**开心农场设计报告**

组别： 第一组

撰写人： 蔡敏瑄

成亚男

黄晓倩

时间：2016年8月28日

**目录**

1. **体系结构设计**

1.1用户界面-----------------------------------------------------------3

1.2业务处理-----------------------------------------------------------3

1.3数据库访问---------------------------------------------------------3

1. **模块设计**

2.1用户登录-----------------------------------------------------------4

2.2用户注册-----------------------------------------------------------4

2.3农场模块-----------------------------------------------------------5

2.4背包模块-----------------------------------------------------------5

2.5商店模块-----------------------------------------------------------5

2.6仓库模块-----------------------------------------------------------6

2.7土地模块-----------------------------------------------------------6

2.8宠物模块-----------------------------------------------------------7

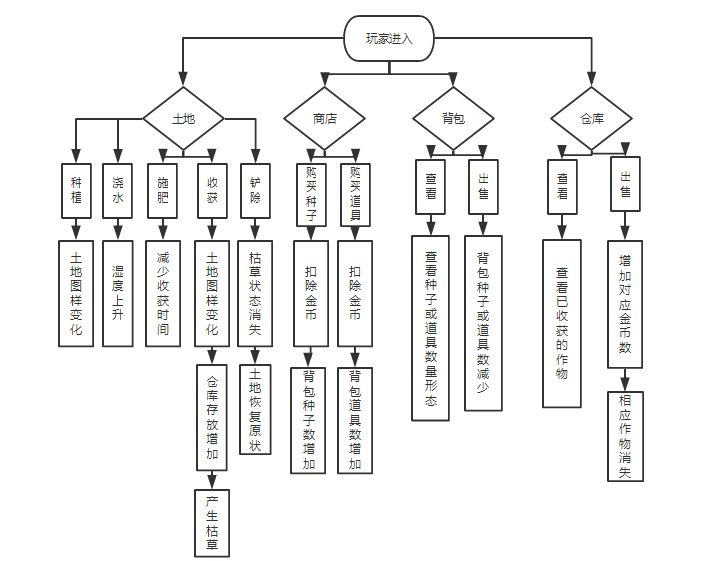
1. **功能设计**-------------------------------------------------------------8
2. **系统实现设计**--------------------------------------------------------9

**5. 农场界面设计**--------------------------------------------------------12

**1.体系结构设计**

**1.1用户界面**

用户界面还是较为简单明了的，具体内容如下面流程图所示：



**1.2业务处理**

业务处理业务处理在整个农场里所需要进行的操作有哪些。

主要是对植物和道具的购买和出售的查询以及修改操作：

植物可以进行增加或减少的按钮以便修改购买或者出售次数，道具同理。

1**.3数据库访问**

数据库主要是用来存放农场所实现的各类数据。

有各类种子数，各种道具的用途，收货农作物数，用户的个人信息，每个土地的操作和使用情况。这些在后面的各个模块会依次对其进行分析。

**2.模块设计**

**2.1用户登录**

|  |  |
| --- | --- |
| 用例名称 | 用户登录 |
| 标识符 | 0001 |
| 用例描述 | 实现游戏者的身份验证 |
| 参与者 | 游戏玩家 |
| 前置条件 | 已经注册 |
| 后置条件 | 游戏信息初始化 |
| 基本操作流程 | 输入账号密码，系统自动连接数据库验证 |
| 分支操作流程 | 用户已登录，提示强迫下线或退出 |
| 异常操作流程 | 登录失败，返回登录页面，系统正在维护。。。 |

**2.2用户注册**

|  |  |
| --- | --- |
| 用例名称 | 用户注册 |
| 标识符 | 0002 |
| 用例描述 | 实现游戏者的账号建立 |
| 参与者 | 游戏玩家 |
| 前置条件 | 进入主界面 |
| 后置条件 | 连接数据库，注册成功，自动登录 |
| 基本操作流程 | 输入账号密码，身份信息 |
| 分支操作流程 | 注册失败，用户已存在 |
| 异常操作流程 | 注册失败，返回注册页面，系统正在维护。。。 |

