一、梳理动画机逻辑

1. 用空动画
2. 子动画机

二、动画帧相关问题

电视、电视广播的

NTSC 欧美 30帧

PAL 亚 25帧

将MAX里面的动画烘培成每一帧都是关键帧的导出，导入到Unity时，Unity会进行优化

Rotation Error

Position Error调整 ，值越小越精确，越大动画大小越小

在Unity动画眼里，只有关键帧和曲线

三、

同一个过渡动画拥有的多条件是并且的关系

从同一个动画片段引出的多条过渡动画拥有同样的条件判断，按优先级， Solo只走这条 Mute禁止

过渡被打断（一般会配合Ordered Interrupt使用，涉及到排序）：

|  |  |
| --- | --- |
| Current State | 能被当前状态打断 |
| Next State | 能被下一个过渡动画打断 |
| Current State Then Next State | 先当前后下一个 |
| Next State Then Current State | 先下一个后当前 |

新导入新的场景：桥的模型

为桥添加Collider：桥体地面、两面侧面空气墙

复制新的桥体，V键可以使它们完美的贴合在一起

下一个场景生成的碰撞点Collider

UnityChan的Collider，打上Player标签

为下一个场景生成的碰撞点Collider添加脚本

|  |  |
| --- | --- |
| Door.cs | pubic GameObject way;    void OnTriggerEnter(Collider coll)  {  Instantiate(way, new Vector3(transform.parent.position.x + 150, 0, 0), Quaternion.identity);  Destory(gameObject); //避免重复触发  } |

如果摄像机在角色背后的，一般会：

添加雾效：Lighting-

天空盒不要背景，调成深褐色

|  |  |
| --- | --- |
| LevelManager.cs | using System.Collections.Generic;  .......    public Transform player;  static public LevelManager lm;  public float wayDis = 0;  public GameObject[] levelWayList; //道路库    public List<GameObject> wayList; //当前道路列表    void Awake()  {  lm = this;  }    public void MakeWay()  {  int r = Random.Range(0, levelWayList.Length);  GameObject thisWay = levelWayList[r];  float thisWayDis = thisWay.GetComponent<Bridge>().myDis;  GameObject newWay = Instantiate(thisWay, new Vector3(wayDis, 0, 0), Quaternion.identity);  wayList.Add(newWay);  wayDis += thisWayDis;  } |

（打断：讲程序贴图）

|  |  |
| --- | --- |
| Bridge.cs | Transform player;  public float myDis = 100; //本段路的长度    void Start(  {  player = LevelManager.lm.player;  }    void Update()  {  //float dis = Vector3.Distance(player.position, transform.rotation);  float dis = player.position.x - transform.position.x; //这样有一个好处就是，往回走的时候，不会删除前面的场景  if(dis>150)  Destroy(gameObject);  } |