Unity Game Developer Postion Interview – Part 2

deepwaterooo

February 9, 2018

Contents

1 第一部分

- http://blog.csdn.net/qq_26270779/article/details/53609069
- 以下哪一个选项不属于 Unity 引擎所支持的视频格式文件（D）
  - A. 后缀为 mov 的文件
  - B. 后缀为 mpg 的文件
  - C. 后缀为 avi 的文件
  - D. 后缀为 swf 的文件
- Unity 引擎使用的是左手坐标系还是右手坐标系（A）
  - A. 左手坐标系
  - B. 右手坐标系
  - C. 可以通过 ProjectSetting 切换右手坐标系
  - D. 可以通过 Reference 切换左手坐标系
- 什么是导航网格（NavMesh）（B）
  - A. 一种用于描述相机轨迹的网格
  - B. 一种用于实现自动寻址的网格
  - C. 一种被优化过的物体网格
  - D. 一种用于物理碰撞的网格
- 下列选项中有关 Animator 的说法错误的是?（D）
  - A.Animator 是 Unity 引擎中内置的组件
  - B. 任何一个具有动画状态机功能的 GameObject 都需要一个 Anim 组件
  - C. 它主要用于角色行为的设置，包括 StateMachine、混合树 BlendTrees 以及同通过脚本控制的事件
  - D.Animator 同 Animation 组件的用法是相同的
- Application.loadLevel 命令为（A）
  - A. 加载关卡
  - B. 异步加载关卡
  - C. 加载动作
  - D. 加载动画

- 下列选项中，关于 Transform 组件的 Scale 参数描述正确的是（A）
  - A.Transform 组件的 Scale 参数不会影响 ParticleSystem 产生粒子的大小
  - B.Transform 组件的 Scale 参数不会影响 GUITexture 的大小
  - C. 添加 Collider 组件后的 GameoObject，其 Collider 组件的尺寸不受 Transform 组件的 Scale 参数影响
  - D. 添加 Rigidbody 组件后的物体，大小将不再受 Transform 组件中 Scale 参数的影响
- 在 Unity 中的场景中创建 Camera 时，默认情况下除了带有 Transform、Camera、GUILayer、Flare Layer 组件之外，还带有以下哪种组件（C）
  - A.Mouse Look
  - B.FPS Input Controller
  - C.Audio Listener
  - D.Character Motor
- 如果将一个声音剪辑文件从 Project 视图拖动到 Inspector 视图或者 Scene 视图中的游戏对象上，该游戏对象会自动添加以下哪种组件（C）
  - A.Audio Listener
  - B.Audio Clip
  - C.Audio Source
  - D.Audio Reverb Zone
- 下列叙述中有关 Prefab 说法错误的是哪一项（B）
  - A.Prefab 是一种资源类型
  - B.Prefab 是一种可以反复使用的游戏对象
  - C.Prefab 可以多次在场景进行实例
  - D. 当一个 Prefab 添加到场景中时，也就是创建了它的一个实例
- 关于 MonoBehaviour.LateUpdate 函数描述错误的是：（B）
  - A. 当 MonoBehaviour 类被启用后，每帧调用一次
  - B. 常被用于处理 Rigidbody 的更新
  - C. 在所有 Update 函数执行后才能被调用
  - D. 常被用于实现跟随相机效果，且目标物体的位置已经在 Update 函数中被更新
- 下列哪个函数不属于碰撞事件（C）
  - A.OnCollisionEnter
  - B.OnCollisionExit
  - C.OnCollisionUpdate
  - D.OnCollisionStay
- 以下关于 MonoBehaviour.OnGUI() 的描述错误的是（D）
  - A. 如果 MonoBehaviour 没有被启用，则 OnGUI 函数不会被调用
  - B. 用于绘制和处理 GUI events
  - C. 每帧可能会被绘制多次，每次对应于一个 GUI event
  - D. 每帧被调用一次
- 以下哪组摄像机中 Normalized View Port Rect 的数值设置可以使摄像机显示的画面位于 1280*720 分辨率的屏幕画面右上角（D）
  - A.X=640,Y=360,W=640,H=360
  - B.X=640,Y=0,W=640,H=360

