

C/C++程序设计案例实战

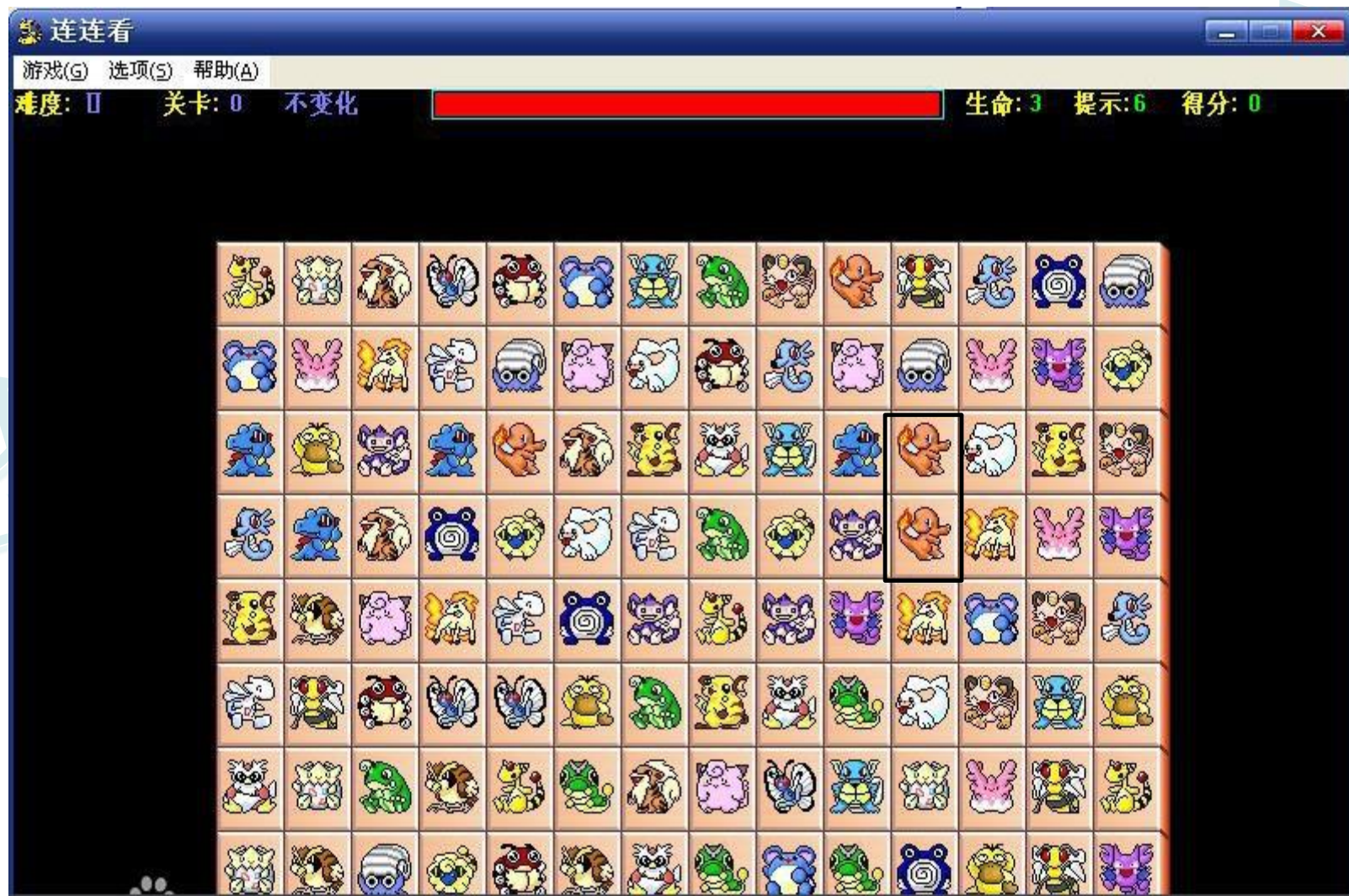
——连连看和猜数游戏

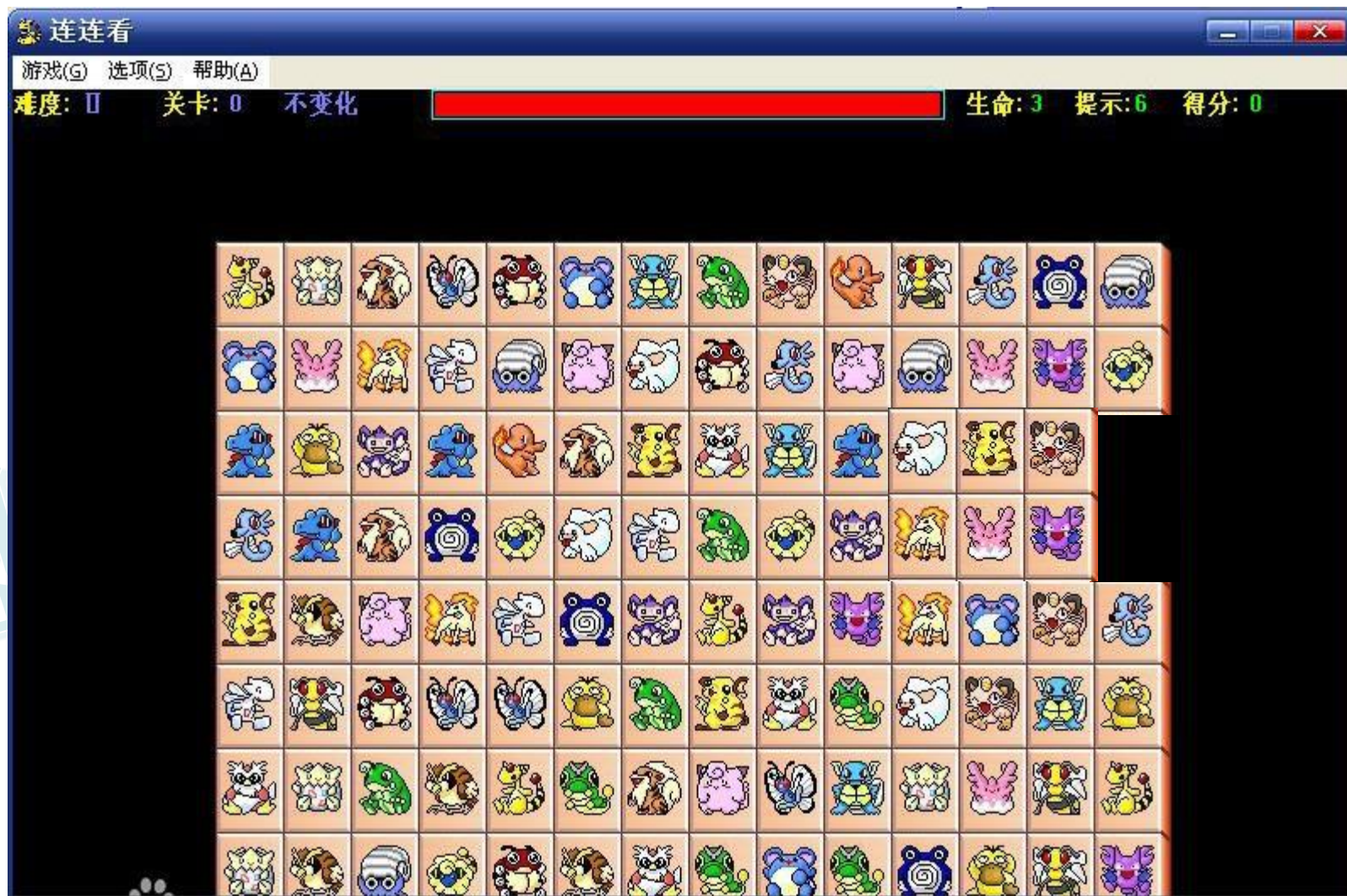
华中农业大学信息学院 李小霞

私

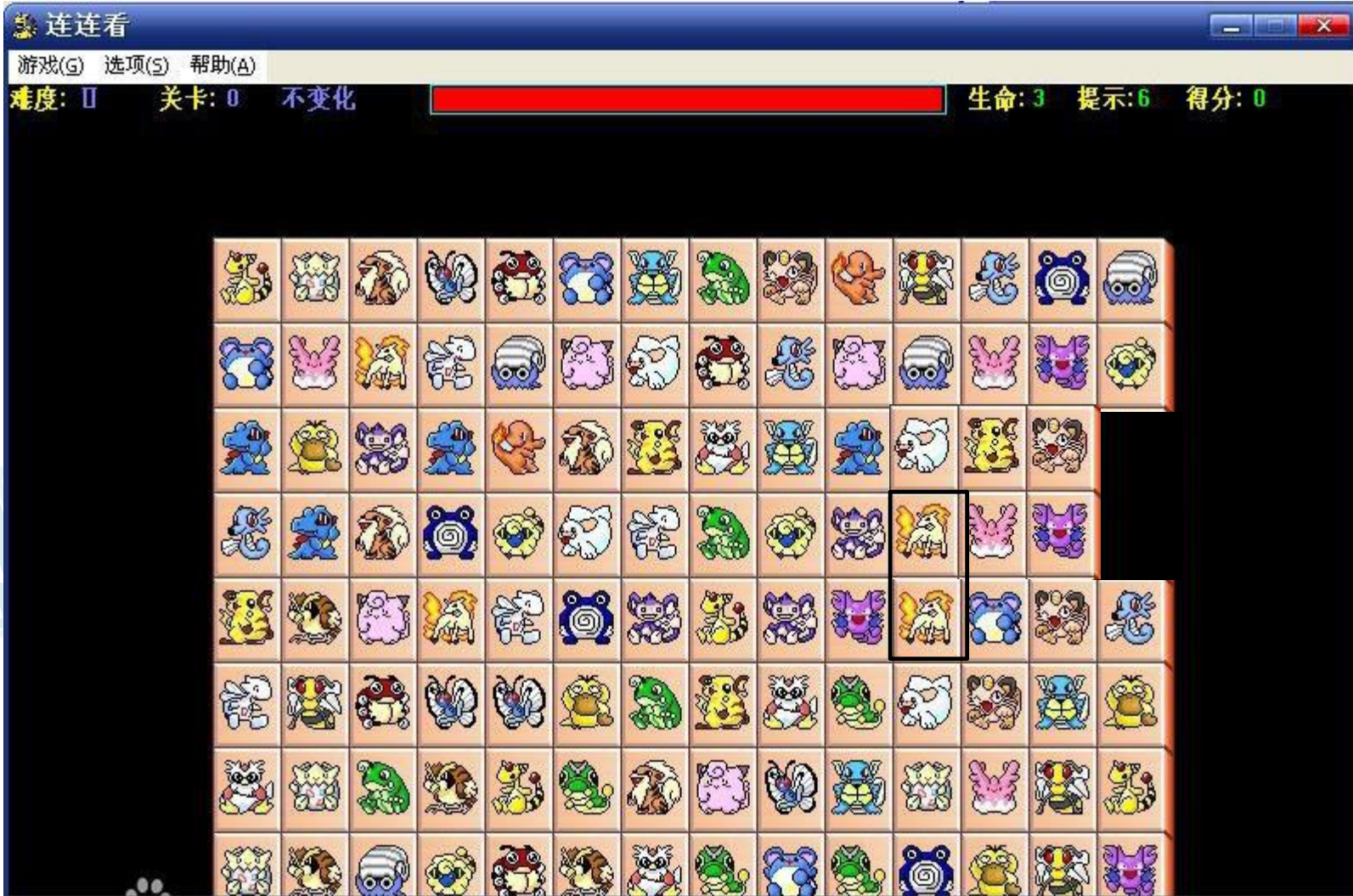


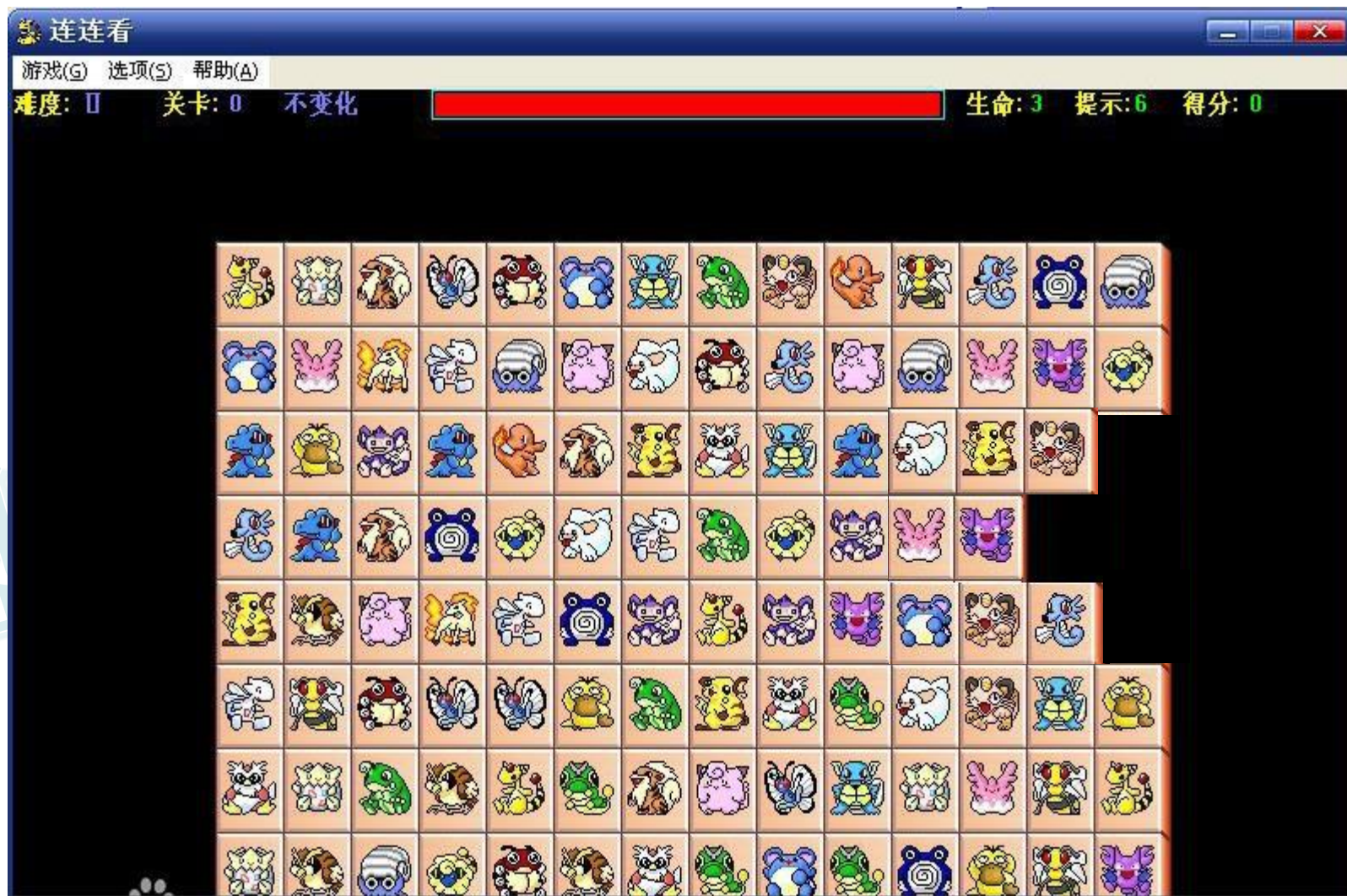
问题引入



