```
wwwwww
1:      SHOW
show databases;   //          ---------------------------------
2:2                       dd---------------------------
create database dd;   //          --------------------------
CREATE DATABASE test5 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci

3:
mysql>  use dd;(                    Database changed
4:
mysql> show tables;


use 002  //        --
show tables WHERE Tables_in_s002 like  'tb_%001%' //             ----------------

create table t      (Id varchar(63) primary key,Kee varchar(255)) CHARSET=utf8; //5:                 ss--
-------------------------

CREATE  TABLE  author  (
                authorid   char(20),
             name   char(20)   ,
             age   char(20)   ,
                country   char(20)
)  ENGINE=InnoDB   DEFAULT  CHARSET=utf8  AUTO_INCREI

alter table name01 //rename name02;---------------

6:          ss
mysql> describe ss;
7:      ss
mysql>  insert  into  ss  values  (”hyq”,”M”);
8:                                              D:/mysql.txt
mysql> LOAD DATA LOCAL INFILE “D:/mysql.txt” INTO TABLE MYTABLE;
9:      .sql                    D:/mysql.sql
mysql>use database;
mysql>source d:/mysql.sql;
10:
mysql>drop TABLE MYTABLE;
11:
mysql>delete from MYTABLE;
12:
mysql>update  MYTABLE  set  sex=”f”  where  name=’hyq’;
                                   MySql            ,
  windows   MySql                                                                net  start  mysql
  Linux             “/etc/rc.d/init.d/mysqld     start”
            MySql                         root
                                                              root
use     mysql;
delete from User where User=””;
update  User  set  Password=PASSWORD(’newpassword’)  where  User=’root’;
                                                     User                Host

mysql  -uroot  -p;
mysql -uroot -pnewpassword;
mysql mydb -uroot -p;
mysql mydb -uroot -pnewpassword;
```

mydb

root root

MySql User

User GRANT

GRANT

```
grant all on mydb.* to NewUserName@HostName  identified  by  “password”  ;
grant usage on *.* to NewUserName@HostName  identified  by  “password”;
grant select,insert,update on mydb.* to NewUserName@HostName  identified  by  “password”;
grant update,delete on mydb.TestTable to NewUserName@HostName  identified  by  “password”;
```

GRANT WITH GRANT OPTION

User Password PASSWORD

User

REVOKE

(www.cn-java.com)

FILE: MySQL

PROCESS:

RELOAD:

SHUTDOWN: MySQL

/ /

ALTER: (/)

CREATE:

DELETE:

DROP:

INDEX:

INSERT:

SELECT: /

UPDATE:

ALL: (root)

USAGE:

//cmd mysql --------------------

//cmd

3306 mysql -u root -p

//cmd

Mysql -P -h mysql \ip -u root -p

//mysql ---- ----------------------

```
PS C:\Users\Administrator> mysql -u root -p
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 5.0.45-community-nt-log MySQL Community Edition (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql>
//                    --------------------------------
PS C:\Users\Administrator> mysql -h 192.168.1.138 -uroot -p
Enter password: ******
```

/// =====================

root cmd--mysql

------------1--------

```
SET PASSWORD FOR root=PASSWORD('ASDfgh132');
```

-------------2--------------

```
   MySQL> update MySQL.user set password=PASSWORD('ASDfgh132') where User='root';
   MySQL> flush privileges;
   MySQL> quit
```

//root ==================================

```
Grant all privileges on *.* to 'root'@'%' identified by 'Abc132' with grant option;
flush privileges;   //(                                MySQL)
```

```
Database changed;   //--                    ,
mysql> SELECT User, Password, Host FROM user;
```

// ---Access denied for user 'root'@'localhost' (using password: YES)

localhost root localhost root

```
create user 'root'@'localhost' identified by '          ';
grant all privileges on *.* to root@'localhost';
```

sql

//------- ------------ “ ” MySQL InnoDB ----------

The 'InnoDB' feature is disabled; you need MySQL built with 'InnoDB' to have it working

my.ini skip-innodb # skip-innodb #

#

// ------------------------------

1 % 0

%%

```
SELECT * FROM [user] WHERE u_name LIKE '%   %'
```

u_name “ ” “ ” “ ” “ ” “ ”

u_name “ ” “ ” and

```
SELECT * FROM [user] WHERE u_name LIKE '%   %' AND u_name LIKE '%   %'
```

2 _

```
SELECT * FROM [user] WHERE u_name LIKE '_   _'
```

“ ” u_name “ ”

3 [ ]

SELECT * FROM [user] WHERE u_name LIKE '[ ] ' “ ” “ ” “ ”
“ ”

[ ] 01234 abcde “0-4” “a-e”
SELECT * FROM [user] WHERE u_name LIKE ' [1-9]' “ 1” “ 2” …… “ 9”

4 [^] []

SELECT * FROM [user] WHERE u_name LIKE '[^] ' “ ” “ ” “ ” “
” “ ”
SELECT * FROM [user] WHERE u_name LIKE ' [^1-4]'; “ 1” “ 4” “ 5” “ 6”
……

//------------------mysql --

MySQL my.ini [mysqld]

```
lower_case_table_names = 0
```

0

1 MySQL

//Server2012 ------------------------------
Win+R
rundll32.exe shell32.dll,Control_RunDLL desk.cpl,,0

```
  using   System;
  using   System.Configuration;
  using   MySql.Data.MySqlClient;
  public  class  TestDatebase{//        :test/    :student/      :id,    ,    ,    /
        public  static  void  Main(String[]  args){
                    String   mysqlStr   =   "Database=test;Data  Source=127.0.0.1;User
  Id=root;Password=Xiahai132;pooling=false;CharSet=utf8;port=3306";
                    MySqlConnection   mysql   =   new   MySqlConnection(mysqlStr);
                String  o_    sql  =  "select  *  from  student";  //
        // o_    sql = "select * from student where     ='    '"; //            _            (
        )
       //  o_    sql = "select * from student where      like '  %'"; //           _
          )
                o_    sql  =  "select        from  student  where  id=12";  //
                String  o_    sql  =  "insert  into  student(    ,    ,    )  values  ('    ',25,'
  /
              String  o_    sql  =  "update  student  set        ='    '   where  id  =  11";  //    id

              String  o_    sql  =  "delete  from  student  where  id  =  12";  //    id
                                                              try{
                                   MySqlCommand   mySqlCommand   =   new   MySqlCommand(o_    sql,
                                             mySqlCommand    =    new    MySqlCommand(o_    sql,
                        //             mySqlCommand    =   new   MySqlCommand(o_    sql,   mysql);
                             //        mySqlCommand    =   new   MySqlCommand(o_    sql,   mysql);


                                                     //                  mySqlCommand.ExecuteNo


                            }catch    (Exception    ex){
                                                                                  Console.Write
                                                              }
                        String    readLine    =    Console.ReadLine();
                  }
        public  static  void  S_                    (MySqlCommand  mySqlCommand){
                        MySqlDataReader    reader    =   mySqlCommand.ExecuteReader();
                                                              try{
                                                                           while       (reader


                                      //               //         Console.WriteLine("id:"    +
  reader.GetString(1)    +    "|    :"    +    reader.GetInt32(2)    +    "|    :"    +    reader.GetStri
                                                       Console.WriteLine("id:"     +     reader.


                                    }catch     (Exception){

                                                              }
                                                              finally{

                                                              }

                  }
  }
```

```
//-------------------     ------------------------
update            set     a=5,b=52      where    n=45  and  m=48
```

```
//----------------                              ------------------------------------
insert       (ID,name) values('jerichen','gdsz'),('alone','gdgz');

//----------------                                     ------------------------------------
   REPLACE INTO       (ID,name) values('4','Jame444s');
   REPLACE INTO       (ID,name) values('4','Jame444s'),('7','Jame444s');

//------------------------          ----------------------------------------
DELETE FROM v       WHERE NAME ='22'
//    -------------------------------------------
SELECT a.X, a.Y FROM TB_CSYX_GRAPHICS a
//-------------          --------------------------and:    or:

 select  count(*)  from `table`  where  `id1`=1  and  `check`='b';

select  *  from          where  user_ID='5ff16828-3895-4f56-acd8-497a8f8fc245'  and(`s02_        `<100  or  `s04_



//--------------------          ---------------------------------------------

select
        case  when  a.A_ID  =  '5ff16828-3895-4f56-acd8-497a8f8fc245'  then  a.B_ID
                          when  a.B_ID  =  '5ff16828-3895-4f56-acd8-497a8f8fc245'  then  a.A_
                                      else     'OK'
         end  as  status
from `      `  a  where  (A_ID='5ff16828-3895-4f56-acd8-497a8f8fc245')  or  (B_ID='5ff16828-3895-4f



//---------------------           ------------------------------------------------------

SELECT * FROM `        `,`         `


//---------------------                  -----------------------------------

select  a.ID,(select  b.A_ID from `tb_        `  b  where  b.B_ID = a.ID )hh  from  tb_user  a
```

20

http://www.daogame.cn/project.html

wwwwww

```
unity3d      -    Edit/Project Settings/Player/Setting for Wed Player/Run In Background* (
//            ----------------
  \Unity\Editor\Data\Mono\lib\mono\2.0\***.dll    Project
//var      :       ;
bute:        , short:      , int:    , long:      , float:      , double:        boolean:
//
                                        .ToString();
    ||
        &&

void   OnDestroy()  //MonoBehaviour  {}    ------------------------

---------------------------------------------------------------c#    ----------------------
////                 ------------------------
\'  :        ,  \"  :        ,  \\  :            ,  \0  :        ,  \a  :        ,  \v  :
\b  :         \f  :            ,  \n  :        ,  \r  :        ,  \t  :            ,

public  int[]  dd  =  new  int[0];

#region            ---------------------------------
#endregion
///      -------------
enum   int{   ----------------------------------------
         0,1,2,............204149866;
}
enum   xxx{   ----------------------------------------
xx,ssa,fff,kgk,
}
xxx   dd1   =   xxx.xx;
xxx   dd2   =   xxx.ssa;


//          ----------------------------------------------
StartCoroutine(ss());//        ---
public   IEnumerator   ss(){
       int  i  =  1;
       yield  return  new  WaitForSeconds(4);    //
        i  =  2;
}
IEnumerator  I_   //(C_     --dd){
    while(dd.o_    y.material.color.a>0){
                                           yield    r
                  }
                 dd.o_    y.SetActive(false);
}
//       ------------------------------------------------
InvokeRepeating("S_001",   4,   2);//4          S_001()
CancelInvoke("S_001");//    S_001()

double    dd    =   double.Parse("1253");    //    do
string    ss    =  //123.ToString();
//          ---------------------------------------------
Mathf.Abs(-10);     10
//          --------
Mathf.Round(2.22f)    =    2;
//      ----------
float    ff=1;ff    =    Mathf.Clamp(ff,2,6    )
Mathf.Pow(4,5);
Mathf.Sqrt(55);
```

```
//          ---------------
51%2 //-------51    2            1

//-----C#                 -------------------------
class         //
struct    //
interface //
abstract //
partial    //
const         //
public    //
private //
static    //
sealed    //
protected //
virtual            //
override    //
internal //
extern //
//------                           ---------------
```

1.

2.

```
//if    --------------------
    if(    ){

} else if(    ){

} else {

}
//switch-----------------------------
var i: int =110;
switch(i){
case 100: //     i    100
             1;
             2;
         break;
case 110: //     i    110
             3;
         break;
default;   //     i
             4;
         break;
}

break;//        ----------
continue;//           -----------

//          ----------------------
tt:int;bb:boolean;-- tt = bb?5:0; ---------    bb            ,         5,         0;//---------------
//    --------------------------------
var d:int[];
d = new int[3];
d[0] = 3;
d[1] = 1220;
d[2] = 23;
//        -----------C#---------
int[,] ii = new int[12,5];

//for    ------------------------------
for (int n=1;n<=5;n++){
         print(n.ToString());
}
break:            ;                 continue:            ;
foreach---------------------------------
return    //

//     ---------------------------------
   gameObject.transform.position,
   gameObject.transform.rotation,
    gameObject.transform.Translate(0,0,m*Time.deltaTime,Space.Self);
   gameObject.transform.position.y = 0;

up         //              Y
right         //              X
forward    //              Z
//    -------
position //
localPosition //
TransformDirection //                              direction
InverseTransformDirection //
InverseTransformPoint //                          //
TransformPoint //                          //
//    ------
eulerAngles    //
localEulerAngles    //
rotation //
localRotation //
//    ----
localScale    //

Translate();
Rotate();
RotateAround (vv,ff);

root //            -----------
parent //      ----------
childCount //              --------------
GetChild(2); //            -----------
DetachChildren //
IsChildOf //

transform.Translate(Vector3.forward);    //Z
transform.Rotate(Vector3.forward);        //Z
Space.World //
Space.Self //
transform.Rotate(transform.right * Input.GetAxis("Mouse Y"),Space.World);   //

Quaternion     =Quaternion.Euler(5,8,2);//------------          ----------
Vector3 vv=      .ToEulerAngles(); //--------          ------------
Vector3 vv2 = o_           .transform.TransformDirection(vv); //-------
Vector3 vv = o_           .transform.InverseTransformDirection(vv2) //----
int ii=gg.Transform.childCount; //                --------------
Transform  tt  =  gg.Transform.GetChild(0);//              -----------




//              ---------------------------------------------------------
                                                  Input.GetKeyDown(KeyCode.Space);
```

```
                                                 Input.GetKeyUp(KeyCode.Space);
//      -------------------------------------------------------------
                                                 Input.anyKey;

//           -------------------------------
Input.GetAxis("    ");
Input.GetAxis("Mouse X")//---------------      X
Input.GetAxis("Mouse Y")//---------------      Y
Input.GetAxis("Mouse ScrollWheel") //----------
Input.mousePosition; //------------------

//         ------------------------------
Input.touchCount>0    //              0-------------------------------------------------------
Input.GetTouch(0).phase   ==   TouchPhase.Moved               //         _        __
Input.GetTouch(0).phase == TouchPhase.Stationary //        _        __
Input.GetTouch(0).phase  ==  TouchPhase.Began    //       _        __
Input.GetTouch(0).phase   ==  TouchPhase.Ended   //       _        __
Input.GetTouch(0).phase  == TouchPhase.Canceled   //
Input.GetTouch(0).deltaPosition                                    //
Input.GetTouch(0).position                                    //



//--------------------        --------------------------------
                =180/π
                =π/180

Mathf.Pow(2,3)==8 //    -----------------
Mathf.Log(8,2)==3 //    -------------------------

//              ------------------------------------------
void OnGUI(){
       Event  Mouse  =  Event.current;
     if  (Mouse.isMouse  &&  Mouse.type  ==  EventType.MouseDown  &&  Mouse.clickCount  ==  2){
                                    print("            -----");
                }else{
                                                                print("    -----");
                }
}
//              2---------------------------------------------------
void OnGUI(){
       if  (Event.current.clickCount  ==  2){
                                    print("            -----");
                }
      Event.current.type  ==  EventType.MouseDown  &&  Event.current.clickCount  ==  2
}
Horizontal
Vertical
Fire
Jump
Mouse



                    --------------------------------------------------------
       "a","b","c" ....
       "1","2","3",....
       "up","down","left","right"
        "[1]","[2]","[3]",......."[+]","[equals]"
       "right shift","left shift","right ctrl","left ctrl","right alt","left alt",
         "mouse 0","mouse 1","mouse 2"
      "backspace","tab","return","escape","space","delete","enter","insert","h
       "f1","f2","f3".....
```

```
@@@@@Input.GetAxisRaw("    ")      +1  -1---------------------------------
//        (  /  /  )-------------------------------------------------------
                                                  Input.GetMouseButton(0/1/2);
//        -----------------------------------------------------------------------
      function  OnMouseEnter(){      ;    }
       function  OnMouseOver(){      ;    }
       function  OnMouseExit(){      ;    }
       function  OnMouseDown(){      ;    }
        function  OnMouseUp(){       ;     }
       function  OnMouseDrag(){      ;    }

//        #---------------------------------------------------------
    Input.GetKey(KeyCode.#);
//        #    ----------------------------------------------------
    Input.GetKeyDown(KeyCode.#);
//      ------------------------------------------------------------
    Random.Range(1,10);       1   10
    Random r = new Random();
    ii2 = r.Next(0, 100);//         0-100
//      ---------------------------------------------------------------------------
                       Transform   dd;       //
                     gameObject.transform.parent   =   dd;

//        ----------------------------------------------------------------------
                @@While(    ){
                //
}
                @@do{
                         //
}While(    );

//      --------------------------------------------------------------------------
 Application

    Application.levelCount;(:int)                                  (     )
   Application.isEditor;(:bool)                    Unity3D           (     )
    Application.dataPath;(:string)                                    (     )
   Application.persistentDataPath

   Application.LoadLevel(          );(:void)
   Application.LoadLevel(          );(:void)
    Application.CaptureScreenshot("C:/Users/fffff.png");(:void)
    Application.OpenURL(     );(void)
     Application.Quit();(void)


DontDestroyOnLoad(o_     y);  //              -----------------------------------
System.Environment.CurrentDirectory  //                   -------------------------
System.Reflection.Assembly.GetExecutingAssembly().Location  //dll        ----------------



//               -------------------------------
function Awake (){
  Application.targetFrameRate = 300;
  }

//          http://localhost/   ----------------------------------
//            --------------------------------------------
GameObject gg = GameObject.CreatePrimitive(PrimitiveType.Cube);

Instantiate(          ,      ,      ,);   //      ------------------------
Destroy(gameObject);  //      \      ---------------------------
//            ----------------------------------------------------
GameObject gg = GameObject.Find(name:String)
GameObject gg = GameObject.Find("/obj/aa");
//            ----------------------------------------------------
Canvas cc= FindObjectOfType<Canvas>();
//                 --------------------
 AstarPath a = GameObject.FindObjectOfType(typeof(AstarPath)) as AstarPath;

//                            Renderer--------------------------------------------------
Renderer[] ff = gameObject.GetComponentsInChildren<Renderer>(true);
//                                   ------------------------------------------------------
string str = "oo";
public GameObject oo;
GameObject dd = (GameObject)this.GetType().GetField(str).GetValue(this);//----
dd.renderer.material.color = Color.blue;
print(oo.GetType() );//                --

//          ----------------------------
Resources.Load("kk");
AssetBundle("pp");
GameObject go = GameObject.CreatePrimitive(PrimitiveType.Sphere);   //
go.renderer.material.mainTexture = (Texture)Resources.Load("kk"); //Resources
WWW www = WWW.LoadFromCachOrDownload(file://D:\\iTunes\\ww001.fbx,1);   //@"C:\ProgramData\"
yield return www;


public class C_         : MonoBehaviour {
        public  Image  o_    ;
        void  Start()  {
                                                                StartCoroutine(I_    StreamingAsset
                 o_    .overrideSprite  =  Resources.Load("ww  /ww  ",  typeof(Sprite))  as  Sprit
                }
        IEnumerator  I_    StreamingAssets()  {
               string  sPath  =  "file://"  +  Application.streamingAssetsPath  +  "/ww  /test.xml";
                   WWW  www  =  new  WWW(sPath);
                          yield   return   www;
                   www.error == null)if{   (
                                      www.text);
                          }   else   {
                                                  www.errorDebug.LogError(sPath);d__"     +
                                                                }
                }
}


/////////////         ---------------------------------------------
  Screen.showCursor  =  false;
//            --------------------------------------------
public Material ss;
RenderSettings.skybox = ss;
//////                ------------------------------------------
gameObject.renderer.enabled  =  false;        //
gameObject.SetActive(false);
gameObject.activeInHierarchy==false;     //
```

```
///////              -----------------------------------------
Screen.SetResolution(1600,900,false,0);
////        -----------------------------------------------------
Screen.fullScreen = !Screen.fullScreen;
   //Screen.width      Screen.height(        )------------------------------------------------------

//          ------------------------------
Time.timeScale = 0;
//unity3d                ---PlayerPrefs-----------------------------------
PlayerPrefs.SetInt("ii", 40); //
int i_ii = PlayerPrefs.GetInt("ii", 0).ToString(); //
PlayerPrefs.DeleteKey(“key”);   //        PlayerPrefs              key
bool  b  =  PlayerPrefs.HasKey(“key”);  //        PlayerPrefs                   key

//UGUI========================================
using UnityEngine.UI;

//               ------------------------
public Sprite jj;
public Image dd;
dd.overrideSprite = jj;
//                 ----------------------------------------------------------
GameObject o_      y;
Button bb = o_      y.GetComponent<Button>();
bb.onClick.AddListener(delegate(){ S_      (); });

///            ---------------------------------------------------------------
RectTransform rr = gg.GetComponent<RectTransform>();
rr.localPosition = new Vector3(200, 60, 0);
rr.sizeDelta = new Vector2(200, 50);

//          ---------------------------------------------------------00---------------------
GetComponent<RectTransform>().SetSiblingIndex(400);

Text     ================================
//----------------------------      ------------------------------------------------------
<color=#ff0000>hi</color>
//--------------------------           ----------------------------------------------
<b>ww</b>
<i>ww</i>
//--------------------------                 --------------------------------------------------------
<size=50>ww</size>

//        -------------------------------
void S_              (Sprite[] v        ,Button v           ,Del_int v      ) {
   v          .gameObject.SetActive(false);
   RectTransform v   =(RectTransform)v           .gameObject.transform.parent;
   RectTransform v      = (RectTransform)v   .parent;
   RectTransform v   =v           .GetComponent<RectTransform>();
   int v      =(int)(v   .sizeDelta.x/v   .sizeDelta.x);
   for(int i=0;i<v        .Length;i++) {
      GameObject gg=(GameObject)Instantiate(v           .gameObject);
      gg.SetActive(true);
       gg.transform.SetParent(v           .gameObject.transform.parent);
      gg.transform.localScale=new Vector3(1,1,1);
```

```
        gg.GetComponent<RectTransform>().anchorMin=new Vector2(0,1);
        gg.GetComponent<RectTransform>().anchorMax=new Vector2(0,1);
        gg.GetComponent<RectTransform>().localPosition=v    .localPosition+
        new Vector3((v    .sizeDelta.x+10)*(i%v       ),-(v    .sizeDelta.y+10)*(i/v       ),0);
        gg.GetComponent<Image>().overrideSprite=v           [i];
         int ii=i;
        gg.GetComponent<Button>().onClick.AddListener(delegate() {
                v       (ii);
           });
    }
    float v       =(v    .sizeDelta.y+10)*((v          .Length+v       -1)/v       )+20;
    if(v       .sizeDelta.y>=v       ){
        v       =v       .sizeDelta.y;
    }
    v    .sizeDelta=new Vector2(v    .sizeDelta.x,v       );
}
//z                       ---------------------------------------------
C_U       .S_Get(o_                   ).o_              = delegate(PointerEventData zData) {
    Vector3 vv = Input.mousePosition - o_                   .position;
    Vector3 z        = new Vector3(vv.x + zData.delta.x, vv.y + zData.delta.y,0);
    float z        = Vector3.Angle(vv, z       );
    Vector3 ff = Vector3.Cross(vv, z       );
    if (ff.z < 0) {
        z        *= -1;
    }
    o_                   .localEulerAngles += Vector3.forward * z       ;
    for (int i = 0; i < o_                   .childCount; i++) {
        o_                   .GetChild(i).eulerAngles = Vector3.zero;
    }
};


//                       ============================================


 String url="http://10.38.13.137:8082/pip-dds-engine/rest";
    void Start() {
         StartCoroutine(I_hh());
    }
    IEnumerator I_hh() {
        WWW getData=new WWW(url);
        yield return getData;
        if(getData.error!=null) {
              Debug.Log(getData.error);
        } else {
              Debug.Log(getData.text);
           }
    }
Convert.ToInt64("F",16)==15  //16                10       ------------------------------

var s:String;
s = "http://hgghjihdtg/4";
s = WWW.EscapeURI(s);                                         //WWW.EscapeURI()
s = WWW.UnEscapeURL(s);                                        //WWW.UnEscapeURL()
```

```
ss.Length                                                   //Length       s.Length  ==8;
ss.Insert(3,"AA") == "httAAp://hgghjihdtg/4"          //Lnsert
ss.Substring(2,3);        //--------------------------------           ------------------------------------------
ss.Remove(3,2) == "htt//hgghjihdtg/4";              //Remove
ss.Replace("hh","AA") == "htt//AggAAjiAAdtg/4";    //Replace                 hh         AA

string[]  sss = ss.Split(":"[0]);      //Split

string[] sss = ss.Split(new char[] { '$','&','%' });  //           ----------------------------------

//   ss.Contains("@")-----------------------                              -------------------------
string ss = "gggaa";
if (ss.Contains("a")){
   print("      a");
}else{
   print("         a");
}

string ss = "ggugaa";
int ii = ss.IndexOf("u");  //-----------      u                     /             -1  //------------------------------


string[] sss2 = ss.Split('a', 'g');  //---------------------              ----------------------------------------------
string[] sss3 = ss.Split(new string[] { "aa" },StringSplitOptions.RemoveEmptyEntries);//---------

ss = ss.ToLower(); //
ss = ss.ToUpper(); //
ss = ss.Trim();  //

bool b = ss.EndsWith("a");  //                         a
bool b2 = ss.StartsWith("a");//                        a
bool b3 = string.IsNullOrEmpty(ss); //                    null         "";
string ss2 = string.Format("ddss{0},ds{1}s", "dd", 45);   //                //-----------------------------------

//                      ----------------------------
string ss ="rrrrrrrr";
ss = ss.ToString("0.200").Substring(0,2);
ss =="rr";
//                     ----------------------
1. String.Compare(str1, str2) == 0         str1.CompareTo(str2) == 0
2. str1.Equals(str2)         String.Equals(str1, str2)
==================================================================================================

string z       = @"\WebSite1\Default.aspx";
string  z        =  System.IO.Path.GetFileName(z       );//            “Default.aspx”
string  z          =  System.IO.Path.GetExtension(z       );//           “.aspx”
string  z          =  System.IO.Path.GetFileNameWithoutExtension(z      );//                       “Default”


Screen.width  Screen.height   //                  ------------------------------
////////
Screen.setResolution(800,600,false,0);
var p:Resolution;   //
  p.width                    //
 p.height                //
 p.refreshRate      //
```

```
Screen.currentResolution;      //            ------------------------------------------------------------

/////////////////               ------------------------------------------------------------
1.MonoBehaviour.OnTriggerEnter( Collider other )
2.MonoBehaviour.OnTriggerExit( Collider other )
3.MonoBehaviour.OnTriggerStay( Collider other )

1.MonoBehaviour.OnCollisionEnter( Collision collisionInfo )
2.MonoBehaviour.OnCollisionExit( Collision collisionInfo )
3.MonoBehaviour.OnCollisionStay( Collision collisionInfo )

//////////////            ------------------------------------------------
Collider[] collider = Physics.OverlapSphere(transform.position,5);
foreach(Collider col in collider) {
}

gameObject.GetComponent<SphereCollider>().isTrigger = true;
gameObject.GetComponent<Rigidbody>().useGravity = false;

//                  ------          --------------------------------------
 GameObject nn;
 void OnCollisionEnter(Collision  dd) {
       nn = dd.gameObject;
       if (nn.transform.tag == "mm"){
          nn.renderer.material.color = Color.red;
         }
  }
Vector3  vv = ToEulerAngles(qq);//        ----------
Quaternion qq = Quaternion.Euler(vv); //         ---------
Vector3 vv = Vector3 .Lerp(v1,v2,0.3f);  //t   0   1 --//         --          ----
Quaternion qq = Quaternion.Slerp(q1,q2,0.3f);//           -----------
Vector3 vv = Vector3 .Slerp(v1,v2,0.3f);//           -----------
Vector3 .magnitude     //               -------------------
Vector3  vv= Vector3.Project(v1,v2);  //      ---1            1        ----------
float ff = Vector3.Distance(v1,v2);//                ---------------------
transform.RotateAround(o_        .transform.position, Vector3.up , 100 * Time.deltaTime);/           ------
transform.LookAt(o_        .transform);//          ---------------------------
Vector3 v2 = gameObject.transform.TransformDirection(v1);//                   ---------------------
Vector3 vv2 = vv1.normalized; //            ----                 1   ---------


DontDestroyOnLoad(gg)    //                   ---//                       -----------------------


//////////////////          -----------------------------------
aa.transform.parent = bb.transform;  //aa      bb

//////////////////                    ---------------------------------
o_anim.Play("Attack01");
o_anim.Stop("Attack01");
o_anim["Attack01"].speed = 2;  //
o_anim.wrapMode = WrapMode.Loop;
if(o_anim.IsPlaying("Attack01")==true) {//      "Attack01"
   print("      'Attack01'          ");
}

//                      other                   ---------------------------------------------------------
```

```
public Transform other;
if (other) {
 var forward = transform.TransformDirection(Vector3.forward);
 var toOther = other.position - transform.position;
 if (Vector3.Dot(forward,toOther) < 0){     //
                //
    }
}

///c#   Animatr
if(GetComponent<Animator>()){ ----------//                              GetComponent<Animator>();\          Ar
     if(GetComponent<Animator>().GetCurrentAnimatorStateInfo(0).IsName("Base Layer.ww")){
 ------------------------------ //            .                    .               /(Base Layer) ---//                ww
  if(Input.GetMouseButton(0)){ ---------------------//
   GetComponent<Animator>().SetBool("jj",true);-----------//      jj
  }else{
   GetComponent<Animator>().SetBool("jj",false);---------------//
   }
  }
}

GetComponent.<Animator>().SetFloat(         ,                    ,                ,            );
     GetComponent.<Animator>().SetFloat("rr",tt,0.25,Time.deltaTime);--------------//rr           tt
GetComponent.<Animator>().SetBool("         ",false/true)
       GetComponent.<Animator>().SetBool("jj",false);---------------------------------------//
GetComponent<Animator>().SetInteger("          ",      );
      GetComponent<Animator>().SetInteger("jj",2);
------------float nnn = GetComponent.<Animator>().GetFloat("rr");

//                     -------------------------------------------------------------------------
GetComponent<Animator>().speed = 0;


//                          ------------------------------------------------------------
if (!gameObject.GetComponent <cc_color>()){
    gameObject.AddComponent<cc_color>();  //cc_color
}
//iTween------------------------------------------------------------------
public GameObject A, B;
iTween.MoveTo(A, iTween.Hash("position", B.transform.position, "time", 2));
iTween.RotateTo(A, iTween.Hash("rotation", B.transform.rotation.eulerAngles, "time", 2));
iTween.ScaleTo(A, iTween.Hash("scale", B.transform.localScale, "time", 2));
//DG.Tween------------------------------------------------------------------
using DG.Tweening;

Tweener  tw= gg.transform.DOScale(new Vector3(0.3f,0.3f,0.3f),0.5f);
tw.SetUpdate(true); //------            Tween      Time.scale
tw.SetEase(Ease.InOutBack);  //-----
tw.onComplete=delegate() {};
//      --------------------
  image.material.DOColor(Color.black,2f).onComplete=delegate()  {
                                                                                          ima
   };

//
```

```
Vector3 speed =GetComponent<Rigidbody>().velocity;   //
speed.x = 0;
speed.z = 4;
GetComponent.<Rigidbody>().velocity = speed;
------------------      ----GetComponent<Rigidbody>().velocity = new Vector3(0,0,5);

//              -------------------------------------
GetComponent<CharacterController>();
void OnAnimatorMove(){
   GetComponent.<CharacterController>().Move(            );
  }
//
GetComponent.<Animator>.speed = 1 + Random.Range(-0.8f,0.8f);
GetComponent.<Animator>.deltaPosition;     //
GetComponent.<Animator>.rootPosition;     //
GetComponent.<Animator>.rootRotation;      //

///IK      ------GetComponent<Animator>.SetIKPosition(AvatarIKGoal.     ,            );
  GameObject mm1;
void OnAnimatorIK(){
   if(layerIndex ==1){
       GetComponent<Animator>.SetIKPosition(AvatarIKGoal.LeftHand,mm1.transform.position);   //
       GetComponent<Animator>.SetIKPositionWeight(AvatarIKGoal.LeftHand,            );
       GetComponent<Animator>.SetIKRotation(AvatarIKGoal.LeftHand,mm1.transform.Rotation);   //
       GetComponent<Animator>.SetIKRotationWeight(AvatarIKGoal.LeftHand,            );
         }
}
------------------------------------------------------------------------------------------------
/////////////-----
//////////////----Edit-Project Settings-Quality-Shadow Distance



//UV            ---------------------------------------------------
float offset = Time.time * 0.05f;
renderer.material.SetTextureOffset ("_MainTex", new Vector2(offset,0));
//        -------------------------------------------------------------------
  gameObject.renderer.material.color = Color.red;     ash   ;white   ;black   ;red   ;Green   ;blue   ;pu



//                     --------------------------------------------------v
Mesh mm=gg.GetComponent<MeshFilter>().sharedMesh;
Vector3[] v      s=mm.vertices;

//                         ----------
C_mm[] cc= FindObjectsOfType<C_mm>();

//                         ----------------------------------------------------------
public  Texture ff;
renderer.material = new Material(Shader.Find("Game/Tree")); //        ---------
gameObject.renderer.material.mainTexture  =  ff;     //        -------------

1.Start()
```

Update Start Start MonoBehaviour
Start Awake

2.Awake()

Awake Start()
Awake (GameObject)
GameObjectFindWithTag

3.Update()

Start

4.FixedUpdate()

Rigdibody FixedUpdate Update
FixedUpdate()

```
        void  FixedUpdate()  {
                                                                rigidbody.AddForce(Vector3.up);
                }
```

5.LateUpdate()

Update
LateUpdate()

6.OnGUI()

OnGUI() (several times) MonoBehaviour enabled false C

7.DontDestroyOnLoad()

GameObj

8.OnDrawGizmos // ----

```
void Reset() {
                Debug.Log("   ");
}
void OnValidate() {
                Debug.Log("              ");
}
```

//MonoBehaviour ----------------------------

Reset // --Reset

OnValidate //

001__Awake___
001.2__OnEnable___
002__OnApplicationPause___
003__Start___
Update

yield return new WaitForSeconds(4);

LateUpdate

101__OnApplicationQuit
102__OnDisable___
102.2__OnDestroy___

1 Unity3d InPut

Input.GetKey(KeyCode.Escape) Input.GetKey(KeyCode.Home) Input.GetKey(KeyCode.Menu)

2 Unity3d

OnApplicationFocus// -- OnApplicationPause// --

void OnApplicationPause() //

```
void OnApplicationFocus()    //----------        ----------------



void aways();



void start(0;

void up();


#if   UNITY_IPHONE||UNITY_ANDROID //        -------------------
  Debug.Log("    ");
#endif
#if UNITY_STANDALONE_WIN //win    ---------------------
  Debug.Log("WINDOWS");
#endif




GameObject gg=GameObject.CreatePrimitive(PrimitiveType.Sphere);//            ------

//-------------------            --------------------------
using UnityEngine;
using System.Collections;
public class C_      : MonoBehaviour {
 public GameObject o_     y,o_        y;
 public float o_     y=200;
  float o_     ;
  Vector3 o_        ;
 void Start () {}
 void Update () {
  o_         =o_     y.transform.position;
  o_     =Vector2.Distance(new Vector2(o_          .x,o_          .z),new Vector2(transform.position.x,tran
  if(o_     >0.2f) {
   transform.LookAt(new Vector3(o_          .x,o_          .y+o_     ,o_          .z));
   transform.Translate(Vector3.forward*Time.deltaTime*o_     y);
  } else {
   Instantiate(o_          y,gameObject.transform.position,Quaternion.identity);
   Destroy(gameObject);
  }
 }
}
//----------------------------        -----------------------------------------
public int v      =1000,v      =2000;
IEnumerator I_         () {
    while(Mathf.Abs(v       -v       )>3) {
        int v     =v       -v       ;
        v        +=(v     +(v     >0?1:-1)*30)/30;
        yield return 5;
    }
     v        =v        ;
}

//----------------     --------------------------------
[ExecuteInEditMode]   //                 --------
[RequireComponent(typeof(Rigidbody))]  //                --------
[DisallowMultipleComponent]   //                  -------
[Serializable]  //          ------------   u3d            ------------
 [SerializeField]//          ------------   u3d            ------------
[NonSerialized]  //          ------------   u3d            ------------
[ExecuteInEditMode]    //                    --------------
[ContextMenu ("Do Something")]   //               -----------
[AddComponentMenu("Tra/dddd")]    //               -----------
[SerializeField,Range(0,5)]   //      -------------------------float
[SerializeField,Space(15)]    //          ---------------------
[SerializeField, MultilineAttribute(2)]   //                    ------
[Tooltip("            ……")]          //
```

//A Pathfinding Project Pro3.5.1----   =================================

//-----------   -----------------------

C_AstarPath.CS

    Graphs   RecastGraph //----------recast

Cell Size //----------------   1

Min Region Size

walkable climb //----------------   0

Character Radius //--------

MaxEdgeError //---------------   1

LayerMask //---------------

//-----------   --------------------

C_AIPath2.CS

FunnelModifier.cs

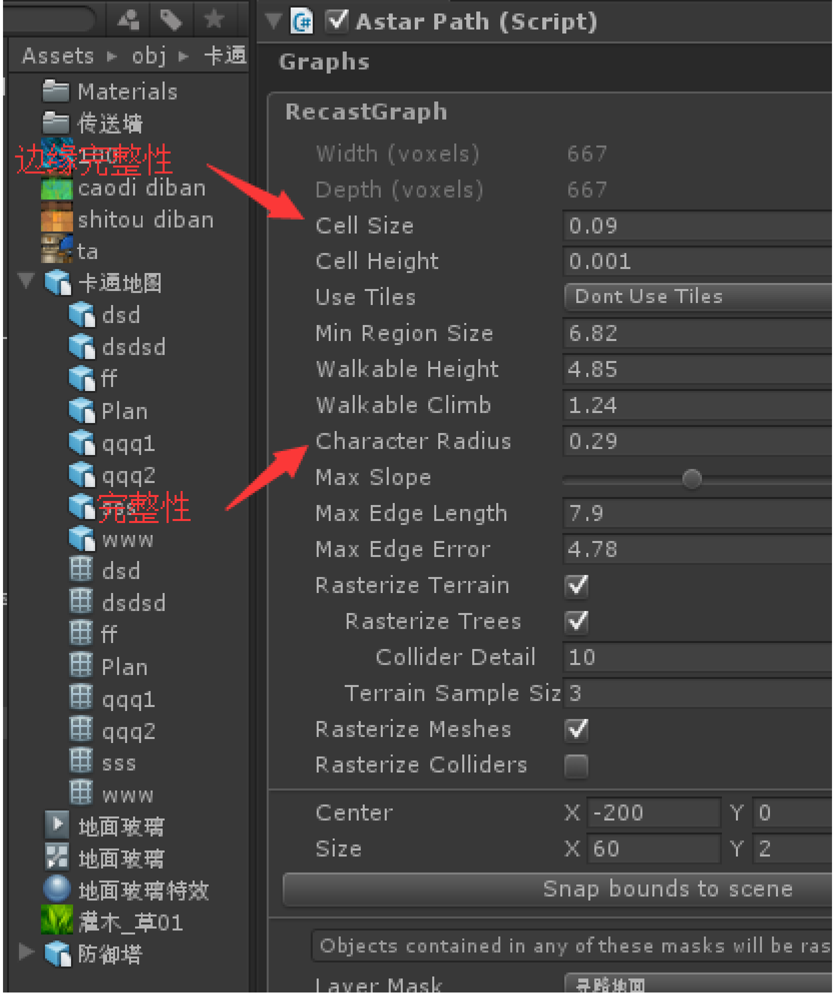
Seeker.cs

"o_   "   "o_   "

//----------   ---------------------

C_NavmeshCut

TileHandlerHelper.CS   //



边缘完整性

完整性

☑ Astar Path (Script)

Graphs

RecastGraph

```
Width (voxels)        667
Depth (voxels)        667
Cell Size             0.09
Cell Height           0.001
Use Tiles             Dont Use Tiles
Min Region Size       6.82
Walkable Height       4.85
Walkable Climb        1.24
Character Radius      0.29
Max Slope
Max Edge Length       7.9
Max Edge Error        4.78
Rasterize Terrain     ☑
  Rasterize Trees     ☑
    Collider Detail   10
  Terrain Sample Siz  3
Rasterize Meshes      ☑
Rasterize Colliders   ☐

Center                X -200        Y 0
Size                  X 60          Y 2
```

