CS1605 程序设计实践

Project 1 - Battle of Pets

2023 年 6 月 20 日 至 2023 年 7 月 3 日

重要说明：

1、尽量提前完成，下个 project 的发布将早于当前 project 的截止日期；

2、严格遵循运行示例显示程序输出（包括文字、标点、空格等所有内容）；

3、严格遵守提交作业的方式与要求。

具体要求：

本次 project 的内容是编写一个关于宠物对战的小游戏。以下先介绍战斗机制，然后介绍本次 project 的具体任务。

战斗分为“开始前”和“过程中”两个阶段，具体机制如下。

战斗开始前：

玩家和敌方的所有宠物的 HP 都处于最大值。

双方各自选择一只首先上场的宠物（互不知道对方的选择）。

在完成选择后，程序应显示双方上场的宠物及其 HP。

战斗过程中：

- 关于行动的说明：

战斗按回合进行，在每回合开始时，程序应显示当前的回合数。

每回合依次进行“选择行动”（包括选择技能、选择换上的宠物、选择用药对象等）、**“实施行动”**（包括发动技能、主动交换宠物、使用药水等）、**“被动交换宠物”**（如有）、**“回合结束”四个阶段。**

以下具体说明各个阶段。

【阶段一：选择行动】

玩家与敌方分别选择该回合的行动，即从以下三种类型的行动中选择一种：

1、让场上的宠物使用一个技能（skill），技能说明（包括其优先级）见表 2。

2、主动交换，即用场下的一只宠物换下场上的宠物（change），优先级为 6。

3、使用药水，优先级为 5。玩家没有药水，只有敌方可以使用药水。

程序提示玩家选择行动类型的信息如下：

Select your action (1 for skill, 2 for change):

当玩家选择技能时，提示信息举例如下：

Select the skill (1 for Tackle, 2 for Leaf):

当玩家选择交换宠物时，提示信息举例如下（L 和 G 代表宠物的名字）：

Select your next pet (2 for L, 3 for G):

【阶段二：实施行动】

玩家与地方根据各自在阶段一中选择的行动来分别实施该行动。

如果双方行动的优先级不同，那么优先级较高的先发动。

如果双方都选择换下宠物，那么会同时发动，即互不知道对方会换宠物。**敌方只会根据本回合开始时玩家场上的宠物来进行决策。**

如果双方都选择使用技能且技能的优先级相同，那么宠物的**速度**数值更高的一方先发动技能。如果技能的优先级和宠物的速度都相同，那么敌方先发动技能。

刚交换上场的宠物不能在本回合中使用技能，也不能在本回合中对它使用药水。

在该阶段，程序应依次显示某方主动换上宠物的信息（如有）、某方宠物使用技能的信息及其效果（如有）、某方宠物被打倒的信息（如有）。

【阶段三：被动交换宠物】

场上的宠物在 HP 减少到零时（负数视为零）无法继续战斗，其所属方须选择一只场下的 HP 不为零的宠物，并将其派上场，即被动交换。

在该阶段，程序应显示某方被动换上宠物的信息（如有）。

【阶段四：回合结束】

在该阶段，如果本回合尚未分出胜负，程序应显示当前场上双方宠物的剩余 HP；如果已分出胜负，则程序应显示胜负结果。

- 关于技能的说明：

攻击类型的技能以【技能发动时】对方场上的宠物为目标。

攻击类型的技能会对目标造成伤害（即减少目标的 HP），计算公式为：

伤害量 = 技能威力 x 攻方攻击力÷守方防御力 x 属性倍率

例如，当技能威力为 20，攻方攻击力为 10，守方防御力为 11，属性倍率为 0.5 时，造成的伤害量为 9。**伤害量按公式最终结果的四舍五入取整。**

公式中的技能威力的取值见表 2，攻击力和防御力的取值见表 1，属性倍率的取值见表 3。

- 关于宣布胜负的说明：

当一方的所有宠物都无法战斗而对方还有宠物可以战斗时，该方失败，对方获胜。如果在第 100 回合的结束阶段仍未分出胜负，那么双方平手。

在分出胜负后，程序应显示胜负结果（You Win / You Lose / Draw）。

表 1 宠物信息表

	属性	最大 HP	攻击力	防御力	速度	技能 1	技能 2
小蛙（W）	草	110	10	10	10	撞击	叶片
小龙（L）	火	100	11	10	11	撞击	火焰
小龟（G）	水	100	10	11	9	撞击	水流