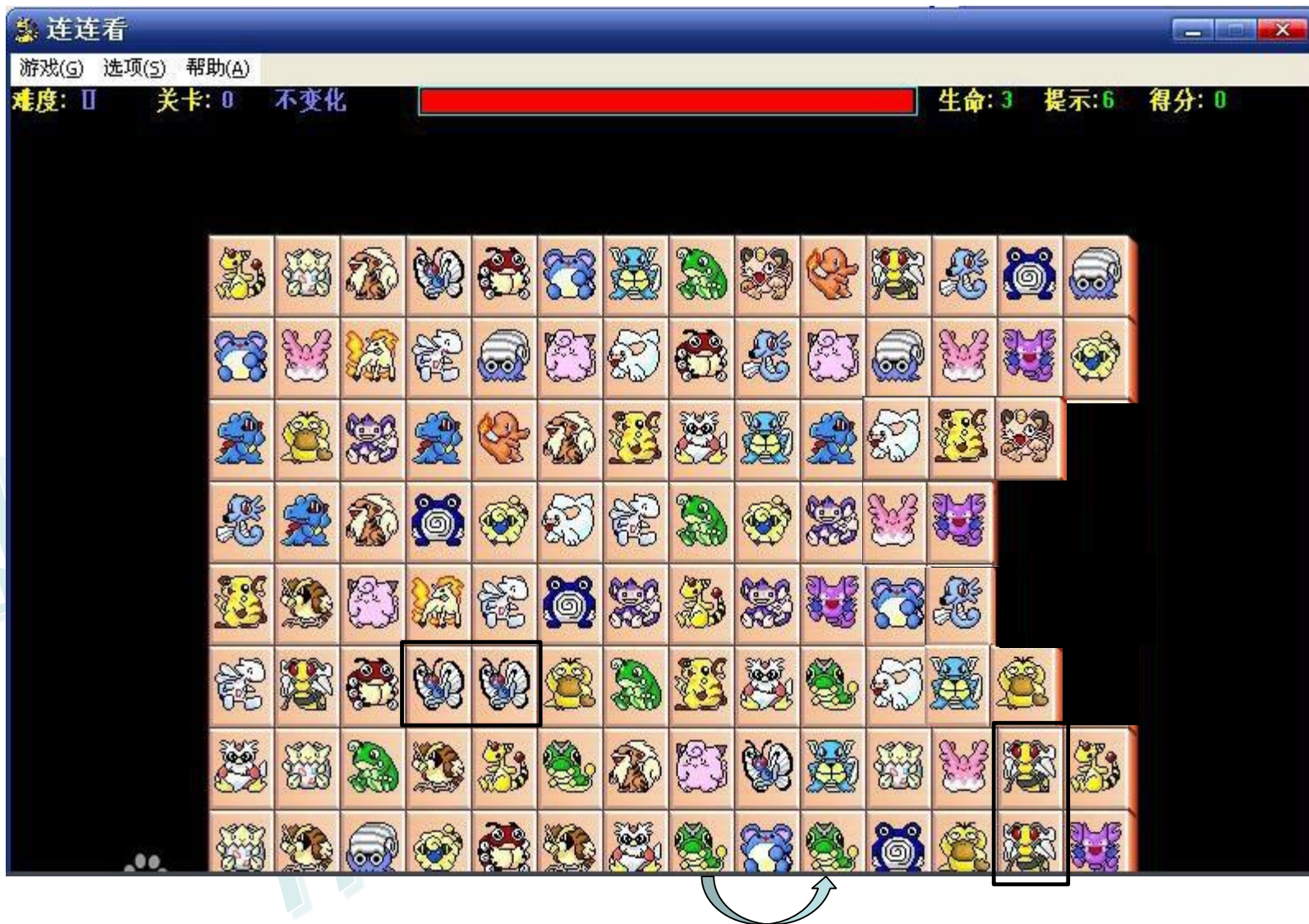


获取坐标—连线—擦除—移动

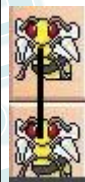




获取坐标—连线—擦除—移动




```
1  int main()  
2  {  
3      获取坐标指令代码;  
4      连接指令代码;  
5      擦除指令代码;  
6      移动指令代码;  
7      获取坐标指令代码;  
8      连接指令代码;  