  - C.X=0,Y=0,W=0.5,H=0.5
  - D.X=0.5,Y=0.5,W=0.5,H=0.5

- 在 Unity 引擎中，Collider 所指的是什么（D）
  - A.collider 是 Unity 引擎中所支持的一种资源，可用作存储网格信息
  - B.Collider 是 Unity 引擎中内置的一种组件，可用对网格进行渲染
  - C.Collider 是 Unity 引擎中所支持的一种资源，可用作游戏对象的坐标转换
  - D.Collider 是 Unity 引擎中内置的一种组件，可用作游戏对象之间的碰撞检测

- 以下关于 WWW.LoadFromCacheOrDownload 描述正确的是（C）
  - A. 可被用于将 Text Assets 自动缓存到本地磁盘
  - B. 可被用于将 Resource 自动缓存到本地磁盘
  - C. 可被用于将 Asset Bundles 自动缓存到本地磁盘
  - D. 可被用于将任意格式的 Unity 资源文件自动缓存到本地磁盘

- 如何实现加载外部视频并播放?
  - 外部视频文件：目前测试仅支持 ogg 格式（设置网络获取视频时，必须将 MIME 设置.ogg 为 application/octet-stream）
  - 场景设置：MainCamera 上添加 AudioSource 脚本；播放物体上（如 Plane）添加 MovieTest 脚本
  - MovieTest 脚本：

```
using UnityEngine;
using System.Collections;

public class MovieTest : MonoBehaviour   {
    //视频纹理
    protected MovieTexture movTexture;
    AudioClip audio;
    AudioSource AudioSource1;
    void Start()   {
        StartCoroutine(DownLoadMovie());
    }
    void OnGUI()   {
        if (GUILayout.Button(" 播放/继续"))   {
            //播放/继续播放视频
            if (!movTexture.isPlaying)   {
                movTexture.Play();
                AudioSource1.Play();
            }
        }
        if (GUILayout.Button(" 暂停播放"))   {
            //暂停播放
            movTexture.Pause();
            AudioSource1.Pause();
        }
        if (GUILayout.Button(" 停止播放"))   {
            //停止播放
            movTexture.Stop();
            AudioSource1.Stop();
        }
    }
    IEnumerator DownLoadMovie()   {
        WWW www = new WWW ("http://127.0.0.1/Wildlife.ogg");//"file://" + Application.dataPath
        yield return www;
        movTexture = www.movie;
```

```
        //获取主相机的声源
        AudioSource1 = Camera.main.GetComponent(typeof(AudioSource)) as AudioSource;
        //获取视频的声音设置到声源上
        AudioSource1.clip = movTexture.audioClip;
        audio = AudioSource1.clip;
        //设置当前对象的主纹理为电影纹理
        renderer.material.mainTexture = movTexture;
        //设置电影纹理播放模式为循环
        movTexture.loop = true;
    }
}
```

