Mini Puzzle Game Full Template

User Manual 1.15

Introduction

First thanks your purchase of the product.
This is a casual puzzle Pack of a lot of mini game collection. You will play the role of landlord Yan and have a series of interesting and touching stories with your beautiful new neighbor.
It's a game of fairly high degree of completion.
You can not only reskin it very easily but also can create new levels of your own.
In my following tutorials. I would not only teach you how to do these job.
The game template is completed designed for indie developers. The pictures can be created from 3d model or 2d illustrations very conveniently.
So don’ t worry you maybe not able to reskin yourself without a profession artist. I would give you advices on the end of the manual to help you create illustration with zero cost yourself.
Attention: The origin game resources is not allowed to publish directly without any change. This will not only raise legal issues but also won’ t help for making money.

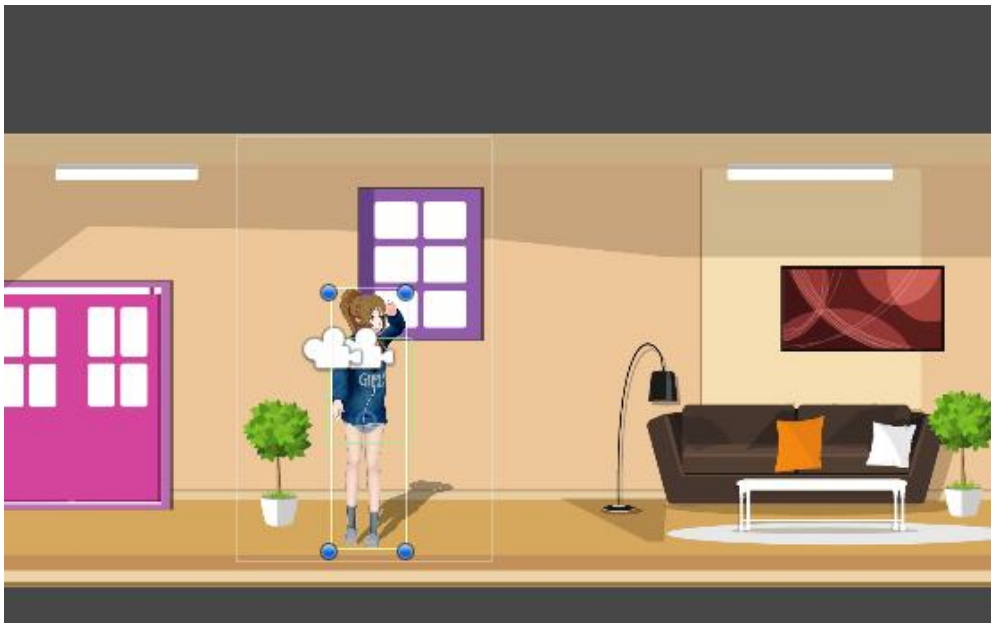
How to start the game

Use at least unity of version 2020.1.12 to open the project.
In the asset folder,find Template/game/_levels folder.
Open any of the level scene files to test of start the game.
If you want to start the game from start. Select the MainMenu file and double click on it.
In the unity menu,in file->build settings,make sure the MainMenu level was listed on the top.

How to reskin

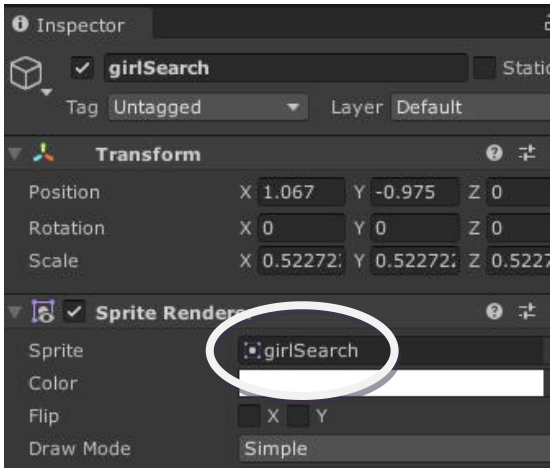
Here is a most easy way to find the picture were it located.

We use level1 as an example.



Here in the Scene Tab. We’ ve already selected the girl in the scene.

We will see inspectors like this.

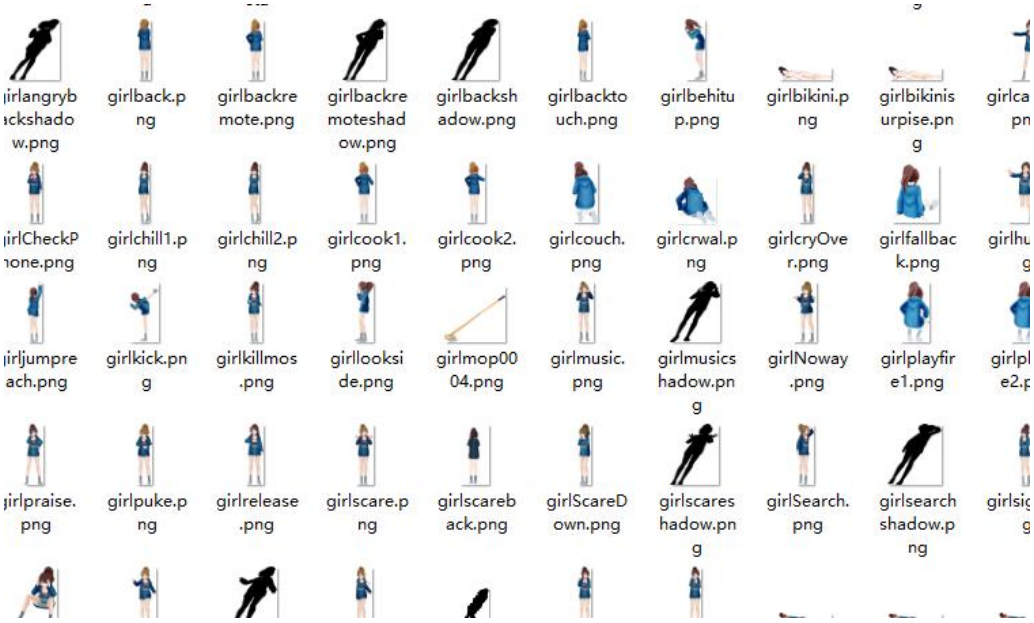


In the sprite Renderer section. We click girlsearch once.

