单词消除游戏开发文档

2017211395 熊智宇

1. 单词消除游戏设计要求

单词消除游戏由两类参与者组成：闯关者（即游戏玩家），出题者（为游戏增加游戏中使用单词）。游戏规则为，游戏每一轮，程序会根据该关卡难度，显示一个单词，一定时间后单词消失。闯关者需要在相应地方输入刚刚显示并消失的单词，如果闯关者输入正确（即闯关者输入的单词与刚刚显示的单词完全一致，包含大小写）则为通过。一关可以由一轮或者多轮组成。

题目一 （20 分）

闯关者属性要求必须含有：闯关者姓名、已闯关关卡数、闯关者经验值、闯关者等级， 若有需要可以自行添加其他属性。

出题者属性要求必须含有：出题者姓名、出题者出题数目、等级，若有需要可以自行添加其余属性。

具体功能包括：

* 实现闯关者，出题者本地的注册、登录。
* 程序支持多人注册，同一时间只有一人可以登录。
* 实现游戏规则：出题者增加游戏中使用单词。游戏每一关，程序会根据该关卡难度，显示一个单词，一定时间后单词消失。闯关者需要在相应地方输入刚刚显示并消失的单词， 如果闯关者输入正确则为通过。
* 任何角色均可查询所有闯关者、出题者，按照属性查找相应闯关者、出题者。
* 可以根据闯关者闯过关卡数、经验、等级等对闯关者排名，根据出题者出题数目、等级对出题者排名。

要求：

* 采用面向对象的方式，使用类设计。
* 在设计类时请注意类的继承关系，关注闯关者、出题者的关联（闯关者与出题者有共同的基类）。
* 使用文件或者数据库作为存储对象，自行设计文件格式，或者数据库表结构。
* 在题目的要求下自行扩展功能，功能扩展适当者酌情加分。

题目二（20 分）

闯关者即为游戏玩家，已经注册并登录的玩家可以在系统进行单词消除游戏。每一关的难度要有所增加，体现为如下三个条件中的一个或者多个：1、单词难度可以递增或者持平（即长度加长或不变）； 2、进行轮数增多（即单词数目增加，如：前三关仅仅通过一个单词就过关，后续需要通过两个、三个甚至更多才过关）； 3、单词显示时间缩短（随着关卡的增加显示时间越来越短）。

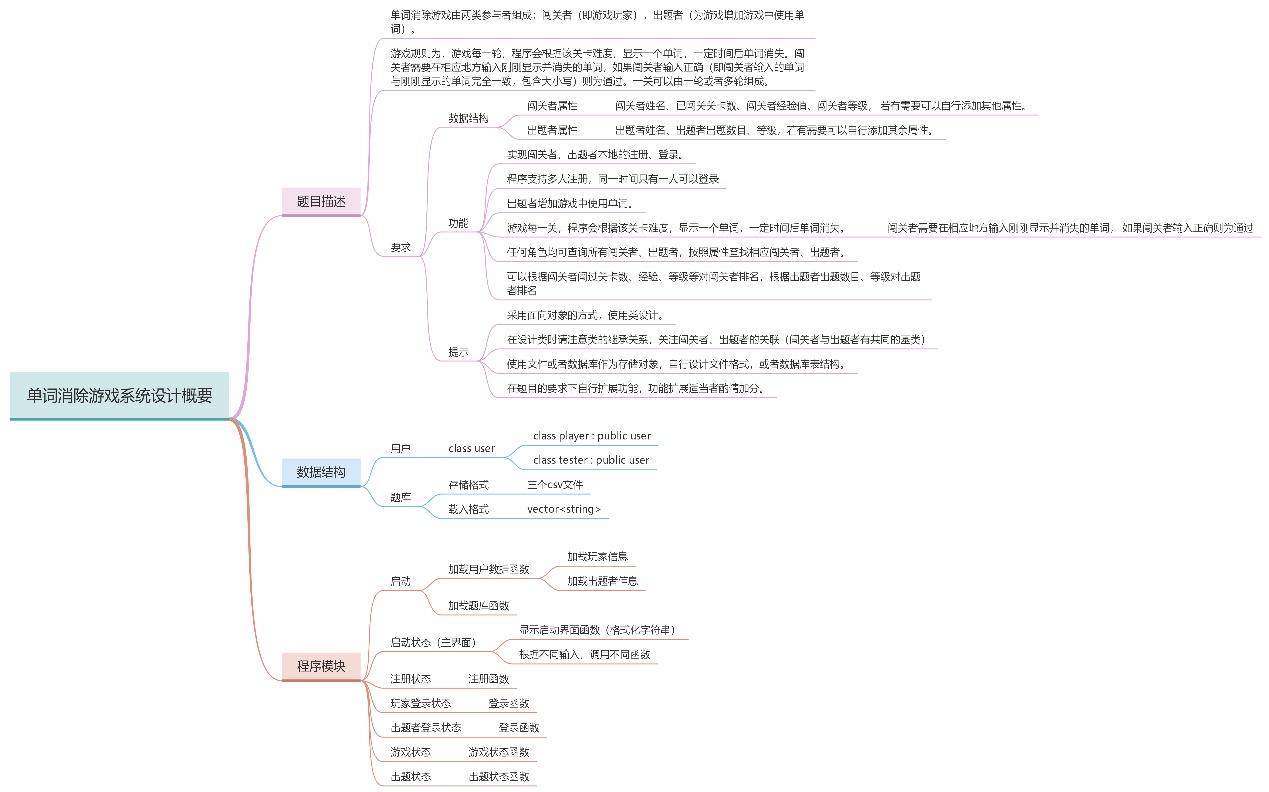
闯关者每闯过一关，增加一定经验值。经验值会根据闯过的该关卡的关卡号、该关的闯关耗费时间共同决定。当经验值累计到一定程度闯关者等级增加。闯关失败需要重新闯该关。