Snap bounds to scene

Objects contained in any of these masks will be ras

Layer Mask 寻路地图

```
//            -------------------------
C_
    void    Start()    {
                                                                  o_     =GetComponent<C_AIPath2>();
                                                                  o_     .d_        =delegate(){

                                                                  };
                                    o_     .d_        =delegate()    {

                                                                  };
}
void S_    _    (Vector3 v      ){
                                                                  o_     .S_          (v       );
}
void S_    _    (GameObject v        ) {
                                                                  o_     .S_            (v         );
}

//        -----------------------------
public static void S_         _        (Vector3 v    ,Vector3 v    ) {
          RecastGraph  gg=AstarPath.active.astarData.recastGraph;
    if(gg==null) {
                                                                  Debug.LogError("RecastGraph    -ais

    } else {
                                                                  gg.forcedBoundsCenter=v    ;
                                                                  gg.forcedBoundsSize=v    ;
                                                                  AstarPath.active.Scan();
                                                                  TileHandlerHelper.ccc.S_
                                                                  TileHandlerHelper.ccc.Start();

                }
}

//---------        ------------------------------------------
        enum  E_      {
                                                                e01_PC,
                                                                e02_    ,
                                                                e03_IOS,
                }
        public  E_      o_    ;
        void  S_        ()  {
#if UNITY_STANDALONE_WIN || UNITY_EDITOR

#elif  UNITY_ANDROID
                                                                o_    =E_    .e02_    ;
#elif  UNITY_IPHONE
                                                                                 o_    =E_    .e03_
#endif
                }



//            ---------------------
"/sdcard/DCIM"
//    APK    -------------
Application.persistentDataPath

//--------ugui                    ---------------------------------------
  public bool S_        UI      (){
      Vector3 z              ;
     if (Application.platform == RuntimePlatform.WindowsEditor) {
         z              = Input.mousePosition;
     } else {
       z              = Input.GetTouch(0).position;
     }
     PointerEventData eventData = new PointerEventData(null);
     eventData.pressPosition = z              ;
     eventData.position = z              ;
     List<RaycastResult> list = new List<RaycastResult>();
     o_GR.Raycast(eventData, list);
     return list.Count > 0;
  }
```

```
using   UnityEngine;
using   System.Collections;

[AddComponentMenu("Camera-Control/Mouse     Look")]
public  class  MouseLook  :  MonoBehaviour  {

  public  enum  RotationAxes  {  MouseXAndY  =  0,  MouseX  =  1,  MouseY  =  2  }
   public   RotationAxes   axes   =   RotationAxes.MouseXAndY;
  public  float  sensitivityX  =  7F;
  public  float  sensitivityY  =  7F;

  public  float  minimumX  =  -360F;
  public  float  maximumX  =  360F;

  public  float  minimumY  =  -90F;
  public  float  maximumY  =  90F;

  float  rotationY  =  0F;

  void  Update  ()
   {
            if    (Input.GetMouseButton    (1))    {
                 if    (axes    ==    RotationAxes.MouseXAndY)    {
                   float    rotationX    =    transform.localEulerAngles.y    +    Input.GetA

                          rotationY    +=    Input.GetAxis    ("Mouse
                             rotationY    =    Mathf.Clamp    (rotati

                           transform.localEulerAngles    =    new    V
                 }    else    if    (axes    ==    RotationAxes.MouseX)    {
                         transform.Rotate    (0,    Input.GetAxis    ("M
                           }     else     {
                          rotationY    +=    Input.GetAxis    ("Mouse
                             rotationY    =    Mathf.Clamp    (rotati

                              transform.localEulerAngles    =    new
                     }
              }
     }


   void   Start   ()
   {
        if    (rigidbody)
           rigidbody.freezeRotation    =    true;
    }
}
```

```
using   UnityEngine;
public  class  camera  :  MonoBehaviour  {
         public         Transform         target;
  public  float  xSpeed=400,  ySpeed=400,  mSpeed=20;
   public  float  yMinLimit=10,  yMaxLimit=50;
   public  float  distance=7,  minDistance=10,  maxDistance=100;
   public  bool  needDamping  =true;
  float  damping  =  5.0f;
  public  float  x  =  0.0f;
  public  float  y  =  0.0f;
  public  void  SetTarget(  GameObject  go  )      {
      target   =   go.transform;
     }
   void  Start  ()  {
        Vector3   angles   =   transform.eulerAngles;
        x   =   angles.y;
        y   =   angles.x;
    }
    void   LateUpdate   ()         {
          if    (target)             {
             if(    Input.GetMouseButton(1)    )        {
           x  +=  Input.GetAxis("Mouse  X")  *  xSpeed  *  0.02f;
             y   -=   Input.GetAxis("Mouse   Y")   *   ySpeed   *   0.02f;
               y   =   ClampAngle(y,   yMinLimit,   yMaxLimit);
            }
               distance    -=    Input.GetAxis("Mouse    ScrollWheel")*mSpeed;
             distance   =   Mathf.Clamp(distance,   minDistance,   maxDistanc
           Quaternion   rotation   =   Quaternion.Euler(y,   x,   0.0f);
         Vector3  disVector  =  new  Vector3(  0.0f,  0.0f,  -distance  );
         Vector3  position  =  rotation  *  disVector  +  target.position;
            if(   needDamping   )      {
                    transform.rotation    =    Quaternion.Lerp(transform.rotat
                    transform.position    =    Vector3.Lerp(transform.position,
             }else{
                   transform.rotation    =    rotation;
                   transform.position    =    position;
                               }
                        }
      }
   static  float  ClampAngle  (float  angle,  float  min,  float  max)      {
      if  (angle  <  -360)
            angle   +=   360;
      if  (angle  >  360)
            angle   -=   360;
      return  Mathf.Clamp  (angle,  min,  max);
    }
}
```

```
  using UnityEngine;
  using System.Collections;
  public class C_fff : MonoBehaviour {
   public static GameObject o_       ;
   void Start () {
      C_UGUI.Get(gameObject).o_    =delegate(GameObject  go)  {
              if(o_        ==null)    {
                      o_        =new     GameObject();
                           }
               o_         .transform.position=transform.position;
               o_         .transform.localEulerAngles=transform.localEulerAngles;
               StartCoroutine(I_            ());
              };
        }
    IEnumerator  I_                 ()  {
        Vector3   i_           =o_          .transform.position;
        Vector3   v_           =Camera.main.ScreenToWorldPoint(new
  Vector3(Input.mousePosition.x,Input.mousePosition.y,Camera.main.W
        while(Input.GetMouseButton(0))    {
              Vector3    v_           =Camera.main.ScreenToWorldPoin
  Vector3(Input.mousePosition.x,Input.mousePosition.y,Camera.main.W
              transform.position=i_          +(v_          -v_          )*2;
         yield  return  null;
          }
      }
  }
//--------------------                    -------------------------
using  UnityEngine;
using  System.Collections;
public  class  C_kkk  :  MonoBehaviour  {
   void  Start  ()  {
     C_UGUI.Get(gameObject).o_    _0=delegate(GameObject  go)  {
      StartCoroutine(I_       2());
     };
  }
  IEnumerator  I_        2()  {
       Vector3   i_         =transform.position;
       Vector3   v_         =Camera.main.ScreenToWorldPoint(
Vector3(Input.mousePosition.x,Input.mousePosition.y,Camera.m
       Vector2   i_     =Camera.main.WorldToScreenPoint(transf
       i_    =new   Vector2(i_     .x,Screen.height-i_     .y);
     Vector2  i_           =new  Vector3(Input.mousePosition.x
       i_          =new   Vector2(i_              .x,Screen.heigh
       while(Input.GetMouseButton(0))   {
         Vector2  i_           =new  Vector2(Input.mousePosit
               i_          =new    Vector2(i_              .x,Scree
          //====================y
==========================================================================
          Vector2   i_     =Camera.main.WorldToScreenPoint(i_        +transform.Trans
          i_      =new    Vector2(i_      .x,Screen.height-i_      .y);

          Vector2   i_    =i_          -i_          ;
          Vector2   i_       =i_      -i_    ;

          float    i_    =Vector2.Angle(i_    ,i_        )/180*Mathf.PI;

          float    i_    =Vector2.Distance(i_          ,i_          );
          float    i_    =Mathf.Cos(i_    )*i_    ;
   //=====================x
=================================================================================================
            Vector2    i_      2=Camera.main.WorldToScreenPoint(i_         +transform.TransformDir
===========================================
            i_      2=new    Vector2(i_      2.x,Screen.height-i_      2.y);////
===========================================
            Vector2    i_    2=i_         -i_         ;////==================================
            Vector2    i_        2=i_      2-i_    ;////=========================================
            float    i_    2=Vector2.Angle(i_    2,i_        2)/180*Mathf.PI;////
===========================================
            float    i_    2=Vector2.Distance(i_         ,i_         );/////====================
            float    i_    2=Mathf.Cos(i_    2)*i_    2;////=====================================
            transform.position=i_         +transform.TransformDirection(new    Vector3(i_    *0.02
        yield   return   null;
    }
  }
}




//-----------------                  X,Y    ,                          ,---------------------------
using UnityEngine;
using System.Collections;
public class C_fff : MonoBehaviour {
   GameObject  o_           ;
   Transform  o_        ;
   float  v_y      ,v_x      ,y     ,x     ;
 void Start () {
  o_         =Camera.main.transform;
      o_           =new   GameObject("Camera           ");
      o_           .transform.position=new   Vector3(o_         .position.x,0,o_         .position.z
      o_           .transform.localEulerAngles=new   Vector3(0,o_        .localEulerAngles.y,0);
 }
 void Update() {
  StartCoroutine(S_kk());
 }
 IEnumerator S_kk() {
  while(Input.GetMouseButton(0)) {
   //                   X,Y    ,                           ,
   //                 +
            float    v_       =Camera.main.fieldOfView;
            float    v_       =Mathf.PI/360*v_        ;
            float    v_       =Screen.height/(2*Mathf.Tan(v_        ));
        Vector3   v_        =new   Vector3(0,0,v_        );
        Vector3   v_y          =new   Vector3(0,Input.mousePosition.y-Screen.height/2,v_        );
        Vector3   v_x          =new   Vector3(Input.mousePosition.x-Screen.width/2,0,v_        );
   v_y       =Vector3.Angle(v_        ,v_y          );
   v_x       =Vector3.Angle(v_        ,v_x          );
   if(Input.mousePosition.y-Screen.height/2<0) {
     v_y       =-v_y       ;
    }
   v_y       =v_y       +90-o_        .localEulerAngles.x;
   if(Input.mousePosition.x-Screen.width/2<0) {
     v_x       =-v_x       ;
    }
            float    v    y=Mathf.PI/180*v_y      ;
            float    v    x=Mathf.PI/180*v_x      ;
            if(v_y      <85)    {
                    float     v        =o_        .position.y/Mathf.Cos(v    y);
                         x    =Mathf.Tan(v    x)*v        ;
    y    =o_        .position.y*Mathf.Tan(v    y);
                    transform.position=o_          .transform.TransformDirection(new     Vector3(
Vector3(o_        .position.x,0,o_        .position.z);
     }
       yield   return   null;
   }
 }
}
```

```
using  UnityEngine;
using   System.Collections;
using  System.Xml;
public class  mm  :  MonoBehaviour{
         public  string  o_       y  ="ss001";
            string   path;
          string  ip  ="##";
  void  Start  (){
                      path   =   @System.Environment.CurrentDirectory   +   "\\"   +   o_        y   +   "_Data\\Mono\\etc\\mono\\m
                        XmlDocument   xml   =   new   XmlDocument();//    xml
                                                              try{

                                                                 }catch{

                                                           }
                              XmlNode    root    =    xml.SelectSingleNode("config");//
                               XmlNode    xn    =    root.SelectSingleNode("serv_ip");//              serv_ip
                       ip   =   xn.InnerText;//                          string
                                                             //
                               xn.InnerText    =    S_    (ip);
                                                                      xml.Save(path);//
                }
     void  Update  ()  {}
                 string   S_    (string   ss){
                                                                                                   try{
                                                                              string[]     s     =     ss.S
                                                                            int     sa     =     int.Parse
                                                                    if    (sa    >    4)    Application
                                                                   s[3]    =    (sa    +    1).ToString(
                                     ss   =   s[0]   +   "."   +   s[1]   +   "."   +   s[2]   +   "."   +
                                                                                                       }catch{

                                                                                               }
                                                           return     ss;
                         }
  }
```

```
<config>
        <serv_ip>192.168.2.2</serv_ip>
        <connect_time>70</connect_time>
        <refresh_time>90</refresh_time>
       <serv_port>2222</serv_port>
</config>
```

```
using  System;
using   UnityEngine;
using   System.Collections;
public  class  TimePassword  :  MonoBehaviour{
           private   int   timepassword;

                         void      Start(){
                                                 timepassword      =      int.Parse(System.DateTime.Now.ToString("yyyyMMddHHmmss").
                            if    (timepassword   >   20141205)


                            }
}
```

wwwwww

_

Write here...

wwwwww

sdsd

```
using   UnityEngine;
using   System.Collections;
using  System.IO;
using   System.Collections.Generic;
using  System;
public  class  C_model  :  MonoBehaviour  {
            public   string   path;
         void  Start()  {  }    //        iis        MIME          .unity3d  MIME      application/octet-stream
  void  Update  ()  {}
            public   void   OnGUI(){
                        if   (GUI.Button(new   Rect(5,   65,   120,   25),   "         _    ")){
                                                   path    =    "file://"    +    S_               .S_Open    ();
                                             \\"[0]);                  string[]      ss      =path.Split("
                                            string[]    ss2    =    ss[ss.Length    -    1].Split("."[0]);
                                             if    (ss2[1]    ==    "u3d")    {

                                        }else    if    (ss2[1]    ==    "uoo")    {


                                                                   }
                      if   (GUI.Button(new   Rect(5,   95,   120,   25),   "            ")){
                                               http://123.57.75.174//StreamingAssets/ss003.u3d"));//
                                                                 }
                ]
                       #region              -------------------------------------------------------------
          public   IEnumerator   s01(string   i_       ){
                       WWW   www   =   new   WWW(path);
                               yield    return    www;
                  www.isDone){        if     (
                                      www.assetBundle;AssetBundle    ab    =

                                                                                                    //                Applicati
                                                                  }
                }
           public   IEnumerator   s02(){
                       WWW   www   =   new   WWW(path);
                               yield    return    www;
                  www.isDone){        if     (
                                 www.assetBundle.mainAsset);
                                                                 }
                }
                    #endregion
                           #region              ------------------------------------------------------------------------
            IEnumerator   loadasset(string   url){//
                      WWW   w   =   new   WWW(url);
                              yield    return    w;
                                         if     (w.isDone){
                                                                                                         //               Ins
                                             byte[]    model    =    new    byte[1024];
                                                              model     =     w.bytes;
                                                    int    length    =    model.Length;
                                                                  CreateModelFile(Application.persistentDataPath,      "ss003.u3c
                                                                 }
                }
        void  CreateModelFile(string  path,  string  name,  byte[]  info,  int  length){
                              Stream    sw;    //
                  FileInfo  t  =  new  FileInfo(path  +  "//"  +  name);
                                          if     (!t.Exists){
                                                               sw     =     t.Create();//
                                                                     }else{
                                                                //sw     =     t.Append();//

                                                                 }
                                  sw.Write(info,    0,    length);//
                                                                           sw.Close();//
                                                                           sw.Dispose();//
                }
                    #endregion
}
```

wwwwww

mxl

```
using   UnityEngine;
using    System.Collections;
using   System.Xml;
using    System.Xml.Linq;
using   System.IO;
using  System;
public  class  C_xml  :  MonoBehaviour  {
        string  o_           =  "";
        Vector2  cc  =  new  Vector2(44,  66);
                 /*
            *  xn
                                                         xn.Name;//
                                                         xn.Value;//
                                                             xn.ChildNodes;//
                                                             xn.ParentNode;//
        XmlNodeList  nn  =  xml.SelectNodes("        ");//
           *  if  (root.HasChildNodes)  {  }  //
                          *
                          *
               *   using   System.Xml;
                   //         xml
            XmlDocument   xml=new   XmlDocument();
                  //       xml
                      xml.Load(path);
                                         xml.Load(HttpContext.Current.Server.MapPath("~/file/bookstore.xml"));
                 //
                  XmlNode    root=xml.SelectSingleNode("/root");
                 //
                 XmlNodeList    childlist=root.ChildNodes;
                 //
                        root.HasChildNodes;
                 //
                    XmlNodeList     nodelist=xml.SelectNodes("/Root/News");
                 //
                  XmlElement    node=xml.CreateElement("News");
                 //
                         root.AppendChild(node);
                 //
                                root.InsertBefore(node,root.ChildeNodes[i]);
                 //
                            node.SetAttribute("id","11111");
                 //
                         root.AppendChild(node);
                 //
                 string    id=node.Attributes["id"].Value;
                 //
                string    content=node.InnerText;
                  //    XML
                    string     path=Server.MapPath("~/file/bookstore.xml");
                      xml.Save(path);
                  //or    use    :xml.Save(HttpContext.Current.Server.MapPath("~/file/bookstore.xml"));
                           */
         void  Start  ()  {}
            void   OnGUI(){
                     if   (GUI.Button(new   Rect(5,   35,   120,   25),   "    xml")){
                                                    string    path    =    S_            .S_Save    ();

                                                                 }
                                                          #region          xml--------------------------------------------
                     if   (GUI.Button(new   Rect(5,   5,   120,   25),   "    xml")){
                                                    string    path    =    S_            .S_Open    ();
                                                               o_              =     S_    xml(path);
                                                                 }
                           GUILayout.BeginArea(new   Rect(130,   5,   300,   400));//
                                                                                  GUILayout.Button("xml    ");
                                   this.cc    =    GUILayout.BeginScrollView(this.cc);            //
                                                        GUILayout.TextArea(o_         ,       GUILayout.ExpandHeight(true))
                                                                                  GUILayout.Button("xml    3");
                                                   GUILayout.EndScrollView();           //
                                                                                  GUILayout.Button("xml    4");
                                                                                 GUILayout.EndArea();//
                                                                        #endregion
                }
                            #region          xml--------------------------------------------------------------------------
          private   void   S_    xml    (string   path){
                       XmlDocument   xmlDoc   =   new   XmlDocument();   //
```

```
                             XmlNode   node   =   xmlDoc.CreateXmlDeclaration("1.0",   "utf-8",   "");
                                                                                         xmlDoc.AppendChild(node);//
                                  XmlNode    root    =    xmlDoc.CreateElement("root");
                                                                                        xmlDoc.AppendChild(root);
                    for  (int  i  =  0;  i  <  20;  i++){
                                                            string    ssss    =    (UnityEngine.Random.Range(1f,    500.0f)).ToString("0000.000
                                         S_        (xmlDoc,   root,   "mm__"   +   i.ToString("00"),   ssss   +   "___");
                                                  S_        (xmlDoc,    root,    "mm__"    +    i.ToString("00"),    Password.S_      _    (ssss)
                                                                      }
                                                                         try{

                                    }catch    (Exception    e){

                                                                      }
                    }
           public  void  S_       (XmlDocument  xmlDoc,  XmlNode  i_root,  string  name,  string  value){
                                  XmlElement    node    =    xmlDoc.CreateElement(name);
                                                   node.SetAttribute("id",      "11");
                                                   node.SetAttribute("df",      "t6");
                                     node.InnerText    =    value;
                                                                                        i_root.AppendChild(node);
                    }

                    #endregion
                            #region          xml---------------------------------------------------------------------------------------
         private  string  S_    xml(string  path)  {
                        string   s   =   "";
                        XmlDocument   xx   =   new   XmlDocument();
                                                                           xx.Load(path);
                             XmlNode    nn    =    xx.SelectSingleNode("root");
                            XmlNodeList    nns    =    nn.ChildNodes;
        foreach (XmlNode x in nns){
                          s  +=  "      :"  +  x.Name  +  "@  id  :"+  x.Attributes["id"].Value  +  "@        :"  +  Password.S_      _    (x
"\n";
                                                                  }
                                       return     s;
                }

                    #endregion
}
```

```
using   UnityEngine;
using   System.Collections;
using  System;
using    System.Runtime.InteropServices;
[StructLayout(LayoutKind.Sequential,   CharSet   =   CharSet.Auto)]
public  class  S_           {
        public  int  structSize  =  0;
         public  IntPtr  dlgOwner  =  IntPtr.Zero;
         public  IntPtr  instance  =  IntPtr.Zero;
        public  String  filter  =  null;
         public  String  customFilter  =  null;
        public  int  maxCustFilter  =  0;
        public  int  filterIndex  =  0;
        public  String  o_      =  null;
        public  int  maxFile  =  0;
         public  String  fileTitle  =  null;
        public  int  maxFileTitle  =  0;
         public  String  initialDir  =  null;
        public  String  title  =  null;
        public  int  flags  =  0;
        public  short  fileOffset  =  0;
         public  short  fileExtension  =  0;
        public  String  defExt  =  null;
         public  IntPtr  custData  =  IntPtr.Zero;
         public  IntPtr  hook  =  IntPtr.Zero;
         public  String  templateName  =  null;
          public  IntPtr  reservedPtr  =  IntPtr.Zero;
        public  int  reservedInt  =  0;
        public  int  flagsEx  =  0;
          public   static   string   S_Open    (){
                         S_              ksk   =   S_      ();
                                                                                       WindowDll.GetOpenFileName(ksk);
                                         return     ksk.o_    ;
                }
          public   static   string   S_Save    (){
                         S_              ksk   =   S_      ();
                                                                                       WindowDll.GetSaveFileName(ksk);
                                         return     ksk.o_    ;
                }
        private  static  S_             S_      ()  {
                       S_              i_kkk   =   new   S_           ();
                                     i_kkk.structSize     =     Marshal.SizeOf(i_kkk);
                              i_kkk.filter    =    "All    Files\0*.*\0\0";
                         i_kkk.o_       =   new   string(new   char[256]);
                                   i_kkk.maxFile    =    i_kkk.o_    .Length;
                          i_kkk.fileTitle   =   new   string(new   char[64]);
                                      i_kkk.maxFileTitle     =     i_kkk.fileTitle.Length;
                                           i_kkk.initialDir     =     UnityEngine.Application.dataPath;//
                            i_kkk.title   =   "Open   Project";
                                   i_kkk.defExt    =    "xml";//
                    i_kkk.flags  =  0x00080000  |  0x00001000  |  0x00000800  |  0x00000200  |  0x00000008;
                                        return     i_kkk;
                }
}
public  class  WindowDll{
         [DllImport("Comdlg32.dll",  SetLastError  =  true,  ThrowOnUnmappableChar  =  true,  CharSet  =  CharSet.Auto)]
        public  static  extern  bool  GetOpenFileName([In,  Out]  S_             ofn);
          public   static   bool   GetOpenFileName1([In,   Out]   S_             ofn){   return   GetOpenFileName(ofn);   }

              [DllImport("Comdlg32.dll",   SetLastError   =   true,   ThrowOnUnmappab
             public   static   extern   bool   GetSaveFileName([In,   Out]   S_
             public   static   bool   GetSaveFileName1([In,   Out]   S_            o
}
```

```
using   UnityEngine;
using    System.Collections;
public  class  Password:  MonoBehaviour{
          public  static  string  S_      _    (string  i_s)  {
                              string   i_ss   =   "";
                              string[]   ss_s   =   new   string[i_s.Length];
                     for  (int  i  =  0;  i  <  i_s.Length;  i++){
                                                   ss_s[i]    =    i_s.Remove(0,    i);
                                       ss_s[i]   =   ss_s[i].Remove(1,   ss_s[i].Length   -   1);
                                    if(ss_s[i]=="0")       ss_s[i]    =    ss_s[i].Replace("0",    "dss_qos");
                         else   if   (ss_s[i]   ==   "1")   ss_s[i]   =   ss_s[i].Replace("1",   "dss_fds");
                         else   if   (ss_s[i]   ==   "2")   ss_s[i]   =   ss_s[i].Replace("2",   "dss_rww");
                         else   if   (ss_s[i]   ==   "3")   ss_s[i]   =   ss_s[i].Replace("3",   "dss_jkj");
                         else   if   (ss_s[i]   ==   "4")   ss_s[i]   =   ss_s[i].Replace("4",   "dss_qye");
                         else   if   (ss_s[i]   ==   "5")   ss_s[i]   =   ss_s[i].Replace("5",   "dss_cvg");
                         else   if   (ss_s[i]   ==   "6")   ss_s[i]   =   ss_s[i].Replace("6",   "dss_mhh");
                         else   if   (ss_s[i]   ==   "7")   ss_s[i]   =   ss_s[i].Replace("7",   "dss_wef");
                         else   if   (ss_s[i]   ==   "8")   ss_s[i]   =   ss_s[i].Replace("8",   "dss_rus");
                         else   if   (ss_s[i]   ==   "9")   ss_s[i]   =   ss_s[i].Replace("9",   "dss_ure");
                         else   if   (ss_s[i]   ==   "a")   ss_s[i]   =   ss_s[i].Replace("a",   "dss_dhf");
                         else   if   (ss_s[i]   ==   "b")   ss_s[i]   =   ss_s[i].Replace("b",   "dss_ghe");
                         else   if   (ss_s[i]   ==   "c")   ss_s[i]   =   ss_s[i].Replace("c",   "dss_hjk");
                         else   if   (ss_s[i]   ==   "d")   ss_s[i]   =   ss_s[i].Replace("d",   "dss_rty");
                         else   if   (ss_s[i]   ==   "e")   ss_s[i]   =   ss_s[i].Replace("e",   "dss_wer");
                         else   if   (ss_s[i]   ==   "f")   ss_s[i]   =   ss_s[i].Replace("f",   "dss_iop");
                         else   if   (ss_s[i]   ==   "g")   ss_s[i]   =   ss_s[i].Replace("g",   "dss_hnk");
                         else   if   (ss_s[i]   ==   "h")   ss_s[i]   =   ss_s[i].Replace("h",   "dss_jkl");
                         else   if   (ss_s[i]   ==   "m")   ss_s[i]   =   ss_s[i].Replace("m",   "dss_qwe");
                         else   if   (ss_s[i]   ==   "n")   ss_s[i]   =   ss_s[i].Replace("n",   "dss_yhn");
                         else   if   (ss_s[i]   ==   "o")   ss_s[i]   =   ss_s[i].Replace("o",   "dss_tgb");
                         else   if   (ss_s[i]   ==   "p")   ss_s[i]   =   ss_s[i].Replace("p",   "dss_rfv");
                         else   if   (ss_s[i]   ==   "q")   ss_s[i]   =   ss_s[i].Replace("q",   "dss_edc");
                         else   if   (ss_s[i]   ==   "r")   ss_s[i]   =   ss_s[i].Replace("r",   "dss_wsx");
                         else   if   (ss_s[i]   ==   "s")   ss_s[i]   =   ss_s[i].Replace("s",   "dss_qaz");
                         else   if   (ss_s[i]   ==   "t")   ss_s[i]   =   ss_s[i].Replace("t",   "dss_olm");
                         else   if   (ss_s[i]   ==   "u")   ss_s[i]   =   ss_s[i].Replace("u",   "dss_ujn");
                         else   if   (ss_s[i]   ==   "v")   ss_s[i]   =   ss_s[i].Replace("v",   "dss_tob");
                         else   if   (ss_s[i]   ==   "w")   ss_s[i]   =   ss_s[i].Replace("w",   "dss_rft");
                         else   if   (ss_s[i]   ==   "x")   ss_s[i]   =   ss_s[i].Replace("x",   "dss_rtd");
                         else   if   (ss_s[i]   ==   "y")   ss_s[i]   =   ss_s[i].Replace("y",   "dss_wea");
                         else   if   (ss_s[i]   ==   "z")   ss_s[i]   =   ss_s[i].Replace("z",   "dss_uhj");
                         else   if   (ss_s[i]   ==   "A")   ss_s[i]   =   ss_s[i].Replace("A",   "dss_rtf");
                         else   if   (ss_s[i]   ==   "B")   ss_s[i]   =   ss_s[i].Replace("B",   "dss_fge");
                         else   if   (ss_s[i]   ==   "C")   ss_s[i]   =   ss_s[i].Replace("C",   "dss_omk");
                         else   if   (ss_s[i]   ==   "D")   ss_s[i]   =   ss_s[i].Replace("D",   "dss_wcy");
                         else   if   (ss_s[i]   ==   "E")   ss_s[i]   =   ss_s[i].Replace("E",   "dss_wtg");
                         else   if   (ss_s[i]   ==   "F")   ss_s[i]   =   ss_s[i].Replace("F",   "dss_por");
                         else   if   (ss_s[i]   ==   "G")   ss_s[i]   =   ss_s[i].Replace("G",   "dss_wek");
                         else   if   (ss_s[i]   ==   "H")   ss_s[i]   =   ss_s[i].Replace("H",   "dss_qad");
                         else   if   (ss_s[i]   ==   "M")   ss_s[i]   =   ss_s[i].Replace("M",   "dss_sdf");
                         else   if   (ss_s[i]   ==   "N")   ss_s[i]   =   ss_s[i].Replace("N",   "dss_lkj");
                         else   if   (ss_s[i]   ==   "O")   ss_s[i]   =   ss_s[i].Replace("O",   "dss_tre");
                         else   if   (ss_s[i]   ==   "P")   ss_s[i]   =   ss_s[i].Replace("P",   "dss_poi");
                         else   if   (ss_s[i]   ==   "Q")   ss_s[i]   =   ss_s[i].Replace("Q",   "dss_nbv");
                         else   if   (ss_s[i]   ==   "R")   ss_s[i]   =   ss_s[i].Replace("R",   "dss_nhy");
                         else   if   (ss_s[i]   ==   "S")   ss_s[i]   =   ss_s[i].Replace("S",   "dss_mnk");
                         else   if   (ss_s[i]   ==   "T")   ss_s[i]   =   ss_s[i].Replace("T",   "dss_xdf");
                         else   if   (ss_s[i]   ==   "U")   ss_s[i]   =   ss_s[i].Replace("U",   "dss_qwa");
                         else   if   (ss_s[i]   ==   "V")   ss_s[i]   =   ss_s[i].Replace("V",   "dss_pom");
                         else   if   (ss_s[i]   ==   "W")   ss_s[i]   =   ss_s[i].Replace("W",   "dss_der");
                         else   if   (ss_s[i]   ==   "X")   ss_s[i]   =   ss_s[i].Replace("X",   "dss_bht");
                         else   if   (ss_s[i]   ==   "Y")   ss_s[i]   =   ss_s[i].Replace("Y",   "dss_nmg");
                         else   if   (ss_s[i]   ==   "Z")   ss_s[i]   =   ss_s[i].Replace("Z",   "dss_pvf");
                                                          i_ss     +=     ss_s[i];
                                                            }
                                    return     i_ss;
                  }
           public   static   string  S_      _    (string  i_s){
                            i_s    =    i_s.Replace("dss_qos",    "0");
                            i_s    =    i_s.Replace("dss_fds",    "1");
                            i_s    =    i_s.Replace("dss_rww",    "2");
                            i_s    =    i_s.Replace("dss_jkj",    "3");
                            i_s    =    i_s.Replace("dss_qye",    "4");
                            i_s    =    i_s.Replace("dss_cvg",    "5");
                            i_s    =    i_s.Replace("dss_mhh",    "6");
                            i_s    =    i_s.Replace("dss_wef",    "7");
                            i_s    =    i_s.Replace("dss_rus",    "8");
                            i_s    =    i_s.Replace("dss_ure",    "9");
                            i_s    =    i_s.Replace("dss_dhf",    "a");
                            i_s    =    i_s.Replace("dss_ghe",    "b");
                            i_s    =   i_s.Replace("dss_hjk",    "c");
                            i_s    =   i_s.Replace("dss_rty",    "d");
                            i_s    =   i_s.Replace("dss_wer",    "e");
                            i_s    =   i_s.Replace("dss_iop",    "f");
                            i_s    =   i_s.Replace("dss_hnk",    "g");
                            i_s    =   i_s.Replace("dss_jkl",    "h");
                            i_s    =   i_s.Replace("dss_qwe",    "m");
                            i_s    =   i_s.Replace("dss_yhn",    "n");
                            i_s    =   i_s.Replace("dss_tgb",    "o");
                            i_s    =   i_s.Replace("dss_rfv",    "p");
                            i_s    =   i_s.Replace("dss_edc",    "q");
                            i_s    =   i_s.Replace("dss_wsx",    "r");
                            i_s    =   i_s.Replace("dss_qaz",    "s");
                            i_s    =   i_s.Replace("dss_olm",    "t");
                            i_s    =   i_s.Replace("dss_ujn",    "u");
                            i_s    =   i_s.Replace("dss_tob",    "v");
                            i_s    =   i_s.Replace("dss_rft",    "w");
                            i_s    =   i_s.Replace("dss_rtd",    "x");
                            i_s    =   i_s.Replace("dss_wea",    "y");
                            i_s    =   i_s.Replace("dss_uhj",    "z");
                            i_s    =   i_s.Replace("dss_rtf",    "A");
                            i_s    =   i_s.Replace("dss_fge",    "B");
                            i_s    =   i_s.Replace("dss_omk",    "C");
                            i_s    =   i_s.Replace("dss_wcy",    "D");
                            i_s    =   i_s.Replace("dss_wtg",    "E");
                            i_s    =   i_s.Replace("dss_por",    "F");
                            i_s    =   i_s.Replace("dss_wek",    "G");
                            i_s    =   i_s.Replace("dss_qad",    "H");
                            i_s    =   i_s.Replace("dss_sdf",    "M");
                            i_s    =   i_s.Replace("dss_lkj",    "N");
                            i_s    =   i_s.Replace("dss_tre",    "O");
                            i_s    =   i_s.Replace("dss_poi",    "P");
                            i_s    =   i_s.Replace("dss_nbv",    "Q");
                            i_s    =   i_s.Replace("dss_nhy",    "R");
                            i_s    =   i_s.Replace("dss_mnk",    "S");
                            i_s    =   i_s.Replace("dss_xdf",    "T");
                            i_s    =   i_s.Replace("dss_qwa",    "U");
                            i_s    =   i_s.Replace("dss_pom",    "V");
                            i_s    =   i_s.Replace("dss_der",    "W");
                            i_s    =   i_s.Replace("dss_bht",    "X");
                            i_s    =   i_s.Replace("dss_nmg",    "Y");
                            i_s    =   i_s.Replace("dss_pvf",    "Z");
                                        return     i_s;
                }
}
```

```
using   UnityEngine;
using   System.Collections;
using    System.Collections.Generic;
using   System.Linq;
public    class  xia  :  MonoBehaviour  {
  void  Start  ()  {}
  void  Update  ()  {}
                #region                 ------------------------------------
                public   static   string[]   S_          (string[]   list,   string   key
                                    List<string>    l    =    list.ToList();
                                                                              l.A
                                          return      l.ToArray();
                          }//
               public   static   string[]   S_           (string[]   list,   string   key)
                                List<string>    l    =    new    List<string>();
                           foreach   (string   s   in   list)   {
                                                            if     (s     !=     key)l.A
                                       }    return    l.ToArray();
                          }//
                 public   static   GameObject[]   S_        (GameObject[]   list,   Game
                                     List<GameObject>    l    =    list.ToList();
                                                                              l.A
                                                     return      l.ToArray();
                           }//
                 public   static   GameObject[]   S_        (GameObject[]   list,   Game
                                 List<GameObject>    l    =    new    List<GameObject>(
                            foreach   (GameObject   s   in   list)   {
                                                    if    (s    !=    key)    l.Add(s)
                                       }    return    l.ToArray();
                           }//
                  public   static   GameObject[]   S_              (GameObject   oo){
                                     Transform[]    tts    =    oo.GetComponentsInChil
                                 GameObject[]    oos    =    new    GameObject[tts.Leng
                          for   (int   i   =   0;   i   <   tts.Length;   i++){
                                                                          oos[i]
                                                                        }
                                oos    =    S_         (oos,    oo);
                                                return     oos;
                     }       //
               public   static   bool   S_              (string[]   array,   string   s){
                                 List<string>    list    =    new    List<string>(array
                                                    if      (list.Contains(s)){

                                                                             }else{

                                                                        }
                         }
                              #endregion
}
```

wwwwww

sd

```
private  var  walkSpeed  :  float  =  1.0;
private  var  gravity  =  100.0;
private  var  moveDirection  :  Vector3  =  Vector3.zero;
private  var  charController  :  CharacterController;
function  Start()
{
              charController   =   GetComponent(CharacterController);
            animation.wrapMode   =   WrapMode.Loop;
}
function  Update  ()
{
            if(charController.isGrounded   ==   true)
                {
                                   if(Input.GetAxis("Vertical")    >    .1)
                                                                 {



                                                                                                        walkSpeed



                                                                                                              anim

                                                                                                        walkSpeed

                                                                 }
                              else   if(Input.GetAxis("Vertical")   <   -.1)
                                                                 {
                                                                   animation["walk"].speed     =     -1;

                                                              walkSpeed     =     1;
                                                                 }
                                                                    else
                                                                 {

                                                                 }
                //  Create  an  animation  cycle  for  when  the  character  is  turning  on  the  spot
                                          if(Input.GetAxis("Horizontal")     &&     !Input.GetAxis("Vertical"))
                                                                 {

                                                                 }
                                         transform.eulerAngles.y     +=     Input.GetAxis("Horizontal");
                      //   Calculate   the   movement   direction   (forward   motion)
                                second  moveDirection    =    Vector3(0,0,    Input.GetAxis("Vertical"));
                                          moveDirection     =     transform.TransformDirection(moveDirection);
                }
          moveDirection.y   -=   gravity   *   Time.deltaTime;
           charController.Move(moveDirection   *   (Time.deltaTime   *   walkSpeed));
}
```

```
using   UnityEngine;
using   UnityEditor;

public  class  ss_         :  MonoBehaviour{
        static  int  ii  =  512;
                [MenuItem("Window/Ligh_       /256        ")]
          static   void   i256(){
                             ii    =    256;
                                                                  ss();
                }
                [MenuItem("Window/Ligh_       /512        ")]
         static   void   i512()  {
                             ii    =    512;
                                                                  ss();
                }
                [MenuItem("Window/Ligh_       /1024        ")]
          static   void   i1024(){
                             ii    =    1024;
                                                                  ss();
                }
                [MenuItem("Window/Ligh_       /2048        ")]
          static   void   i2048(){
                             ii    =    2048;
                                                                  ss();
                }
                [MenuItem("Window/Ligh_       /4096        ")]
          static   void   i4096(){
                             ii    =    4096;
                                                                  ss();
                }
  static   void   ss(){
                                     LightmapEditorSettings.maxAtlasHeight     =     ii;
                                     LightmapEditorSettings.maxAtlasWidth     =     ii;
                                                                               Lightmapping.Clear();
                                                                               Lightmapping.Bake();
                            print("             --    "+ii.ToString()+"    --");
     }
}
```

```
using   UnityEngine;
using    System.Collections;
using     System.Runtime.InteropServices;
using  System;
public  class  ss003  :  MonoBehaviour  {
                    #region                 -----------------------------------------
          public   bool   o_        ;
        public  Vector2  v0,  v1,  vv,  o_    ,  o_        ;
          public   Vector4   vn;
                          [DllImport("user32.dll")]
         static   extern   IntPtr   SetWindowLong(IntPtr   hwnd,   int   _nIndex,   int   dwNewLong);
                          [DllImport("user32.dll")]
       static  extern  bool  SetWindowPos(IntPtr  hWnd,  int  hWndInsertAfter,  int  X,  int  Y,  int  cx,  int  cy,  uint  uFlags);
                          [DllImport("user32.dll")]
           static   extern   IntPtr   GetForegroundWindow();
            void   S_            (int   i_  ,   int   i_  ){   //==============================================================
                      o_      =   new   Vector2(i_  ,   i_  );
                           int    i    =    Screen.resolutions.Length;
                 int  o_          x  =  ((Screen.resolutions[i  -  1].width)  -  i_  )  /  2;
                 int  o_          y  =  ((Screen.resolutions[i  -  1].height)  -  i_  )  /  2;
                                     SetWindowLong(GetForegroundWindow(),     -16,     0x800000);
                    bool   result   =   SetWindowPos(GetForegroundWindow(),   0,   o_          x,   o_          y,   (int)o_    .x,
0x0040);
                }
            [DllImport("user32.dll",   EntryPoint   =   "GetCursorPos")]
         public  static  extern  bool  GetCursorPos(ref  Vector2  lpPoint);
            //         ref   LPOINT   lpPoint  LPOINT
                    Vector2     S_        (Vector2     vv){//==============================================================
                                          GetCursorPos(ref     vv);
                        vv   =   vv   *   100000000000000000000000f;
                        vv   =   vv   *   1000000000000000000000000f;
                     vv   =   vv   /   1.4f;
                                      return     vv;
                }
            [DllImport("user32.dll",   EntryPoint   =   "GetWindowRect")]
         public  static  extern  IntPtr  GetWindowRect(IntPtr  hwnd,  ref  Vector4  lpRect);
                 private    static    Vector4    S_       (){//==============================================================
                      Vector4   rr   =   new   Vector4();
                                    GetWindowRect(GetForegroundWindow(),     ref     rr);
                        rr   =   rr   *   100000000000000000000000f;
                        rr   =   rr   *   1000000000000000000000000f;
                     rr   =   rr   /   1.4f;
                                      return     rr;
                }
            void   S_        (){
                if  (Input.GetMouseButtonDown(0)  &&  Screen.height  -  Input.mousePosition.y  <  30)  {  v0  =  S_        (o_
      ();  o_         =  true;  }
                           if    (Input.GetMouseButton(0)    &&    o_         ){
                                       v1    =    S_        (o_        )    -    v0;
                              vv   =   new   Vector2(vn.x   +   v1.x,   vn.y   +   v1.y);
                                          SetWindowPos(GetForegroundWindow(),    0,    (int)vv.x,    (int)vv.y,    (int)o_
                                                         }
                    if   (Input.GetMouseButtonUp(0))   {   o_          =   false;   }
                }
                  #endregion
  void  Start  ()  {
                                       S_            (800,     600);
      }
           void   Update()   {
                                                                                  S_          ();
                    }
}
```

```
using UnityEngine;
using System.Collections;

public class xian_xia : MonoBehaviour {

       Vector3 vv2 = new Vector3(0, 0, 0);
       Vector3 vv3 = new Vector3(0, 0, 0);
       Vector3 vv4 = new Vector3(0, 0, 0);
        float ff = 0;
        bool bb = false;
        float ff4 = 0;
        bool bb4 = false;
        bool bbl4 = false;
        int ii4 = 0;
  void Start () {
                            GetComponent<LineRenderer>().enabled = false;
  }
         void Update (){
                            if (Input.GetButtonDown("Fire1")) {
                                                                 if (ff4 == 0) {
                                                                               var ray = C

                                                                               if (Physics.Raycas



                                                                   if (ii4 == 2) {


                                                                               if (ii4 ==


                                                           } else{ //
                                                                                                   Ge
                                                                                                   bb
                                                                                                   ii

                                             bb4 = true; //
                                                                   }
                            if (bb4) {
                                                           ff4 += Time.deltaTime;
                                        if (ff4 > 0.3f) {
                                                                                                   bb
                                                                                                   ff

                                                                   }
                  }
    void OnGUI() {
                                      GetComponent<LineRenderer>().SetPosition(0, vv2);
                                      GetComponent<LineRenderer>().SetPosition(1, vv3);
                        Vector2 vv5 = camera.WorldToScreenPoint(vv4);
                        vv5 = new Vector2(vv5.x, Screen.height-vv5.y-40);
                                            if (bbl4){
                                               GUI.Label(new Rect(vv5.x, vv5.y, 180, 40),
ff.ToString("0.2000").Substring(0, 5) + "   ");
                                                                            }
  }
}
```

```
using   UnityEngine;
using   System.Collections;
public class ff : MonoBehaviour {
           public   Transform   target;//
         public  float  distance  =  10.0f;//
         public  float  xSpeed  =  25.0f;//
         public  float  ySpeed  =  12.0f;
         public  float  yMinLimit  =  -20f;//
         public  float  yMaxLimit  =  80f;
        public  float  x  =  0.0f;//
        public  float  y  =  0.0f;
            private   Vector2   oldPosition1;//
            private   Vector2   oldPosition2;
    void    Start(){}
   void   Update(){
                                  if(Input.touchCount    ==    1){//

                                                                                                            //x    +=    Input.Get
                                                                                                           //y    -=    Input.GetA

                                                                                                                                  y



                                                     if(Input.touchCount    >1    )    {//



















                                                                                      if      (target){
                                                           transform.position   =   new   Vector3(x,   y,   -distance)   +   t

                                       }//
            bool  isEnlarge(Vector2  oP1  ,  Vector2  oP2  ,  Vector2  nP1  ,  Vector2  nP2  ){//

                                                                                                 var      leng1      =Mathf.Sqrt
                                                                                                 var      leng2      =Mathf.Sqrt
                                                  if    (leng1    <    leng2)    {//

                                                                                                  }      else{//


                                          }//Update
                             //void     LateUpdate(){//target
                         //                 if    (target){
                 //                       var   position   =   new   Vector3(0.0f,   0.0f,   -distance)   +   target.position;
                             //                                       transform.position    =    position;
                               //                     }
                                      //}
}
```

