游戏开发课堂测验 19.04.12

1. 计算机内部使用（ A ）存储数据和进行运算，最小的单位是比特（bit）

A. 二进制 B.八进制 C.十进制 D.十六进制

1. 十进制数是逢十进一，使用十进制表示数量，需要用10种不同的符号。

二进制数是逢二进一，使用二进制表示数量，需要用（ B ）种不同的符号

A. 1 B. 2 C. 10 D. 16

1. 一位二进制数（1比特）的可能取值范围是（ B ）

A. 0到9 B. 0到1 C. 0到2 D. 0到15

1. 十进制的2，用二进制数表示为（ C ）

A. 0 B. 1 C. 10 D. 11

1. 十进制的4，用二进制数表示为（ D ）

A. 0 B. 1 C. 10 D. 100

1. 二进制的111，相当于十进制的（ C ）

A. 5 B. 6 C. 7 D. 8

1. 已知n位二进制数可以表示“2的n次方”种不同状态，那么8位二进制数（8比特）可以表示（ C ）种状态

A. 64 B. 128 C. 256 D. 512

1. 字节（byte）是数据计量单位，1字节等于8比特，用一个字节来存放数字，这个数字的取值范围可以是（ A ）

A. 0到255 B. 0到511 C. 0到1023 D. 0到正无穷

1. 整型（int）数值每个在内存中占用4个字节，相当于（ C ）比特。

A. 8 B. 16 C. 32 D. 64

1. 当声明一个整型（int）类型变量时，系统会为这个变量分配（ C ）个字节的内存空间。

A. 1 B. 2 C. 4 D. 8

1. 当声明一个可以存放50个元素的整型数组时，系统会为这个数组分配（ D ）个字节的内存空间。

A. 4 B. 32 C. 50 D. 200

1. 换行字符 '\n' 可以控制文本在何处换行显示，请选出以下显示效果所对应的字符串（ D ）

CN-171-11

领航员

苏拉威西

A. CN-171-11领航员苏拉威西

B. CN-171-11领航员\n苏拉威西

C. CN-171-11\n\n领航员\n\n苏拉威西

D. CN-171-11\n领航员\n苏拉威西

1. 以下哪条语句声明了一个布尔变量？（ D ）

A. string myStr;

B. int myNum;

C. const int MYCONST = 5;

D. bool myBool;

1. 以下哪条语句调用了一个方法？（ B ）

A. string msg;

B. msg = Console.ReadLine();

C. msg = "Hello";

D. return msg;

15. 刚体碰撞体是指一个游戏对象（ A ）

A. 有刚体，有碰撞体

B. 有刚体，无碰撞体

C. 无刚体，有碰撞体

D. 无刚体，无碰撞体

16. 运动学刚体对象只能通过改变位置来移动，它对应的刚体组件选项是（ B ）

A. Is Kinematic = false

B. Is Kinematic = true

C. Use Gravity = false

D. Use Gravity = true

17. （ C ）的游戏对象称为静态碰撞体对象，游戏运行过程中移动这类对象会造成较大的性能开销，应尽量避免。

A. 有刚体，有碰撞体

B. 有刚体，无碰撞体

C. 无刚体，有碰撞体

D. 无刚体，无碰撞体

18. 有两个碰撞体对象，以下的哪种组合之间不能发生碰撞（ D ）

A. 刚体碰撞体，刚体碰撞体

B. 运动学刚体碰撞体，刚体碰撞体

C. 刚体碰撞体，静态碰撞体

D. 运动学刚体碰撞体，静态碰撞体

19. 以下的哪种组合之间能够发生实体碰撞（ D ）

A. 刚体碰撞体，静态触发碰撞体

B. 刚体触发碰撞体，静态碰撞体

C. 运动学刚体碰撞体，静态碰撞体

D. 刚体碰撞体，静态碰撞体

20. Unity脚本的（ D ）方法可以用于在场景视图中描绘调试图形

A. Start B. Update C. OnTriggerEnter D. OnDrawGizmos

21. 二维碰撞体之间如果发生了实体碰撞，会触发以下的哪个方法（ C ）

A. OnCollisionEnter

B. OnTriggerEnter

C. OnCollisionEnter2D

D. OnTriggerEnter2D

22.C#的文件读写功能的所在命名空间是（ B ）

1. System B. System.IO C. UnityEngine D. File
2. File类的哪个方法可以读取文件的全部内容，并以字符串数组的形式返回（ B ）
3. ReadAllBytes B. RealAllLines C. ReadAllText D. Read
4. 使用FileStream对象访问文件后，需要调用（ D ）方法来关闭文件
5. Read B. ReadByte C. Write D. Close