- 游戏对象 B 是游戏对象 A 的子物体，游戏对象 A 经过了旋转，请写出游戏 B 围绕自身的 Y 轴进行旋转的脚本语句，以及游戏对象 B 围绕世界坐标的 Y 轴旋转的脚本语句。
  - 绕世界坐标旋转：transform.Rotate (transform.up*speed*Time.deltatime);
  - 绕自身 Y 轴旋转：transform.Rotate (Vector.up*speed*Time.deltatime);
- Unity 中用过哪些插件？具体功能
  - FXMaker，制作粒子特效；NGUI，制作 2D 界面；EasyTouch，摇杆；shaderForge，制作 shader；Itween，制作动画；
- 当删除 Unity 工程 Assets 目录下地 meta 文件时会导致什么？为什么？
  - 会导致在场景中游戏对象看不到，或者报错，材质找不到资源。多人协作的时候会导致资源的重复产生。因为每个资源文件都对应一个.meta 文件，这个.meta 文件中的 guid 就是唯一标识这个资源的。材质就是通过这个 guid 来记录自己使用了那些资源，而且同一个资源的 guid 会因为不同的电脑而不同，所以当你上传了丢失了.meta 文件的资源的时候，到了别人的机器上就会重新产生 guid，那个这个资源就相当于垃圾了。
- 频繁创建 GameObject 会降低程序性能为什么？怎么解决?
  - 频繁创建游戏对象，会增加游戏的 Drawcall 数，降低帧率，GPU 会一直在渲染绘制。可以通过对象池来管理对象：当需要创建一个游戏对象时，先去对象池中查找一下对象池中是否存在没有被正在使用的对象，如果有的话直接使用这个对象，并把它标记为正在使用，没有话就创建一个，并把它添加到池中，然后标记为使用中。一个游戏对象使用完毕的时候，不要销毁掉，把它放在池中，标记为未使用。
- 关于 Vector3 的 API，以下说法正确的是（BC）
  - A.Vector3.normalize 可以获取一个三维向量的法线向量
  - B.Vector3.magnitude 可以获取一个三维向量的长度
  - C.Vector3.forward 与 Vector3(0,0,1) 是一样的意思
  - D.Vector3.Dot(向量 A, 向量 B) 是用来计算向量 A 与向量 B 的叉乘
- 以下哪个函数在游戏进入新场景后会被马上调用（B）
  - A.MonoBehaviour.OnSceneWastLoaded()
  - B.MonoBehaviour.OnSceneEnter()
  - C.MonoBehaviour.OnLevelEnter()
  - D.MonoBehaviour.OnLevelWastLoaded()
- 采用 Input.mousePosition 来获取鼠标在屏幕上的位置，以下表达正确的是（C）
  - A. 左上角为原点（0，0），右下角为（Screen.Width, Screen.Height）
  - B. 左下角为原点（0，0），右下角为（Screen.Height, Screen.Width）
  - C. 左下角为原点（0，0），右上角为（Screen.Width, Screen.Height）
  - D. 左上角为原点（0，0），右下角为（Screen.Height, Screen.Width）
- 如何通过脚本来删除其自身对应的 Gameobject（A）

  - A.Destroy(gameObject)
  - B.this.Destroy()
  - C.Destroy(this)
  - D. 其他三项都可以

- 某个 GameObject 有一个名为 MyScript 的脚本，该脚本中有一个名为 DoSomething 的函数，则如何在该 Gameobject 的另外一个脚本中调用该函数?（A）
  - A.GetComponent().DoSomething()
  - B.GetComponent
  - C.GetComponent().Call("DoSomething")
  - D.GetComponent

- Animator.CrossFade 命令作用是:（B）
  - A. 动画放大
  - B. 动画转换
  - C.Update()
  - D.OnMouseButton()

- OnEnable,Awake,Start 运行时的发生顺序?（A）
  - A.Awake->OnEnable->Start
  - B.Awake->Start->OnEable
  - C.OnEnable-Awake->Start
  - D.Start->OnEnable->Awake

- 以下选项中，正确的是（D）
  - A.Mathf.Round 方法作用是限制
  - B.Mathf.Clamp 方法作用是插值
  - C.Mathf.Lerp 方法作用是四舍五入
  - D.Mathf.Abs 方法作用是取得绝对值

- 以下选项中，将游戏对象绕 Z 轴逆时针旋转 90 度（C）
  - A.transform.rotation = Quaternion.Euler(0,0,90)
  - B.transform.rotation = Quaternion.Angle(0,0,90)
  - C.transform.Rotate(new Vector3(0,0,90))
  - D.transform.Rotate(new Vector3(90,0,0))

- public static function InitializeServer(connections:int,listenPort:int,useNat:bool):NetworkConnectionError; 解释一下函数，参数以及返回值的意思。
  - 初始化服务器。connections 是允许的入站连接或玩家的数量，listenPort 是要监听的端口，useNat 设置 NAT 穿透功能。如果你想要这个服务器能够接受连接使用 NAT 穿透，使用 facilitator，设置这个为 true。如果有错误会有返回错误。

- 请写出以下函数的含义和运算结果

```
delegate b Func<a, b>(a a1);
static void Main(string[] args) {
    Func<int, bool> mFunc = x => x == 5;
    Console.WriteLine(mFunc(6));
}
```

  - false，就是定义一个 delegate，返回值类型为 b，有一个参数，类型为 a。

- 编写一个函数，输入一个 32 位整数，计算这个整数有多少个 bit 为 1.

```
uint BitCount (uint n) {
    uint c = 0; // 计数器
    while (n > 0) {
        if ((n & 1) == 1) // 当前位是 1
            ++c; // 计数器加 1
        n >>= 1; // 移位
    }
    return c;
}
```