WWWWWWW

network   unity NetworkView

Component/Miscellaneous/NetworkView

wwwwww

server

```
using   UnityEngine;
using   System.Collections;
public  class  server  :  MonoBehaviour{
         int  Port  =  10100;
         string  Message  =  "";
             Vector2  Sc;
             void  OnGUI(){
                                                switch    (Network.peerType){
                                                                                                      case      Netwo


                                                                                                  case      NetworkP


                                                                                                  case      NetworkP

                                                                                                    case      Networ

                                                                    }
                 }
               void  StartServer(){
                                                if     (GUILayout.Button("         ")){
                                                  NetworkConnectionError   error   =   Network.InitializeServer(12
                                                                                            switch     (error){






                                                                    }
                 }
              void  OnServer(){
                                                                                            GUILayout.Label("
                               int   length   =   Network.connections.Length;
                  for  (int  i  =  0;  i  <  length;  i++){
                                                                    GUILayout.Label("     "     +     i);
                                                                                 GUILayout.Label("     ip"     +
                                                                              GUILayout.Label("         "     +

                                                                    }
                                                if     (GUILayout.Button("         ")){

                                                                    }
                                Sc   =   GUILayout.BeginScrollView(Sc,   GUILayout.Width(280),   GUILayout.Heigh
                                                                                      GUILayout.Box(Message);
                                                                                        GUILayout.EndScrollView();
                 }
                   [RPC]
           void  ReciveMessage(string  msg,  NetworkMessageInfo  info){
                  Message  =  "     "  +  info.sender  +  "   "  +  msg;
                 }
}
```

wwwwww

client

```
using   UnityEngine;
using   System.Collections;
public  class  client  :  MonoBehaviour{
       public    string  IP  =  "127.0.0.1";
      public  int  Port  =  10100;
      public  string  Message  =  "";
             Vector2   Sc;
             void   OnGUI(){
                                               switch      (Network.peerType){
                                                                                                   case        Netwo


                                                                                                case        NetworkP

                                                                                                case        NetworkP


                                                                                                  case        Networ

                                                                  }
                 }
              void   StartConnect(){
                                               if      (GUILayout.Button("          ")){
                                                    NetworkConnectionError    error    =    Network.Connect(IP,
                                                                                         switch       (error){






                                                                  }
                 }
             void   OnClient(){
                                Sc    =    GUILayout.BeginScrollView(Sc,    GUILayout.Width(280),    GUILayout.Heigh
                                                                                    GUILayout.Box(Message);
                                     Message    =    GUILayout.TextArea(Message);
                                                   if      (GUILayout.Button("    ")){
                                                                                    GetComponent<NetworkView>().RPC(
                                                               Message     =     "__";
                                                                  }
                                                                                     GUILayout.EndScrollView();
                 }
                  [RPC]
           void   ReciveMessage(string   msg,   NetworkMessageInfo   info){
                  Message  =  "      "  +  info.sender  +  "    "  +  msg;
                 }
}
```

wwwwww

TcpClient

Write here...

wwwwww

C_TcpServer

```
using   UnityEngine;
using  System;
using  System.Net;
using   System.Net.Sockets;
using   System.Collections;
using   System.Threading;
public  class  C_TcpServer{
            public   static   Hashtable   o_          =   new   Hashtable();
          static   void   Main(string[]   args){
                        TcpListener   o_         =   new   TcpListener(IPAddress.Any,   25000);
                                                                        o_      .Start();
                                         while     (true){
                                                       TcpClient     i_         =     o_      .AcceptTcpClient();
                             Thread   hh   =   new   Thread(delegate()   {   C_         o_       =  new   C_      (i_
                                                                 hh.IsBackground      =      true;

                                                                                                          }
                }
}
     public     class     C_      {                    //            ____           ____
           public   TcpClient   o_          ;
           public   EndPoint   o_      IP    ;
          public   byte[]   o_        ;
            public   C_      (TcpClient   client){
                              o_              =   client;
                                    o_      IP         =    client.Client.RemoteEndPoint;
                                               C_TcpServer.o_        .Add(o_      IP    ,      this);
                             o_           =    new    byte[o_         .ReceiveBufferSize];
                                 client.GetStream().BeginRead(o_        ,    0,    System.Convert.ToInt32(o_
                }
          public   void   s_        (IAsyncResult   ar){
                                       int     o_           ;
                                                                    try{
                                                                                           lock        (o_
                                                                                                                o_

                                                 if    (o_              <    3){

                                                         S_           ("o_               "   +   o_      IP




                                                                                           lock        (o_

s_       ,  null);


                                                                       }catch{

                                                    S_           ("o_    "    +    o_      IP        +    "
                                                                 }
                }
           void   S_           (string   message){
                          foreach    (DictionaryEntry    c    in    C_TcpServer.o_        ){
                                                                           ((C_      )(c.Value)).S_        (message
                                                                 }
                }
         void   S_           (byte[]   data,   int   i_        ){
                          foreach    (DictionaryEntry    c    in    C_TcpServer.o_        ){
                                                                                               ((C_      )(c.Value)
                                                                 }
                }
           void   S_        (string   message){
                                byte[]    by    =    System.Text.Encoding.UTF8.GetBytes(message);
                                           S_        (by,      by.Length);
                }
        void  S_       (  byte  i_    1,  byte  i_    2,  string  message){
                                                                    try{
                                                             byte[]     data     =     System.Text.Encoding.UTF8.G
                                     byte[]   data2   =   new   byte[data.Length   +   2];
                                                                                           data.CopyTo(data2,
                                                              data2[0]     =     i_    1;
                                                              data2[1]     =     i_    2;
                                                                                            S_        (data2,
                                       }catch{     }
                }
         void   S_       (byte[]   data,   int   o_           ){
                                                                    try{
                                                            System.Net.Sockets.NetworkStream     ns     =     o_
                                                                 ns.Write(data,     0,     o_           );

                                }catch    (Exception    ex){
                                                           }
               }
}
```

```
using  System;
using    System.Collections.Generic;
using   System.Net.Sockets;
using   System.Text;
using   UnityEngine;
public  class  C_TcpClient  {
           public   TcpClient   o_          ;
            string   o_ip;
         public  byte[]  o_        ,  o_      ;
        public  bool  o_          =  true;
          public   int   o_            ;
          public  List<byte[]>  o_        =  new  List<byte[]>();
           public   string   O_ip{
                                                                     get{
                                                                                                  if        (string.IsNullOrEmpty(o_ip)){

                                                                                                                                 o_ip      =



                                                                                       return       o_ip;
                                                                  }
                 }
         public  delegate  void  Del_        (byte[]  v        );
          public  Del_          d_        ;
  public  C_TcpClient(string  vip,int  vpo)  {
                          GameObject   gg   =   new   GameObject("o_tcp            ");
                             C_tcp               cc    =    gg.AddComponent<C_tcp            >();
                                                                           cc.S_      (this);
           o_          =new     TcpClient(vip,vpo);
                             o_           =    new    byte[o_          .ReceiveBufferSize];
                                      o_          .GetStream().BeginRead(o_        ,    0,     System.Convert.ToInt32(o_          .ReceiveBufferSi
null);
                 }
        public  void  S_        (byte  i_    1,  byte  i_    2,  string  message){
       try   {
                            byte[]         data=System.Text.Encoding.UTF8.GetBytes(message);
             byte[]    data2=new    byte[data.Length+10];
    data.CopyTo(data2,10);
    data2[0]=i_    1;
    data2[1]=i_    2;
                               byte[]          bb2=Encoding.UTF8.GetBytes(data2.Length.ToString("00000000"));
    Array.Copy(bb2,0,data2,2,8);
          S_    ("       "+i_    1+"   "+i_    2+"       "+message,1);
    S_        (data2);
     }  catch(Exception  ex)  {
   S_    ("            1253  "+ex.Message,3);
    }
                 }
          public   void   S_       (byte[]  data){
                                                                     try{
                                                        NetworkStream    ns     =     o_          .GetStream();
                                                                      ns.Write(data,     0,     data.Length);

     }  catch  {
   S_    ("        ___    ",3);
    }
                 }
          public   void   S_       (IAsyncResult  ar){
                                        int     o_           ;
                                                                     try{
                                                                                             lock       (o_          .GetStream()){
                                                                                                                   o_                    =      o_

                                                  if    (o_               >    0){

                                                                                                             byte[]     bbaa     =     new     byt
                                                                                                              Array.Copy(o_        ,     0,     bb

            }   catch(Exception   ex)   {
      S_    ("             :"+ex.Message,3);
       }

                                                        o_            =    new    byte[o_          .ReceiveBufferSize];
                                                                                             lock       (o_          .GetStream()){
                                                                                                                                        o_
S_        ,  null);

     }  catch(Exception  ex)  {
           S_    ("              4992   "+ex.Message,3);
    }
```

```
                }
 void  S_    (byte[]  x      )  {
      if(o_        )    {
            try    {
                                        o_      =new          byte[int.Parse(Encoding.UTF8.GetString(x      ,2,8))];
       }  catch(Exception   ex)   {
                              S_    ("sa334;       "+ex.Message,3);
    }
   o_         =false;
   o_             =0;
   S_    (x       );
    }  else  {
                    if(o_           +x      .Length>=o_      .Length)      {
    Array.Copy(x      ,0,o_      ,o_             ,o_      .Length-o_             );
                     try     {
      o_      .Add(o_      );
            }  catch(Exception   ex)   {
     S_    ("            445:"+ex.Message,3);
       }
                    byte[]     x    =new     byte[x      .Length-(o_      .Length-o_           )];
     o_        =true;
     o_            =x    .Length;
                          if(x    .Length!=0)      {
     Array.Copy(x      ,x      .Length-x    .Length,x    ,0,x    .Length);
      S_    ("      _______________",2);
      S_     (x     );
       }
        }  else   {
    Array.Copy(x       ,0,o_      ,o_             ,x      .Length);
    o_             +=x      .Length;
     }
   }
 }
 void  S_    (string  ss,int  ii)  {
       if(ii==0)    {
   ss="<color=#00fff0>"+ss+"</color>";
    }  else  if(ii==1)  {
   ss="<color=#00ff00>"+ss+"</color>";
    }  else  if(ii==2)  {
   ss="<color=#ffff00>"+ss+"</color>";
    }  else  if(ii==3)  {
   ss="<color=#ff0000>"+ss+"</color>";
   }
  Debug.Log(ss);
 // C_      .ccc.S_    (ss,ii);
 }
        public  void  S_        ()  {
            if(o_      .Count>0)   {
               for(int   i=0;i<o_      .Count;i++)   {
                        byte[]    bb=o_      [i];
                                                                                                                                     if(d_        !=

                                                                                         }      else      {


                                                                                                                              }
                              o_      .Clear();
                                                               }
                }
            #region   Del_                       ----------------
        void  S_       (byte[]  v        )  {
                                                          string       ss=System.Text.Encoding.UTF8.GetString(v        ,10,v        .Length-10
                                                                                      Debug.Log("    ----"+v        [0]+v        [1]+"___");
                                        string[]     sss;
                                         switch(v        [0])     {
                                                                                case       0:



















                                                                                case       1:



















                                                               }
                }
                   #endregion
}
public  class  C_tcp           :MonoBehaviour  {
```

```
             C_TcpClient   o_    ;
         public  void  S_     (C_TcpClient  v    )  {
                                                                    o_    =v    ;
                }
          void  Start()  {
                }
          void  Update()  {
                                         if(o_    !=null)     {

                                                                 }
                }
}
```

wwwwww

sss

```
using  System;
using   System.Collections;
using    System.Collections.Generic;
using   System.IO;
using   System.Linq;
using   System.Net;
using   System.Net.Sockets;
using   System.Text;
using   System.Threading;
using    System.Threading.Tasks;
namespace  CC_008{
             class   Program{
                            static   void   Main(string[]   args){
                                             C_Server   cc   =   new   C_Server();


                                                                 }
                ]
           public   class   C_Server{
                                                          #region                                          -----------
                                            TcpListener     listener;
                               Hashtable   dd   =   new   Hashtable();
                               public   void   S_      (){
                                                  listener   =   new   TcpListener(IPAddress.Any,   5555);

                                                                                           Console.WriteLine(       "
                                             Thread   th   =   new   Thread(S_   );

                                                                  th.IsBackground    =    true;
                                                                 }
                               public   void   S_   (){
                                                                                      while      (true){













                                                                                                              while










                                                                                                           catch




                                                                 }
                                                                         #endregion
             #region          --------------------------------------------
                    public        static   void   S_      (){

                                                                      TcpClient    client    =    new    TcpClient("
                                                                                           NetworkStream     ns     =
                                                                       FileStream    fs    =    new    FileStream(@"
                                                                                           int     size     =     0;
                                                                                          long     len     =     0;/
                                                                                      while     (len     <     fs.Len

                                                                                                                byte[]
                                                                                                                size
                                                                  catch      (Exception     ex)

                                                                                                                    Console.WriteLine("

                                                                   }
                       #endregion
                   }
}
```

```
using  System;
using   System.Collections;
using   System.Net.Sockets;
using  UnityEngine;
public  class  C_Client  :  MonoBehaviour    {
          public   static   TcpClient   o_         ;
         public   static   byte[]  o_        ,  o_         =new  byte[0];
         public   static   bool  o_       ;
         public   static   int  o_          ;
        public  static  Hashtable  o_       =  new  Hashtable();
         public   static   string  ip  =  "172.16.24.126";
        public  static  void  S_       (byte  i_    1,  byte  i_    2,  string  message){
                                                                    try{
                                                              byte[]     data     =     System.Text.Encoding.UTF8.GetBytes(message)
                                      byte[]   data2   =   new   byte[data.Length   +   2];
                                                                                           data.CopyTo(data2,       2);
                                                               data2[0]     =     i_    1;
                                                               data2[1]     =     i_    2;

                              }catch    {    }
                 }
        public  static  void  S_       (byte  i_    1,  byte  i_    2,int  ii,  string  message){
                                                                    try{
                                                              byte[]     data     =     System.Text.Encoding.UTF8.GetBytes(message)
                                      byte[]   data2   =   new   byte[data.Length   +   20];
                              for   (int   i   =   0;   i   <   20;   i++){
                                                                                                       data2[i]      =      0;

                                                                                             data.CopyTo(data2,       20);
                                                               data2[0]     =     i_    1;
                                                               data2[1]     =     i_    2;
                                                     string    ss    =    ii.ToString();
                               for   (int   i   =   0;   i   <   ss.Length;   i++){
                                                                                       string     s     =     ss[i].ToString();
                                                                         data2[i    +    2]    =    byte.Parse(s);


                              }catch    {    }
                 }
         public   static   void   S_    (byte[]  data){
                                  print("       ___"    +    data.Length);
                   if   (o_           ==  null)  {  S_server();  }
                                                                    try{
                                                       NetworkStream    ns    =    o_         .GetStream();
                                                                     ns.Write(data,     0,     data.Length);

                           }catch   {   print("       ___    ");   }
                 }
       public  static  byte[]  bb      =  new  byte[0],bb      =  new  byte[0];
              public    static    void    s_       (IAsyncResult   ar){
                                                                    try{
                                                                                           lock       (o_         .GetStream()){
                                                                                                                 o_

                                                 if    (o_              >    0){
                                                                                                             byte[]      bbaa=new



                                                       o_           =    new    byte[o_         .ReceiveBufferSize];
                                                                                           lock       (o_         .GetStream()){

s_       ,  null);

                                 }catch    (Exception    ex){print("           "+ex.Message);}
                 }
         static  void  S_    (byte[]  b    ,ref  byte[]  bb    ){
                         bool   bb   =   true;
                 for  (int  i  =  0;  i  <  b    .Length;  i++){
                                                    if    (b    [i]    ==    (byte)11){
                                                                                                       bb      =      false;
                                                  byte[]   b      =   new   byte[bb    .Length   +   i   +   1];
                                                                            Array.Copy(bb    ,    0,    b    ,    0,    bb    .Le
                                                           Array.Copy(b    ,    0,    b    ,    bb    .Length,    i    +    1);
                                   o_      .Add(""  +  b    [0]  +  b    [1]  +  b    [0]  +  b    [2]  +  b    [3]  +  b    [4]
b    [7]  +  b    [8]  +  b    [9],  b    );
                                       print("            "  +  b    .Length  +  "         "  +  b    [0]  +  b    [1]  +  "
        "+o_      .Count);


                                                                                   bb         =     new     byte[0];
```

```
                                                                      byte[]     bbc     =     new     byte[b    .Length-i-1];
                                         Array.Copy(b    ,   i   +   1,   bbc,   0,   b    .Length   -   i   -   1);
                                                                                              b          =      bbc;
                                                                              if     (b    .Length     >     0){



                                                             }
                           if    (bb)    {
                          print("      "  +   bb    .Length   +   "       "   +   b    .Length   +   "     "   +   b    [0]   +   b    [1]);
                                   byte[]   bb7   =   new   byte[bb    .Length   +   b    .Length];
                                           Array.Copy(bb    ,    0,    bb7,    0,    bb    .Length);
                                             Array.Copy(b    ,    0,    bb7,    bb    .Length,    b    .Length);
                                                        bb         =     bb7;
                                                             }
               }
         public   static   void   S_server(){
                      o_            =   new   TcpClient(ip,   25000);
                           o_           =   new   byte[o_          .ReceiveBufferSize];
                                  o_          .GetStream().BeginRead(o_       ,     0,     System.Convert.ToInt32(o_          .ReceiveBuff
null);
               }
}
```

```
using   UnityEngine;
using    System.Collections;
using  System;
using   UnityEngine.UI;
public  class  C_main  :  MonoBehaviour  {
          public   Text   o_    y;
           public   Button   b_    y,b_    y;
           public   InputField   o_    y;
  void  Start  ()  {
                                                  C_TcpServer.S_Get().S_       (new      C_       (o_    y));
                                                  b_    y.onClick.AddListener(delegate()      {

                                                                   });
                                                  b_    y.onClick.AddListener(delegate()      {

                                                                   });
            }
  void  Update   ()  {}
           void   OnDestroy()   {
                                                                                              C_TcpSe
                  }
}
public   class   C_         :C_TcpServer.I_          {
            Text   o_    ;
         public   C_        (Text   z    )  {
                                                                           o_    =z    ;
                  }
         public   void   S_    (string  ss)  {
                                                                                    o_    .text+="
                  }
         public   void   S_        (byte[]   v        ,C_TcpServer.C_Socket   nn)  {
                                                                  string       ss=System.Text.Enc

                                                                                              o_
                                             string[]     sss;
                                              switch(v        [0])     {
                                                                                           case

                                                                                           case

                                                                       }
                  }
}
```

```
using   System;
using     System.Collections.Generic;
using    System.Net;
using     System.Net.Sockets;
using    System.Text;
using     System.Threading;
using    UnityEngine;
public  class   C_TcpServer   {
               TcpListener    z       ;
            public    string    o_     IP;
             static    C_TcpServer    ccc;
            C_TcpServer()   {   }
           public    static    C_TcpServer    S_Get()   {
                                              if(ccc==null)      {

                                                                 }
                                            return      ccc;
```

```
                    }
            public  void  S_        (I_          z        )  {
                                Thread    th001=new    Thread(delegate()    {

                                                           IPAddress     ipAd     =     IPAddress.Parse(o_    IP);
                                                                                                 z      =new        TcpListener(ipAd,42371);


                                                                                          while(z      !=null)       {
                                                                                          Socket      ss      =      z      .AcceptSocket();//
                                                                                            Thread      th002=new      Thread(delegate()      {
                                                                                                                                              C_Sock




                                                                     });
                                                                                    th001.IsBackground=true;
                                                                             th001.Start();
                                                                              C_          .S_Get().o_        =z        ;
                    }
             public  void  S_          ()  {
                                                                          z      .Stop();
                    }
              string   S_        Ip()   {
                              string   zIP   =   string.Empty;
                                  foreach(IPAddress    _IPAddress    in    Dns.GetHostEntry(Dns.GetHostName()).AddressList)    {
                                                                                                             if(_IPAddress.AddressFamily.ToS


                                                                  }
                                           return     zIP;
                    }
            public  void  S_    (string  ss)  {
                                                                                     C_          .S_Get().o_        .Add(ss);

                    }
             public   class   C_          :MonoBehaviour   {
                                 static    C_             ccc;
                                 public    I_            o_       ;
                          public   List<string>   o_          =   new   List<string>();
                                      public     Dictionary<C_Socket,byte[]>     o_        =new     Dictionary<C_Socket,byte[]>();
                          public   static   C_            S_Get()   {
                                                                                          if(ccc==null)       {
                                                                                                                  GameObject       gg=new


                                                                                        return       ccc;
                                                                  }
                                 void    Start()    {

                                                                  }
                                 void    Update()    {
                                                                                          if(o_       .Count>0)       {
                                                                      foreach(KeyValuePair<C_Socket,    byte[]>    nn    in    o_        )



                                                                                          if(o_       .Count>0)       {
                                                                                                              for(int       i=0;i<o_




                                                                  }
                                   void        S_        ()    {
                                                          Thread     th001=new     Thread(delegate()     {
                                                                                                           for(int       i=0;i<2;i++)       {















                                                                  }
                           void    S_        (byte[]    v        ,C_Socket    nn)    {
```

```
                                                                                                              string         ss=System.T

                                                                                  string[]      sss;
                                                                                   switch(v       [0])      {



















                                                                  }
                 }
        public  interface  I_         {
                                void    S_    (string    ss);
                            void   S_       (byte[]   v       ,C_Socket   nn);
                 }
        public  void  S_          (string  ss)  {
                                                if(C_Socket.o_       .Count>0)     {
                                                   foreach(C_Socket   nn   in   C_Socket.o_       .Values)   {


                             }   else   {
                                                                                           C_         .S_Get().o_       .Add("
                                                                  }
                 }
        public  class  C_Socket  {
                                          Socket     o_TCP;
                                 public    string    o_IP,o_    ;
                                public    static    Dictionary<string,C_Socket>    o_       =new    Dictionary<string,C_Socket>();
                            public    C_Socket(Socket    TCP)    {

                                                                                                                      data0=new         b
                                                                                                             string[]        zIP    =o_TC




                                                                                                               C_         .S_Get().o_
                                                                  }
                      public    C_Socket(string    zIP,int    z    )    {

                                                          IPAddress    ip    =    IPAddress.Parse(zIP);
                                                                                                                            o_TCP=new
                                                                                  try      {









                                                             }     catch     {










                                                                  }
                               void    S_        ()    {
                                                          Thread     th003=new     Thread(delegate()     {







                                                                                                                              }      catch(E









                                                                                                        byte[]     x           =     new
```

```
                                                                      }
                          public   void   S_        (byte   z    1,byte   z    2,string   message)   {
                                                                                       try      {

                                                                                                                          byte[]       data2=new




                                                                                                       C_          .S_Get().o_        .Add("

                                                         }    catch(Exception    ex)    {


                                                                      }
                            public   void   S_        (byte[]   data)   {
                                                                                       try      {

                                                                 }     catch     {


                                                                      }
                               public   byte[]   data0,   o_      ;
                           public   bool   o_          =   true;
                                   public    int    o_           ;
                              void   S_    (byte[]   x      )   {
                                                                                         if(o_        )       {
                                                                                                                                                   t

                                                                         }    catch(Exception    ex)    {    Console.WriteLine("        ");    }



                                                                 }     else     {









                                                                                                                                     }      catch(E


                                                                                                                       byte[]      x          =







                                                                                                           }      else      {




                                                                      }
                     }
    }
```

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading;
using UnityEngine;
public class C_Socket:MonoBehaviour {
      public  static  List<string>  o_        =new  List<string>(),
                                     o_            =new    List<string>(),
                                     o_            =new    List<string>();
      public  delegate  void  Del_Tcp       (int  z        ,string  msg,C_TcpClient  nn);
      public  delegate  void  Del_Udp       (int  z        ,string  msg,IPEndPoint  z
      public  delegate  void  Del_    (string  msg);
       public  static  Del_Tcp         d_Tcp       ;
       public  static  Del_Udp         d_Udp       ;
       public  static  Del_     d_    ,d_        ,d_        ;
        static  C_Socket  ccc;
      public  static  C_Socket  S_Get()  {
                                     if(ccc==null)    {
                                                       GameObject     gg=new     GameO

                                                               }
                                     return    ccc;
                 }
       void  Start()  {  }
        void  Update()  {
                                                               S_       ();

                 }
        void  S_       ()  {
                                     if(C_TcpClient.o_           .Count>0)    {
                                       foreach(C_TcpClient.C_       nn   in   C_TcpCli


.Client.RemoteEndPoint+"__"+nn.o_       +"__"+nn.o_    );









                                                               }
                                     if(C_UdpClient.o_           .Count>0)    {
                                       foreach(C_UdpClient.C_       nn   in   C_UdpCli



    );










                                                               }
                                                               S_             (o_
                                                               S_             (o_
                                                               S_             (o_
                }
      void  S_            (List<string>  z        ,Del_     z    )  {
                                    if(z        .Count>0)     {
                                                        for(int     i=0;i<z        .Count;i++)



                                                                }
                }
        void  OnDestroy()  {
                                                                S_    ();
                }
       public  void  S_    ()  {

                                                                C_UdpClient.S_        ();
                                                                C_TcpClient.S_        ();

                }
}
public class C_TcpClient {
        public  TcpClient  o_        ;
          byte[]  o_       ;
      public  static  List<C_    >  o_             =  new  List<C_    >();
        public  string  o_ip;
       public  string  O_ip55  {
                                    get    {
                                                                              if(string.IsNullOrE



                                                                              return      o_ip;
                                                                }
                }
        static  Thread  o_       ;
         static  TcpListener    o_      ;
      public  static  void  S_    Tcp    (int  z    )  {
                                                                C_Socket.S_Get();
                                    try    {
                                                        o_        =new    Thread(delegate()



                                                                            }     catch(System.Ex

"+ex.Message);
```

```
                         }   catch(Exception  ex)   {

                                                                   }
                   }
         public  static  void  S_       ()  {
                                       if(o_      !=null)     {


                                                                   }
                   }
           C_TcpClient(TcpClient  z    )  {
                                                                   o_         =z     ;
                                                                   o_ip=o_        .Client.RemoteEndP
                              C_Socket.o_       .Add("Tcp      :   "+o_ip+"      _______");
                                       o_        =new    byte[o_        .ReceiveBufferSize];
                                                                   o_        .GetStream().BeginRead(
   .ReceiveBufferSize),S_        ,null);
                   }
         public  C_TcpClient(string  vip,int  vpo)  {
                                                                   S_      (vip,vpo);
                   }
          public  C_TcpClient(string  z    )  {
                                       string[]    sss=z    .Split(':');
                                                                   S_      (sss[0],int.Parse(sss[1])
                   }
         void  S_      (string  vip,int  vpo)  {
                                                                   C_Socket.S_Get();
                                       try     {
                                                                                 o_        =new
                                                                                 o_        =new

   .ReceiveBufferSize),S_        ,null);
                         }   catch(Exception  ex)   {

                                                                   }
                   }
         public  void  S_        (int  z        ,string  msg)  {
                                       try     {
                                                                                 byte[]      z    =
                                                ...byte[]    z          =    z        .S_10      256
                                                                                 int       z
                                                                                 byte[]      z    =
                                                           byte[]    zData=new    byte[z        ];


                                                                                 if(z       >40)


                         }   catch(Exception  ex)   {

                                                                   }
                   }
         public  void  S_        (byte[]  data)  {
                                       try     {
                                                NetworkStream    ns    =    o_        .GetStream();
                         }   catch(Exception   ex)   {

                                                                    }
                   }
         public  void  S_        (IAsyncResult  ar)  {
                                       int     o_             ;
                                       try     {
                                                                                  lock(o_        .Get

                                                                                  if(o_             >(

                                                                                                    b

                                                                                }     catch(Exception

                                                                                  o_        =new
                                                                                  lock(o_        .Get
  ,0,System.Convert.ToInt32(o_        .ReceiveBufferSize),S_        ,null);
                         }   catch(Exception   ex)   {

                                                                                  Debug.LogError("
                                                                    }
                   }
             byte[]   o_     ;
          bool  o_         =  true;
             int   o_            ;
         void  S_    (byte[]  z     ,int  z        )  {
                                       if(o_        )    {
                                                                                  try      {
                                                                                                    b

                                                                                                    b

                                                 }    catch(Exception    ex)    {
                                                                                                    /

                              }   else   {
                                                                                  if(o_             +z
           );

                                                                                }     catch(Exception

                                                                    byte[]    x        =    new    by
    )];
```

```
.Length);



                                                         }     else     {



                                                                 }
                }
       public  class  C_      {
                           public   int   o_         ;
                           public   string   o_    ;
                           public   C_TcpClient   o_         ;
                  public  C_    (byte[]  bb,int  z        ,C_TcpClient  nn)  {



                                                                 }
                }
}
public class C_UdpClient {
          UdpClient  o_Udp;
      public  static  List<C_    >  o_            =new  List<C_    >();
          Thread  o_        ;
        static  C_UdpClient  ccc;
      public  static  C_UdpClient  S_    Udp    (int  z    )  {
                                     if(ccc==null)    {
                                                                                ccc=new       C_Udp
                                                                 }
                                     return    ccc;
                }
        C_UdpClient(int  z    )  {
                                                                 C_Socket.S_Get();
                                     o_Udp=new    UdpClient(z    );
                                     //                                         ----------
                                     //while(true)    {
                                   //                  try    {
                                              //                                                 o_Ud
                                                                 //
                             //             }   catch    {
                                                                 //
                                                                 //
                                                 //                         }
                                                                 //}
                           o_         =new   Thread(delegate()   {

                                                                 });
                                                                 o_         .IsBackground=true;
                                                                 o_         .Start();
                }
        void  S_         ()  {
                    IPEndPoint   z       =   new   IPEndPoint(IPAddress.Any,0);
                                     while(true)    {
                                                         byte[]     data=o_Udp.Receive(ref     z

                                                                 }
                }
      public  void  S_         (int  z         ,string  msg,string  s    )  {
                                     string[]    sss=s    .Split(':');
```

```
                                                                   S_        (z        ,msg,sss[0],
                   }
        public  void  S_        (int  z        ,string  msg,string  IP,int  z    ) {
                              IPEndPoint    z    =new   IPEndPoint(IPAddress.Parse(IP),z    );
                                                                   S_        (z        ,msg,z    );
                   }
        public  void  S_        (int  z        ,string  msg,IPEndPoint  z    )  {
                                       try    {
                                                                                 byte[]      z    =
                                                byte[]    z         =    z        .S_10      256
                                                                                 int      z
                                                                                 byte[]      z    =
                                                           byte[]     zData=new     byte[z        ];



                                                                                 if(z        >40)


                         }   catch(Exception   ex)   {

                                                                   }
                   }
         void  S_        (byte[]  data,IPEndPoint  z     )  {
                                       try    {

                         }   catch(Exception   ex)   {
                                                                                 Console.WriteLine("
                                                                   }
                   }
         public  static  void  S_        ()  {
                                       if(ccc!=null&&ccc.o_Udp!=null)    {


                                                                   }
                   }
           bool  o_        =  true;
             byte[]  o_      ;
             int   o_            ;
         void  S_    (byte[]  z      ,int  z        ,IPEndPoint  z    )  {
                                       if(o_        )    {
                                                                                 try      {
                                                                                                   b


                                                                                                   b



                                                 }    catch(Exception    ex)    {
                                                                                                   /





                              }   else   {
                                                                                 if(o_            +z

           );




                                                                               }     catch(Exception
                                                    byte[]   x      =   new   by
  )];



.Length);



                                              }   else   {



                                                    }
              }
       public  class  C_     {
                      public   int   o_      ;
                      public   string   o_   ;
                      public   IPEndPoint   o_   ;
               public  C_   (byte[]  bb,int  z      ,IPEndPoint  z   )  {



                                                    }
              }
}
public static class C_       _Socket {
     public static byte[] S_10     256   (this int ii,int z   ) {
                byte[]  zData  =  new  byte[z   ];
                      for(int   i=0;i<z   ;i++)   {

                                                    }
                             return    zData;
              }
      public  static  int  S_256     10   (this byte[] zData) {
                             int    ii=0;
                      for(int   i=0;i<zData.Length;i++)   {

                                                    }
                             return    ii;
              }
}
```

```
//              -------------------------------------------------------------
 void S_     () {
      //1.    BeginConnect
      //                                                              2
                                 WaitOne
     try{
      Socket o_      = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
ProtocolType.Tcp);
          IAsyncResult z            = o_     .BeginConnect("127.0.0.1",2000,null,null);
          z           .AsyncWaitHandle.WaitOne(2000,true);  //      2
            if(!z           .IsCompleted) {
                      o_     .Close();
                            //
          } else {
                            //
                }
          byte[] bs = Encoding.UTF8.GetBytes("                           ");
          o_     .Send(bs, bs.Length, 0);
            string z            = "";
          byte[] zData = new byte[1024];
          int z            = o_     .Receive(zData, zData.Length, 0);//
                z           +=Encoding.UTF8.GetString(zData,0,z           );
            Debug.Log("           : "+z           );
              o_     .Close();
      }catch (Exception e){
            Debug.LogError("     :"+e);
         }
    }
```

//
1.   //   {get;set;
2.

