

中山大学移动信息工程学院本科生实验报告

(2016 学年春季学期)

课程名称：Data structures and algorithms

任课教师：张子臻

年级	15	专业 (方向)	
学号	15352144	姓名	黄志浩
电话	13727024650	Email	670994114@qq.com
开始日期	20160420	完成日期	20160507

1.实验题目

【井字棋】

要求实现人、机间的相互博弈， 在操作界面中应可自由选择两人互玩、人机对战或者是两个 NPC 对战，并任意指定先后手。

程序设计时要求实现 game, player, playerHuman, playerComputer 以及 board/digits 这些类，但不必局限于此。

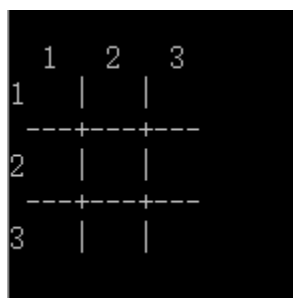
2.实验目的

熟悉类的封装，继承和多态，并提前了解算法的相关信息。

3.程序设计

界面是一个 3*3 方格，玩家通过输入 x (1~3) 行 y (1~3) 列，可以确定下子位置。如果玩家输入的位置超出范围或已经有棋子了，则要求玩家重新输入。

界面如下：



有三种游戏模式可以选择：1、双人模式 2、人机对战 3、电脑对战

```
Please input your choice. 1.双人对战 2.人机对战 3.电脑对战
1
```

模式一：双人模式

可以两个人对战。

模式二：人机对战

可以选择电脑先手或人先手，电脑先手的话，电脑会在九个方格内随机选择一个下子。电脑的 AI 采用极大极小搜索算法，可以保证电脑不败。

Min-max Search 算法常用于棋类等由两方较量的游戏和程序。该算法是一个零总和算法，即一方要在可选的选项中选择将其优势最大化的选择，另一方则选择令对手优势最小化的一个。很多棋类游戏可以采取此算法，例如 tic-tac-toe。

这种双人轮流下子的游戏，可以生成一棵博弈树，

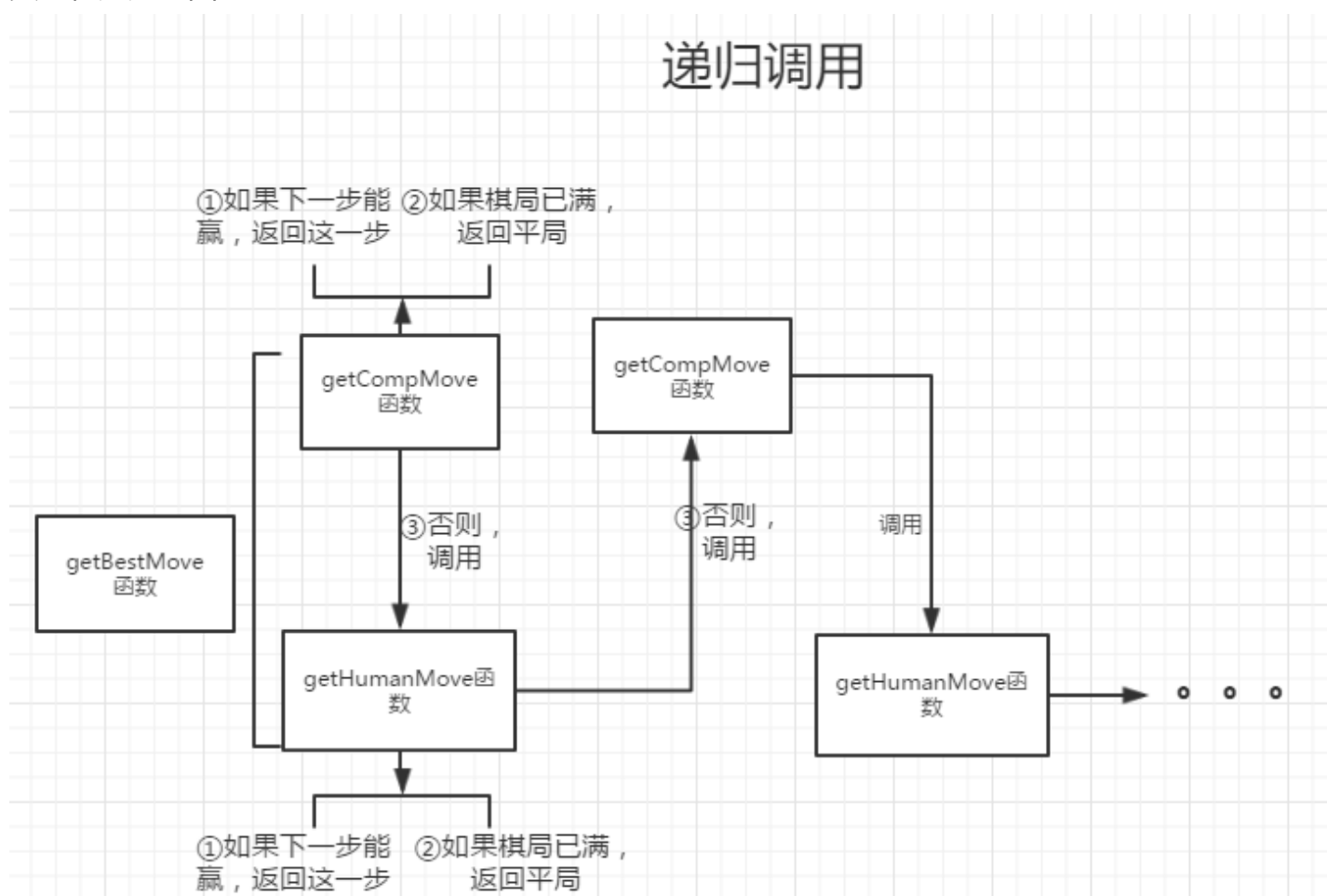
1. 以空棋局为根节点，以某个棋局 1 步后的棋局的该棋局节点的子节点，广度优先生成一棵博弈树。然后从树根开始轮流给每层结点赋予 MAX 和 MIN 的称号。