- 某游戏中的装备系统有 16 种附加属性，每种附加属性使用一个 32 位的 ID 表示 (比如 10001 表示加人物 hp 的附加属性，10002 表示加人物 mp 的附加属性)，一件装备做多有 4 个附加属性，请写一个程序输出所有附加属性的组合。

- 请实现如下函数，在 Unity 中有一副骨骼树，请使用递归方式与非递归方式实现先序遍历，在 Unity 的 Console 输出所有骨骼名。
  - 可能用到的函数

```
public Transform GetChild (int index);
public int Transform.childCount
public void OutputTree(Transform root) {}
```

- 简要解释下数据库中 ACID 的含义。
  - ACID 是指在可靠数据库管理系统（DBMS）中，事务所具有的四个特性：原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）。
  - 原子性是指事务是一个不可分割的工作单位，事务中的操作要么都发生，要么都不发生。
  - 一致性是指在事务开始之前和事务结束以后，数据库的完整性约束没有被破坏。这是说数据库事务不能破坏关系数据的完整性以及业务逻辑上的一致性。例如：对银行转帐事务，不管事务成功还是失败，应该保证事务结束后 ACCOUNT 表中 aaa 和 bbb 的存款总额为 2000 元。
  - 隔离性多个事务并发访问时，事务之间是隔离的，一个事务不应该影响其它事务运行效果。这指的是在并发环境中，当不同的事务同时操纵相同的数据时，每个事务都有各自的完整数据空间。由并发事务所做的修改必须与任何其他并发事务所做的修改隔离。事务查看数据更新时，数据所处的状态要么是另一事务修改它之前的状态，要么是另一事务修改它之后的状态，事务不会查看到中间状态的数据。
  - 持久性，意味着在事务完成以后，该事务所对数据库所作的更改便持久的保存在数据库之中，并不会被回滚。即使出现了任何事故比如断电等，事务一旦提交，则持久化保存在数据库中。

