

实验课 2：仿真环境脚本语言实习

一、实验目的

1. 实习虚拟化身、姿势和动作制作方法
 - 掌握虚拟世界化身体型、皮肤、头发、眼睛制作方法
 - 掌握化身服饰制作操作方法
 - 掌握虚拟世界姿势、动作、手势制作方法
2. 三维虚拟世界脚本语言实习
 - 掌握脚本语言程序结构
 - 掌握脚本语言数据类型和流程控制
 - 熟悉 LSL 库函数
3. 人与虚拟机器人简单对话实验
 - 掌握虚拟世界听感知通道以及会话控制方法
 - 实现简单人机对话

二、实习内容

2.1 三维化身体型和皮肤等操作方法



2.2 化身服饰制作

见 pdf 文档《The Clothes Make the Avatar:Creating Your Own Clothing》，相应模板资源在实验 2 文件夹。

2.3 化身动作、姿势和手势制作

见 pdf 文档《Making a Statement with Poses, Animations, and Gestures》，相应软件工具在实验 2 文件夹。

三、LSL脚本语言

3.1 LSL脚本语言程序结构

林登脚本语言（LSL, Linden Scripting Language），是一种简单易学而又具有强大功能的编程语言。林登脚本语言在语法上类似于 C 语言和 Java。它运用的是事件触发模式，也就是说，事件会引发特定代码的运行。事件句柄是那些对事件进行设置、复位或其他反应的函数命令。而库函数（任何以 ll 开头的单词）会调用和预编译你程序中有用的模块。

LSL 语言类似 Java，通过仿真服务器的虚拟机将脚本编译成可执行的字节码执行。每个脚本在服务器分配的时间单元内执行，当服务器上运行的脚本文件很多的时候，为了不降低服务器本身的性能，服务器会自动地减少分配给每个脚本的时间，因此会引起一些延迟。另外，每一个脚本文件执行时都有自己的内存块，不会与服务器会的内存或其他脚本内存冲突，因此脚本一般不会引起服务器失效。

同一个物体可以添加多个脚本，因此可以将脚本写成一些小的单功能的可重用的代码，以便进行组合完成复杂的行为。

3.1.1 Hello Avatar 示例

- 创建脚本文件：在库存里点击新建脚本
- 写状态（States）
- 调用函数
- 处理事件
- 添加到物体内容里

LSL脚本实质上是一个状态机，而且必须有一个缺省状态 `default {}`，当脚本启动或者重置时首先进入的状态。

```
default
```

```

{
    state_entry()
    {
        llSay(0, "Hello, Avatar!");
    }
    touch_start(integer total_number)
    {
        llSay(0, "Touched.");
    }
}

```

3.1.2 LSL程序结构

```

// 全局变量
vector gStartPosition;

// 全局函数
make_physical_and_spin(vector torque)
{
    // double the torque because we want to
    vector double_torque = 2.0*torque;
    llSetTorque(double_torque, FALSE);
}

// 进入状态代码
// default state - 缺省状态，不需声明，所有脚本必须有一个缺省状态
default
{
    // when a state is transitioned to, the state_entry event is raised
    // 在状态转换的时候首先触发state_entry事件
    state_entry()
    {
        // Make physical so it can move
        llSetStatus(STATUS_PHYSICS, TRUE);
        // get the base stating position
        gStartPosition = llGetPos();
        // Move to 2 meters above the start position
        gStartPosition.z += 2.0;
        llMoveToTarget(gStartPosition, 0.5);
        // transition to spin state
        state SpinState;
    }
    // when a state is transitioned out of, the state_exit event is raised
    state_exit()
    {
        // set critical damping to keep us on our stating location, with

```

```

time constant of 0.5 seconds
    llMoveToTarget(gStartPosition, 0.5);
}
}
// SpinState is a user defined state
state SpinState
{
    state_entry()
    {
        // say hello on channel 0
        llSay(0, "Starting to spin");
        // call global function
        make_physical_and_spin(<0.1, 0.1, 0.0>);
    }
}

```

3.1.3 数据类型

- **整型 integer**
Signed, 32-bit integer value
- **浮点 float**
32-bit floating point value
- **字符串 string**
A sequence of characters
“\” functions as an escape character, so use “\n” to enter a carriage return into a string, “\t” to enter a 4 space tab, “\\” to enter a backslash, “\” to enter a double quote, and “\” followed by any other character to enter that character. Note that the C conventions of \ooo and \xhh are not supported.
- **UUID 类型 key**
在 **secondlife** 里，每个物体都有一个唯一的 **UUID** 代码
- **向量 vector**
用三个浮点数组，用于表达位置坐标、颜色、力等。
- **选装 rotation**
4 个浮点数，用于表达一个旋转量。通过 .x, .y., .z, or .w 来访问
- **链表 list**
string StringVar = "Hello, Carbon Unit";
list MyList = [1234, ZERO_ROTATION, StringVar];
Give this list:
[1234, <0,0,0,1>, "Hello, Carbon Unit"]

3.1.3 类型转换

- Integer to String
- Float to Integer
- ·Float to String
- ·Vector to String
- ·Rotation to String
- ·Integer to List
- ·Float to List
- ·Key to List
- ·String to List
- ·Vector to List
- ·Rotation to List
- ·String to Integer
- ·String to Float
- ·String to Vector
- ·String to Rotation