So, now in the project panel, we found its related picture.

Right click the picture I circled upon.

And select show in explorer



Then you would found the picture on your hardware.

Just overwrite this pictures with your own to finish the reskin.

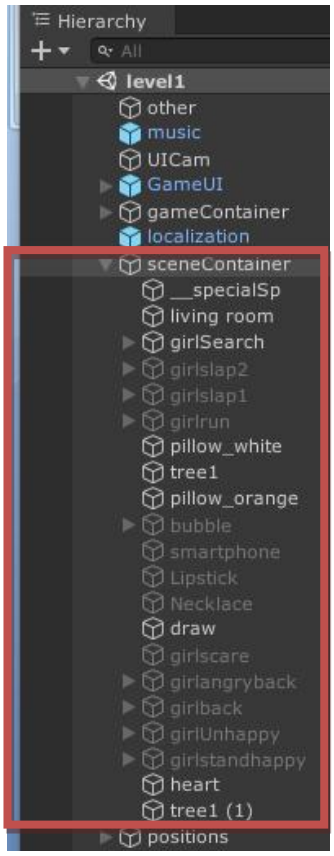
Things should be mentions

To overwrite a picture means you copy your new generated picture and paste to overwrite the origin in the game folder. Do not delete and move a picture with the same name into its origin place,that won’t work.

If you uses picture larger or smaller then the origin one.You should test the effect like recreate its collider or adjust their position.

Make your own levels

The best way to create a new level is duplicate an already existed level file from _levels folder and write your own game logic script.
Things is not that difficult if you got basic C# skill.
Here I’ d explain details of each element in a scene file as an example.
Let’ s take a look the previous level like level1.
In Assets/helpyou/_levels we double click the file level1
In level1,we expand its hierarchy and see there are many items in the



In the red frames, there are all the game elements on the scene.
Do not change the name of __specialSp,it is necessary for the game framework.
Here we see many gameobjects like girlslap2,girlrun were greyed.
This because this pictures only showed when certain events triggered.
For example,when this level starts.The initial state of the girl is standing and searching her missing items. So the girlsearch is the only activated gameobject of the character.
But when you touch the girl directly. The girlslap1gameobject would be triggered and the girlSearch gameobject would be deactivated(hide). This looks like you touching the girl and make her annoying.
When we select the girlsearch gameobject.

We would see there is a green frame above the picture.In inspector, it is a boxcollider2d component. You can set it by yourself. This is the touching area where would trigger the events.

About how to make the item can be interactive, see the section of interactive items.

The location mark

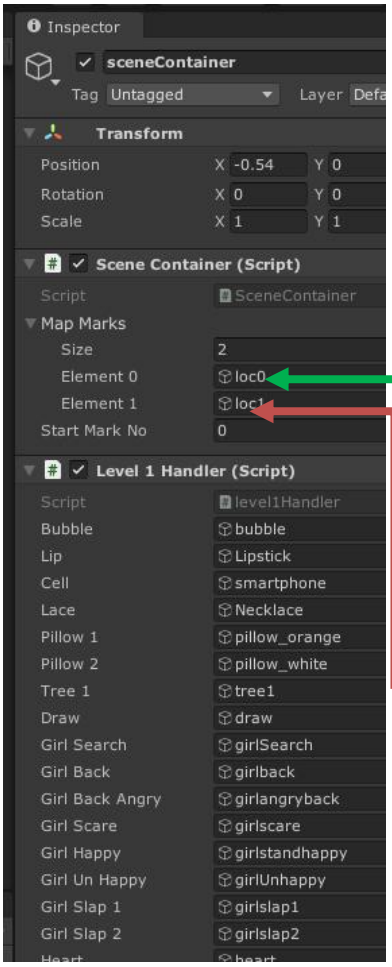
Let’ s check the inspector of sceneContainer.

Here we see there are 2 scripts attached on the sceneContainer gameobject.

We talk about the first script first.

The sceneContainer gameobject got a scenecontainer script already. This script deal with some frame jobs so it is necessary.

We see it exposed an array named Map Marks



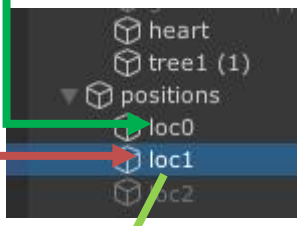
This is a very import things.

The game got a cool feature is that it can separate on large scene into several part by switch scenes.

We use arrows to switch between these scenes to find clues or doing the interactive.

These arrows would generated by the setting of map marks.

Lets take a look back to the hierarchy.



here we see there are 3 gameobject name from loc0-lock2.But 1 is deactivated. They are location marks. So this means in this level,we sperate the scene into 2 parts because we only uses 2 location marks.

Location mark is an empty gameobject without any script,it’ s only usage is mark the x postion.

