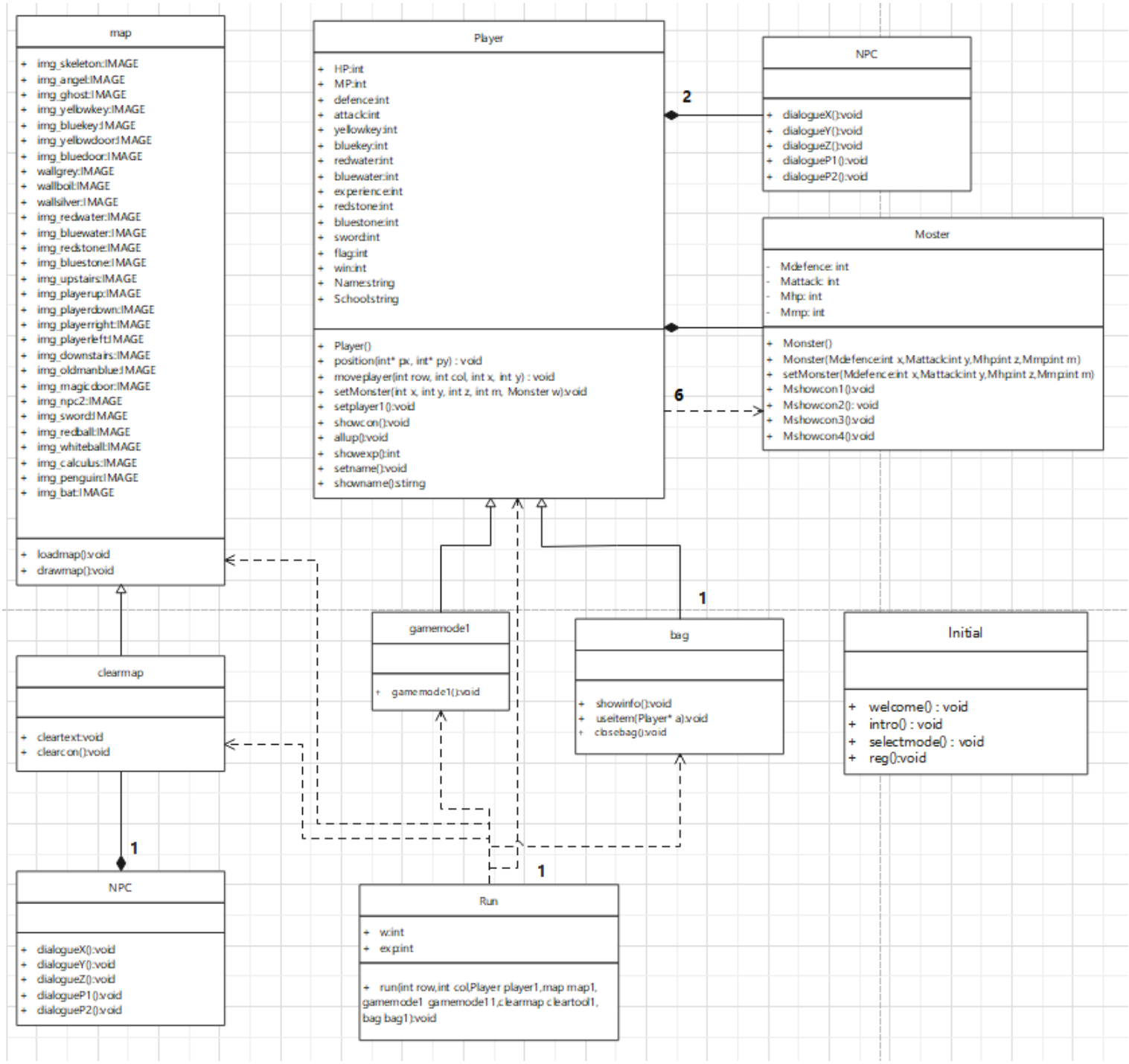
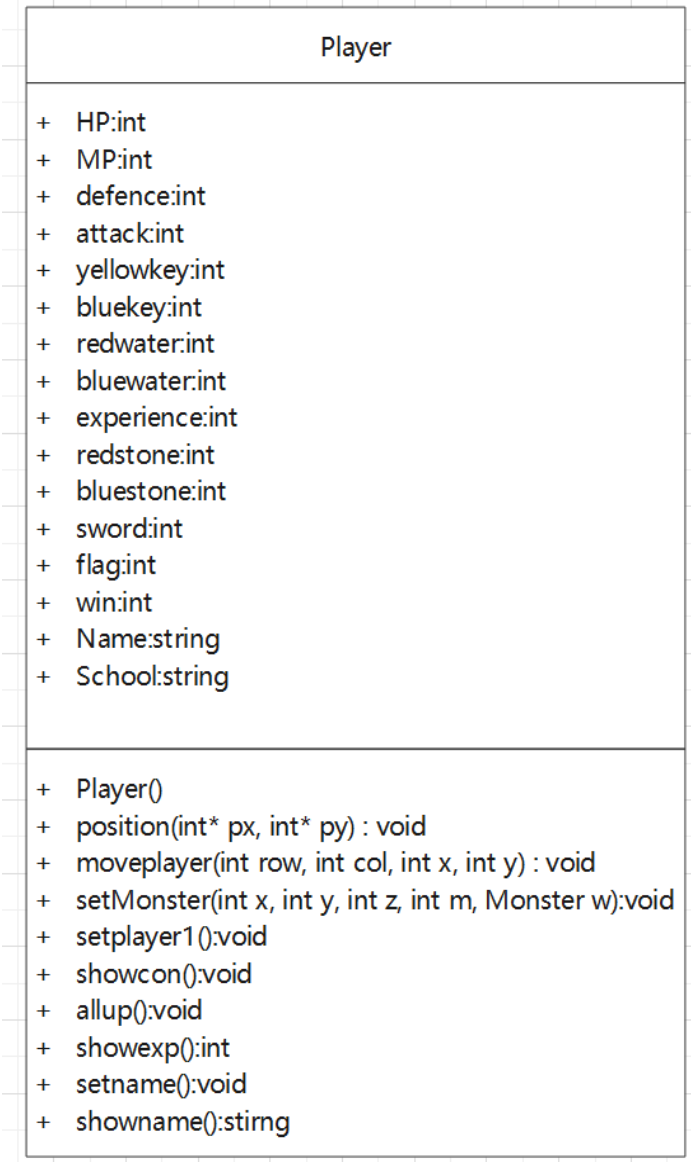
# 设计说明书:勇闯博雅塔

