

Unity3D 与数据库通信方法的研究

张利利¹, 李仁义², 李晓京¹, 马 进¹, 惠铎铎¹

(1. 第四军医大学 航空航天医学教育部重点实验室, 陕西 西安 710032;
2. 西安热工研究院有限公司, 陕西 西安 710032)

摘 要: 基于 Unity3D 实现系统与两种关系型数据库之间的连接。随着虚拟仿真技术在各领域的广泛应用, 虚拟仿真过程中海量数据的实时性已经成为当前研究的热点。文中主要研究基于 Unity3D 的 Monodevelop 编辑器下, 采用 C# 编程语言实现系统与数据库的连接, 并完成海量数据的传输和存储, 从而达到系统与数据库的双向交互。文中给出了具体的代码实现, 并结合实际开发过程中遇到的问题给出了具体可行的解决方法。实践结果表明, 通过这种方法连接数据库具有简单、高效等特点。

关键词: Unity3D; Access; SQL Server2000

中图分类号: TP301

文献标识码: A

文章编号: 1673-629X(2014)03-0229-03

doi: 10.3969/j.issn.1673-629X.2014.03.057

Research on Communication Method between Unity3D and Database

ZHANG Li-li¹, LI Ren-yi², LI Xiao-jing¹, MA Jin¹, HUI Duo-duo¹

(1. Key Laboratory of Aerospace Medical of Ministry of Education, Fourth Military Medical University, Xi'an 710032, China;
2. Xi'an Thermal Power Research Institute Co., Ltd., Xi'an 710032, China)

Abstract: The connection between system and two relational databases based on Unity3D is implemented. With the wide application of virtual simulation technology, the real-time of mass data in the virtual simulation has been the hotspot in this field. Mainly research the connection of system and database by C# language, based on Monodevelop of Unity3D, completing the transmission and storage of mass data, consequently reaching bidirectional alternation. It also shows the detailed code realization and gives the concrete solution with the practical process. The result shows this is simple and efficient.

Key words: Unity3D; Access; SQL Server2000

0 引言

随着计算机硬件及软件的不断发展, 虚拟仿真技术已经渗透到了各个行业及领域^[1]。虚拟仿真旨在通过计算机为用户提供最为真实的虚拟环境。用于研发虚拟平台的技术很多, 这两年 Unity3D 由于其价格更为便宜^[2], 并支持多种系统平台, 受到越来越多业界人士的追捧^[3]。文中研究以团体测试系统为研究对象, 借助 Unity3D 平台开发了一个团体测试系统。即为测试系统, 就避免不了要将测试的数据存储在数据库中方便以后统计^[4]。文中针对 Unity3D 在连接数据库过程中出现的一些问题提出了具体可行的解决方案, 并且将其与实际的应用相结合, 为进一步的研究

提供了依据。

1 系统简介

1.1 Unity3D 简介

Unity3D 是由丹麦 Unity 公司开发的一款跨平台的创新思维的图形化^[5]三维引擎, 能够提供如渲染引擎、无力引擎、音乐音效、脚本引擎及场景管理^[6]等基本功能。Unity3D 具有方便的可视化创作环境^[7-8], 并支持各种脚本语言包括 C#、JavaScript^[9], 兼容各种操作系统, 真正地实现了跨平台, 让开发者能简单直观地开发基于各种平台三维硬件设备的应用程序^[10]。Unity3D 默认的脚本编辑器是 UnisciTE, 它用起来特别

轻巧快捷,就像记事本一样,但是当开发一些大型项目时,显然 UnisciTE 简单的功能就不足以支撑它的工作了,所以文中选用 Monodevelop 和 C#作为系统的开发平台和编程语言。

1.2 系统结构

该系统分为三部分,测试的服务器端、客户端和数据库。其中服务器端和客户端都是基于 Unity3D 平台开发的。服务器主要实现与数据库的交互,包括管理员登录、系统初始化、测试者信息的录入、给客户端下发测试任务、保存测试所得的原始数据以及计算和保存测试的结果数据。客户端通过与服务器通信得到自己的编号,通过该编号与姓名就可以登录测试系统进行相应的测试,并将测试的原始数据发送给服务器。