Here we see the loc1 gameobject were put in this place in the scene . If you start the game,then touch the right arrow.

We will scene the game camera pointed at the position of were loc1 located and a left arrow appeared automatically.

You can set the map Mark size to any number and assign location gameobject with the same counts.

Game Logics by Scripts

Let’ s open Level1Handler first.

```
public GameObject bubble;
public GameObject lip, cell, lace;
public GameObject pillow1, pillow2, tree1;
public GameObject draw;
public GameObject girlSearch, girlBack, girlBackAngry, girlScare, girlHappy, girlUnHappy,girlSlap1,girlSlap2;
public GameObject heart;
public GameObject girlRun;
```

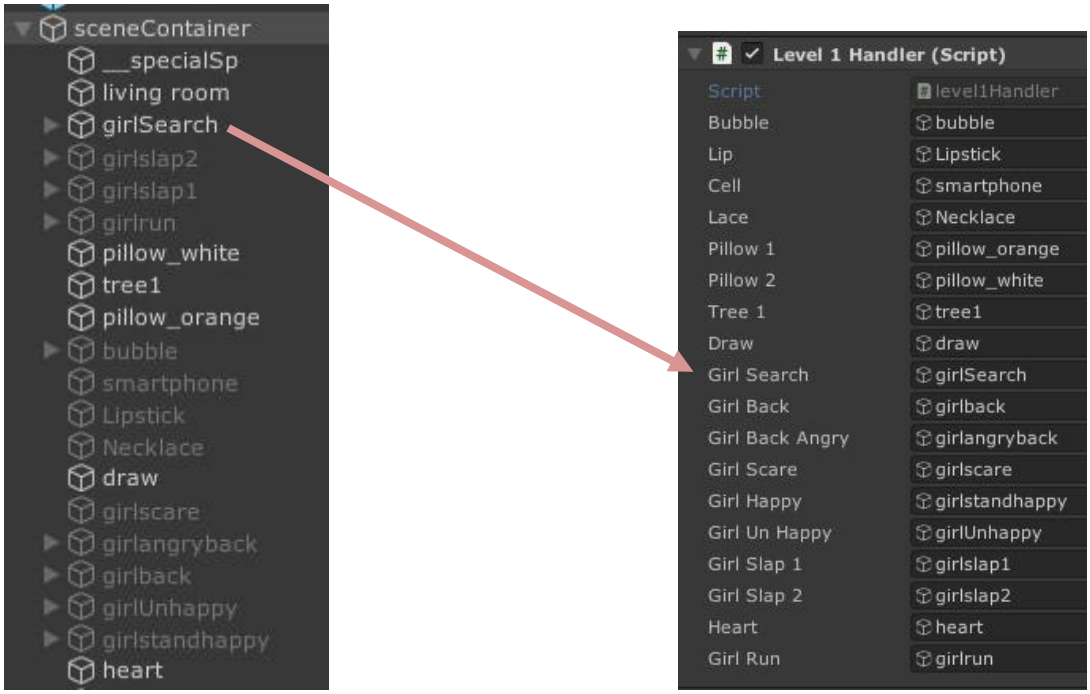
On the top of the script, We see there are many public variables being declared.

These where we need to refer in the game, so we must assign it.

There is a away to assign a gameobject to a script is using script like

GameObject g = GameObject.Find(”gameobject name”);

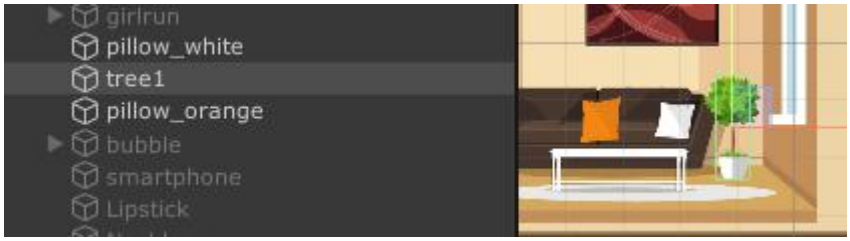
But in level we assign it by manually drag gameobject from hierarchy into the inspector.



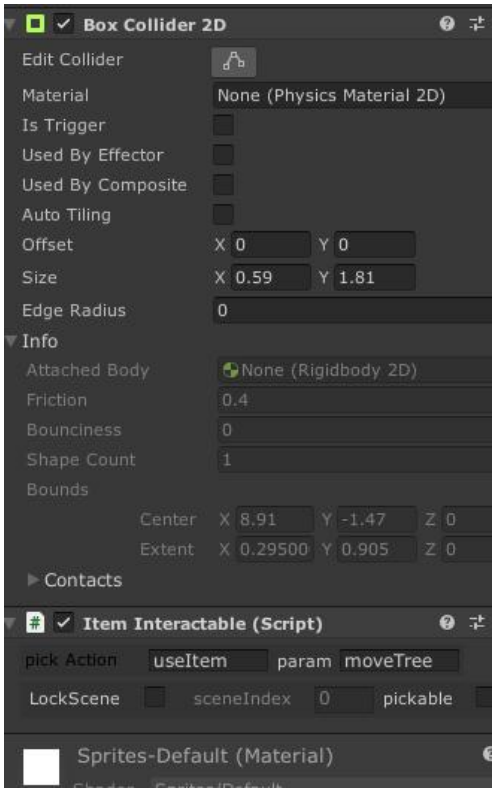
Attenion, since we improve the game making flowing during our work. The way of assign a gameobject can be more efficient in later levels. We would talk it later.

After finish all the assignment, we can use the gameobject in the script now.

First we select a gameobject such as the tree on the very right of the scene.