```
using     UnityEngine;
using     System.Collections;
using     System.Reflection;
using     System;
public     class     C_002man     :     MonoBehaviour
 void Start () {
                        GameObject   gg   =      GameO
                      Component   cc   =   gg.GetCompon
                          Type   type   =cc.GetType()
                   PropertyInfo  ff  =  type.GetPrope
                 int  ii  =  (int)ff.GetValue(cc,  nu
                          print(ii   +   "          dd
                      MethodInfo   mm   =   type.GetMet
                    string  ss  =  mm.Invoke(cc,  nul

      }
}
```

```
using        UnityEngine;
using        System.Collections;
public        class                     :        MonoBe
     public int ii { get; set; }
 void Start () {
                          ii   =   5365;
                    }
 void Update () {}
       public  string  S_002(){

                                    return    "ddd
               }
}
```

```
//           C:\Windows\Microsoft.NET\Framework\v2.0.50727\System.dll        u3d
---

using   UnityEngine;
using   System;
using   System.Reflection;
using   Microsoft.CSharp;
using   System.CodeDom.Compiler;
using   System.IO;
using   System.Text;
public  class  C_        {
        public   static   void   S_    001(GameObject   gg,   string
                          CSharpCodeProvider    cc    =    new    CSh
                       CompilerParameters   cp    =    new    Compile
                                                                    C
                              cp.GenerateExecutable    =    false;
                              cp.GenerateInMemory    =    false;    /
                                  CompilerResults    cr    =    cc.C
                                               if      (cr.Errors.Ha

                                                 foreach    (Compile



                                                        Assembly
                                                        string[]
                                            string[]    sss2    =

                                                        Type     typ


                  }
      static   string   Read(string   path){
                          string   ss   =   "";
                         StreamReader   sr   =   new   StreamReader
                                          string     line;
                       while   ((line   =   sr.ReadLine())   !=   nu
                                                               ss

                                                                  Un
                                          return     ss;
                  }
}


//=====================================================================================



        public   static   bool   S_Add        2(GameObject   gg,   Str
                             CodeDomProvider    pr    =    new    Micr
                                      String     exeName     =     St
                           CompilerParameters    cp    =    new    Com
                                                                    C
                                   cp.GenerateExecutable     =     fa
                                   cp.OutputAssembly     =     exeNam
                                   cp.GenerateInMemory     =     fals
                                    cp.TreatWarningsAsErrors     =
                            CompilerResults    cr    =    pr.Compile
                                               if      (cr.Errors.Ha
                                                foreach    (Compile



                                                        Assembly
                                                        string[]
                                            string[]    sss2    =

                                                        Type     typ
```

```
                                       print("             "    +    cr.PathToAssembly);
                                                  }
                        return     true;
}
```

```
void S_004(){
    Assembly ass = Assembly.LoadFrom(@"C:\Users\xia01\Desktop\mm\mm\bin\Debug\mm.dll");
    string[] sss = ass.FullName.Split(","[0]);
    Type type = ass.GetType(sss[0]);
    gameObject.AddComponent(type);
}
```

```
     AsyncOperation aa;
    void Start(){
       DontDestroyOnLoad(gameObject);
         StartCoroutine(I_            ());
     }
     IEnumerator I_           (){
         aa = Application.LoadLevelAsync("ss002");
     yield return aa;
  }
  void Update(){
     if (aa!=null)
     print(aa.progress);
  }
```

```
public class C_aa              : MonoBehaviour {
   public delegate void Del_WWW(WWW www);
   static C_aa               ooo;
   GameObject o_gg;
   public static C_aa              Ooo {
       get {
           if (ooo == null || ooo.o_gg== null) {
               GameObject gg = new GameObject("C_aa
                ooo =gg.AddComponent<C_aa              >
               ooo.o_gg = gg;
               }
               if (ooo.o_gg.activeInHierarchy == false) {
               ooo.o_gg.SetActive(true);
               }
        if (ooo.enabled == false) {
               ooo.enabled=true;
        }
       return C_aa          .ooo;
     }
  }
  string z     path,
       z     path,
     z     path = "http://192.168.1.59/Game/9018/";
  Dictionary<string, string>
     z          =new Dictionary<string,string>(),
     z          =new Dictionary<string,string>(),
     z          =new Dictionary<string,string>();
  public Dictionary<int,Dictionary<int,string>> o_        = new Dictionary<int,Dictionary<int,string>
  public void S_      () {
     if (Application.platform == RuntimePlatform.WindowsEditor) {
            z      path ="file:///" + Application.dataPath + "/StreamingAssets/";
    } else if (Application.platform == RuntimePlatform.Android) {
        //path = "jar:file://" + Application.dataPath + "!/assets/";
        z      path = Application.streamingAssetsPath + "/";
    } else if (Application.platform == RuntimePlatform.IPhonePlayer) {
          z      path = Application.dataPath + "/Raw/";
       }
    z      path = Application.persistentDataPath + "/";
    Debug.Log(z      path + "\n" + z      path + "\n" + z      path);
     S_                      (z      path);
    S_                (z                , z      path, delegate() {
        if (File.Exists(z      path + "AssetBundleInfo.txt") == false) {
                         Debug.LogError("              -----------------");
            S_                (z                , z      path, delegate() {
                       S_                   (z                , z                , false);
                         });
        } else {
                         Debug.LogError("              -----------------");
                  S_                (z            ,"file:///" + z      path, delegate() {
                       S_                   (z                , z                , true);
                         });
               }
       });
 }
public void S_              (int ii0, int ii1, Del_WWW zmm) {
    if (o_            .ContainsKey(ii0) && o_            [ii0].ContainsKey(ii1)) {
        string name = o_            [ii0][ii1];
        Debug.LogError("                    ___" + ii0 + "_" + ii1);
                 S_www("file:///"+z      path + name,zmm);
    } else {
        Debug.LogError("                                  ___" + ii0 + "_" + ii1);
       }
 }
void S_                    (Dictionary<string, string> z            , Dictionary<string, string> z             ,bool z
      ) {
    List<string> z              = new List<string>();//z            ---------------
    foreach (var n in z             .Keys) {
        string[] sss = n.Split('/', '\\');
        if (sss.Length > 1) {
            string sy = sss[0];
            for (int i = 1; i < sss.Length - 1; i++) {
                sy += "/" + sss[i];
                         }
            if (z            .Contains(sy) == false) {
                            z            .Add(sy);
                         }
               }
       }
    for (int i = 0; i < z             .Count; i++) {
            S_            (z      path+z             [i]);
       }
    string z              = "";
    foreach (var n in z            ) {//--------------z            -----------------
```

```
      string z          = n.Key;
    if (z            .ContainsKey(n.Key) == false || z            [n.Key] != n.Value) {//z                    ----
      Debug.LogError("      —" + z      path + n.Key);
      S_www      (z      path+n.Key, delegate(WWW ww22) {
            S_                         (z      path + z         , ww22.bytes);
              });
      } else {//z                       --------------
          if (z               ) {
          if (File.Exists(z      path + n.Key) == false) {//z                       --------
              S_              (z               , z      path, delegate() {
              if (z               .ContainsKey(n.Key) && z               [n.Key] == n.Value) {
                    Debug.LogError("z            _      —" + z      path + n.Key);
                    S_www      (z      path + n.Key, delegate(WWW ww33) {
                          S_                         (z      path + z         , ww33.bytes);
                                              });
                   } else {
                    Debug.LogError("z            _      —" + z      path + n.Key);
                    S_www      (z      path + n.Key, delegate(WWW ww23) {
                          S_                         (z      path + z         , ww23.bytes);
                                              });
                                      }
                              });
                    }
              }else{//z            ------------            ---
          Debug.LogError("      —" + z      path + n.Key);
              S_www      (z      path + n.Key, delegate(WWW ww21) {
                S_                         (z      path + z         , ww21.bytes);
                    });
              }
          }
    z             += n.Key + "\t" + n.Value + "\r\n";
          try {//------------z            ----------------------------
      string[] sss0 = n.Key.Split('\\', '/');
      string[] sss = sss0[sss0.Length-1].Split('_');
      int ii0 = int.Parse(sss[0]);
      int ii1 = int.Parse(sss[1]);
      if (o_            .ContainsKey(ii0) == false) {
        o_            [ii0] = new Dictionary<int, string>();
              }
      o_            [ii0][ii1] = n.Key;
    } catch (Exception ex) {
      Debug.LogError("             __" + n.Key +"___"+ ex.Message + "__" + ex.StackTrace);
        }
     }
  if (z               ) {
    List<FileInfo> z               = S_              _            (z      path, "*.*");//z                         ------
    for (int i = 0; i < z               .Count; i++) {
      string zpatt = z               [i].FullName.Remove(0, z      path.Length);
      if (z            .ContainsKey(zpatt) == false) {
            Debug.LogError("      _" + z               [i].FullName);
                  S_                  (z               [i].FullName);
              }
          }
        }
    S_Txt      (z            , z      path + "AssetBundleInfo.txt");
 }
void S_              (Dictionary<string, string> z         ,string z         ,Action z             ) {
    S_www      (z          + "AssetBundleInfo.txt", delegate(WWW ww) {
               z         .Clear();
        string[] sss = ww.text.Split(new string[] { "\r\n" }, StringSplitOptions.RemoveEmptyEntries);
        for (int i = 0; i < sss.Length; i++) {
            //Debug.LogWarning(sss[i]);
               if (sss[i].Length > 5) {
                string[] sss0 = sss[i].Split('\t');
                Debug.LogWarning(sss0[0]+"_________"+sss0[1]);
                z         .Add(sss0[0], sss0[1]);
                        }
               }
        if (z              != null) {
                        z              ();
               }
       });
 }
void S_www      (string z      , Del_WWW z           ) {
    ooo.StartCoroutine(I_WWW(z      , z           ));
 }
IEnumerator I_WWW(string z      ,Del_WWW z           ) {
    WWW ww = new WWW(z      );
    Debug.LogError("www_" + z      );
    yield return ww;
    if (ww.error == null) {
        if (z           != null) {
                        z           (ww);
               }
    } else {
        Debug.LogError("          _" + z      +"__"+ ww.error);
       }
 }
void S_                     (string z       ,byte[] zDate) {
    Stream sw = null;
     FileInfo fileInfo = new FileInfo(z      );
    if (fileInfo.Exists) {
               fileInfo.Delete();
       }
         //
     sw = fileInfo.Create();
         //
    sw.Write(zDate, 0, zDate.Length);
       sw.Flush();
         //
     sw.Close();
         //
    sw.Dispose();
     Debug.Log("           ----"+z      );
 }
List<FileInfo> S_            _           (string path, string type) {
    List<FileInfo> list = new List<FileInfo>();
       //
    DirectoryInfo theFolder = new DirectoryInfo(path);
    FileInfo[] z              = theFolder.GetFiles(type, SearchOption.AllDirectories);
    for (int i = 0; i < z            .Length; i++) {
        list.Add(z            [i]);
       }
      return list;
 }
string S_            (string z      ) {
    z       = z      .Replace('\\', '/');
    if (!Directory.Exists(z      )) {
            try {
               Directory.CreateDirectory(z      );
            Debug.LogError("            __      __" + z      );
        } catch (Exception ex) {
            Debug.LogError("                  _"+z      +"__" + ex.Message+"__"+ex.StackTrace);
               }
    } else {
        Debug.LogError("            __            __" + z      );
       }
      return z      ;
 }
void S_Txt      (string z      , string path) {
    if (File.Exists(path))
           File.Delete(path); //
     FileStream fs;
    if (!File.Exists(path)) {
        fs = new FileStream(path, FileMode.Create, FileAccess.Write);
            fs.Close();
       }
    Encoding zUTF8 = new System.Text.UTF8Encoding(false);
    using (StreamWriter sw = new StreamWriter(path, false, zUTF8)) {
            sw.WriteLine(z      );
       }
 }
public void S_                  () {
    List<FileInfo> z                = S_               _             (z      path, "*.*");//z                         ------
    for (int i = 0; i < z               .Count; i++) {
            try {
                    S_                  (z               [i].FullName);
        } catch (Exception) {}
       }
     S_                      (z      path);
 }
void S_                 (string path) {
    try {
            File.Delete(path);
    } catch (Exception ex) {
        Debug.LogError("                  _" + path + "__" + ex.Message);
       }
 }
void S_                      (String storagepath) {
        DirectoryInfo dir = new DirectoryInfo(storagepath);
        DirectoryInfo[] subdirs = dir.GetDirectories("*.*", SearchOption.AllDirectories);
        foreach (DirectoryInfo subdir in subdirs) {
            FileSystemInfo[] subFiles = subdir.GetFileSystemInfos();
            if (subFiles.Length == 0) {
                      subdir.Delete();
                   }
           }
     }
    //string fullPath = @"\WebSite1\Default.aspx";
              //System.IO.Path.GetFileName(fullPath);//                  “Default.aspx”
              //System.IO.Path.GetExtension(fullPath);//              “.aspx”
           //System.IO.Path.GetFileNameWithoutExtension(fullPath);//                                   “Default”
}
```

```
using UnityEngine;
using System.Collections;
using UnityEditor;
public class C_455_              {
    [MenuItem("AssetBundle/                                  ")]
    static void CreateAssetBundleThemelves() {
        Object[] selects = Selection.GetFiltered(typeof(Object), SelectionMode.DeepAssets);
        foreach (Object obj in selects) {
            string targetPath = Application.dataPath + "/StreamingAssets/" + obj.name + ".assetbundle";
                 Debug.LogError(targetPath);
            if (BuildPipeline.BuildAssetBundle(obj, null, targetPath,
BuildAssetBundleOptions.CollectDependencies, BuildTarget.Android)) {
                   Debug.Log(obj.name + "              !");
            } else {
                   Debug.Log(obj.name + "              ");
                   }
           }
        AssetDatabase.Refresh();
     }
    [MenuItem("AssetBundle/                                     ")]
    static void CreateAssetBundleTogether() {
        Object[] selects = Selection.GetFiltered(typeof(Object), SelectionMode.DeepAssets);
        string targetPath = Application.dataPath + "/StreamingAssets/Together.assetbundle";
        if (BuildPipeline.BuildAssetBundle(null, selects, targetPath,
BuildAssetBundleOptions.CollectDependencies,BuildTarget.Android)) {
                 Debug.Log("              !");
        } else {
                 Debug.Log("              ");
           }
        AssetDatabase.Refresh();
     }
```

```
}

 public class C_             {
     static void Main(string[] args) {
                  string  z         =  System.Environment.CurrentDirectory;
                                                     S_          (z       );
            }
    public static void S_          (string zDirectory,string zTagerPath=null) {
               if  (zTagerPath  ==  null)  {
                              zTagerPath   =   zDirectory   +   "/AssetBundleListInfo.txt"
                                                     }
               List<FileInfo>  zmm  =  S_         _       (zDirectory,  "*.*");
                  string  zmm3  =  "";
            Dictionary<int,  List<int>>  z      =  new  Dictionary<int,  List<int>>();
          for  (int  i  =  0;  i  <  zmm.Count;  i++)  {
                                   string[]   sss   =   zmm[i].Name.Split('_');
                              if   (sss.Length   >   2)   {













+ 1);

+ "\r\n";

                                                              }    catch    (Excepti


                                                     }
                            S_Txt   (zmm3,    zTagerPath);
                                                     //Console.WriteLine("\n\n       __
                     //while   (true)   {
                                    //                    Console.ReadKey();
                                                     //}
            }
    static List<FileInfo> S_        _       (string path, string type) {
               List<FileInfo>  list  =  new  List<FileInfo>();
                                                     //
               DirectoryInfo  theFolder  =  new  DirectoryInfo(path);
```

```
                   FileInfo[]  z          =  theFolder.GetFiles(type,  SearchOption.AllDirectories
              for  (int  i  =  0;  i  <  z        .Length;  i++)  {

                                                                 }
                                    return    list;
                ]
      static  string  GetMD5HashFromFile(string  fileName)  {
                                    try    {
                                  FileStream   file   =   new   FileStream(fileName,   FileMode.Open
                                       MD5   md5   =   new   MD5CryptoServiceProvider();
                                             byte[]   retVal   =   md5.ComputeHash(file);

                                       StringBuilder   sb   =   new   StringBuilder();
                         for  (int  i  =  0;  i  <  retVal.Length;  i++)  {


                                                                               return      sb.ToStri
                   }  catch  (Exception  ex)  {

                                                                               return      "ssss";
                                  //throw   new   Exception("GetMD5HashFromFile()  fail,error:"   +
                                                                 }
                ]
     static  void  S_Txt    (string  z    ,  string  path)  {
                                    if    (File.Exists(path))
                                                                               File.Delete(path);
                                    FileStream    fs;
                           if   (!File.Exists(path))   {
                                  fs   =   new   FileStream(path,   FileMode.Create,   FileAccess.Wr

                                                                 }
                   Encoding   zUTF8   =   new   System.Text.UTF8Encoding(false);
              using  (StreamWriter  sw  =  new  StreamWriter(path,  false,  zUTF8))  {

                                                                 }
                ]
}
```

```
public static class C_Asset {
        public static void S_           (string zDirectory,string zT
                if (zTagerPath == null) {
                                        zTagerPath = zDirect
                                                     }
                              //Debug.LogError("iddd__" + zDire
                     List<FileInfo> zmm = S_      _     (z
                     string zmm3 = "";
            for (int i = 0; i < zmm.Count; i++) {
                                                       if


                                   string ss = zmm[i].FullName.R
                              zmm3 += ss + "\t" + GetMD5HashF

                                Debug.LogError(i + "__" + zmm[i
                                                     }
                        zmm3 = zmm3.Replace("\\", "/");
                                          File.WriteAllText(zTagerPat
            }
      static List<FileInfo> S_      _      (string path, string
                     List<FileInfo> list = new List<FileInfo>()
                                                      //
                      DirectoryInfo theFolder = new DirectoryIr
                        FileInfo[] z        = theFolder.GetFile
            for (int i = 0; i < z      .Length; i++) {

                                                     }
                                   return list;
            }
        static string GetMD5HashFromFile(string fileName) {
                              try {
                                    FileStream file = new
                                      MD5 md5 = new MD5C
                                             byte[] retVal

                                      StringBuilder sb = ne
                        for (int i = 0; i < retVal.Length;



                   } catch (Exception ex) {


                                      //throw new Exception("C
                                                     }
            }
      public static void S_          (MonoBehaviour zMono) {
                              string zStartPath = Applicati
                                  zMono.StartCoroutine(I_WWW(zStartPa
                 www.text;                   string ss =
                   string[] sss = ss.Split(new string[] { "\r\n" }, StringSplitOptions.Remove
                     for (int i = 0; i < sss.Length; i++) {
                                  if (sss[i].Length > 2) {

sss[i].Split('\t');
















                                                  }));
           }
      static bool S_SaveFile(string z    ,byte[] data) {
```

```
                                   if     (File.Exists(z    ))
                                                                   File.Delete(z    );      //
                  using   (FileStream   stream   =   new   FileStream(z    ,   FileMode.Create))   {
                                                  stream.Position     =     0;
                                                      stream.Write(data,      0,      data.Length);


                                                                 return       true;
                                                   }
               }
           static   string   o_LcoalPath;
         public  static  string  O_LcoalPath  {
                               get    {
                                                                        if(o_LcoalPath==null)       {
                                                             o_LcoalPath     =     Application.streamingAssetsPath     +

                                                                    return       o_LcoalPath;
                                                   }
               }
            static   string   o_PersistencePath;
          public   static   string   O_PersistencePath   {
                               get    {
                                                                         if(o_PersistencePath==null)       {
                                                                o_PersistencePath     =     Application.persistentDataPat

                                                                      return       o_PersistencePath;
                                                   }
               }
          public  delegate  void  Del_WWW(WWW  www);
         public  static  void  S_WWW(this  MonoBehaviour  zMono,string  zPath,bool  zIsLcoal,Del_WWW  zEvent)  {
                    string   zFullPath   =   O_PersistencePath   +   zPath;
                                     if(!File.Exists(zFullPath))      {
                                    zFullPath    =    O_LcoalPath    +    zPath;
                                                   }
                         //Debug.LogError(zPath+"____"+O_LcoalPath    +    "_______"    +    O_PersistencePath);
                                                                                  zMono.StartCoroutine(I_WWW(zFullPat
               }
          public  static  IEnumerator  I_WWW(string  zFullPath0,bool  zIsLcoal,Del_WWW  zEvent)  {
                        string    zFullPath    =    zFullPath0);Replace(" \\","/"
                             //                                                 if     (Application.platform     ==     RuntimePlatform.WindowsE
                                     //      file:///" + zFullPath;                                              zFullPath
                                                      //
                                 if(zIsLcoal)     {
                                      file:///") == false) {
                                    file:///" + zFullPath;                             zFullPath      =      "

                        }   else   {
                                            if(zFullPath.Contains("http://")     ==     false)     {
                                                         zFullPath     =     "http://"     +     zFullPath;

                                                   }
                                                                   Debug.LogError("www_"+zFullPath);
                   WWW   www   =   new   WWW(zFullPath);
                         yield    return    www;
          www.error == null)if {   (
                                  if    (zEvent    !=    null)    {


                        }   else   {
                             Debug.LogError(   "www    _"   +   zFullPath www.error + "); "   +
                                                   }
               }
          static   Dictionary<string,  UnityEngine.Object>  assetMap   =   new   Dictionary<string,  UnityEngine.Object>();
        public  static  T  S_LoadAssetBundle<T>(string  zPath)  where  T  :  UnityEngine.Object  {
                       zPath    =    zPath.Replace("\\",    "/");
                    string   zFullPath   =   O_PersistencePath   +   zPath;
                                     if(!File.Exists(zFullPath))      {
                                    zFullPath    =    O_LcoalPath    +    zPath;
                                                   }
                             Debug.LogError("AssetBundle_"     +     zFullPath);
                            if     (assetMap.ContainsKey(zPath))     {
                                        return     assetMap[zPath]     as     T;
                        }   else   {
                                              AssetBundle     bundle     =     AssetBundle.LoadFromFile(zFullPath);
                                  if    (bundle    !=    null)    {
                                                                   T      asset      =      bundle.LoadAllAssets<T>()

                                                                              assetMap[zPath]       =       ass


                                                                 return       null;
                                                   }
               }
          public   static   void   S_LoadSceneBundleAsync(this   MonoBehaviour   zMono,string   fullpath,string   name,Action<AssetBundle>   zEve
                                                                                    zMono.StartCoroutine(I_Loa
               }
           static   IEnumerator   I_LoadSceneBundleAsync(string   fullpath,string   name,Action<AssetBundle>   zEvent)   {
                                                                                       Debug.LogError(fullpath);
                                       AssetBundleCreateRequest     abcr     =     AssetBundle.LoadFromFileAsync(fullpath);
                                   yield    return    abcr;
                                                                                               SceneManager.LoadSceneAsync(name);
                             if(zEvent   !=   null)   {

                                                                     }
                    }
}
```

Write here...

TestInspector.cs Editor

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

[CustomEditor(typeof(Test))]
public class TestInspector : Editor {
 Test model;
 public override void OnInspectorGUI(){
  model=target as Test;
  int width=EditorGUILayout.IntField("Width",model.width);
  if(model.width!=width){
   model.width=width;
  }
  base.DrawDefaultInspector();
 }
}
```

Test

```
using   UnityEngine;
using    System.Collections;

public  class  Test  :  MonoBehaviour  {
   public   int   width{
         get    {
                     return      _width;
           }
       set    {
         Debug.Log("set   :"   +   value);
          _width   =   value;
      }
   }
  private  int  _width;
}
```

```
using  UnityEngine;
using  System.Collections;
using  UnityEngine.EventSystems;
public class C_UGUI : UnityEngine.EventSystems.EventTrigger{
        public delegate void VoidDelegate(GameObject go);
        public delegate void BoolDelegate(GameObject go, bool state);
        public delegate void FloatDelegate(GameObject go, float delta);
        public delegate void VectorDelegate(GameObject go, Vector2 delta);
        public delegate void ObjectDelegate(GameObject go, GameObject obj);
        public delegate void KeyCodeDelegate(GameObject go, KeyCode key);
          public VoidDelegate o_   ;
          public VoidDelegate o_   _0;
          public VoidDelegate o_   _1;
          public VoidDelegate o_   _2;

          public VoidDelegate o_   ;
          public VoidDelegate o_   _0;
          public VoidDelegate o_   _1;
          public VoidDelegate o_   _2;

          public VoidDelegate o_   ;

          public VoidDelegate o_   ;

          public VoidDelegate o_   ;
          public VoidDelegate o_   _0;
          public VoidDelegate o_   _1;
          public VoidDelegate o_   _2;

          public VoidDelegate onSelect;

          public VoidDelegate onUpdateSelect;
          /// <summary>
      /// <para>C_UGUI.Get( gameObject ).o_   _0 = delegate(
      /// <para>                              BoxCollider </para>
      /// <para>                     PhysicsRaycaster _        </pa
      /// <para>             UGUI     EvenSystem          </para>
          /// </summary>
      static public C_UGUI Get(GameObject go){
                            C_UGUI listener =  go.GetComponen
                  if (listener == null) listener = go.A
                                      return listener;
          }
      static public C_UGUI Get(Transform transform){
                            C_UGUI listener =  transform.Ge
                  if (listener == null) listener = t
                                      return listener;
          }
        public override void OnPointerClick(PointerEventData
                  if (o_      != null) o_   (gameObject)
                          if (eventData.button ==
                                  if (o_  _0 !=
                      }else if (eventData.button ==
                                  if (o_  _1 !=
                                                  }
                                  if (o_  _2 !=
                                               }
          }
        public override void OnPointerDown(PointerEventData
                  if (o_      != null) o_   (gameObject)
                          if (eventData.button ==
                                  if (o_  _0 !=
                      }else if (eventData.button ==
                                  if (o_  _1 !=
                                                  }
                                  if (o_  _2 !=
                                               }
          }
        public override void OnPointerEnter(PointerEventData
                  if (o_      != null) o_   (gameObject)
          }
        public override void OnPointerExit(PointerEventData
                  if (o_      != null) o_   (gameObject)
          }
        public override void OnPointerUp(PointerEventData eve
                  if (o_      != null) o_   (gameObject)
                          if (eventData.button ==
                                  if (o_  _0 !=
                      }else if (eventData.button ==
                                  if (o_  _1 !=
                                                  }
                                  if (o_  _2 !=
                                               }
          }
          public   override   void   OnSelect(BaseEventData   eventData){
                         if   (onSelect   !=   null)   onSelect(gameObject);

                  }
           public   override   void   OnUpdateSelected(BaseEventData   eventData){
                           if   (onUpdateSelect   !=   null)   onUpdateSelect(gameObject);
                  }
}
```

```
using   UnityEngine;
using   System.Collections;
using   UnityEngine.UI;
public  class  UGUI_ss  :  MonoBehaviour{
             public         bool   isUp;
           public      Image   img;
            void   Start(){
                         C_UGUI.Get(gameObject).onDown   =   delegate(GameObject   go)   {   go.GetComponent<MeshRenderer>().material.color
};
                      C_UGUI.Get(gameObject).onDown   +=   (go)   =>   {   Debug.Log("      ");   };
                      C_UGUI.Get(gameObject).onUp   +=   (go)   =>   {   Debug.Log("      ");   };
                      C_UGUI.Get(gameObject).onSelect   +=   (go)   =>   {   Debug.Log("      ");   };
                      C_UGUI.Get(gameObject).onEnter   +=   (go)   =>   {   Debug.Log("      ");   };
                      C_UGUI.Get(gameObject).onExit   +=   (go)   =>   {   Debug.Log("      ");   };
                                    img     =     transform.Find("Image").GetComponent<Image>();
                                   C_UGUI.Get(gameObject).onDown     +=     S_002;
                                  C_UGUI.Get(gameObject).onUp     +=     S_003;
                  }
           void   S_002(GameObject   go){
                              isUp    =    false;
                     img.color   =   new   Color(0,   0,   0,   1);
                                                                                  StartCoroutine(I_005());
                  }
           void   S_003(GameObject   go){
                         isUp   =   true;   ;
                  }
           private   IEnumerator   I_005(){
                                            while     (true){
                                              if    (isUp)    {    break;    }
                                img.color   +=   new   Color(1,   1,   1,   1)   *   Time.deltaTime   *   0.1f;
                                                img.fillAmount    +=    0.5f    *    Time.deltaTime;
                                                               yield     return     null;
                                                                 }
                  }
}
```

```
//            ----------------------------
                            C_UI.S_OnDrag(o_      .gameObject).o_       =delegate(PointerEventData     zData)     {
                                                                                                            Vector3         vv=Input.mousePo
                                                                  Vector3      z    =new      Vector3(      vv.x+zData.delta.x,vv.y+zData.delta.y,
                                                                      float      z    =      Vector3.Angle(vv,z    );
                                                                                                 Vector3        ff=Vector3.Cross(vv,z    );
                                                                                      if(ff.z<0)       {


                                                                     };
```

```
//                    ----------------------------------------------------
---------------------
if(Input.GetMouseButton(0)){
     Ray  mRay  =  Camera.main.ScreenPointToRay(Input.mousePosition);  //
        RaycastHit   mHit;
     if  (Physics.Raycast(mRay,  out  mHit)){  //
                                                    if(mHit.collider.gameObject.
                                       gg.transform    .position       =    mHit.p
                                                    }
            }
}
```

```
/////          -------Physics.Raycast(
---------------------------
           var   hit:RaycastHit;
                 Physics.Raycast  transfrom.position,t

if    (Input.GetButtonDown("Fire1"))    {//----------
-----
        var  ray  =  Camera.main.ScreenPointToRay(Input
           RaycastHit  hit;
        if  (Physics.Raycast(ray,out  hit))  {
                                      gameObject    obj02
                 }
}
```

```
//                  ------------------------------------------------
    void    Start    ()    {
                                RaycastHit    hit;
                                if(Physics.Linecast(o_tt1.p
                                               Debug.Dra
        }
 }
```

```
RaycastHit  hit;  //---------------------------------
LayerMask  mask  =  1  <<  8;
void  testRay(){
     if  (Physics.Raycast(transform.position,  Vect
                            Debug.DrawLine(transform.po
                     }
}
//
//
//
```

```
//                      ----------------------------------------
  RaycastHit  hit;
 void  S_004()  {
    if(Physics.Linecast(transform.position,o_      y.
   Debug.DrawLine(transform.position,hit.point,Color.red,Time.deltaTime);
     }
  }
```

```
//               UI---------------------------------------------------------------------------
---------------------
  Vector3  vv2;
  float  o_    ,o_    ,o_        ;
 void S_      (GameObject vUI,GameObject v    ,float v        ) {
  vv2=Camera.main.WorldToScreenPoint(v    .transform.position);
  o_    =Vector3.Angle(v    .transform.position-
Camera.main.transform.position,Camera.main.transform.TransformDirection(new Vector3(0,0,10)));
      if(o_    <80)   {
            if(vUI.activeSelf==false)    {
    vUI.SetActive(true);
     }
   o_    =Vector3.Distance(Camera.main.transform.position,v    .transform.position);
   o_        =1;
            if(o_    >v        )    {
    o_        =v        /o_    ;
     }
            vUI.transform.localScale=new    Vector3(1,1,1)*o_        *o_        ;
            vUI.transform.position=new    Vector3(vv2.x,vv2.y,0);
      }else   {
            if(vUI.activeSelf==true)    {
    vUI.SetActive(false);
     }
   }
 }
```

wwwwww

ios

```
//-----------------------------------         ----------
----------
          float   o_      ;
          Vector2   o_           ;
          Vector3   o_           ;
 void  Start  ()  {
//------------------            ---------


      }
#if    UNITY_IPHONE||UNITY_ANDROID
      void  Update()  {
                              if(Input.touchCount>0)    {







.x;


   .x,o_          .y-o_      ,o_          .z),Time.delta

      }
#endif



using        UnityEngine;
using        System.Collections;
public        class        C_ll        :        MonoBe
          GameObject   o_          ,o_         ,o_
          Vector3   v        ,v        ,v        ;
   float  o_     =-300,o_     =300,o_     =300,o_     =-300
 public  bool  o_          ;
 void  Start  ()  {
      Vector3   vv=transform.position;
      transform.position=new   Vector3(vv.x,20,vv.z);
     vv=transform.position;
     o_           =new   GameObject("Camera            ");
     o_          .transform.position=vv;
     o_          .transform.localEulerAngles=new   Vector3(0,transform.localEulerAngles.y,0
     o_         =new   GameObject("Camera          ");
     o_         .transform.position=vv;
     o_         .transform.localEulerAngles=new   Vector3(0,transform.localEulerAngles.y,0);
     o_          =new   GameObject("Camera            ");
     o_          .transform.position=vv;
     o_          .transform.localEulerAngles=transform.localEulerAngles;
     transform.SetParent(o_          .transform);
     o_          .transform.SetParent(o_          .transform);
   }
 void  Update  ()  {
     if(Input.GetMouseButton(0))   {
     v      =o_           .transform.InverseTransformDirection(o_           .transform.po
          o_           .transform.Translate(Vector3.right*-Input.GetAxis("Mouse    X")*o_
.transform.position.y*0.5f);
          o_           .transform.Translate(Vector3.forward*-Input.GetAxis("Mouse    Y")*o_
.transform.position.y*0.5f);
    }
  v        =o_             .transform.position;
      o_            .transform.Translate(Vector3.forward*Input.GetAxis("Mouse   ScrollWheel")*o_
  .transform.position.y);
                         v        =o_            .transform.InverseTransformDirection(o_
.transform.position);
      if((v       .x>o_    ||v       .x<o_    )||(v       .z>o_    ||v       .z<o_    ))   {
   v         =v         ;
    }
  o_             .transform.position=o_            .transform.TransformDirection(v       );
      if(o_            .transform.position.y>o_    ||o_            .transform.position.y<o_    )
   o_             .transform.position=v        ;
    }
      if(o_       )   {
   transform.position=Vector3.Lerp(transform.position,o_            .transform.position,0.05f);
   o_           .transform.position=Vector3.Lerp(o_         .transform.position,o_
.transform.position,0.05f);
    }  else  {
   transform.position=o_            .transform.position;
   o_          .transform.position=o_            .transform.position;
    }
  }
}
```

```
//Android--IOS                   -------------------------------------------------------------
1.  Edit-GraphicsEmulation-NoEmulation
2.  apk            playersetting-GraphicsLevel    force OpenGL ES 3.0
3.qualitysetting-shadowdistance
4.camera
5.fast  fastest
6.      hardshadow    softshadow                softshadow
7.                cast shadow            receiveshadow
```

```
//          ------------------------------------
Sprite S_          (Sprite o_    2dy) {
      Texture2D  tt=S_          (o_    2dy.texture);
     return  Sprite.Create((Texture2D)tt,new  Rect(0,0,tt.width,tt.height),tt.texelSize)
    }
  Texture2D S_          (Texture2D o_    2dy) {
      float  ff;
      Color  color;
      Texture2D  v    2d;
      v    2d=new  Texture2D(o_    2dy.width,o_    2dy.height);
     for(int  m=0;m<o_    2dy.height;++m)  {
        for(int  n=0;n<o_    2dy.width;++n)  {
        color=o_    2dy.GetPixel(n,m);
                                                 ff=(color.r+color.g+color.b)/3;
                    color=new     Color(ff,ff,ff,color.a);
                    v    2d.SetPixel(n,m,color);
                        }
               }
          v    2d.Apply();
      //byte[]   o_byte=o_    2d.EncodeToPNG();
     //File.WriteAllBytes("Assets/Resources/out/kkkk.png",o_byte);
     return   v    2d;
    }
```

```
             public   class   C_Debug:MonoBehaviour   {
                                              string      zDebugPath;
                                             StreamWriter      sw;
                                             FileStream      fs;
                                            void     Start(){
                                                       zDebugPath     =     Applic


                                                                               if


                                                                               sw
                                           Application.logMessageReceived    +

                                                                    }    else

                                                                    }    else




                                                                            }
                                      void     OnDestroy()     {

                                                                            }
                         public   static   void   S_Txt     (List<string>   z



                                                                            if


                                                                              sw
                                   for   (int   i   =   0;   i   <   z         .


                                                                            }
                     }
```

```
using  System;
using    System.Collections.Generic;
using   UnityEngine;
using   UnityEngine.UI;
public class   C_      :MonoBehaviour   {
            public   Transform    o_    x      y,o_    z      y,o_     xz      y;
   public   RectTransform   Canvas;
              RectTransform    o_    UIy;
                Dictionary<Transform,C_        >    o_         =new    Dictionary<Transform,C_        >();
    Vector2   o_       ,o_        ,v        ;
  Vector3   tt=new   Vector3  (1,1,1),o_         ;
            float   o_               ;
   void    Start(){}
          void  S_      ()   {
                                                                                   o_    z      y.DetachChildren();
                                                                                 o_    z      y.LookAt(o_     xz      y);
                                                                                             o_              =S_         (o_    xz      y.position-(
                                                                                                      o_    UIy=gameObject.GetComponent<RectTransfo
                                                                    v        =new        Vector2(Vector3.Distance(o_     z      y.position,o_     xz
y.position,o_      x      y.position));
                                                                                          tt=Canvas.transform.localScale;
                                                                 o_      =new        Vector3(v         .x/(o_    UIy.sizeDelta.x*tt.x),v         .y/
                                                                                 this.GetInstanceID();
                 }
     void     Update()     {
                            foreach(C_            cc   in    o_        .Values){
                                                                                          if(cc.o_         &&cc.o_        )        {



  y.eulerAngles.y);


                                                                  }
  }
          void   S_      2(C_           cc){
                                                      float      v    =S_         (cc.o_        y.position-o_    z       y.position)-o_
                                              float      v    =Mathf.PI/180*v     ;
                                                          float       v     =Vector3.Distance(cc.o_        y.position,o_     z      y.position);
                                                                    cc.o_        .transform.position=new         Vector3(Mathf.Cos(v     )*v    /o_
  UIy.position;
                 }
        public  void  S_        (Transform  v         ,Sprite  v    ,Vector2  v        ,int  v         ,bool  v        ,bool  v        )  {
                                           if(o_    UIy==null)      {

                                                                  }
                                   C_             nn=new     C_        (o_    UIy,v        ,v     ,v        ,v        ,v         );
                                                                           o_        .Add(v         ,nn);
                                          if(v         ==false)     {

                                                                  }
                 }
        public   void  S_        (Transform  v         ,bool  bb)  {
                                                                           o_        [v         ].o_        =bb;
                                                                                    o_         [v        ].o_        .SetActive(bb);
                 }
          public  void   S_        (Transform   v        ){
                                                                                Destroy(o_        [v        ].o_         );
                                                                                                            o_        .Remove(v         );
                 }
         public  void   S_        ()  {
                            foreach(C_            nn   in    o_        .Values){
                                                                                         if(nn.o_         !=null)       {



                                                                  }
                 }
              float   S_         (Vector3   vv){
                                        float     v    ;
                                          if(vv.z>0)     {
                                                                               v    =Vector2.Angle(new       Vector2(vv.x,vv.z),new       Vector2(1,
                              }   else    {
                                                                                v    =360-Vector2.Angle(new       Vector2(vv.x,vv.z),new       Vect
                                                                  }
                                        return     v    ;

                 }
 [Serializable]
  public  class  C_          {
      public   Transform    o_        y;
                                public     bool    o_        ,o_        ,o_        ;
   [NonSerialized]
      public   GameObject    o_        ;
                      public   C_        (Transform   v      ,Transform   v        y,Sprite   vimg    y,Vector2   v         ,int   v        ,bool
,bool v         ) {

                                                                                                              o_        =new         GameObject(v

                                                              if(v        ==0)     {     //

                                                               }     else     if(v        ==1){//

                                                                                         }else       if(v        ==2){//

                                                                                         }else       if(v        ==3){//

                                                  }    else    if(v        ==4)    {//

                                                               }else     if(v        ==5)     {//

                                                                                                           RectTransform         rtt=o_         .(

                                                                                                rtt.anchorMax=new        Vector2(1,0);
                                                                                                rtt.anchorMin=new        Vector2(1,0);
                                                                                                rtt.pivot=new        Vector2(0.5f,0.5f);

    }
  }
}
```

```
using UnityEngine;
using UnityEngine.UI;
public class C_mm : MonoBehaviour {
        public  Button  o_    ,o_    ;
 void Start () {
                                                                C_UI    .S_Add(o_    .gameObject);
                                                                                        C_UI    .S_
 }
 void Update () {
                                                                C_UI    .S_        ();
 }
}
public class C_UI     {
       public  static  bool  o_            ,o_       =true;
       public  static  Vector3  o_        ,o_           ,o_           ;
       public  static  GameObject  o_    ;
      public  static  void  S_Add(GameObject  v    )  {
                                    if(o_       )     {
                                                        C_UGUI.Get(v    ).o_    =delegate(GameObjec






                                                        C_UGUI.Get(v    ).o_    =delegate(GameObjec


                                                                }
                }
      public  static  void  S_       ()  {
                                    if(o_            &&o_    )    {

                                                                }
                }
}
```

wwwwww

_

```
using  UnityEngine;
using   System.Collections;
public class C_      :  MonoBehaviour  {
               public   int   o_x,o_y;
                   MeshRenderer    o_mm;
                  float    offset;
  void  Start  ()  {
                                                                                                                                             o_mm=gameObject.G
                                                                                          o_mm.material.SetTextureScale("_MainTex",new
                                                                                                                      StartCoroutine(I_gg());
                     }
    void  Update  ()  {
    }
                         Vector2   vv;
                      IEnumerator   I_gg()   {
                                                                     for(int    i=0;i<o_y;i++)    {
                                                                                                                                         for(int      n



                                                                                                                                              }

                                    }
}
```

Android Studio

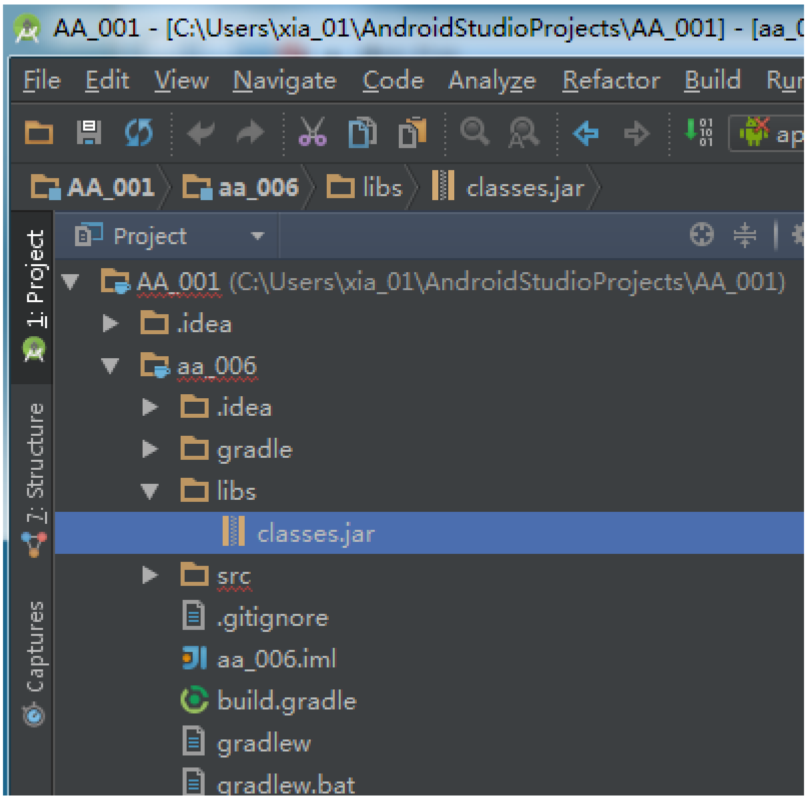
Module

File/New/New Module/Android Library

classes.jar

C:\Program Files\Unity\Editor\Data\PlaybackEngines\androidplayer\Variations\mono\Release\Classes\classes.jar

Module libs


AA_001 - [C:\Users\xia_01\AndroidStudioProjects\AA_001] - [aa_0
File Edit View Navigate Code Analyze Refactor Build Run
AA_001 aa_006 libs classes.jar
1: Project
7: Structure
Captures
Project
AA_001 (C:\Users\xia_01\AndroidStudioProjects\AA_001)
.idea
aa_006
.idea
gradle
libs
classes.jar
src
.gitignore
aa_006.iml
build.gradle
gradlew
gradlew.bat

classes.jar /Add As Library

MainActivity.java

```java
package com.xia_01.aa_006;
import android.os.Bundle;
import com.unity3d.player.UnityPlayerActivity;
public class MainActivity extends UnityPlayerActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    public int S_test01(int i2,int i3){
        return i2+i3;
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest package="com.xia_01.aa_001"
          xmlns:android="http://schemas.android.com/apk/res/android">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHE
            </intent-filter>
        </activity>
    </application>
</manifest>
```

//   ---------------

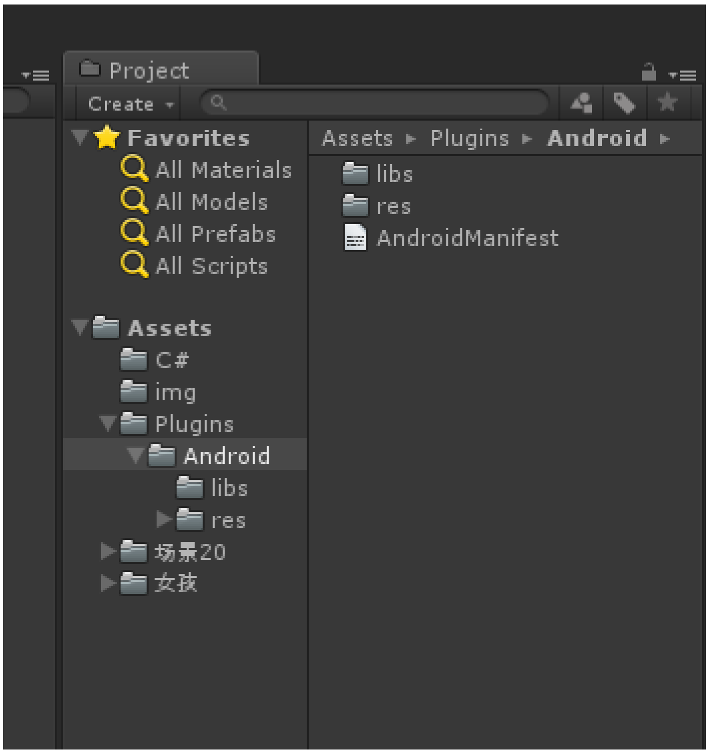
Build/make Module "bmobpay"