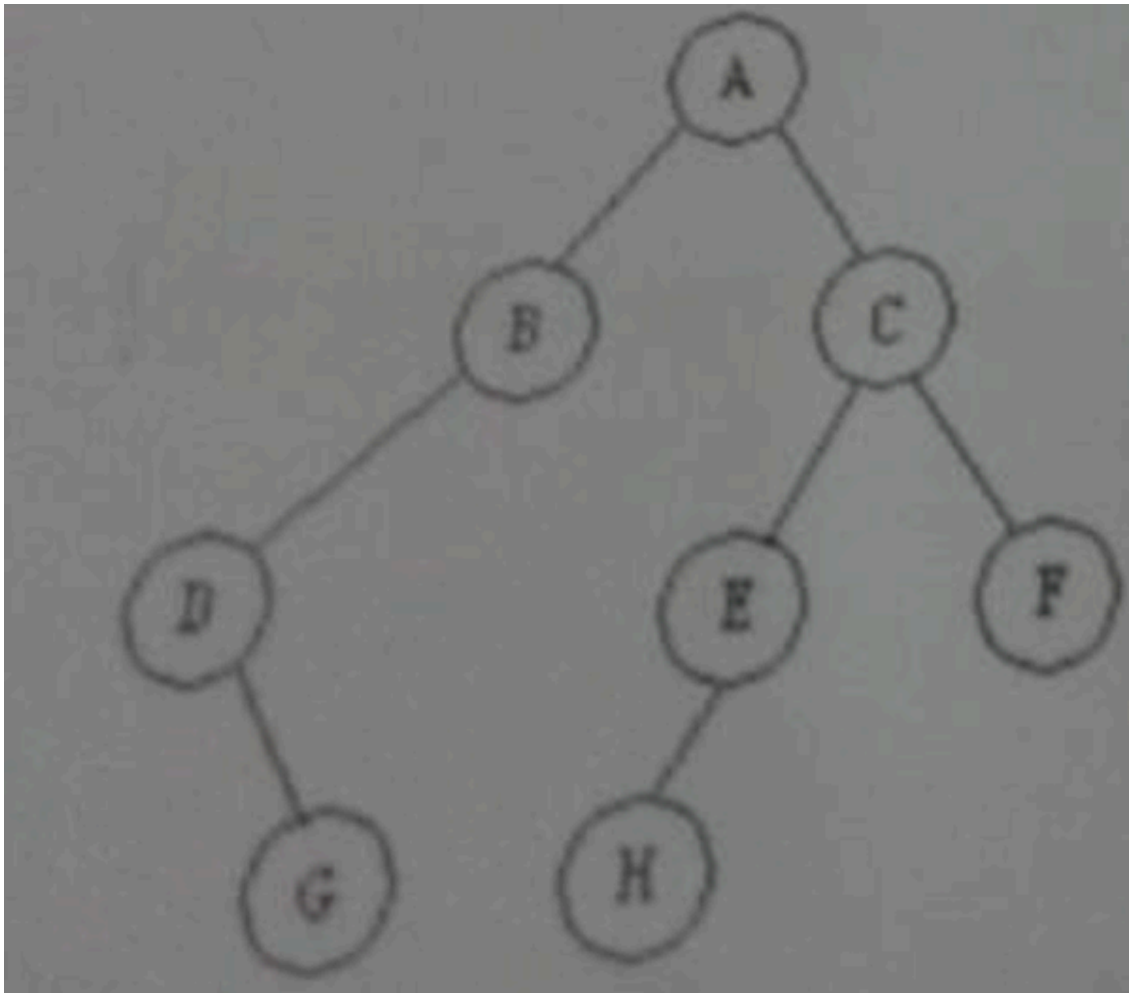
- 32 位整数 256 和 255 按位异或后的结果是（511）

- unix/linux 系统将所有的 I/O 设备模型化为文件,c 语言中的（stdin）,(stdout) 和 (stderr) 分别表示标准输入，标准输出，标准错误。

- k 层二叉树最多有 2^k - 1 个结点。

- 中缀算式 (8 + x*y)-2x/3 的后缀算式是

- 对以下二叉树进行前序遍历的结果是（ABDGCEHF）



- 写出一个 C 类 ip 地址（192.168.1.105），该 ip 地址一个合法的网格掩码是（255.255.255.224）.
- c/c++ 程序运行时有堆内存与栈内存之分，请写一个语句在堆中分配一个整数：(int a = new int(4)), 在栈内存中分配一个整数：(int a = 5)。
- 简述从 C/C++ 源代码生成可执行文件所经历的几个过程和每个过程中所做的事情。

预编译-» 编译-» 链接-» 执行

- 简述 TCP 建立连接的过程，最好能画出时序图。
  - 三次握手；
- 有一个 c 语言函数

```
unsigned int getN(unsigned int v){
    v--;
    v |= v >> 1;
    v |= v >> 2;
    v |= v >> 4;
    v |= v >> 8;
    v |= v >> 16;
    v++;
    return v;
}
```

请问这个函数的功能是什么?

- 返回的结果是 v 或者 2 ^ 跟下 v 次方 + 1（比如 125 返回 128，128 返回的就是 128）
- 列出 c 语言中 static 关键字的用法。

  - static 关键字至少有下列几个作用：
    * （1）函数体内 static 变量的作用范围为该函数体，不同于 auto 变量，该变量的内存只被分配一次，因此其值在下次调用时仍维持上次的值；
    * （2）在模块内的 static 全局变量可以被模块内所用函数访问，但不能被模块外其它函数访问；
    * （3）在模块内的 static 函数只可被这一模块内的其它函数调用，这个函数的使用范围被限制在声明它的模块内；
    * （4）在类中的 static 成员变量属于整个类所拥有，对类的所有对象只有一份拷贝；
    * （5）在类中的 static 成员函数属于整个类所拥有，这个函数不接收 this 指针，因而只能访问类的 static 成员变量。

- 解释一下 Unity3D 中的协程（coroutine）是什么？并举例说明
  - 在主线程运行的同时开启另一段逻辑处理，来协助当前程序的执行，协程很像多线程，但是不是多线程，Unity 的协程实在每帧结束之后去检测 yield 的条件是否满足。

- 物体自旋转使用的函数叫什么？物体绕某点旋转使用函数叫什么？
  - 物体自旋转函数 transform.Rotate，物体绕某点旋转函数：transform.RotateAround

- 使用 prefab 预制物体有什么好处？
  - 1.Prefab 一个重要的优势就是编辑 prefab 资源后，场景中所有使用 Prefab 克隆的游戏对象将全部使用新编辑的资源，无需一个一个的给场景中的对象赋值。
  - 2. 当游戏中需要频繁创建一个物体时，使用 Prefab 能够节省内存。
  - 3. 当你在一个场景中增加一个 Prefabs，你就实例化了一个 Prefabs。