Then let’ s see its inspector.



The most important part of an interactive object requires 2 things.

First is a collider. Here we see we’ ve used a boxcollider2d component.

The other is a script named ItemInteractable

This script got several sections to fill.

The pick action means when you can touch this gameobject.

A message would be send to sceneContainer.

So,do not change the place or name of sceneContainer.

Here we give it a string name useItem.

This mean the message would try to find a function named useItem from all the scripts under sceneContainer.

Let’ s back to the Level1Handler script.

```
public void useItem(string param)
{
    if (GameData.instance.isLock) return;
    switch (param)
    {
        case "touchMe":
            if (!meTouched)
            {
                meTouched = true;

                girlSearch.SetActive(false);
                StopCoroutine("loop");
                bubble.SetActive(false);
                girlSlap1.SetActive(true);
```

Here we see there is a function named useItem with a parameter required.

So this is what we need to give in the param section of ItemInteractable component

We use moveTree param for this tree gameobject.

So,when you touch the tree, in the Level1Handler script.

These scripts would be called.

```
case "moveTree":
    if (!treeMoved)
    {
        GameData.instance.isLock = true;
        treeMoved = true;
        lace.SetActive(true);
        tree1.transform.DOMoveX(tree1.transform.position.x + .5f,
        { GameData.instance.isLock = false; });
    }
    break;
```

Of course to read these scripts require C# skills. But they are just basic logics and very easy to understand.

There are some variables like treeMoved being used in this part of the script.

This variable uses for the reason of avoiding unnecessary multiple touches to a same object.

These limitation is very using for to avoiding bugs. Do not forget to use as many as you can when you try to write your only logic.

The other import thing in this part of scripts is

GameData.instance.isLock = false;

This phase can lock the game to forbidden all the operations which may cause bugs during a game. Always use this when you begin to play an animation clip or some time when you don't want your player to do other operation for a while. Do not forgot to set the isLock back to true when you finish your demonstration.

Useful scripts

Play game sound effect: `GameManager.instance.playSfx("soundName");`
Play game music: `GameManager.instance.playMusic("musicName");`
Stop all sound effect: `GameManager.getInstance().stopAllSFX();`
Stop all music: `GameManager.instance.stopBGMusic();`
You wins: `GameData.instance.main.gameWin();`
You failed: `GameData.instance.main.gameFailed();`
Lock the game: `GameData.instance.isLock = true;`

Useful Functions

We see each level got a levelXHandler(x is level index)script on the sceneContainer gameobject.
There are many scripts maybe easy to reuse,here are some example.

Sometimes we need to make some delay for a next action. You can use the function as following:

```
StartCoroutine(Util.DelayToInvokeDo(() => {
    //do some thing
}, .4f));
```

The .4f means the scripts you write in the do something section would execute after 0.4 seconds.

Sometimes we need to switch the scene camera by script. You can find functions like below in some levelHandler scripts.

```
  void changeScenePos(int index)
    {

GameObject.Find("sceneContainer").GetComponent<SceneContainer>().manualScene(index);
    }
```

Call this function by use

```
changeScenePos(3);
```

This would automatically move the camera to the third location mark.

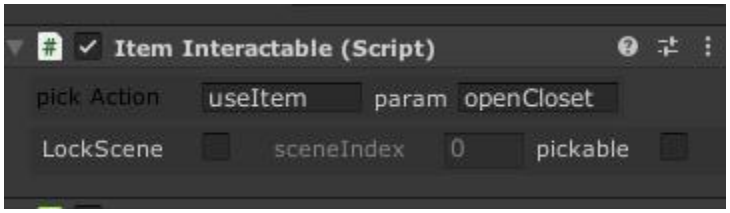
Interactive items

In order interactive with the items on the scene. We need to add a collider first.

As it is a 2d game, we can use boxcollider2d, circleCollider2d or polyCollider2d to design your touchable areas.

The other thing is the script componet.

Add the script named Iteminteractable to the gameobject which you need to interactive with the player.

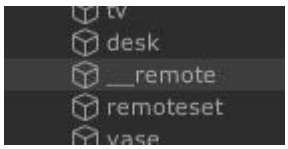


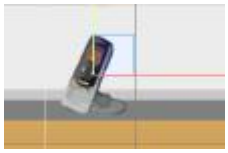
There pick Action and the param section tell the game which function would be called after you touch the gameobject.

This explained previous here

Let's not talk about lockscene checkbox but talk about the pickable first.

For example. Let open level4 and select the __remote gameobject on the hierarchy.

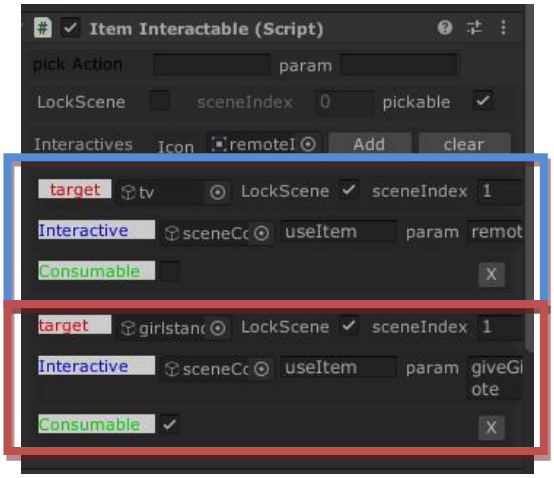




This remote is surely an item which can be interactive with. So let's see its inspector.

First we found the pickable is being checked. This uses for an item which can be pickup. If an object(such as a charactor)being touched, it need not to be pick up, so its pickable checkbox were be checked on.