## 一、类设计

### 1. UML 图



## 2.具体类说明



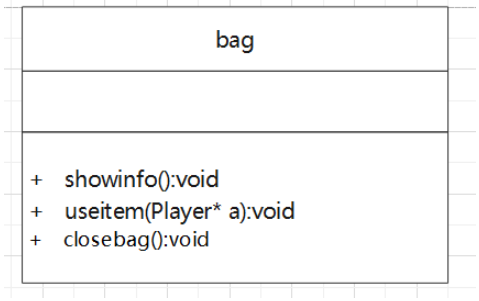

### Class Player:（玩家类）

***数据成员：**

HP:玩家血量值，用于战斗结果判定。
MP:玩家魔法值，用于战斗结果判定。
defence:玩家防御力，用于战斗结果判定。
attack:玩家攻击力，用于战斗结果判定。
yellowkey:土之钥的数量，用于开启土之门。
bluekey:水之钥的数量，用于开启水之门。
redwater:红色药剂，使用后增加血量和攻击力。
bluewater:蓝色药剂，使用后增加魔法值、防御力。
experience:经验值，击杀怪物经验增加，用于排名。
redstone:红色宝石，使用后增加攻击力。
bluestone:蓝色宝石，使用后增加防御力
sword:宝剑数，用于开启最终密藏
flag:用于判定游戏状态，0 值继续游戏，1 值退出。
win:用于判定游戏胜负，0 值游戏失败，1 值胜利。
Name:用于存储玩家注册姓名。
School:用于存储玩家注册学校、