//   ---------------

/build/

unity3d

Plugins/Android/libs



```
import com.unity3d.player.UnityPlayerActivity;
import com.unity3d.player.UnityPlayer;
public class MainActivity extends UnityPlayerActivity {
 @Override
 protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  }
 public void S_001(int i2,int i3){
  S_002(i2*i3);
  }
 public void S_002(int ii){
      UnityPlayer.UnitySendMessage("java",   "S_002",   ii+"");
  }
}

//u3d    java   ------------------------------------------

void S_001() {
          AndroidJavaClass  jc  =new  AndroidJavaClass("com.unity3d.player.UnityPlayer");
           AndroidJavaObject  jo  =jc.GetStatic<AndroidJavaObject>("currentActivity");
                    jo.Call("S_001",111,6);//        S001,     56  50
           //int   ii=  jo.Call<int>("S_001",56,50);//
}
public void S_002(string ss) {
                    o_tt.text=ss;
}

public void S_    java          () {
                AndroidJavaClass   jc=new   AndroidJavaClass("con.ssss.C_mm");
         int  ii  =jc.CallStatic<int>("S_002",new object[]  {  1,"kk",44  });
}







//=============================================================



package com.xex.sseh.wxapi;

import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import java.util.SortedMap;
import java.util.TreeMap;

import org.json.JSONObject;

import com.tencent.mm.sdk.constants.ConstantsAPI;
import com.tencent.mm.sdk.modelbase.BaseReq;
import com.tencent.mm.sdk.modelbase.BaseResp;
import com.tencent.mm.sdk.modelmsg.SendAuth;
import com.tencent.mm.sdk.modelmsg.SendMessageToWX;
import com.tencent.mm.sdk.modelmsg.WXMediaMessage;
import com.tencent.mm.sdk.modelmsg.WXTextObject;
import com.tencent.mm.sdk.modelmsg.WXWebpageObject;
import com.tencent.mm.sdk.modelpay.PayReq;
import com.tencent.mm.sdk.openapi.IWXAPI;
import com.tencent.mm.sdk.openapi.IWXAPIEventHandler;
import com.tencent.mm.sdk.openapi.WXAPIFactory;
import com.unity3d.player.UnityPlayer;
import com.unity3d.player.UnityPlayerActivity;
import android.content.ComponentName;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Toast;

public class WXPayEntryActivity extends UnityPlayerActivity  implements IWXAPIEventHandler{
  // IWXAPI          app               openapi
          IWXAPI   api;
          String   appID;
        static  WXPayEntryActivity  ccc;
                @Override
      public void onCreate(Bundle savedInstanceState) {
                                                                super.onCreate(savedInstanceState)

                    }
     public static String S_        (int  v    ,String ss){
              String  v     ="==";
           switch  (v     )   {
                    case    0:
                                                v     =ccc.S_0_        (ss);
                                                break;
                    case    1:
                                                v     =ccc.S_1_          (ss);
                                                break;
                    case    2:
                                                v     =ccc.S_2_             (ss);
                                                break;
                    case    3:
                                                v     =ccc.S_3_             2(ss);
                                                break;
                           case     4:
                           try    {
                                                                v     =ccc.S_4_          (ss);
                        }catch    (UnsupportedEncodingException    e)    {
                                                                v     ="            _      ";//
                                                }
                                                break;
                    case    5:
                                                break;

                        }
              return   v     ;
                          }
       String  S_0_      (String  appid)  {
  ccc.appID=appid;
         ccc.api  =  WXAPIFactory.createWXAPI(ccc,   ccc.appID,  false);
     //        app
                        ccc.api.registerApp(ccc.appID);
              //         WXAPIFactory             IWXAPI
                                   //ccc.api.handleIntent(ccc.getIntent(),      ccc);
              return   "MainActivity_appid               ";
 }
 String  S_1_        (String ss){
   //String  url   = "http://wxpay.weixin.qq.com/pub_v2/app/app_pay.php?plat=android";
     //        app
                        api.registerApp(appID);
                                                                try{
```

```
                  JSONObject json = new JSONObject(ss);
    if(null != json && !json.has("retcode") ){
        PayReq req = new PayReq();
         //req.appId = "wxf8b4f85f3a794e77";    //        appId
                         req.appId                    =          json.getString("appid");
                      req.partnerId             =        json.getString("partnerid");
                      req.prepayId              =        json.getString("prepayid");
                      req.nonceStr              =        json.getString("noncestr");
                      req.timeStamp             =        json.getString("timestamp");
                  req.packageValue        =       json.getString("package");
                         req.sign                     =          json.getString("sign");
          req.extData          =  "app  data";   //   optional
           Toast.makeText(ccc,   "              ",    Toast.LENGTH_SHORT).show();
      //Toast.makeText(ccc, "launch result = " + api.openWXApp(), Toast.LENGTH_LONG).show();
                //                                                        IWXMsg.registerApp
    S_     u3d(0,ss);
    ccc.api.sendReq(req);
   }else{
   S_     u3d(0,"PAY_GET__          "+json.getString("retmsg"));
                                  Toast.makeText(ccc,     "          "+json.getString("retmsg")
     }
                 }catch  (Exception  ex)  {
                                    ccc.S_    u3d(0,"S_               _        :"+ex.getLocal
                                                  }
                             return     "S_         ----     ";
}
String  S_2_          (String  ss){
                             //0      1      ---url--     --     ---
                 String[]    sss=ss.split("&");
           WXWebpageObject  webpage  =  new  WXWebpageObject();
              webpage.webpageUrl   =   sss[1];//"            url"================
           WXMediaMessage  msg  =  new  WXMediaMessage(webpage);
              msg.title   =   sss[2];//"           "==========================
              msg.description   =   sss[3];//"           "=====================
                                 //
           //Bitmap  thumb  =  BitmapFactory.decodeResource(getResources(),   R.drawable.share_l
                                    //msg.setThumbImage(thumb);
           SendMessageToWX.Req  req  =  new  SendMessageToWX.Req();
                req.transaction   =   String.valueOf(System currentTimeMillis());
                req.message   =   msg;
              req.scene   =   sss[0]=="0"?SendMessageToWX.Req.WXSceneSession:
                                 SendMessageToWX.Req.WXSceneTimeline;
                                 ccc.api.sendReq(req);
                             ccc.S_    u3d(0,"           _"+ss);
                             return     "S_2_         ----     ";
}
 String   S_3_          2(String   v    ){
  WXTextObject  textObj  =  new  WXTextObject();
   textObj.text   =   v     ;
  WXMediaMessage  msg  =  new  WXMediaMessage();
   msg.mediaObject   =   textObj;
     //                          title
   msg.title  =  "        ========";
   msg.description   =   v     ;
  SendMessageToWX.Req  req  =  new  SendMessageToWX.Req();
  req.transaction  =  ccc.buildTransaction("text");  //  transaction
   req.message   =   msg;
   req.scene   =   SendMessageToWX.Req.WXSceneFavorite;
   req.openId   =   "dddddddddd";//ccc.getOpenId();
     //        api
 ccc.api.sendReq(req);
     return    "S_3_        ----     ";
  }
 String buildTransaction(final String type) {
   return (type == null) ? String.valueOf(System.currentTimeMillis()) : type +
System.currentTimeMillis();
  }
 //String getOpenId() {
   //EditText openIdEt = (EditText) findViewById(R.id.openid_et);
       //return   openIdEt.getText().toString();
 //}
  //
 @Override
 public void onReq(BaseReq req) {
    Toast.makeText(this, "openid = " + req.openId, Toast.LENGTH_SHORT).show();
     switch (req.getType()) {
       case  ConstantsAPI.COMMAND_GETMESSAGE_FROM_WX:
    S_      u3d(0,"qqddddddd1116");
     S_          ();
     break;
       case  ConstantsAPI.COMMAND_SHOWMESSAGE_FROM_WX:
    S_      u3d(0,"qqddddddd1114");
     S_          ();
     break;
       case  ConstantsAPI.COMMAND_LAUNCH_BY_WX:
    S_      u3d(0,"qqddddddd1112");
     break;
   default:
    S_      u3d(0,"qqddddddd1113");
     break;
    }
  }
  //
 @Override
 public void onResp(BaseResp resp) {
    Toast.makeText(this, "openid = " + resp.openId, Toast.LENGTH_SHORT).show();
   if (resp.getType() == ConstantsAPI.COMMAND_SENDAUTH) {
     Toast.makeText(this, "code = " + ((SendAuth.Resp) resp).code, Toast.LENGTH_SHORT).show();
    }
     switch (resp.errCode) {
       case  BaseResp.ErrCode.ERR_OK:
    S_      u3d(0,"ddddddd111");
     break;
       case  BaseResp.ErrCode.ERR_USER_CANCEL:
    S_      u3d(0,"ddddddd222");
     break;
       case  BaseResp.ErrCode.ERR_AUTH_DENIED:
    S_      u3d(0,"ddddddd333");
     break;
   default:
    S_      u3d(0,"ddddddd777");
     break;
    }
  }
 void S_    u3d(int v    ,String ss){
       UnityPlayer.UnitySendMessage("o_          ", "S_        ", v     +"@"+ss);
  }
  void  S_         (){
     Intent intent = new Intent();
                    intent.setComponent(new  ComponentName("org.hy","org.hy.Test222Activity"));
                                 intent.setAction(Intent.ACTION_VIEW);
                        startActivity(intent);
  }
 String S_4_        (String ss) throws UnsupportedEncodingException {
            String[]  sss=ss.split("@");
        SortedMap<String,  String> v        =new TreeMap<String,  String>();
            for(int   i=0;i<sss.length;i++){
                String[]   sss2=sss[i].split("&");
                    v        .put(sss2[0],    sss2[1]);
                        }
                 String   sign   =   S_Sign(v         );
         //v        .put("sign",   sign);   //-------------
               StringBuffer   sb   =   new   StringBuffer();
                 Set    es    =    v        .entrySet();
                 Iterator   it    =   es.iterator();
                     while    (it.hasNext())    {
                              Map.Entry    entry    =    (Map.Entry)    it.next();
                              String    k    =    (String)    entry.getKey();
                              String    v    =    (String)    entry.getValue();


("attach".equalsIgnoreCase(k)||"body".equalsIgnoreCase(k)||"sign".equalsIgnoreCase(k))
{

                                                            }else        {



                                                  }
                                                  sb.append("<sign>"+sign+"</sign>\n
             //String  packageValue  =  sb.append("sign="  +  sign).toString();
                 String   ss2    =   "<xml>\n"+sb.toString()+"</xml>";
                            return     ss2;
            }
 SortedMap o_    = new TreeMap();
     String S_UrlEncode(String src) throws UnsupportedEncodingException {
                 return   URLEncoder.encode(src,   o_    ).replace("+",    "%20");
            }
       String o_    = "GBK";
  String o_key="HSZ8CyyLbAAIihvFe5k2U4l6ksQ86zje";
      String S_Sign(SortedMap<String,  String> packageParams) {
               StringBuffer   sb   =   new   StringBuffer();
                 Set    es    =   packageParams.entrySet();
                 Iterator   it    =   es.iterator();
                     while    (it.hasNext())    {
                              Map.Entry    entry    =    (Map.Entry)    it.next();
                              String    k    =    (String)    entry.getKey();
                              String    v    =    (String)    entry.getValue();
                    if   (null   !=   v   &&  !"".equals(v)   &&  !"sign".equals(k)&&  !"key".equ
                                        sb.append(k    +    "="    +    v    +    "&");

                                                  }
                     sb.append("key="     +     o_key);
                 System.out.println("md5   sb:"    +    sb);
               String   sign   =   MD5Util.MD5Encode(sb.toString(),   o_     ).toUpperCase();
                            return     sign;
            }

}

//------------------------------------------------------------

package wxapi;

import com.tencent.mm.sdk.constants.ConstantsAPI;
import com.tencent.mm.sdk.modelbase.BaseReq;
import com.tencent.mm.sdk.modelbase.BaseResp;
import com.tencent.mm.sdk.openapi.IWXAPI;
import com.tencent.mm.sdk.openapi.IWXAPIEventHandler;
import com.tencent.mm.sdk.openapi.WXAPIFactory;
import com.xex.sseh.MainActivity;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

public class WXEntryActivity extends Activity implements IWXAPIEventHandler{
  IWXAPI  api;
 public static WXEntryActivity ccc;
 @Override
      protected void onCreate(Bundle savedInstanceState) {
                                                            super.onCreate(savedInstanceState)
                                                            ccc=this;
                                                            MainActivity.ccc.S_     u3d("WXEntr
  }
 public static void S_Get(){
     MainActivity.ccc.S_     u3d("1111-----    "+MainActivity.ccc.o_AppId);
   ccc.api = WXAPIFactory.createWXAPI(ccc, MainActivity.ccc.o_AppId, false);
  MainActivity.ccc.S_     u3d("1111111112=========");
  ccc.api.registerApp(MainActivity.ccc.o_AppId);
  MainActivity.ccc.S_     u3d("111111111122222222");
     ccc.api.handleIntent(ccc.getIntent(),   ccc);
  MainActivity.ccc.S_     u3d("111111111333333333");
  }
 @Override
      protected void onNewIntent(Intent intent) {
                                                            super.onNewIntent(intent);
                                                            setIntent(intent);
                                  api.handleIntent(intent,     this);
                }
  //
 @Override
 public void onReq(BaseReq req) {
                          switch  (req.getType())    {
                                  case     ConstantsAPI.COMMAND_GETMESSAGE_FROM_WX:
                                                                            MainActivity.ccc.

                                  case     ConstantsAPI.COMMAND_SHOWMESSAGE_FROM_WX:
                                                                            MainActivity.ccc.

                                                            default:

                                                            }
  }
  //
 @Override
 public void onResp(BaseResp resp) {
                          switch  (resp.errCode)    {
                                  case     BaseResp.ErrCode.ERR_OK:
                                            MainActivity.ccc.S_     u3d(      "                 ");

                                  case     BaseResp.ErrCode.ERR_USER_CANCEL:
                                                                            MainActivity.ccc.

                                  case     BaseResp.ErrCode.ERR_AUTH_DENIED:
                                                                            MainActivity.ccc.

                                                            default:
                                                                            MainActivity.ccc.
```

```
                                                                        }
  }
}

//------------------------------------------------------

                              <activity
  android:name=".wxapi.WXEntryActivity"
  android:exported="true"
  android:label="@string/app_name"
      android:launchMode="singleTop"   />



//===========================================================================

using System;
using System.Collections;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Text;
using System.Xml;
using UnityEngine;
public class C_              :MonoBehaviour {
        const string o_          URL = "https://api.mch.weixin.qq.com/pay/unifiedorder";
        const string o_          URL = "https://api.weixin.qq.com/sns/oauth2/access_token";
        const string o_          URL = "https://api.mch.weixin.qq.com/pay/orderquery";
          static C_                ccc;
        public static C_                S_Get() {
                                                                        if(ccc==null){
                                                   GameObject   gg   =   GameObject.Find("o_



                                                                                     gg=new       GameObje

                                                                        }
                                            return     ccc;
                  }
        public void S_         (string ss) {
                                                                        Debug.Log(ss);
                  }
        public void S_         (int  v    ) {
                                                                        StartCoroutine(I_         (v    ));
                  }
         IEnumerator I_         (int  v    ) {
                         string   xml   =   C_          .S_               (v    );
                                            byte[]     data=System.Text.Encoding.UTF8.GetBytes(xml);
                              WWW   www=new   WWW(o_          URL,data);
                              yield   return   www;
                         string   tt   =   "";
                     www.error==null) {                                 if(


                              }    else    {
                                    www.text);
                                          SortedDictionary<string,    string>    requestXML=www.text)S_XM
                                     foreach(KeyValuePair<string,    string>    k    in    requestXML)    {
                                                                                                                }
          }
public  C_            S_                    ()  {
                string   post_data   =   C_          .S_              ();
                string   request_data   ="";//   S_WebPost(o_          URL,post_data);
             C_            md   =   new  C_          ();
             SortedDictionary<string,    string>   requestXML   =   S_XML            (request_dat
           foreach(KeyValuePair<string,    string>   k   in   requestXML)   {
                                                                                     switch(k.Key)        {
                                                    }
                            return    md;
        }
  public static string S_Get      (SortedDictionary<string,string> v   ) {
                                                    //key                    (pay.we
                            string    key="192006250b4c09247ec02edce69f6a2d";
                int  i  =  0;
                string  sign  =  string.Empty;
            StringBuilder  sb  =  new  StringBuilder();
          foreach(KeyValuePair<string,  string>  temp  in  v   )  {
                                                              if(temp.Value==""||t




                                                    }
                                                    sb.Append("key="+key.Trim()+"");
                                                    sign=S_GetMD5(sb.ToString(),"utf-8
                            return    sign;
        }
  public static string S_GetMD5(string encypStr,string charset) {
                            string    retStr;
            MD5CryptoServiceProvider  m5  =  new  MD5CryptoServiceProvider();
                                                    //    md5
                            byte[]    inputBye;
                            byte[]    outputBye;
                                                    //    GB2312
                            try   {

                }  catch(Exception  ex)  {


                                                    }
                                                    outputBye=m5.ComputeHash(inputBye)
                                                    retStr=System.BitConverter.ToStrin
                                                    retStr=retStr.Replace("-","").ToUp
                            return    retStr;
        }
  public static string S_     () {
            System.Random  random  =  new  System.Random();
                            return    S_GetMD5(random.Next(1000).ToString(),"GBK").ToLower
        }
  public static string S_Get      () {
            TimeSpan  ts  =  DateTime.UtcNow-new  DateTime(1970,1,1,0,0,0,0);
```

```
                        return    Convert.ToInt64(ts.TotalSeconds).ToString();
        }//   1970
    SortedDictionary<string,string>  S_XML          (string  xmlstring)  {
            SortedDictionary<string,  string>  sParams  =  new  SortedDictionary<string,stri
                        try    {
                          XmlDocument   doc   =   new   XmlDocument();

                              XmlElement    root    =    doc.DocumentElement;
                              int    len    =    root.ChildNodes.Count;
                          for(int    i    =    0;i<len;i++)    {
                                                  string     name     =



            }   catch   {   }
                        return    sParams;
        }
    public  class  C_          {
                        public     string
                                                  #region        MyRegio
                        o_     ID="wx6b82f53daeeec518",




                       http://wxpay.weixin.qq.com/pub_v2/pay/notify.v2.php",


                  static    C_          ccc;
            public  static  string  S_            (int   v    )   {
                                                  if(ccc==null)        {


                              string     xml     =     string.Empty;
                      SortedDictionary<string,    string>   v      =    new   SortedDict













                          StringBuilder    ssb    =    new    StringBuilder();
                      foreach(KeyValuePair<string,    string>   k   in   v    )    {


                                                                             }




                              byte[]     byteA     =     Encoding.UTF8.GetBytes(xml);
                                                  return        Encodin
                                          }
            public  static  void  S_            (string  prepay_id)  {
```

```
                                                                                                            if(ccc==null)        {

                                                SortedDictionary<string,    string>    v        =    new    SortedDict






                                                      StringBuilder    ssb    =    new    StringBuilder();
                                                foreach(KeyValuePair<string,    string>    k    in    v    )    {


                                                                                                            string        ss=ssb.T

                                                               AndroidJavaClass     jc     =new     AndroidJavaClass("cc
                                                                              AndroidJavaObject     jo     =jc.GetStatic<


                                                                                           }
                        }
           public  class  C_            {
                                                   ///      <summary>
                                    ///           ID(                          ID)
                                                   ///      </summary>
                            public   string   appid   =   "";
                                                   ///      <summary>
                                                   ///            (                          )
                                                   ///      </summary>
                            public   string   mch_id   =   "";
                                                   ///      <summary>
                                                   ///
                                                   ///      </summary>
                            public   string   transaction_id   =   "";
                                                   ///      <summary>
                                                   ///
                                                   ///      </summary>
                            public   string   out_trade_no   =   "";
                                                   ///      <summary>
                              ///                        32
                                                   ///      </summary>
                            public   string   nonce_str   =   "";
                                                   ///      <summary>
                                                   ///                            appid   mch_id   transaction_id   out_
                                                   ///      </summary>
                            public   string   sign   =   "";
                                      static   C_          ccc;
                            public   static   string   S_          ()   {
                                                                                                            if(ccc==null)        {

                                                               string     return_string     =     string.Empty;
                                                      SortedDictionary<string,    string>    sParams    =    new
 SortedDictionary<string,string>();
```

```
                                        StringBuilder  sbPay  =  new  StringBuilder();
                                  foreach(KeyValuePair<string,  string>  k  in  sParams)  {



                                                                                                    }




                                                                                  return      return_s
                                                                  }
                }
        public  class  C_          {
                                    ///      <summary>
                           ///               SUCCESS/FAIL
trade_state
                                    ///      </summary>
                    public  string  return_code  =  "";
                                    ///      <summary>
                       ///
                                    ///      </summary>
                    public  string  return_msg  =  "";
                                    ///      <summary>
                           ///          ID(                    ID)
                                    ///      </summary>
                    public  string  appid  =  "";
                                    ///      <summary>
                                    ///           (                    )
                                    ///      </summary>
                    public  string  mch_id  =  "";
                                    ///      <summary>
                                    ///                        32
                                    ///      </summary>
                    public  string  nonce_str  =  "";
                                    ///      <summary>
                                    ///
                                    ///      </summary>
                    public  string  sign  =  "";
                                    ///      <summary>
                                    ///             , SUCCESS/FAIL
                                    ///      </summary>
                    public  string  result_code  =  "";
                                    ///      <summary>
                                    ///
                                    ///      </summary>
                    public  string  err_code  =  "";
                                    ///      <summary>
                                    ///
                                    ///      </summary>
                    public  string  err_code_des  =  "";
                                    ///      <summary>
                                    ///
                                                                  ///SUCCESS—
                                                                  ///REFUND—
                                                                  ///NOTPAY—
                                                                  ///CLOSED—
                                                                  ///REVOKED—
                                                                  ///USERPAYING--
                                    ///NOPAY--      (                    )     PAYERROR--
                                    ///     </summary>
                    public string trade_state = "";
                                    ///     <summary>
                                    ///
                                    ///     </summary>
                    public string device_info = "";
                                    ///     <summary>
                                    ///              appid
                                    ///     </summary>
                    public string openid = "";
                                    ///     <summary>
                                    ///                           Y-      N-
                                    ///     </summary>
                    public string is_subscribe = "";
                                    ///     <summary>
                                    ///           ,JSAPI  NATIVE  MICROPAY  APP
                                    ///     </summary>
                    public string trade_type = "";
                                    ///     <summary>
                                    ///
                                    ///     </summary>
                    public string bank_type = "";
                                    ///     <summary>
                                    ///
                                    ///     </summary>
                    public string total_fee = "";
                                    ///     <summary>
                                    ///                 <=                    -
                                    ///     </summary>
                    public string coupon_fee = "";
                                    ///     <summary>
                               ///                 ISO  4217                              CNY
                                    ///     </summary>
                    public string fee_type = "";
                                    ///     <summary>
                                    ///
                                    ///     </summary>
                    public string transaction_id = "";
                                    ///     <summary>
                                    ///
                                    ///     </summary>
                    public string out_trade_no = "";
                                    ///     <summary>
                                    ///
                                    ///     </summary>
                    public string attach = "";
                                    ///     <summary>
                                    ///                         yyyyMMddhhmmss    2009  12  27  9

                               ///          GMT+8   beijing
                                    ///     </summary>
                    public string time_end = "";
                }
}
```

```
//-----------    AndroidManifest.xml--------------------------
<manifest package="nn.cn.uj_002" xmlns:android="http://schemas.android.com/apk/res/android">
   <application>
      <activity android:name="nn.cn.uj_002.C_main">
               <intent-filter>
                   <action android:name="android.intent.action.MAIN"/>
               <category android:name="android.intent.category.LAUNCHER"/>
               </intent-filter>
        </activity>
    </application>
</manifest>
```

```
///----------      arr    ----         build.gradle---------------
apply plugin: 'com.android.library'
android {
    compileSdkVersion 24
    buildToolsVersion "24.0.1"
    defaultConfig {
        minSdkVersion 14
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
    repositories {
        flatDir {
            dirs 'libs'
        }
    }
}
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:24.+'
    testCompile 'junit:junit:4.12'
    compile files('libs/classes2.jar')
    compile(name: 'js_sdk_2.0.0', ext: 'aar')
```

```
   implementation files('libs/classes2.jar')
}
```

```
///--------------          --------------------------------------------------------

void S_Init(){
    o_Button =(Button)findViewById(R.id.button_05);
    o_Text=(TextView)findViewById(R.id.ss_004);
    o_Button.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View view) {
            if(o_Text.getText().length()>5){
                o_Text.setText("dd");
            }else {
                o_Text.setText("dddd____________ddddd");
                S_    (o_Button);
            }
        }
    });
}
public void S_    (Button but){
    but.setOnClickListener(new View.OnClickListener() {//
        @Override
        public void onClick(View v) {
            new AlertDialog.Builder(MainActivity.this).setTitle("       ")//
                    .setMessage("                                  ")//
                    .setPositiveButton("    ",new DialogInterface.OnClickListe
                        @Override
                        public void onClick(DialogInterface dialog, int which
                            finish();
                        }
                    }).setNegativeButton("    ",new DialogInterface.OnClickLi
                @Override
                public void onClick(DialogInterface dialog, int which) {//
                    Log.i("alertdialog","             ");
                }
            }).show();//
        }
    });
}
```

```
      ----RelativeLayout------
```

Write here...

```
//----PC------------------------
using  UnityEngine;
using   System.Collections;
using  UnityEngine.UI;
using  System;
public  class  C_Microphone:MonoBehaviour  {
           public  GameObject  o_   y;
           public  AudioSource  o_   y;
           private  AudioClip  clip;
           private  byte[]  recordData;
                              //                        44100              44
                      //
        public  const  int  SamplingRate  =  8000;
                      //

          void  Start()  {
                                        C_UGUI.Get(o_   y).o_   =delegate(Gam

                                                                     };
                                        C_UGUI.Get(o_   y).o_   =delegate(Gam

                                                                     };
                 }
          void  OnRecording(bool  v   )  {
                                              if(v   )    {//


                                  }   else   {//


                                                            int    lastPos

                                                              }    else    {






                                                                     //int
                                                //DbSound   dbs   =   new   N



                                                                    }
                 }
}
```

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;
using System.IO;
using UnityEditor;
using System;
public class C_MicroPhoneInput:MonoBehaviour {
                /*
                              :
     Starting Microphone failed. result=25 (Unsupported file or o_audio format. )
       UnityEngine.Microphone:Start(String, Boolean, Int32, Int32)
        C_MicroPhoneInput:S_StartRecord        () (at Assets/Script/C_MicroPhoneInput.
        C_MicroPhoneInput:OnGUI() (at Assets/Script/C_MicroPhoneInput.cs:41)
            unity3d        http://answers.unity3d.com/search.html?
f=&type=question&redirect=search%2Fsearch&sort=relevance&q=MicroPhone
                                                                  .net

                                                                *&#47;
        public  AudioSource  o_audio;
          int  deviceCount;
        void  Start()  {
                                                                o_audio=GetComponent<Auc
                      string[]  ms  =  Microphone.devices;
                                                                Debug.Log("          :
                                                                deviceCount=ms.Length;
                                    if(deviceCount==0)    {
                                                        Debug.Log("no    microphone
                                                                }
                }
        void  OnGUI()  {
                                    if(deviceCount>0)    {

",GUILayout.Height(Screen.height/20),GUILayout.Width(Screen.width/5)))            {



",GUILayout.Height(Screen.height/20),GUILayout.Width(Screen.width/5)))            {



",GUILayout.Height(Screen.height/20),GUILayout.Width(Screen.width/5)))            {



if(!Microphone.IsRecording(null)&&GUILayout.Butt
20),GUILayout.Width(Screen.width/5)))


                }
        void  S_StartRecord          ()  {
                                    int    z          ,
                                    Microphone.GetDeviceCa
                       Debug.Log("            =  "+z
.ToString());
```

```
                              while(!(Microphone.GetPosition(null)>0))   {   }
                         Debug.Log("devices'   Rand   =   "+Microphone.devices.Rank.ToString());
                   }
           void  S_StopRecord        ()  {
                                       if(!Microphone.IsRecording(null))    {

                                                                   }
                                                                   Microphone.End(null);
                                                                   o_audio.Stop();
                   }
           void  PrintRecord()  {
                                       if(Microphone.IsRecording(null))    {

                                                                   }
                                                                   S_SavWav.S_Save("GGD",o_audio.cl
                   }
           void  S_PlayRecord        ()  {
                                       if(Microphone.IsRecording(null))    {

                                                                   }
                                       if(o_audio.clip==null)    {

                                                                   }
                                                                   o_audio.Play();
                   }
          public  byte[]  S_GetClipData()  {
                                       if(o_audio.clip==null)    {
                                                Debug.Log("GetClipData   audio.clip   is   null")
                                                                                 return      null;
                                                                   }
                      float[]  samples  =  new  float[o_audio.clip.samples*o_audio.clip.channels];
                                                                   o_audio.clip.GetData(samples,0);
                              Debug.Log("sample.lenght   =   "+samples.Length);
                      byte[]  outData  =  new  byte[samples.Length*4];
                         int   top   =   0;
                      for(int  i  =  0;i<samples.Length;i++)  {
                                                var    by    =    System.BitConverter.GetBytes(sampl
                                                for(int    j    =    2;j<4;j++)

                                                for(int    j    =    0;j<2;j++)

                                                                   }/**/
                                       if(o_audio.clip.channels==1)    {
                      }  else  if(o_audio.clip.channels==2)  {  }
                                       if(outData==null||outData.Length<=0)    {
                                                Debug.Log("GetClipData   intData   is   null");
                                                                                 return      null;
                              }   else   {
                                                                   }
                                       return    outData;
                   }
  }
  public static class S_SavWav {
         const  int  HEADER_SIZE  =  44;
        public  static  bool  S_Save(string  z      ,AudioClip  clip)  {
                                       if(!z      .ToLower().EndsWith(".wav"))    {

                                                                   }
                         var   z           =   Path.Combine(Application.dataPath,z      );
                                                                   //
                                                                   Directory.CreateDirectory(Path.Ge
                      using(FileStream  nn  =  S_CreateEmpty(z        ))  {
                                                                   }
                         return   true;   //                               false
                   }
        public  static  AudioClip  S_TrimSilence(AudioClip  clip,float  min)  {
                      var   samples   =   new   float[clip.samples];
                                                                   clip.GetData(samples,0);
                              return    S_TrimSilence(new    List<float>(samples),min,clip.channels,c
                   }
        public  static  AudioClip  S_TrimSilence(List<float>  samples,float  min,int  channels,int  hz)
                                       return     S_TrimSilence(samples,min,channels,hz,false,false)
                   }
        public  static  AudioClip  S_TrimSilence(List<float>  samples,float  min,int  channels,int  hz,
   _3D,bool stream) {
                                       int     i;
                                       for(i=0;i<samples.Count;i++)     {
                                                                                 if(Mathf.Abs(sample


                                                                   }
                                                                   samples.RemoveRange(0,i);
                                       for(i=samples.Count-1;i>0;i--)     {
                                                                                 if(Mathf.Abs(sample


                                                                   }
                                                                   samples.RemoveRange(i,samples.Cou
                         var   clip   =   AudioClip.Create("TempClip",samples.Count,channels,hz,_3D,
                                                                   clip.SetData(samples.ToArray(),0)
                                       return     clip;
                   }
         static  FileStream  S_CreateEmpty(string  filepath)  {
                      var   fileStream   =   new   FileStream(filepath,FileMode.Create);
                      byte   emptyByte   =   new   byte();
                  for(int  i  =  0;i<HEADER_SIZE;i++)  {  //preparing  the  header

                                                                   }
                                       return     fileStream;
                   }
         static  void  S_ConvertAndWrite(FileStream  nn,AudioClip  clip)  {
                      var   samples   =   new   float[clip.samples];
                                                                   clip.GetData(samples,0);
                      Int16[]   intData   =   new   Int16[samples.Length];
                //converting  in  2  float[]  steps  to  Int16[],  //then  Int16[]  to  Byte[]
                      Byte[]   bytesData   =   new   Byte[samples.Length*2];
                   //bytesData  array  is  twice  the  size  of
                //dataSource  array  because  a  float  converted  in  Int16  is  2  bytes.
                 int  rescaleFactor  =  32767;  //to  convert  float  to  Int16
                      for(int   i   =   0;i<samples.Length;i++)   {

                                         Byte[]    byteArr    =    new    Byte[2];


                                                                   }
                                                                   nn.Write(bytesData,0,bytesData.Le
                   }
         static  void  S_WriteHeader(FileStream  nn,AudioClip  clip)  {
                         var   hz   =   clip.frequency;
                         var   channels   =   clip.channels;
                         var   samples   =   clip.samples;
                                                                   nn.Seek(0,SeekOrigin.Begin);
                      Byte[]   riff   =   System.Text.Encoding.UTF8.GetBytes("RIFF");
                                                                  nn.Write(riff,0,4);
                      Byte[]   chunkSize   =   BitConverter.GetBytes(nn.Length-8);
                                                                  nn.Write(chunkSize,0,4);
                      Byte[]   wave   =   System.Text.Encoding.UTF8.GetBytes("WAVE");
                                                                  nn.Write(wave,0,4);
                   Byte[]  fmt   =  System.Text.Encoding.UTF8.GetBytes("fmt  ");
                                                                  nn.Write(fmt,0,4);
                      Byte[]   subChunk1   =   BitConverter.GetBytes(16);
                                                                  nn.Write(subChunk1,0,4);
                      UInt16   two   =   2;
                      UInt16   one   =   1;
                      Byte[]   audioFormat   =   BitConverter.GetBytes(one);
                                                                  nn.Write(audioFormat,0,2);
                      Byte[]   numChannels   =   BitConverter.GetBytes(channels);
                                                                  nn.Write(numChannels,0,2);
                      Byte[]   sampleRate   =   BitConverter.GetBytes(hz);
                                                                  nn.Write(sampleRate,0,4);
                Byte[]  byteRate  =  BitConverter.GetBytes(hz*channels*2);  //  sampleRate  *
bytesPerSample*number of channels, here 44100*2*2
                                                                  nn.Write(byteRate,0,4);
                      UInt16   blockAlign   =   (ushort)(channels*2);
                                                                  nn.Write(BitConverter.GetBytes(bl
                      UInt16   bps   =   16;
                      Byte[]   bitsPerSample   =   BitConverter.GetBytes(bps);
                                                                  nn.Write(bitsPerSample,0,2);
                      Byte[]   datastring   =   System.Text.Encoding.UTF8.GetBytes("data");
                                                                  nn.Write(datastring,0,4);
                      Byte[]   subChunk2   =   BitConverter.GetBytes(samples*channels*2);
                                                                  nn.Write(subChunk2,0,4);
                                       //    nn.Close();
                }
}
```

```
  //----------------            ------------------------------
                [MenuItem("Assets/  AssetBundle-     -       ")]
       static  void  ExportResourceNoTrack2()  {
                      UnityEngine.Object[]   z           =   Selection.objects;
                   string   path   =   EditorUtility.SaveFilePanel("Save   Resource
                           string[]    z        =new    string[z        .Length];
                           List<Object[]>    o_           =new    List<Object[]>(
                           for(int    i=0;i<z        .Length;i++)    {

                                       o_           .Add(new    Object[]    {    z
                                                                     }
                           for(int    i=0;i<z        .Length;i++)    {

                                                                     }
                }
```

```
//-------------------              ------------------------
                    [MenuItem("Assets/  AssetBundle-
          static   void   ExportResourceNoTrack(){
                            string   aa_name   =   Select
                  string  path  =  EditorUtility.Save
                            if   (path.Length   !=   0){

BuildPipeline.BuildAssetBundle(Selection.activeObj

                    }
```

EditorWindow

```
using   System;
using   UnityEngine;
using   UnityEditor;
public  class  C_EditorWindow {    :
          public   Transform   o_        ;
           public   string   TagName="      ";
             GameObject[]   objs;
                 //
               [MenuItem("Window/C_             %Q")]
                     //
         static   void   Init()   {
                                                       C_                  window=(C_          )EditorWindow.GetWindow(typeof(C_
                                                                           window.Show();
                }
                      //  GUI
          void   OnGUI()   {
                                                                                      GUILayout.BeginHorizontal();
                                                     GUILayout.Label("      ",       EditorStyles.boldLabel);
                                   o_        =EditorGUILayout.ObjectField(o_        ,    typeof(Transform))     as     Transform;
                                                                                    GUILayout.EndHorizontal();
                                                                                      GUILayout.BeginHorizontal();
                                                     GUILayout.Label("        ",       EditorStyles.boldLabel);
                                                             TagName=EditorGUILayout.TextField(TagName,        EditorStyles.boldL
                                                                                    GUILayout.EndHorizontal();
                                              if(GUILayout.Button("    "))      {

                                                                 }
                     if(GUI.Button(new   Rect(20,   100,   200,   30),   "dddd"))   {

                                                                 }
                }
          void   S_    mm()   {
                                                                                                    objs=GameObject.FindGameObjec
                        foreach(GameObject   obj   in   objs)   {
                                                                                Instantiate(o_        ,       obj.transform.positi
                                                                 }
                }
}
```

Write   here...

.   All Rights Reserved   Copyright ©2018

```
public class FileIO {
                //            --------------
                //PC  ------Application.dataPath
                //      -----Application.persistentDataPath
          string   o_Path;
         static     FileIO   ccc;
       public  FileIO  S_Get()  {
                                    if(ccc==null)    {
                                                                              c

                                                                }
                                    return    ccc;
                }
        FileIO()  {  }
         public   void   S_SaveFile(byte[]  data,string   z    )  {
                   FileStream   stream   =   new   FileStream(o_Path+z    ,FileMo
              //  byte[]  zData  =  Encoding.UTF8.GetBytes("ffff"  );
                                                                stream.Positio
                                                                stream.Write(da
                                                                stream.Flush()
                                                                stream.Close()
                }
      public  byte[]  S_ReadFile(string  z    )  {
                   FileStream   stream   =   new   FileStream(o_Path+z    ,FileMo
                                    byte[]    zData=null;
                                    if(stream.Length<1)    {
                                                                              r
                          }   else   {
                                             zData    =    new    byte[1024];

                                                                }
                                    return    zData;
                }
}
```

```
using   UnityEngine;
using   System.Collections;
public  class  C_124_          :  MonoBehaviour  {
           public   Transform[]   o_      ;
          Vector3[]   o_     ,   o_     ;
         void   Start()   {     }
         private   void   S_     (float   z     ){
                           o_         =    new    Vector3[o_       .Length];
                           o_         =    new    Vector3[o_       .Length];
                 for   (int   i   =   0;   i   <   o_       .Length;   i++){
                                     int    z0    =    i    -    1;
                                     int    z2    =    i    +    1;
                              if   (z0   <   0)   {   z0   +=   o_       .Length;
                                          z2    =    z2    %    o_       .Length
                                          Vector3    z          =    o_       [z0
                                          Vector3    z          =    o_       [z2
                                Vector3   z              =   new   Vector3(z
                                                                      bool
                                             float    z         =    (Vector3.Ar

                                                       z         =    z         /

                                                       z        =(360-    z

                                       float    z         =    z         /    Mathf.
                                        Vector3    z             =    Vector3.Sl


                                                          if      (z

                                                                  z

                                                                   z

                                           o_     [i]     =     z              +
                                   o_     [i]    =    z              *    -1    +    o_
                                                                s}
                }
              void    OnDrawGizmos(){
                                                                    S_    (1);
                                            S_     (o_       ,      Color.yellow)
                                           S_     (o_      ,      Color.blue);
                                          S_     (o_      ,      Color.red);
                }
        public   static   void   S_     (Transform[]   z     ,   Color   z     ){
                              Gizmos.color     =     z     ;
                 for   (int   i   =   0;   i   <   z     .Length;   i++){



                                                                }
                }
        public   static   void   S_     (Vector3[]   z     ,   Color   z     ){
                              Gizmos.color     =     z     ;
                 for   (int   i   =   0;   i   <   z     .Length;   i++){



                                                                }
                }
}
```

Write here...

