

(3).如何安裝google chrome的網頁線上英文字典工具:

->更多工具->擴充功能

->左上角主選單->開啟chrome線上應用程式商店

->勾選:google製作, 免費

->搜尋:google dictionary->安裝

->到chrome右上擴充功能->點按google dictionary的『詳細資料』->擴充功能選項->my language=chinese

->打勾2個:Pop-up definitions:

(1)反白單字翻譯: Display pop-up when I double-click a word

(2)ctrl+拖曳整段翻譯: Display pop-up when I select a word or phrase

期末作業

1. 內容:

(1) 使用以下4個技術, 85分

1. 滑鼠/觸控技術, 射線ray移動,
2. AI自動導航網格移動,
3. 自動不斷產生物件(例如, 怪物),
4. 發射子彈, 摧毀怪物

(2) 加分

1. 子彈碰撞怪物, 的爆炸效果: 5分
2. 怪物碰撞到player, player被KO, 爆炸: 5分
3. 子彈打到怪物, 計算得分: 5分

4. 其它創意， 其它技術， 都可以加分

(3) 鼓勵原創

請寫出原創的百分比% (例如： 95%)

(若是100%參考外面的遊戲教學影片/網站， 則註明原創的百分比% = 0)

2. 繳交日期期限：1月9日（週一）晚上12點之前

3. 繳交作業方式：

(1) 拍成影片：前面demo遊戲， 後面介紹你使用的技術

(2) 同時上傳一個文字檔 (word/或txt) , 註明：原創的百分比%

也可以把這個『原創百分比%』， 放在影片中說明， 那就不需要上傳文字檔)

範例：

https://drive.google.com/file/d/1JcO1o9cOdoGa7ehI_xmjdt_RrNAkqyeZ/view?usp=share_link

(3) 上傳到『數位學習平台：作業區：期末考作業』

https://drive.google.com/file/d/1UASUAsBjnJ1WYzALkp3jIVe0-Q94y_K0/view?usp=share_link

https://drive.google.com/file/d/1UASUAsBjnJ1WYzALkp3jIVe0-Q94y_K0/view?usp=share_link

https://drive.google.com/file/d/1LeS087XRjH6exyCs5Lv8tlt9F03gZrsD/view?usp=share_link

下載3爆炸素材資源：

打到敵機爆炸的粒子動畫:WFX_ExplosiveSmoke Big Alt

玩家被敵軍碰撞的爆炸(核子爆炸)動畫:War FX:

WFX_Nuke

子彈飛出去的粒子動畫:flame:WFX_FlameThrower Big

Alt Looped

https://drive.google.com/file/d/15ih8GgcEW3Ilo5X9wIdOvA72ywxBmNM8/view?usp=share_link

下載1爆炸素材資源：

https://drive.google.com/file/d/1crGjOmqv-9ZSPrwrXMRI698HJuPb4rNi/view?usp=share_link

下載爆炸音效(nuclear_explode_aduio)

https://acupun.site/lecture/unity/example/exp11-6/nuclear_explode_aduio.rar

https://drive.google.com/file/d/1UASUAsBjnJlWYzALkp3jIVe0-Q94y_K0/view?usp=share_link

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.AI;

public class player_move : MonoBehaviour
{
    public GameObject a1;
    public GameObject a2;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        //1.get mouse
        if(Input.GetMouseButtonDown(0))
        {
            //1.ray
            Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
            //2.hit
            RaycastHit h1;
            //3.if hit or not?
            if(Physics.Raycast(r1,out h1)==true)
            {

GameObject.Find("player").GetComponent<NavMeshAgent>().Set
Destination(h1.point);
            }
        }

        //2.get key

```

```

        if (Input.GetKeyDown(KeyCode.F))
        {
            GameObject b1 = Instantiate(a1,a2.transform);
            b1.transform.position =
GameObject.Find("gun").transform.position;
        }
    }
}

```

```

void Update()
{
    //帧
    float dz = 5 * Time.deltaTime;
    //          m/s * 1/60 s = m
    this.gameObject.transform.Translate(0, 0, dz);
}

```

----- n2a -----

```

void Update()
{
    float dz = 5*Time.deltaTime;
    //          m/s  1/60   dt
    this.gameObject.transform.Translate(0,0, dz);

    //this.gameObject.transform.Translate(0, 0,
0.001f);
}

```

```

----- n2a -----
-----cs03 player 觸控移動程式碼-----
    void Update()
    {
        if(Input.GetMouseButtonDown(0))
        {
            Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
            RaycastHit h1;
            if(Physics.Raycast(r1, out h1)==true)
            {

GameObject.Find("player").GetComponent<NavMeshAgent>().Set
Destination(h1.point);
                }
            }
        }
    }

```

-----cs02生出怪物程式碼-----

```

public GameObject a1;
public GameObject a2;
// Start is called before the first frame update
void Start()
{
    InvokeRepeating("monsterAdd",0 ,3);
}
void monsterAdd()
{
    Instantiate(a1, a2.transform);
}

```

```
}
```

-----cs01怪物移动程式碼-----

```
void Update()
{

this.gameObject.GetComponent<NavMeshAgent>().SetDestination(GameObject.Find("player").transform.position);

}
```

預鑄件

prefab

----- 2c -----

-----PlayerMove程式碼-----

```
void Update()
{
    if(Input.GetMouseButtonDown(0))
    {

    }

}
```

-----MonsterCreate程式碼-----

```
public GameObject a1;
public GameObject a2;
// Start is called before the first frame update
void Start()
{
    InvokeRepeating("monsterAdd",0,3);
}
```

```

void monsterAdd()
{
    Instantiate(a1, a2.transform);
}

```

-----MonsterMove程式碼-----

```

void Update()
{

this.gameObject.GetComponent<NavMeshAgent>().SetDestination(GameObject.Find("player").transform.position);
}

```

預鑄件

prefab

----- n3a -----

-----MonsterCreate程式碼-----

```

public GameObject a1;
public GameObject a2;
// Start is called before the first frame update
void Start()
{
    InvokeRepeating("monsterAdd", 0, 3);
}

void monsterAdd()
{
    Instantiate(a1, a2.transform);
}

```

-----MonsterMove程式碼-----

```
void Update()
{

this.gameObject.GetComponent<NavMeshAgent>().SetDestination(GameObject.Find("player").transform.position);
}
```

預鑄件

prefab

```
----- n2a-28-----
void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        //1.ray
        Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
        //2.hit
        RaycastHit h1;
        //3.if hit ?
        if (Physics.Raycast(r1, out h1) == true)
        {

GameObject.Find("player").GetComponent<NavMeshAgent>().SetDestination(h1.point);
        }

    }
}
```

```

----- 2c-28-----
void Update()
{
    if(Input.GetMouseButtonDown(0))
    {
        //1.ray
        Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
        //2.hit
        RaycastHit h1;
        //3.if hit ?
        if(Physics.Raycast(r1,out h1))
        {

GameObject.Find("player").GetComponent<NavMeshAgent>().Set
Destination(h1.point);

        }

    }

}

}

}

----- n3a-----

```

```

void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        //1.ray
        Ray r1 =
Camera.main.ScreenPointToRay (Input.mousePosition);
        //2.hit
        RaycastHit h1;
        //3.if hit or not?
        if (Physics.Raycast (r1,out h1)==true)
        {

GameObject.Find("player").GetComponent<NavMeshAgent>().Set
Destination();

        }

    }
}

```

https://drive.google.com/file/d/1IGiqslhvAMJ9QxxPUIq7TeBBBDBR9Xxn/view?usp=share_link

----- n2a-----

```

void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        //1.ray

```

```

        Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
        //2.hit
        RaycastHit h1;
        //3.if hit or not?
        if(Physics.Raycast(r1,out h1)==true)
        {

GameObject.Find("Sphere").transform.position = h1.point;
        //mouse-collide
        if(h1.collider.gameObject.name=="Cube" ||
h1.collider.gameObject.name == "Cylinder" ||
h1.collider.gameObject.name == "Capsule" ||
h1.collider.gameObject.name == "Fox" ||
h1.collider.gameObject.name == "FantasyBee")
        {
                Destroy(h1.collider.gameObject);
        }
    }
}

```

----- hai 40 -----

```

private void OnTriggerEnter(Collider other)
{
    if(other.gameObject.name == "enemy(Clone)")
    {

```

```

        }
    }

    void Start()
    {
        Invoke("delbullet", 4);
    }

    void Update()
    {

        //1.move
        if(Input.GetMouseButtonDown(0))
        {

            Ray ray1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
            RaycastHit hit;
            if(Physics.Raycast(ray1,out hit))
            {

                //this.gameObject.transform.position = new
Vector3 (hit.point.x, hit.point.y + 1f, hit.point.z);

this.gameObject.GetComponent<NavMeshAgent>().SetDestinatio
n(hit.point);
            }
        }

        //2.fire a bullet
        if(Input.GetKeyDown(KeyCode.F))
        {
            GameObject node = Instantiate(b1, clone1);
            node.transform.position =
firepoint.transform.position;

```

```
}
```

```
}
```

```
void Update()  
{  
  
this.gameObject.transform.Translate(0,0,-2*Time.deltaTime)  
;  
}
```

```
----- 2a -----
```

```
void Update()  
{  
    if(Input.GetMouseButtonDown(0))  
    {  
        //1.ray  
        Ray r1 =  
Camera.main.ScreenPointToRay(Input.mousePosition);  
        //2.hit  
        RaycastHit h1;  
        //3.if hit or not?  
        if(Physics.Raycast(r1,out h1)==true)  
        {  
  
GameObject.Find("Sphere").transform.position = h1.point;  
        //collision  
        if(h1.collider.gameObject.name=="Cube" ||  
h1.collider.gameObject.name == "Cylinder" ||  
h1.collider.gameObject.name == "Capsule" ||
```

```

h1.collider.gameObject.name == "FantasyBee" ||
h1.collider.gameObject.name == "Fox")
    {
        Destroy(h1.collider.gameObject);
    }
}
}
}

```

```

----- n3c -----
void Update()
{
    if(Input.GetMouseButtonDown(0))
    {
        //1.ray
        Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
        //2.hit
        RaycastHit h1;
        //3.if hit or not?
        if(Physics.Raycast(r1,out h1)==true)
        {

GameObject.Find("Sphere").transform.position = h1.point;

        //4.判別碰撞對方物件的名稱
        if(h1.collider.name == "Capsule")
        {
            Destroy(h1.collider.gameObject);

```

```
        }  
    }  
  
}
```

```
public GameObject m1;  
// Start is called before the first frame update  
void Start()  
{  
    //Invoke("create", 0);  
    InvokeRepeating("create", 0, 5);  
}
```

```
void create()  
{  
    Instantiate(m1, this.gameObject.transform);  
}
```

```
void Update()
```

```

    {
        //1.ray
        Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
        //2.hit
        RaycastHit h1;
        //3.if hit happen
        if(Physics.Raycast(r1, out h1))
        {

this.gameObject.GetComponent<NavMeshAgent>().SetDestinatio
n(h1.point);
        }
    }

```

```

}

```

```

void Update()
{

```

```

this.gameObject.GetComponent<NavMeshAgent>().SetDestinatio
n(GameObject.Find("player").transform.position);
}

```

```

-----n21 4-22-----

```

```

void Update()
{
    if(Input.GetMouseButtonDown(0))
    {
        //1.ray
        Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);

```