***函数成员：**

Player:默认构造函数，所有数据设为默认值。
Position:寻找玩家位置函数。在函数内部修改坐标参数直到寻找到玩家所在坐标。
moveplayer:移动玩家位置函数。通过检测输入值改变玩家坐标并进行游戏判定。
setMonster:设置怪物函数。用于设置一个怪物
setplayer1:设置玩家函数。用于新建一个玩家。
showcon：显示状态函数。用于显示玩家属性状态。
allup:全物品增加函数。用于增加玩家的药剂与宝石。
showexp:显示经验函数。用于返回玩家的经验值。
setname:设置信息函数。用于玩家注册姓名和学校。
showname:显示姓名函数。用于返回玩家姓名。




### Class bag:（背包类）

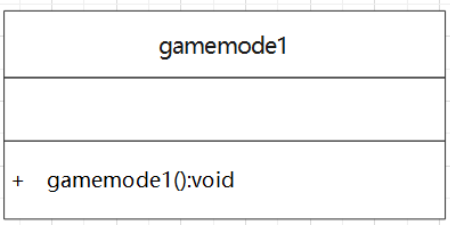
***函数成员：**

showinfo:显示背包信息函数。显示背包中的使用信息。
useitem:使用背包物品函数。使用背包中的药剂和宝石。
closebag:关闭背包函数。关闭已经打开的背包。

Class gamemode1:（游戏模式类）

***函数成员：**

Gamemode1:设置无敌模式。设置游戏模式为无敌版，用于作弊和测试使用。




Class map:（地图类）

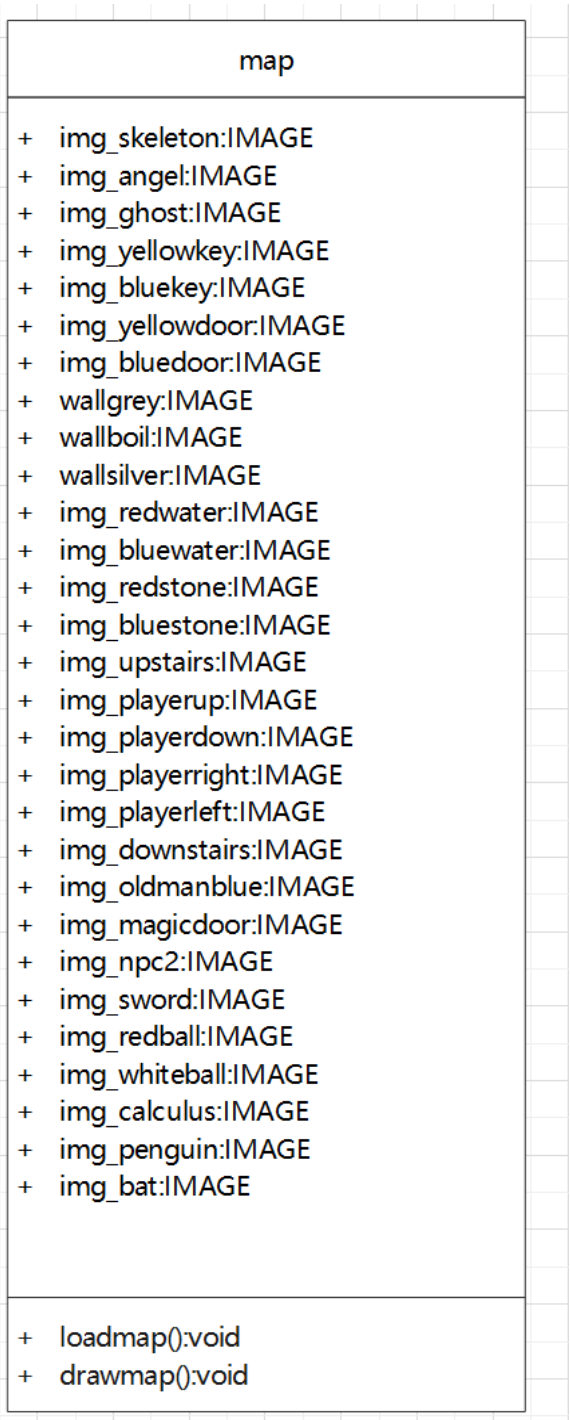
***数据成员：**

(数据成员为 easyX 库中 IMAGE 类型，即图像对象)

