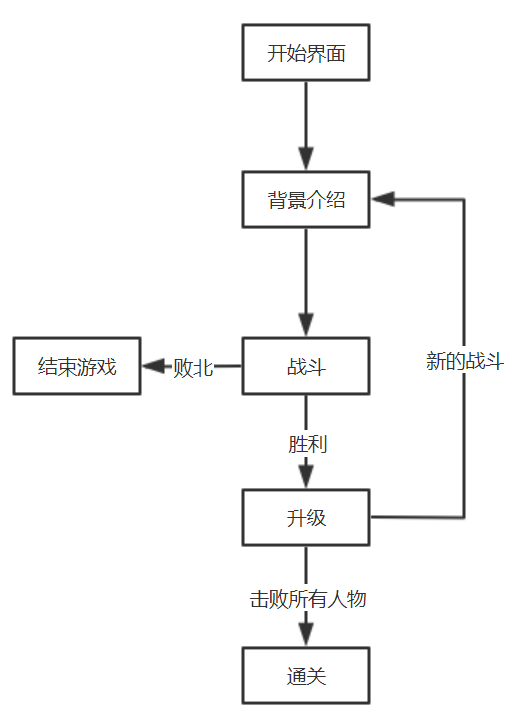
一个基于数据库的卡牌类游戏设计

数据库设计报告

1. 需求分析

（一）收集用户需求

本次大作业我们将设计一个基于数据库的卡牌类游戏。游戏的基本流程如下图所示，经过开始界面后，会有每次的战斗，和不同的人物进行PK，成功后循环进入下一个战斗，失败则结束游戏；成功击败所有人物则闯关成功。这些部分，包括游戏的机制和前端等部分，我们将使用python来实现。



通过数据库实现的部分主要在于对战斗的信息进行管理、查询和更新，这一部分是用户需求的重点。

打斗的规则有：

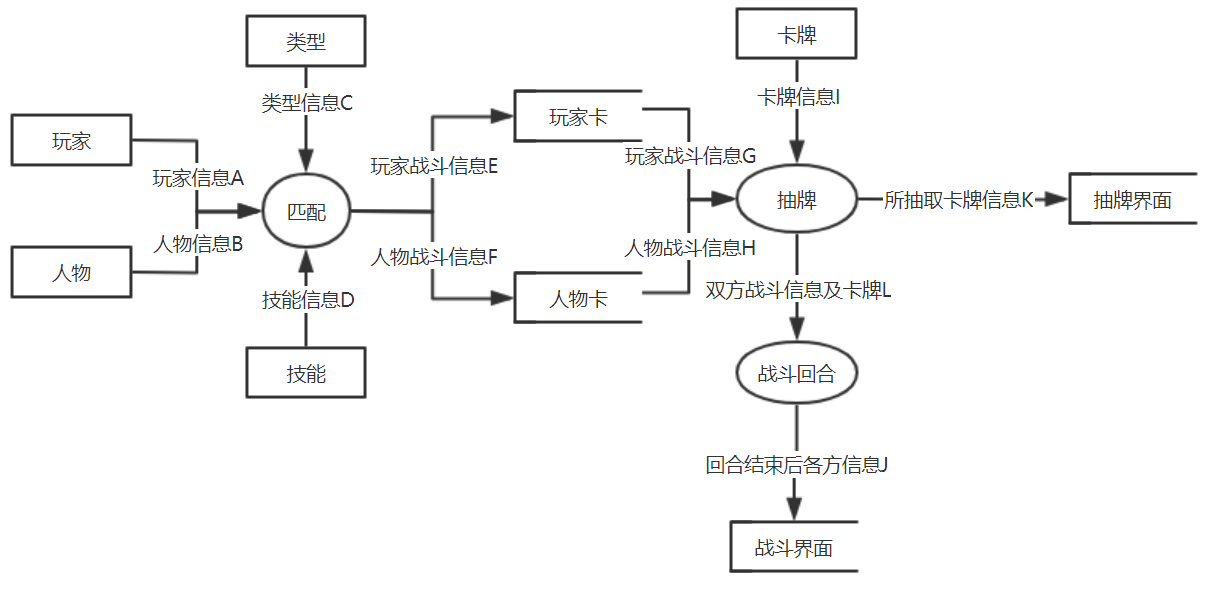
1. 玩家和人物打斗的过程中，玩家首先开始一轮回合，抽取三张卡牌，对自己或对方的的攻、防、血量等造成影响。三张卡牌的操作结束后，玩家回合结束，人物回合开始，同样抽取三张卡牌并进行操作。
2. 玩家和人物都具有各自的技能，这些技能是依附于人物和玩家一直起作用，区别于只在一个回合发生作用的卡牌。技能是对自己或对方的的攻、防、血量，或者对卡牌的属性造成影响。
3. 玩家和人物都具有各自的属性，属性与属性之间是相克的，属性被克制的人物的技能在战斗中失效。
4. 玩家和人物对战过程中，只要有一方的血量将为零则战斗结束，战斗结束后血量值不为零的一方获胜。