```
using  System;
using   System.Collections.Generic;
using  System.Linq;
using  UnityEngine;
public class C_124_        : MonoBehaviour {
          public  Transform[]  o_     ;
         void  Start()  {    }
        void  S_   (List<C_55  >  z     ,  float  z    )  {
               for  (int  i  =  0;  i  <  z     .Count;  i++){
                                int   z0   =   i   -   1;
                                int   z2   =   i   +   1;
                                                                     if(z0<0)      {



                                                                                     Vector3        z      =z
                                                                                     Vector3        z      =z
                         Vector3   z           =   new   Vector3(z      .z,   z      .y,   -1   *   z      .x);
                                                           bool      z          =Vector3.Angle(z          ,      z
                                    float    z        =    (Vector3.Angle(z      ,    z      ));
                                                                    float       z    ;
                                                                       if       (z          ==false){




                              float   z        =   z        /   Mathf.Sin(z    );
                               Vector3   z           =   Vector3.Slerp(z      ,   z      ,   0.5f);


                                                                                       if       (z
                                                                                                 z

                                                                                                   z




                                                                                              }
                      }
           Dictionary<Transform,    C_55   >   z          =new   Dictionar
          Dictionary<int,   C_55   >   z          2=new   Dictionary<int,
            C_mm           mm;
               void    OnDrawGizmos(){


                                                 List<Vector3>     zmmm=new     Lis
                                  for(int    i=0;   i<o_       .Length;   i++)   {
                                                                                              z
                                                                                            z

                                                                                              }

                                                         if(mm==null)      {


                                                                                              }
                                                                 mm.S_          (zmmm,
                                                                  List<C_55   >      z

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








                                                                                              }

                                                                                                     S
                       }
         public   void   S_         (List<C_55   >   z        )   {

                                                  List<List<C_55   >>      z
                                                       S_           (z       ,       z
                foreach(var  m  in  z        )  {
                              for(int   i=0;   i<m.Count;   i++)   {







                                                  }
           }
     void  S_   _     (List<C_55  >  z     )  {
                for(int  x=0;  x<z     .Count;  x++)  {

                              for(int   y=0;   y<z     .Count;   y++)   {























                                                  }
                foreach(var  n0  in  z     )  {
                                                                      if(n0.o_        .Count>0)      {
                                              n0.o_        =n0.o_        .OrderBy(n   =>   n.Value).ToDictionary(n   =>

                                                  }
           }
     void  S_        (List<C_55  >  z     ,  List<List<C_55  >>  z         ,string  ssw)  {
                              //S_mmmmds(ssw+"z     __",     z     );
                for(int  i=0;  i<z     .Count;  i++)  {
                                                    List<C_55  >     z          =new     List<C_55  >();
                               foreach(var   n   in   z     [i].o_        .Keys)   {




                             foreach(var   n   in   z          )   {



                                                  }
                              //S_mmmmds("z     __",     z     );
                  List<C_55  >  z     m88=z     .Where(n  =>  n.o_        .Count>0).OrderBy(n  =>  n.o_
 .Values.ElementAt(0)).ToList();
                             if(z     m88.Count==0)     {
                                                               string      ss="";
                              foreach(var   n   in   z     )   {





                                              S_        (z     ,    z         ,    ssw);

                                                  }
                            C_55      Aa=z     m88[0];
                                 C_55      Ba=Aa.o_        .Keys.ElementAt(0);
                                                          Aa.o_        .Remove(Ba);
                         List<C_55  >    z     A=new   List<C_55  >();
                         List<C_55  >    z     B=new   List<C_55  >();
                                                                                                    //Debug.L
                                                              ssw+=Aa.o_   +"-"+Ba.o_   +"==";
                for(int  i=0;  i<z     .Count;  i++)  {


                                                                                 }     else     {



                                                  }
                                                        z     B.Add(Aa.S_      ());
                                                        z     B.Add(Ba.S_      ());
                          z     B=z     B.OrderBy(n    =>    n.o_   ).ToList();
                                                         //S_mmmmds("",z     A);
                                                         //S_mmmmds("",z     B);
                             if(z     A.Count==3)     {
                            }   else   {
                                                                                    S_          (z      A,       z          ,ssw);
                                                               }
                                          if(z      B.Count==3)     {

                            }   else   {
                                                                                    S_          (z      B,       z          ,ssw);
                                                                                }
                */
         void  S_mmmmds(string  ss,List<C_55  >  z      )  {
                                                                                  ss+=z      .Count+"<==          ________";
                      foreach(var   n   in   z      )   {

                                           foreach(var    m    in    n.o_          )    {



                                                               }
                                                                              Debug.LogError(ss);
                }
        public  class  C_55    :  ICloneable  {
                               public    Transform    o_  ;
                              public    int    o_    ;
                                 public    Vector3    o_      ,o_      0,o_      ,o_        ;
                         public    float    o_    ,    o_    ;
                         public    Dictionary<C_55  ,    float>    o_          =new    Dictionary<C_55  ,    float>();
                     public   C_55  (Transform   z    ,   int   z    )   {


                                                               }
                         public    C_55     S_        ()    {
                                            C_55      mm=    this.Clone()    as    C_55  ;
                                                                    mm.o_          =new      Dictionary<C_55  ,      float>();
                                          foreach(var    n    in    o_          )    {


                                                                                  return       mm;
                                                               }
                          public    object    Clone()    {
                                                                                             return        this.MemberwiseClone();
                                                               }
                }
}
public  class  C_mm       :  MonoBehaviour  {
          public   MeshCollider   o_      ;
           public   MeshRenderer   o_       ;
            Mesh   o_Mesh;
        public  static  C_mm        S_Get()  {
                                  GameObject     gg=new     GameObject("C_      ");
                                               C_mm           nn=gg.AddComponent<C_mm      >();
                                                                                          nn.o_      =gg.AddComponent<MeshCollider>();
                                           nn.o_Mesh=new      Mesh();
                                                                                  nn.o_Mesh.name="ccc_Mesh";
                                                   MeshFilter       ff=gg.AddComponent<MeshFilter>();
                                                                                ff.sharedMesh=nn.o_Mesh;

                                                                                          nn.o_        =gg.AddComponent<MeshRenderer>();
                                 //nn.o_        .sharedMaterial=new    Material(Shader.Find("Legacy     Shaders/Transparent/Cutout/Soft
                                                         nn.o_        .sharedMaterial=new       Material(Shader.Find("Standard"));
                                                                                            nn.o_        .sharedMaterial.color=Color.white
                                       return     nn;
                }
        public  void  S_         (List<Vector3>  z    0,Vector3  z    ,bool  z            )  {
                          C_mm34    mm=S_           (z    0,    z    ,    z            );
                                                                                      o_Mesh.vertices=mm.o_    .ToArray();
                                                                                      o_Mesh.triangles=mm.o_    .ToArray();
                                                                               o_      .sharedMesh=o_Mesh;
                }
       public  void  S_        (List<Vector3>  z    0,  Vector3  z    ,  bool  z            )  {
                          C_mm34    mm=S_           (z    0,    z    ,    z            );
                                                                                      mm.o_    .AddRange(o_Mesh.vertices);
                                                                                      mm.o_    .AddRange(o_Mesh.triangles);
                                                                                      o_Mesh.vertices=mm.o_    .ToArray();
                                                                                      o_Mesh.triangles=mm.o_    .ToArray();
                }
       public  C_mm34  S_           (List<Vector3>  z    0,  Vector3  z    ,  bool  z            )  {
                                   //Vector3[]     z    =new     Vector3[z    0.Count*2];
                                  //int[]     z    =new     int[(z    0.Count-1)*6];
                       //for(int   i=0;   i<z    0.Count;   i++)   {
                                                     //                            z    [i*2]=z    0[i]-z    ;
                                                       //                            z    [i*2+1]=z    0[i]+z    ;
                                                                  //}
                        //for(int   i=0;   i<z    0.Count-1;   i++)   {
                                    //                    if(z            )     {
```

```
                                                                     //
                                                                       //
                                                                       //
                                                                       //
                                                                       //
                                                                       //
                              //                }   else   {
                                                                     //
                                                                       //
                                                                       //
                                                                       //
                                                                       //
                                                                       //
                                                  //                          }
                                                                    //}
                                    List<Vector3>    z    =new    List<Vector3>();
                                  List<int>    z    =new    List<int>();
                        for(int   i=0;   i<z    0.Count;   i++)   {


                                                                 }
                        for(int   i=0;   i<z    0.Count-1;   i++)   {
                                                                                   if(z             )       {






                                                            }     else     {







                                                                 }
                                 C_mm34    mm=new    C_mm34();
                                                                       mm.o_    =z    ;
                                                                       mm.o_    =z    ;
                                        return     mm;
                 }
         public  class  C_mm34  {
                                 public    List<Vector3>    o_    ;
                                public    List<int>    o_    ;
                 }
}
```

Write   here...

Write here...

```
using UnityEngine;
using UnityEditor;
using System.Collections;
using System;
public class MyEditor {
  [InitializeOnLoadMethod]
  static void Start() {
    Action OnEvent = delegate {
       Event e = Event.current;
       switch(e.type) {
          case EventType.mouseDown:
                Debug.Log("mousedown");
                  e.Use();
                    break;
          case EventType.mouseUp:
                Debug.Log("mouseup");
                  e.Use();
                    break;
          case EventType.MouseMove:
                Debug.Log("move");
                  e.Use();
                    break;
          case EventType.DragPerform:
                  Debug.Log("DragPerform");
                  e.Use();
                    break;
         case EventType.DragUpdated:
                Debug.Log("DragUpdated");
                  e.Use();
                    break;
          case EventType.DragExited:
                Debug.Log("DragExited");
                  e.Use();
                    break;
         }
      };
      EditorApplication.hierarchyWindowItemOnGUI+=delegate(int instanceID,Rect selectionRect) {
          OnEvent();
      };
      EditorApplication.projectWindowItemOnGUI+=delegate(string guid,Rect selectionRect) {
          OnEvent();
      };
  }
}
//MenuItem------          -----------
```

1.%X : ctrl + X;
2.#X : shift + X;
3.&X : alt + X;
4._X   X

5.%&X: ctrl + alt + X

```
//              ------------------------------------------------
using UnityEditor;
using UnityEngine;
public class C_SceneEditor:EditorWindow {
    RaycastHit o_Ray;
    GameObject o_       ;
    static C_SceneEditor ccc;
     [MenuItem("Example_      /          ----          ")]
    static void Init() {
        ccc=EditorWindow.GetWindow(typeof(C_SceneEditor)) as C_SceneEditor;
           SceneView.onSceneGUIDelegate=delegate(SceneView svv) {
              ccc.S_            (svv);
           };
        ccc.o_      =GameObject.CreatePrimitive(PrimitiveType.Sphere);
        ccc.o_      .transform.localScale=Vector3.one*0.2f;
        ccc.o_      .GetComponent<Collider>().enabled=false;
     }
    void OnEnable() {}
    void OnDisable() {}
    void OnDestroy() {}
    void OnGUI() {}
    void OnInspectorGUI() {
          Debug.Log("OnInspectorGUI");
     }
    void S_            (SceneView svv) {
        if(Event.current!=null) {
              switch(Event.current.type) {
                    case EventType.mouseDown:
                               Debug.LogError("            ------454-----");
                                       break;
                    case EventType.mouseUp:
                               Debug.LogError("            -------444----");
                                       break;
                  }
           }
        Camera cameara = svv.camera;
        Ray ray = HandleUtility.GUIPointToWorldRay(Event.current.mousePosition);
        if(Physics.Raycast(ray,out o_Ray,10000,-1)) {
            Vector3 z           = o_Ray.point;
                o_      .transform.position=z          ;
           }
        SceneView.RepaintAll();
     }
}
```

Write here...

```
    using   UnityEngine;
    using   System.Collections;
    public  class  C_       :  MonoBehaviour  {
                  WebCamTexture   z      ;
                 string   z          ="";
                  bool   isPlay=false;
              void   Start()   {
                                                                                         StartCoroutine(Test());

                     }
               IEnumerator   Test()   {
                                                     yield      return      Application.RequestUserAuthorization(UserAuthorization.WebCam);
                                                                     if(Application.HasUserAuthorization(UserAuthorization.WebCam))       {
                                                                                                                 WebCamDevice[]        z

                                                              z      =new     WebCamTexture(z         ,     400,     300,     15);


                                                                          }
                     }
          void   OnGUI()   {
                                            if(isPlay)     {
                                            GUI.DrawTexture(new   Rect(0,   0,   400,   300),   z      ,   ScaleMode.ScaleToFit);
                                                                      }
                 }
}
```

Write here...

GPS

```
using   UnityEngine;
using    System.Collections;
using   UnityEngine.UI;
public  class  C_106_GPS  :  MonoBehaviour  {
          public   Text   o_     ;
          void   Start()   {

                 }
           IEnumerator   StartGPS()   {
                                                                             o_
                                  if(!Input.location.isEnabledByUser)     {     /

                                                                   }
                                           o_    .text="            ";//

                                  int     o_        =20;                 //
                                                                whille(Input.lo

                                                         yield     return     no

                                                                   }
                               if(o_        <1)    {//


                                                                   }
                                           if(Input.location.status==LocationS

                              }    else    {






                                                                   }
                                                   Input.location.Stop();//
                 }
}
```

Write   here...

```
using   UnityEngine;
using   UnityEditor;
using    System.Collections.Generic;
public   class   ExportAssetBundles{
                                 [MenuItem("Assets/AssetBundle/AssetBundle-            -    ")]
          static   void   ExportResource()   {
                              string    aa_name    =    Selection.activeObject.name;
                      string   path   =   EditorUtility.SavePanel("Save   Resource",   "",   aa_name,   "ab");
                      if   (path.Length   !=   0)   {
                                                        Object[]     selection     =     Selection.GetFiltered(typeof(Object),
                                                                                 BuildPipeline.BuildAssetBundle(Selection.activeO
BuildAssetBundleOptions.UncompressedAssetBundle      |      BuildAssetBundleOptions.UncompressedAssetBundle);
                                                                  Selection.objects      =      selection;
                                                                }
                }
                                 [MenuItem("Assets/AssetBundle/  AssetBundle-    -    ")]
            static   void   ExportResourceNoTrack(){
                              string    aa_name    =    Selection.activeObject.name;
                      string   path   =   EditorUtility.SaveFilePanel("Save   Resource",   "",   aa_name,   "ab");
                          if    (path.Length   !=   0){
                                                                                    BuildPipeline.BuildAssetBundle(Selection.acti
BuildAssetBundleOptions.CollectDependencies,       BuildTarget.Android);
                                                                }
                }
                                 [MenuItem("Assets/AssetBundle/  AssetBundle-    -       ")]
           static   void   ExportResourceNoTrack2()   {
                              UnityEngine.Object[]    z           =    Selection.objects;
                          string    path    =    EditorUtility.SaveFilePanel("Save    Resource","","{0}",    "ab");
                                  string[]    z        =new    string[z        .Length];
                                    List<Object[]>    o_           =new    List<Object[]>();
                                 for(int    i=0;i<z        .Length;i++)    {

                                            o_           .Add(new    Object[]    {    z        [i]    });
                                                                }
                                 for(int    i=0;i<z        .Length;i++)    {
                                                                         BuildPipeline.BuildAssetBundle(z        [i],      o_
BuildAssetBundleOptions.CollectDependencies,       BuildTarget.Android);
                                                                }
                }
                                [MenuItem("Assets/AssetBundle/  BuildPlaye-       ")]
            static   void      MyBuild(){
                              string    o_name    =    Selection.activeObject.name;
                            string[]    path       =    {    AssetDatabase.GetAssetPath(Selection.activeInstanceID)    };
                          string    o_          =   EditorUtility.SaveFilePanel("    u3d    __          ______
"u3d");
                                            if    (path.Length    !=    0){
                                                                                                                                    Bu
BuildOptions.BuildAdditionalStreamedScenes);
                                                                                  }
                }
}
```

Write here...

```
public  class  C_Debug  :  MonoBehaviour  {
              string   zDebugPath;
             StreamWriter   sw;
            FileStream   fs;
         void  Awake()  {
                            zDebugPath    =   Application.persistentDataPath   +   "/Debug.txt";
                                              if     (File.Exists(zDebugPath))
                                                                                              File.Delete(zDebugPath);       //

                                   if     (!File.Exists(zDebugPath))     {
                                               fs     =     new     FileStream(zDeb

                                                                  }
                                   sw     =     File.AppendText(zDebugPath);
                     Application.logMessageReceived   +=   delegate   (string   c
                                              if     (type     ==     LogType.Excep
                                                    S_         (type    +    "@@--
                                                            }     else     {
                                                                           S_


                }
          void   Start()   {
                                                                           S_UI_

                }
         List<string>   z       =   new   List<string>();
           public   string   o_Show="";
         void   S_         (string   ss)   {
                                                                              sw
                                                                          z     .A
                   o_Show   =   ss   +   "\n"   +   o_Show;
                      if   (z     .Count   >   400)   {

                                                StringBuilder     ssb     =     new
                                                  int     z         =     z     .Co
                            for   (int   i   =   0;   i   <   z     ;   i++)   {
                                                          ssb.Append(z     [z

                                                                 o_Show      =
                                                                  }
                }
          void   OnDestroy()   {
                                                                           sw.Cl
                }
             void    S_UI_Init()    {
                           o_width    =Screen.width    -    100;
                                  o_height=Screen.height     -     200;
                          o_Box=new   Rect(20,   100,   o_width,   o_height);
                     o_Bottom=new   Rect(0,   0,   o_width   -   60,   9600);
                }
          Rect   o_Box,   o_Bottom;
          int   o_width,   o_height;
         public   Vector2   scrollPosition   =   Vector2.zero;
        public   float   scrollVelocity   =   0f;
         public   float   timeTouchPhaseEnded   =   0f;
         public   float   inertiaDuration   =   0.5f;

            public   Vector2   lastDeltaPos;
             bool   o_Switch;
          void   OnGUI()   {
                                   //GUI.skin.box.fontSize     =     25;
                    //if   (GUI.Button(new   Rect(5,   5,   250,   60),   "Debug
                                 //                 o_Switch     =     !o_Switch;
                                                                      //}
                               //if    (o_Switch)    {
                          //             scrollPosition   =   GUI.BeginScrollVi
                  //          //for   (int   i   =   0;   i   <   32;   i++)   {
                   //          //           GUI.Button(new   Rect(0,   i   *   50,
                                                   //                       //
                                          //                       GUI.Box(o_Bott
                                                         //
                                                                      //}
                }
          void   Update()   {
                               if    (o_Switch)    {
                                              if    (Input.touchCount    >    0)



                                                               }    else    if
```

```
                                                                                                                                scrollVelocity




                                                            }     else     {
                                                                         if    (scrollVelocity    !=    0.0f)    {

                                                                                   float    t    =    (Time.time    -    timeTouchPhaseEnded)
                                                                                                      float     frameVelocity     =     Mathf.Lerp
                                                                                                                     scrollPosition.y      +=

                                                                                                                                if      (t      >



                                                                 }
                sep}
}
```

wwwwww

C#

Write here...

```
//     GUID----------------------------------------
System.Guid.NewGuid().ToString()
//c#          -----------------------------------------------
Random ran = new Random();
int dd = ran.Next(100, 999);

//              -----------------------------
int z      =0;
Random ran = new Random();
do {
    z      =ran.Next(9000);
    Console.WriteLine(z      );
} while(z      <8900);

//      ---------------------------------------------
Stopwatch  ww  =  new  Stopwatch();
ww.Start();
ww.Stop();
Console.WriteLine(ww.ElapsedMilliseconds);

///VS2013                ------------------------
ctrl+H
^(?([^\r\n])\s)*\r?$\r?\n

^(?([^\r\n])\s)*\r?$\r?\n

^(?([^\r\n])\s)*\r?$\r?\n

[\t]*//[^\n]*\n
[\t]*//[^\n]*\n

//            ----------------------
 Process[] ps = Process.GetProcesses();

//                     -------------------------
using  System.IO;
Path//-------------
Directory//----------
File  //--------------
DirectoryInFo
FileInFo
//       ------------------------------------------
using(FileStream  o_        =  new  FileStream("d:\\
       byte[]  o_    s  =  new  byte[1024];
       int  o_          =   o_        .Read(o_    s,
       string  o_        =  Encoding.Default.GetStr
        byte[]  o_    2s  =  Encoding.UTF8.GetBytes
}
void  S_             (){    //-----------------------
      FileStream  o_       _    =  new  FileStream(@'
      FileStream  o_       _    =  new  FileStream(@'
      byte[]  i_    s  =  new  byte[1024  *  1024];
       int  i_            =  0;
                 do{
                     i_           =   o_      _  .Rea
                             o_      _  .Write(i_
       }while  (i_             ==  i_    s.Length);
```

```
                o_      _  .Dispose();
                o_      _  .Dispose();
                Console.Write("        ");
}
//        ------------------------
using(FileStream o_       = new FileStream("D:\\1.txt", FileMode.Open)){
      byte[]  o_    s  =  new  byte[1024];
      int  o_          =   o_       .Read(o_    s,  0,  o_    s.Length);
     for  (int  i  =  0;  i  <  o_        ;i++  ){
                   o_    s[i]   =   (byte)(255   -   o_    s[i]);
                }
     using  (FileStream  ff  =  new  FileStream("d:\\2.txt",  FileMode.Create)){
                          ff.Write(o_    s,    0,    o_        );
                }
}
//                    (        )------------------------------------------------------
byte[] S_        (){
       byte[]  bb  =  System.IO.File.ReadAllBytes("o_    ");
      byte[]  bb2  =  new  byte[bb.Length  +  1];
          bb.CopyTo(bb2,   1);
        bb2[0]  =  1;
          return   bb2;
                }
void S_        (byte[] bb){
          switch   (bb[0]){
                                    case    0:


                                    case    1:


                }
//      -------------------------------------------------------------------------------------------
using System.Threading;
Thread dd = new Thread(S_    );
dd.IsBackground = true;//                  ;
dd.Name = "dd      ";
dd.Start();//
dd.Abort();//
void S_    (){
    lock  (this)  {  //            ;

                                                  Thread.Sleep(500);//            500
                 Thread   hh   =   Thread.CurrentThread;//
         }
}
TextBox.CheckForIllegalCrossThreadCalls = false;//
//            ---------------------------------------------
Thread hh2 = new Thread(new ParameterizedThreadStart(delegate { S_    ("    001", "    002"); }));
hh2.Start();

//     ---------------------------------------------------------------------------------------------
-
delegate void ss();
ss k =new ss(       );
k+=    2;
k();

//      ------------------------------------------------------------
MessageBox.Show("ddddd");
```

```
//                --------------------------------------------------------------------------------
System.Diagnostics.Process.Start(@"C:\Users\gff.exe ");

//                    ----------------------------------------------------------------------------
using System.Drawing;
string path = @"C:\Users\Administrator\Desktop\";
Image img = new Bitmap(path+"ss.jpg");
Image img2 = new Bitmap(path + "ss2.jpg");
Graphics g = Graphics.FromImage(img);
g.DrawImage(img2, new PointF(10, 10));
g.DrawString("      ", new Font("        ", 30), Brushes.Red, new PointF(10, 10));
img.Save(path + "ss2.jpg");

//      --        -------------------------------------------------------
Hashtable hh = new Hashtable();
hh.Add("u", "ddd001");//
hh["ss"] ="eeeeeeee";   //
hh["u"]   ="eee";              //
string ss = hh["u"]; //
hh.Remove("u"); //
hh.ContainsKey("abc") //      (key)                    ;
hh.Count         //
foreach (var c in hh.Values/Keys){  //
    print(c.ToString());
}

//                      -----------------------
string ss = "";
if(z    .TryGetValue("ccc",out ss)==true) {
                Console.WriteLine(ss);
} else {
          Console.WriteLine("ccc        ---------");
}

z    .Values.ElementAt<string>(5)   //--------------                        -----------Ling--------
------------------

//Dictionary<string, C_    > ----------------    -----------------------------
//SortedDictionary<string, C_    > ---          -------
//ConcurrentDictionary   <string,  C_    >  //         -------------------
Lookup<string, C_     > ---        -------
foreach (KeyValuePair<string, C_    > kk in         ){
     Console.WriteLine(kk.Key  +  "        "  +  kk.Value);
 }

void  S_           <T>(Dictionary<int,T>  vdd)  {    //------            -------------------
          if(vdd.Count>0)   {
                           List<KeyValuePair<int,T>>    lst=new    List<KeyValuePair<int,T>>(vdd);
                      lst.Sort(delegate(KeyValuePair<int,T>   s2,KeyValuePair<int,T>   s1)   {
                                                                              return      s2.Key.Co
                                                                 });
                                                                 vdd.Clear();
                }
}
Dictionary<string,GameObject> v        =new Dictionary<string,GameObject>(); //         List<T>---
List<GameObject> v      =new List<GameObject>(v        .Values);
```

```
//        -----------------------------------------------------------------------
int[] ii = { 2, 6, 65, 12, 4 };
ArrayList ll = new ArrayList();
ll.Add(1);//
ll[0] = 100; //      --
ll.RemoveAt(0); //
ll.Add("ddd");//
ll.AddRange(ii);//        --
bool bb = ll.Contains("ddd"); //    "ddd"                     ;
int i_     = ll.Capacity;
for (int i = 0; i < ll.Count; i++){
        print("ddd"  +  ll[i]);
}

//Ling    -------------------using System.Linq;------------------------
void S_List    (List<GameObject> v    ,GameObject v      ) { //              OrderBy
OrderByDescending        )------------------
         List<GameObject>  x    2=v    .OrderBy(c   =>  Vector3.Distance(c.transform.position,v
.transform.position)).ToList();
}
XXX.RemoveAll(x => x==null); //             --------------

//            ----------------------------------------
List<int> ss1= ss.Distinct().ToList();



//            ------------------------------------
Queue<string> z        =new Queue<string>();
z        .Enqueue("dddddddddd");//      --
z        .Enqueue("dddddd333dddd");
z        .Enqueue("dddd");
string ss = z        .Dequeue();//                 --
string ss2 = z        .Dequeue();
Console.WriteLine(z        .Count+    "_____    "+ss+"    _____"+ss2);



//      --------------------------------------------------
int im1 = System.Environment.TickCount;
//
int im2 = System.Environment.TickCount;
int im3 = im2 - im1;//




//----------------------md5    ----------------------------
                   public   static   string   S_md5(string   ss){
                                     using    (MD5    m5    =    MD5.Create()){
                                                                            byte[]     bb     =
                                                                                                bb

                                                                StringBuilder     bb5     =     new
                                          for    (int    i    =    0;    i    <    bb.Length;    i++){
                                                                 }



public static string S_            (string  str)  {  //---------------------
        if(Regex.IsMatch(str,  @"[\u4e00-\u9fa5]"))  {
                           StringBuilder    ssb=new    StringBuilder();
                                    var     charS=str.ToCharArray();
                   for(int   i=0;  i<charS.Length;   i++)   {
                                                                              if(charS[i]>=0x4e00&&



                                                        }     else     {


                                                                 }
                                                                 str=ssb.ToString();
        }  else  {
                                                                 Console.WriteLine("          __"+st
                }
          return   str;
}


sealed



//-------------win    TreeView--------------------------------------------
   void  S_    (string  ss,int  ii) {
            if(ii==0)    {
    S_    2(ss,Color.FromArgb(255,255,255));
      }  else  if(ii==1)  {
    S_    2(ss,Color.FromArgb(0,0,255));
      }  else  if(ii==2)  {
    S_    2(ss,Color.FromArgb(0,0,0));
      }  else  if(ii==3)  {
    S_    2(ss,Color.FromArgb(255,0,0));
     }
   }
   void  S_    2(string  ss,Color  cc){
            if    (o_TreeView.Nodes.Count>50){
    o_TreeView.Nodes.RemoveAt(0);
     }
       TreeNode   nn=new   TreeNode();
            int    ii=this.o_TreeView.Nodes.Add(nn);
            TreeNode    n=this.o_TreeView.Nodes[ii];
   n.Text=ss;
   n.ForeColor=cc;
   }



//-----------------C#                  ---------------------------------------
Console.BackgroundColor=ConsoleColor.Blue; //
Console.ForegroundColor = ConsoleColor.White; //
Console.WriteLine("          .");
Console.ResetColor(); //
Console.BackgroundColor=ConsoleColor.Green;
Console.ForegroundColor = ConsoleColor.DarkGreen;
string value = "             ";
Console.WriteLine(value.PadRight(Console.WindowWidth-value.Length)); //


   public static void S_    (string ss,int ii) {
             if(ii==0)    {
     Console.ForegroundColor=ConsoleColor.Cyan;//
        }else  if(ii==1)  {
     Console.ForegroundColor=ConsoleColor.White;//
       }  else  if(ii==2)  {
     Console.ForegroundColor=ConsoleColor.Yellow;//
       }  else  if(ii==3)  {
     Console.ForegroundColor=ConsoleColor.Red;//
      }
             Console.WriteLine(DateTime.Now.ToString("hh:mm:ss")+"-->    "+ss);
    }

//------------lock      --------------------------------------------------------------------
 public static C_file Ccc{
                get{
                                    lock     (typeof(C_file)){
                                             if    (ccc    ==    null){
                                                                GameObject    gg    =    new    Gam
                                                                                                 ccc

                                                                              return       ccc;
                                                                }
                }
}
try {
   Monitor.Enter(obj);
    //        ();
} catch(Exception ex) {
} finally {
    Monitor.Exit(obj);
}

//Monitor               --------------------------------------

Enter(Object)

Exit(Object)

IsEntered

Pulse

PulseAll

TryEnter(Object)

TryEnter(Object, Boolean)

Wait(Object)


//  C#                 ---------------------------------------

       —     —               .  (      )

unsaf e{
}

 //        ----------------------------
                          public static void S_      (){
                                     DateTime v//          = DateTime.Now;
                                      DateTime v  =new  Date//Time(2016,5,31,23,30,12);
                                       string ss=v  .ToString("yyyy,MM,dd,HH,mm,ss");          //
                                DateTime v 2 = v  .AddDays(60);      //
              if(DateTime.Compar//e(v    ,v    2)>=0)  {

                                                }   else   {




                                                     }
                     public static string S_(DateTime v      , DateTime v     ){
                                    TimeSpan v     = new TimeSpan(v      .Ticks);
                                    TimeSpan v     = new TimeSpan(v      .Ticks);
                                             TimeSpan ts = v  .Subtract(v  );//Subtract(  )//

                                                                                                 re
.ToString()+"  ";
                                                         }
Days        //
Hours        //
Minutes        //
Seconds        //
Milliseconds        //

string     ss=         now.ToString("yyyy,MM,dd,HH,mm,ss,fff");
   //        -----------------------------------

//              ----//          ---------------------------------------
long              =   Convert.ToInt32((DateTime.Now-new   DateTime(1970,1,1,8,0,0)).TotalSeconds);



//-      ----        -----------------------------------------------------
-------------------
```

```
using System.Linq;
   public static List<string> S_          (this IEnumerable<string> z     ,string z        ) {
      IEnumerable<string> z       = from ss in z
                                              where  ss.Contains(z         )
                         select ss;
      return z      .ToList();
   }

//                  IP--------------------------------------
string web = "www.aaa9000.com";
IPHostEntry host = Dns.GetHostByName(web);
IPAddress ip = host.AddressList[0];
web = ip.ToString();

//           ------------------operator-----------
   public class C_mm {
      public string o_ss;
      public C_mm(string ss) {
         o_ss=ss;
         Console.WriteLine(o_ss);
      }
      public static C_mm operator+(C_mm a,C_mm b) {
         return new C_mm(a.o_ss+"______"+b.o_ss);
      }
   }
```

---AssetDatabase

```
using  System;
using   System.Collections.Generic;
using   System.ComponentModel;
using  System.Data;
using  System.Drawing;
using  System.Linq;
using  System.Text;
using  System.IO;
using  System.Net;
using  System.Net.Sockets;
namespace      _001{
          public  partial  class  C_001{
                            private   static   Socket   o_socket;
                           public   static   TcpClient   o_         ;
                           public   static   byte[]   o_   ;
                   public  static  byte[]  o_data  =  new  byte[1024  *  1024];
                      public  static  int  i_          =  1024*1024,i_    ;
                           public   static   FileStream   o_     _  ;
                      public   static   void   Main(String[]   args)   {
                                                  o_     _     =    new    FileStream(@"d:\1.mp4",    FileMode.Open);


                                                                 }
                           private   static   void   S_         (){
                                               o_            =    new    TcpClient("127.0.0.1",    8884);
                                                        o_        =    new    byte[o_         .ReceiveBufferSize];

                                                                    o_         .GetStream().BeginRead(o_    ,     0,     System.C
null);
                                                                  }
                                          void     S_dd(){
                                                                                         #region       MyRegion
                                                                  //o_socket     =     o_         .Client;
                                                  //int    i_          =    0;
                                              //FileStream    i_         =    new    FileStream(@"d:\1.mp4",    FileMode.Open);
                                              //byte[]    i_    s    =    new    byte[1024];
                                                 //BinaryReader    br    =    new    BinaryReader(i_      );

                                                  //int    log    =    0;
                                                    //while    (log    ==    0){
                                           //           i_          =   br.Read(i_    s,    0,    i_    s.Length    );
                                    //          if   (i_          <   i_    s.Length){   log   =   1;   }
                                                                    //                       o_socket.Send(i_    s,     i_




                                              FileStream    o_      _      =    new    FileStream(@"d:\1.mp4",    FileMode.Open);
                                              FileStream    o_      _      =    new    FileStream(@"d:\2.mp4",    FileMode.Create)
                                     byte[]   i_    s   =   new   byte[1024   *   1024];
                                                  int    i_          =    0;

                                                                            i_           =    o_      _  .Read(i_    s,    0,
                                                                                                             o_      _  .Write(i_
                                              }    while    (i_          ==    i_    s.Length);


                                                                  }
                           public   static   void   s_        (){


                                                                                       if     (i_            ==     o_data.Length
                                                                                                                 i_            =




                                                                  }catch     (Exception     ex){}
                                                                  }
                          public   static   void   s_       (IAsyncResult   ar){

                                                                                            int     o_              =     o_
                                                                                                                  Console.Write("


                                                                                                            o_          .GetStream(
        ,  null);
```

```
                                                                 }catch      (Exception      ex){

                                                                 }
                ]
}

using  System;
using    System.Collections.Generic;
using   System.Linq;
using   System.Text;
using    System.Threading.Tasks;
using   System.Net;
using    System.Net.Sockets;
using    System.Collections;
using   System.IO;
namespace       _001{
             class   Program{
                            static    void    Main(string[]    args){
                                            TcpListener    o_          =    new    TcpListener(IPAddress.Any,    8884);

                                                                     Console.WriteLine("           :      {0}----------------      ...\n----
                                                                                     while       (true){
                                                                              ss_UDP_       o_         =     new     ss_UDP_  (o_      .Accep

                                                                 }
                }
              class    ss_UDP_  {              //              ____            ____
                     public   static   Hashtable  o_          =   new   Hashtable();
                                private    TcpClient    o_          ;
                               public    string    o_      IP    ;
                               private    byte[]    o_        ;
                                          FileStream     o_      _  ;
                                       int     i_    ;
                                   public     ss_UDP_  (TcpClient    client){
                                                  o_      _      =    new    FileStream(@"c:\2.mp4",    FileMode.Create);
                                                             o_              =     client;
                                                                         o_      IP          =      client.Client.RemoteEndPoint.ToString();
                                                                                         o_        .Add(o_      IP    ,       this);
                                                       o_            =     new     byte[o_          .ReceiveBufferSize];
                                                                   client.GetStream().BeginRead(o_        ,      0,      System.Convert.ToInt
   null);
                                                                 }
                          public    void    s_           IAsyncResult    ar){
                                                                                   int       o_            ;



                                                                                 }if     (o_                 <     1){







                                                                                                                                 s_

                                                                                                             }       lock       (o_           .

.ReceiveBufferSize),  s_              null);

                                                                 }catch      (Exception      ex){

                                                                                                                     s_                 ("o_

                                                                 }
                            public    void    s_               (string    message){

                                              foreach    (DictionaryEntry    c    in    o_        ){
                                                                                                                      ((ss_UDP_  )(c.Value)).

                                                                 }
                          public    void    s_           string    message){







                                                                                  byte[]     by     =     System.Text.Encoding.UTF8.GetBytes
                                                                                                               ns.Write(by,       0,       by.

                                                                 }catch      (Exception      ex){

                                                                 }
                }
}
```

```
System.Timers.Timer t = new System.Timers.Timer(10000);    //     Timer                    10000
t.Elapsed += new System.Timers.ElapsedEventHandler(theout); //
t.AutoReset = true;     //                false            (true)
t.Enabled = true;       //        System.Timers.Timer.Elapsed
public void theout(object source, System.Timers.ElapsedEventArgs e)  {
      MessageBox.Show("OK!");
}
```

```
//                 -------------    C_           .S_       (1000, S_00
public class C_           {
    public delegate void dell();
    static System.Timers.Timer t;
    static dell del;
    public static void S_      (int        , dell de){
        t = new System.Timers.Timer(      *1000);
        t.Elapsed += S_002;
      del = de;
      t.AutoReset = false;
       t.Enabled = true;
    }
    static void S_002(object source, ElapsedEventArgs e){
       del.Invoke();
    }
}
```

```
     public class C_         2 {
         public static void S_       (float       ,ThreadStar
            Thread th =new Thread(delegate(){ S_      2
                th.IsBackground=true;
                th.Start();
           }
           static void S_      2 (float       ,ThreadStart de
               Thread.Sleep ((int)(     *1000));
                   del.Invoke();
              S_      2(      ,del);
           }
     }
```

```
Thread.Sleep(1000);
```

```
using  System;
using   System.Reflection;



                        C_fff   c1   =   new   C_fff();

                                    Assembly    aa    =    GetType().Assembly;    //
                                          Type     type     =    aa.GetType("C_fff");//
                                   Type    type2    =    typeof(C_fff);    //
                                   Type[]    types    =    aa.GetTypes();    //
                                          Type[]     types2     =    aa.GetExportedTypes();
        FieldInfo   ff   =  type.GetField("o_name");//      C_fff    o_name
                                                 MethodInfo     mm     =     type.GetMethod

                                   C_fff    cc    =    Activator.CreateInstance(type)
                                                       ff.SetValue(cc,     "     ");     //
                                         string    ss    =    mm.Invoke(cc,    null).ToSt

                                                                    type.Assembly;       //     t
                                                                    type.FullName;       //     t
                                                                        type.Name;            //
                                                                    type.IsArray;       //     ty
                                                                   type.IsEnum;      //     type
                                         //          ----
                                                        type.IsAssignableFrom(Type     i);
                                                          type.IsSubclassOf(type     father
                                                         type.IsInstanceOfType(object     o
                                                                                                   //-
                                                                             type.GetField("gender
                                                                             type.GetMethod("SayHi


                                                         Assembly[]     aaa     =          Ap
                                                 Assembly     aa2     =     this.GetType().
                                                     Assembly     aa3     =     Assembly.Lo

                         }
                void   S_    (){
                                string   path   =   @"C:\d";
                                  string[]    paths    =   System.IO.Directory.GetFiles(path,    "
                             foreach   (string   ss   in   paths){
                                                                 Assembly     aa     =     Assembly.Lo
                                                                 Type[]     types     =     aa.GetExpo
                                                                Type     plug01     =     typeof(CC_
                                                     foreach    (Type    t    in    types){




                                                                            }
                     }
```

```
using System.Reflection;




        void S_004(){
                    Assembly ass =
Assembly.LoadFrom(@"C:\Users\xia01\Desktop\mm\mm\bin\Debug\mm.dll");
                string[] sss = ass.FullName.Split(","[0]);
                Type type = ass.GetType(sss[0]);
                                                  gameObject.AddComponent(type);
            }




        void S_003(){
              var fs = new FileStream(@"C:\Users\xia01\Desktop\mm\mm\bin\Debug\mm.dll",
FileMode.Open);
              var b = new byte[fs.Length];
                    fs.Read(b, 0, b.Length);
                                                  fs.Close();
                var assembly = System.Reflection.Assembly.Load(b);
                var type = assembly.GetType("Test");
                                                        gameObject.AddComponent(ty
            }




void S_002(){
                Assembly ass = Assembly.LoadFrom(@"C:\Users\xia01\Desktop\CC_      \CC_
\bin\Debug\CC_      .dll");
                Type tt = ass.GetType("C_001");
                MethodInfo S_s = tt.GetMethod("S_001");
                Type i_       = typeof(I_xia001);
                if (tt != null){
                                                              if     (i_     .IsAs
                                                            I_xia001           =
                            object[] objs = new object[] { ss /* variable */

                                                                             ss


                                                  }
            }









using UnityEngine;
using System.Collections;
using System;
public class C_006 : MonoBehaviour{
        IEnumerator Start(){
                string url="file:///" +"Application.dataPath + "/Plugins/Test.unity3d";//
          , Test.unity3d
```

```
                        WWW www = new WWW(url); yield return www;
                              www.error != null)if{ (
                                              www.error);
                                                                                                    }else{
                                                                          AssetBundle www.assetBundle;bundle =
                                                        TextAsset txt = bundle.Load("Test") as TextAsset;

                                                                                           System.Reflection.Assembly assembly
System.Reflection.Assembly.Load(txt.bytes);
                                                                          Type type = assembly.GetType("Test");
                                                               GameObject obj = new GameObject();

                                                                                                    }
                          }
}
```

```
using System;

using System.Collections.Generic;
using System.Linq;

using System.Text;
using UnityEngine;
using System.Collections;
public class Test : MonoBehaviour{
            void Start()            {
                          Debug.Log("sucess");             }
}
```

```
//==             --
=============================================================================

static void Main(){
         MyClass cc = new MyClass();
           cc.Property1 = 2;
         Type type = cc.GetType(); //
         PropertyInfo ff = type.GetProperty("Property1"); //
        int ii = (int)ff.GetValue(cc, null); //
          ff.SetValue(cc, 5, null); //
}

public class MyClass{
        public int Property1 { get; set; }
}
```

```
using  System;
using    System.Collections.Generic;
using   System.Linq;
using   System.Reflection;
using   System.Text;
using    System.Threading.Tasks;
namespace  CC_003{
             class   Program{
          static   void  Main(string[]   args){
                          string   ss    =   "sssss";
                          MethodInfo    i_           =   typeof(Program).GetMethod("S_001",    BindingFlags.Static    |    BindingFla
                         if   (i_           !=   null){
                              object[]  objs   =   new   object[]  {   ss   /*   variable   */   };
                                                                                            i_        .Invoke(null,       objs);
                                                                ss      =      (string)objs[0];


                                                                                   }
                 }
      static  void  S_001(ref  string  arg){
                                       arg     =     "test_______________________jjjjjjjjjjjjj";
                                                                 }
                 }
}
```

```
using  System;
using   System.Reflection;
using   System.Globalization;
using   Microsoft.CSharp;
using   System.CodeDom;
using   System.CodeDom.Compiler;
using   System.Diagnostics;
using   System.IO;
using   System.Text;
namespace  CC_   {
             class   Program{
                            static   void   Main(string[]   args){



                                                                 }
                            static   void   S_   001(string   path){
                                                  CSharpCodeProvider    cc    =    new    CSharpCodeProvider();
                                            CompilerParameters   cp   =   new   CompilerParameters();//

                                                                     cp.OutputAssembly     =     "xxx.exe";//exe
                                                        cp.GenerateExecutable    =    true;    //   exe
                                                       cp.GenerateInMemory    =    false;    //<--
                                                        CompilerResults    cr    =    cc.CompileAssemblyFromSource(cp,    Read(p
                                                                                             if      (cr.Errors.HasErrors){

                                                                             foreach    (CompilerError    err    in    cr.Errors)