img_skeleton:骷髅图像对象。作为怪物。
img_angel:天使图像对象。作为 NPC。
img_ghost:幽灵图像对象。作为怪物。
img_yellowkey:土之钥图像对象。作为物品。
img_bluekey:水之钥图像对象。作为物品。
img_yellowdoor:土之门图像对象。作为建筑。
img_bluedoor:水之门图像对象。作为建筑。
wallgrey:灰色地板图像对象。作为建筑。
wallboil:棕色地板图像对象。作为建筑。
wallsilver:银色地板图像对象。作为建筑。
img_redwater:红色药水图像对象。作为药剂。
img_bluewater:蓝色药水图像对象。作为药剂。
img_redstone:红色宝石图像对象。作为宝石。
img_bluestone:蓝色宝石图像对象。作为宝石。
img_upstairs:上楼梯图像对象。作为建筑。
img_playerup:人物上行图像对象。人物行走。
img_playerdown:人物下行图像对象。人行走。
img_playerright:人物右行图像对象。人行走。
img_playerleft:人物左行图像对象。人行走。
img_downstairs:下楼梯图像对象。作为建筑。
img_oldmanblue:蓝色老人图像对象。NPC。
img_magicdoor:魔法门图像对象。作为建筑。
img_npc2:魔法老人图像对象。作为 NPC。
img_sword:宝剑图像对象。作为物品。
img_redball:红色球怪图像对象。作为怪物。
img_whiteball:白色球怪图像对象。作为怪物。
img_calculus:魔法书图像对象。作为密藏。
img_penguin:企鹅图像对象。作为 BOSS。
img_bat:蝙蝠人图像对象。作为 BOSS。

***函数成员：**

loadmap:上传图像函数。将数据成员中的图像从本地文件中上传。



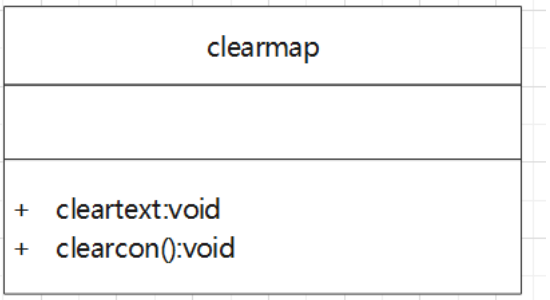
drawmap:绘制函数图像.按照三维数组约定的地图进行地图的绘制。

**Class clearmap:（清理地图类）**

***函数成员：**

cleartext:清理文本函数。将对话框中的文本清空。

clearcon:清空人物状态函数。用于人物属性数据的更新。




**Class Moster:（怪物类）**

***数据成员：**

Mdefence:怪物防御力。用于战斗判定。

Mattack:怪物攻击力。用于战斗判定。

Mhp:怪物生命值。用于战斗判定。

Mmp:怪物魔法值。用于战斗判定。

***函数成员：**

Monster:默认构造函数。设置属性为默认值

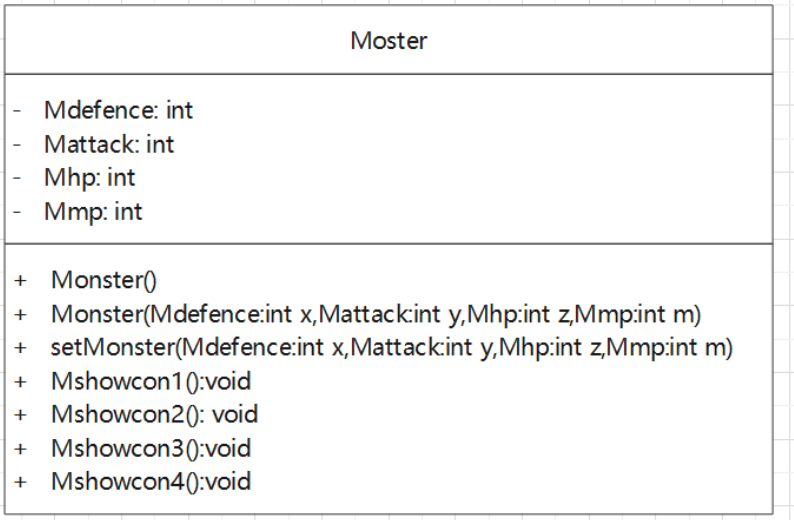
setMonster:建立怪物函数。设置一个怪物。

Mshowcon1:怪物对话函数。怪物对话 1.