- 设有如下关系表 R1 和 R2
  - R1(NO, NAME, SEX, AGE, CLASS)
  - R2(NO, SUBJECT, SCORE)
  - 主关键字是 NO, 其中 NO 为学号，NAME 为姓名，SEX 为性别，AGE 为年龄，CLASS 为班号，SUBJECT 为科目，SCORE 为成绩。写出实现下列功能的 SQL 语句。查找学号为 20 的学生的姓名，科目，成绩。
  - SELECT NAME,SUBJECT,SCORE FORM R1 INNER JOIN R2 ON R1.NO=R2.NO WHERE R1.NO = 20

- 描述多人联网中位置的同步和聊天通讯的实现方法，并编程写出如何实现多人协同中交互操作。（交互操作例如，血值为 100 的两个角色人物可以相互射击双方，射击一次减去血值 10，当血值为 0 的时候证明已经死亡）

- 当实例化一个 prefab 对象，prefab 对象上面挂载了一个继承 MonoBehaviour 的脚本
  - 1. 请问这个脚本分别会按顺序调用哪些函数，并列出哪些函数是反复进去的
  - 2. 请问当这个实例化对象先调用了 SetActive(false)，然后又调用了 SetActive(true), 那么这个脚本依次会调用哪些函数方法?
  - 3. 如果在脚本的 Awake () 函数中，调用了 this.gameObject.AddComponent()(PS:GameController 为另一个继承 Mono 的脚本类)，请问此时脚本函数的调用顺序是否发生变化?

- if(go.CompareTag(“Enemy”)) 和 if（go.tag == “Enemy”）两种判断方法哪种是合适的，为什么?
  - 第一种合适，因为第二种会占用更多内存。

- DestroyImmediate 和 Destroy 的区别是?
  - DestroyImmeditate 销毁对象的时候，会立即释放资源。Destroy 只是从该场景销毁，但是还在内存当中。

- 详细说明 Unity 中资源加载的方法，以及他们的区别?
  - 1. 通过 Resources 模块，调用它的 load 函数：可以直接 load 并返回某个类型的 Object，前提是要把这个资源放在 Resource 命名的文件夹下，Unity 不关有没有场景引用，都会将其全部打入到安装包中。

  - 2. 通过 bundle 的形式：即将资源打成 asset bundle 放在服务器或本地磁盘，然后使用 WWW 模块 get 下来，然后从这个 bundle 中 load 某个 object。

- 从代码角度上如何进行内存优化?
- 分别写出 Invoke 和协程的几种用法?
- 以下函数的功能是计算斐波那契数列的第 n 项，请填空

```
int func(int n) {
    if(n<=2) return 1;
    return n + func(n-1);
}
```

- C 语言中宏定义中若有多行，可以使用字符（\）.
- C 语言中 32 位整数能表达的最小的数是-2^{31}~2^{31}-1.
- 使用（点乘）向量运算可以很方便地计算出三维空间中两个单位向量之间夹角的 cos 值。
- 类 unix 系统中某文件的权限为：drw-r–r–，用八进制数值形式表示该权限为 (411), 首位 d 代表目录（文件夹）
- 若有如下程序段，其中 s，a，b，c 均已定义为整型变量，且 a，c 均已赋值（c 大于 0）

```
s = a;
for(b = 1; b <= c; b++)
    s = s + 1;
```