If it was checked on, we must assign an Icon to the Icon section. Only after that can we continue our job.



Here,it looks a bit complicate but don't worry.

Actually there are 2 part of this interactive setting. I have framed them with different colors.

This because we can interactive with 2 different objective with this remote.

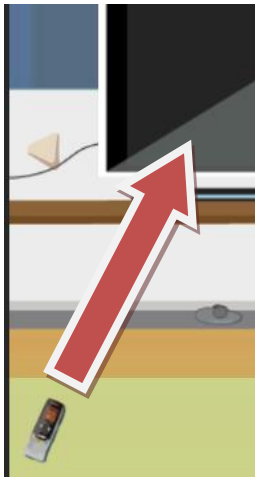
In this level the remote can interactive with both the TV and the girl

You can create as many as interactive by click the Add button on this script editor

So let just talk about one interactive in the blue frame.

The first is target

Here we assign the tv gameobject into this section. Which means when we drag the remote which being picked up already to the tv, things would be happen.



But what would be happen? Now I think it is quite clear for you if you have already read the previous tutorial.

The game would call a function useItem which in a script attached on the sceneContainer gameobject

And the function got a parameter string named remoteOnTv

We take a look to the level4Hanler.cs(it was attached on the sceneContainer gameobject)

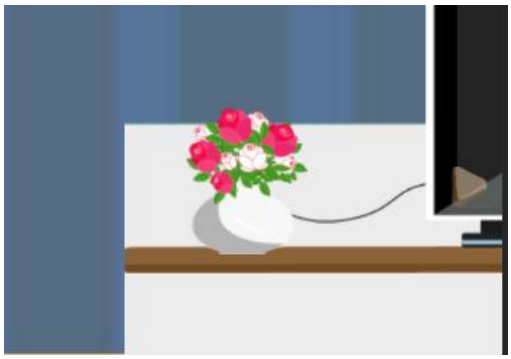
```
public void useItem(string param)
{
    if (GameData.instance.isLock) return;
    switch (param)
    {
        case "remoteOnTv":
            if (plugIsSet)
            {
                GameData.instance.isLock = tru
                if (TvIsFixed)
                {
```

The script surely is exist.

The next is the lockScene checkbox.

The LockScene checkbox means whether you must interactive the item in a certain location mark.

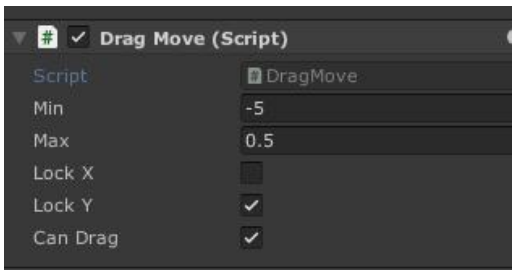
For example, In level 4, we can see the tv visible when the camera switched to left.



But if you drag on remote the TV in this place. Nothing will happened.
This is because here we've checked on the LockScene checkbox and set the sceneIndex to 1
This sceneIndex is exactly the location mark which we mentioned before.
The left place mean you are at location index 0, it is not index 1 so this interactive only can be triggered when you are at the middle location of the scene.
The last thing is consumable, if you check on this. The object would be destroyed after being interactive once.
There is also a lockscene checkbox on the top of this script editor. It means where you can pick up or interactive the gameobject when you check it on.

Other useful scrip examples

You may need to make some creative levels of special operations. Here are some levels which maybe useful for your reference.
On level6, select boxClose gameobject. There is drag script on it.



This editor means the gameobject can be drag only on X axis(y is locked)
And its moving range is 5 to left and 0.5 to right from its origin place.

In level10, there is a script which could simulate the operation of holding on the screen.
Take a look into level10Handler for detail.

In level12, the level must be tested on the device. This level simulate an operation of tilt the device to move the item. The book on the shelf would start to move when you tilt your device.
See detail in the script of level12Hanlder.

In level 15, there is simulation of shake the device. See level15Handler for details.

More tricks

As I mentioned before, there is an easier way to assign many gameobjects on the scene.
Open level5hanlder for example. We don’ t see the assign section on this inspector editor.
We just declare the variable and can be used directly in the game.
This because we use a trick called reflection.

It is realized by the following scripts.

```
foreach (System.Reflection.MemberInfo m in typeof(level5Handler).GetMembers())
        {
            var tempVar = GameObject.Find(m.Name);
            if (tempVar != null)
            {
                this.GetType().GetField(m.Name).SetValue(this,tempVar);
            }
        }
```

So by this way, only we need to do is keep the gameobject on the scene is the same name of the variable we declared. This would save a lot of time when you have too many gameobject in a scene.

Localization

There are 3 step to add a new localization language.

add a new language in script.