Mshowcon2:怪物对话函数。怪物对话 2.

Mshowcon3:怪物对话函数。怪物对话 3.

Mshowcon4:怪物对话函数。怪物对话 4.




**Class NPC:（NPC 类）**

***函数成员：**

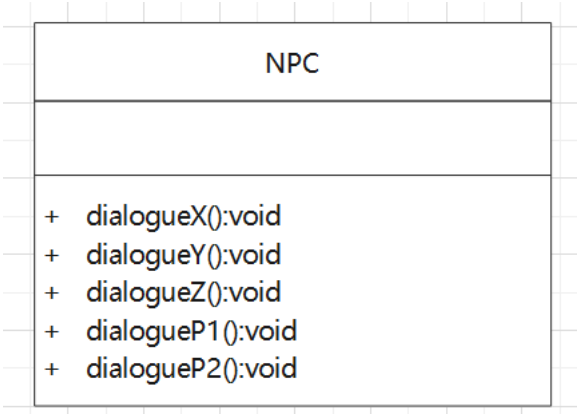
dialogueX:NPC 对话函数。NPC 对话 1.

dialogueY:NPC 对话函数。NPC 对话 2.

dialogueZ:NPC 对话函数。NPC 对话 3.

dialogueP1:NPC 对话函数。NPC 对话 4.

dialogueP2:NPC 对话函数。NPC 对话 5.




**Class Run:（游行运行类）**

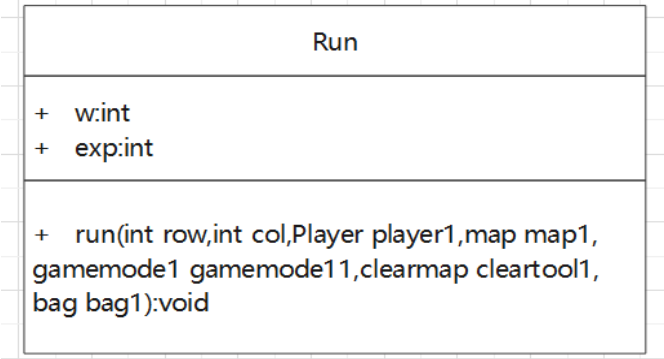
***数据成员**

w:储存游戏胜负判定数据。用于游戏胜负判断。

exp:储存人物经验值数据。用于游戏排行。

***函数成员：**

run：游戏运行函数。形参为各种类，在主函数中进行游戏主要操作运行的函数。



## 二、主要技术难点和及实现方案/算法设计

本款游戏的设计思路是还原并创新魔塔类游戏。设计者本学期第一次进行程序设计相关课程的学习在大作业的设计过程中遇到不少困难，现列举如下：

**技术难点 1：实现游戏程序可视化。**

设计者在游戏设计之初本想使用命令行程序进行游戏的编写。但发现效果很差且可玩性、美感大大削弱。且如果使用命令行程序，许多人物对话、人物运行操作难以通过设计者已掌握的知识进行解决。

**实现方案 1：设计者将 EasyX 库加入到程序设计。**

使用 EasyX 库中封装的一系列函数进行游戏欢迎界面、游戏运行界面、人物操作模块的实现。实际效果良好。以下展示作者使用的部分函数与实现的游戏功能。

使用 initgraph()函数进行可控大小的可视化图形界面的绘制。设计了游戏运行窗口。

使用 outtextxy()函数进行可控坐标的可视化文字输出，设计了欢迎界面与 NPC 对话窗口。

使用 loadimage()函数上传图片，载入魔塔游戏所需要的贴纸。

使用 putimage()函数进行照片的输出，进行魔塔游戏的地图绘制。

**技术难点 2：人物位置的寻找。**

不同层数游戏初始化后人物位置的确定方法，在游戏设计之处给设计者带来不少困难。由于不同地图之间的差异巨大，且人物移动后需要重新判定人物位置。因此人物位置的寻找需要设计合适的函数进行实现。