```

        //2. hit
        RaycastHit h1;
        //3. if hit or not?
        if (Physics.Raycast(r1, out h1) == true)
        {
            //避免點按到自己, 越來越大
            if (h1.collider.gameObject.name !=
"Cylinder")
            {

GameObject.Find("Cylinder").transform.position = h1.point;
            }
        }
    }
}

```

```

-----n21 4-21-----
    private void OnMouseDown()
    {
        //1.ray
        Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
        //2.hit
        RaycastHit h1;
        //3.判別射線有沒有產生碰撞
        if (Physics.Raycast(r1, out h1) == true)
        {
            GameObject.Find("Cylinder").transform.position
= h1.point;

```

```
    }  
}
```

----hai 40 -----

```
void Update()  
{  
    if (Input.GetMouseButtonDown(0))  
    {  
        //1.ray  
        Ray r1 =  
Camera.main.ScreenPointToRay (Input.mousePosition);  
        //2.hit  
        RaycastHit h1;  
        //3.if hit or not?  
        if (Physics.Raycast(r1,out h1)==true)  
        {  
  
this.gameObject.GetComponent<NavMeshAgent>().SetDestinatio  
n(h1.point);  
        }  
    }  
}
```

https://drive.google.com/file/d/1s8cmumOIMeAXdD9c2lsQ-7FB2eqXcMcC/view?usp=share_link

-----2c -----

```
void Update()
{
    if(Input.GetMouseButtonDown(0))
    {
        //1.ray
        Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
        //2. hit
        RaycastHit h1;
        //3.if hit or not?
        if(Physics.Raycast(r1, out h1)==true)
        {

GameObject.Find("Cylinder").transform.position = h1.point;
        }
    }
}
```

```

-----n3a -----
void Update()
{
    if(Input.GetMouseButtonDown(0))
    {
        //1.r1
        Ray r1 =
Camera.main.ScreenPointToRay(Input.mousePosition);
        //2.h1
        RaycastHit h1;
        //3.if hit or not?
        if(Physics.Raycast(r1, out h1)==true)
        {

GameObject.Find("Cylinder").transform.position = h1.point;
        }

    }
}

```

```

----hai 40 -----
    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            //1.ray
            Ray ray1 =
Camera.main.ScreenPointToRay(Input.mousePosition);

            //2.hit
            RaycastHit hit1;

            //3.if hit or not
            if (Physics.Raycast(ray1, out hit1) ==
true)
            {
                //this.gameObject.transform.position
= hit1.point;

this.gameObject.GetComponent<NavMeshAgent>().SetDest
ination(hit1.point);

            }
        }
    }

```

期末作業（取代考試）

1. 內容：

（1）可以用上課教過的Unity程式技術
滑鼠/觸控技術，射線ray移動，
AI自動導航網格移動，
自動不斷產生物件（例如，怪物），
遊戲失敗的Game Over畫面，
遊戲過關的過關畫面，

寫一個小小遊戲 85

（2）可以參考外面的遊戲教學影片/網站
但是要寫網址：
以及佔據作業專案的百分比%

（3）若寫出自動產生的很多怪物朝主角靠近攻擊，主角打怪物/或被打，過關或
/Game Over（+10）

（4）若寫出塔防遊戲的（+10）

2. 繳交作業方式：

（1）拍成影片：前面demo遊戲，後面介紹你使用的技術

（2）上傳一個文字檔(word/或txt)，註明：

A. 使用技術包括：自動導航，滑鼠觸控，鍵盤控制，第一人稱，左轉右轉，怪物
不斷產生，GameOver/過關畫面

B. 參考外面的遊戲教學影片/網站的網址（可寫多個）

C. 參考外面網站技術，佔據作業專案的百分比%

D.

（2）上傳到『數位學習平台：作業區：期末考作業』

期中作業（取代考試）

1. 內容：

(1) 可以用上課教過的Unity程式技術

(旋轉/碰撞, 縮放, 改變顏色, 音效, 鍵盤控制), 寫一個小小遊戲 85

(2) 可以用上課沒有教過的Unity程式4技術, 寫一個小小遊戲 85

2. 繳交作業方式：

(1) 拍成影片：前面demo遊戲, 後面介紹你使用的技術

(2) 上傳到『數位學習平台：測驗區：期中考作業』

```
-----n2a-----
void Update()
{
    Vector3 v1;
    //(1)move
    //1.v1.x, v1.y, v1.z
    v1.x = Input.GetAxis("Horizontal")*0.05f;
    v1.z = Input.GetAxis("Vertical")*0.05f;
    v1.y = 0;

    //2.move

    this.gameObject.transform.Translate(v1,Space.Self);

    //(2)rotate
    Vector3 r1;
    r1.x = 0;
    r1.z = 0;
    r1.y = Input.GetAxis("Rotate")*0.1f;
    this.gameObject.transform.Rotate(r1);

}
```

```

private void OnTriggerEnter(Collider other)
{

    if (other.gameObject.name== "Cylinder" ||
other.gameObject.name == "Cylinder (4)" ||
other.gameObject.name == "Cylinder (3)" ||
other.gameObject.name == "Cylinder (2)" ||
other.gameObject.name == "Cylinder (1)")
    {

        Destroy(other.gameObject);

this.gameObject.GetComponent().Play();
        score += 20;
        GameObject.Find("Text
(Legacy)").GetComponent<Text>().text = score.ToString();
    }
}

```

```

----hai 40-----
using UnityEngine.UI;
public class cs01 : MonoBehaviour
{
    int score = 0;
    public AudioClip a1;
    // Start is called before the first frame update
    void Start()
    {

    }
}

```

```

// Update is called once per frame
void Update()
{
    if(Input.GetMouseButtonDown(0))
    {
        //1.ray
        Ray ray1 =
Camera.main.ScreenPointToRay(Input.mousePosition);

        //2.hit1
        RaycastHit hit1;

        //3.判別是否真的發生碰撞
        if(Physics.Raycast(ray1, out hit1)==true)
        {

//GameObject.Find("Sphere").transform.position =
hit1.point;

GameObject.Find("Sphere").transform.position = new
Vector3(hit1.point.x, 0.5f+hit1.point.y, hit1.point.z);
        }

        //4.判別碰撞物件名稱
        if(hit1.collider.gameObject.name == "Capsule"
|| hit1.collider.gameObject.name == "Cylinder" ||
hit1.collider.gameObject.name == "Cylinder (1)" ||
hit1.collider.gameObject.name == "Fox" ||
hit1.collider.gameObject.name == "FantasyBee")
        {

            Destroy(hit1.collider.gameObject);
        }

        this.gameObject.GetComponent<AudioSource>().Play();
        score += 20;

```

```

GameObject.Find("Text").GetComponent<Text>().text =
score.ToString() + "分";
        if(score==100)
        {

GameObject.Find("Text").GetComponent<Text>().text =
"Good";

this.gameObject.GetComponent<AudioSource>().clip = a1;

this.gameObject.GetComponent<AudioSource>().Play();
        }

    }

}

```

-----2c-----

```

void Update()
{
    Vector3 v1;
    //(1)move
    //1.v1.x, v1.z
    v1.x = Input.GetAxis("Horizontal")*5;

```

```

v1.z = Input.GetAxis("Vertical")*5;

//6.转换成局部坐标
Vector3 localv1;
localv1 = v1.x * Vector3.right + v1.z *
Vector3.forward;

//2. keyup(A)---> v1.x = 0
if (Input.GetKeyUp(KeyCode.A) ||
Input.GetKeyUp(KeyCode.D))
{
    v1.x = 0;
}
if (Input.GetKeyUp(KeyCode.W) ||
Input.GetKeyUp(KeyCode.S))
{
    v1.z = 0;
}
//3.keydown(space)-->jump
if(Input.GetKeyDown(KeyCode.Space))
{

this.gameObject.GetComponent<Rigidbody>().AddForce(Vector3
.up * 300);
}
//4.v1.y
v1.y =
this.gameObject.GetComponent<Rigidbody>().velocity.y;

//5.move
this.gameObject.GetComponent<Rigidbody>().velocity
= v1;

//(2)rotate
Vector3 r1;

```

```

        r1.x = 0;
        r1.z = 0;
        r1.y = Input.GetAxis("Rotate")*0.1f;
        this.gameObject.transform.Rotate(r1);
    }
}

```

-----n3a-----

```

    int score=0;
    public AudioClip a1;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        Vector3 v1;
        //1. v1.x, v1.z
        v1.x = Input.GetAxis("Horizontal")*5;
    }

```

```

        v1.z = Input.GetAxis("Vertical")*5;
        v1.y = 0;

        //5.move
        this.gameObject.GetComponent<Rigidbody>().velocity
= v1;
    }
    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.name == "Cylinder" ||
collision.gameObject.name == "Cylinder (3)" ||
collision.gameObject.name == "Cylinder (1)" ||
collision.gameObject.name == "Cylinder (2)" ||
collision.gameObject.name == "Cylinder (4)")
        {
            Destroy(collision.gameObject);

this.gameObject.GetComponent<AudioSource>().Play();
            score += 20;
            GameObject.Find("Text
(Legacy)").GetComponent<Text>().text = score.ToString() +
"分";

            if(score==100)
            {

this.gameObject.GetComponent<AudioSource>().clip = a1;

this.gameObject.GetComponent<AudioSource>().Play();
                GameObject.Find("Text
(Legacy)").GetComponent<Text>().text = "Good !";
            }
        }
    }
}

```

----hai 40-----

```
void Update()
{
    if(Input.GetMouseButtonDown(0))
    {
        //1.build ray
        Ray ray1 = Camera.main.ScreenPointToRay(Input.mousePosition);

        //2.build hit1
        RaycastHit hit1;

        //3.if hit==true or not
        if(Physics.Raycast(ray1,out hit1)==true)
        {
            GameObject.Find("Cylinder").transform.position = hit1.point;
        }
    }
}
```

----hai 40-----

```

private void OnMouseDown()
{
    //1.建立1射線
    Ray ray1 = Camera.main.ScreenPointToRay(Input.mousePosition);

    //2.建立1射線碰撞點
    RaycastHit hit1;

    //3.判別是否真的發生射線碰撞？
    if(Physics.Raycast(ray1,out hit1)==true)
    {
        GameObject.Find("Cylinder").transform.position = hit1.point;
    }
}

```

-----n2a-----

```

using UnityEngine.UI;

```

```

public class cs01 : MonoBehaviour
{
    int score = 0;
    public AudioClip a1;
    // Start is called before the first frame update
    void Start()

```

```

{

}

// Update is called once per frame
void Update()
{
    Vector3 v1;
    //1.v1.x, v1,z
    v1.x = Input.GetAxis("Horizontal") * 5;
    v1.z = Input.GetAxis("Vertical") * 5;
    v1.y = 0;

    //5.move
    this.gameObject.GetComponent<Rigidbody>().velocity = v1;
}
private void OnCollisionEnter(Collision collision)
{
    if(collision.gameObject.name== "Cylinder" || collision.gameObject.name
== "Cylinder (4)" || collision.gameObject.name == "Cylinder (3)" ||
collision.gameObject.name == "Cylinder (2)" || collision.gameObject.name ==
"Cylinder (1)")
    {
        Destroy(collision.gameObject);
        this.gameObject.GetComponent<AudioSource>().Play();
        score += 20;
        GameObject.Find("Text (Legacy)").GetComponent<Text>().text =
score.ToString() + "分";
        if(score == 100)
        {
            GameObject.Find("Text (Legacy)").GetComponent<Text>().text = "
恭喜, 過關";

```

