软件设计

Unity3d游戏制作

Unity3D是由Unity Technologies开发的一个让玩家轻松创建诸如[三维](https://baike.baidu.com/item/%E4%B8%89%E7%BB%B4)视频游戏、建筑可视化、实时[三维动画](https://baike.baidu.com/item/%E4%B8%89%E7%BB%B4%E5%8A%A8%E7%94%BB/1575383)等类型互动内容的多平台的综合型游戏开发工具，是一个全面整合的专业[游戏引擎](https://baike.baidu.com/item/%E6%B8%B8%E6%88%8F%E5%BC%95%E6%93%8E)。在本课题中，我们使用unity3d程序进行患者步行运动配套的游戏制作。在游戏中建立模拟场景和模拟患者的小人，小人根据患者腿部运动的频率和步幅同步前进。患者在游戏的激励下将会对康复训练感到兴趣，康复训练不再是疲倦的机械重复操作。成品效果图如图1所示。



在unity3d制作中，我们使用预制件制作出模拟人形和背景；使用自己编写的脚本文件控制人形运动，形成走路姿态；摄像机通过脚本和人形同步移动，形成第三人视角，真实舒适；在游戏中添加pickups，在人形走过时能够吃掉pickups得分，增添游戏乐趣；通过material设计，增添游戏暖色彩，给患者良好感受；利用physics物理世界模块，仿真出真实世界的物理规律，使游戏更加真实有趣。同时在游戏中加入计步，测距等可视化功能，便于控制和读取信息；最终的结束奖励机制更会给患者带来完成康复的成就感，带给患者康复的信心，生活的乐趣和希望，这也是患者迫切需要的。下面介绍本次游戏制作的关键与难点。

1. 人形的运动

在创建了最基础的人形和背景后，本部分主要是让人形运动起来。这套游戏做到先在电脑中输入模式，再把患者的腿放置在机器上后，人形立即随着电机开始运动。人形的运动基于模式的选择和压力传感器的数据，这需要script脚本文件来实现。在脚本代码中，我们运用到了addforce函数来实现人形的移动；使用if语句实现压力传感器的判断；定义浮点与三维向量来实现精确控制；运用rigidbody来让人形具有物理性质，例如保持人形走在地面上。具体的代码流程如图2所示。代码如下，attach到人形上。

初始化

设置模式

压力传感器数值是否大于0

人形不动

人形按照模式向前移动直到终点

结束

图2 人形运动流程图

|  |
| --- |
| using UnityEngine;  using UnityEngine.UI;  using System.Collections;  public class PlayerController : MonoBehaviour  {  public float speed;  public Text countText;  public Text winText;  private Rigidbody rb;  private int count;  void Start()  {  rb = GetComponent<Rigidbody>();  count = 0;  SetCountText();  winText.text = "";  }  void FixedUpdate()  {  float moveH = Input.GetAxis("Horizontal");  float moveV = Input.GetAxis("Vertical");  Vector3 now = new Vector3(moveH, 0.0f, moveV);  rb.AddForce(now);  }  void OnTriggerEnter(Collider other)  {  if (other.gameObject.CompareTag("Pick Up"))  {  other.gameObject.SetActive(false);  count = count + 1;  SetCountText();  }  }  void SetCountText()  {  countText.text = "Count: " + count.ToString();  if (count >= 3)  {  winText.text = "You Win!";  }  }  } |

1. Camera的移动

Camera在游戏中必须跟随人物移动，否则游戏界面上只有一个人渐行渐远，而不是人物持续地行走。这样不会有良好的游戏体验，不会给患者代入感。较为简单的方法是使相机作为人形的子部件，相机就会跟随人形移动，但是相机有可能上下翻滚，毫无游戏体验。为了使相机跟随人移动且上下翻滚的现象，我们编写了具体的代码如下，attach到camera上。

|  |
| --- |
| using UnityEngine;  using System.Collections;  public class CameraController : MonoBehaviour  {  public GameObject player;  private Vector3 offset;  void Start()  {  offset = transform.position - player.transform.position;  }  void LateUpdate()  {  transform.position = player.transform.position + offset;  }  } |

1. pickup与计分的制作与应用

在游戏中，为了增加游戏体验，我们设计了奖励机制，即pickup。在人形经过pickup的时候将会吃掉pickup并为之计分。在本游戏中，pickup是一个小正方体。为了不让pickup因为有物理性质（即重力）而掉落，我们对它关闭了use gravity功能，使它保持悬浮在空中。当人形碰到pickup时，我们让pickup消失。这使用到了trigger触发功能。在人的认知中，动态的，有色彩的，与背景不同的东西更易引人关注。Pickup需要引起患者的关注，所以我们使用material给pickup增添暖色彩，编辑rotate角度使它处于45度角的位置，并编写代码使它旋转，营造动态效果。为了动态旋转我们运用到了rotate函数，代码如下，attach到pickup上。

|  |
| --- |
| using UnityEngine;  using System.Collections;  public class Rotator : MonoBehaviour  {  void Update()  {  transform.Rotate(new Vector3(15, 30, 45) \* Time.deltaTime);  }  } |

游戏有计分和奖励机制才能带来乐趣，所以我们创立计分和结束奖励，使用到了unity中的text文本功能。我们用count = count + 1代码实现连续计分，使用if (count >= 3)代码进行结束判断，使用winText.text = "You Win!"代码实现结束奖励，让患者在游戏中找到乐趣，康复也就不再那么枯燥。代码在“人形的运动”中已经展示，attach到人形上。