2.1与2.2里可以包含一个用户表。

主要内容：用户名username、用户密码password、玩家金币数money。

1. 用户进行注册时需要输入属于自己的密码，这时后台将此密码进行再一层加密后再存入数据库中。
2. 在用户开始登陆时输入用户名和密码，在数据库中查找是否匹配，以判别能否登陆游戏。
3. 若用户一直在同一IP地址登陆游戏，每次进入时就可以直接进入游戏画面而不用再进行登陆。
4. 用户的金币数：在进行购买操作时，若购买所需金币数小于等于用户本身的金币数，则可以进行扣除操作，否则扣除失败；在进行出售操作时，只需要增加用户的金币数即可。

**2.3农场模块**

|  |  |
| --- | --- |
| 用例名称 | 农场模块 |
| 标识符 | 0003 |
| 用例描述 | 显示玩家用户，等级，金钱，升级经验等 |
| 参与者 | 游戏玩家 |
| 前置条件 | 已经登录 |
| 后置条件 | 玩家自由操作 |
| 基本操作流程 | 登录游戏，游戏初始化 |
| 分支操作流程 | 农场信息更新 |
| 异常操作流程 | 农场信息更新失败，返回主页面，系统正在维护中。。。 |

**2.4背包模块**

|  |  |
| --- | --- |
| 用例名称 | 背包模块 |
| 标识符 | 0004 |
| 用例描述 | 显示玩家拥有的种子，金钱，道具等 |
| 参与者 | 游戏玩家 |
| 前置条件 | 已经登录 |
| 后置条件 | 使用种子、道具，售卖种子、道具 |
| 基本操作流程 | 登录游戏，点击背包图标 |
| 分支操作流程 | 背包信息更新 |
| 异常操作流程 | 背包信息更新失败，返回农场，系统正在维护中。。。 |

**2.5商店模块**

|  |  |
| --- | --- |
| 用例名称 | 商店模块 |
| 标识符 | 0005 |
| 用例描述 | 显示商店售卖拥有的种子，道具、宠物等 |
| 参与者 | 游戏玩家 |
| 前置条件 | 已经登录 |
| 后置条件 | 购买种子、道具、宠物 |
| 基本操作流程 | 登录游戏，点击商店图标 |
| 分支操作流程 | 商店信息更新 |
| 异常操作流程 | 商店信息更新失败，返回农场，系统正在维护中。。。 |

此模块中包含两个数据库：农作物表和道具表

农作物表：名称plantname，生长时间growtime，种子价格seedmoney，成熟果实价格fruitmoney（这个也可以在仓库模块中看见，方便仓库出售）

主要内容：

1. 对各类植物进行系统分类并管理。
2. 购买时可以显示种子各种信息（名称，价格，生长时间，成熟果实形态，成熟后的售价）

道具表：名称toolname，道具价格toolmoney

主要内容：与植物一样，在商店页面可以显示对道具选择。可以在商店中购买所需要的道具。在使用道具时，根据道具表面意思上的功能相应的对土地的时间进行增减操作。

**2.6仓库模块**

|  |  |
| --- | --- |
| 用例名称 | 仓库模块 |
| 标识符 | 0006 |
| 用例描述 | 显示玩家收获的农作物，鲜花，宠物等 |
| 参与者 | 游戏玩家 |
| 前置条件 | 已经登录 |
| 后置条件 | 出售农作物，鲜花、宠物等 |
| 基本操作流程 | 登录游戏，点击仓库图标 |
| 分支操作流程 | 仓库信息更新 |
| 异常操作流程 | 仓库信息更新失败，返回农场，系统正在维护中。。。 |

仓库表：用户id，果实名fruitid，果实数fruitnum

主要内容

1. 用户收获时，根据收获的农作物的名称进行仓库数据的更新，若有相应的名字，则只需要增加数量即可，若存在新的果实，则需要添加。
2. 仓库可以进行出售果实的操作。选择需要出售的数量（最大上限为已有果实的数量），然后进行相应的数据更新。若全部出售，则在仓库里删除此农作物的信息。
3. 在出售果实后，会有用户金币的相应增加操作。