```

        this.gameObject.GetComponent().clip = a1;
        this.gameObject.GetComponent().Play();
    }
}
}

```

-----hia 40-----

```

public Image img1;
//public GameObject img1;
// Start is called before the first frame update
void Start()
{

}

// Update is called once per frame
void Update()
{
    //1.rotate
    Vector3 rot;
    rot.x = 0;
    rot.z = 0;
    rot.y = Input.GetAxis("Rotate");
    this.gameObject.transform.Rotate(rot);

    //2.move

```

```

        Vector3 vel, localvec;
        vel.x = Input.GetAxis("Horizontal")*5;
        vel.z = Input.GetAxis("Vertical")*5;
        vel.y = 0;
        localvec = vel.x * transform.right + vel.z *
transform.forward;
        vel.x = localvec.x;
        vel.z = localvec.z;

        this.gameObject.GetComponent<Rigidbody>().velocity
= vel;

    }

    private void OnCollisionEnter(Collision collision)
    {
        if(collision.gameObject.name == "Cylinder (1)")
        {
            Destroy(collision.gameObject);
            img1.gameObject.SetActive(true);
        }
    }
}

```

Cancel

----hia 40-----

```

using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEditor;

```

```

public class cs02 : MonoBehaviour

```

```

{
    public void show()
    {
        SceneManager.LoadScene("s1");
    }

    public void myexit()
    {
        Application.Quit();
        EditorApplication.isPlaying = false;
    }
}

```

https://drive.google.com/file/d/1-pp5Ur_Et8wu28-b-LM6nmju_dH2-nxM/view?usp=share_link

-----2c-----

```

void Update()
{
    Vector3 vel;
    //1.move
    vel.x = Input.GetAxis("Horizontal")*0.01f;
    vel.z = Input.GetAxis("Vertical") * 0.01f;
    vel.y = 0;
    this.gameObject.transform.Translate(vel,Space.Self);

    //2.rotate
    Vector3 rot;
    rot.x = 0;
    rot.z = 0;
    rot.y = Input.GetAxis("Rotate")*0.2f;
    this.gameObject.transform.Rotate(rot);
}

```

```
}
```

```
private void OnTriggerEnter(Collider other)
{
    if(other.gameObject.name == "Cylinder" || other.gameObject.name ==
"Cylinder (4)" || other.gameObject.name == "Cylinder (1)" ||
other.gameObject.name == "Cylinder (3)" || other.gameObject.name ==
"Cylinder (2)")
    {
        Destroy(other.gameObject);
    }
}
```

-----n3a-----

```
void Update()
{
    Vector3 vel;
    //(2)判斷鍵盤ADWS則移動→設定vel.x = ....., vel.y
= ....(可以用2種方式: Input.GetAxis, Input.GetKeyDown
)
    vel.x = Input.GetAxis("Horizontal")*5;
```

```
vel.z = Input.GetAxis("Vertical")*5;
```

//(3) A鍵盤放開則停止移動→判別鍵盤放開

```
Input.GetKeyUp則vel.x = 0....
```

```
    if(Input.GetKeyUp(KeyCode.A) ||  
Input.GetKeyUp(KeyCode.D))  
    {  
        vel.x = 0;  
    }  
    if (Input.GetKeyUp(KeyCode.W) ||  
Input.GetKeyUp(KeyCode.S))  
    {  
        vel.z = 0;  
    }
```

//(4) 有向上跳躍的力學效果→按space鍵則使用
addforce移動(但沒有設定vel.y)

```
    if(Input.GetKeyDown(KeyCode.Space))  
    {
```

```
        this.gameObject.GetComponent<Rigidbody>().AddForce(  
Vector3.up*300);  
    }
```

//(5) 注意: 要設定vel.y(=
this.gameObject.GetComponent<Rigidbody>().velocity.y

```
    vel.y =  
this.gameObject.GetComponent<Rigidbody>().velocity.y  
;
```

//(1) 如何能夠連續移動→在update內設定幀動畫移動公式→剛體.velocity=vel, 才能控制鍵盤放開讓速度 = 0) (不能使用transform.Translate, 剛體.AddForce)

```
this.gameObject.GetComponent<Rigidbody>().velocity =  
vel;  
  
}
```

```
----hai 40 -----
```

```
int score=0;  
// Start is called before the first frame update
```

```
void Start()
```

```
{
```

```
}
```

```
// Update is called once per frame
```

```
void Update()
```

```
{
```

```
    Vector3 vel;
```

```
    //1.vel.x, vel.y
```

```
    vel.x = Input.GetAxis("Horizontal") * 5;
```

```
    vel.z = Input.GetAxis("Vertical") * 5;
```

```
    //6.localVec
```

```
    Vector3 localvec;
```

```
    localvec = vel.x * transform.right + vel.z * transform.forward;
```

```
    vel.x = localvec.x;
```

```
    vel.z = localvec.z;
```

```
    //2.A,D,W,S ==>keyup ==> vel.x = 0
```

```
    if (Input.GetKeyUp(KeyCode.A) || Input.GetKeyUp(KeyCode.D))
```

```
    {
```

```
        vel.x = 0;
```

```
    }
```

```
    if (Input.GetKeyUp(KeyCode.W) || Input.GetKeyUp(KeyCode.S))
```

```
    {
```

```
        vel.z = 0;
```

```
    }
```

```
    //3.addforce-y
```

```
    if(Input.GetKeyDown(KeyCode.Space))
```

```
    {
```

```
this.gameObject.GetComponent<Rigidbody>().AddForce(Vector3.up*300);
```

```

    }
    //4.vel.y = ??
    vel.y = this.gameObject.GetComponent<Rigidbody>().velocity.y;

    //5.move
    this.gameObject.GetComponent<Rigidbody>().velocity = vel;

    //11.rotate
    Vector3 rot;
    rot.x = 0;
    rot.z = 0;
    rot.y = Input.GetAxis("Rotate");
    this.gameObject.transform.Rotate(rot);
}

private void OnCollisionEnter(Collision collision)
{
    if(collision.gameObject.name == "Cylinder" || collision.gameObject.name
== "Cylinder (1)")
    {
        Destroy(collision.gameObject);
        score += 50;
        if(score==100)
        {
            SceneManager.LoadScene("s2");
        }
    }
}
}

```

```

-----2c -----
private void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.name == "Cylinder (2)" ||
collision.gameObject.name == "Cylinder (3)" || collision.gameObject.name ==
"Cylinder (4)")
    {
        Destroy(collision.gameObject);
        this.gameObject.GetComponent().Play();
        score += 30;
        GameObject.Find("Text (Legacy)").GetComponent<Text>().text =
score.ToString() + "分";
    }
}

```

```

void Update()
{
    Vector3 vel;
    //1.vel.x, vel.z
    vel.x = Input.GetAxis("Horizontal")*5;
    vel.z = Input.GetAxis("Vertical")*5;
    //2. keyup ==> vel.x = 0

```

```

        if (Input.GetKeyUp(KeyCode.A) || Input.GetKeyUp(KeyCode.D) ||
Input.GetKeyUp(KeyCode.LeftArrow) ||
Input.GetKeyUp(KeyCode.RightArrow))
        {
            vel.x = 0;
        }
        if (Input.GetKeyUp(KeyCode.W) || Input.GetKeyUp(KeyCode.S) ||
Input.GetKeyUp(KeyCode.UpArrow) ||
Input.GetKeyUp(KeyCode.DownArrow))
        {
            vel.z = 0;
        }
//3.space ==> jump vel.y =...
if(Input.GetKeyDown(KeyCode.Space))
{

this.gameObject.GetComponent<Rigidbody>().AddForce(Vector3.up*300);
}
//4.vel.y = ...
vel.y = this.gameObject.GetComponent<Rigidbody>().velocity.y;

//5.move
this.gameObject.GetComponent<Rigidbody>().velocity = vel;
}

```

期中作業（取代考試）

1. 內容：

（1）可以用上課教過的Unity程式技術（旋轉/碰撞，縮放，改變顏色，音效，鍵盤控制），寫一個小小遊戲 85

（2）可以用上課沒有教過的Unity程式4技術，寫一個小小遊戲 85

2. 繳交作業方式：

（1）拍成影片：前面demo遊戲，後面介紹你使用的技術

（2）上傳到『數位學習平台：測驗區：期中考作業』

3. 在學校電腦，如何下載google雲端硬碟的資料：（設定/安全性/第三方cookie/全部允許）

SQL 入門課程：14 天從語法到真實企業案例

<https://program.alphacamp.co/sql-basic>

Python、R、Julia 三大程式語言怎麼選？來自資料科學家的觀點

<https://tw.alphacamp.co/blog/r-python-julia-in-data-science-a-comparison>

<https://www.youtube.com/watch?v=S2PzRXXD4bA>

15 分鐘上手 Power BI！我一旦認真起來連我自己都會害怕 ~

https://www.youtube.com/watch?v=9RcQUh1Ib_Y

大數據時代一定要會的 SQL 商業資料分析術

<https://www.books.com.tw/products/0010786846>

第05 期Python/SQL 商業資料分析& 視覺化與網路爬蟲入門共學 ...

<https://www.accupass.com/event/2104181001161184735677>

----hai 40 -----

```
public void showairplane()
{
    GameObject.Find("Fox").SetActive(false);
    GameObject.Find("FantasyBee").SetActive(false);
    GameObject.Find("Gunship").SetActive(true);
}
public void showfox()
{
    GameObject.Find("Fox").SetActive(true);
    GameObject.Find("FantasyBee").SetActive(false);
    GameObject.Find("Gunship").SetActive(false);
}
```

----hai 40 -----

```
void Update()
{
    Vector3 vel;
    //(2)判斷鍵盤ADWS則移動→設定vel.x = ....., vel.y = ....(可以用2種方式:
Input.GetAxis, Input.GetKeyDown)
    vel.x = Input.GetAxis("Horizontal")*5;
    vel.z = Input.GetAxis("Vertical")*5;
    //(3)鍵盤放開則停止移動→判別鍵盤放開Input.GetKeyUp則vel.x = 0....
    if(Input.GetKeyUp(KeyCode.A) || Input.GetKeyUp(KeyCode.D))
    {
        vel.x = 0;
```

```

}
if (Input.GetKeyUp(KeyCode.W) || Input.GetKeyUp(KeyCode.S))
{
    vel.z = 0;
}

```

//(4) 有向上跳躍的力學效果→按space鍵則使用addforce移動(但沒有設定 vel.y)

```

if(Input.GetKeyDown(KeyCode.Space))
{
    this.gameObject.GetComponent<Rigidbody>().AddForce(Vector3.up *
300);
}

```

//(5) 注意: 要設定 vel.y(=
this.gameObject.GetComponent<Rigidbody>().velocity.y
vel.y = this.gameObject.GetComponent<Rigidbody>().velocity.y;

//(1) 如何能夠連續移動→在update內設定幀動畫移動公式→剛體.velocity=
vel, 才能控制鍵盤放開讓速度 = 0)(不能使用transform.Translate, 剛體.
AddForce)

```

    this.gameObject.GetComponent<Rigidbody>().velocity = vel;

}

```

```

----hai 40 -----
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class cs01 : MonoBehaviour
{
    int score = 0;
    public AudioClip aul;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        Vector3 vel;
        //(1) 5 steps to move
        //1.vel.x, vel.y (世界座標)
        vel.x = Input.GetAxis("Horizontal")*5;
        vel.z = Input.GetAxis("Vertical") *5;