3.1.4 流程控制

LSL 支持C/java的流程控制语法结构。If else ， for循环， do while循环， while循环， 条件跳转。

运行示例脚本程序。

3.1.5 示例脚本

`change color and size.txt;`

`Changing an object's appearance over time.txt`

`Changing an object's position over time.txt`

四、实验：人-机器人对话

4.1 实验目的

实验简单的自然语言处理，实现与机器人的人机对话。要求实现机器人能随机的回答yes/no的问题。实验要求掌握LSL脚本语言结构，掌握文本处理技巧，掌握LSL听感知函数和说函数调用。

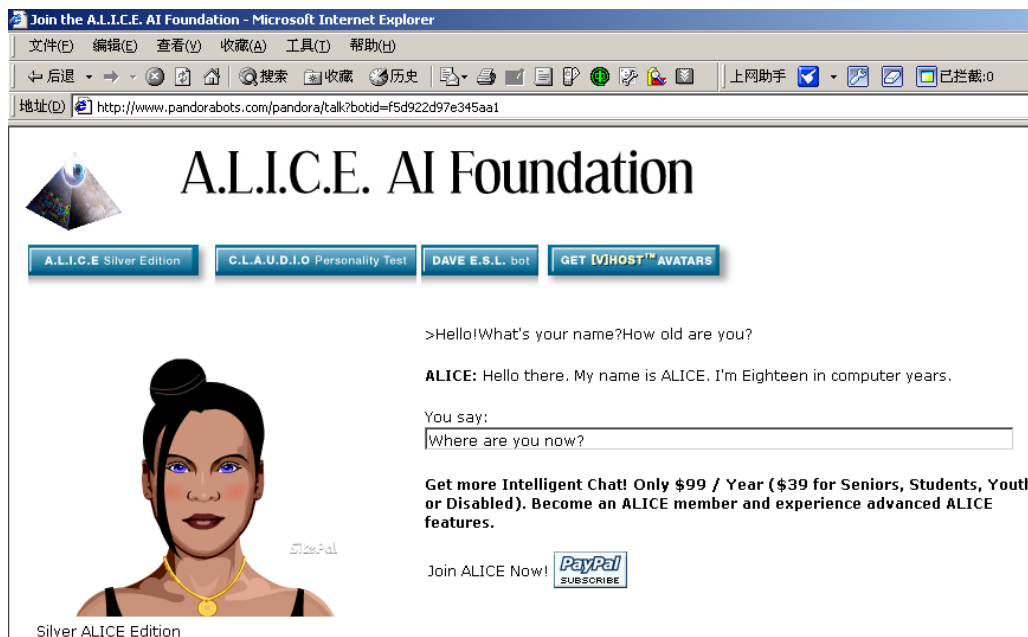
4.2 实验原理

聊天机器人背景：

聊天机器人Verbot是体现人工智能的一种形式，让机器人理解自然语言，并能与人对话。这种类型的机器人叫做VerBot，就是Verbal Robot。这类机器人最早是用来参加图灵测试的，没什么实用价值。现在聊天机器人已经可以实用了，例如把这种机器人放到了网上，就好象商店门口摆了机器人招揽生意一样，在一些

公司的网站上和浏览者聊天，主动介绍产品。有些能Verbot能和网页相互作用，在一个frame里聊天，聊到公司负责人或者产品，就在另外的frame里调出相应的网页。有些把语音合成的功能结合了进来，和你谈的所有话都是“说”出来的。在回答几个关键问题的时候，它们各有不俗的表现。

艾丽丝（ALICE）是目前世界上最好的人工智能程序之一。ALICE的发明者理查德·华莱士教授（Richard S. Wallace）将这个程序设计得跟人类的思维太相似了，以至于很多人把艾丽丝当成了真人。人们在网上与艾丽丝谈论他们的生活和梦想，倾诉自己的隐私，也谈对小甜甜布兰妮的喜爱。



聊天机器人原理

在问到“*How old are you?*”的时候，一个回答“*I am old enough*”，一个回答很实在，被做出来多长时间就说多长时间。还有一个回答“*9 years old*”不知道是怎么算的。

在问到“*are you a male or a female?*”时，一个和形象非常吻合地描述自己的性别，一个回答“*I don't have gender, neither am I interested in it*”还有一个装模作样地说：“*Now it's not really a good time to talk about this*”。

在问到“*Who made you?*”时，它们都能回答出自己的设计者。

有一位在被问到“*Can you talk Chinese?*”的时候，答曰“*I am not designed to talk in Chinese, but the technology is language independent.*”

它们的共同特点就是，凡是找不到合适的话题或接收了一句不能理解的话的时候，往往就扯到公司的产品和值得炫耀的技术上去。

聊天机器人的原理是，把机器人能说的所有的话都写在了脚本里，并定义一些变量，根据需从与人的对话里抽取具体的取值。当人的对话中呈现出某种模式时，对应的应答就被激活，经过某种代换后输出。

4.3 实例程序：0624_Magic_Answer_Ball_3.txt