Open the GameData.cs under Assets/commonSrc/Scripts
Focus on this function

```
public int GetSystemLaguage()
    {
        int manualLanguage = PlayerPrefs.GetInt("language", -1);
        int returnValue = 0;
        switch (Application.systemLanguage)
        {
            case SystemLanguage.Chinese:
                returnValue = 1;
                break;
            case SystemLanguage.ChineseSimplified:
```

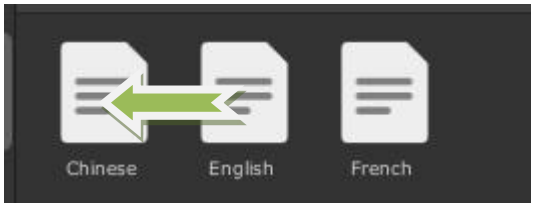
```
                returnValue = 1;
                break;
            case SystemLanguage.ChineseTraditional:
                returnValue = 1;
                break;
            default:
                returnValue = 0;
                break;

        }
        //returnValue = 0;//test
        if (manualLanguage != -1)
        {
            returnValue = manualLanguage;
        }
        return returnValue;
    }
```

Here we see there are several cases in this function but only returns 1 or 0
This is because current game only support 2 languages.
0 is English default and 1 is simplified Chinese.
If you want to add another Language.
Add a new case like this.

```
case SystemLanguage.ChineseTraditional:
                returnValue = 1;
                break;
            case SystemLanguage.French:
                returnValue = 2;
                break;
            default:
                returnValue = 0;
                break;
```

modify the localization files

under Assets/commonSrc/Resouce/Localizations
There are already 2 localization files.
If you want to add a new language file. Just Duplicate 1 first.



Now we duplicated the French as a new language

Let open the txt file for taking a look.

```
28  btnCancel = cancel
29  btnBuy = buy coin
30  btnClose = close
31  btnTitle = Title
32  btnMenu = Menu
33  btnRetry = Retry
34  btnStart = Start Game
35  btnReview =Rate & Review
36  btnMore = More Game
37  btnContinue = Cointinue
38  levelScore = Total Score:
39  levelTitle =   Level Menu
40
41  titleShop = Coin Store
42  price1tip = Get 300 Coin
```

As you will see, a localization file got short phases in each of the line.

For example at the line 28

btnCancel = cancel

which means, if you call a localization script in the game by using the Key named "btnCancel",the French language will show the word cancel.

How use call localization by script, Just do the following:

```
Localization.Instance.GetString ("btnCancel");
```

You can use to assign text by using scripts such as:

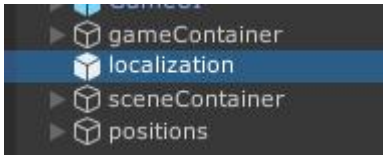
```
xxx.text  = Localization.Instance.GetString ("btnCancel");
```

Remember the key which you try to search must exist in the current language file. Otherwise an error would be happen.

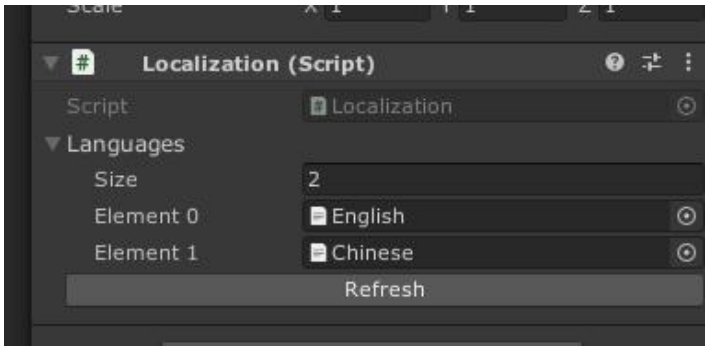
Setup localization GameObjects

The localization is a singleton component so it requires to be added in each scene where you need to use localizations.

For example, Let's open the scene file of level1



Here select the localization gameobject in the hierarchy. Then take a look at its inspector.



We see there are already 2 files loaded in this prefab.

If you already added a new language, just change size to 3 and drag the new language files into the section.

Apply the prefab to make the change being applied to all the localization gameobjects in the game.

So you just required to change the gameobject once.

Switch language

Everything is ready, but requires to tell the system what language your player want to use.

Let open MainMenu scene file to do this job.

First take a look its hierarchy , the PanelSetting gameobject is defaultly deactivated.

So you active it first to make it visible in the editor.

So This is what to selected in the scene.

This radio box got a function of onToggle on the scripts panelsetting.cs

Select gameobject panelSetting and open the script

```
case "Radio 0":
    GameManager.getInstance().playSfx("click");
    if (GameObject.Find("Radio 0").GetComponent<Toggle>().isOn)
    {
        PlayerPrefs.SetInt("language", 0);
        refresh();
    }
    break;
case "Radio 1":
    GameManager.getInstance().playSfx("click");
    if(GameObject.Find("Radio 1").GetComponent<Toggle>().isOn){
        PlayerPrefs.SetInt("language", 1);
        refresh();
    }
    break;
```

```
void refresh()
    {
        int clan = GameData.getInstance().GetSystemLaguage();
        Localization.Instance.SetLanguage(clan);
        initText();
        panelMain.initView();
    }
```

Here I marked the important script by yellow.

We see the most important script is tell the system what we language we set.

When we click on the language button, we just

Player.setInt(“language”,number)

This stored a key named language to make the system know whether we have set the language manually or just using the default.

We back to the Gamedata.cs and this the function Getsystemlanguage

```
public int GetSystemLaguage()
    {
        int manualLanguage = PlayerPrefs.GetInt("language", -1);
        int returnValue = 0;
…..
if (manualLanguage != -1)
        {
            returnValue = manualLanguage;
        }
        return returnValue;
```

We see in this function, the system judges whether uses the default language refers to the variable manualLanuage which get playerprefs variables from the save.

So, now you can duplicate toggle buttons in the setting panel and add cases like the radio0 and radio1 to add new language setting for your customer.

The Tip system

Still, let’s open the language file English.txt

```
92
93    level1tips0 = Seems that the girl can't fi
94    level1tips1 = You'd better give things cor
95    level1tips2 = Although all three things ar
96    level2tips0 = Watch the cat's state, somet
97    level2tips1 = When the cat snores (zzz on
98    level2tips2 = However, don't forget to cov
99
00
01    level3tips0 = Can't open the window? Where
02    level3tips1 = Touch on the potted plant, t
03    level3tips2 = A caterpillar just crawls by
```

We see those phases being included.