        //6.轉成局部座標合成向量localvec;
        //Vector3.right, Vector3.forward (世界座標單位向量)
        //transform.right, transform.forward (局部座標單位向
量)

        Vector3 localvec;

```

```

        localvec = transform.right * vel.x +
transform.forward * vel.z;
        vel.x = localvec.x;
        vel.z = localvec.z;

        //2.keyup(A)==> vel.x = 0
        if (Input.GetKeyUp(KeyCode.A) ||
Input.GetKeyUp(KeyCode.D) ||
Input.GetKeyUp(KeyCode.LeftArrow) ||
Input.GetKeyUp(KeyCode.RightArrow))
        {
            vel.x = 0;
        }
        if (Input.GetKeyUp(KeyCode.W) ||
Input.GetKeyUp(KeyCode.S) ||
Input.GetKeyUp(KeyCode.UpArrow) ||
Input.GetKeyUp(KeyCode.DownArrow))
        {
            vel.z = 0;
        }

        //3.space ==> addforce(up)
        if (Input.GetKeyDown(KeyCode.Space))
        {

this.gameObject.GetComponent<Rigidbody>().AddForce(Vector3
.up*300);
        }

        //4.vel.y =
        vel.y =
this.gameObject.GetComponent<Rigidbody>().velocity.y;

        //5.velocity
        this.gameObject.GetComponent<Rigidbody>().velocity
= vel;

```

// (2). 第一人稱視角，球會跟著滾動

// 解決方法：限制剛體的轉動角度-->constraints

// (3) rotation

Vector3 rot;

rot.x = 0;

rot.z = 0;

rot.y = Input.GetAxis("Rotate");

this.gameObject.transform.Rotate(rot);

}

private void OnTriggerEnter(Collider other)

{

if (other.gameObject.name == "Cylinder (1)" ||

other.gameObject.name == "Cylinder (2)" ||

other.gameObject.name == "Cylinder (3)" ||

other.gameObject.name == "Cylinder (4)" ||

other.gameObject.name == "Cylinder")

{

this.gameObject.GetComponent().Play();

score += 20;

GameObject.Find("Text").GetComponent<Text>().text =

score.ToString() + "分";

if (score == 100)

{

GameObject.Find("Text").GetComponent<Text>().text = "恭喜,
过关";

this.gameObject.GetComponent<AudioSource>().clip = au1;

```

this.gameObject.GetComponent<AudioSource>().Play();
    }
    Destroy(other.gameObject);
}
}
}

```

-----2c -----

```

void Update()
{
    Vector3 vel;
    //1. move
    // (2) 判斷鍵盤ADWS則移動→設定vel.x = ....., vel.z =
    .... (可以用2種方式: Input.GetAxis, Input.GetKeyDown)
    vel.x = Input.GetAxis("Horizontal")*3;
    vel.z = Input.GetAxis("Vertical")*3;
    // (3) 鍵盤放開則停止移動→判別鍵盤放開Input.GetKeyUp則
    vel.x = 0....

```

```

        if (Input.GetKeyUp(KeyCode.A) ||
Input.GetKeyUp(KeyCode.D) ||
Input.GetKeyUp(KeyCode.LeftArrow) ||
Input.GetKeyUp(KeyCode.RightArrow))
        {
            vel.x = 0;
        }
        if (Input.GetKeyUp(KeyCode.W) ||
Input.GetKeyUp(KeyCode.S) ||
Input.GetKeyUp(KeyCode.UpArrow) ||
Input.GetKeyUp(KeyCode.DownArrow))
        {
            vel.z = 0;
        }
        // (4) 有向上跳躍的力學效果→按space鍵則使用addforce移動
        (但沒有設定vel.y)
        if (Input.GetKeyDown(KeyCode.Space))
        {

this.gameObject.GetComponent<Rigidbody>().AddForce(Vector3
.up * 300);
        }

        // (5) 注意：要設定vel.y(=
this.gameObject.GetComponent<Rigidbody>().velocity.y
        vel.y =
this.gameObject.GetComponent<Rigidbody>().velocity.y;

        // (1) 如何能夠連續移動→在update內設定幀動畫移動公式→剛
體.velocity=vel, 才能控制鍵盤放開讓速度 = 0) (不能使用
transform.Translate, 剛體.AddForce)
        this.gameObject.GetComponent<Rigidbody>().velocity
= vel;

        //2. rotate
        Vector3 rot;

```

```

    rot.x = 0;
    rot.z = 0;
    rot.y = Input.GetAxis("Rotate")*0.3f;
    this.gameObject.transform.Rotate(rot);
}

```

```

-----n3a -----
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.A) ||
Input.GetKeyDown(KeyCode.LeftArrow))
        {
            this.gameObject.transform.Translate(-0.5f, 0,
0);
        }
        if (Input.GetKeyDown(KeyCode.D) ||
Input.GetKeyDown(KeyCode.RightArrow))
        {
            this.gameObject.transform.Translate(0.5f, 0,
0);
        }
    }
}

```

```

    }
    if (Input.GetKeyDown(KeyCode.W) ||
Input.GetKeyDown(KeyCode.UpArrow))
    {
        this.gameObject.transform.Translate(0, 0,
0.5f);
    }
    if (Input.GetKeyDown(KeyCode.S) ||
Input.GetKeyDown(KeyCode.DownArrow))
    {
        this.gameObject.transform.Translate(0, 0,
-0.5f);
    }
    if (Input.GetKeyDown(KeyCode.Space))
    {
        this.gameObject.transform.Translate(0, 1, 0);
    }
    if (Input.GetKeyDown(KeyCode.E))
    {
        this.gameObject.transform.Rotate(0, 10, 0);
    }

}

```

```

----hai 40-----
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

```

```

public class cs01 : MonoBehaviour
{
    int score = 0;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        Vector3 vel;
        //1.move
        vel.x = Input.GetAxis("Horizontal")*0.05f;
        vel.z = Input.GetAxis("Vertical")*0.05f;
        vel.y = 0;
        this.gameObject.transform.Translate(vel.x,
vel.y,vel.z, Space.Self);

        //2.rotate
        Vector3 rot;
        rot.x = 0;
        rot.z = 0;
        rot.y = Input.GetAxis("Rotate")*0.5f;
        this.gameObject.transform.Rotate(rot.x, rot.y,
rot.z);
    }

    private void OnTriggerEnter(Collider other)
    {
        if(other.gameObject.name== "Cylinder" ||
other.gameObject.name == "Cylinder (1)" ||
other.gameObject.name == "Cylinder (2)" ||
other.gameObject.name == "Cylinder (3)" ||

```

```

other.gameObject.name == "Cylinder (3)" ||
other.gameObject.name == "Cylinder (4)")
{

this.gameObject.GetComponent().Play();
    Destroy(other.gameObject);
    score += 20;

GameObject.Find("Text").GetComponent<Text>().text =
score.ToString() + "分";
    }
}
}

```

```

----n2a-----
void Update()
{
    if (Input.GetKeyDown(KeyCode.A) ||
Input.GetKeyDown(KeyCode.LeftArrow))
    {
        //this.gameObject.transform.Translate(-1, 0, 0);

this.gameObject.GetComponent<Rigidbody>().AddForce(-1, 0,
0);
    }
    if (Input.GetKeyDown(KeyCode.D) ||
Input.GetKeyDown(KeyCode.RightArrow))
    {
        //this.gameObject.transform.Translate(1, 0,
0);

```

```

this.gameObject.GetComponent<Rigidbody>().AddForce(1, 0,
0);
    }
    if (Input.GetKeyDown(KeyCode.W) ||
Input.GetKeyDown(KeyCode.UpArrow))
    {
        //this.gameObject.transform.Translate(0, 0,
1);

this.gameObject.GetComponent<Rigidbody>().AddForce(0, 0,
1);
    }
    if (Input.GetKeyDown(KeyCode.S) ||
Input.GetKeyDown(KeyCode.DownArrow))
    {
        //this.gameObject.transform.Translate(0, 0,
-1);

this.gameObject.GetComponent<Rigidbody>().AddForce(0, 0,
-1);
    }
    //2.rotation
    if (Input.GetKeyDown(KeyCode.E))
    {
        this.gameObject.transform.Rotate(0, -20, 0);
    }
    if (Input.GetKeyDown(KeyCode.R))
    {
        this.gameObject.transform.Rotate(0, 20, 0);
    }
    //3.jump
    if (Input.GetKeyDown(KeyCode.Space))
    {
        this.gameObject.transform.Translate(0, 2, 0);
    }

```

```

//unity 的移動有2種：
//(1)幽靈模式移動： transform.Translate()

//(2)真實物體移動： 剛體.addforce()
}

```

-----海青-----

```

int score=0;
public AudioClip a1;

// Update is called once per frame
void Update()
{
    Vector3 vel;

    //2.vel.x, vel.z
    vel.x = Input.GetAxis("Horizontal")*5;
    vel.z = Input.GetAxis("Vertical")*5;
    vel.y = 0;
    //3.keyup---vel.x=0, vel.z = 0
    if(Input.GetKeyUp(KeyCode.A) ||
Input.GetKeyUp(KeyCode.LeftArrow) ||
Input.GetKeyUp(KeyCode.D) ||
Input.GetKeyUp(KeyCode.RightArrow))
    {
        vel.x = 0;
    }
}

```

```

        if (Input.GetKeyUp(KeyCode.W) ||
Input.GetKeyUp(KeyCode.UpArrow) ||
Input.GetKeyUp(KeyCode.S) ||
Input.GetKeyUp(KeyCode.DownArrow))
        {
            vel.z = 0;
        }
        //4.space -- gravity
        if(Input.GetKeyDown(KeyCode.Space))
        {

this.gameObject.GetComponent<Rigidbody>().AddForce(V
ector3.up * 300);
        }
        //5.vel.y
        vel.y =
this.gameObject.GetComponent<Rigidbody>().velocity.y
;
this.gameObject.GetComponent<Rigidbody>().velocity =
vel;
        }

        private void OnCollisionEnter(Collision
collision)
        {

            if(collision.gameObject.name == "Cylinder
(3)" || collision.gameObject.name == "Cylinder (1)"
|| collision.gameObject.name == "Cylinder")
            {

this.gameObject.GetComponent<AudioSource>().Play();

```

```

        Destroy(collision.gameObject);
        score += 30;

GameObject.Find("Text").GetComponent<Text>().text =
score.ToString()+"分";
        if (score == 100)
        {
this.gameObject.GetComponent<AudioSource>().clip =
a1;
this.gameObject.GetComponent<AudioSource>().Play();
GameObject.Find("Text").GetComponent<Text>().text =
"恭喜, 過關";
        }
    }

    private void OnTriggerEnter(Collider other)
    {
        if(other.gameObject.name == "Cylinder (2)"
|| other.gameObject.name == "Cylinder (4)")
        {
            score += 5;
            Destroy(other.gameObject);

this.gameObject.GetComponent<AudioSource>().Play();
GameObject.Find("Text").GetComponent<Text>().text =
score.ToString() + "分";
            if (score == 100)
            {

this.gameObject.GetComponent<AudioSource>().clip =
a1;

```

```

this.gameObject.GetComponent<AudioSource>().Play();
GameObject.Find("Text").GetComponent<Text>().text =
"恭喜，過關";
    }
}
}

```

-----2c-----

