TowerFS文件系统游戏

计52 于纪平 2015011265

计55 张瑞喆 2015011371

计55 王逸松 2015011369

二〇一八 年 六 月

**摘要**

本课程要求同学们在大作业中开发实现一款和存储技术原理相关的计算机软件。我们选择独立设计和开发一款与传统文件系统所不同的文件系统，这个文件系统参考和借鉴了经典游戏“魔塔”中的游戏机制，结合了打怪升级、获得物品等游戏机制，又综合设计了“传送门”机制，使得整个游戏变得非常有趣。TowerFS文件系统是我们将魔塔游戏与文件系统相有机结合，开发出的一款基于文件系统的游戏。游戏中的各种操作，均与文件系统中的操作相对应，创意十分新颖。~~游戏的开发使用了面向对象程序设计技术进行封装，为二次开发和维护提供了便利条件。~~通过本次作业，我们团队学习到了如何编写一个简易的，而且是自定义的文件系统的技术，又编写出一个有趣的游戏，可以说是一举两得。

**关键词**：文件系统；游戏；fuse

目录

[设计灵感与初衷 3](#_Toc516454497)

[设计目标 3](#_Toc516454498)

[设计思路 3](#_Toc516454499)

[实现效果 3](#_Toc516454500)

[代码结构 3](#_Toc516454501)

[未来工作 3](#_Toc516454502)

[总结 3](#_Toc516454503)

# 设计灵感与初衷

在2017-2018春季学期，我们三人选修了“存储技术基础”课程，课程要求我们完成一个与存储系统相关的作业设计。在讨论相关方案时，我们一开始希望能实现一个文件系统作为最终的作品。但是主流文件系统都已经有了具体实现，再次实现未免枯燥乏味，没有新意。而进行主流文件系统的开发，往往还面临着开发方面的各种困难，因此，我们认为开发一个新的文件系统并非一个好的选择。

在此基础上，我们考虑以文件系统为基础，做出新的扩展应用，该应用能基于文件系统本身，给使用者提供不同于一般文件系统的体验。我们在讨论后决定，利用文件系统的特性，开发一款不同于通常游戏的一款游戏。

我们选择了魔塔游戏作为原型，我们在讨论中发现了魔塔游戏和文件系统的许多共同之处，并以此为契机，基于文件系统，开发了一款和魔塔形式类似的命令行探险游戏。

# 《魔塔》游戏简介

《魔塔》游戏是一款经典的策略类的固定数值RPG游戏。游戏需要动很多脑筋，任何一个轻率的选择都可能导致游戏的失败。魔塔游戏虽不大，但是制作精美，道具很多，而且难度不低，对智商是一次艰巨的考验。

在《魔塔》的关卡中，玩家需要通过打败怪物、获得道具、打开门锁、利用道具提高各项参数等方式进行游戏，游戏操作简单，容易上手，又十分有趣。

《魔塔》自推出以来，经过反复的开发，衍生出了同一个游戏框架下的许多游戏版本。0ksh制作了RMXP用的魔塔样板，有兴趣的人可以使用该样板，编辑配置文件，制造出属于自己的魔塔关卡，这一改进使得制作魔塔的能力普及。许多玩家纷纷编写不同的关卡配置文件，使得魔塔的开发得到普及，衍生出许多不同的版本。

# 设计思路

在经过考虑之后，我们选择设计一个名为《TowerFS》的文件系统游戏，作为课程的大作业。

我们考虑过许多不同的游戏，最终选择了《魔塔》系列，是因为它和文件系统具有许多共通之处，下面一段将结合这些共同关系，叙述整个文件系统的设计思想：

1. 基本设定

文件系统的使用者即对应着魔塔中的冒险者，因此，我们为文件系统的使用者设计了以下基本属性：

* 血量（HP）
* 攻击力（ATK）
* 防御力（DEF）
* 金币（GOLD）

使用者在使用的过程中，与文件系统发生各种交互时，同魔塔游戏中一样，这些参数会随着使用者进行各种操作，而发生改变。这与游戏中随着玩家操作，游戏角色的各项参数发生改变本质是具有共通之处的。