**实现方案 2：设置参数为坐标指针的函数。**

设计者在学习指针知识之后，设计了人物位置寻找函数。人物位置寻找函数以坐标整型指针作为参数，可以对于坐标变量直接进行修改，可以理解为引用的变量。然后设置两重循环搜索全界面的每一点的图片信息，当检测到玩家角色图片时停止，返回。此时经过人物位置寻找函数的运行之后，系统的 x、y 坐标变量的值就变成了玩家角色所在的位置。因为全图只有一个玩家角色图片，因此该函数结束后 x、y 坐标的实际值是唯一的。这就实现了玩家角色位置的寻找。作为游戏实际运行的基础。

**技术难点 3：人物与怪物属性的设计**

在游戏设计之处，设计者本以为属性很容易进行规划与设定。但在实际设计过程中设计者发现怪物属性和人物属性的设计需要达到一定的平衡，否则就会出现游戏过于简单而丧失了游戏可玩性与玩家区分度，或者是游戏过于困难导致游戏无法按照正常版本所规定的要求通关。

**实现方案 3：反复调试，寻找合适参数。**

设计者通过多次自己试玩、寻找同学作为内测玩家，终于找到了合适的通关条件与通关参数。使得游戏在可通关的同时，加入了经验值变量，使得不同玩家通关后具有不同的经验值。这就使得游戏可以成功通关，但是通关后的经验值也将作为评判玩家游戏水平的标准。

设计者在游戏中设置了经验值与难度系数成正比的游戏模式，即玩家若想获得更多经验值，那么游戏的难度也将相应提高。最终排行的标准:成功通关且经验高者排名靠前。

**技术难点 4：游戏测试方法的寻找。**

设计者第一次设计游戏程序。发现游戏需要多次进行测试才能够正式进行使用。而重复打开调试界面或反复修改代码使得游戏测试难度加大，耗费大量的时间和精力。

**实现方案 4：设置游戏作弊按钮和无敌版本。**

设计者在游戏设计过程中加入了游戏作弊按钮和无敌版本的选择。游戏作弊按钮为全道具提升函数，在游戏测试过程中，设计者可以通过计算作弊按钮的使用次数以及每关使用的频率测试游戏难度设置情况，提高了测试效率的同时，使得游戏可完成度增高。

无敌模式的开启是设计者用于计算全地图经验值总量以及测试每个位置怪物触发情况。通过加入无敌版本，设计者可以不用考虑游戏属性而进行游戏的运行操作。可以检测玩家角色的各种辅助功能的使用。

**技术难点 5：游戏内容的设计**

设计者在全游戏设计的过程中发现游戏剧情的规划与 NPC 对话框的实现具有一定的难度。设计者本打算采用 hwnd 窗口句柄进行对话框的实现，但发现如果使用 hwnd 会使得整体运行过程中弹出大量窗口，影响游戏流畅度，可能会导致卡顿。

**实现方案 5：文本类函数的实现。**

设计者建立文本类函数 CPP 文件，在该文件中进行剧情的设计。最终采取了步步吸引与趣味式通关提醒的模式，实现了内容与趣味并行的游戏模式。文本类函数采取统一格式与操作方法，使得程序可读性大大提高。

**尚未实现的功能以及未来优化展望：**

由于设计者第一次学习编程知识，缺乏算法基础与相关程序设计经验。因此本款游戏有许多功能并未上线。但设计者已有完善的构思与一定的优化想法。同时在本款游戏的设计过程中设计者也对于程序设计、游戏策划有了一定的了解与认识，改变了过去只能玩不能设计的能力缺陷。

未来在笔者学习更多编程知识后，设计者计划将购买装备、提升等级、学习技能、动画式对战显示、3D 游戏场景等更为复杂的交互模式与显示模式加入到勇闯博雅塔的游戏当中。同时希望将本款游戏的设计作为我编程学习与锻炼的开始，在未来编程学习过程中我将会不断提高程序的完善性与设计的完整性。

感谢课程老师的帮助、感谢课程助教王勇和余翔的指导、感谢课程同学的帮助。让我制作完成这样一份充满成就感的大作业。