```

void Update()
{
    // (2) 判斷鍵盤ADWS則移動→設定vel.x = ....., vel.y =
    .... (可以用2種方式: Input.GetAxis, Input.GetKeyDown)
    // (3) 鍵盤放開則停止移動→判別鍵盤放開Input.GetKeyUp則
    vel.x = 0....
    // (4) 有向上跳躍的力學效果→按space鍵則使用addforce移動
    (但沒有設定vel.y)
    // (5) 注意: 要設定vel.y(=
    this.gameObject.GetComponent<Rigidbody>().velocity.y

    // (1) 如何能夠連續移動→在update內設定幀動畫移動公式→剛
    體.velocity=vel, 才能控制鍵盤放開讓速度 = 0) (不能使用
    transform.Translate, 剛體.AddForce)
    this.gameObject.GetComponent<Rigidbody>().velocity
}

```

-----2c-----

// Update 里面可以做24小時監測--》鍵盤

// 1/60秒，監測一次

//結論：鍵盤偵測，一定是在update裡面，才能不斷偵測

//移動無法連續，會有斷點的解決方法：改成幀動畫

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.A) ||
Input.GetKeyDown(KeyCode.LeftArrow) )
    {
        this.gameObject.transform.Translate(-1, 0, 0);
    }
    if (Input.GetKeyDown(KeyCode.W) ||
Input.GetKeyDown(KeyCode.UpArrow) )
    {
        this.gameObject.transform.Translate(0, 1, 0);
    }
    if (Input.GetKeyDown(KeyCode.D) ||
Input.GetKeyDown(KeyCode.RightArrow) )
    {
        this.gameObject.transform.Translate(1, 0, 0);
    }
    if (Input.GetKeyDown(KeyCode.S) ||
Input.GetKeyDown(KeyCode.DownArrow) )
    {
        this.gameObject.transform.Translate(0, -1, 0);
    }
    if (Input.GetKeyDown(KeyCode.E) )
    {
        this.gameObject.transform.Rotate(0, -15, 0);
    }
    if (Input.GetKeyDown(KeyCode.R) )
    {
```

```

        this.gameObject.transform.Rotate(0, 15, 0);
    }
    if (Input.GetKeyDown(KeyCode.Space))
    {
        this.gameObject.transform.Translate(0, 1, 0);
    }
}

```

-----n3a-----

//重點1：監測是否按了鍵盤，必須在update

//重點2：使用變數前，要先宣告

```
void Update()
```

```
{
```

```
    float dx, dy, dz;
```

```
    //1.移動
```

```
    dx = Input.GetAxis("Horizontal")*0.05f;
```

```
    dz = Input.GetAxis("Vertical") * 0.05f;
```

```

dy = Input.GetAxis("Jump") * 0.05f;
this.gameObject.transform.Translate(dx, dy, dz);

//2.旋轉
float rx, ry, rz;
rx = 0;
rz = 0;
ry = Input.GetAxis("Rotate");
this.gameObject.transform.Rotate(rx, ry, rz);
}

```

Rotate

-----海青40-----

```

Vector3 vel;
//跳躍變數
int num=0;
// Start is called before the first frame update
void Start()
{

}

// Update is called once per frame
void Update()
{

```

```

        // (2) 判斷鍵盤ADWS則移動→設定vel.x = . . . . , vel.y =
        . . . . (可以用2種方式 : Input.GetAxis, Input.GetKeyDown)
        //if (Input.GetKeyDown(KeyCode.D) ||
Input.GetKeyDown(KeyCode.RightArrow))
        //{
        //    vel.x = 10;
        //}
        //if (Input.GetKeyDown(KeyCode.A) ||
Input.GetKeyDown(KeyCode.LeftArrow))
        //{
        //    vel.x = -10;
        //}
        vel.x = Input.GetAxis("Horizontal")*20;
        vel.z = Input.GetAxis("Vertical")*20;

        // (3) 鍵盤放開則停止移動→判別鍵盤放開Input.GetKeyUp則
        vel.x = 0. . . .
        if (Input.GetKeyUp(KeyCode.A) ||
Input.GetKeyUp(KeyCode.LeftArrow) ||
Input.GetKeyUp(KeyCode.D) ||
Input.GetKeyUp(KeyCode.RightArrow))
        {
            vel.x = 0;
        }
        if (Input.GetKeyUp(KeyCode.W) ||
Input.GetKeyUp(KeyCode.UpArrow) ||
Input.GetKeyUp(KeyCode.S) ||
Input.GetKeyUp(KeyCode.DownArrow))
        {
            vel.z = 0;
        }
        // (4) 有向上跳躍的力學效果→按space鍵則使用addforce移動
        (但沒有設定vel.y)
        // (4 - 1) 注意 : 除了按了space鍵, 還要判斷是否跳躍變數num

```

```
        // (4 - 2) 注意：當發生碰撞，on_collision() 則設定 num = 0, 跳躍變數歸零
```

```
        if (Input.GetKeyDown(KeyCode.Space) && num < 2)
        {
            num += 1;
        }
    }
}
```

```
this.gameObject.GetComponent<Rigidbody>().AddForce(Vector3.up * 350);
    }
}
```

```
        // (5) 還要要設定 vel.y (=
        this.gameObject.GetComponent<Rigidbody>().velocity.y
        vel.y =
        this.gameObject.GetComponent<Rigidbody>().velocity.y;
```

```
        // (1) 如何能夠連續移動→在update內設定幀動畫移動公式→剛體.velocity=vel, 才能控制鍵盤放開讓速度 = 0) (不能使用 transform.Translate, 剛體.AddForce)
```

```
        this.gameObject.GetComponent<Rigidbody>().velocity
        = vel;
    }
}
```

```
//碰撞
```

```
private void OnCollisionEnter(Collision collision)
{
    num = 0;
}
}
```

-----連跳2次限制-----

```
Vector3 vel;
int num = 0;
```

```

// Start is called before the first frame update
void Start()
{

}

// Update is called once per frame
void Update()
{

    //          完美移動演算法的5步驟：
    // (2) 判斷鍵盤ADWS則移動→設定vel.x = ....., vel.y =
    .... (可以用2種方式：Input.GetAxis, Input.GetKeyDown)
    vel.x = Input.GetAxis("Horizontal") * 9;
    vel.z = Input.GetAxis("Vertical") * 9;

    // (3) 如何鍵盤停止則移動停止→判別鍵盤放開Input.GetKeyUp
    則vel.x = 0....
        if (Input.GetKeyUp(KeyCode.A) ||
Input.GetKeyUp(KeyCode.D))
        {
            vel.x = 0;
        }
        if (Input.GetKeyUp(KeyCode.W) ||
Input.GetKeyUp(KeyCode.S))
        {
            vel.z = 0;
        }

    // (4) 如何能夠有向上跳躍的力學效果→space鍵盤則使用
    addforce (但沒有設定vel.y)
        //if(Input.GetKeyDown(KeyCode.Space) &&
this.gameObject.GetComponent<Rigidbody>().velocity.y ==0)
        if (Input.GetKeyDown(KeyCode.Space) && num < 1)
        {

```

```
this.gameObject.GetComponent<Rigidbody>().AddForce(Vector3  
.up * 290);
```

```
    num += 1;
```

```
    }
```

```
    if
```

```
(this.gameObject.GetComponent<Rigidbody>().velocity.y ==  
0)
```

```
    {
```

```
        num = 0;
```

```
    }
```

```
    // (5) 注意：要設定vel.y(=
```

```
this.gameObject.GetComponent<Rigidbody>().velocity.y
```

```
    vel.y =
```

```
this.gameObject.GetComponent<Rigidbody>().velocity.y;
```

// (1) 如何能夠連續運動→在update內設定幀動畫移動公式→剛
體.velocity=vel, 才能控制鍵盤放開讓速度 = 0) (不能使用

transform.Translate, 剛體.AddForce)

```
    this.gameObject.GetComponent<Rigidbody>().velocity  
= vel;
```

```
    //輸出變數有2種方法：
```

```
    //Debug.Log(this.gameObject.transform.position.y);
```

```
    //print(this.gameObject.transform.position.y);
```

```
    print(vel);
```

```
    }
```

-----n2a-----

```
    // 要監測鍵盤，必須在update裡面監測
```

```
    // 注意1：使用變數前，要先宣告
```

```
    // 注意2：使用變數前，要先給起始值
```

```
void Update()
```

```
{
```

```
    float dx,dz,dy;
```

```
    float v1=0.05f, v2=0.05f, v3=0.05f;
```

```

//1.移動
dx = Input.GetAxis("Horizontal")*v1;
dz = Input.GetAxis("Vertical")*v2;
dy = Input.GetAxis("Jump")*v3;
this.gameObject.transform.Translate(dx, dy, dz);

//2.旋轉
float rx, ry, rz;
rx = 0;
ry = Input.GetAxis("Rotate")*0.2f;
rz = 0;
this.gameObject.transform.Rotate(rx, ry, rz);

}

```

--海青40-----

```

Vector3 vel;
int num = 0;
// Start is called before the first frame update
void Start()
{

}

```

```

// Update is called once per frame

```

```

void Update()
{

    //          完美移動演算法的5步驟：
    // (2) 判斷鍵盤ADWS則移動→設定vel.x = ....., vel.y =
    ..... (可以用2種方式：Input.GetAxis, Input.GetKeyDown)
    vel.x = Input.GetAxis("Horizontal")*5;
    vel.z = Input.GetAxis("Vertical")*5;

    // (3) 如何鍵盤停止則移動停止→判別鍵盤放開Input.GetKeyUp
    則vel.x = 0.....
    if (Input.GetKeyUp(KeyCode.A) ||
Input.GetKeyUp(KeyCode.D))
    {
        vel.x = 0;
    }
    if (Input.GetKeyUp(KeyCode.W) ||
Input.GetKeyUp(KeyCode.S))
    {
        vel.z = 0;
    }
    // (4) 如何能夠有向上跳躍的力學效果→space鍵盤則使用
    addforce (但沒有設定vel.y)
    //if(Input.GetKeyDown(KeyCode.Space) &&
this.gameObject.GetComponent<Rigidbody>().velocity.y ==0)
    if (Input.GetKeyDown(KeyCode.Space) && num <1)
    {

this.gameObject.GetComponent<Rigidbody>().AddForce(Vector3
.up*350);

        num += 1;
    }

    if(this.gameObject.GetComponent<Rigidbody>().velocity.y==0
)
    {

```

```

        num = 0;
    }
    // (5) 注意：要設定vel.y(=
this.gameObject.GetComponent<Rigidbody>().velocity.y
    vel.y =
this.gameObject.GetComponent<Rigidbody>().velocity.y;

    // (1) 如何能夠連續運動→在update內設定幀動畫移動公式→剛
體.velocity=vel, 才能控制鍵盤放開讓速度 = 0) (不能使用
transform.Translate, 剛體.AddForce)
    this.gameObject.GetComponent<Rigidbody>().velocity
= vel;

    //輸出變數有2種方法:
    //Debug.Log(this.gameObject.transform.position.y);
    //print(this.gameObject.transform.position.y);
    print(vel);
}

```

-----2c-----

```

// Update 里面可以做24小時監測--》鍵盤
// 1/60秒, 監測一次
//結論：鍵盤偵測，一定是在update裡面，才能不斷偵測

