**数据库分工（陈文杰）**

1. **后端设计方面**

完成了基于数据库管理系统的游戏应用（“马保国勇闯南大”）的后端开发设计，根据设定的游戏机制，利用Python语言实现了游戏的主体功能。后端的内容如下：

1. **数据库连接**

利用pymssql中的GetConnect()函数连接本地的数据库DBMS，编写查询语句

cur.execute(“select \* from ...”)，读取MBG数据库中player、figure、card、skill等数据表同时根据需要做表连接返回查询结果，在Python内存中生成相应的DataFrame（数据框）结构数据，为程序的运行提供基本的数据支撑。

1. **程序主体设计**

采用“自顶向下，模块化设计”的程序设计思路，对游戏应用进行解构，根据数据流图和游戏流程图，分离出其需要实现的各项子功能，具体包括：

①人物、卡牌、技能初始化（通过”gne\_card（）”、”gne\_roe（）”、”gne\_skill（）”实现）；

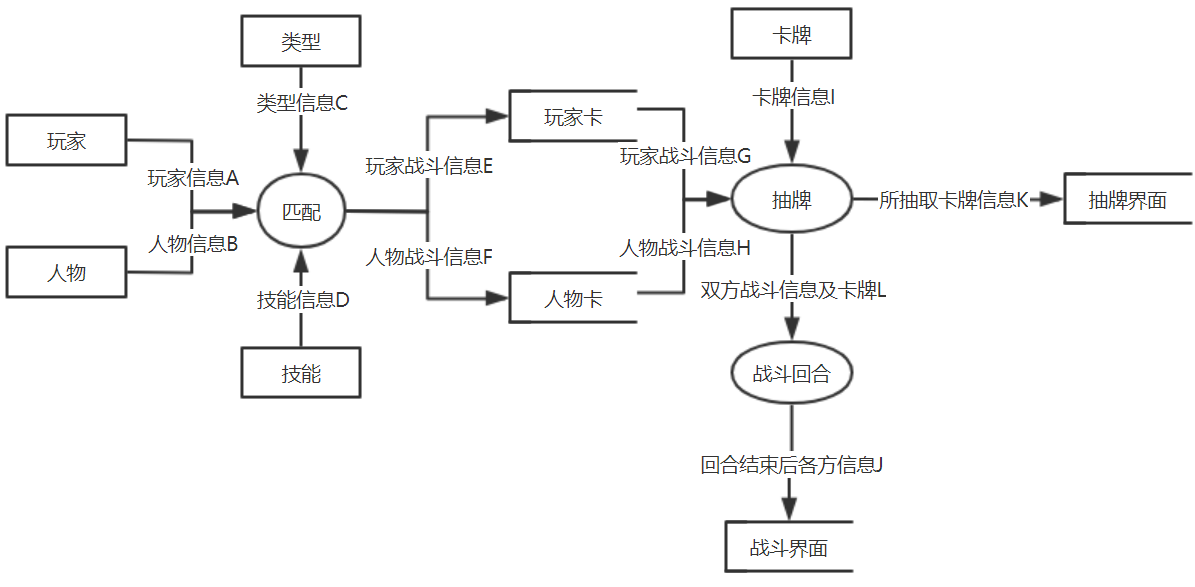
②场景更换与玩家匹配（通过”scene（）、”role\_match（）”实现）；

③随机抽牌、发牌（通过”card\_shuffle( )”、”get\_card( )”实现）；

④人物战斗（涉及到技能与卡牌的使用，通过”use\_card( )”、”use\_skill( )”实现）；

⑤结果反馈（与人物战斗同步，涉及到对人物属性的总体更改，通过”info\_show\_( ...)”系列函数实现）

图 1 游戏流程图



1. **面向对象编程的应用**

在数据库设计的过程中，我们绘制了ER图，明确了此数据库管理系统的实体，实体属性，以及实体之间的相互关系，完成了数据库系统的抽象建构。

在后端程序设计时，我也采用了这种思路，将人物、技能、类型、卡牌等抽象为面向对象程序设计中的“类”概念，在Python中生成了“Role”、“Skill”、“Card”等Class，每个类都对应有自己的数据成员（即属性），也对应有自己的成员函数（即行为），从而完成封装。同时利用数据库中的数据记录，完成相应类的实例的创建。总的看来面向对象编程与关系数据库的原理有异曲同工之妙。

