;

SaveData.\_Sav.SaveGameData();

if(saveData.\_IsUclockSkill[id].\_a[i])

{

item.\_Skill[i].image.sprite=s;

}

// 更新所有道具图片

UpdateAllItemPic();

break;

}

}

}});

}

/// <summary>

/// 更新所有道具图片

/// </summary>

void UpdateAllItemPic(){

KeepData saveData = SaveData.\_Sav.GetSaveData ();

for(int i = 0; i <\_itemPic.Count; i++){

Gx (\_itemPic[i]);

}

}

/// <summary>

/// 更新一个道具上所有技能点

/// </summary>

/// <param name="item">Item.</param>

void Gx(Pic item)

{ KeepData saveData = SaveData.\_Sav.GetSaveData ();

for (int t = 0; t < item.\_Skill.Count; t++)

{ if(saveData.\_IsUclockSkill[item.\_Id].\_a[t])

item.\_Skill[t].image.sprite=s;

}

}

/// <summary>

/// 点击道具栏，卸下道具

/// </summary>

/// <param name="q">Q.</param>

/\*void ReturnItem(int q,Pic item)

{

KeepData saveData = SaveData.\_Sav.GetSaveData ();

Itemsave s = GameObject.FindObjectOfType<Itemsave> ();

\_djl [q].onClick.AddListener (delegate {

if(saveData.\_HaveItemOn[q]!=-1){

saveData.\_ItemId.Remove(q);

saveData.\_HaveItemId.Add(saveData.\_HaveItemOn[q]);

\_itemPic[q].gameObject.SetActive(true);

saveData.\_HaveItemOn [q] = -1;

SaveData.\_Sav.SaveGameData ();

if (saveData.\_HaveItemOn [q] == -1) {

\_djl [q].image.sprite = s1;

}

saveData.\_Djnum--;

SaveData.\_Sav.SaveGameData ();

}

});

}

\*/

/// <summary>

/// 卸下该栏装备

/// </summary>

/// <param name="i">The index.</param>

void DropItem(int i){

// 卸下空道具，退出

if (\_djl [i].image.sprite == null) {

return;

}

KeepData saveData = SaveData.\_Sav.GetSaveData();

// 卸下道具，图片为空

//saveData.\_HaveItemId.Add(saveData.\_HaveItemOn[i]);

\_itemPic[saveData.\_HaveItemOn [i]].gameObject.SetActive(true);

saveData.\_ItemId.Remove (saveData.\_HaveItemOn [i]);

saveData.\_HaveItemOn [i] = -1;

SaveData.\_Sav.SaveGameData ();

\_djl [i].image.sprite = null;

// 更新道具栏

}

}