二、概念结构设计

（一）数据字典

1.数据流图

通过用户的需求描述，我们总结为如下的数据流图来描述一个战斗回合：



2.数据项描述

2.1 cno

数据项：cno

含义说明：唯一标识每个人物和玩家

别名：人物/玩家编号

类型：字符型

长度：8

取值范围：0至99999999

取值含义：目前为各个人物或玩家的出现次序的顺序号，0在前为人物，反之为玩家

与其他数据项的逻辑关系：Cno->character, cno->hair, cno->attack, cno->defend, cno->level, cno->ctype，是all-key关系“人技匹配”的外键。

2.2 character

数据项：character

含义说明：人物的名字

别名：人物

类型：人物名

长度：20

取值范围：all

取值含义：人物的名字

与其他数据项的逻辑关系：cno->character

2.3 hair

数据项：hair

含义说明：血量

别名：发量

类型：整型

长度：int

取值范围：0-9999

取值含义：设定血量有限，但随着游戏开发，可能会提高上限

与其他数据项的逻辑关系：cno->hair, sno->hair, cardno->hair

2.4 attack

数据项：attack

含义说明：显示攻击值，数值是对目标的血量的减少值

别名：攻击值

类型：整型

长度：int

取值范围：0-9999

取值含义：设定攻击值有限，但随着游戏开发，可能会提高上限

与其他数据项的逻辑关系：cno->attack, sno->attack, cardno->attack

2.5 defend

数据项：defend

含义说明：显示防守值，数值是对攻击数值的减少值

别名：防守值

类型：整型

长度：int

取值范围：0-9999

取值含义：设定防守值有限，但随着游戏开发，可能会提高上限

与其他数据项的逻辑关系：cno->defend, sno->defend, cardno->defend

2.6 level

数据项：level

含义说明：显示级数，数值人物或者玩家的级别

别名：级别

类型：整型

长度：int

取值范围：0-9999

取值含义：设定级别数有限，但随着游戏开发，可能会提高上限

与其他数据项的逻辑关系：cno->level

2.7 ctype,restrict,restricted

数据项：ctype

含义说明：人物的类型属性

别名：类型/克制/被克制

类型：字符型

长度：2

取值范围：(“智”,”攻”,”防”)

取值含义：智代表智力型人物或玩家，防代表防御型人物或玩家，攻代表攻击型人物或玩家

与其他数据项的逻辑关系：cno->ctype

2.8 sno

数据项：sno

含义说明：唯一标识每一个技能

别名：技能号

类型：字符型

长度：8

取值范围：字母和数字

取值含义：标识号

与其他数据项的逻辑关系：sno->skill, sno->hair, sno->attack, sno->defend, sno->process, sno->object, 是all-key关系“人技匹配”的外键。

2.9 skill

数据项：skill

含义说明：技能的名字

别名：技能

类型：字符串

长度：20

取值范围：all

取值含义：技能的名字

与其他数据项的逻辑关系：sno->skill

2.10 process

数据项：process

含义说明：记录技能的各种值对对象产生作用的运算

别名：运算操作

类型：字符型

长度：2

取值范围：(“+”,”-”,”\*”,”/”)

