

湖北第二师范学院

《面向对象程序设计》

课程设计报告书

题目： 纸牌游戏——“21点”

院系： 计算机学院

专业： 网络工程

姓名： 李林

学号： 0850311013

同组人员： 陈易俊、吴鑫、易龙、张帅

指引老师： 张琪

设计时间： 2月26日~3月19日

目 录

一. 课程设计目的	3
二. 问题描述	3
三. 需求分析	4
四. 概要设计	5
4.1 系统中类类型定义及系统各类类成员功能阐明	5
4.2 系统各类间关系	6
五. 具体设计	7
5.1 数据类型定义及伪代码描述	7
5.2 系统重要子程序具体设计（加注释的源程序）	8
六. 测试分析	19
七. 技术难点及创新	19
八. 自己完毕模块阐明	23
九. 源程序清单	23

一. 课程设计目的

在 Visual C++6.0 或者 C++ Builder 6.0 环境中编译运营纸牌游戏——“21 点”，通过编程练习、程序调试、程序纠错等过程训练，真正掌握所学知识、提高编程水平。

二. 问题描述

“21 点”是一种古老的扑克牌游戏，游戏规则是：各个参与者设法使自己的牌达到总分 21 而不超过这个数值。扑克牌的分值取它们的面值，A 充当 1 分或者 11 分（由玩家自己选择一种分值），J、Q 和 K 人头牌都是 10 分。庄家对付 1 到 7 个玩家。在一局开始时，涉及庄家在内的所有参与者均有两张牌。玩家可以看到他们所有牌以及总分，而庄家有一张牌临时是隐蔽的。接下来，只有乐意，各个玩家均有机会依次再拿一张牌。如果某个玩家的总分超过了 21（称为“引爆”），那么这个玩家就输了。在所有玩家都拿了额外的牌后，庄家将显示隐藏的牌。只要庄家的总分等于或小于 16，那么他就必须再拿牌，如果庄家引爆了，那么还没有引爆的所有玩家都将获胜。引爆的玩家打成平局。否则，将余下的各玩家的总分与庄家的总分做比较，如果玩家的总分大于庄家的总分，则玩家获胜。如果两者的总分相似，则玩家与庄家打成平局。

(1) 将所有的函数与相应的数据封装到类中，并改写主程序，使程序简化。

(2) 序中一共有 13 张扑克牌可以随机抽取，大于 10 的点数为 10，A 根据状况点数不同，目前点数小于 11 时为 11 点，大于 10 时则为 1 点。

(3) 规定庄家永远不会暴庄，即庄家可以查看下一张牌，若大于 21 点，则回绝，固然，在游戏规则上不能泄露这点秘密。

(4) 将界面的提示阐明所有设计为中文界面，界面的解释具体和谐，可以吸引更多的玩家。

三. 需求分析

设计程序的重要目的是编写程序模拟游戏。游戏为纸牌游戏——“21 点，”也叫“Black Jack”即黑杰克，该游戏合用于手机，也是网上比较流行的小游戏之一。游戏以计算机作为庄家，1—7 个人作为一般玩家参与游戏。可用字符代表扑克牌面值（A，2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K）和花色（梅花 C，方块 D，红心 H，黑桃 S），模拟实质扑克进行游戏。以便大众在空闲时一种人就可以达到多人娱乐的效果！

四. 概要设计

4.1 系统中类类型定义及系统各类类成员功能阐明

“21 点游戏”程序中的类

类	基类	阐明
Card	无	一张扑克牌
Hand	无	一手牌, Card 对象集合
Deck	Hand	一副牌它具有 Hand 所不具有的额外功能, 如洗牌和发牌
GenericPalayer	Hand	一种抽象玩家, 具有庄家和玩家的共同元素
Player	GenericPalayer	一般玩家
House	GenericPalayer	庄家
Game	无	一种游戏

4.2 系统各类间关系

Card 类

成员	阐明
rank m-Rank	牌的面值。Rank 是 13 个面值等级的枚举
suit m-Suit	牌的花色。Suit 是 4 花色的枚举
bool m-IsFace	标示牌与否正面朝上的状态，正面朝上为 true，否则 false
int GetVable ()	返回牌面值所相应的分值

Hand 类

成员	阐明
vector<card*>m-Cards	牌面向量，存储指向 Card 对象指针
void Add (Card*pCard)	向手中添加一张牌，即将一种指向 Card 的指针添加到向量 m-Cards 中
void Clear()	清除手中所有牌，即 删除向量 m-Card 中的所有指针删除所有动态创立 Card 对象
int GetTotal() const	返回这首派总值

GenericPalayer 类（抽象类）

成员	阐明
sring m-Name	通用玩家的名字
vrtual bool IsHitting () const=0	指出通用玩家与否想要另一张牌，纯属函数
bool IsBusted () const	指出通用玩家与否引爆
void Bust () const	宣布通用玩家引爆

Player 类

成员	阐明
vrtual bool IsHitting () const	指出玩家与否想要另一张牌
void Win () const	宣布玩家获胜
void Lose () const	宣布玩家失败
void Push () const	宣布玩家获得平局