**2.7土地模块**

|  |  |  |
| --- | --- | --- |
| 用例名称 | 土地模块 | |
| 标识符 | 0007 | |
| 用例描述 | 显示土地是否被开垦，土地上种植作物的状态 | |
| 参与者 | 游戏玩家 | 玩家好友 |
| 前置条件 | 已经登录 | 互为好友 |
| 后置条件 | 种植作物，使用道具，收获果实 | 偷取果实（待定） |
| 基本操作流程 | 登录游戏 | |
| 分支操作流程 | 土地信息更新 | |
| 异常操作流程 | 土地信息更新失败，返回农场，系统正在维护中。。。 | |

关于数据库的分析：土地表：可用土地数useland，土地是否已经种植（水果一类，鲜花一类）planted，围绕土地的一系列应用（浇水，施肥，铲除枯草）worktime，收获剩余时间lefttime

主要内容：

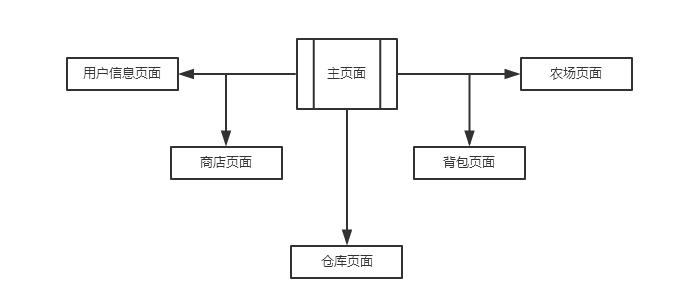
1. 初始登陆进去时关于土地的所有类别都未使用，初值为0.
2. 可用土地数为9块， 不可用为3块。当有足够的金币后，即可开垦土地。
3. 种植农作物时，只有当土地状态为空才可以种植操作。此时土地形态有所改变。
4. 农场功能较为初级，只有在开始玩农场时，种植的农作物才会生长，其余未打开页面的时候，农场时间是暂停的。
5. 是根据土壤的湿度来进行浇水的。要判定湿度的百分比来浇水。浇一次水可以增加40%的湿度，100%为正常。若未能及时浇水，土地状态又会发生改变，此时种植农作物失败，进行铲除操作。
6. 施肥可以减少土地的收获时间。根据所选肥料的种类来判别减少时间的多少。在此之前，设定一个每个土壤限制施肥总数（比如说每个土壤只能最多只能施一次肥），以保证游戏的可玩性。当施肥减少的时间大于剩余时间后，土地作物直接进入收获状态。收获后，土地保持初始状态以待下次继续播种。

**2.8宠物模块（待定）**

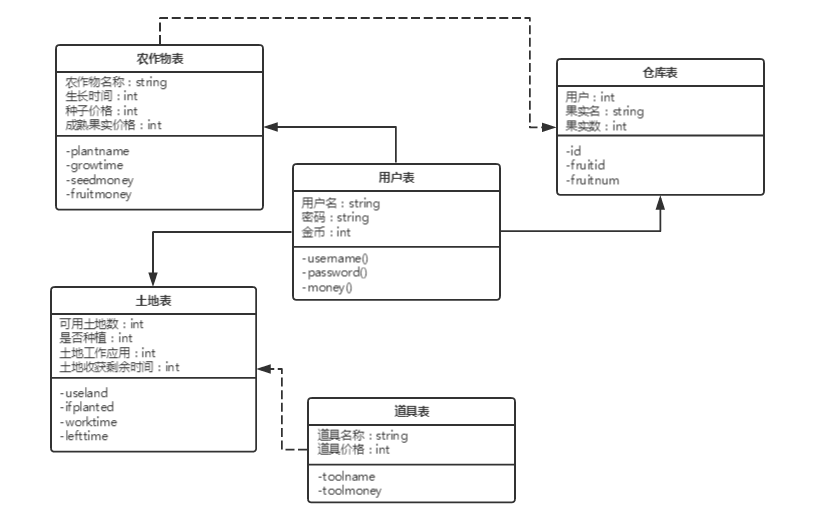
|  |  |  |
| --- | --- | --- |
| 用例名称 | 宠物模块 | |
| 标识符 | 0008 | |
| 用例描述 | 显示看守宠物及其状态 | |
| 参与者 | 游戏玩家 | 玩家好友 |
| 前置条件 | 已经登录 | 互为好友 |
| 后置条件 | 使用道具 | 使用道具 |
| 基本操作流程 | 登录游戏 | |
| 分支操作流程 | 宠物信息更新 | |
| 异常操作流程 | 宠物信息更新失败，返回农场，系统正在维护中。。。 | |