游戏自带词库，而且已经注册的出题者可以为系统出题，即增加词库的新词，已经存

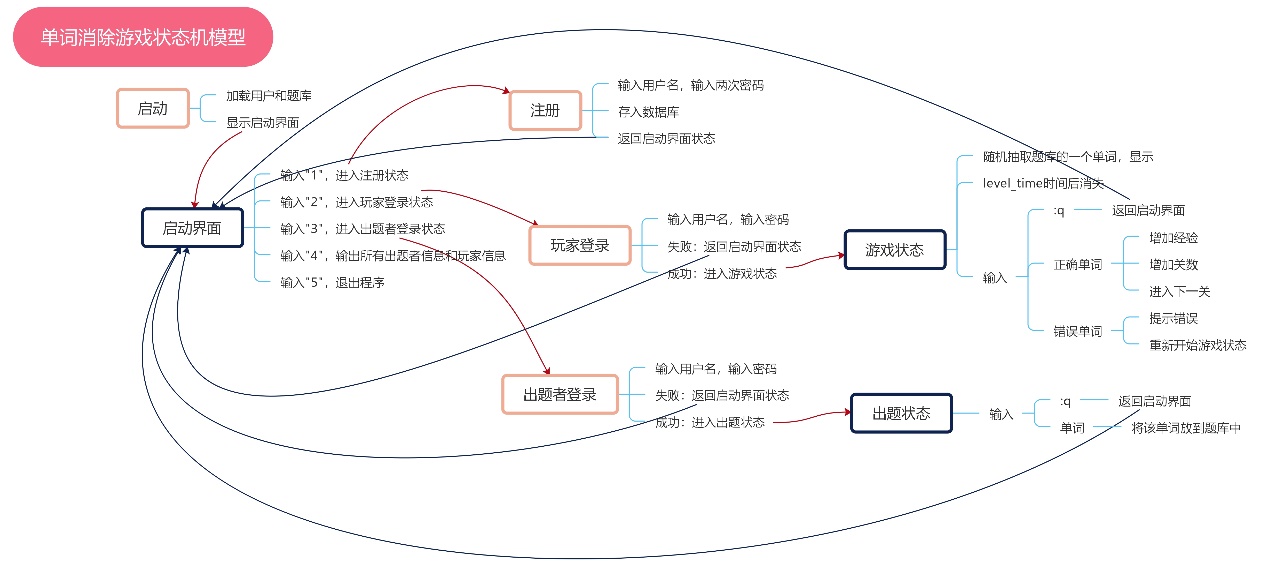
在的单词不能再次添加（词库中的单词构成一个单词池，但建议根据单词的长度来组织存储。每次出题时，系统从该单词池中按照关卡难度随机的选择相应长度的单词）。每成功出题一次，更新该出题者的出题数目。出题者等级根据出题人成功出题数目来升级。