系统模式为一台服务器控制 3 个客户端。所有客户端的测试情况都实时地显示在服务器端,同时各个客户端也可以根据系统需要选择可见或者不可见其他客户端的测试情况。为了实现数据的同步,服务器和客户端必须进行数据的实时通信,并将数据保存在数据库中,便于以后对数据进行相应的统计分析和回放。其结构如图 1 所示。

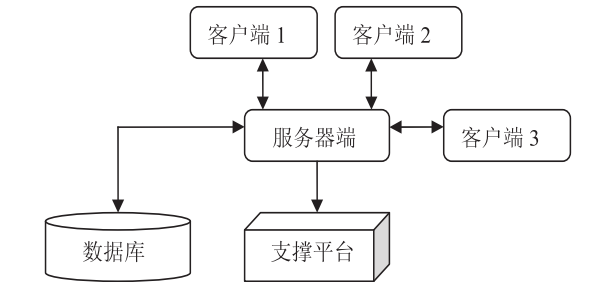


图 1 系统结构示意图

2 数据库原理与设计

2.1 数据库工作原理

系统采用软件驱动来实现数据库的调用。在使用数据时,需要有一个数据库服务器在运行,然后应用程序就可以通过软件编程的方式实现系统与数据库的连接,以及对数据库进行读写操作。

2.2 数据库设计

系统的数据库主要有以下 5 个表,管理员表(admin)、测试人员表(user)、任务表(task)、原始数据表(data)和结果表(result)。

管理员表主要记录操作服务器端的管理者,其中 ID 为关键字,唯一标识一位管理员。管理员可以通过 ID 和密码登录服务器端。结构如表 1 所示。

表 1 管理人员表

ID	Name	Password
自动编号	文本	文本

测试人员表主要记录测试的人员信息, ID 为关键字,唯一标识一位测试人员。测试人员可以通过 ID 和 Name 登录测试系统。结构如表 2 所示。

表 2 测试人员表

ID	Name	Sex	Year	Edu	Dep
自动编号	文本	文本	文本	文本	文本

任务表主要记录管理员给哪些测试人员分配了哪些测试任务。测试批次、测试人员 ID 和测试任务 ID 同为关键字,唯一标识测试人员的一次测试。结构如表 3 所示。

表 3 任务表

测试批次	管理员 ID	测试人员 ID	测试任务 ID
文本	数字	数字	数字

原始数据表记录每个批次,测试人员进行测试任务时所得的所有原始数据,在该系统中,设计的有 47 个字段。

结果表记录了每个批次、每个测试人员进行测试所得的结果数据,管理员可以快速直观地看出测试者的测试成绩。

由于篇幅有限,就不列出原始数据表和结果表的结构。

2.3 代码实现

①连接 ACCESS 数据库。

此段代码以 ACCESS 2003 数据库为例实现 Unity3D 与数据库的通信,如果选用不同版本的 ACCESS 数据库,那么对数据库的后缀名要根据需要进行修改。

```
using System. Data;
using System. Data. Odbc;
public class ACCESSREADER : MonoBehaviour
{
    void OnGUI( )
    {
        if( GUILayout. Button( "增加测试人员信息" ) )
        {
            InsertInto( Application. dataPath + "/teamdata. mdb" );
        }
        void InsertInto( string fileto read)
        {
            string con = " Driver = { Microsoft Access Driver ( *. mdb,
            *. accdb) } ; DBQ = "+fileto read;
            string query = " ";
            string[ ] values = { "王平", "男", "1858", "大学", "学员队" };
            query = " INSERT INTO user( Name, Sex, Year, Edu, Dep)
VALUES( "+ values[ 0];
            for ( int i = 1; i < values. Length; ++i)
```

```

{
query += ", " + values[i];
}
query += ")";
OdbcConnection oCon = new OdbcConnection(con);
OdbcCommand oCmd = new OdbcCommand(query, oCon);
oCon.Open();
oCmd.ExecuteNonQuery();
oCon.Close();
}
}