                                                                  Assembly   ass   =   cr.CompiledAssembly;   //
                                                                                            object     oo     =     ass.CreateIns
                                                                                               MethodInfo     mi     =     oo.Get


                                                                 }
                            static   string   Read(string   path){
                                                  string   ss   =   "";
                                             StreamReader   sr   =   new   StreamReader(path,   Encoding.Default);
                                                                                      string      line;
                                          while   ((line   =   sr.ReadLine())   !=   null){
                                                                      ss    +=    line    +    "\n";

                                                                                     return      ss;
                                                                 }
                }
}
```

wwwwww

Cddd.cs

```
using  System;
public  class  Cddd    {
    static  void  Main(string[]  args){
                                                                  Console.WriteLine("dfffffffffffffff----------======
                                              Console.ReadLine();
     }
  public  void  S_002(){
                                                                                Console.WriteLine("dddd-------------
      }
}
```

wwwwww

txt

```
                    static   void   S_    (string   i_    ,string   i_path,string   name){
                                                          if     (!Directory.Exists(i_path
                                                                                StreamWrit
                                                                                FileStream
                               string   path   =   i_path   +   name   +   ".txt";
                                    if   (!File.Exists(path)){fs   =   new   FileStream(pa
  FileAccess.Write);   fs.Close();   }
                                                              sw     =     File.AppendText


                                                                  }
                        static   string   S_    (string   path){
                                    StreamReader   sr   =   new   StreamReader(path,   Enco
                                                                                String
                                               string    s    =    "";
                                    while   ((line   =   sr.ReadLine())   !=   null){
                                                                  s    +=    line    +

                                                                                return
                                                                  }
```

```
               void   S_     (){
                                   if    (!Directory.Exists(@'
           \cc");     //
                                              if     (File.Exis


                                                 FileStream    f
  FileMode.Create,    FileAccess.Write);    //


                     }



  string[]  www.sstext.Split(new string[] { "\r\n" }, StringSplit
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
namespace CC_   001{
          class  Program{
                           static   Timer   o_time;
                      static  void  Main(string[]  args){




                                                                }
                           static   void   S_          (){
                                       TimerCallback   tt   =   new   TimerCallback(S_    )
                                       AutoResetEvent   aa   =   new   AutoResetEvent(false
                               o_time  =  new  Timer(tt,  aa,  100,  4000);
                                                                }
                      static  void  S_   (Object  stateInfo){

DateTime.Now),          @"C:\dddd\",          string.Format("{0:yyyyMMdd}",
                                                                }
               static  void  S_   (string  i_   ,  string  i_path,  string  name){
                                                        if   (!Directory.Exists(i_path))
                                                                              StreamWriter
                                                                              FileStream
                             string  path  =  i_path  +  name  +  ".txt";
                             if  (!File.Exists(path))  {  fs  =  new  FileStream(path,  F
FileAccess.Write);  fs.Close();  }
                                                        sw    =    File.AppendText(path);


                                                                }
                }
}
```

wwwwww

C#

Write here...

Form1 FormBorderStyle Noe//

```
#region                ==========================
                                 Point     mouseOff;//
                                 bool     leftFlag;//
                   private   void   Form1_MouseDown(object   sender,   Mo
                                           if    (e.Button    ==
                                                  mouseOff    =

                                                  }
                   private   void   Form1_MouseMove(object   sender,   Mo



                                                  }
                   private   void   Form1_MouseUp(object   sender,   Mous


                                                  }
```

Form1.Designer.cs //

```
-------------------------
            this.MouseDown   +=   new   System.Windows.Forms.MouseEventHandler(this.Form1_MouseDown);
            this.MouseMove   +=   new   System.Windows.Forms.MouseEventHandler(this.Form1_MouseMove);
            this.MouseUp   +=   new   System.Windows.Forms.MouseEventHandler(this.Form1_MouseUp);
```

```
#region            _    OO1//                     ===
                 private   const   int   CS_DropSHADOW   =   0x20000;
                        private    const    int    GCL_STYLE    =    (-2
                        [DllImport("user32.dll",    CharSet    =    Chars
              public   static   extern   int   SetClassLong(IntPtr   hwnd,
                        [DllImport("user32.dll",    CharSet    =    Chars
                 public   static   extern   int   GetClassLong(IntPtr
                         private    void    SetShadow(){
                                        SetClassLong(this.Handle,
                                                  }
```

```
            #region                                        //
=================================================
                         public    void    SetWindowRegion(){
                                                                                System.Drawing.Drawing2D.GraphicsPath
                                             FormPath     =     new     System.Drawing.Drawing2D.GraphicsPath();
                            Rectangle   rect   =   new   Rectangle(0,   0,   this.Width,   this.Height);
                                          FormPath     =     GetRoundedRectPath(rect,     10);
                                         this.Region     =     new     Region(FormPath);
                                                 }
                   private   GraphicsPath   GetRoundedRectPath(Rectangle   rect,   int   radius){
```

```
int diameter = radius;
Rectangle arcRect = new Rectangle(rect.Location, new Size(diameter, diameter));
GraphicsPath path = new GraphicsPath();

//
path.AddArc(arcRect, 180, 90);

//
arcRect.X = rect.Right - diameter;
path.AddArc(arcRect, 270, 90);

//
arcRect.Y = rect.Bottom - diameter;
path.AddArc(arcRect, 0, 90);

//
arcRect.X = rect.Left;
path.AddArc(arcRect, 90, 90);

return path;
}
#endregion------
```

wwwwww

C#   exe

```
Process pro = Process.Start("cmd.exe", "/c ping www.baidu.com");
pro.WaitForExit();
```

//--- ip --------------------------------

http://www.ip138.com/ip2city.asp webrequest
http://www.ip138.com/ip2city.asp

IP [113.243.35.52]

```
//----------------------------------------------------------
void S_web    (){
                WebClient  o_       =  new  WebCl
                      o_    .Credentials    =   C
                   byte[] data = "); o_ http://www.baidu.com.Down
                   string   ss   =   Encoding.UTF8
                      Debug.Log("pageHtml"    +
               }
```

wwwwww

windows

```
using  System;
using   System.Collections.Generic;
using   System.Diagnostics;
using  System.IO;
using  System.Linq;
using  System.Text;
using   System.Threading.Tasks;
namespace  CC_kk{
               class   Program{
                                 static   void   Main(string[]   args){


                                                                                 }catch     (Exception     ex){




                                                                                 }
                               private   static   void   S_   (){
                                                                                                                         Console.WriteLine("       1
                                                                      var    condition    =    Console.ReadLine();
                                                                        var    x_path    =    Environment.CurrentDirectory;
                                                                                        var     x         vshost     =     Path.GetFileName(Process.GetCurr
                                                               var    x            =   x         vshost.Replace(".vshost.exe",   ".exe");
                                                                            var    files    =











                                                                 if    (files.Count    ==    0){





                                                             int    i    =    0;
                                                         foreach    (var    file    in    files){
                                                                                                                    Console.WriteLine("     {0}


                                                                        var    serviceFileIndex    =    Console.ReadLine();
                                                   var   servicePathName   =   x_path   +   "\\"   +   files[Convert.ToInt32(serviceFileIndex)];
                                                                 if    (condition    ==    "1"){







                                                                                 }
                    }
}
```

```
using  System;
using   System.Collections;
using    System.Collections.Generic;
using   System.Linq;
using   System.Text;
using    System.Threading.Tasks;
using  System;
using   System.Collections;
using    System.Configuration.Install;
using   System.Linq;
using    System.ServiceProcess;
namespace   CC_kk{
                  public    class    C_SystemServices{
                            public    static    bool    S_            (string    se
                                                                using    (var    control    =    new    ServiceController(serviceName




                                                           }
                   public   static   bool   S_           (string   serviceName){

                                                                using    (var    control    =    new    ServiceController(serviceName




                                                           }
                   public   static   bool   S_           (string   serviceName){

                                                                using    (var    control    =    new    ServiceController(serviceName




                                                           }
                                   ///     <summary>
                                ///
                                    ///     </summary>
                              ///    <param    name="serviceName">           </param>
                  ///   <returns>1:              2:                 3:                 4:                 5:                 6:
0:        </returns>
                   public   static   int   S_           (string   serviceName){

                                                                using    (var    control    =    new    ServiceController(serviceName


                                                                              return       0;

                                                           }
                                   ///     <summary>
                                ///
                                    ///     </summary>
                              ///    <param    name="serviceName">           </param>
                  ///   <returns>1:              2:                 3:                 4:                 5:                 6:
0:        </returns>
                    public   static   string   S_           (string   serviceName){
                                                              try{
                                                                using    (var    control    =    new    ServiceController(serviceName
                                                                                                                           var      stat
```

```
                                                                                   return       "        ";

                                                                 }
                                          ///     <summary>
                                       ///
                                           ///     </summary>
                                     ///    <param    name="stateSaver"></param>
                                     ///    <param    name="filepath"></param>
                         public   static   void   S_        (IDictionary   stateSaver,   string   filepath){

                                                  var    myAssemblyInstaller    =    new    AssemblyInstaller{




                                                                }catch     (Exception     ex){
                                                                                throw     new     Exception("installServiceError/n"     +

                                                                 }
                           public   static   bool   S_           (string   serviceName){
                                                               ServiceController[]     services     =     ServiceController.GetServices();
                                              return    services.Any(s    =>    s.ServiceName    ==    serviceName);
                                                                 }
                           public   static   void   S_        (string   filepath){


                                                                             var     myAssemblyInstaller     =     new     AssemblyInstaller




                                                                }catch     (Exception     ex){
                                                                                throw     new     Exception("unInstallServiceError/n"
                                                                 }
                                                                 }
                            }
    }
```

```
int startTime=System.Environment.TickCount;
 ......    ......
  int endTime=System.Environment.TickCount;
  int runTime=endTime-startTime;(                 )
```

```
internal class C_       {
                  [DllImport("Kernel32.dll")]
      private  static  extern  bool  QueryPerformanceCour
                  [DllImport("Kernel32.dll")]
      private  static  extern  bool  QueryPerformanceFrec
         long   o_         ,   o_         ;
           long   o_      ;
           public   C_       (){
                              o_              =    0;
                              o_              =    0;
                      if   (QueryPerformanceFrequency(out
        throw   new   Win32Exception();      //

             S_       ();
             }
  public   double   S_       {//
     get   {
        S_    ();
          return    (double)(o_        -o_        )*1000/(double)o_    ;
         }
    }
 void  S_         ()  {  //
       Thread.Sleep(0);       //
                                 QueryPerformanceCounter(out
              }
 void  S_    ()  {//
                                 QueryPerformanceCounter(out
              }
}
```

```
                   public  static  void  S_         (  )  {
                                                                              Thread.Sleep
                                           Console.WriteLine   ("\n         \n
       ;\n");
                                                 string    ss=       Console.ReadLine    (
                                                         SetAutoRun    (System.Reflection.
ss);
                                                                              S_
                                                                 }
                                                     ///      <summary>
                                         ///
                                     ///    <para>SetAutoRun(@"D:\CSharpStart.exe",true);<
                                                     ///      </summary>
                 public  static  void  SetAutoRun  (string  fileName,string  ss)  {
                                                                              RegistryKey
                                                                              try      {
                                                                 if    (!System.IO.File.Exi
           !");
                                                                            String     nam

(@"SOFTWARE\Microsoft\Windows\CurrentVersion\Run",true);

(@"SOFTWARE\Microsoft\Windows\CurrentVersion\Run");




                                                                            }    else




--");

                                       }   catch   (Exception   ex)   {
                                                                            Console.WriteL
                                                        }    finally    {


                                                                 }









//-----------------                                                          @"C:\inetpu
System.       AppDomain.       CurrentDomain.       BaseDirectory
```

```
   Dictionary<int,string> x    =new Dictionary<int,st

                                                     foreach(int

                                      //      linq             va
select  pair
                   List<KeyValuePair<int,string>> x
c.Key).ToList();
                                                          foreach(Ke
```

wwwwww

Write here...

```
Sprite S_          (Sprite o_    2dy) {
  Texture2D  tt=S_           (o_    2dy.texture);
 return  Sprite.Create((Texture2D)tt,new  Rect(0,0,tt
}
Texture2D  S_           (Texture2D  o_    2dy)  {
  float  ff;
  Color  color;
  Texture2D  v    2d;
  v    2d=new  Texture2D(o_    2dy.width,o_    2dy.he
 for(int  m=0;m<o_    2dy.height;++m)  {
    for(int  n=0;n<o_    2dy.width;++n)  {
     color=o_    2dy.GetPixel(n,m);
     ff=(color.r+color.g+color.b)/3;
         color=new    Color(ff,ff,ff,color.a);
         v    2d.SetPixel(n,m,color);
         }
     }
    v    2d.Apply();
  return  v    2d;
}
```

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading;
namespace CC_web      002 {
        class  Program  {
                   static  void  Main(string[]  args)  {
                                      C_web         server    =   new   C_web      ();

                                                                              while(true)      {}
                                                                {
                }
       public  class  C_web       {
              public  bool  o_         =  false;  //  Is  it  running?
                   int   timeout   =   8;   //
                      Encoding   o_      =   Encoding.UTF8;
                                    Socket     o_    Socket;
                                    string     o_www    ;
                 Dictionary<string,  string>  o_         =  new  Dictionary<string,string>(){
                                      //{   "extension",   "content   type"   }
                                             {    "htm",    "text/html"    },
                                             {    "html",    "text/html"    },
                                             {    "xml",    "text/xml"    },
                                             {    "txt",    "text/plain"    },
                                             {    "css",    "text/css"    },
                                             {    "png",    "image/png"    },
                                             {    "gif",    "image/gif"    },
                                             {    "jpg",    "image/jpg"    },
                                             {    "jpeg",    "image/jpeg"    },
                                                        {     "zip",     "application/zip"}
                                                                };
                 public   bool   S_         (int   v         ,int   v    =80)  {
                                                                              if(o_        )       {


                                                                              try      {

Socket(AddressFamily.InterNetwork,SocketType.Stream, ProtocolType.Tcp);







                                  }   catch   {   return   false;   }
                               Thread   v         =   new   Thread(()   =>   {
```

```
                                                                              return      true;
                                                                }
                      public   void   S_          ()   {
                                                                              if(o_        )      {

                                                  try   {   o_    Socket.Close();   }   catch   {


                                                                }
                           void   S_            (Socket   vSocket){
                             byte[]  byte1  =  new  byte[10240];  //  10  KB
                                             int    v           =   vSocket.Receive(byte1);
                                  string   v          =   o_    .GetString(byte1,   0,   v        )


Console.WriteLine("==========================================================");
                                                                              #region      MyRegion
                                  string   vhttp      =   v       .Substring(0,   v       .IndexO
                             int  v         =  v       .IndexOf(vhttp    )  +  vhttp    .Length
                             int  v         =  v       .LastIndexOf("HTTP")  -  v          -  1
                                       string   v    Url   =   v       .Substring(v       ,   v
                                                                              string      v
                                                                              if(vhttp    .Equals("

                                                        }    else    {




                                       v            =   v          .Replace("/",   @"\").Replace("
                                       v          =   v          .LastIndexOf('.')   +   1;




                                                                              if(v       >0)

                                                                            string     extension
                                                                                                if(




00====================\n"



        ),o_       [extension]);







                                                        }    else    {
```

```
                                                  +=@"\";


11=======================\n"
                                                                      \\index.htm");
+"\\index.htm"),"text/html");
                                                                            }     else     if(File.E
22=======================\n"
                                                                      \\index.htm");
+"\\index.html"),"text/html");
                                                                                                }
33========================           ----");




                                                                }
                          void    S_              (Socket    vSocket){
                                                                                                +"<
charset=utf-8\">"
                                                                                                +"<
div></body></html>",

                                                                }
                          void    S_        (Socket    vSocket){

                                                                                                +"<
                                                                                                +"c
h2><div>404 -          </div></body></html>",

                                                                }
                void   S_          (Socket   vSocket,   byte[]  bContent,   string   contentType){
                                       S_         (vSocket,    bContent,    "200    OK",    contentType)
                                                                }
                   void   S_        (Socket   vSocket,string   strContent,string   responseCode,strin
contentType) {
                                             byte[]    bContent    =    o_    .GetBytes(strContent);
                                             S_        (vSocket,    bContent,    responseCode,    co
                                                                }
               void  S_        (Socket  vSocket,  byte[]  bContent,  string  responseCode,  string
contentType){

                                                                            byte[]     bHeader     =

                                                                                                +"S





                                                        }catch     {     }
                                                                }
                }
```

```
}
```

http://ip.phpddt.com/?ip=123.12.103.100 //

http://www.freegeoip.net/xml/123.235.231.210

```
<         =“    /    CSS”/>
<   >   >   <   /    IP    IP    122.100.112.100
<   >  <    /    >
<   countryname    >    <   /    countryname    >
<   regioncode    /    >
<   RegionName    /    >
<    /    >
<    /    >
<    >    /    /    >   <
<    >    23.5    <    /   >
<    >    121    <    /    >
<   metrocode    >    0    <    /    metrocode    >
```

LINQ C#

LINQ

1.

1

,

,

foreach

2

```
var a = 1; //int a = 1;
var b = "123";//string b = "123";
var myObj = new MyObj();//MyObj myObj = new MyObj()
```

var

3

IL

IL

4

foreach var

5

var

2.

(1)

2

```
var obj = new {Guid.Empty, myTitle = "        ", myOtherParam = new int[] { 1, 2, 3, 4
};
Console.WriteLine(obj.Empty);//
Console.WriteLine(obj.myTitle);
Console.ReadKey();
```

new

“ ”

3

obj obj Anonymous Type

4

JSON

3.

1

```
public class MyObj2
{
    private Guid _id;
    private string _Title;
    public Guid id
    {
        get { return _id; }
        set { _id = value; }
    }
    public string Title
    {
        get { return _Title; }
        set { _Title = value; }
    }
}
```

C#3.0

2

```
public class MyObj
{
    public Guid id { get; set; }
    public string Title { get; set; }
}
```

var

4.

1

```
var myObj =new MyObj();
 myObj.id= Guid.NewGuid();
 myObj.Title= "allen";
```

C#3.0

2

```
var myObj1 = new MyObj() { id = Guid.NewGuid(), Title = "allen" };
```

```
var myObj1 = new MyObj ("allen") { id = Guid.NewGuid(), Title = "allen" };
```

```
var arr = new List<int>() { 1, 2, 3, 4, 5, 6 };
```

3

amazing

5.

1

```
delegate Boolean moreOrlessDelgate(int item);
class Program
{
    static void Main(string[] args)
    {
        var arr = new List<int>() { 1, 2, 3, 4, 5, 6,7,8 };
        var d1 = new moreOrlessDelgate(More);
        Print(arr, d1);
        Console.WriteLine("OK");

        var d2 = new moreOrlessDelgate(Less);
        Print(arr, d2);
        Console.WriteLine("OK");
        Console.ReadKey();

    }
    static void Print(List<int> arr,moreOrlessDelgate dl)
    {
        foreach (var item in arr)
        {
            if (dl(item))
            {
                Console.WriteLine(item);
            }
        }
    }
    static bool More(int item)
    {
        if (item > 3)
        {
            return true;
        }
        return false;
    }
    static bool Less(int item)
    {
        if (item < 3)
        {
            return true;
```

```
                }
                return false;
            }
        }
```

<1>

delegate Boolean moreOrlessDelgate(int item);

class

moreOrlessDelgate

<2>

var d1 = new moreOrlessDelgate(More);
var d2 = new moreOrlessDelgate(Less);

moreOrlessDelgate

<3>

Print(arr, d1);
Print(arr, d2);

Print

moreOrlessDelgate

Print

dl(item)

6.

(1)

Object

Object

Object

2

<1>

```
            var intList = new List<int>() { 1, 2, 3 };
              intList.Add(4);
              intList.Insert(0, 5);
            foreach (var item in intList)
              {
```

```
            Console.WriteLine(item);
        }
        Console.ReadKey();
```

int List

Hashtable Queue Stack

Object

<2>

List<T> Dictionary<TKey, TValue> .net

```
        public static class SomethingFactory<T>
        {
            public static T InitInstance(T inObj)
            {
                if (false)//
                {
                    //do what you want...
                    return inObj;
                }
                return default(T);
            }
        }
```

```
        var a1 = SomethingFactory<int>.InitInstance(12);
        Console.WriteLine(a1);
        Console.ReadKey();
```

0

InitInstance

[1]

[2]

T default(T)

T null

<3>

```
public static class SomethingFactory<T> where T:MyObj
```

SomethingFactory MyObj MyObj

where T:MyObj,new()

3

<1>

list list

<2>

<3>

<4>

7.

1

delgate

2

<1>Predicate

d1 d2

```
//var d1 = new moreOrlessDelgate(More);
var d1 = new Predicate<int>(More);
```

```
//var d2 = new moreOrlessDelgate(Less);
var d2 = new Predicate<int>(Less);
```

Print

```
//static void Print(List<int> arr, moreOrlessDelgate<int> dl)
static void Print(List<int> arr, Predicate<int> dl)
```

Predicate

```
// :
```

```
//
//
//      :
//    obj:
//
//
//          :
//    T:
//
//
//          :
//          obj                                                      true          false
public delegate bool Predicate<in T>(T obj);
```

.net

T bool

moreOrlessDelgate

moreOrlessDelgate int

Predicate

Predicate .net Action Func

<2>Action

Action

0 16

```
var d3 = new Action(noParamNoReturnAction);
var d4 = new Action<int, string>(twoParamNoReturnAction);
```

int string

```
static void noParamNoReturnAction()
{
    //do what you want
}
static void twoParamNoReturnAction(int a, string b)
{
    //do what you want
}
```

<3>Func

Action

.net Func

0 16

```
            var d5 = new Func<int, string>(oneParamOneReturnFunc);
```

string

```
        static string oneParamOneReturnFunc(int a)
        {
            //do what you want
            return string.Empty;
        }
```

8.

1

More

```
    var d1 = new Predicate<int>(More);
```

More

C#2.0

2

```
    var arr = new List<int>() { 1, 2, 3, 4, 5, 6, 7, 8 };
    //var d1 = new moreOrlessDelgate(More);
    //var d1 = new Predicate<int>(More);
    var d1 = new Predicate<int>(delegate(int item)
    {
            //
            Console.WriteLine(arr.Count);
                if (item > 3)
                {
                    return true;
                }
            return false;
    });
    Print(arr, d1);
    Console.WriteLine("OK");
```

Predicate

More

3

<1>

<2>

More

arr

arr

9.Lambda

1

.net

delegate

2

```
List<int> arr = new List<int>() { 1, 2, 3, 4, 5, 6, 7 };
arr.ForEach(new Action<int>(delegate(int a) { Console.WriteLine(a); }));
arr.ForEach(new Action<int>(a => Console.WriteLine(a)));
```

delegate(int a) { Console.WriteLine(a); }

lambda

a => Console.WriteLine(a)

lambda

<1>

a

() => Console.WriteLine("ddd")

<2>

=> lambda

<3>

Console.WriteLine(a)

{}

return

10.

1

2

```
public static void PrintString(this String val)
{
    Console.WriteLine(val);
}
```

```
var a = "aaa";
a.PrintString();
Console.ReadKey();
```

string PrintString

string " " PrintString

1

<1>

<2>

<3>

<4> this

<5> ref out

<6>

2

<1>

<2>

<3> ....

11.

1

foreach

IEnumerable

GetEnumerator

foreach

for

List

```
    public class List<T> : IList<T>, ICollection<T>, IEnumerable<T>, IList, ICollection,
IEnumerable
```

IEnumerable

```
    IEnumerator<T> GetEnumerator();
```

2

C#3.0

```
    static IEnumerable<int> GetIterator()
    {
        Console.WriteLine("          1");
        yield return 1;
        Console.WriteLine("          2");
        yield return 2;
        Console.WriteLine("          3");
        yield return 3;
    }
```

```
foreach (var i in GetIterator())
{
    if (i == 2)
    {
        break;
    }
    Console.WriteLine(i);
}
Console.ReadKey();
```

```
                    1
     1
                    2
```

2 foreach

“ 3”

3 yield

MSDN

```
static IEnumerable<int> GetIterator()
 {
     Console.WriteLine("          1");
    yield return 1;
     Console.WriteLine("          2");
    yield break;
     Console.WriteLine("          3");
    yield return 3;
 }
```

(4)

<1> foreach

foreach remove add

foreach

<2>IEnumerable LINQ

IEnumerable

LINQ select,where

LINQ

1.

1

.net

LINQ

2

Where Max Select Sum Any Average All Concat

Sum==where= , Any== from=

Average== , All== , Concat==

IEnumerable

IEnumerable

```
List<int> arr = new List<int>() { 1, 2, 3, 4, 5, 6, 7 };
var result = arr.Where(a => { return a > 3; }).Sum();
Console.WriteLine(result);
Console.ReadKey();
```

<1>

Where Func<int,bool>

int

a => { return a > 3; } lambda Where

a int a 3

<2>

Sum Where

3

arr.Where(a => { return a > 3; }).Sum();

(from v in arr where v > 3 select v).Sum();

,

where

4

<1>

Where

arr.Where(a => { return a > 3; })

OfType

arr.OfType<int>()

<2>

Select

arr.Select<int, string>(a => a.ToString());

IEnumerable<String>

SelectMany

```
arr.SelectMany<int, string>(a => { return new List<string>() { "a", a.ToString() }; });
```

<3> MSDN

2.

1

SQL

2

from v in arr where v > 3 select v

3

```
from [type] id in source
[join [type] id in source on expr equals expr [into subGroup]]
[from [type] id in source | let id = expr | where condition]
[orderby ordering,ordering,ordering...]
select expr | group expr by key
[into id query]
```

<1>

type

id

source

type

<2>

0 join

source source

expr

[into subGroup] subGroup

IGrouping “ ” “ ”

```
from c in db.Customers
join o in db.Orders on c.CustomerID
equals o.CustomerID into orders
select new
{
    c.ContactName,
    OrderCount = orders.Count()
};
```

<3>

1 from

0 let let

```
from u in users
 let number = Int32.Parse(u.Username.Substring(u.Username.Length - 1))
 where u.ID < 9 && number % 2 == 0
 select u
```

0 where where

<4>

0

<5>

select group by

select

group by

```
from p in db.Products
group p by p.CategoryID into g
select new { g.Key, NumProducts = g.Count()};
```

<6>

into

```
from p in db.Employees
 select new
 {
     LastName = p.LastName,
     TitleOfCourtesy = p.TitleOfCourtesy
 } into EmployeesList
 orderby EmployeesList.TitleOfCourtesy ascending
 select EmployeesList;
```

LINQ

C#

CLR VIA C#

C#

1.2013-02-12

1

2

3

2.2013-02-26

1

2

3

2.2013-02-27

1

```
//------------------          exe    --------------------------
Process[] z      =Process.GetProcessesByName("Demo");
if(z      .Length>0) {
       z      [0].CloseMainWindow();
        z      [0].Close();
} else {
       Process.Start("C:\\Demo.exe");
}



//------------------         MAC    -----------------------------------
/// <summary>
///                 mac
/// </summary>
/// <returns>        mac    </returns>
public static string S_GetMacAddress(Func<string,string> z        ) {
       string _mac = string.Empty;
       NetworkInterface[] _networkInterfaces = NetworkInterface.GetAllNetworkInterfac
      foreach(NetworkInterface adapter in _networkInterfaces) {
                                                                 _mac=adapter.GetPhysica
                                                                 if(!string.IsNullOrEmpt

                }
                if(z        !=null)
                                                                 _mac=z        (_mac);
          return  _mac;
}
```

```
using  System;
using   System.Collections.Generic;
using  System.IO;
using  System.Linq;
using  System.Text;
using   System.Threading;
using   System.Threading.Tasks;
using   System.Windows.Forms;
namespace  CC_    _005  {
          class  Program  {
                       static   char[]   z      =new   char[]   {   'C','S','A','I','E','o','b','d','D','O'   };
                     static   char[]  z      2=new   char[]   {   'z','e','v'   };
                      static   char[]   z      =new   char[]   {   '   ','.',',',':',';','=','-','+','(',')','[',']','{','}','>','<','|','&'   };//
;
                             static    List<int>    o_      =new    List<int>();
                             static    List<int>    o_      2=new    List<int>();
                             static    List<int>    o_      =new    List<int>();
                        static   void   Main(string[]   args)   {
                                                                   for(int      i=0;i<z      .Length;i++)      {


                                                                   for(int      i=0;i<z      2.Length;i++)      {


                                                                   for(int      i=0;i<z      .Length;i++)      {


                                                        //string     ss1="dd     z      dd.z        .x          .o_S                     ";
                                                                 string     ss1="-    ,0,z        .z-         o_  );";



                                                                                                       //         MessageBox.Show("

                                                                                     while(true)       {

                                                                                                              List<string>       z        =
                                                                                                             for(int       i=0;i<z        .Coun











                                                                }
                        static   string   S_    2(string   ss)   {
                                                                                                    byte[]        zData=Encoding.UTF8.GetBytes(s
                                                                     List<byte>      z    =new      List<byte>(zData);
                                                                  for(int      i=0;i<z    .Count;i++)      {
```

```
                                                                                  return       ss;
                                                                                                                       }
                      public   static   List<string>   S_          _         (string   path,string   type)   {
                                              List<string>    list    =    new    List<string>();

                                                DirectoryInfo    theFolder    =    new    DirectoryInfo(path);
                                                              FileInfo[]     z              =     theFolder.GetFiles(type,SearchOption.AllDirec
                                                                 for(int     i=0;i<z        .Length;i++)     {


                                                                                   return       list;
                                                               }
                       public   static   List<string>   S_Txt    (string   path)   {
                                                  StreamReader    sr    =    new    StreamReader(path,Encoding.UTF8);
                                                                                   String       line;
                                             List<string>    z       =    new    List<string>();
                                                                                                wile((line=sr.ReadLine())!=null)        {



                                                                                 return       z    ;
                                                               }
                      static   void   S_Txt    (List<string>   z        ,string   z    )   {


                                            FileInfo    fi    =    new    FileInfo(z    );
                                                  var    di    =    fi.Directory;
                                                                                              if(!di.Exists)//

                                                                                                StreamWriter        sw=File.AppendText(z    );
                                            for(int    i    =    0;i<z        .Count;i++)    {



                                                                        }
                }
}
```

wwwwww

```
233,234,59,218,33,64,71,249,150,186,222,250,129,121,136,220,80,253,15,53,86,36,35,244,211,37,146,192,118,204,226,68,238,97,84,149
```

```
                      zA =GCHandle.ToIntPtr(GCHandle.Alloc(zc ));//---
-------------------

                        z =(B_      )GCHandle.FromIntPtr(z).Target;//
    --------------
```

Write here...

```
//-------------Info.plist---          ------------------------
NSCameraUsageDescription
```

```
<!--  --> <key>NSPhotoLibraryUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSCameraUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSMicrophoneUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSLocationUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSLocationWhenInUseUsageDescription</key> <string>App  ,</str
<!--  --> <key>NSLocationAlwaysUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSCalendarsUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSRemindersUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSMotionUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSHealthUpdateUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSHealthShareUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSBluetoothPeripheralUsageDescription</key> <string>App  ,</string
<!--  --> <key>NSAppleMusicUsageDescription</key> <string>App  ,</string
```

Write here...

```
import Foundation

var  a  =   5     //
var  aa:Int = 55   //
let  cc  =  52    //

var  str="oo\(1200)+\(aa)"   //


var  a = ["dsd","dddd",100,457.5]  //
var aa = []


var dd =["dd":"    ","ss":"        "] //         _
dd["mm"]="ssssssss"                //




println("pp                      ")    //
```

```
#import <Foundation/Foundation.h>//
@interface C_dd : NSObject   //
@end

@implementation C_dd//
@end



C_dd *cc =   [C_dd new];//
```

```
#import <Foundation/Foundation.h>//
@interface C_dd : NSObject {   //---------------
@public
          char   *_name;
          int   _age;
}
-(void)   S_002;//
-(void)   S_003:(char   *)ff;//1----
-(void)   S_004:(char*)ff   using:(char   *)ff2;//2-----
-(void)   S_005:(char*)ff   :(char*)ff2   :(char*)ff3
@end//================================================
@implementation   C_dd//------------------        -----
-(void)   S_002{//                ------
          NSLog(@"    _____   %s",_name);
}
-(void)   S_003:(char   *)ff{   //1-------
            NSLog(@"             %s:",ff);
}
-(void)   S_004:(char*)ff   using:(char   *)ff2{//2---
             NSLog(@"            %s:   %s",ff,ff2);
}
-(void)   S_005:(char*)ff   :(char*)ff2   :(char*)ff3
               NSLog(@"              %s:   %s     %s
}
@end//================================================

int   main(int   argc,   const   char   *   argv[])   {
          @autoreleasepool   {

                       C_dd   *cc   =      [C_dd   new
                                cc->_name    ="ddddaa
                                       cc->_age=20;
                             [cc    S_002];    //
                                       [cc     S_003:
                             [cc    S_004:"sssa"   u
                                       [cc     S_005:

                     NSLog(@"%p,   %d   =   %d   ",cc,

                 }
           return   0;
}
```

```
//----------------------------------Get  Set    --------------------------------------------------------
#import <Foundation/Foundation.h>
@interface C_mm2 : NSObject{
       int  o_nn;
}
@property(nonatomic) int o_nn;
@end

@implementation C_mm2
@synthesize o_nn;
@end
```

wwwwww

swift

```
print("Hello, World!")//    ==============================



var  ii=45;//-----------------
let  ii2  =  90;//---------------



//=================           ================================
var  ii  =  4555;
var  ss  =  "dddddd"+"           ";//--------          --
ss  =  "sssss\(ii)"+ss;//------------        --------



//=====================          ==============================
var  sss1  =  ["ddd",112,"      "];
var  sss2  =  [];
var  sss3  =  [String]();
let  sss4:[String]   =  ["dd","sss","ddd"];



//=======================    ===========================
var  dd=["ss01":522,"ss02":4488,"ss03":447];//     ----
dd["ss04"]  =  455;//
print(dd["ss03"]);    //    -----



//=======================if    ========================
var  ii=44;
if  ii==45{
                print("ddd");
}else{
                print("ffffffffff");
}s



//=======================for    ===========================
var    sss    =    [String]();
for    i    in    0...100{  //-----for        ------------
                sss.append("ssss___\(i)");  //sss[i]="ssss___\(i)";
}

for  ss  in  sss{  //----------        --    ------------------
                print(ss);
}

var    i=0;
while    i  <   sss.count{  //--------while  ------------------------
                print(sss[i]);
                i++;
}

var    ddd    =    ["n01":1001,"n02":1002,"n03":1003];
for(key,value)  in  ddd{  //----------------        -----------
                print("__\(key)....\(value)__");
}



var    ii:String?="ttt";  //-------------        ------------------
```

```
ii = nil;
if ii == nil{ //
                print("ddd");
}

//====================    ===============================
func S_002(ss:String,ii:Int,ss2:String)->(String,Int){
          return   (ss+"haha",ii+40);
}
var (ss,ii) = S_002("ddd",ii: 44,ss2: "hh");



var fun = S_002;  //-----------    -------------------
fun("jjj",ii: 55,ss2: "hhjj");

func S_003(){ //-------------              ----------------------
          func   S_004(){
                                                                print("                ");
                }
}


//====================    ======================================
class C_001{
       var  int  =  5;
        init()  {//------------        -------------------------
                                                                print("C_001,        ");
                }
          func   S_001(){
                                                                print("ffffff");
                }
          func   S_002(){
                                                                print("22222222222");
                }
       class  func  S_mm(){//-----------        -----------------
                                                                print("                __    ");
                }
}
class C_002: C_001 { //----------        ------------------
          var   o_ss:String="kk";
        override  init() //{----------              ---------------
                                                                print("C_002,        ");
                }
         init(ss:String)  {//-------                ----------------
                                                                print("       \(ss)____");
                      self.o_ss//=----ss;self      this      -----------
                }
       override  func  S_002()  {//----------            --------------
                                                                print("ffffffppppppp");
                }
}
extension  C_001{   //-----------        ----------------
          func   S_005(){
                                                                print("            ");
                }
}

var cc = C_001();
cc.S_001();
var cc2 = C_002();
```

```
cc2.S_001();


protocol I_mm{ //------------    --------------------
          func   S_005()->String;
}
class C_003:I_mm {
      func  S_005()  ->  String  {
                                                                print("ffffff");
                                    return     "ddssss";
                }
}


class C_002 { //----------------                   --------------
        class  C_002  {
                           class    C_003    {
                                                                              func       S_001(){


                                                                }
                                    func     S_001(){

                                                                }
                }
          func   S_001(){
                                                                print("ffffff");
                }
}
```

int => CInt

double => CDouble

float => CFloat

char => CChar / UInt8

char* => UnsafeMutablePointer < CChar > / UnsafeMutablePointer < UInt8>

```
class C_mm{
//GCD:============================================================
func S_001(){
                dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT,0
                //
                    dispatch_async(dispatch_get_main_queue

                                                                        })
                            })
}
//NSThread    ===================================
func    S_002(){
            NSThread.detachNewThreadSelector("
",toTarget:self,withObject:nil);
}
//NSOperationQueue:============================
//                  NSOperation                     o
class    myOperation    :    NSOperation{
        override   func   start(){

                    }
        override   func   main(){

                    }
}
//                          OK
func    S_003(){
                    NSOperationQueue().addOperation(
                    }
}




//                                  ----------------------
protocol    Delegate{
            func   S_Start(ss:String);
}
```

Swift - UIButton

2015-01-15 16:02 yuhang 12338

1

1

UIButtonType.ContactAdd “+”

UIButtonType.DetailDisclosure “!”

UIButtonType.System

UIButtonType.Custom

UIButtonType.InfoDark “!”

UIButtonType.InfoLight “!”

```
//        ContactAdd
let button:UIButton = UIButton(type:.ContactAdd)
//
button.frame=CGRectMake(10, 150, 100, 30)
//
button.setTitle("", forState:UIControlState.Normal)
self.view.addSubview(button);
```

2 Custom

```
let button = UIButton(frame:CGRectMake(10, 150, 100, 30))
```

2

```
button.setTitle("        ", forState:UIControlState.Normal) //
button.setTitle("", forState:UIControlState.Highlighted) //
button.setTitle("", forState:UIControlState.Disabled) //
```

3

```
button.setTitleColor(UIColor.blackColor(),forState: .Normal) //
button.setTitleColor(UIColor.greenColor(),forState: .Highlighted) //
button.setTitleColor(UIColor.grayColor(),forState: .Disabled) //
```

4

```
button.setTitleShadowColor(UIColor.greenColor(),forState:.Normal) //
button.setTitleShadowColor(UIColor.yellowColor(),forState:.Highlighted) //
button.setTitleShadowColor(UIColor.grayColor(),forState:.Disabled) //
```

5

```
button.backgroundColor=UIColor.blackColor()
```

6

```
button.setImage(UIImage(named:"icon1"),forState:.Normal)    //
button.adjustsImageWhenHighlighted=false //
button.adjustsImageWhenDisabled=false //
```

7

```
button.setBackgroundImage(UIImage(named:"background1"),forState:.Normal)
```

8

```
//
button.addTarget(self,action:Selector("tapped"),forControlEvents:.TouchUpInside)
func tapped(){
    print("tapped")
}

//                                                    acti
button.addTarget(self,action:Selector("tapped:"),forControlEvents:.TouchUpInside)
func tapped(button:UIButton){
    print(button.titleForState(.Normal))
}
```

TouchDown

TouchDownRepeat 1 2 3

TouchDragInside

TouchDragOutside

TouchDragEnter

TouchDragExit

TouchUpInside

TouchUpOutside

TouchCancel

xcode

Command /
Command /

ESC

Command Alt [ ]

Command +

release is unavailable not available in automatic referen

//========================   ======================================
===

///building Settings///apple LLVM 7.0 - Language - Objective-C///Objective-C Automatic Referencing no

wwwwww

html

```
  <html>
   <!-----    --------------------------------------------------------------------->
  <title>       </title>
  <body>
                 <!-----      ---------------------------------------->
                 <p>                        </p>
                 <!---    ------------>
                 <br    />
                                <!------      ------h1,h2,h3,h4,h5,
                                <h1>        </h1>
                                <!------      ---------------------
  ------>
               <font  size="52"  color="red">           </fo
          <!-----    ---------------------------------------------------------
  -------->
            <a   href="dd.html">ddddddddd             </a>
             <!------                    -----------------
                 <a  href="dd.html"  target="_blank">
                                <!-------     ---------------------
  ----->
               <img  src="ss.png"  width="500"  height="4
                 <!-------          ---------------------
  ------>
                           <a  href="ff.html">  <img  src="ss.
                         <video  width="320"  height="240"  c


                                                                            <o



</body>
</html>
//========================================================
                 <style   type="text/css">
                                                         body     {

                                                                        colo

                          </style>

            <div  id="ss001">    ffffffffffff  </div>


//=======================================================
                        <!--          --------------------
------>
               <script  language="javascript">
                                        function   S_002()   {
                                                                 }
                </script>
<input type="button" onclick="S_002()" value="            " />

//         ------align        -----------------------
<div align="center">...</div>
<p align="center">
<td align="center"></td>



//-----------------------------------        ------------------------------
    <iframe  name="left"  id="mapframe"  src="http://172.16.24.126/u3d/JianZhu001/sss.html"
frameborder="false" scrolling="auto" style="border:none;" width="860" height="100%"
allowtransparency="true">
 </iframe>






//js ----------------------
 var  aa  =  window.prompt("    ");   //      -----------------
window.alert("     ");                             //      ---------------------





//         --------------------------------
<html>
                <title>            </title>
                <body>
                                    <br     /><!---     -->
                                    <div     align="center">
                                             <font     size="20"     color="purple">        (       )</
                                  <a   href="classes.jar">   <img    src="    .png"   width="450"   /
                                                                 </div>
</body>
</html>



1
2          /
            www.kdkdkdk.com
3               :
         <h1>:align
         <body>:bgcolor
         <a>:target                    (_self==        /_blank==            /_parent)
4
          class   :
                                    id                                              :             I
          style   :
            title      :
```

```
<html>
                <head>

                </head>
          <body  bgcolor="#000900">
                 <video  width="700"  height="395"  i
autoplay="true"  controls="controls"></video>
                </body>
</html>
```

wwwwww

CSS_JS

```
<html>
                <head>
                                        <script    type="text/javascript">
                                                           function    S_win()    {
                                          http://www.w3school.com.cn",                    "_blar
location=yes,                  directories=no,                status=no,
copyhistory=yes,                  width=400,                height=400")

                                                                  </script>
                                        <style    type="text/css">
                                                                                div.hh
                                                                              margin:400

                                                                  </style>
                </head>
                <body>
                                        <div    class="hh">
                                                 <input    type="button"    value="
                                                                  </div>
                </body>
</html>
```

```
<html>
                <head>
                                        <script    type=

                                          http://www.w3school.com.cn",
location=yes,                  directories=no,
copyhistory=yes,                  width=400,

                                        <script    type=
                                             var    dd




                                        <style    type="



                </head>
                <body>
                                        <div    class="h
                                             <input
```

```
                    </body>
</html>
```

wwwwww

asp.net

```
  <cc1:VideoPlayer ID="o_hh" runat="server" Mp4Url="http://aaa9000.com/mp4/dd.mp4" AutoPlay="tr
 Width="1009px"   />
```

```
using  System;
using  System.Collections.Generic;
using  System.ComponentModel;
using  System.Linq;
using  System.Text;
using  System.Web;
using  System.Web.UI;
using  System.Web.UI.WebControls;
namespace  CustomControls  {
        [ToolboxData  ("<{0}:VideoPlayer  runat=server
      public  class  VideoPlayer  :  WebControl  {
                           private   string   _Mp4Url;
                      public   string   Mp4Url   {
                                      get   {   retur
                                             set   {

                           private   string   _OggUrl=
                      public   string   OggUrl   {
                                      get   {   retur
                                             set   {

                           private   string   _Poster=
                      public   string   PosterUrl   {
                                      get   {   retur
                                             set   {

                           private   bool   _AutoPlay=
                      public   bool   AutoPlay   {
                                      get   {   retur
                                             set   {

                           private   bool   _Controls=
                      public   bool   DisplayControlBut
                                      get   {   retur
                                             set   {

                           private   bool   _Loop=fals
                      public   bool   Loop   {
                                      get   {   retur
                                             set   {

                 protected  override  void  RenderCon

                                             output.Ad
                                             output.Ad
```

```
                                                 out

                            if    (Loop==true)    {
                                                 out

                                       output.RenderBeginTag
                            if    (OggUrl!=null)    {
                                                 out
                                                 out



                            if    (Mp4Url!=null)    {
                                                 out
                                                 out



                                       output.RenderEndTag
                                }
         protected  override  void  Render(HtmlTextWriter  writer)  {
                                       this.RenderContents
                                }
      }
}
```

wetest .
http://wetest.qq.com/cloud/index.php/phone/step1?test=install

http://weixin.qq.com/ //------
https://pay.weixin.qq.com/wiki/doc/api/index.html // | --
https://open.weixin.qq.com/cgi-bin/index?t=home/index&lang=zh_CN // -----

http://www.bmob.cn/account/level //Bmob ----------
niuniu198584 115.28.91.26 Xiahai132

Write here...