Each level got 3 tips. So we have 3 lines for each levels.

You can modify them by yourself for each language.

How to add a new levels

By default the game got 20 levels and 1 tutorial level.

There are files from level1-level20 and level0 is the tutorial file.

Now if you got a level of level21

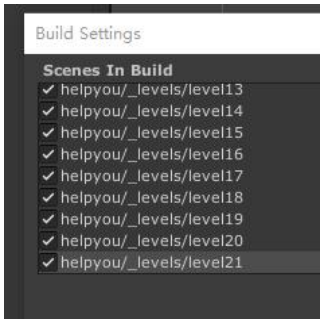
You should let the system know the level count is 21 first

```
29        public GameObject maing;//not used
30        public static int totalLevel = 20;//total levels
31        public int[] tipUsed = new int[totalLevel];
```

Open Gamedata.cs and locate at this line.

Change the variable totalLevel from 20 to 21.

Then goto file->build settings and drag the new created scene file into the list.



Now when you start the game and enter the menu, you will see a new level is already existed.

By default the level is locked because you must finish level 20 to unlock it.

You should add tips for level21 also

Use level21tip0-level21tip2 to set all the level tip localization texts.

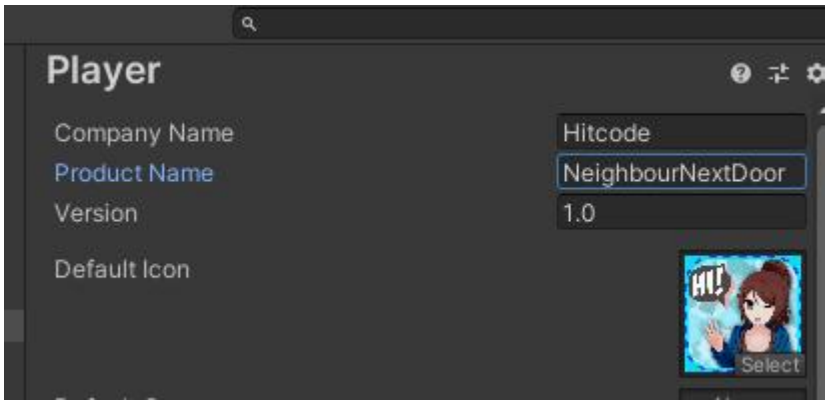
For detail, read the chapter of Localization.

About more detail of how to create a new level. See the chapter of make your own levels

Setup your package Ids

Game package Id

Before you publish your game, you should set up your unique package id.
Here in project settings



Game Center Id

If you pubish your game to appstore.
In order to make game center works. You should set up the learder board id and appid also.
Search and Open Const.cs

```
//system
public static string LEADER_BOARD_ID = "com.Hitcode.NeighbourNextDoor";
public static int appid = 1516154589;
```

Change these variables to your own on your apple developer ends.

Set up the advertisement（deprecated）

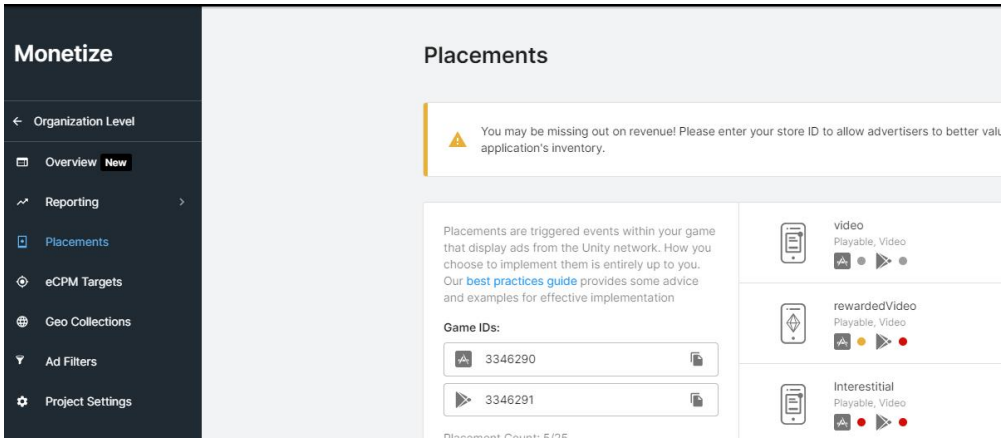
View AdsAndTips since 1.15

Setup Id

The most import part maybe how to earn revenue from your product. This game only support ads but it will be a more easier and safety way to earn.
Because in app purchase included games are not allowed to be published in the big market such as China without the publish code.
It is impossible for indie developer to meet Chinese store requirement.
So ads will be your best choice.
This game support most popular advertisements and you can set up easily without any script.
To set up your own ads ids, do the following:
Goto GameManger.cs find

```
bool testmode = true;
string androidGameId = "3346291";
string iosGameId = "3346290";
string gameId;
string rewardPlacementId = "rewardedVideo";
```

Change these Ids with your own. You can find them in unity developer backend dashboard.



Learn detail about this from Official tutorial

Turn off test mode

Find `bool testmode = true;` in Gamemanager.cs
Set to bool testmode = false;
Before you want to publish the game. Otherwise the game would always show fake ads.

Basic knowleage must know

build and submit a game on app store with unity.

link

build and apk on android platform

link

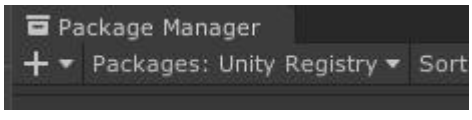
FAQ

Why I get errors during a import?