```

连接成功之后添加数据并显示数据的界面如图2所示。



图2 增加测试人员信息并显示的界面

② 连接 SQL Server 数据库。

此段代码以 SQL Server2000 数据库为例实现 Unity3D 与数据库的通信。

```

using System.Data;
using System.Data.SqlClient;
public class datatest : MonoBehaviour
{
void Start ()
{
SqlConnection sqlcon=new SqlConnection("data source=(local);uid=sa;pwd=;database=teamdata");//连接 teamdata 数据库的字符串
SqlCommand cmd=new SqlCommand();
cmd.Connection=sqlcon;
cmd.CommandType=System.Data.CommandType.Text;
sqlcon.Open(); //打开连接
string query="";
string[] values={ "王平","男","1858","大学","学员队"};
query="INSERT INTO user( Name,Sex,Year,Edu,Dep)VALUES("+values[0];
for(int i=1;i<values.Length;++i)
{
query += ", " + values[i];
}
query += ")";

```

```

cmd.CommandText=query;
cmd.ExecuteNonQuery(); //向 user 表中插入一条记录
sqlcon.Close(); //关闭连接
}
}

```

3 数据库性能评价

在一般数据库中,评价性能的好坏主要参考 SQL 操作执行的时间消耗,SQL 的执行时间消耗又由以下几个因素决定^[11]:SQL 执行时间消耗=网络传输消耗+CPU 执行消耗+存储器读写消耗。网络传输消耗主要指 SQL 操作从客户端经过网络传输到服务器端所消耗的时间,存储器读写消耗是指从获取数据到将数据存储到数据库以及从数据库读取数据所消耗的时间。

由于该系统是基于局域网内部,而且服务器端和数据库在同一台计算机上,计算机也都是最新配置,所以可以忽略网络传输消耗和 CPU 执行消耗,而只考虑存储器读写消耗。

系统一次完整的测试时间为 3 min,为了使测试界面流畅,借鉴了电视机的数据传输速率,即 25 帧/s^[12],那就意味着系统在一次完整的测试过程中必须存储 4 500 条数据。经过两种方法对数据库的性能进行测试。

第一种方法:为了保障数据库的安全性,每插入一条记录就打开和关闭一次数据库,经过测试插入一条记录大概需要 60 ms,因为每 40 ms 需要传输一条记录,显然这种方法不可取。

第二种方法:打开一次数据库,等所有的记录都插入之后再关闭数据库,那么插入一条记录大概需要 2 ms,速度大大提高。

4 需要注意的问题

(1) 配置文件:要连接数据库,需要引用 unity 安装目录下的一个动态链接库,即. \ Unity \ Editor \ Data \ MonoBleedingEdge \ lib \ mono \ 2.0 目录下的 System.Data.Dll,即将 System.Data.Dll 文件拷贝到当前项目文件的 Assets 文件夹中即可,然后在程序中添加引用 using System.Data。虽然这样可以成功连接,但是发布为 web 或者 exe 后可能会出现连接不上数据库的情况,这就需要另外三个动态链接库文件 I18N.dll, I18N.West.dll, I18N.CJK.dll,并将此 3 个文件与 System.Data.dll 放在同一个目录下。

(2) 中文显示乱码:该系统选用 Mono develop 编辑器下的 C#作为开发语言,由于 C#会默认的将文件保存为 UTF-8 格式,该格式对中文会出现乱码,所以

必须用记事本将脚本打开,再另存为 Unicode 格式即可显示中文。

(3)打补丁:如果需要连接 Access 数据库,则必须将数据库存放在工程目录的 Assert 目录下。但是如果系统连接 SQL Server2000 的数据库,连接时会出现“由于目标主机积极拒绝连接”错误,这是因为 SQL Server 的 SP4 以下的版本不允许外连,所以必须打 SP4 的补丁。

(4)系统发布:当需要将系统编译为可执行程序时,会提示错误“Extracting referenced dlls failed”,这就需要在 Building settings 的 Player settings 面板中找到对应的发布平台,然后找到 Api Compatibility level 下拉菜单,将默认的设置由 .Net 2.0 subset 改为 .Net 2.0,然后再次发布就可以成功生成 exe 文件。

5 结束语

通过各种方法实现 Unity3D 与数据库的连接实践表明,在 unity 中利用 System. Data. dll 中提供的方法实现与数据库的连接比较简单实用。文中系统所用的范围仅限于单位的局域网内,暂时没有考虑数据库安全问题。

如果开发基于 B/S 模式的系统,数据库的安全就显得十分重要。数据库安全是数据库的生命,作为数据库安全性管理者,每个应用程序首先要保证数据库的安全。

参考文献:

