

《人机交互原理与应用》实验报告

班级	信息十班	实验日期	2022.11.3	实验成绩	
姓名	陈澄	学号	32420212202930		
实验名称	人机交互原理与应用第五次实验				
实验目的、要求	1、应用 unity 实现多线程运行，并考虑巨量计算网络通讯和资源加载等因素； 2、应用 unity 实现数据持久化并在运行时实现文件，资源的读取。				
实验内容、步骤及结果	1. 实现一个登录界面 A，包含： <ul style="list-style-type: none">• 用户名和密码输入• 登录和退出按钮 点击登录按钮后，将获取的登录时间、用户名和密码保存下来，供之后的场景使用。 2.中间场景 B 包含一个随时间增长进度条。 <ul style="list-style-type: none">• 处理 CSV 文件• 至少显示 2 秒 在进入最终场景 C 前显示该场景 3. 场景 C 中显示以下内容 <ul style="list-style-type: none">• 用户名、密码、登录时间• CSV 文件中日期为你的出生日期的行				

	<p>(不再读取 CSV 文件)</p> <ul style="list-style-type: none">• 将结果装入 Scroll View
回 达 相 关 问 题	<p>1.如何保存将要在场景 C 中显示的信息（登录时间、用户名、密码、CSV 文件特定行）</p> <pre>void Start() { //创建表 设置表名 DataTable dt = new DataTable("Sheet1"); //创建列 有三列 dt.Columns.Add("账号"); dt.Columns.Add("密码"); dt.Columns.Add("登录时间"); //创建行 每一行有三列数据 DataRow dr = dt.NewRow(); dr["账号"] = PlayerPrefs.GetString("zhanghao"); dr["密码"] = PlayerPrefs.GetString("mima"); dr["登录时间"] = (System.DateTime.Now.Year+"/"+System.DateTime.Now.Month+"/"+System.DateTime.Now.Day); dt.Rows.Add(dr); DataRow dr2 = dt.NewRow(); dr2["登录时间"] = ("2004/05/19"); dt.Rows.Add(dr2); string filePath = Application.streamingAssetsPath + "\\账号密码储存.csv"; SaveCSV(filePath, dt); }</pre> <p>(创建 datatable 表)</p>

```

// 将DataTable中数据写入到CSV文件中
1 reference
public static void SaveCSV(string filePath, DataTable dt)
{
    FileInfo fi = new FileInfo(filePath);
    if (!fi.Directory.Exists)
    {
        fi.Directory.Create();
    }
    using (FileStream fs = new FileStream(filePath, FileMode.Create, FileAccess.Write))
    {
        using (StreamWriter sw = new StreamWriter(fs, System.Text.Encoding.UTF8))
        {
            string data = "";
            //写入表头
            for (int i = 0; i < dt.Columns.Count; i++)
            {
                data += dt.Columns[i].ColumnName.ToString();
                if (i < dt.Columns.Count - 1)
                {
                    data += ",";
                }
            }
            sw.WriteLine(data);
            //写入每一行每一列的数据
            for (int i = 0; i < dt.Rows.Count; i++)
            {
                data = "";
                for (int j = 0; j < dt.Columns.Count; j++)
                {
                    string str = dt.Rows[i][j].ToString();
                    data += str;
                    if (j < dt.Columns.Count - 1)
                    {
                        data += ",";
                    }
                }
                sw.WriteLine(data);
            }
            sw.Close();
            fs.Close();
        }
    }
}

```

(按顺序写入 csv 文档)

答：构建 DataTable 数据将其创建为 csv 文档。

2.如何实现异步场景加载

```

7 public class 加载 : MonoBehaviour
8 {
9     // Start is called before the first frame update
10    3 references
    public Slider slider;
11    0 references
    void Start()
12    {
13        slider.value=0;
14    }
15
16    // Update is called once per frame
17
18    0 references
    void Update()
19    {
20        slider.value+=Time.deltaTime*0.5f;
21        if(slider.value==1&&SceneManager.LoadSceneAsync(2).progress==1)
22        {
23            SceneManager.LoadSceneAsync(2);
24        }
25    }
26 }

```

答：使用 SceneManager.LoadSceneAsync 方法异步加载，并用 SceneManager.LoadSceneAsync().progress 函数判断是否已完成加载。

3.如何读取 CSV 文件得出想要的结果

```
0 references
public class csv读取 : MonoBehaviour
{
    1 reference
    public Text zhanghao;
    1 reference
    public Text mima;
    1 reference
    public Text time;
    1 reference
    public Text borntime;
    0 references
    void Start()
    {
        string filePath = Application.streamingAssetsPath + "\\账号密码储存.csv";
        DataTable dt = OpenCSV(filePath);
        zhanghao.text="账号="+dt.Rows[0][0];
        mima.text="密码="+dt.Rows[0][1];
        time.text="登录时间="+dt.Rows[0][2];
        borntime.text="我的出生日期"+dt.Rows[1][2];
    }
}
```

```
1 reference
public static DataTable OpenCSV(string filePath)//从csv读取数据返回table
{
    DataTable dt = new DataTable();
    using (FileStream fs = new FileStream(filePath, FileMode.Open, FileAccess.Read))
    {
        using (StreamReader sr = new StreamReader(fs, Encoding.UTF8))
        {
            //记录每次读取的一行记录
            string strLine = "";
            //记录每行记录中的各字段内容
            string[] aryLine = null;
            string[] tableHead = null;
            //标示列数
            int columnCount = 0;
            //标示是否是读取的第一行
            bool IsFirst = true;
            //逐行读取csv中的数据
            while ((strLine = sr.ReadLine()) != null)
            {
                if (IsFirst == true)
                {
                    tableHead = strLine.Split(',');
                    IsFirst = false;
                    columnCount = tableHead.Length;
                    //创建列
                    for (int i = 0; i < columnCount; i++)
                    {
                        DataColumn dc = new DataColumn(tableHead[i]);
                        dt.Columns.Add(dc);
                    }
                }
                else
                {
                    aryLine = strLine.Split(',');
                    DataRow dr = dt.NewRow();
                    for (int j = 0; j < columnCount; j++)
                    {
                        dr[j] = aryLine[j];
                    }
                    dt.Rows.Add(dr);
                }
            }
        }
    }
}
```

```
1
{
    if (IsFirst == true)
    {
        tableHead = strLine.Split(',');
        IsFirst = false;
        columnCount = tableHead.Length;
        //创建列
        for (int i = 0; i < columnCount; i++)
        {
            DataColumn dc = new DataColumn(tableHead[i]);
            dt.Columns.Add(dc);
        }
    }
    else
    {
        aryLine = strLine.Split(',');
        DataRow dr = dt.NewRow();
        for (int j = 0; j < columnCount; j++)
        {
            dr[j] = aryLine[j];
        }
        dt.Rows.Add(dr);
    }
}
}
```

	答：将 csv 文件数据按顺序读取到 datatable 再读取 datatable 表。
--	---