则与上述程序段功能等价的赋值语句是（B）

- A.s = a + b
- B.s = a + c
- C.s = s + c
- D.s = b + c
- 简述 static 和 const 关键字的作用
  - static 关键字至少有下列几个作用：
    * （1）函数体内 static 变量的作用范围为该函数体，不同于 auto 变量，该变量的内存只被分配一次，因此其值在下次调用时仍维持上次的值；
    * （2）在模块内的 static 全局变量可以被模块内所用函数访问，但不能被模块外其它函数访问；
    * （3）在模块内的 static 函数只可被这一模块内的其它函数调用，这个函数的使用范围被限制在声明它的模块内；
    * （4）在类中的 static 成员变量属于整个类所拥有，对类的所有对象只有一份拷贝；
    * （5）在类中的 static 成员函数属于整个类所拥有，这个函数不接收 this 指针，因而只能访问类的 static 成员变量。
  - const 关键字至少有下列几个作用：
    * （1）欲阻止一个变量被改变，可以使用 const 关键字。在定义该 const 变量时，通常需要对它进行初始化，因为以后就没有机会再去改变它了；
    * （2）对指针来说，可以指定指针本身为 const，也可以指定指针所指的数据为 const，或二者同时指定为 const；
    * （3）在一个函数声明中，const 可以修饰形参，表明它是一个输入参数，在函数内部不能改变其值；
    * （4）对于类的成员函数，若指定其为 const 类型，则表明其是一个常函数，不能修改类的成员变量；
    * （5）对于类的成员函数，有时候必须指定其返回值为 const 类型，以使得其返回值不为“左值”。
- 用你熟悉的语言及你认为最简洁的方法书写计算 s = 1!+2!+3!+…+num! 的代码。num 为输入，s 为输出。(! 代表阶乘 3! = 1 * 2 * 3

```
Console.ReadLine(num)
int s = 0;
for(int i = 1; i <= num; i++) {
    s += JieCheng(num);
}
public int JieCheng(int num) {
    if(num < 0) {
        Console.WriteLine("error");
        return;
    }
    if(num <=1) {
        return 1;
    } else {
        return num * JieCheng(num - 1)
            }
}
```

- 用你熟悉的语言从一个字符串中去掉相连的重复字符，例如原字符串“adffjkljaalkjhl”变为“adfjkljalkjhl”

```
int GetResult(char[] input, char[] output)   {
    int i, j, k = 0;
    int flag;
    int length;
    if(input == NULL || output == NULL)   {
        return -1;
    }
    length=strlen(input);//求数组的长度
    for(i = 0; i<length; i++)   {
        flag = 1;
        for(j = 0; j < i; j++)   {
            if(output[j] == input [i])
                flag = 0;
        }
        if(flag)
            output[k++] = input[i];
    }
    printf(" 最终的字符串为: ");
    output[k] = '\0';
    for(int m = 0; m < output.Length; m++) {
        print (output [m]);
    }
    return 0;
}
```

2 第二部分

- 哪种实时光源是 Unity 中没有的？（D）
  - A. 点光源
  - B. 方向光
  - C. 聚光灯
  - D. 日光灯
- 如何在 Unity 中创建地形系统？(D)
  - A.Terrain->Create Terrain
  - B.Component->Create Terrain
  - C.Asset->Create Terrain

  - D.Windows->Create Terrain
- 以下哪种操作步骤可以在场景中添加“Wind Zone”?(B)
  - A.Terrain->Wind Zone
  - B.GameObject->Create other->Wind Zone
  - C.Component->Physics->Wind Zone
  - D.Assets->Create->Wind Zone
- 在 Unity 编辑器中创建一个 Directional Light，以下步骤正确的是?（B)
  - A.Edit->Rendering Setting->Directional Light
  - B.GameObject->Create Other->Directional Light
  - C.Component->Rendering->Directional Light
  - D.Assets->Directional Light
- 下列哪一项不属于 Camera 中的“Clear Flags”?（D）
  - A.Skybox
  - B.Solid Color
  - C.Depth Only
  - D.Background
- 以下哪种脚本语言是 Unity 编辑器所不支持的?（D）
  - A.Javascript
  - B.C#
  - C.Boo
  - D.Perl
- 对于 Prefab，以下说法错误的是?（D）
  - A.Prefab 资源可以在项目中多次重复使用
  - B. 由 Prefab 实例出的 GameObject，其在 Hierarchy 试图中表现为蓝色
  - C.Prefab 上的组件信息一经改变，其实例出的 GameObject 也会自动改变
  - D. 实例出的 GameObject 上的组件信息一经改变，其对应出的 Prefab 也会自动改变
- 下面哪种做法可以打开 Unity 的 Asset Store?（A）
  - A.Windows->Asset Store
  - B.Edit->Asset Store
  - C.File->Asset Store
  - D.Assets->Asset Store
- 在哪个面板中可以修改物体的空间属性，如位置，朝向，大小等（B）
  - A.Project
  - B.Inspector
  - C.Hierarchy
  - D.Toolbar
- 如何为一个 Asset 资源设定一个 Label，从而能够方便准确的搜索到?（D）
  - A. 在 Project 窗口中选中一个 Asset，右键->Create->Label
  - B. 在 Project 窗口中选中一个 Asset，右键->Add Label
  - C. 在 Project 窗口中选中一个 Asset，在 Inspector 窗口中点击添加 Label 的图标
  - D. 在 Project 窗口中选中一个 Asset，在 Inspector 窗口中点击按钮“Add Label”