2. 使用评分函数 $f(p)$ 计算各个叶子节点分值。

3. 当叶子节点的分值计算出来后，再逆层向推算出各层非叶子节点的得分。推算的方法是：对于处于 MAX 层的节点，选其子节点中一个最大的得分作为该节点的得分，这是为了使自己在可供选择的方案中选一个对自己最有利的方案；对处于 MIN 层的节点，选其子节点中一个最小的得分作为该节点的得分，这是为了立足于最坏的情况，这样计算出的父节点的得分为倒推值。

4. 如此反推至根节点下的第一层孩子，如果其中某个孩子能获得在其兄弟节点中最大的倒推值，则它就是当前棋局最佳的走法。

实践程序原理图



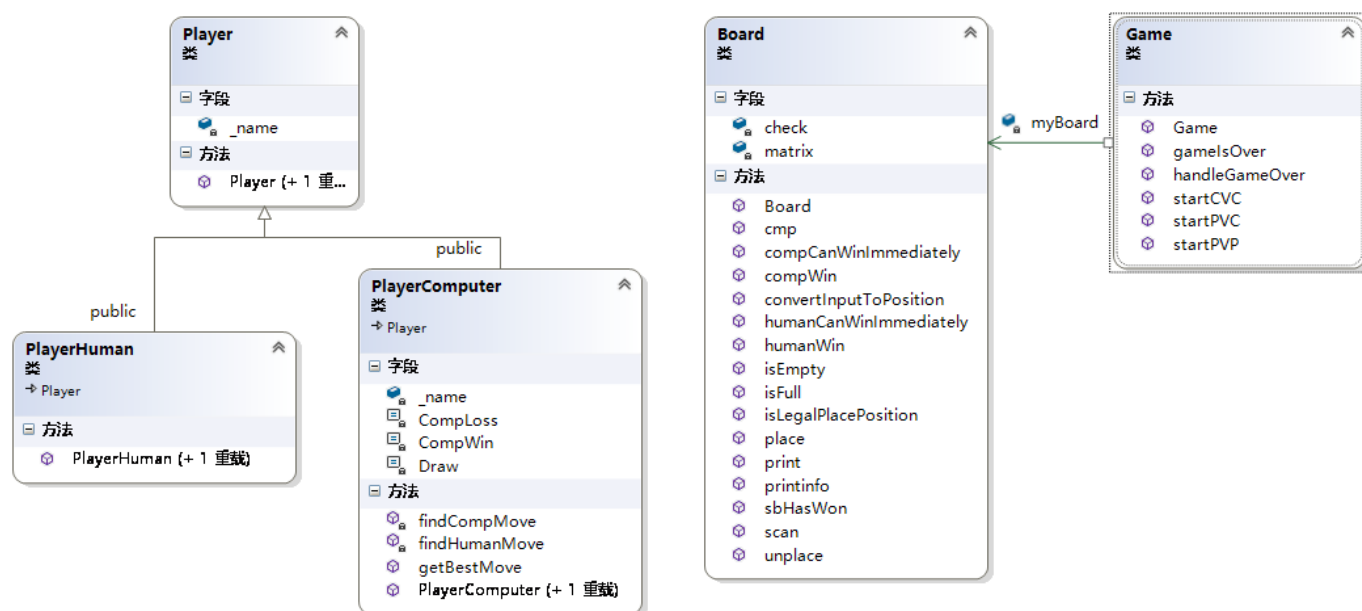
井字棋可以算是极大极小搜索算法的一种简单的特例，可以一直搜索到底部叶子，不用定义深度，当然，如果采用 alpha-beta 剪枝算法，能提高程序运行的效率（剪枝算法还没弄懂，也就不采用了。）