House 类

成员	阐明
<code>virtual bool IsHitting () const</code>	指出玩家与否再拿一张牌

Deck 类

成员	阐明
<code>void Populate ()</code>	创立一共 52 张的原则扑克牌
<code>void Shuffle ()</code>	洗牌
<code>void Deal (Hand&Hand)</code>	将一张牌发到一手中
<code>void AdditionalCards (GenericPalayer&GenericPala yer)</code>	只要通用玩家可以并且想要另一张牌,则在给他发一张牌

Game 类

成员	阐明
<code>Deck m-Deck</code>	一副牌
<code>House m-House</code>	庄家
<code>vector<Player>m-Player</code>	一般玩家的集合, Player 对象的一种向量
<code>void Play ()</code>	玩一局游戏

五. 具体设计

5.1 数据类型定义及伪代码描述

1. 一方面定义表达扑克牌花色与面值的枚举类型, 这样程序可读性更强
2. 另一方面, 措施 **Game**

() 用来运营游戏，下面为用伪代码描述的整个游戏的流程：

洗牌

给庄家和玩家发最初的二张牌

隐藏庄家的第一张牌

显示庄家和玩家手中的牌

依次向玩家发额外的牌

显示庄家的第一张牌

给庄家发额外的牌

if(庄家引爆)

 没有引爆的玩家赢

 引爆的玩家平局

else

 for(每个玩家)

 if(玩家没有引爆且玩家的总分比庄家大) 宣布玩家赢

 else if(玩家的总分与庄家相等) 宣布平局

 else 宣布玩家赢

5.2 系统重要子程序具体设计（加注释的源程序）

(1) 新建头文献 Card.h，为一张单独的扑克牌定义 Card 类。

```
#ifndef _Card_h_
#define _Card_h_
#include<iostream>
using namespace std;
```



```

class Card
{
public:

enum
rank{ACE=1,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,JACK,QUEEN,KING};
enum suit{CLUBS,DIAMONDS,HEARTS,SPADES};
Card(rank r=ACE,suit s=SPADES,bool ifu=true);
//friend ostream& operator<<(ostream& os,const Card& aCard);
int GetValue() const;
void Flip();

rank m_Rank;
suit m_Suit;
bool m_IsFaceUp;
};
#endif

```

(1) 新建头文件 Card.h，实现 Card 类。

```

#include "Card.h"
Card::Card(rank r,suit s,bool ifu):m_Rank(r),m_Suit(s),m_IsFaceUp(ifu)
{}
int Card::GetValue() const
{int value=0;
if(m_IsFaceUp)
{value=m_Rank;
if(value>10)value=10;
}
return value;
}
void Card::Flip()
{m_IsFaceUp=!m_IsFaceUp;
}

```

(2) 新建头文件 Hand.h，为纸牌的集合定义 Hand 类。

```

#ifndef _Hand_h_
#define _Hand_h_
#include "Card.h"
#include <vector>
using namespace std;
class Hand
{public:
Hand();

```

```

virtual ~Hand();
void Add(Card* pCard);
void Clear();
int GetTotal() const;
public:
    vector<Card*>m_Cards;};
#endif

```

(3) 新建 Cpp 文献 Hand.cpp, 实现 Hand 类。

```

#include "Hand.h"
Hand::Hand()
{m_Cards.reserve(7);}
Hand::~~Hand()
{Clear();}
void Hand::Add(Card* pCard)
{m_Cards.push_back(pCard);}
void Hand::Clear()
{vector<Card*>::iterator iter=m_Cards.begin();
for(iter=m_Cards.begin();iter!=m_Cards.end();++iter)
{delete *iter;
*iter=0;
}
m_Cards.clear();
}
int Hand::GetTotal() const
{if(m_Cards.empty()) return 0;
if(m_Cards[0]->GetValue()==0)
return 0;
int total=0;
vector<Card*>::const_iterator iter;
for(iter=m_Cards.begin();iter!=m_Cards.end();++iter)
total+=(*iter)->GetValue();
bool containsAce=false;
for(iter=m_Cards.begin();iter!=m_Cards.end();++iter)
if((*iter)->GetValue()==Card::ACE)
containsAce=true;
if(containsAce&&total<=11)
total+=10;
return total;
}

```

(4) 新建头文献 GenericPlayer.h, 为通用玩家定义抽象的 GenericPlayer 类。

```

#ifndef _GenericPlayer_h_
#define _GenericPlayer_h_
#include "Hand.h"
#include<string>
using namespace std;
class GenericPlayer:public Hand
{
public:
    friend ostream &operator<<(ostream &os,const GenericPlayer &aGenericPlayer);
    GenericPlayer(const string& name=" ");
    virtual~GenericPlayer();
    virtual bool IsHitting()const=0;
    bool IsBusted() const;
    void Bust() const;

    string m_Name;
};
#endif

```

(5) 新建 cpp 文献 GenericPlayer.cpp，实现 GenericPlayer 类的成员函数。

```

#include "stdafx.h"
#include "GenericPlayer.h"

GenericPlayer::GenericPlayer(const string& name):m_Name(name)
{

}

GenericPlayer::~~GenericPlayer()
{

}

bool GenericPlayer::IsBusted() const
{
    return (GetTotal(>21);
}

void GenericPlayer::Bust()const
{
    cout<<m_Name<<"引爆!\n";
}

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/288047103021006100>