表 2 技能信息表

	类型	属性	威力	命中率	优先级
撞击（Tackle）	攻击	普通	20	100%	0
叶片（Leaf）	攻击	草	20	100%	0
火焰（Flame）	攻击	火	20	100%	0
水流（Stream）	攻击	水	20	100%	0

表 3 属性倍率表（左侧为攻方技能属性，上方为守方宠物属性）

	普通	草	火	水
普通	1	1	1	1
草	1	0.5	0.5	2
火	1	2	0.5	0.5
水	1	0.5	2	0.5

名词解释：“克制”。当宠物 A 的属性作为表 3 中的攻方技能属性时对宠物 B 的属性倍率大于 1，则称宠物 A 克制宠物 B。

- 任务 1：简单敌方（得分 60%）

根据上述战斗机制编写一个程序，符合以下要求：

1、玩家和敌方都有小蛙、小龙、小龟各一只。

2、允许用户作为玩家，通过键盘输入来选择首先上场的宠物和每回合的行动。程序应显示可选操作的提示信息。

3、敌方选择的首先上场的宠物总是克制玩家选择的首先上场的宠物。

4、在每个回合的阶段一，如果敌方场上的宠物克制玩家场上的宠物，那么敌方选择让该宠物使用技能 2，否则选择使用技能 1。

5、敌方不会主动交换宠物，只会让场上的宠物使用技能，只有当场上的宠物被打倒了才会换上下一个（即被动交换宠物）。在被动交换宠物时，敌方优先选择能克制玩家场上宠物的宠物，其次优先选择和玩家场上宠物相同的宠物。

- 任务 2：贪心敌方（得分 20%）

基于任务一再写一个程序，与任务一的区别如下：

1、在每个回合的阶段一，如果敌方场下有宠物能克制玩家场上的宠物，那么敌方会选择换上该宠物。

2、在每个回合的阶段一，如果敌方场上的宠物被玩家场上的宠物克制，那么敌方会选择换下该宠物（除非敌方的其它宠物都已无法战斗）。

- 任务 3：吃药贪心敌方（得分 20%）

基于任务二再写一个程序，与任务二的区别如下：

1、敌方有 1 瓶复苏药（Revival Potion），可以让场下一只无法战斗的宠物回复最大 HP 的一半。敌方会在有宠物无法战斗时使用复苏药，但注意不是在宠物被打倒的回合，而是要在下个回合才能用药。

2、敌方有 2 瓶强攻药（Attack Potion），可以让场上宠物的攻击力翻倍，药效持续到该宠物下场（包括主动换下场或因被打倒而下场）为止，药效不能叠

加。敌方会在场上的宠物不被玩家场上的宠物克制时使用强攻药，但不会对已具有强攻药效的宠物重复用药。敌方不会主动换下已具有强攻药效的宠物。注意每瓶药只能用一次，用药需占用一个行动回合（用药的行动优先级为 5）。如果本回合的情况既符合用药的条件也符合主动交换宠物的条件，那么敌方会选择用药而不是交换宠物。在两种药都符合使用条件时，敌方会选择复苏药。程序应显示用药信息。