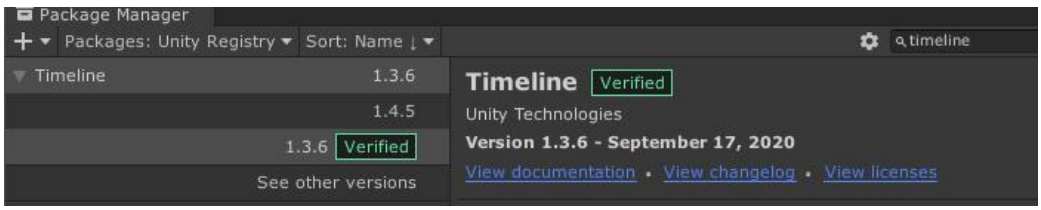
As unity start to use package manager, sometimes it may got import error like below:

> Project has invalid dependencies:
>
> com.unity.2d.sprite: Package [com.unity.2d.sprite@1.0.0] cannot be found
>
> com.unity.2d.tilemap: Package [com.unity.2d.tilemap@1.0.0] cannot be found
>
> com.unity.modules.androidjni: Package [com.unity.modules.androidjni@1.0.0] cannot be found
>
> com.unity.package-manager-ui: Package [com.unity.package-manager-ui@2.2.0] cannot be found
>
> com.unity.timeline: Package [com.unity.timeline@1.1.0] cannot be found

If you face such problems, do not be afraid. Just see there are some keywords in these errors.
Here above, there are many things in the branch before a version.
Like[com.unity.timeline@1.0.0] so this will be a native plugin in unity.
And this plugin is now not be imported correctly.
To fix this just
Goto window->package manage
Select unity registry like below



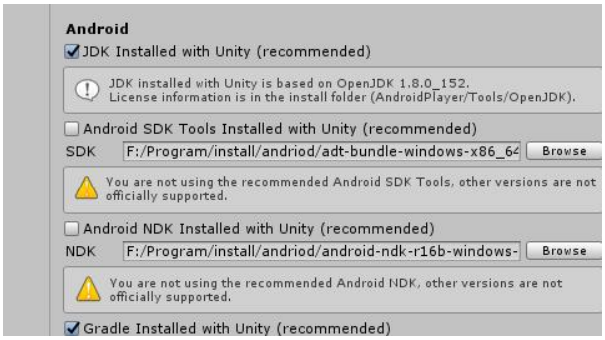
Then search timeline on the right.



Here you can choose to reinstall or uninstall the package.
Most probably your error would be fixed.

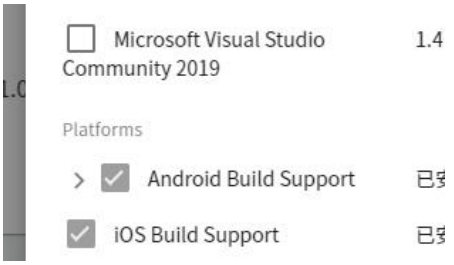
I can not create my own keystore

Goto edit-perference-external tools



Make sure your JDK,SDK and NDK were all assigned correctly.

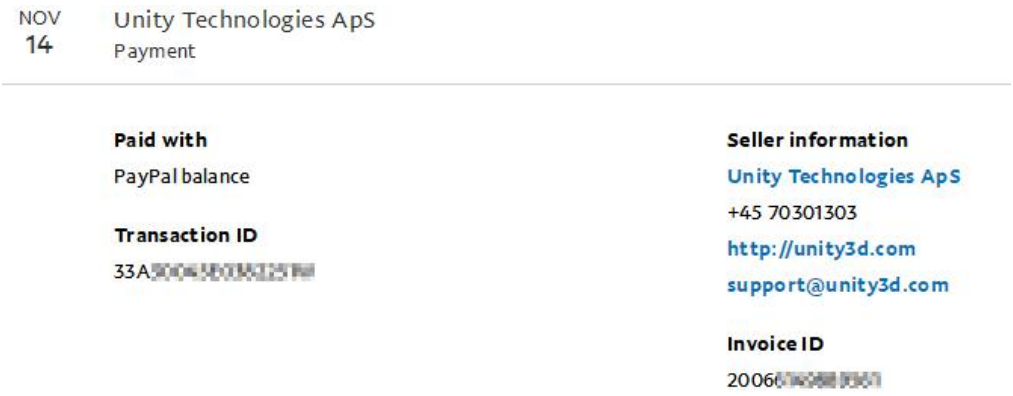
You can install them with hub if you install the unity editor by with it. Otherwise download the related sdk , ndk package from unity official, unzip them and assign manually in the external tools section.



Author support

Contact to us E-mail

Remember attach your invoice otherwise there would not be my reply.


NOV 14 Unity Technologies ApS
Payment

Paid with
PayPal balance

Transaction ID
33A

Seller information
Unity Technologies ApS
+45 70301303
http://unity3d.com
support@unity3d.com

Invoice ID
2006

Log in to the Unity Asset Store https://assetstore.unity.com/

Go to your purchase history， Click your avatar in the top right → My Assets

Find the asset you purchased, and click View Invoice on the right (Sometimes you'll need to open Order History first to locate it.)

Check the invoice number，In the invoice details, you'll see a line like Invoice Number: INxxxxxxxx

The invoice you can get from your paypal account records also.
As we are busy to dealt with many emails. Reply maybe would last 1-2 days. Sorry for the inconvenience.

If you want support our work or feel interested in other assets, take a look at More Games