取值含义：+为加操作，-为减操作，\*为乘操作，/为除操作

与其他数据项的逻辑关系：sno->process

2.11 object

数据项：object

含义说明：是卡牌操作的作用对象

别名：作用对象

类型：字符型

长度：20

取值范围：(“card”,”character”)

取值含义：card是指对象为卡牌，character是指对象为人物

与其他数据项的逻辑关系：sno->object

2.12 cardno

数据项：cardno

含义说明：卡牌号

别名：唯一标识卡牌

类型：字符型

长度：8

取值范围：0-9999

取值含义：标识卡牌

与其他数据项的逻辑关系：cardno->card, cardno->hair, cardno->attack, cardno->defend

2.13 card

数据项：card

含义说明：卡牌的名称

别名：卡牌名

类型：字符型

长度：20

取值范围：all

取值含义：卡牌的名称

与其他数据项的逻辑关系：cardno->card

3.数据结构描述：

3.1 人物

数据结构：人物

含义说明：定义了一个人物的有关信息

组成：cno, character, hair, attack, defend, level

3.2 玩家

数据结构：玩家

含义说明：定义了一个玩家的有关信息

组成：cno, character, hair, attack, defend, level

3.3 人物类型

数据结构：任务类型

含义说明：说明了玩家和类型的匹配关系

组成：cno, ctype

3.4 类型相克

数据结构：类型相克

含义说明：规定了几种人物类型之间的相克关系

组成：restrict, restricted

3.5 技能

数据结构：技能

含义说明：定义了一个技能的相关信息

组成：sno, skill, hair, attack, defend, process, object

3.6 人技匹配

数据结构：人技匹配

含义说明：规定了人物和技能之间的匹配关系，即什么人物拥有什么技能

组成：cno, sno

3.7 人物卡牌

数据结构：人物卡牌

含义说明：定义了一个人物卡牌的相关信息，任务卡牌是人物可以抽取的卡牌

组成：cardno, card, hair, attack, defend

3.8 玩家卡牌

数据结构：玩家卡牌

含义说明：定义了一个玩家卡牌的相关信息，玩家卡牌是玩家可以抽取的卡牌

组成：cardno, card, hair, attack, defend

4.数据流：

4.1 玩家信息

数据流：A

说明：将玩家信息提供战斗匹配

数据流来源：玩家

数据流去向：匹配

组成：cno, character, hair, attack, defend, level

4.2 人物信息

数据流：B

说明：将人物信息提供战斗匹配

数据流来源：人物

数据流去向：匹配

组成：cno, character, hair, attack, defend, level

4.3 类型信息

数据流：C

说明：将类型和人物之间的匹配关系以及类型之间相克关系传递给战斗匹配

数据流来源：类型

数据流去向：匹配

组成：cno, ctype, restrict, restricted

4.4 技能信息

数据流：D

说明：将技能和人物之间的匹配关系以及技能的相关属性传递给战斗匹配

数据流来源：技能

数据流去向：匹配

组成：cno, sno, skill, hair, attack, defend, process, object

4.5 玩家战斗信息

数据流：E，G

说明：是玩家和技能匹配以及自己的类别信息

数据流来源：匹配

数据流去向：玩家卡

组成：玩家(cno, character, hair, attack, defend, level), 技能(sno, skill, hair, attack, defend, process, object), ctype, restricted

4.6 人物战斗信息

数据流：F，H

说明：是人物和技能匹配以及自己的类别信息

数据流来源：匹配

数据流去向：人物卡

组成：人物(cno, character, hair, attack, defend, level), 技能(sno, skill, hair, attack, defend, process, object), ctype, restricted

4.7 卡牌信息

数据流：I

说明：人物卡牌和玩家卡牌的各项信息

数据流来源：卡牌

数据流去向：抽牌

组成：cardno, card, hair, attack, defend

4.8 双方战斗信息及卡牌

数据流：L

说明：双方当前的各个属性值、类型、技能的相关信息、所抽取卡牌的相关信息

数据流来源：抽牌

数据流去向：战斗回合

组成：人物(cno, character, hair, attack, defend, level), 玩家(cno, character, hair, attack, defend, level), 技能(sno, skill, hair, attack, defend, process, object), ctype,restricted, 卡牌(cardno, card, hair, attack, defend)