1. 布局关系

在我们的设计中，一个文件夹的内容和魔塔中一层的内容具有相似性。在魔塔中，到达一层的时候，往往需要玩家先清除某些怪物，然后获得一些道具。再进入到下一层中。而在文件系统游戏中，我们设计了进入文件夹时可以看到一些文件，而在进行了一些操作之后，文件夹中可能会出现一些新的文件，这也与魔塔中的关卡有相似之处。

我们对文件系统游戏进行设计时，简化了这一流程，在进入每个文件夹后，会先出现若干“事件”，通常是文件夹中出现了一些文件，这些文件对应着魔塔中的“怪物”。在打败了这些怪物（对这些文件做出相应的处理）之后，才会出现新的道具或到达新的关卡。

1. 连接关系

魔塔中，通过楼梯可以实现上下楼之间的移动，通过特殊道具可以实现跨越层级的移动。

文件系统中，通过子文件夹可以实现在上下级文件目录中切换，通过链接可以实现跳到其他文件夹的移动。

1. 行动的处理

在文件系统中，通过“cd”命令打开文件夹在我们的设计中和魔塔中的进入下一层（也有可能是上一层），甚至可能有多个“后继”相类似。通过“cd”本身，也可以使得终端实现到达某一层的操作。

1. 战斗事件的处理

在魔塔中，玩家通过向怪物方向移动，来完成和怪物的对战，具体来说，玩家在怪物面前一格，朝着怪物移动时，会和该怪物发生对战。在对战的过程中，参照回合制对战的方式计算伤害，玩家在每个回合先行攻击。被攻击的一方会损失，其中，ATK表示攻击方的攻击力，def表示被攻击方的防御力。直到有一方血量为0时结束，血量不为9的一方获胜。如果玩家获得胜利，则玩家可以获得该怪物携带的所有金币。如果怪物获得胜利，则玩家死亡，游戏失败。特别地，如果双方都不能对对方造成伤害，则也判定为玩家失败。

而在文件系统游戏中，我们设置了一个“fight”命令来进行这一操作。玩家可以通过“./fight green\_slime”命令来和一只“绿色史莱姆”发生对战。对战的记录会被记录下来，并打印在屏幕上反馈给玩家。对战记录中详细记叙了玩家和怪物发生战斗的每一回合发生的事件。

1. 门与钥匙

在魔塔游戏中，门与钥匙是两个很重要的元素，玩家可以通过“钥匙”打开“门”，打开门之后，会进入到一定的区域内，而这个区域内可能会具有新的怪物和物品。

在我们的文件系统游戏中，也设计有相应的机制。在文件系统中，有些地方可以获取到一定的key，而通过使用这些Key，可以打开相应的一些文件夹（进入到新的空间中）。

1. 权限控制与作弊

为了防止玩家在游戏中，通过强制的修改、删除文件达到作弊的效果。我们的设计中，文件系统的使用者对其中的各种文件和文件夹均无权修改、复制、删除。这也保证了游戏的公平性与趣味性。

# 实现效果

# 代码结构

# 未来工作

半个学期的时间过得飞快，我们的作业从无到有，从框架搭建到具体的功能实现，取得了较好的效果。但是，设计中的部分内容并没有被完全实现，例如，对于锁和门的机制虽然已经在框架中预留，但尚未开发。在将来，希望能进行相关方向的进一步开发，完善这一功能。

关于锁功能的详细设计为：在进入passage时，需要先调用NodeLinkStatus 中的“can\_reach”函数判定是否能进入到该通道，而当前这个判定是直接返回“true”的，在未来，如果继续这一工作，需要对这个地方添加判断逻辑，确定是否能具有相应的钥匙来通过这一通道。同时需要增加钥匙文件和标记获取钥匙的功能。

# 总结

在本次大作业中，我们队伍的三位同学都对文件系统的构成有了进一步的深入了解。同时在设计中将所学到的文件系统知识与有趣的游戏相结合，开发了一个这样的非常有意义的作品。在这里，要感谢老师和助教对我们的指导和帮助。课程临近尾声，希望在未来，新的同学能选修存储技术基础课程，继续我们这项有意义的作业。