```

```

// (1) 按 : A, D : Input.GetAxis("Horizontal");
// (2) 按 : W, S : Input.GetAxis("Vertical");
// (3) 按 : space : Input.GetAxis("Jump");
//注意：使用變數前要先宣告
void Update()
{
    //區域變數
    float dx, dz, dy;
    float a1= 0.3f, a2 = 0.3f, a3 = 7;
    dx = Input.GetAxis("Horizontal")*a1;
    dz = Input.GetAxis("Vertical")*a2;
    dy = Input.GetAxis("Jump")*a3;
    //1.動力學移動（沒有考慮力學，沒有摩擦力不會滾動，沒有重力加速減速，只會等速）
    //GameObject.Find("Sphere").transform.Translate(dx, dy, dz);
    //2.力學移動（考慮力學，有摩擦力會滾動，有重力會加速減速）
    GameObject.Find("Sphere").GetComponent<Rigidbody>().AddForce(dx,dy,dz);

    //轉動
    float ry;
    ry = Input.GetAxis("Rotate");
    GameObject.Find("Sphere").transform.Rotate(0,ry,0);
}

```

遊戲的物件移動有2種模式：

（1）動力學模式（不考慮力學，只是移動）

使用**Translate**函數

```
GameObject.Find("Sphere").transform.Translate(1, 0, 0);
```

（2）力學模式（考慮重力，摩擦力，角動量）

要加上**Rigidbody**剛體，才會有力學效果

使用**AddForce**函數

期中作業（取代考試）

1. 內容：

（1）可以用上課教過的Unity程式技術（移動，旋轉，縮放，改變顏色，音效，鍵盤控制），寫一個小小遊戲 85

（2）可以用上課沒有教過的Unity程式4技術，寫一個小小遊戲 85

2. 繳交作業方式：

（1）拍成影片：前面demo遊戲，後面介紹你使用的技術

（2）上傳到『數位學習平台：測驗區：期中考作業』

3. 在學校電腦，如何下載google雲端硬碟的資料：（設定/安全性/第三方cookie/全部允許）

-----n3c-----

```
using UnityEngine.UI;
```

```
public InputField a1;
```

```
//1.物件三種基本運動
```

```
//移動的函數：Tranlate(1,0,0)
```

```
//轉動的函數：Rotate(0,1,0)
```

```
//縮放的參數（沒有函數）：localScale
```

```
GameObject.Find("Sphere").transform.Rotate(0,  
float.Parse(a1.text)/60, 0);
```

```
try
```

```

        {
GameObject.Find("Sphere").transform.Rotate(0,
float.Parse(a1.text) / 60, 0);
        }
        catch
        {
GameObject.Find("Sphere").transform.Rotate(0, 0, 0);
        }

```

```

using UnityEngine.UI;

```

```

public class cs01 : MonoBehaviour

```

```

{
    //int a1;
    public InputField a1;

    //1秒内進來執行60次 frame, 1/60 秒
    //形成卡通動畫效果: frame animaiton = 幀動畫
    void Update()
    {
        //1.移動
        //GameObject.Find("Sphere").transform.position =
new Vector3(1,0,0)

//GameObject.Find("Sphere").transform.Translate(1,0,0)

        //2.轉動
        //GameObject.Find("Sphere").transform.rotation =
new Vector3(1, 0, 0)
        try
        {

```

```

        this.gameObject.transform.Rotate(0,
float.Parse(a1.text) / 10, 0);
    }
    catch
    {
        this.gameObject.transform.Rotate(0, 0, 0);
    }

}

```

```

-----n2a-----
    void Update()
    {
        //1.移動
        //GameObject.Find("Sphere").transform.position =
new Vector3(1,0,0)

//GameObject.Find("Sphere").transform.Translate(1,0,0)

        //2.轉動
        //GameObject.Find("Sphere").transform.rotation =
new Vector3(1, 0, 0)
        this.gameObject.transform.Rotate(0, 1, 0);
    }

```

-----海40-----

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.A))
    {
        this.gameObject.transform.Translate(-1, 0, 0);
    }
    if (Input.GetKeyDown(KeyCode.D))
    {
        this.gameObject.transform.Translate(1, 0, 0);
    }
}
```

-----海40-----

```
void Update()
{
    //1. 鍵盤移動
    float dx, dy, dz;
    float v1=1, v2=3, v3=1;
    dx = Input.GetAxis("Horizontal")*v1;
    dy = Input.GetAxis("Jump")*v2;
    dz = Input.GetAxis("Vertical")*v3;
```

```
GameObject.Find("Sphere").GetComponent<Rigidbody>().AddForce(dx, dy, dz);
```

```
    //2. 滑鼠移動
    float mx, my, mz;
    float m1=6, m2=6;
    mx = Input.GetAxis("Mouse X")*m1;
    my = 0;
    mz = Input.GetAxis("Mouse Y")*m2;
```

```
this.gameObject.GetComponent<Rigidbody>().AddForce(mx, my, mz);
```

```

        //3.旋轉
        float rx, ry, rz;
        rx = 0;
        ry = Input.GetAxis("Rotate");
        rz = 0;
        this.gameObject.transform.Rotate(rx, ry, rz);
    }

```

-----2c-----

```

        //int a1;
        public AudioClip a1, a2;

        private void OnMouseDown()
        {
            //1.縮小
            GameObject.Find("Cube").transform.localScale = new
            Vector3(0.5f, 0.5f, 0.5f);
            //2.顏色
            this.gameObject.GetComponent<Renderer>().material.color =
            Color.red;
            //3.音效

            this.gameObject.GetComponent<AudioSource>().Play();

        }

        private void OnMouseUp()
        {

```

```

        //1.放大
        this.gameObject.transform.localScale = new
Vector3(1, 1, 1);
        //2.顏色
        this.gameObject.GetComponent<Renderer>().material.color =
Color.white;
    }

```

-----2c-----

1.比較2種點按事件：

**(1)點按button事件(限制:只能設定按下的事件):按下
onClick事件**

方法:點按button事件

-->按下onClick事件

-->public void showBlue()

**(2)滑鼠點按物件事件:按下onMouseDown事件,放開
onMouseUp事件**

方法:滑鼠點按物件事件:

-->void OnMouseDown()

-->void OnMouseUp()

-----40-----

```

public AudioClip a1,a2;

```

```

    private void OnMouseDown()
    {
        //1.縮小
        GameObject.Find("Cube").transform.localScale = new
Vector3(0.5f, 0.5f, 0.5f);
    }

```

```

        //2.顏色
GameObject.Find("Cube").GetComponent<Renderer>().material.
color = Color.red;
        //3.音效
GameObject.Find("Cube").GetComponent<AudioSource>().clip =
a1;
GameObject.Find("Cube").GetComponent<AudioSource>().Play()
;

    }

    private void OnMouseUp()
    {
        //1.放大
        GameObject.Find("Cube").transform.localScale = new
Vector3(1, 1, 1);
        //2.顏色
GameObject.Find("Cube").GetComponent<Renderer>().material.
color = Color.white;
        //3.音效
GameObject.Find("Cube").GetComponent<AudioSource>().clip =
a2;
GameObject.Find("Cube").GetComponent<AudioSource>().Play()
;

    }

```

-----n3a-----

1.比較2種點按事件：

(1)點按**button**事件(限制：只能設定按下的事件)：按下
onClick事件

方法：點按button事件

-->按下onClick事件

-->public void showBlue()

(2) 滑鼠點按物件事件：按下onMouseDown事件，放開onMouseUp事件

方法：滑鼠點按物件事件：

-->void OnMouseDown()

-->void OnMouseUp()

-----40-----

//全域變數

 //float dx;

 public float v1 = 0.2f;

 float v2;

 //24小時監測，user是否按了鍵盤

 void Update()

 {

 //使用變數前，要先宣告

 //區域變數

 float dx, dy, dz;

 //1. 鍵盤移動

 dx = Input.GetAxis("Horizontal")*v1;

 dz = Input.GetAxis("Vertical") * v1;

 dy = Input.GetAxis("Jump") * v1/3;

 //GameObject.Find("Sphere").transform.Translate(dx, dy, dz);

 GameObject.Find("Sphere").GetComponent<Rigidbody>().AddForce(dx, dy, dz);

 }

----n2a--4-9-----

```
private void OnMouseDown()
{

this.gameObject.GetComponent().Play();

}
```

----n2a-4-5-----

```
public AudioClip a1, a2;
private void OnMouseDown()
{
    //1.縮小
    GameObject.Find("Cube").transform.localScale = new
Vector3(0.5f, 0.5f, 0.5f);
    //2.顏色
    GameObject.Find("Cube").GetComponent<Renderer>().material.
color = Color.red;
    //3.音效
    GameObject.Find("Cube").GetComponent<AudioSource>().clip =
a1;
    GameObject.Find("Cube").GetComponent<AudioSource>().Play()
;
}
private void OnMouseUp()
{
    //1.放大
    GameObject.Find("Cube").transform.localScale = new
Vector3(1, 1, 1);
```

```

        //2.顏色
GameObject.Find("Cube").GetComponent<Renderer>().material.
color = Color.blue;
        //3.音效
GameObject.Find("Cube").GetComponent<AudioSource>().clip =
a2;
GameObject.Find("Cube").GetComponent<AudioSource>().Play()
;
    }

```

-----n2a-----

1.比較2種點按事件：

(1)點按button事件(限制:只能設定按下的事件):按下
onClick事件

方法:點按button事件

-->按下onClick事件

-->public void showBlue()

(2)滑鼠點按物件事件:按下**onMouseDown**事件, 放開
onMouseUp事件

方法:滑鼠點按物件事件:

-->void OnMouseDown()

-->void OnMouseUp()

-----40-----

```

using UnityEngine.UI;

float n1;
public InputField a1;

```

// 幀動畫：在update裡面的運動，就是幀動畫

```

// 60 次/sec
void Update()
{
    try
    {
        n1 = float.Parse(a1.text);
    }
    catch
    {
        n1 = 0;
    }

    //移動 = Translate()
    //轉動 = Roate()
    //GameObject.Find("Sphere").transform.Rotate(new
Vector3(0,1,0));
    GameObject.Find("Sphere").transform.Rotate(0, n1, 0);
    //GameObject.Find("Sphere").transform.Rotate(Vector3.up);

}

```

-----40-----

```

public AudioClip a1, a2;
private void OnMouseDown()
{
    //1.color
    GameObject.Find("Cube").GetComponent<Renderer>().material.color
= Color.red;
    //2.audio
    GameObject.Find("Cube").GetComponent<AudioSource>().clip = a1;
    GameObject.Find("Cube").GetComponent<AudioSource>().Play();
    //3.scale
    GameObject.Find("Cube").transform.localScale = new
Vector3(0.5f, 0.5f, 0.5f);

}

```

```

private void OnMouseUp()
{
    //1.color
    GameObject.Find("Cube").GetComponent<Renderer>().material.color
= Color.blue;
    //2.audio
    GameObject.Find("Cube").GetComponent<AudioSource>().clip = a2;
    GameObject.Find("Cube").GetComponent<AudioSource>().Play();
    //3.scale
    GameObject.Find("Cube").transform.localScale = new Vector3(1,
1, 1);
}

```

-----40-----

1.比較2種點按事件：

**(1)點按button事件(限制：只能設定按下的事件)：按下
onClick事件**

方法：點按button事件

-->按下onClick事件

-->public void showBlue()

**(2)滑鼠點按物件事件：按下onMouseDown事件，放開
onMouseUp事件**

方法：滑鼠點按物件事件：

-->void OnMouseDown()

-->void OnMouseUp()

--2c-----