4.9 回合结束后各方信息

数据流：J

说明：战斗回合结束后，双方的各类属性值

数据流来源：战斗回合

数据流去向：战斗界面

组成：人物(cno, character, hair, attack, defend, level), 玩家(cno, character, hair, attack, defend, level)

4.10 所抽取卡牌信息

数据流：K

说明：所抽取卡牌的各项信息

数据流来源：抽牌

数据流去向：抽牌界面

组成：cardno, card, hair, attack, defend

5.数据存储

5.1 玩家卡

数据存储：玩家卡

说明：记录玩家参与战斗时的各项属性

流入数据流：E

流出数据流：G

组成：玩家(cno, character, hair, attack, defend, level), 技能(sno, skill, hair, attack, defend, process, object), ctype, restricted

数据量：每个战斗开始前一张

存取方式：用python实现部分，呈现到界面上

5.2 人物卡

数据存储：人物卡

说明：记录人物参与战斗时的各项属性

流入数据流：F

流出数据流：H

组成：人物(cno, character, hair, attack, defend, level), 技能(sno, skill, hair, attack, defend, process, object), ctype, restricted

数据量：每个战斗开始前一张

存取方式：用python实现部分，呈现到界面上

5.3 抽牌界面

数据存储：抽牌界面

说明：记录双方所抽取的各三张牌的信息

流入数据流：K

流出数据流：无

组成：cardno, card, hair, attack, defend

数据量：每一个回合3张

存取方式：用python实现部分，呈现到界面上

5.4 战斗界面

数据存储：战斗界面

说明：记录一个回合结束之后的各方西信息

流入数据流：J

流出数据流：无

组成：人物(cno, character, hair, attack, defend, level), 玩家(cno, character, hair, attack, defend, level)