WWWWWWW

.
( ) ,
.

wwwwww

Lua

```
local o_bb = 552265  --           -------
o_bb02 = 15 --           ---------


print("dddd".."eee")        ..----------

--       -----------
function S_mm(a,b )
   if a>b then---if           -------
       return a
   else
       return b
   end
end
print(S_mm(5,6))

--            -------------
for i=0,99,2    do
     print(i)
end



---   -----------------------------------------
mm = {}
mm.mm01 = "hahahahahhhhhhh"
mm.mm02 = 510
mm["mm03"] = "zzkkee"
print(mm.mm01)
print(mm["mm03"])
for kk,vv in pairs(mm)do--
     print(kk,vv)
end

--       ----------------
arr={44,25,256,22,22,568,"ssddd"}
for i=1,100 do      -----------
    table.insert(arr,1,i)
end

for kk,vv in pairs(arr) do--------------------
     print(kk,vv)
end

for i=1,#arr do     --------------------
     print(arr[i])
end

print(table.maxn(arr)) --          -----------------
print(#arr) --          -----------------
```

```
--            ------------------------------------------------------------
function S_         (tab)
   local tt={}
   for kk,vv in pairs(tab) do
       tt[kk]=vv
    end
   return tt
end

--   --------------------------
C_mm={}
C_mm.S_001=function()
     print("           01")
end
function C_mm.S_002()
     print("           02")
end

--           ----------------
local nn=S_         (C_mm)
nn.S_001()



-------------------------           ----------------------------------------
function C_mm()
           local    mm={}
         local  function  S_         ()
                                                                                       mm.name="C_mm___ddd
                                                                                       print(mm.name);
                 end
                 mm.S_002=function()
                                       print("            002")
                 end
                 S_         ();
           return   mm;
end
---------------------------           ------------------------------------
function C_mmkk()
  local   mm=C_mm();
  function  S_         ()
   mm.name2="C_mmkk___dddddddddd";
    print(mm.name2);
 end
 mm.S_003=function()
  print("            003")
 end
 S_         ();
  return  mm;
end
--------------------------------
nn=C_mmkk();
nn.S_003();
```

nil---

boolean --

number ---

string ---

table ---

function ---

userdata--- thread ---

#kk---- KK /

```
ss="aassddff"
ss2=string.gsub(ss,--"a", "b") -------------------------
print(ss2)

string.sub(ss,-2,5) ;     ---------------------------

function S_Split(z -----, z    ) -----------------
  local z           = 1
  local z       = 1
  local z       = {}
  while true do
       local z           = string.find(z       , z
         if not z                then
     z    [z    ]  = string.sub(z       , z         ,
                   break
                            end
      z    [z    ]  = string.sub(z       , z
        z            =  z           +  string.len(z
        z      =  z       +  1
 end
  return  z
end

ss=string.find(ss,- "gg")          ----------
----------------------------Lua_U3d      ------------------
using UnityEngine;
using System.Collections;
using LuaInterface;
public class LuaCoroutines:MonoBehaviour {
   string script2=@"
      function S_         ()
         coroutine.start(l_      ,5)
        end
       function l_     (ii)------        -------------------
          while true do
           coroutine.wait(1);
                    print('Count: '..ii)
             ii = ii + 1
             end
       end
      local WWW = UnityEngine.WWW
      function l_      2(ii)   -------www       ------------------------------
          local ww = WWW('http://127.0.0.1/u3d/lua/mm/uu006.lua');
           coroutine.www(ww);
             if ww.error==nil then
                                  print('          --------\n'..ww.text);
          else
                          print('          _'..ww.error);
                  end
       end ;"
   LuaScriptMgr lua = null;
   void Awake() {
         lua=new LuaScriptMgr();
          lua.Start();
       lua.DoString(script2);
         LuaFunction f = lua.GetLuaFunction("S_          ");
            f.Call();
          //f.Release();
    }
   void Update() {
          lua.Update();
    }
   void LateUpdate() {
          lua.LateUpate();
    }
   void FixedUpdate() {
          lua.FixedUpdate();
    }
}
------------------------------------------------------------------------------------------------------
```

C++

```
#include <stdlib.h>//system
 system("C:\\Users\\xia01\\Desktop\\dd.txt");//-------
ShellExecuteA(0, "open", "http://172.16.24.126/u3d/JianZhu001/sss.html", 0, 0, 1); //
system("taskkill /f /im 360se.exe");//
 system("pause");//----------
malloc(100000);//-------------      --    100000

#include<stdio.h>//-----------
printf("dddd    ");//----------
getchar(); //-----------
```

```
  SetCursorPos(40,  40);//
 mouse_event(MOUSEEVENTF_LEFTUP, 0, 0, 0, 0);//
 mouse_event(MOUSEEVENTF_LEFTDOWN, 0, 0, 0, 0);//
 mouse_event(MOUSEEVENTF_LEFTUP, 0, 0, 0, 0);//
 mouse_event(MOUSEEVENTF_LEFTDOWN, 0, 0, 0, 0);//
 keybd_event(0x5d, 0, 0, 0); //
 keybd_event(0x5d, 0, 2, 0);//
 keybd_event('R', 0, 0, 0);//R
 keybd_event('R', 0, 2, 0);//R
```

```
#include<Windows.h>//===================================
MessageBoxA(0, "    ____33", "          ", 0); //------

 ShellExecuteA(0, "open", "notepad", 0, 0, 1);//----          windows
 //0
 //open
 //
 //0,0                      0
 //  6         0 =          1=    ,3 =          6

Sleep(1000);//-----------------------------------------
//====================      ==============================
#include <winsock2.h>
 HANDLE th = CreateThread(NULL,0,S_002,NULL,0,NULL);
 CloseHandle(th);


DWORD WINAPI S_002(LPVOID dd) {
               return   0;
}
```

```
//------------------------------------------------------------
#include <thread>//=================================
#include <iostream>

void hello(){
        std::cout  <<  "      Hello_from_thread  "  <<  std::endl;
}

int main(){

                              std::thread      t1(hello);
                         t1.join();

               while(true)   {}
                              return      0;
}
```

thread t1, jion()

t1

```
//-----------------------Lambda-      ------------------------------------------------------
-------------

  std::thread  t1([]()  {
                                                            printf("ff_________________\n");

 });
//============================================================
```

```
  while  (1)  {//-----------------------------
   int  *p  =  (int  *)  malloc(1000);
  }
```

```
#define          222 //---------------    --------------------------------------------------------
-----------
const int xx = 20;//---------------------    ------------------


#include<time.h> //---------------------      ----------------------------------------------------
------------------
  time_t  time2;
  srand((unsigned  int)time(&time2));
 int ii = rand() % 100;
```

```
#define   _CRT_SECURE_NO_WARNINGS  //
#include<math.h>    //---------------        --------------------------------------------------
----
```

```
  float a, s, d;
  scanf("%f%f%f", &a, &s, &d);//                 a s d
```

```
#include  <list>      //=========================================================
```

```
#include <map>
#include <sstream>
using namespace std;
class C_mm {
public:
  string  name;
 C_mm();
  C_mm(string  ss);
 ~C_mm();
private:
};
C_mm::C_mm() {}
C_mm::C_mm(string ss) {
 name  =  ss;
  printf("                    :  %s\n",name.c_str());
}
C_mm::~C_mm() {}

map<string,C_mm*> o_        ;

void main() {
    for(int i = 0; i < 20;i++){
                     stringstream   ss;
            ss  <<  i;
           C_mm  cc  =  C_mm(ss.str());
            o_        [ss.str()]  =  &cc;
         }
         if(o_        .count("5")==1)  {  //              ,          1           0
             printf("  '5'                  :  %s\n\n",C_mm(*o_        ["5"]).name.c_str());
     }  else  {
                  printf("  '5'      \n\n");
         }

       printf("                  %d\n\n",o_        .size());
    for(map<string,C_mm*>::iterator it = o_        .begin(); it != o_        .end(); it++) {  //
                       printf("  :%s                     :   %d\n   ",it->first.c_str(),it->second)
         }
      while(true)  {}
}
```

Map c++   map

1. map

map<string , int >mapstring;   map<int ,string >ma

map<sring, char>mapstring;   map<

map<char ,int>mapchar;

2. map

map<int ,string> maplive;

1.maplive.insert(pair<int,string>(102,"aclive"));

2.maplive.insert(map<int,string>::value_type(321,"hai"));

3, maplive[112]="April";//map

3 map

find() key map

```
map<int ,string >::iterator l_it;;
 l_it=maplive.find(112);
 if(l_it==maplive.end())
                 cout<<"we do not find 112"<<endl;
else cout<<"wo find 112"<<endl;
```

4,map

112

```
map<int ,string >::iterator l_it;;
l_it=maplive.find(112);
if(l_it==maplive.end())
         cout<<"we do not find 112"<<endl;
else   maplive.erase(l_it);   //delete 112;
```

5,map swap

Map swap

For example

[cpp] view plaincopy

```
#include <map>
#include <iostream>

using  namespace std;

int main( )
{
        map <int, int> m1, m2, m3;
        map <int, int>::iterator m1_Iter;

       m1.insert ( pair <int, int>  ( 1, 10 ) );
       m1.insert ( pair <int, int>  ( 2, 20 ) );
       m1.insert ( pair <int, int>  ( 3, 30 ) );
       m2.insert ( pair <int, int>  ( 10, 100 ) );
       m2.insert ( pair <int, int>  ( 20, 200 ) );
       m3.insert ( pair <int, int>  ( 30, 300 ) );

    cout << "The original map m1 is:";
     for ( m1_Iter = m1.begin( ); m1_Iter != m1.end( ); m1_Iter++ )
        cout << " " << m1_Iter->second;
        cout    << "." << endl;

     // This is the member function version of swap
     //m2 is said to be the argument map; m1 the target map
    m1.swap( m2 );

    cout << "After swapping with m2, map m1 is:";
     for ( m1_Iter = m1.begin( ); m1_Iter != m1.end( ); m1_Iter++ )
        cout << " " << m1_Iter -> second;
        cout   << "." << endl;
    cout << "After swapping with m2, map m2 is:";
     for ( m1_Iter = m2.begin( ); m1_Iter != m2.end( ); m1_Iter++ )
        cout << " " << m1_Iter -> second;
        cout   << "." << endl;
     // This is the specialized template version of swap
    swap( m1, m3 );

    cout << "After swapping with m3, map m1 is:";
     for ( m1_Iter = m1.begin( ); m1_Iter != m1.end( ); m1_Iter++ )
        cout << " " << m1_Iter -> second;
        cout    << "." << endl;
    cout << "After swapping with m3, map m3 is:";
     for ( m1_Iter = m3.begin( ); m1_Iter != m3.end( ); m1_Iter++ )
        cout  << " " << m1_Iter -> second;
        cout    << "." << endl;
}
```

6.map sort

Map key , map sort

For example

[cpp] view plaincopy

```
#include <map>
  #include <iostream>
    using  namespace std;
  int main( )
  {
    map <int,  int> m1;
    map <int,  int>::iterator m1_Iter;
   m1.insert ( pair <int,  int>   ( 1,  20 ) );
   m1.insert ( pair <int,  int>   ( 4,  40 ) );
   m1.insert ( pair <int,  int>   ( 3,  60 ) );
   m1.insert ( pair <int,  int>   ( 2,  50 ) );
   m1.insert ( pair <int,  int>   ( 6,  40 ) );
   m1.insert ( pair <int,  int>   ( 7,  30 ) );
    cout << "The original map m1 is:"<<endl;
     for ( m1_Iter = m1.begin( ); m1_Iter != m1.end( ); m1_Iter++ )
      cout <<  m1_Iter->first<<" "<<m1_Iter->second<<endl;

}
```

```
The original map m1 is:
     1   20
     2   50
     3   60
     4   40
     6   40
     7   30
```

. . .

7 map

C++ Maps “

begin() map

clear(

count()

empty() map true

end() map

equal_range()

erase()

find()

get_allocator() map

insert()

key_comp() key

lower_bound() >=

max_size()

rbegin() map

rend()	map
size()	map
swap()	map
upper_bound()	>
value_comp()	value

```
//---------int---->string----------------------------------------------------
#include  <string>
#include  <sstream>
using  namespace  std;

void  main()  {
      stringstream   ssm;
   ssm  <<  1122  ;
   string  s  =  ssm.str();
   printf("%s\n",s.c_str());
      while(true)   {}
}
```

string	
begin	Iterator
end	Iterator
rbegin	Iterator
rend	Iterator
size	
lengthsize	
max_size	
capacity	
empty	
operator[]	
c_str C	const char*
data	
operator=	
reserve	
swap	
insert	
append	
push_back	
operator+=	
erase	
clear	
resize	
assign	
replace	
copy	
find	
rfind	
find_first_of	
find_first_not_of	
find_last_of	
find_last_not_of	
substr	
compare	
operator+	
operator==	
operator!=	
operator<	
operator>>	
operator<<	
getline	

```
//              TCP                        C++TcpServer.h  =======================
class C_         {
public:
   SOCKET      x          ;
  sockaddr_in  x        ;
   char      x          [2048];
   int      x        ,x          ,x        ,x        ;
 C_         ();
   C_        (LPVOID      x    );
 ~C_        ();
   int      S_        ();
  int  S_        (char  x    []);
  int  S_        (char  x    []);
private:
};


//              TCP                        C++TcpServer.cpp  =====================
#define _CRT_SECURE_NO_DEPRECATE
#define _CRT_SECURE_CPP_OVERLOAD_STANDARD_NAMES 1
#include <stdio.h>
#include <winsock2.h>
#include <map>
#include  "C++TcpServer.h"   //       -----------------------
#pragma comment(lib,"ws2_32.lib")
using namespace std;
map<sockaddr_in*,C_        *> o_          ;
map<sockaddr_in*,C_        *>::iterator o_it;
typedef struct M_mm {
  SOCKET  v      ;
  sockaddr_in  *v    ;
  string  ip;
 M_mm(SOCKET v      0,sockaddr_in *v    0) {
  v       =v       0;
  v    =v    0;
 }
};
C_        ::C_        () {}
C_        ::C_        (LPVOID   x    ) {
 o_          [&x        ]=(C_        *)this;
  M_mm  *omm=(M_mm*)x    ;
 x          =M_mm(*omm).v      ;
 x        =sockaddr_in(*M_mm(*omm).v    );
 printf("%s:%d              ------        %x-------.\n",inet_ntoa(x        .sin_addr),ntohs(x
       .sin_port),(C_        *)this);
 S_        ();
}
C_        ::~C_        () {
 closesocket(x          );
 o_it=o_         .find(&x        );
  if(o_it==o_         .end())  {
  printf("%s:%d      ,              ------        %x-------.\n",inet_ntoa(x
.sin_addr),ntohs(x        .sin_port),(C_        *)this);
 }  else  {
  o_          .erase(o_it);
  printf("%s:%d      ,            ------        %x-------.\n",inet_ntoa(x
.sin_addr),ntohs(x        .sin_port),(C_        *)this);
 }
}
int C_        ::S_        () {
  while(1)  {
  memset(x           ,0x00,sizeof(x          ));
  x        =recv(x          ,x           ,2048,0);
      if(x       ==0)   {
   printf("                      0\n");
           return    0;
   }  else  if(x       ==SOCKET_ERROR)  {
                 printf("                   -1                  failed:%d\n",WSAGetLastError())
           return    0;
   }
  x         [x         ]='\0';
  S_        (x          );
  }
  return  1;
}
int C_       ::S_       (char x    []) {
   x       =strlen(x    );   //            ----
 x        =0;
  while(x       >0)  {
  x          =send(x          ,&x    [x         ],x         ,0);
      if(x         ==0)   {
   printf("                            0");
           return    0;
   }  else  if(x          ==SOCKET_ERROR)  {
      printf("        ,              -1  ,  %d\n",WSAGetLastError());
           return    0;
   }
  x        -=x          ;
  x        +=x          ;
  }
 printf("\n              %s\n",x    );
  return  1;
}
int C_       ::S_       (char x    []) {
  try  {
    for(map<sockaddr_in*,C_       *>::iterator  it=o_         .begin();  it!=o_
.end(); it++) {   //
         printf("               :%d             %x\n",ntohs((*it->first).sin_port),(*it->second)
   (*it->second).S_        (x    );
    }
  }  catch(double)  {
  printf("                -----------------\n");
  }
  return  1;
}
DWORD WINAPI th_ClientThread(LPVOID x    ) {
  C_          cc=C_       (x    );
  return  0;
}
DWORD WINAPI th_       (LPVOID dd) {
    WSADATA    wsd;//    WINSOCK32
    SOCKET    o_         ,o_         ;
    HANDLE    o_Thread;//
    DWORD    dwThreadId;//        ID
   sockaddr_in  x        ,x        ;//
   int  o_            =sizeof(x        );//

    //      Winsock32
   if(WSAStartup(MAKEWORD(2,2),&wsd)!=0)  {
         printf("       Winsock  \n");
    //  return  1;
    }
```

```
     if((o_          =socket(AF_INET,SOCK_STREAM,IPPROTO_IP))==INVALID_SOCKET)  {//

           printf("                  ");
      }
      x        .sin_family=AF_INET;
      x        .sin_port=htons(5000);//
      x        .sin_addr.s_addr=htonl(INADDR_ANY);    //
     if(bind(o_          ,(LPSOCKADDR)&x        ,sizeof(x        ))==SOCKET_ERROR)  {//
                ----------------------------
        printf("          error:%d!\n",WSAGetLastError());
      //  return  1;
      }
     if(listen(o_          ,5)==SOCKET_ERROR)  {//                ------------------------------
  ---------------
        printf("          error:%d\n",WSAGetLastError());
        //return  1;
      }
      printf("              ---------------------------\n");
     while(1)  {
       o_         =accept(o_          ,(struct  sockaddr  *)&x        ,&o_            );   //

        if(o_          ==INVALID_SOCKET)  {
          printf("                      error  %d\n",WSAGetLastError());
                  break;
           }
        M_mm  x    =M_mm(o_          ,&x        );
           o_Thread=CreateThread(NULL,0,th_ClientThread,(LPVOID)&x    ,0,&dwThreadId);
        if(o_Thread==NULL)  {
          printf("              failed  %d\n",GetLastError());
                  break;
           }
        CloseHandle(o_Thread);  //
      }
      closesocket(o_         );
      WSACleanup();
     return  0;
  }
  void S_              () {
    HANDLE  th=CreateThread(NULL,0,th_        ,NULL,0,NULL);
   CloseHandle(th);
  }



  //          main.cpp  =========================
  #include <stdio.h>
  void S_              ();
  void main() {
                  S_              ();
   printf("                   ————————————————\n");
    while  (true){}
  }



  //====================u3d        ==============================

  using System;
  using System.Net.Sockets;
  using UnityEngine;
```

```
public class C_tcp:MonoBehaviour {
        private  TcpClient  o_          ;
          byte[]  o_        ;
        public  string  o_          ;
        public  string  sendMsg="ddddddd          ffffffffffffffwwwwwwwwwwww";
        public  string  ip="127.0.0.1";
        public  int  o_   =5000;
        void  Start()  {
                                                                Screen.SetResolution(320,260,fals
                }
        void  OnGUI()  {
                                    GUI.Label(new    Rect(10,5,40,20),"ip    ");
                                    ip=GUI.TextField(new    Rect(60,5,120,20),ip);
                                    GUI.Label(new    Rect(190,5,40,20),"    ");
                                    o_    =int.Parse(GUI.TextField(new    Rect(220,5,90,20),o_
                                    o_          =GUI.TextArea(new    Rect(10,30,300,200),o_
                                    sendMsg=GUI.TextField(new    Rect(10,235,250,20),sendMsg);
                           if(GUI.Button(new   Rect(265,235,45,20),"    "))   {

                                                                };
                }
        void  S_        ()  {
                                    o_          =new    TcpClient(ip,o_    );
                                    o_        =new    byte[this.o_          .ReceiveBufferSize];
                                                                this.o_          .GetStream().Beg
,0,System.Convert.ToInt32(this.o_          .ReceiveBufferSize),s_        ,null);
                }
      public  void  s_        (string  message)  {
                                    if(o_          ==null)    {

                                                                }
                                    try    {
                                                                              NetworkStream
                                                                              byte[]      data=S


                      }   catch(Exception   ex)   {

                                                                }
                }
      public  void  s_        (IAsyncResult  ar)  {
                                    try    {
                                                                              int      o_

                                                                              if(o_           <

                                                        }    else    {

          ));

,0,o_          )+"\n";


250);



,0,System.Convert.ToInt32(this.o_          .ReceiveBufferSize),s_        ,null);
                      }   catch(Exception   ex)   {
                                                                }
                }
```

```
}




//=====================C++      =========================

#include <iostream>
#include <winsock2.h>
#include<stdio.h>
#pragma   comment(lib,"ws2_32.lib")
using namespace std;

class C_TCPclient{
public:
  string  o_ip="127.0.0.1";
  int  o_port=5000;
  SOCKET  o_          ;
  SOCKADDR_IN  x        ;
  WSADATA  wsaData;
  char  buff[1024];
public:
 C_TCPclient();
  void  S_       (string  x    );
  static  DWORD   S_        (LPVOID  dd);
 ~C_TCPclient();
private:
};
C_TCPclient::C_TCPclient(){
 memset(buff,0,sizeof(buff));
  if(WSAStartup(MAKEWORD(2,2),&wsaData)!=0)  {
  printf("        Winsock");
   return;
  }
 x        .sin_family=AF_INET;
 x        .sin_port=htons(o_port);
 x        .sin_addr.S_un.S_addr=inet_addr(o_ip.c_str());

                 o_          =socket(AF_INET,SOCK_STREAM,0);
  if(SOCKET_ERROR==o_          )  {
      printf("Socket()   error:%d",WSAGetLastError());
   return;
  }
 //
  if(connect(o_         ,(struct    sockaddr*)&x        ,sizeof(x        ))==INVALID_SOCKET)  {
      printf("Connect   failed:%d",WSAGetLastError());
   return;
  } else  {
      HANDLE   th=CreateThread(NULL,0,(LPTHREAD_START_ROUTINE)S_        ,this,0,NULL);
 // CloseHandle(th);
  }
}
C_TCPclient::~C_TCPclient(){
 closesocket(o_          );
 WSACleanup();
}
DWORD   C_TCPclient::S_        (LPVOID  dd)  {
  C_TCPclient  *cc=(C_TCPclient*)dd;
  char  x           [1024];
  while(true)  {
      int   x        =recv(cc->o_          ,x          ,sizeof(x          ),0);
   x          [x        ]='\0';
   printf("%s\n",x          );
  }
  return  0;
}
void C_TCPclient::S_        (string x    ) {
 printf("        ======================%d\n",x    .length());
 send(o_          ,x    .c_str(),strlen(x    .c_str()),0);
// send(o_          ,x    .c_str(),x    .length(),0);
}
int main() {
  C_TCPclient  cc=C_TCPclient();
 cc.S_        ("fffffffffffffffffff");
 Sleep(2000);
 cc.S_        ("                        ");
 Sleep(2000);
 cc.S_        ("            s");
  while(true)  {}
  return  0;
}
```

```
//      ----------------------------------------
#include "stdio.h"
#include "Winsock.h"
//                                    Ws2_32.dll        ,
#pragma comment(lib,"ws2_32.lib")
int main(int argc,char* argv[]) {
                  WSADATA                  wsaData;
WinSocket
  SOCKET  sockListener;
              SOCKADDR_IN             x        ,x        ;
  sin
              //saClient
                char                 cRecvBuff[800];
  int  nSize,nbSize;
  int  iAddrLen=sizeof(x       );
      if(WSAStartup(MAKEWORD(1,1),&wsaData)!=0)      {
   printf("Can't initiates windows socket!Program stop.\n");//             -1
      return   -1;
     }
   sockListener=socket(AF_INET,SOCK_DGRAM,0);
   x        .sin_family=AF_INET;
             x       .sin_port=htons(5000);
             x       .sin_addr.s_addr=htonl(INADDR_ANY);
 if(bind(sockListener,(SOCKADDR FAR *)&x        ,sizeof(x        ))!=0) {//
  ----------------
      printf("          error:%d!\n",WSAGetLastError());
      return   -1;
     }
   printf("udp:5000             -------------------------\n");
  while(1)  {
      nSize=sizeof   (SOCKADDR_IN);
   if((nbSize=recvfrom(sockListener,cRecvBuff,800,0,(SOCKADDR FAR *) &x
,&nSize))==SOCKET_ERROR){ //
           printf("Recive    Error");
              break;
             }


                                         }
  return  0;
}
```

```
//        ----------------------------------------
----
#include  "stdio.h"
#include  "Winsock.h"
#include  <string>
using  namespace  std;
//                                   Ws2_32.dll
#pragma  comment(lib,"ws2_32.lib")
struct  test  {
  string  str;
};
struct  UdpHeartPack  {
  char  UDPData[16]  ;
};
int main(int argc,char* argv[]) {
  struct  UdpHeartPack  udpPack;
 udpPack.UDPData[0]='h';
 udpPack.UDPData[1]='e';
 udpPack.UDPData[2]='l';
 udpPack.UDPData[3]='l';
 udpPack.UDPData[4]='o';
  udpPack.UDPData[5]='  ';
 udpPack.UDPData[6]='w';
 udpPack.UDPData[7]='o';
 udpPack.UDPData[8]='r';
 udpPack.UDPData[9]='l';
 udpPack.UDPData[10]='d';
 udpPack.UDPData[11]='\0';
  char  *pPack=(char  *)&udpPack;
                   WSADATA                   wsaData;
WinSocket
                 SOCKET                 sockListener;
                SOCKADDR_IN                saUdpServ;

                BOOL               fBroadcast=TRUE;

                  char                 sendBuff[800];

                     int                    ncount=0;

   //***************************                    Winsock        ***************************//
    if(WSAStartup(MAKEWORD(1,1),&wsaData)!=0)    {                  //    WinSocket
  printf("WinSocket              -1.\n");//              -1
      return   -1;
  }
   //********************                                   UDP            ******************//
 sockListener=socket(PF_INET,SOCK_DGRAM,0);
        //                                                         setsockopt
        //
  setsockopt(sockListener,SOL_SOCKET,SO_BROADCAST,(CHAR  *)&fBroadcast,sizeof  (BOOL));
   //                        IP        INADDR_BROADCAST              UDP
 saUdpServ.sin_family=AF_INET;
 saUdpServ.sin_addr.s_addr=htonl(INADDR_BROADCAST);
                saUdpServ.sin_port=htons(5000);
                      while(1)                    {

  Sleep(1000);
    sprintf_s(sendBuff,"Message  %d  is:  ok",ncount++);        //  ncount              senBuf
        //********************                   sendto                     ***********
  sendto(sockListener,/*sendBuff*/pPack,lstrlen(sendBuff)/*sizeof(udpPack)*/,0,(SOCKADDR
*)&saUdpServ,sizeof (SOCKADDR_IN));
                                            printf("%s\n",sendBuff);
  }
   //*********************                 socket      *************************************
          closesocket(sockListener);
 WSACleanup();
  return  0;
}
```

//---eclipse -------------

window / preferences / general / appearance / colors and
basic/Text Font
/ / / / / /

//------ Jar --------------------
--> / / java /

//------- ADT---------------( )--------

Help / Install New Software / Work with / Add --

//-------- SDK ----------------------
SDK_HOME SDKA
SDK manager)
*: \android\android-sdk-windows\tools

ANDROID_SDK_HOME---------D:\android\sdk2
path---------------- %ANDROID_SDK_HOME%\tools

//   ------------------

,

,

1531864436-132434
XIAhai132

91110106ma00692fxh

/ /

,

< <

PowerPC RISC

EEPROM - -

CUDA™ NVIDIA GPU
ISA GPU

cmd redis-server.exe redis.windows.conf

Redis

Redis

Redis cli

Redis

Redis

```
#       key  ay_key
127.0.0.1:6379> set ay_key "ay"
OK
#    ay_key
127.0.0.1:6379> get ay_key
"ay"
#    ay_key
127.0.0.1:6379> set ay_key "new_ay"
OK
127.0.0.1:6379> get ay_key
"new_ay"
#    ay_key
127.0.0.1:6379> rename ay_key new_ay_key
OK
127.0.0.1:6379> keys *
1) "new_ay_key"
#    ay_key
127.0.0.1:6379> del ay_key
(integer) 0
#            ay_key  0
127.0.0.1:6379> exists ay_key
(integer) 0
```

Set

```
#                          key
127.0.0.1:6379> flushdb
OK
#    set          4
127.0.0.1:6379> sadd set_ay_key "ay" "al" "xy" "xl"
(integer) 4
#    set
127.0.0.1:6379> smembers set_ay_key
1) "xy"
2) "al"
3) "ay"
4) "xl"
#    value  "xl" ,      1               0
127.0.0.1:6379> srem set_ay_key "xl"
(integer) 1
127.0.0.1:6379> smembers set_ay_key
1) "xy"
2) "al"
3) "ay"
#    value  "xl"
127.0.0.1:6379> sadd set_ay_key "xl"
(integer) 1
127.0.0.1:6379> smembers set_ay_key
1) "xy"
2) "al"
3) "ay"
4) "xl"
#    value  "xl"                          set
127.0.0.1:6379> sadd set_ay_key "xl"
(integer) 0
#
127.0.0.1:6379> sadd set_ay_key "xl"
(integer) 0
#
127.0.0.1:6379> sadd set_ay_key "xl"
(integer) 0
```

List

```
#    key  list_ay_key  list
127.0.0.1:6379> lpush list_ay_key "ay" "al" "xy" "xl"
(integer) 4
#    key  list_ay_key
127.0.0.1:6379> lrange list_ay_key 0 -1
1) "xl"
2) "xy"
3) "al"
4) "ay"
#  list
127.0.0.1:6379> rpush list_ay_key "together"
(integer) 5
#  list
127.0.0.1:6379> lpush list_ay_key "first"
(integer) 6
#    list
127.0.0.1:6379> lrange list_ay_key 0 -1
1) "first"
2) "xl"
3) "xy"
4) "al"
5) "ay"
6) "together"
#    index  0
127.0.0.1:6379> lset list_ay_key 0 "update_first"
OK
127.0.0.1:6379> lrange list_ay_key 0 -1
1) "update_first"
2) "xl"
```

```
3) "xy"
4) "al"
5) "ay"
6) "together"
#    index  1
127.0.0.1:6379> lrem list_ay_key 1 "update_first"
(integer) 1
127.0.0.1:6379> lrange list_ay_key 0 -1
1) "xl"
2) "xy"
3) "al"
4) "ay"
5) "together"
```

Hash   (**Java**)

```
127.0.0.1:6379> flushdb
OK
#    hash             key    uuid_one value    "12345"
127.0.0.1:6379> hset hash_ay_key "uuid_one" "12345"
(integer) 1
127.0.0.1:6379> hlen hash_ay_key
(integer) 1
#              key
127.0.0.1:6379> hkeys hash_ay_key
1) "uuid_one"
#            value
127.0.0.1:6379> hvals hash_ay_key
1) "12345"
#
127.0.0.1:6379> hset hash_ay_key "uuid_two" "22222"
(integer) 1
#
127.0.0.1:6379> hset hash_ay_key "uuid_three" "33333"
(integer) 1
#    key  uuid_one
127.0.0.1:6379> hget hash_ay_key  uuid_one
"12345"
#    key  uuid_three
127.0.0.1:6379> hdel hash_ay_key uuid_three
(integer) 1
127.0.0.1:6379> hkeys hash_ay_key
1) "uuid_one"
2) "uuid_two"
#              key  value
127.0.0.1:6379> hgetall hash_ay_key
1) "uuid_one"
2) "12345"
3) "uuid_two"
4) "22222"
#    key  uuid_one
127.0.0.1:6379> hset hash_ay_key uuid_one "11111"
(integer) 0
127.0.0.1:6379> hset hash_ay_key "uuid_one" "11111"
(integer) 0
127.0.0.1:6379> hgetall hash_ay_key
1) "uuid_one"
2) "11111"
3) "uuid_two"
4) "22222"
```

SortedSet

SortedSet   set

```
#sorted set      ay          1
```

```
127.0.0.1:6379> zadd zset_ay_key 1 "ay"
(integer) 1
127.0.0.1:6379> zadd zset_ay_key 2 "al"
(integer) 1
127.0.0.1:6379> zadd zset_ay_key 3 "xy"
(integer) 1
127.0.0.1:6379> zadd zset_ay_key 4 "xl"
(integer) 1
#
127.0.0.1:6379> zrange zset_ay_key 0 -1
1) "ay"
2) "al"
3) "xy"
4) "xl"
#
127.0.0.1:6379> zrem zet_ay_key "xl"
(integer) 0
127.0.0.1:6379> zrange zset_ay_key 0 -1
1) "ay"
2) "al"
3) "xy"
4) "xl"
```

```
cd c:\//                      --     C     --------------

d:  //              --     D   --------------------

dir //                        ----------------------
```

dddddddddddd

C#

```
using  System;
using   System.Net.Sockets;
using   System.Text;
//
//1                                                 +         +OK\r\n
//2:                              -           -ERR  unknown  command  'mush'\r\n
//3:
//4                 512M                                                                            $    +                     $4\r\n
//5
namespace  CC_Redis_001  {
         class   Program   {
                     static   void   Main(string[]   args)   {
                                      C_Redis        redis    =    new    C_Redis    ();
                                           redis.S_           (E_Redis    .INFO,"lpush    plist    ch0    ch1    ch2(integer)")
                                                                                                       Console.WriteLine("====
                                                     while(true)     {     }
                                                       }
              }
        public  enum  E_Redis      {
                                   GET,     //        key
                                       INFO,     //Redis
                                  SET,    //
                                   EXPIRE,     //
                                   MULTI,     //
                      EXEC,   //          MULTI
              }
        public  class  C_Redis      {
                                public    string    o_IP="192.168.1.104";
                             public    int    o_    =6379;
                              public    bool    o_        Nagle=false;
                              private    Socket    o_Socket;
                    private   byte[]   v          =   new   byte[204800];
                                  int     o_        ;
                          void    S_       ()    {


                                                                         o_Socket=new       Socket(AddressFamily.InterNetwor


                                                                    try      {






                                            }    catch(Exception    ex)    {




                                                       }
                        public    void    S_Close()    {
                                                                    try      {


                                            }    catch(Exception    ex)    {


                                                       }
                   public   void   S_           (E_Redis       z    ,params   string[]   sss)   {
                                                 //                *<number    of    arguments>\r\n
                                         const    string    z           =    "*{0}\r\n";
                                         //                             $<number    of    bytes    of    argument    N>\r\n<argum
                                          const    string    z           =    "${0}\r\n{1}\r\n";
                                           StringBuilder    sb    =    new    StringBuilder();

                                               string    cmd    =    z    .ToString();
                                                                                         Console.WriteLine("        :<<-

                                         foreach(string    ss    in    sss)    {



                                                     byte[]     data     =     Encoding.UTF8.GetBytes(sb.ToString());
                                                                    try      {





                                              }    catch(SocketException    e)    {
                                                                    }
                                  void    S_       ()   {
                                                                                                              string         ss=System.Text.En
                                                                 if(ss[0]=='-')     {     //

                                                       }     else    if(ss[0]=='+')    {


                                                                    }
                    }
   }
```

Write here...

,

5 ,

, , ,

),

)

70

,

2 5 ,

2

1v1

4 4 , 2

12

1. 2. / 3. / 4. /
5. 6. 7. 8. 9.

1. 2. 3. 4

wwwwww

Write here...

Visual---

Gestures---

Simple---
Listener---

swipe---

Completed---

wwwwww

cmd

```
netstat -ano
```

//   ------------   PID-------

//-------win10 --   -------------

```
C:\Windows\system32>netsh wlan set hostednetwork mode=allow ssid=Xia_005 key=123456789

C:\Windows\system32>netsh wlan start hostednetwork
```

Write here...

Alt+F11VBA

```
Private  Sub  Worksheet_SelectionChange(ByVal  Target  A
Dim  Rng  As  Range
Set  Rng  =  Target.Range("a1")
Cells.Interior.ColorIndex  =  0  '
'Rng.EntireColumn.Interior.ColorIndex  =  40  '
Rng.EntireRow.Interior.ColorIndex  =  38  '
End  Sub
```

//----excel   -------------------

=OR(AND(CELL("row")=ROW(),CELL("col")),AND(CELL("c

- - -

- - - - - - - -   - - - - -

- - - -
99%

QQ   - - - -

//   - - - - -   -

1   -   - --

2   --   ----

- - -   - -

-

- - -

- - - - - - - - - - - - - - - - - - - -

Write here...

2018

‘ ’ ‘ ’

20

……

06 09

2002

N

‘ ’

XX

“ ”

XX
XX

94 02

1

2

3

“

” “ ” “

” “ ”

e

.

“ ”

10 100 1000 5 1000 1 3

5

,

100

10

“

1 2 4 8 64

”

……

.

```
A&B  //-----------------  -&&---
A|B  //------------------  -----
  ~A:  //matlab
  A^B;  //matlab
  A<<B;  //----------------  A    B
  A>>B;
        A>>>B;  //java
```

```
1        void  clr(int&  a,int  i){a&=~(1<<i);}
    2,        :  int  get(int  a,int  t){return  a&(1<<
         3,                  a=a&oxFF;
      4,          2     bool   isPow2(int   a){return   (
    5,           bool  isOdd(int  a){return  a&1==1;}

1        void  set(int&  a,int  i){a|=(1<<i);}
    2,        int  opposite(int  a){return  ~a+1;}

 1       void  swap(int&  a,int&  b){a=a^b;b=a^b;a=a^b;}//
          2:          void   reverse(int&   a,int   i){a^=(1
 1  2    a<<2        //--  a   4
              2           void   rol(int&   a,int   k){
```

```
average   //     -----int   average   =(x&y)+((x^y)>>1);
abs    //     ------int  abs(int  x){int  y=x>>31;  return  (x+y)^y;}
max    //        -------int  max(int  x,int  y){return  x^((x^y)&((x-y)>>
```

Write here...