运行示例：

要求参照以下运行示例来显示程序的运行结果。

其中，蓝色字符表示用户的输入，在实际运行中应显示为默认颜色（如白色）。

- 同一个回合中，优先级高的行动的信息先显示。如果双方都主动交换宠物，先显示玩家交换宠物的信息，再显示敌方交换宠物的信息。
- 用户的输入一定是整数，不用考虑其它类型的不合法输入。如果用户输入的整数不在选项范围内，就让用户重复输入，直到用户输入合法为止。
- 如果宠物受到的伤害大于等于其剩余 HP，那么不显示倒下的宠物的 HP，而是换上新宠物后再显示新宠物的剩余 HP。
- 当玩家只剩两只宠物时，如果选择主动交换宠物，那么 Select your next pet 只显示一个选项。
- 当玩家只剩一只宠物时，Select your action 只有一个选项（1 for skill）。
- 如果第 100 轮还未分出胜负，且又有宠物被打倒，那么要先完成被动交换宠物，然后再输出 Draw。

```
Welcome to Battle of Pets!
You have W, L and G. So does Enemy.
Select your starting pet (1 for W, 2 for L, 3 for G): 4
Select your starting pet (1 for W, 2 for L, 3 for G): 2
You start with L
Enemy starts with G
Your L: HP 100 || Enemy's G: HP 100
Battle starts!
--------------------------------------------
```

```
Round 1
Select your action (1 for skill, 2 for change): 1
Select the skill (1 for Tackle, 2 for Flame): 2
Enemy uses Attack Potion on G
Your L uses Flame! Damage: 10
Your L: HP 100 || Enemy's G: HP 90
------------------------------------------------
Round 2
Select your action (1 for skill, 2 for change): 2
Select your next pet (1 for W, 3 for G): 1
You send W
Enemy's G uses Stream! Damage: 20
Your W: HP 90 || Enemy's G: HP 90
------------------------------------------------
Round 3
Select your action (1 for skill, 2 for change): 1
Select the skill (1 for Tackle, 2 for Leaf): 2
Your W uses Leaf! Damage: 36
Enemy's G uses Tackle! Damage: 40
Your W: HP 50 || Enemy's G: HP 54
------------------------------------------------
(此处省略若干行)
Your W uses Leaf! Damage: 36
Enemy's G is beaten
Enemy sends L
Your W: HP 10 || Enemy's L: HP 100
------------------------------------------------
Round 6
Select your action (1 for skill, 2 for change): 2
Select your next pet (2 for L, 3 for G): 3
You send G
Enemy uses Revival Potion on G
Your G: HP 100 || Enemy's L: HP 100
------------------------------------------------
Round 7
Select your action (1 for skill, 2 for change): 1
Select the skill (1 for Tackle, 2 for Stream): 1
```

```
Enemy sends W
Your G uses Tackle! Damage: 20
Your G: HP 100 || Enemy's W: HP 90
------------------------------------------------
Round 8
Select your action (1 for skill, 2 for change): 1
Select the skill (1 for Tackle, 2 for Stream): 1
Enemy uses Attack Potion on W
Your G uses Tackle! Damage: 20
Your G: HP 100 || Enemy's W: HP 70
------------------------------------------------
Round 9
Select your action (1 for skill, 2 for change): 1
Select the skill (1 for Tackle, 2 for Stream): 1
Enemy's W uses Leaf! Damage: 73
Your G uses Tackle! Damage: 20
Your G: HP 27 || Enemy's W: HP 50
------------------------------------------------
Round 10
Select your action (1 for skill, 2 for change): 1
Select the skill (1 for Tackle, 2 for Stream): 1
Enemy's W uses Leaf! Damage: 73
Your G is beaten
Select your next pet (1 for W, 2 for L): 2
You send L
Your L: HP 100 || Enemy's W: HP 50
------------------------------------------------
Round 11
Select your action (1 for skill, 2 for change): 1
Select the skill (1 for Tackle, 2 for Flame): 2
Your L uses Flame! Damage: 44
Enemy's W uses Tackle! Damage: 40
Your L: HP 60 || Enemy's W: HP 6
------------------------------------------------
Round 12
Select your action (1 for skill, 2 for change): 1
Select the skill (1 for Tackle, 2 for Flame): 2
```

Your L uses Flame! Damage: 44
Enemy's W is beaten
Enemy sends G
Your L: HP 60 || Enemy's G: HP 50

Round 13
Select your action (1 for skill, 2 for change): 1
Select the skill (1 for Tackle, 2 for Flame): 1
Your L uses Tackle! Damage: 20
Enemy's G uses Stream! Damage: 40
Your L: HP 20 || Enemy's G: HP 30