```
using UnityEngine.UI;
```

//1.四種基本變數宣告

```
int a1;  
float a2;  
string a3;  
bool a4;
```

//2.物件變數

```
public Text a5;  
public InputField a6;  
public InputField c1, c2, c3;
```

```
public void show()  
{  
    a5.text = "90分";  
    a6.text = "90分";  
}
```

```
public void show2()  
{  
    c3.text = (int.Parse(c1.text) +  
int.Parse(c2.text)).ToString();  
}
```

-----2c-----

//1.四種基本變數宣告

```
int a1;  
float a2;  
string a3;  
bool a4;
```

```
//2.物件變數  
public Text a5;
```

```
public void show()  
{  
    a5.text = "90分";  
}
```

-----2c-----

//如何抓場景物件，有2個方法

//1.方法1：三層抓法

```
//GameObject.Find("Sphere").transform.position += new  
Vector3(1, 0, 0);
```

//2.方法2：直接前台設定

```
using UnityEngine.UI;  
public Text a5;  
a5.text = "90分";
```

```
public AudioClip a1, a2;  
public void showblue()  
{  
    //1.三層抓法  
    //音效播放
```

```
GameObject.Find("Cube").GetComponent().clip =  
a1;
```

```
GameObject.Find("Cube").GetComponent().Play()  
;
```

//改變顏色

```
GameObject.Find("Cube").GetComponent<Renderer>().material.  
color = Color.blue;  
}
```

```
public void showred()  
{
```

```
GameObject.Find("Cube").GetComponent<AudioSource>().clip =  
a2;
```

```
GameObject.Find("Cube").GetComponent<AudioSource>().Play()  
;
```

```
GameObject.Find("Cube").GetComponent<Renderer>().material.  
color = Color.red;  
  
}
```

1.比較2種點按事件：

(1)點按button事件(限制：只能設定按下的事件)：按下
onClick事件

方法：點按button事件

-->按下onClick事件

-->public void showBlue()

(2) 滑鼠點按物件事件: 按下 **onMouseDown** 事件, 放開 **onMouseUp** 事件

方法: 滑鼠點按物件事件:

-->void OnMouseDown()

-->void OnMouseUp()

```
using UnityEngine.UI;

//1.4種基本變數宣告
int a1;
float a2;
string a3;
bool a4 = true;

//2.物件變數
public Text a5;
public InputField a6;
public InputField b1, b2, b3;

public void show()
{
    //如何抓場景物件, 有2個方法
    //1.方法1: 三層抓法
    //GameObject.Find("Text (Legacy)").

    //2.方法2: 直接前台設定
    a5.text = "90分";
    a6.text = "90分";
}

public void show2()
{
```

```
        b3.text = (int.Parse(b1.text) +
int.Parse(b2.text)).ToString();
    }
```

```
public void showblue()
{

}
```

```
using UnityEngine.UI;
using TMPro;
```

```
//4.tmp
public TMP_InputField r1;
public TMP_Text area;

public void show3()
{
    area.text = (3.14 * float.Parse(r1.text) *
float.Parse(r1.text)).ToString();
}
```

```
public void show2()
{
    t3.text = (int.Parse(t1.text) +
int.Parse(t2.text)).ToString();

}
```

```
//1.基本變數，就4種
int a1;
float a2;
string a3;
bool a4;
//2.物件變數：很多很多種
public Text a5;
public InputField a6;

public void show()
{
    //如何抓場景物件，有2個方法
    //1.方法1：三層抓法
    //GameObject.Find("Text")

    //2.方法2：前台直接設定
    a5.text = "90分";
    a6.text = "90分";

}
```

```
public void show()  
{  
  
}
```

```
using UnityEngine.UI;
```

```
//如何抓場景物件，有2個方法  
//1.方法1：三層抓法  
//GameObject.Find("Text  
(Legacy) ") .
```

```
//2.方法2：直接前台設定
```

遊戲的物件移動有2種模式：

(1) 動力學模式（不考慮力學，只是移動）

使用Translate函數

```
GameObject.Find("Sphere").transform.Translate(1, 0, 0);
```

(2) 力學模式（考慮重力，摩擦力，角動量）

要加上RigidBody剛體，才會有力學效果

使用AddForce函數

```
-----  
  
    //1. 動力學的移動  
    public void right()  
    {  
  
        GameObject.Find("Sphere").transform.Translate(Vector3.righ  
t);  
    }  
  
    //2. 力學移動方法  
    public void left()  
    {  
  
        GameObject.Find("Sphere").GetComponent<Rigidbody>().AddFor  
ce(Vector3.left*30);  
    }
```

```
GameObject.Find("Sphere").GetComponent<Rigidbody>().AddForce(Vector3.left);
```

```
public void left()  
{  
    //2. 力學移動模式
```

```
GameObject.Find("Sphere").GetComponent<Rigidbody>().AddForce(Vector3.left*40);  
}
```

```
public void right()  
{  
    //1. 動力學的移動方法  
    // (1) 原始方法  
    //GameObject.Find("Sphere").transform.position += new  
    Vector3(1, 0, 0);  
  
    // (2) 常用方法  
    //GameObject.Find("Sphere").transform.Translate(new  
    Vector3(1, 0, 0));  
    //GameObject.Find("Sphere").transform.Translate(1, 0,  
    0);  
  
    GameObject.Find("Sphere").transform.Translate(Vector3.right);  
}
```

```
-----  
    //1.原始寫法  
    //GameObject.Find("Sphere").transform.position += new  
Vector3(1, 0, 0);  
  
    //2.常用寫法  
    //GameObject.Find("Sphere").transform.Translate(new  
Vector3(1, 0, 0));  
    //GameObject.Find("Sphere").transform.Translate(1, 0,  
0);  
  
GameObject.Find("Sphere").transform.Translate(Vector3.right);  
  
-----
```

```
GameObject.Find("Sphere").GetComponent<Rigidbody>().  
AddForce(Vector3.left);
```

//1.移動到固定點

```
//GameObject.Find("Sphere").transform.position = new  
Vector3(2, 0.5f, 0);  
    //整數 int  
    //小數點有2點: float (173.5) , double (  
35454885488418154544489.464646644666454)
```

//2.不斷累計移動（再向右）（使用參數法）

```
//GameObject.Find("Sphere").transform.position +=  
new Vector3(1, 0, 0);
```

//3.再向右移動 Translate(使用函數法)

```
//GameObject.Find("Sphere").transform.Translate(new  
Vector3(1,0,0));
```

```
//GameObject.Find("Sphere").transform.Translate(1,  
0, 0);
```

```
GameObject.Find("Sphere").transform.Translate(Vector  
3.right);
```

遊戲的物件移動有2種模式：

(1) 動力學模式（不考慮力學，只是移動）

使用Translate函數

```
GameObject.Find("Sphere").transform.Translate(1, 0, 0);
```

(2) 力學模式（考慮重力，摩擦力，角動量）

要加上Rigidbody剛體，才會有力學效果

使用AddForce函數

```
GameObject.Find("Sphere").GetComponent<Rigidbody>().AddForce(Vector3.left);
```

1. 課程名稱：行動遊戲設計

2. 使用遊戲引擎：Cocos Creator
(簡稱C.C.)

3. 遊戲三大引擎
(在台灣就業市場的前三大，可以到
104人力銀行查詢)

一.unreal

- (1) 若用程式語言開發：c++
- (2) 用非程式開發工具：blueprint 藍圖

二.Unity

- (1) 若用程式語言開發：c#
- (2) 用非程式開發工具：bolt, playmaker

三.Cocos Creator

- (1) 若用程式語言開發：javascript
- (2) 用非程式開發工具：

4.Unity互動程式元件化設計

(1) Unity學習三步驟：

Unity基礎 → 各種Unity互動效果之程式設計 →
Unity專案

(2) 原理：

學習各種效果的程式寫法，是基本功，可以將它元件化學習，每種效果一個元件，可以複製程式碼

(3) 遊戲程式學習技巧：

元件化，模組化，單元化，網頁化，隨查即得，快速應用

這20多年來，多媒體領域的幾波浪潮

1. 20多年前：網頁設計（前端網頁，後端網頁）

2. 9年前：手機APP設計

3. 5年前：人工智慧

4. 今年開始：元宇宙（VR虛擬實境，AR擴增實境，MR混合實境）

未來的VR眼鏡會成為主流

比較程式學

一, python

1. 基本

(1) 輸出

```
print("hi...")
```

(3) 輸入

```
input()
```

(2) 換行 \n

```
newline
```

(4) 變數

變數使用前不用宣告

```
a2 = "john";
```

2. 變數, 數字的運算

(1) 注解

```
c++: //
```

```
python: #
```

(1) **type(變數)**

- `a = 123` , `type(a)`會告訴我們 `int` (整數)。
- `b = '456'` , `type(b)`會告訴我們 `str` (字串)。

- `c = 8.70` , `type(c)`會告訴我們這是一個 `float` (浮點數)

(2) 計算, 加減乘除, 商數, 餘數, 次方

(3) 不要把字串和數值加在一起、否則會產生`TypeError`

解決方法：用 `int()`, `str()`, `float()` 來轉換變數型態

(4) 應用題：不要把字串和數值加在一起 (`a1 = input(...)`)

3. 選擇結構`if`

(1) 雙向選擇：判讀分數`a`, 及格, 或不及格?

```
if a>=60:
    print('及格')
else:
    print('不及格')
```

(2) 練習：判斷輸入的數字, 是奇數, 或偶數?

(3) 多向選擇：判斷分數`a`, 是優等 (`a>85`), 或中等, 或不及格 (`a<60`)

```
a = int(input('輸入a = '))
if a>=85:
    print('優等')
elif a>=70 and a<85:
    print('中等')
else:
    print('不及格')
```

(4) 練習：判斷輸入的數字a,b，誰大？

(5) 練習：人臉辨識，屬於哪一種人工智慧？1. 語音辨識，2. 自然語言，3. 影像辨識（10分）

4. 迴圈結構for, while

(1) for迴圈

公式：for 變數 in range(起始值,終止值,遞增減值)：

要重複做的程式

注意：range的範圍為終止值的前一個值

範例：range(1,10)，表示：1,2,3,4,5,6,7,8,9

範例6-1：印出10行恭喜發財

```
for i in range(1,11,1):  
    print(i, '恭喜發財')
```

(2) 練習4-2：印出99乘法表

(3) 簡化寫法：只寫終止值（若是起始值=1，遞增1，可以省略不寫）

```
for i in range(10):
```

(4) .while迴圈

公式：

i=起始值

while i <= 終止值：

 運算程式

 i +=1

注意：在python的i=i+1不能夠寫成i++

範例4-4：印出10行恭喜發財

```
i=1
```

```
while i <= 10:
```

```
print(i, '恭喜發財')  
i +=1
```

(5). 練習4-5：計算1到10的總和

5. 串列 (**list**)：類似傳統的陣列 **array**

(1). 建立方法：

```
a = ['tom','mike','peter']
```

(2). 取出其中元素：**a[0]**

```
a = ['tom','mike','peter']
```

```
print(a[0])
```

(3). 計算串列(**list**)長度函數：**len(tuple)**

```
a=['tom','mike','peter']
```

```
b=len(a)
```

(4). 修改串列(**list**)的某個元素：**a[1] = '新值'**

```
a=['tom','mike','peter']
```

```
a[1] = 'jane'
```

```
print(a)
```

(5). 顯示串列(**list**)的某個元素的編號**index**：**a.index(元素)**

```
a=['tom','mike','peter']
```

```
b = a.index('peter')  
print('b')
```

(6).新增一個元素到串列(list)的最後: a.append(元素)