- [1] Goldstone W. Unity game development essential[M]. Birmingham: Packt Publishing Ltd., 2009.
- [2] 朱惠娟. 基于 Unity3D 的虚拟漫游系统[J]. 计算机系统应用, 2012, 21(10): 36-39.
- [3] 方 凯. Unity3D 第一人称视角游戏者水下视觉效果的实现[J]. 计算机光盘软件与应用, 2012(15): 193-194.
- [4] 吴松涛, 龚家伟. 在 LabVIEW 中利用 LabSQL 实现数据库访问[J]. 国外电子测量技术, 2006, 25(4): 53-56.
- [5] UNITY: Game development tool[EB/OL]. 2012. <http://Unity3D.com/>.
- [6] Kenneth C. Finney 3D 游戏开发大全[M]. 齐兰博, 肖 奕, 译. 北京: 清华大学出版社, 2005.
- [7] 贺 杰, 郭 慧, 龚 平. 应用 Unity3D 引擎实现旅游景点 WEB 虚拟漫游的方法[J]. 技术与市场, 2012, 19(4): 21-22.
- [8] 付永明, 汪云甲. 一个 3D GIS 的设计与实现[J]. 测绘与空间地理信息, 2005, 28(1): 49-51.
- [9] 伍传敏, 张 帅, 邱锦明. 基于 Unity3D 的 FPS 游戏设计与开发[J]. 三明学院学报, 2012, 29(2): 35-40.
- [10] 凌东伟, 李 欣, 朱美正. 基于 GIS 的三维虚拟地球的研究与发现[J]. 计算机工程与设计, 2007, 28(19): 4774-4777.
- [11] 黄德才. 数据库原理及其应用[M]. 第 3 版. 北京: 科学出版社, 2010.
- [12] Yu Yang, Li Zhu, Larry S. Cross-layer optimization for state update in mobile gaming[J]. IEEE transactions on multi-media, 2008, 10(5): 701-710.
- [13] 杨 永. 基于 ASP. NET 和 B/S 模式的试题库管理系统的研究与开发[J]. 微计算机应用, 2008(10): 12-16.
- [14] 侯 虹. B/S 模式职业技术学院教学教务管理信息系统的设计与实现[J]. 中国科技信息, 2006(2): 46-47.
- [15] 康亚娟. 浅谈站内信息搜索系统[J]. 硅谷, 2011(13): 191-191.
- [16] 穆慧琳, 高 岭, 杨建峰, 等. 基于 B/S 模式的网络教学系统的设计与实现[J]. 中国教育信息化·高教职教, 2009(1): 48-51.
- [17] Baldi P. Bioinformatics: The Machine Learning Approach[M]. [s. l.]: MIT Press, 2001.
- [18] 杨红飞, 李振坤, 梅松青, 等. 基于 Struts 和 Hibernate 的教务系统研究与设计[J]. 计算机技术与发展, 2008, 18(2): 202-204.
- [19] Sebastian F. Machine learning in automated text categorization[J]. ACM computing surveys, 2002, 34: 45-47.
- [20] 明日科技. ASP. NET 程序开发范例宝典[M]. 北京: 人民邮电出版社, 2009.
- [21] 蒋 薇. 基于 B/S 架构的“神龙一号”数据库系统的实现[J]. 强激光与粒子束, 2010, 22(3): 642-646.
- [22] Han Jiawei, Kamber M. Data mining: Concepts and techniques[M]. 2nd ed. Beijing: China Machine Press, 2006.
- [23] Dietterich T G. Machine learning for sequential data[D]. Oregon: Oregon State University, 2003.

(上接第 228 页)

作者：

张利利, 李仁义, 李晓京, 马进, 惠铎铎, ZHANG Li-li, LI Ren-yi, LI Xiao-jing, MA Jin, HUI Duo-duo

作者单位：

张利利, 李晓京, 马进, 惠铎铎, ZHANG Li-li, LI Xiao-jing, MA Jin, HUI Duo-duo(第四军医大学 航空航天医学教育部重点实验室, 陕西 西安, 710032), 李仁义, LI Ren-yi(西安热工研究院有限公司, 陕西 西安, 710032)

刊名：

计算机技术与发展

ISTIC

英文刊名：

Computer Technology and Development

年, 卷(期):

2014(3)

本文链接: http://d.wanfangdata.com.cn/Periodical_wjfz201403056.aspx