25. Unity脚本中加载资源文件的方法是（ B ）

1. Resource.Load()
2. Resources.Load()
3. GameObject.Instantiate()
4. File.ReadAllLines()

26. Unity游戏场景中存在的游戏对象的类型名称是（ A ）

A. GameObject B. Monobehaviour C. Component D. Prefab

1. Unity游戏对象的位置、旋转、缩放信息都存放在（ B ）组件中
2. GameObject B. Transform C. Monobehaviour D. Rigidbody

28. Unity中表示三维向量的类型是（ D ）

A. vector3 B. Vector C. Vector2 D. Vector3

29. 向量是有方向，有长度的线段，从点 A（x1,y1）指向点 B（x2,y2）的向量 AB 可以表示为（x2-x1,y2-y1），由上推断，从点 B 指向点 A 的向量可以表示为（ C ）

A. x1-x2,y1-y2 B. x1+x2,y1+y2 C. y2-y1,x2-x1 D. x2-y2,x1-y1

30. 以下选项中，与Unity项目的Assets文件夹对应的是（ B ）

A. Application.streamingAssetsPath

B. Application.dataPath

C. Application.persistentDataPath

D. Application.temporaryCachePath

31. 如果希望Unity项目中的某个文件在导出可执行文件后仍能使用System.IO文件类访问，应该将这个文件存放在（ D ）文件夹内。

A. Assets

B. Assets\Resources

C. Assets\StreamingAssets

D. C:\

32. 有一个Unity项目，文件夹路径为 D:\HelloProject，则该项目的Application.streamingAssetsPath对应的路径是（ B ）

A. D:\HelloProject\StreamingAssets

B. D:\HelloProject\Assets\StreamingAssets

C. D:\StreamingAssets

D. D:\HelloProject\Resources\StreamingAssets

33. 已知三维坐标系中两个点 A 和 B，如果要求出两点连线 AB 上某一位置的坐标，在Unity中可以通过线性插值方法（ D ）来计算

A. Vector3.Distance()

B. Mathf.Max()

C. Mathf.Lerp()

D. Vector3.Lerp()

34. 已知数轴上的点 A = 2，点 B = 8，那么线段 AB 的中点 C = （ C ）

A. 3 B. 4 C. 5 D. 6

35. 线性插值运算中，使用线段的起点 A，终点 B，以及插值系数 t 可以求出线段 AB 上任一点 P 的值，计算公式为（ D ）

A. P = A + B \* t

B. P = A - B \* t

C. P = (B - A) \* t

D. P = A + (B - A) \* t

36. 线性插值运算中，对于任意的线段 AB，插值系数 t = 0，t = 0.5，t = 1 分别对应线段上的（ C ）

A. 点 A，点 B，中点

B. 点 B，点 A，中点

C. 点 A，中点，点 B

D. 中点，点 A，点 B

37. 在Unity场景中有两个独立的游戏对象，名称分别是 objA 和 objB，（ A ）身上有一个脚本，其中的（ A ）语句可以将 objB 设置为 objA 的子对象

A. objA, GameObject.Find("objB").transform.SetParent(transform);

B. objB, GameObject.Find("objA").transform.parent = transform;

C. objA, gameObject.Find("objB").transform.SetParent(this);

D. objB, transform.Find("objA").parent = transform;

38. 三维向量 v(1,1,0) 所指方向与 X 轴的夹角是（ B ）度

A. 30 B. 45 C. 60 D. 90

39. 长度为 1 的向量叫做单位向量，求出和某个向量方向相同的单位向量的运算叫做（ C ）

A. 求模 B. 泛化 C. 归一化 D. 点乘

40. 向量(1,1,0)的长度（ A ）它的归一化向量

A. 大于 B. 等于 C. 小于 D. 小于等于

41. 一个Unity脚本的 Update 方法中有以下语句：

float f = Mathf.Sin(2 \* Mathf.PI \* Time.time) \* 5 + 5; 关于变量 f 的描述，错误的是（ D ）

A. 它的值每隔一秒重复一次

B. 它的最大值是 10

C. 它的最小值是 0

D. 它的最初的值是 2

42. 一个游戏对象从原点出发，沿着(1,-1,0)方向移动，速度大小是 10m/s，以下说法正确的是（ C ）

A. 它在坐标系的右上方

B. 它在 X 方向的速度大小等于 10m/s

C. 它在 Y 方向的速度大小小于 10m/s

D. 经过 1 秒钟后，它的坐标将变成(10,-10,0)

43. 三维向量 v = Vector3.left + Vector3.up,它的值是（ C ）

A. (1,1,0) B. (-1,-1,0) C. (-1,1,0) D. (1,-1,0)

44. 在Unity脚本中通过transform变量可以查询到游戏对象局部坐标系的XYZ轴的正方向，以下属性哪一个是不存在的（ B ）

A. transform.right

B. transform.left

C. transform.up

D. transform.forward

45. 在Unity场景中有两个游戏对象 objA 和 objB，其中 objB 是 objA 的子对象，objA 的位置是 (3,4,5)，objB 相对于 objA 的位置是 (-1,4,0)，那么（ A ）

