

DASH 项目设计文档

181830044 董宸邳

2020.10

概述

- Dash的设计思路来源于经典的飞机大战、节奏大师、俄罗斯方块等游戏。
- 游戏中，文本以单个字符的形式从控制台上方不断落下，玩家需要在字符掉落到分隔线之前完成对应按键的敲击，以获取得分。
- 游戏分为若干个难度，难度的区别主要表现在字符下降的速度。
- 除主体游戏外，本程序还为普通用户和管理员设计了不同的操作内容和权限，详见后续讲解。

模块划分

- 1. Game
 - 游戏的运行
 - 游戏记录的结算
- 2. User
 - 玩家信息的初始化、修改
 - 游戏记录的存储、查阅
- 3. Admin
 - 程序整体的运行逻辑控制
 - 游戏生成信息的维护、修改、存取

- FYI: 以下仅展示各类的核心成员，一些辅助变量、函数不再冗述...

GAME

- class Game
 - {
 - private:
 - static Game* g; //保证Game类仅有一个实例
 - float duration; //游戏持续时间
 - float correctness; //正确率
 - int score = 0; //游戏得分
 - difficulty diff; //难度等级
 - void checkInput(); //检查输入是否正确 多线程
 - public:
 - static Game* getInstance(string str, float du, difficulty di);
 - void mainGame(); //负责游戏的主体逻辑及画面的构建、刷新
 - friend class Admin;
 - };
- ```
enum difficulty
{
 EASY, MEDIUM, HARD, HELL
};
```

## USER

- //一次游戏的记录
- struct record
- {
- int index;                     //记录编号
- difficulty diff;             //难度等级
- int score;                    //得分
- float duration;             //持续时间
- float correctness;         //正确率
- };
- 与Game类的相关成员一一对应!

# USER

- class User
- {
- private:
- string id;                     //唯一标识符
- string nickname;             //昵称
- string pwd;                  //密码
- vector<record> records;       //练习记录
- public:
- //每个用户都用一个单独文件存储，文件名即为id
- //所有id用"ids.txt"另行存储
- User(string id);
- User(string id, string nm, string pwd);       //构造函数重载
- void displayRecord();                         //展示练习记录
- void storeRecord();                            //将记录保存到文件中
- friend ostream& operator<<(ostream& os, record& r);   //重载record输出
- friend class Admin;
- };

# ADMIN

- class Admin
- {
- private:
- //保证Admin类**仅有一个实例**
- **static** Admin\* a;
- map<string, User> users;     //用户组成的**字典**
- vector<string> src;     //打字文本集合
- User\* activeUser;     //当前活跃用户
- void storeIDs();     //保存所有id至"**ids.txt**"
- void login();     //登录
- void regist();     //注册
- void resetPwd(string);     //重置密码
- bool changePwd(string);     //更改密码
- void play();     //加载游戏
- };

- class Admin
- {
- void storeTxts();     //保存所有文本名至"**src.txt**"
- void addTxt(string); //增加文本
- void rmTxt(string); //移除文本
- void userMenu();     //用户菜单
- void adminMenu(); //管理员菜单
- public:
- **static** Admin\* getInstance();
- void mainMenu();     //主菜单
- };



## 菜单切换

- void mainMenu();
- void adminMenu();
- void userMenu();

## 文本管理

- void storeTxts();
- void addTxt(string);
- void rmTxt(string);

## 玩家操作

- void login();
- void regist();
- void resetPwd(string id);
- bool changePwd(string);
- void play();

请选择您的操作序号:

1. 玩家登录
2. 玩家注册
3. 管理员操作
4. 退出程序

请选择以下玩家操作:

1. 开始游戏!
2. 查看练习记录
3. 修改密码
4. 返回主菜单

请选择您的管理员操作序号:

1. 查看可用文本
2. 增加练习文本
3. 删除练习文本
4. 为用户重置密码
5. 返回主菜单

h

a

r

e

t

s

a

i

s

d

o

n

t

h

a

r

t

a

e

h

当前得分: 100

当前正确率: 62.50%

提示: 按下Esc可提前结束游戏!

**THE END...**