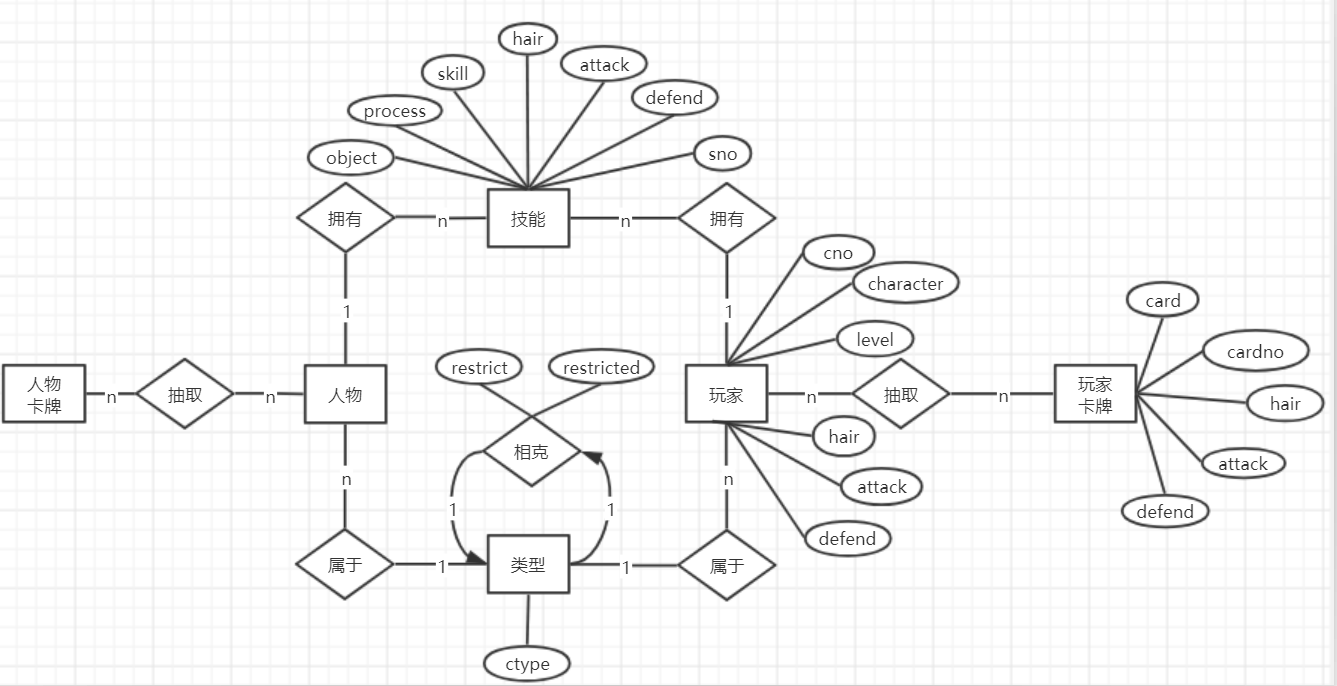


图 2 数据库ER图（面向对象OOP）

1. **与前端的对接**

模块化编程，由”\_\_main\_\_\_”函数实现统一调度，也相应地为前端提供相应的功能接口。

编写info\_show\_( )系列函数，为前端界面信息的展示提供相应的获取渠道。

例如，前端需要显示人物当前的信息，那么可以调用我的内置函数，获得人物的名称、发量、攻击值、防御值、类型等属性，并映射到QT界面进行实时展示。

1. **数据库读写**

此前所叙述的程序模块主要是将数据读取进内存，并直接在内存中运行，根据用户行为进行数据的操作。

但数据库管理系统涉及到更新操作，所以我们还编写程序实现对数据库的实时更新，具体形成了玩家“存档/独挡”的游戏功能。在玩家通过一个场景后，玩家可以选择存档并退出，这时后端利用Python语言将缓存中的数据写入数据库相应的存档表中。玩家再次进入游戏时可以选择读档，同时Python读入上次存储的数据，便于用户继续游戏。

1. **数据库设计方面**

**1）数据库建构的讨论与设计**

与小组成员一起讨论确定数据库的构成，ER中主要的实体、实体属性与实体间关系；

确定组成数据库的几张基本表，表的基本属性与属性取值。

**2）游戏机制的设计**

讨论确定游戏应用的总体流程，分析得出实现游戏所需要的模块以及所需的数据与功能，并基于此进行游戏的整体设计。