9      擦除指令代码;  
10     移动指令代码;  
11     .....  
227   获取坐标指令代码;  
228   连接指令代码;  
229   擦除指令代码;  
230   移动指令代码;  
231   return 0;  
232 }
```

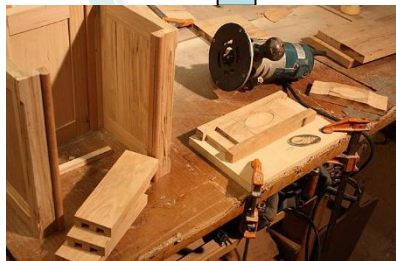


有没有简单的方法，将用于消除相同动物的代码只写一次？



函数的定义和调用

返回值类型 函数的名称 (参数列表) {
具体实现代码;
}



```

1 void eraser()
2 {
3     获取坐标指令代码;
4     连接指令代码;
5     擦除指令代码;
6     移动指令代码;
6     return;
7 }
8 int main()
9 {
10    eraser();
11    eraser();
12    .....
65    eraser();
66    return 0;
67 }

```

VS

```

1 int main()
2 {
3     获取坐标指令代码;
4     连接指令代码;
5     擦除指令代码;
6     移动指令代码;
7     获取坐标指令代码;
8     连接指令代码;
9     擦除指令代码;
10    移动指令代码;
11    .....
227  获取坐标指令代码;
228  连接指令代码;
229  擦除指令代码;
230  移动指令代码;
231  return 0;
232 }