* 必须在题目一的基础上进行修改。
* 请根据要求设计每一关的出题方式，注意随着关卡数增加，题目难度增加。请合理处理出题人新添加新词的使用方式，并且新加词组不会影响游戏难度。
* 设计闯关者经验值，等级增加策略。设计出题者等级升级策略。

1. 单词消除系统设计概要：（大图见附件）



1. 单词消除系统状态机模型：（大图见附件）



1. 程序命名规范
   1. 变量名称以小写开头，单词间用大写分割；
   2. 函数名称以小写开头，单词间用下划线分割；
   3. 自定义类/结构全部小写，单词间用下划线分割；
2. 程序模块划分

Database.h//指定数据格式

Global.h//引用外部依赖库

Mainfunc.h//程序中所有的函数

Database.cpp//类的初始化

Game.cpp//游戏部分

Init.cpp//程序运行加载部分

Login.cpp//用户登录部分

Main.cpp//主程序

Search.cpp//查询用户信息函数，以及其他功能函数

Testmake.cpp//出题部分

User\_register.cpp//用户注册部分

1. 数据类型

Class user//用户类，包括用户的名称和密码。

{

Private:

String name;

String password;

Public:

Get\_name();

Get\_password();

}

Class player : public user

{

Private:

Int exp;//玩家经验

Int level;//玩家等级

Int best\_round;//玩家的到达的最大关数

Public:

player(string newName, string newPassword,int newExp=0, int newLevel=0,int newbestRound=0);

~player();

Get\_level();

Get\_exp();

Get\_best\_round();

Void rewards(int bestRound);//这个函数用于在游戏结束后，根据玩家的本轮游戏的最大关数刷新经验值、最大关和等级。

}

Class tester : public user

{

private:

int problemNum;//出题者贡献的题目数

int level;//出题者的等级

public:

tester(string newName, string newPassword, int newProblemNum = 0, int newLevel = 0);

~tester();

short get\_level();

int get\_problem\_num();

void rewards(int wordAdd);//根据出题者本次出题数更新出题总数和等级

};

Vector<string>wordlist;//单词库，运行时从csv文件加载，加入新对象时同步更新文件

Vector<player>playerList;//玩家列表，运行时从csv文件加载，加入新对象时同步更新文件

Vector<tester>testerList; //出题人列表，运行时从csv文件加载，加入新对象时同步更新文件

1. 开发进程

V1.0

修复了文件最后一个用户读取两次的bug

修复了非法输入导致程序死循环的bug

修复了排名显示bug

修复了文件读取bug

修复了重名用户注册bug

添加了一些默认选项

V2.0

增加了经验的递增奖励机制。现在在每次游戏中完成更多的单词可以获得更多的经验了。当你在一定时间内成功输入单词时，会获得一个奖励关。

增加了升级所需经验的递增机制。现在每次升级需要更多的经验。

增加了游戏难度。随着游戏进行，单词显示时间会逐渐减少，长度会逐渐增加。可能包括含数字、字符的单词。

增加了失败条件：当输入失败5次时游戏结束。

1. 程序使用说明

打开exe文件。根据主界面提示，输入对应数字，进行对应操作。

输入1：进入注册账号界面。再输入1表示注册一个玩家账号，输入2表示注册一个出题人账号，其他输入会退回主界面。选择账号类型后，根据提示输入用户名和密码。

输入2：进入游戏。首先要求您登录一个玩家账号，输入用户名和密码正确后游戏开始，否则退回主界面。输入”:q”可以随时终止游戏，根据您本次游戏情况，账号的经验和等级可能会提升。

输入3：进入出题模式。首先要求您登录一个出题者账号。输入用户名和密码正确后可以自由出题，否则退回主界面。输入”:q”可以随时终止出题。根据您本次游戏情况，账号的出题数量和等级可能会提升。

输入4：进入查询模式。选择1查询所有玩家，选择2查询所有出题者，选择3查询特定名称的用户。模式1和2下可以将用户以不同方式排序显示。

输入0：退出游戏。