```
a=['tom','mike','peter']  
a.append('marry')  
print(a)
```

#錯誤寫法:

```
b = a.append('marry')  
print(b)
```

(7).新增一個元素到串列(list)的某個位置: a.insert(位置, 元素)

```
a=['tom','mike','peter']  
a.insert(1,'marry')  
print(a)
```

(8).刪除第幾個元素: del a[編號]

```
a=['tom','mike','peter']  
del a[1]  
print(a)
```

(9).刪除第幾個元素: a.pop(編號)

```
a=['tom','mike','peter']  
a.pop(1)
```

```
print(a)
a.pop() #若是沒有指定編號，則刪除最後一個
print(a)
```

(10).讀取串列裡面的每個元素:for 元素 in 串列

```
a=['tom','mike','peter']
for 元素 in a:
    print(元素)
```

#(11).排序:a.sort()

```
a=['tom','mike','peter']
a.sort()
print(a)
```

6. 大數據資料的視覺化 (Matplotlib模組)

#1. 常用繪圖的基本語法：

```
# (1) 畫出一條紅色線條：(原始寫法)
# pyplot : 繪圖函式庫
import matplotlib.pyplot
x = [1,2,3,4,5]
y = [10,8,13,16,20]
matplotlib.pyplot.plot(x,y,'r--')
matplotlib.pyplot.show()
```

(2) 常用繪圖的標準語法：

```
import matplotlib.pyplot as plt
x = [1,2,3,4,5]
y = [10,8,13,16,20]
plt.plot(x, y, 'r--')
```

(3) 畫出二條線，紅色，藍色

```
import matplotlib.pyplot as plt
x = [1,2,3,4,5]
y1 = [9,3,5,6,2]
y2 = [8,10,8,5,1]
plt.plot(x,y1, 'r-', label='john')
plt.plot(x,y2, 'b--', label='tom')
#標題
plt.title('my money')
plt.xlabel('month')
plt.ylabel('money')
#顯示圖例標籤
plt.legend()
plt.show()
```

(4) 畫出左右兩個子圖：

```
import matplotlib.pyplot as plt
x = [1,2,3,4]
y = [2,4,6,8]
#左半部，紅色，圓點圖
#上下1個，左右2個，在第1個位置：subplot(1,2,1)
plt.subplot(1,2,1)
plt.plot(x,y, 'ro')
#右半部，實線，綠色圖
#上下1個，左右2個，在第2個位置：subplot(1,2,2)
plt.subplot(1,2,2)
plt.plot(x,y, 'g-')
plt.show()
```

(5) 顯示一個圖片：

pyplot : 繪圖函式庫

image : 圖片函數庫

import matplotlib.pyplot as plt

import matplotlib.image as img

plt.imshow(img.imread("https://mg.lhu.edu.tw/var/file/3/1003/msys_1003_4796605_43808.png"))

7 . Numpy : 矩陣

(1) **Numpy**專門用來處理矩陣，它的運算效率比list串列更高效。

(2) 串列轉成矩陣，指令：**矩陣 = np.array(串列)**

(2) 高等數學運算，都是以矩陣的方式來進行運算，例如人工智慧，機器學習，深度學習，類神經網路計算等。(3) 高等數學計算，numpy是大數據分析的最重要工具

(4) 範例1

```
import numpy as np
```

```
//陣列, 串列
```

```
list1 = ['tom','john','peter','jolin']
```

```
//把陣列, 轉成矩陣
```

```
name = np.array(list1)
```

```
//也可以寫成: np.array(['tom','john','peter','jolin'])
```

```
#印出第2個人名
```

```
print('第2個人名=', name[1])
```

```
#印出全部人名
```

```
for item in name:
```

```
print(item)
```

#印出矩陣的維度, 長度

print('矩陣的維度, 長度=', name.shape, '個人')

print('矩陣的維度, 長度=', len(name))

8. 人工智慧程式撰寫

(1) 機器學習/深度學習，數學模型流程的5步驟

A. 建立模型

B. 建立目標函數(就是：誤差函數, **loss function**)

C. 訓練模型

D. 測試模型，避免**overfitting**(過度擬合)

D. 模型的預測

#(2) tensorflow AI程式實作的6個步驟

#目的：目的：從一堆 x, y 對應數據，找出迴歸線（歸納找出趨勢）

#方法：建立深度學習模型，來找出 x, y 的對應關係式

(1) 步驟1：載入AI函式庫：`import tensorflow library (TensorFlow and keras)`

(2) 步驟2：建立神經網絡模型，單層神經，一個神經元`units=1`

(3) 步驟3：建立目標函數，設定求解方法(最少需要設定兩個參數：A. 損失函數`loss`, B. 優化函數`optimizer`)

(4) 步驟4：設定輸入數據 (x, y) ,

(5) 步驟5：訓練模型：

(6) 步驟6：預測

1.AI的學習內容

- (1) 常識, 知識
- (2) 數學模型
- (3) AI程式

1.AI的數學模型

- (1) 機器學習, 深度學習的流程: 5步驟

- A.建立模型
- B.建立目標函數(就是:誤差函數, **loss function**)
- C.訓練模型
- D.測試模型, 避免**overfitting**(過度擬合)
- D.模型的預測

- (2)機器學習, 深度學習的用途:

- A.分類 **classification**
- B.迴歸 **Regression**

- (3)AI練習網站:

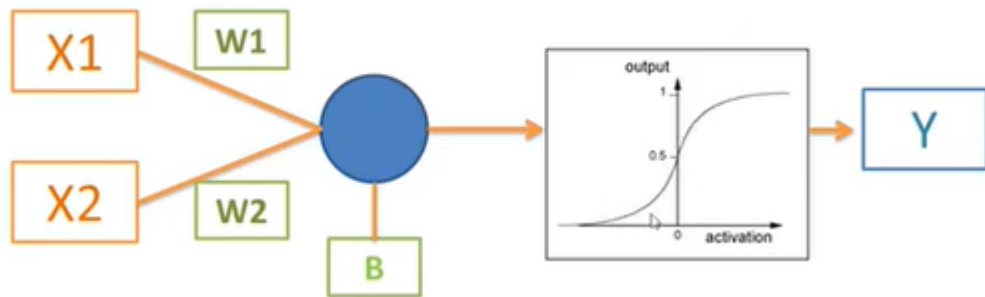
AI練習網站: 數據分類的線上雲端神經網絡

- (4)一顆神經元**neuron**的數學模型:

數學模型: $y = \text{Sigmoid}(x_1 * w_1 + x_2 * w_2 + b)$

w: 權重值, **weight**

b: 偏差值, **bias**



$$Y = X1 * W1 + X2 * W2 + B$$

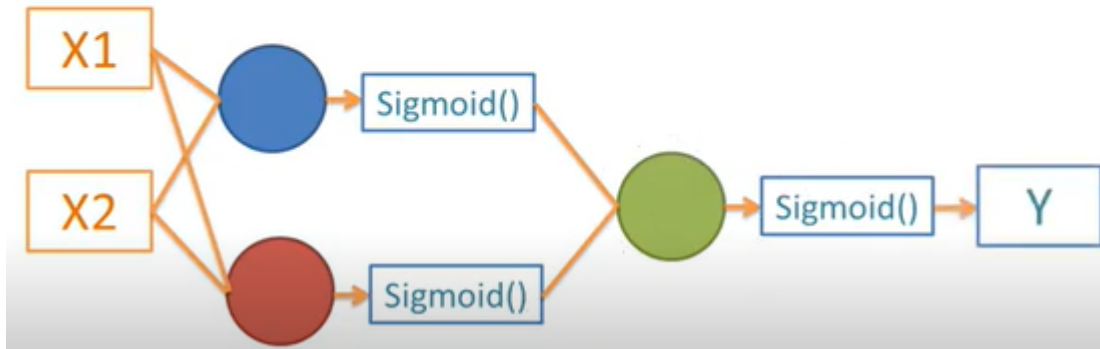
$$Y = \text{Sigmoid}(X1 * W1 + X2 * W2 + B)$$

(5) 多顆神經元(神經網路)的數學模型:

數學模型:

$$\begin{aligned} y = & \text{Sigmoid}(\\ & x11 * w11 + x21 * w21 + b1 + \\ & x12 * w12 + x22 * w22 + b2 + \\ & x13 * w13 + x23 * w23 + b3 + \\ & \dots\dots\dots \\ &) \end{aligned}$$

解決方法



(6)深度學習的核心內容：

☞深度學習，學什麼？

⇒學神經元之間號的**權重值weight**

⇒看權重值，就可以知道誰是主要的影響訊號

☞AI，得權重值者，得天下

二, C++

1. 基本

(1) 輸出

```
//cout<<"hi...";
```

(2) 換行 \n

```
newline
```

(3) 輸入

```
cin>>a;
```

(4) 變數

注意：變數使用前要先宣告

```
int a1;
```

```
a1 = 100;
```

```
float a3;
```

```
a3 = 173.5;
```

```
double a3;
```

```
a3 = 173.5;
```

```
string a2;
```

```
a2 ="john";
```

3. 選擇結構if

(1) 雙向選擇：判讀分數a，及格，或不及格？

```
if (a>=60)
{
    echo<<及格<<endl;
}
else
{
    echo<<不及格<<endl;
}
```

(3) 多向選擇：判斷分數a，是優等 (a>85)，或中等，或不及格 (a<60)

```
a = int(input('輸入a = '))
if(a>=85)
{
    echo<<優等<<endl;
}
else if(a>=70 and a<85)
{
    echo<<中等<<endl;
}
else
{
    echo<<不及格<<endl;
}
```

4. 迴圈結構for, while

(1) for迴圈

```
for(int i=1;i<=10, i++)  
{  
    cout<<i<<"，恭喜發財"<<endl;  
}
```

(2) while迴圈

```
int i =1;  
while(i<=10)  
{  
    cout<<i<<"，恭喜發財"<<endl;  
}
```


遊戲三大引擎 (在台灣就業市場的前三大)

1.unreal

- (1) 若用程式語言開發: `c++`
- (2) 用非程式開發工具: `blueprint` 藍圖

2.Unity

- (1) 若用程式語言開發: `c#`
- (2) 用非程式開發工具: `bolt, playmaker`

3.Cocos Creator

- (1) 若用程式語言開發: `javascript`
- (2) 用非程式開發工具:

元宇宙需要的工具

(VR虛擬實境, AR擴增實境, MR混合實境)

1. 3D建模, 3D動畫
2. 2D建模, 3D動畫
3. 平面美工
4. 程式 : C#
5. 遊戲引擎 : unity + Vuforia
unreal+ Vuforia

結論 : C#程式, unity遊戲引擎, 這兩個很重要

```
-----  
--  
-- main.lua  
--  
-----  
-----  
  
-- Your code here
```

```
a1 = display.newImageRect("world.jpg", 320, 480);  
a1.x = 160;  
a1.y = 240;
```

```
b1 = display.newImageRect("Icon-60.png", 60, 60);  
b1.x = 0;  
b1.y = 0;
```

```
b2 = display.newImageRect("Icon-60.png", 60, 60);  
b2.x = 320;  
b2.y = 0;
```

```
b3 = display.newImageRect("Icon-60.png", 60, 60);  
b3.x = 0;  
b3.y = 480;
```

```
b4 = display.newImageRect("Icon-60.png", 60, 60);  
b4.x = 320;  
b4.y = 480;
```

```
display.newText("新年快樂",160, 240, native.systemFont, 75);
```