```




```

1 void eraser()
2 {
3     获取坐标指令代码;
4     连接指令代码;
5     擦除指令代码;
6     移动指令代码;
6     return;
7 }
8 int main()
9 {
10    eraser();
11    eraser();
12    .....
65    eraser();
66    return 0;
67 }

```

VS

```

1 int main()
2 {
3     获取坐标指令代码;
4     连接指令代码;
5     擦除指令代码;
6     移动指令代码;
7     获取坐标指令代码;
8     连接指令代码;
9     擦除指令代码;
10    移动指令代码;
11    .....
227    获取坐标指令代码;
228    连接指令代码;
229    擦除指令代码;
230    移动指令代码;
231    return 0;
232 }

```



```
1 void eraser()  
2 {  
3     获取坐标指令代码;  
4     连接指令代码;  
5     擦除指令代码;  
6     移动指令代码;  
6     return;  
7 }  
8 int main()  
9 {  
10    eraser();  
11    eraser();  
12    .....  
65    eraser();  
66    return 0;  
67 }
```

函数的定义必须放在主函数前面吗？




```

1 #include<iostream>
2 using namespace std;
3 void eraser();
4
5
6
7
8 int main()
9 {
10     eraser();
11     eraser();
12     .....
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65     eraser();
66     return 0;
67 }
68 void eraser()
69 {
70     获取坐标指令代码;
71     连接指令代码;
72     擦除指令代码;
73     移动指令代码;
74     return;
75 }

```

需要在主函数前面进行原型声明

也可以放在主函数后面

怎样进行函数声明?



```
1 #include<iostream>
2 using namespace std;
3 void eraser();
4
5 int main()
6 {
7     eraser();
8     eraser();
9     .....
10    eraser();
11    return 0;
12 }
13
14 void eraser()
15 {
16     获取坐标指令代码;
17     连接指令代码;
18     擦除指令代码;
19     移动指令代码;
20     return;
21 }
```

返回值类型 函数的名称 (参数列表);

案例进阶

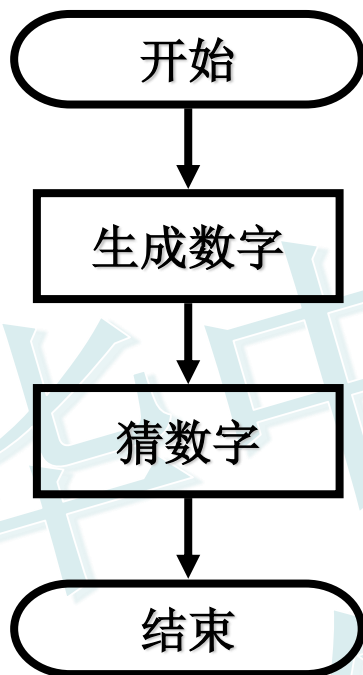


什么游戏规则？

猜一次，猜对：right!
猜错：wrong!并提示
大小



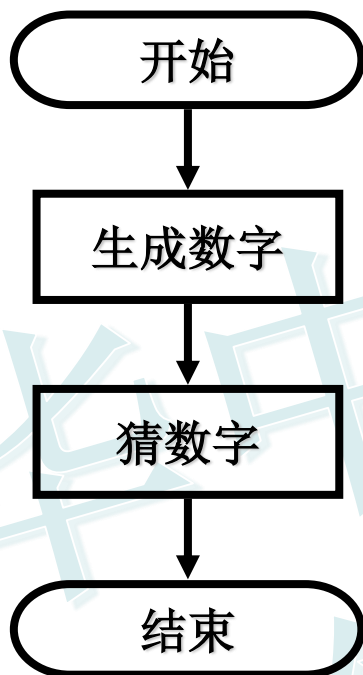
案例分析



```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
int main()
{
    int number;
    srand(time(NULL));
    number=MakeNumber();
    GuessNumber(number);
    return 0;
}
```

生成数字函数

案例分析



```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
int main()
{
    int number;
    srand(time(NULL));
    number=MakeNumber();
    GuessNumber(number);
    return 0;
}
```

