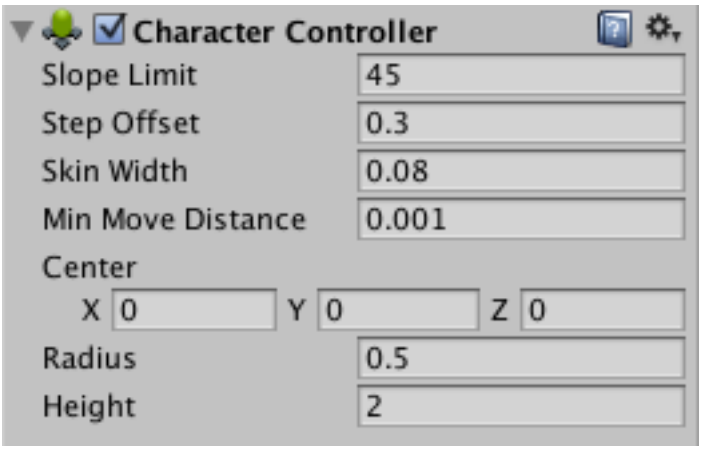
Unity中的角色控制器

角色控制器主要用于第三人称或第一人称游戏主角控制，并不使用刚体物理效果。

它允许你在受制于碰撞的情况下很容易的进行运动，而不用处理刚体。角色控制器不受力的影响，仅仅当你调用Move函数时才运动。然后他将执行运动，但是受制于碰撞。

Unity3D封装了一个非常好用的组件来实现第一人称视角与第三人称视角游戏开发，我们称他为角色控制器组件，几乎不用写一行代码就可以完成一切的操作---- Character Controller（角色控制器)。给游戏角色添加该组件的方式是Add Component->Physics->Character Controller。如下所示。



其中Slope Limit表示角色控制器最大能爬的坡度，比如上图的45，表示该控制器只能爬到45°以下的坡。Step Offset表示以米为单位的角色控制器的台阶偏移量，Skin Width表示角色控制器的碰撞器皮肤厚度，如果发现游戏中角色老是卡住，那么一般调节这个就可以(其大小在控制器宽度的十分之一最佳)。Min Move Distance表示最小移动距离，当小于该值，角色不会动。Center表示类似胶囊的角色控制器的中心，Radius表示其半径，Height表示其高度值。

除了以上几个变量以外，在脚本中还可以获取到角色控制器对象，然后用角本控制挂载有该组件的对象。

看如下代码。

```
using UnityEngine;
using System.Collections;
public class NewBehaviourScript : MonoBehaviour {
	public CharacterController c;
	public EasyJoystick joy;
	private Vector3 moveDirection = Vector3.zero;
	public float moveSpeed = 5.0f;
	public float pushPower = 2.0F;

	void Update () {
		float x = joy.JoystickTouch.x > 0 ? 1:-1;
		float z = joy.JoystickTouch.y > 0 ? 1:-1;
		x = joy.JoystickTouch.x == 0?0:x;
		z = joy.JoystickTouch.y == 0?0:z;

		moveDirection = new Vector3(0,0,z);
		moveDirection =
transform.TransformDirection(moveDirection);
		CollisionFlags flgs = c.Move(moveDirection *
moveSpeed*Time.deltaTime);//根据flgs的值可以判断角色控制是否碰触地面也可以用下面的c.isGrounded来判断。
//		Debug.Log(c.isGrounded);
//		Debug.Log((flgs & CollisionFlags.Below)!=0);
	}
```

```
    void OnControllerColliderHit(ControllerColliderHit hit) {
        Debug.Log("Hit");
        Rigidbody body = hit.collider.attachedRigidbody;
        if (body == null || body.isKinematic)
            return;
        if (hit.moveDirection.y < -0.3F)
            return;
        Vector3 pushDir = new Vector3(hit.moveDirection.x, 0,
hit.moveDirection.z);
        body.velocity = pushDir * pushPower;
    }
}
```

其中OnControllerColliderHit，表示当角色控制器在执行移动(Move)的时候触碰到其他物体时，OnControllerColliderHit被调用。