Round 14
Select your action (1 for skill, 2 for change): 1
Select the skill (1 for Tackle, 2 for Flame): 1
Your L uses Tackle! Damage: 20
Enemy's G uses Stream! Damage: 40
Your L is beaten
Select your next pet (1 for W): 3
Select your next pet (1 for W): 1
You send W
Your W: HP 10 || Enemy's G: HP 10

Round 15
Select your action (1 for skill): 1
Select the skill (1 for Tackle, 2 for Leaf): 1
Enemy sends L
Your W uses Tackle! Damage: 20
Your W: HP 10 || Enemy's L: HP 80

Round 16
Select your action (1 for skill): 1
Select the skill (1 for Tackle, 2 for Leaf): 1
Enemy's L uses Flame! Damage: 44
Your W is beaten
You Lose

提交作业的方式与要求：

本次作业共有 3 个任务，要求写成 3 个各自独立的程序，每个程序的所有代码文件（仅包括源文件和头文件）都放在同一个文件夹中，并将该文件夹命名为“**P1-姓名-任务编号**”。

例如张三同学编写的任务 1 的文件夹应命名为“P1-张三-任务 1”。

将按要求命名后的 3 个文件夹打包为 zip 文件（不加密），并将压缩包命名为“**P1-姓名-学号.zip**”，然后在 Canvas 网站上提交。

不按要求提交的作业得零分。

注意！请勿上传除了代码文件之外的任何其它文件。

本次作业提交截止至 7 月 3 日晚 23:59，过期未交的作业得零分。

关于作业要求的 Q&A：

Q：可以使用 StanfordCppLib 等第三方库吗？

A：不可以，请熟悉使用标准库，因为当你以后写其它程序（尤其是和别人合作时），不宜默认使用某个第三方库。

Q：是否每个函数都需要写注释？

A：不需要每个都写。特别简单的（比如 getHP 之类的）函数不需要写。比较复杂的（比如使用技能的）函数就需要写注释。

Q：注释一般是只说明函数的功能，还是要说明函数是如何实现的？

A：两种情况都有。如果是头文件里的函数声明，一般只写功能，因为这个是给库的使用者看的；如果是源文件里的函数定义，有的会简述如何实现，以便同一个项目组的其他开发人员理解。在本次作业中对此不作区分。

Q：作业批改的严格程度？

A：我们是人工批改作业，这也就意味着，如果你在输出信息中拼错了极少量单词或多打了空格，其实并不会扣分（但你不能把很多单词都拼错，或连续多打了很多个空格）。

需要注意的知识点（根据去年同学们的反馈总结的）：

- **调用函数时注意“值传递”的“副作用”**

➢ 副作用 1：实参和形参是不同的对象，一不小心可能混淆

```
int main () {
    int firstPet;
    cin >> firstPet; //  输入 2
    vector<Pet> allPets = {W, L, G}; // W, L, G 都是 Pet 类对象
    Pet *p = &allPets[firstPet];
    Play (p, allPets);
    return 0;
}

void Play (Pet *p, vector<Pet> allPets) {
    通过  p 来间接修改 allPets[2]，再显示 allPets[2]的信息却发现未被修改
}
```

➢ 副作用 2：浅拷贝导致的提前和重复释放动态内存空间

```
int main () {
    IntArray arr1; // IntArray 有一个成员指针，析构函数会释放其指向的动态内存空间
    Compute (arr1);
    此处无法正确访问 arr1 的成员指针指向的动态内存
    return 0;
}

void Compute (IntArray arr2) {
    该函数返回的时候 arr2 会析构，于是把成员指针指向的动态内存空间释放掉了
}
```

- **两个类互以对方为成员函数的参数类型**

```
class A {
    void functionA (B b);
};

class B {
    void functionB (A a);
};
```

编译不通过，因为编译到第二行时，编译器不能识别 B 这个标识符。
所以要在最前面加一句 class B; 从而告诉编译器 B 是一个类。

- **用 cin.get()时要注意之前的回车符**

```
int num;
cin >> num;
(中间省略若干行代码)
char choice = cin.get(); //  或  char choice; cin.get(choice);
```

最后一行会直接读取此前执行 cin>>num;时用户输入的回车符，即 choice 的值为'\n'。