A. objB 的世界坐标是 (2,8,5)

B. objB 的世界坐标是 (-1,4,0)

C. objB.transform.position 的值是(-1,4,0)

D. objB.transform.localPosition 的值是(2,8,5)

46. 垂直于某个平面的单位向量叫做这个平面的（ B ）

A. 切线 B. 法线 C. 中线 D. 割线

47. 要使一个游戏对象的旋转角度变为(0,0,90)，以下写法正确的是（ A ）

A. transform.rotation = Quaternion.Euler(0,0,90);

B. transform.rotation = new Quaternion(0,0,90,0);

C. transform.rotation = new Vector3(0,0,90);

D. transform.rotation.z = 90;

48. 语句RaycastHit2D hit = Physics2D.Raycast(transform.position, transform.up, 1f) 的功能是（ A ）

A. 从游戏对象的位置往局部坐标的正上方作一条射线，射线的长度是 1，将射线检测的结果保存在 hit 变量中

B. 从游戏对象的位置往局部坐标的正上方作一条射线，将射线接触的碰撞体保存在 hit 变量中

C. 从游戏对象的位置往世界坐标的正上方作一条射线，射线的长度是 1，将射线检测的结果保存在 hit 变量中

D. 从游戏对象的位置往世界坐标的正上方作一条射线，射线长度无限大，将射线检测的结果保存在 hit 变量中

49. Unity中用于设置不同动画片段之间的迁移条件的组件是（ B ）

A. Animation B. Animator C. Transform D. Sprite Renderer

50. 有一个指向 Animator 组件的变量 anim，以下语句中，可以通过这个变量将动画状态机中的一个布尔变量 walk 设为 true的是（ D ）

A. Animator.SetBool(walk,true);

B. Animator.SetBool("walk",true);

C. anim.SetBool(walk,true);

D. anim.SetBool("walk",true);

51. 以下四条语句的顺序被打乱了，它们的其中一种正确顺序是（ C D A B ）

A. go.AddComponent<SpriteRenderer>();

B. go.GetComponent<SpriteRenderer>().sprite = sprite;

C. GameObject go = new GameObject("ship");

D. Sprite sprite = Resources.Load<Sprite>("ShipSprite");

1. 根据以下代码回答问题

public class Character

{

int hp;

int atk;

Equipment equipment;

public Character(int hp, int atk)

{

// ...

}

public void Hurt(int amount)

{

// ...

}

public void Attack(Character target)

{

// ...

}

public void Equip(Equipment equipment)

{

// ...

}

public int GetDefense()

{

// ...

}

}

以上代码中，Character是一个 类

hp和atk都是它的 属性

Hurt和Attack都是它的 方法

equipment的数据类型是 Equipment

public Character(int hp, int atk) 是它的 构造方法 ，用于创建Character的 实例

上机操作：

1. 创建一个2D的Unity项目，名称是”姓名拼音\_Test\_190411”（例：LiSan\_Test\_190411）
2. 在上述项目中新建一个名称为UnityTest的场景
3. 在UnityTest场景中新建一个空白游戏对象，改名为Test
4. 将Test对象的位置设为 (0,0,0)
5. 在Assets文件夹下新建一个文件夹，命名为Scripts
6. 新建一个Unity脚本，命名为HelloTest，存放到Scripts文件夹下
7. 将HelloTest脚本添加到Test游戏对象上
8. 在HelloTest脚本中完成 9 - 17 题的上机操作
9. 声明一个公共整型变量myInt，在检视面板将其值设为255
10. 声明一个私有字符串变量myStr，在Start方法中将其值设为”Hello Unity”
11. 声明一个公共Vector3变量myVec，在检视面板将其值设为(30,40,50)
12. 在OnDrawGizmos方法中用半径0.5的红色小球标出myVec的位置
13. 在OnDrawGizmos方法中用黄色线段连接myVec和Test对象的位置
14. 创建一个私有整型数组myArray，长度为100
15. 在Start方法中将1,2,3, ... 100依次存入myArray数组
16. 实现一个私有方法Sum1to100，无参数，整型返回值，求myArray所有元素相加的和。
17. 在Update方法中检测键盘S键，每次按下S键调用一次Sum1to100方法，用Debug.Log显示求和结果。
18. 新建一个名称为Gradius的场景，在此场景完成18 - 24题的上机操作
19. 在Assets文件夹下新建一个文件夹，命名为Sprites
20. 将宇宙巡航机案例中使用的图片Gradius Sprite.png导入项目，保存到Sprites文件夹下
21. 使用Unity的自动切割Sprite功能来切割Gradius Sprite.png
22. 利用切割好的Sprite来创建玩家控制的飞机游戏对象，游戏对象命名为Player
23. 新建一个Unity脚本，命名为Player，存放到Scripts文件夹下
24. 使用Player脚本来控制飞机，WASD上下移动，J发射子弹，按K自杀并显示爆炸特效
25. 制作22题所需的子弹，爆炸特效，脚本等
26. 将完成项目上传到GitHub