猜数字
函数

案例分析

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
int main()
{
    int number;
    srand(time(NULL));
    number=MakeNumber();
    GuessNumber(number);
    return 0;
}
```

函数调用的提示:

1.有返回值时

返回值 = 函数名(实参表列);

2.无返回值时

函数名(实参表列);

案例——代码实现

```
int MakeNumber()  
{  
    int number;  
    srand(time(NULL));  
    number=rand()%100+1;  
    return number;  
}
```


案例——代码实现

```
int MakeNumber()  
{  
    int number;  
    srand(time(NULL));  
    number=rand()%100+1;  
    return number;  
}
```

案例——代码实现

```
int MakeNumber()  
{  
    int number;  
    srand(time(NULL));  
    number=rand()%100+1;  
    return number;  
}
```

案例——代码实现

```
void GuessNumber(int magic)
{
    int guess;
    cout<<"Please guess a magic number:";
    cin>>guess;
    if(guess>magic)
        cout<<"Wrong! Too high!"<<endl;
    else if(guess<magic)
        cout<<"Wrong! Too low!"<<endl;
    else
        cout<<"Right!"<<endl;
}
```


案例——代码实现

```
void GuessNumber(int magic)
{
    int guess;
    cout<<"Please guess a magic number:";
    cin>>guess;
    if(guess>magic)
        cout<<"Wrong! Too high!"<<endl;
    else if(guess<magic)
        cout<<"Wrong! Too low!"<<endl;
    else
        cout<<"Right!"<<endl;
}
```

案例——代码实现

```
void GuessNumber(int magic)
{
    int guess;
    cout<<"Please guess a magic number:";
    cin>>guess;
    if(guess>magic)
        cout<<"Wrong! Too high!"<<endl;
    else if(guess<magic)
        cout<<"Wrong! Too low!"<<endl;
    else
        cout<<"Right!"<<endl;
}
```

案例进阶

4

Please guess a magic number:4
Wrong! Too low!

14



案例进阶

只猜一下就
没机会了，
不好玩



请多给点机会吧！



案例——代码实现

```
void GuessNumber(int magic)
{
    int guess;
    do{
        cout<<"Please guess a magic number:";
        cin>>guess;
        if(guess>magic)
            cout<<"Wrong! Too high!"<<endl;
        else if(guess<magic)
            cout<<"Wrong! Too low!"<<endl;
        else
            cout<<"Right!"<<endl;
    }while(guess!=magic);
}
```

案例进阶

10

Wrong! Too low!

71



案例进阶

60



Wrong! Too low!

71



案例进阶

80



Wrong! Too high!

71



案例进阶

75



Wrong! Too high!

71



案例进阶

72



Wrong! Too high!

71



案例进阶

71



Right!

71



案例进阶

欧耶！

我等的很着急



每个人只给有限次机会



案例——代码实现

```
void GuessNumber(int magic)
{
    int guess;
    int count=0;
    do{
        count++;
        cout<<"Please guess a magic number:";
        cin>>guess;
        if(guess>magic)
            cout<<"Wrong! Too high!"<<endl;
        else if(guess<magic)
            cout<<"Wrong! Too low!"<<endl;
        else
            cout<<"Right!"<<endl;
    }while(guess!=magic && count<=3);
}
```

每人只能
猜3次

案例进阶



案例进阶

```
int BuyTicket()  
{  
    int tickets;  
    cout<<"please tell me the number of  
        tickets you want!"<<endl;  
    cin>>tickets;  
    return tickets;  
}
```

案例进阶

```
int BuyTicket()  
{  
    int tickets;  
    cout<<"please tell me the number of  
        tickets you want!"<<endl;  
    cin>>tickets;  
    return tickets;  
}
```


案例进阶

```
int main()  
{  
    int num;  
    int tick=0;  
    tick=BuyTicket();  
    while(tick>0){  
        num=MakeNumber();  
        GuessNumber(num);  
    }  
    return 0;  
}
```

案例进阶

```
int main()  
{  
    int num;  
    int tick=0;  
    tick=BuyTicket();  
    while(tick>0){  
        num=MakeNumber();  
        GuessNumber(num);  
    }  
    return 0;  
}
```

小结

函数的定义

函数的调用

函数的声明

华中农业大学
信息学院

小结

函数的定义

函数的调用

函数的声明

延申

请将连连看游戏按照功能拆分成多个模块，
每个模块用函数实现，请完成整个游戏的代码编写。