**3.功能设计**

**1.各组件之间关系**



**2.各个模块之间类图对应关系**



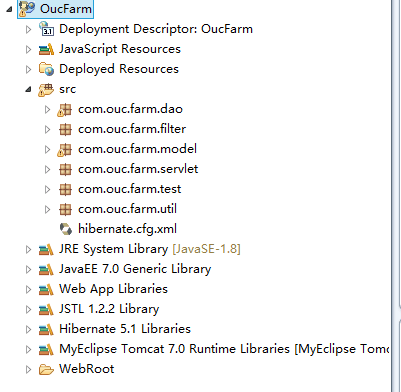
**4.系统实现设计**

经过我们讨论，决定采用web用户前台和java管理员后台相结合的模式，

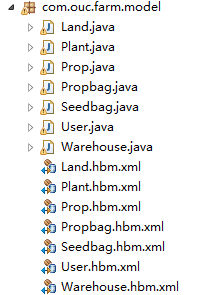
使用web前端技术html + css + javascript ，写出美观大方的用户界面供用户使用，用java搭建后台数据库，供管理员使用，实现数据的增删查改。

下面展示部分截图

1. 项目总体结构图 引入了最新的hibernate数据库持久层框框，方便对数据的持久化操作和事务的管理

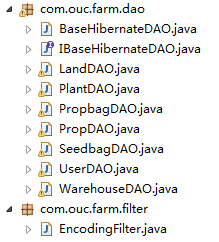


1. 根据数据库映射出的实体类和对应的hibernate文件xxx.hbm.xml。因为使用较为方便的注解有时会产生版本上的错误，所以仍然使用了比较早期的xml配置形式



3）项目dao包下面的dao类 dao时data access object 数据访问对象 定义了数据访问的相应方法 方便servlet或者action中进行调用

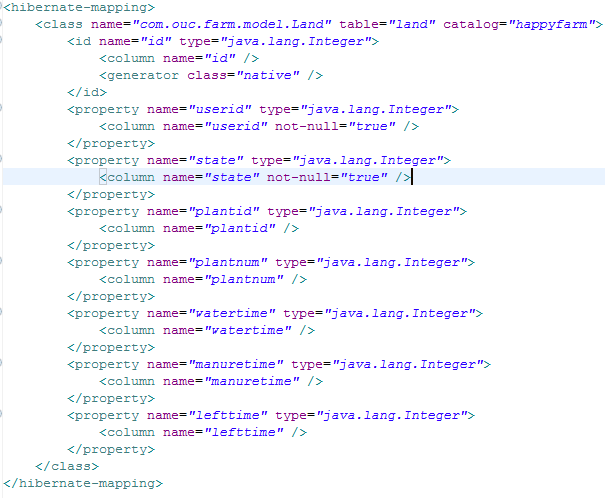
另外还写了HibernateUtil类用来进行SessionFactory和Session的管理



C:\Users\Lenovo\Documents\Tencent Files\605947802\Image\C2C\AY`7E39]GMU~WISOA%IGK~W.png、

这是land表映射出来的land.hbm.xml

就是表之间的关联关系



**5.农场界面设计**

因为项目正在开发中所以目前只有一个简单的页面，土地的平铺立体化未实现，总体来说还算是简洁大方。