数据量：每个回合结束时两张

存取方式：用python实现部分，呈现到界面上

6.处理过程

6.1匹配

处理过程：匹配

说明：对于玩家和人物的战斗信息进行匹配

输入：ABCD

输出：EF

处理：根据玩家表、人物表、人物类型表、类型相克表、技能表、人机匹配表，来匹配玩家和人物的战斗信息。

6.2 抽牌

处理过程：抽牌

说明：给回合开始方抽取三张卡牌

输入：GHI

输出：LK

处理：如果是玩家的回合，就从玩家卡牌随机抽取三张卡牌；如果是人物的回合，就从人物卡牌随机抽取三张卡牌

6.3 战斗回合

处理过程：战斗回合

说明：使用三张卡牌进行攻击、防御、加血操作等

输入：L

输出：J

处理：用卡牌的信息对双方的相对应的属性值进行运算

（二）概念结构模型

我们将概念抽象为六个实体：人物、人物卡牌、玩家、玩家卡牌、技能、类型。

根据需求分析，我们抽象出七个联系：

①人物抽取人物卡牌：是多对多的联系，一个人物可以抽取多个卡牌，一个卡牌可以被多个人物抽取。这个是m:n的联系。

②玩家抽取玩家卡牌：是多对多的联系，一个玩家可以抽取多个卡牌，一个卡牌可以被多个玩家抽取。这个是m:n的联系。

③人物属于某种类型：是多对一的联系，一个人物属于固定的一种类型，一种类型下可以包含多个人物。这个是n:1的联系。

④玩家属于某种类型：是多对一的联系，一个玩家属于固定的一种类型，一种类型下可以包含多个玩家。这个是n:1的联系。

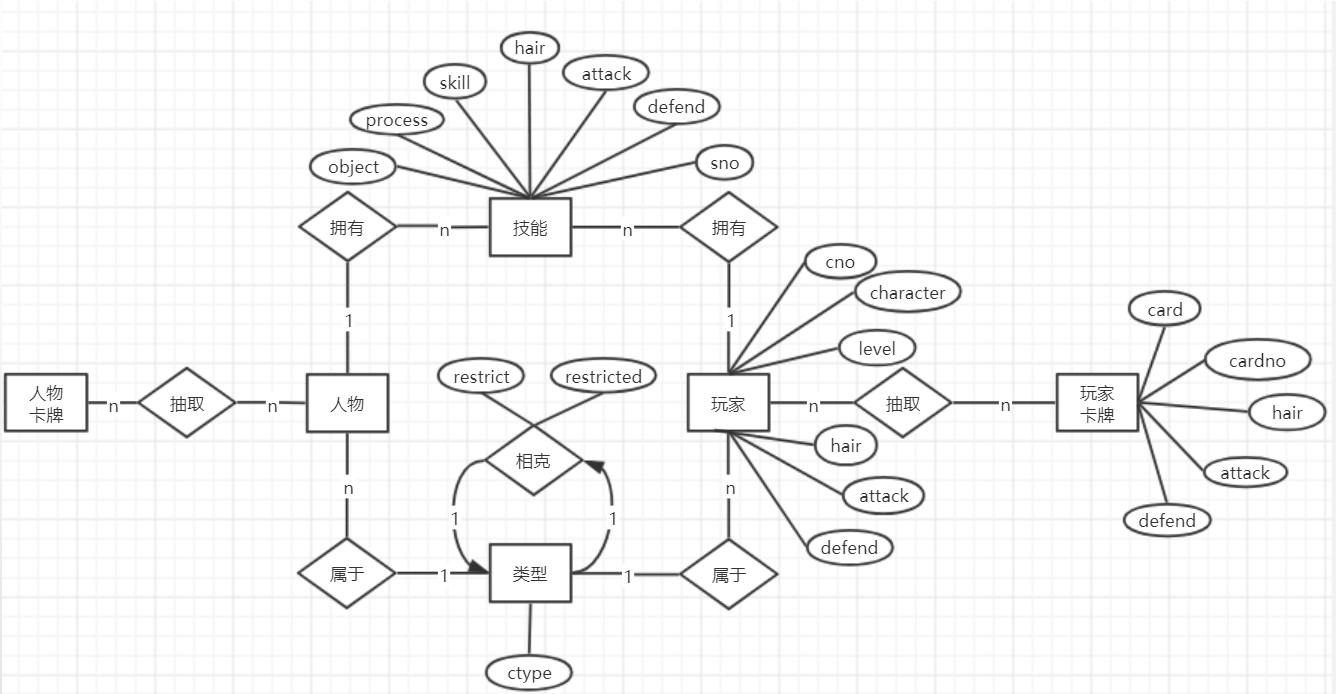
⑤人物拥有某些技能：是一对多的联系，一个人物可以拥有多个技能，一种技能只能属于一个玩家。这个是1:n的联系。

⑥玩家拥有某些技能：是一对多的联系，一个玩家可以拥有多个技能，一种技能只能属于一个玩家。这个是1:n的联系。

⑦类型与类型相克：是一对一的联系，一种类型克制另一种类型，每一种类型也被一种类型克制。这个是1：1的联系。

在这个概念结构模型中，三种关系（一对一，一对多，多对多）都被应用到联系当中。

再将前面在数据字典中定义的数据项作为属性添加到相应的主体和联系上，特别注意的是restrict和restricted是在联系“相克”上的属性。得到的E-R图如下：



三、逻辑结构设计

按照概念结构模型的设计构建关系数据模型。经过优化，得到8个关系：

人物(cno, character, hair, attack, defend, level)

玩家(cno, character, hair, attack, defend, level)

人物类型(cno, ctype)

类型相克(restrict, restricted)

技能(sno, skill, hair, attack, defend , process, object)

人技匹配(cno,sno)

人物卡牌(cardno, card, hair, attack, defend)

玩家卡牌(cardno, card, hair, attack, defend)

其中：人物、玩家、类型相克、技能、人物卡牌和玩家卡牌都是非主属性完全函数依赖于主键，而人物类型和人技匹配为all-key。此逻辑模型做到了：

①所有非主属性对每一个码都是完全函数依赖；

②所有的主属性对每一个不包含它的码，也是完全函数依赖；

③没有任何属性完全函数依赖于非码的任何一组属性。

因此此逻辑结构设计的范式为BC范式。

数据库实施

数据库运行和维护