- Mecanim 系统中，Body Mask 的作用是?（D）
  - A. 指定身体的某一部分是否参与骨骼动画
  - B. 指定身体的某一部分是否参与物理模拟
  - C. 指定身体的某一部分是否可以输出骨骼信息
  - D. 指定身体的某一部分是否参与渲染
- 以下哪种操作步骤可以打开 Unity 编辑器的 Lightmapping 视图?（C）
  - A.File->Lightmapping
  - B.Assets->Lightmapping
  - C.Windows->Lightmapping
  - D.Component->Lightmapping
- 下列关于光照贴图，说法错误的是?（C）
  - A. 使用光照贴图比使用实时光源渲染要快
  - B. 可以降低游戏内存消耗
  - C. 可以增加场景真实感
  - D. 多个物体可以使用同一张光照贴图
- 如何为物体添加光照贴图所使用的 UV?(B)
  - A. 不用添加，任何时候都会自动生成
  - B. 更改物体导入设置，勾选“Generate Lighting UVs”
  - C. 更改物体导入设置，勾选“Swap UVs”
  - D. 更改物体导入设置，在 UVs 选项中选择“Use Lightmaps”
- 在哪个模块下可以修改 Render Path?（A）
  - A.Camera
  - B.Light
  - C.Render Settings
  - D.Project Setting->Quality
- 以下哪项技术下不是目前 Unity 所支持的 Occlusion Culling 技术?（D）
  - A.PVS only
  - B.PVS and dynamic objets
  - C.Automatic Portal Generation
  - D.Dynamic Only
- 关于 Vector3 的 API，以下说法正确的是?（C）
  - A.Vector3.normalize 可以获取一个三维向量的法线向量
  - B.Vector3.magnitude 可以获取一个三维向量的长度
  - C.Vector3.forward 与 Vector3（0，0，1）是一样的
  - D.Vector3.Dot(向量 A，向量 B) 是用来计算向量 A 与向量 B 的叉乘
- 下列那些选项不是网格层属性的固有选项?（B）
  - A.Default
  - B.Walkable
  - C.Not Walkable
  - D.Jump
- 写出你对游戏的理解及游戏在生活中的作用，对 Unity3D 软件理解最深入的地方。

3 C++

- 请写代码打印 100 之内的素数，讲求效率（请做你的解法的效率分析）
- 求 m,n 的最大公约数
- 输入 10 个字符串，打印出其中重复的字符串以及重复的次数
- 请画图例（UML 最好），给出 windows 下的文件目录的设计模式
- 用 OO 表示狼吃羊羊吃草
- 什么是 subversion? 它与 vss,cvs 的区别在哪? 或者有什么优势?
- 什么是 wiki，关于程序项目的 wiki 你使用过哪些?wiki 对你有什么帮助吗?wiki 与程序文挡的差别在哪?
- 什么是 tdd? 你使用过吗?tdd 的关键在哪? 跟传统的单元测试相比，有什么优越性?
- 什么是单元测试? 你用过哪些单元测试工具? 他们的区别和好处各有哪些? 你主要倾向于哪一种?
- 什么是编程规范? 你倾向于什么样的规范? 他的好处在哪?
- 什么是 mfc? 你经常使用那些 mfc 类? 那么为什么很多人不主张使用 mfc?
- 什么是头文件依赖? 你注意过这些问题吗? 你注意过编译的时间吗? 你怎么改进编译时间?
- 什么是面向对象? 你在哪些方面用过面向对象? 带来了什么好处? 又有什么弊端?
- 什么是接口编程.com，他带来了什么好处? 适用于什么地方?
- 什么是设计模式? 使用设计模式有什么好处? 列举你使用过的设计模式知识:
- 一寸山河一寸血，_____ _____
- 抗战历时 _____ _____ _____