**Linux操作系统大作业报告**

1. 简介：

项目名称：俄罗斯方块

项目类型：游戏

项目目标：成功实现游戏，希望能够找到添加背景音乐的方法

操作方法：通过左右移动，旋转等方式来实现最佳的方块堆砌位置及状态。

参考资料：网络游戏代码

1. 涉及的函数及方法：

1、两个主要函数  
RunAsDisplayer()、RunAsKeyReceiver()。

主要命令有：翻转、左移、右移、下移、直接下落到底、退出。  
2、绘制函数  
  Initdraw() #初始绘制  
3、指令处理函数  
  BoxRotate() #接收上方向键，用于旋转方块  
  BoxRight() #右移  
  BoxLeft() #左移  
  BoxDown() #下移，下移时要判断，如果到底了，要写入背景，并创建新方块  
  BoxAllDown() #直接下移到最底部  
4、移动可行性判断函数  
  BoxMove() #该函数包含两个表示目的地址的参数，测试是否可以移动，如果可以移动，则返回0，否则返回1  
5、写入背景函数  
  Box2Map() #该函数用于把到底的方块写入到背景当中，并消去可以消去的行  
6、退出函数  
  MyExitNoSub() #用于恢复终端并退出指令接收进程  
  MyExit() #关闭两个进程，调用其他两个退出函数实现  
  ShowExit() #关闭显示进程

1. 部分实验代码：

1.七中不同的方块的定义

#通过旋转，每种方块的显示的样式可能有几种

box0=(0 0 0 1 1 0 1 1)

box1=(0 2 1 2 2 2 3 2 1 0 1 1 1 2 1 3)

box2=(0 0 0 1 1 1 1 2 0 1 1 0 1 1 2 0)

box3=(0 1 0 2 1 0 1 1 0 0 1 0 1 1 2 1)

box4=(0 1 0 2 1 1 2 1 1 0 1 1 1 2 2 2 0 1 1 1 2 0 2 1 0 0 1 0 1 1 1 2)

box5=(0 1 1 1 2 1 2 2 1 0 1 1 1 2 2 0 0 0 0 1 1 1 2 1 0 2 1 0 1 1 1 2)

box6=(0 1 1 1 1 2 2 1 1 0 1 1 1 2 2 1 0 1 1 0 1 1 2 1 0 1 1 0 1 1 1 2)

2.判断输入了何种键

if [[ $key == $cESC && ${aKey[1]} == $cESC ]]

then

#ESC键

MyExit

elif [[ ${aKey[0]} == $cESC && ${aKey[1]} == "[" ]]

then

if [[ $key == "A" ]]; then sig=$sigRotate #<向上键>

elif [[ $key == "B" ]]; then sig=$sigDown #< 向下键>

elif [[ $key == "D" ]]; then sig=$sigLeft #< 向左键>

elif [[ $key == "C" ]]; then sig=$sigRight #< 向右键>

fi

elif [[ $key == "W" || $key == "w" ]]; then sig=$sigRotate #W, w

elif [[ $key == "S" || $key == "s" ]]; then sig=$sigDown #S, s

elif [[ $key == "A" || $key == "a" ]]; then sig=$sigLeft #A, a

elif [[ $key == "D" || $key == "d" ]]; then sig=$sigRight #D, d

elif [[ "[$key]" == "[]" ]]; then sig=$sigAllDown #空格键

elif [[ $key == "Q" || $key == "q" ]] #Q, q

then

MyExit

3.处理显示和游戏流程的主函数

function RunAsDisplayer()

{

local sigThis

InitDraw

#挂载各种信号的处理函数

trap "sig=$sigRotate;" $sigRotate

trap "sig=$sigLeft;" $sigLeft

trap "sig=$sigRight;" $sigRight

trap "sig=$sigDown;" $sigDown

trap "sig=$sigAllDown;" $sigAllDown

trap "ShowExit;" $sigExit

while :

do

#根据当前的速度级iLevel不同，设定相应的循环的次数

for ((i = 0; i < 21 - iLevel; i++))

do

sleep 0.02

sigThis=$sig

sig=0

#根据sig变量判断是否接受到相应的信号

if ((sigThis == sigRotate)); then BoxRotate; #旋转

elif ((sigThis == sigLeft)); then BoxLeft; #左移一列

elif ((sigThis == sigRight)); then BoxRight; #右移一列

elif ((sigThis == sigDown)); then BoxDown; #下落一行

elif ((sigThis == sigAllDown)); then BoxAllDown; #下落到底

fi

done

#kill -$sigDown $$

BoxDown #下落一行

done

}

4.游戏主程序

if [[ "$1" == "-h" || "$1" == "--help" ]]; then

Usage

elif [[ "$1" == "--version" ]]; then

echo "$APP\_NAME $APP\_VERSION"

elif [[ "$1" == "--show" ]]; then

#当发现具有参数--show时，运行显示函数

RunAsDisplayer

else

bash $0 --show& #以参数--show将本程序再运行一遍

RunAsKeyReceiver $! #以上一行产生的进程的进程号作为参数

Fi

1. 运行结果：



1. 实验中遇到的困难：

硬件：电脑出故障，经一番努力后，重置系统。

软件：软件安装时出现权限等问题。

其他：没有找到能够成功插入背景音乐的方法。

1. 项目总结：

借鉴网络以及同学的帮助下安装系统并成功运行出结果，但经过此项目，对Linux操作系统和shell有了新的认知。