模式三：电脑对战

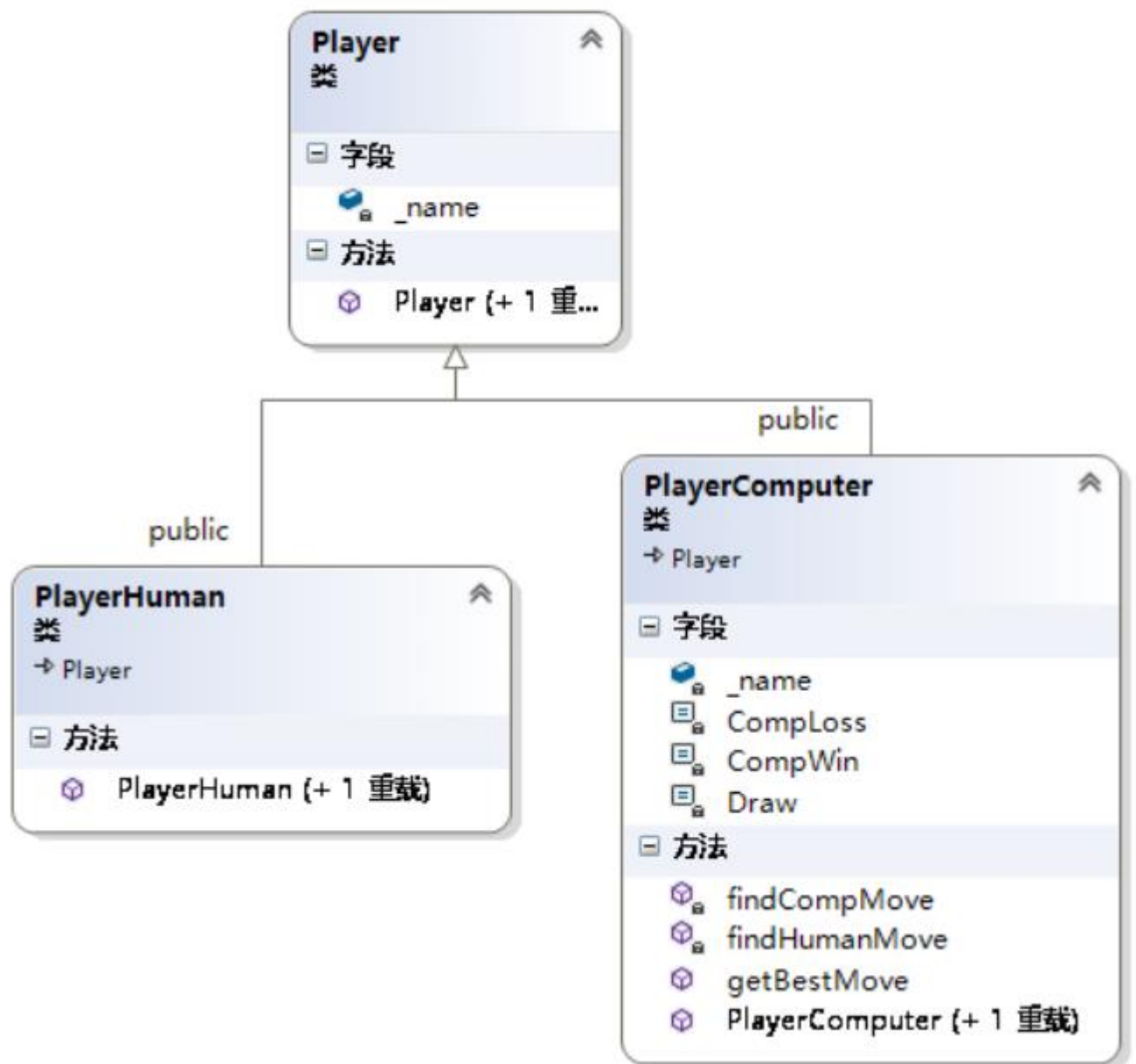
第一个电脑会在九个方格内随机选择一个下子，最终一定为平局。

总计三种情形：第一步下在角；第一步下在中间；第一步下在边上（非角）。

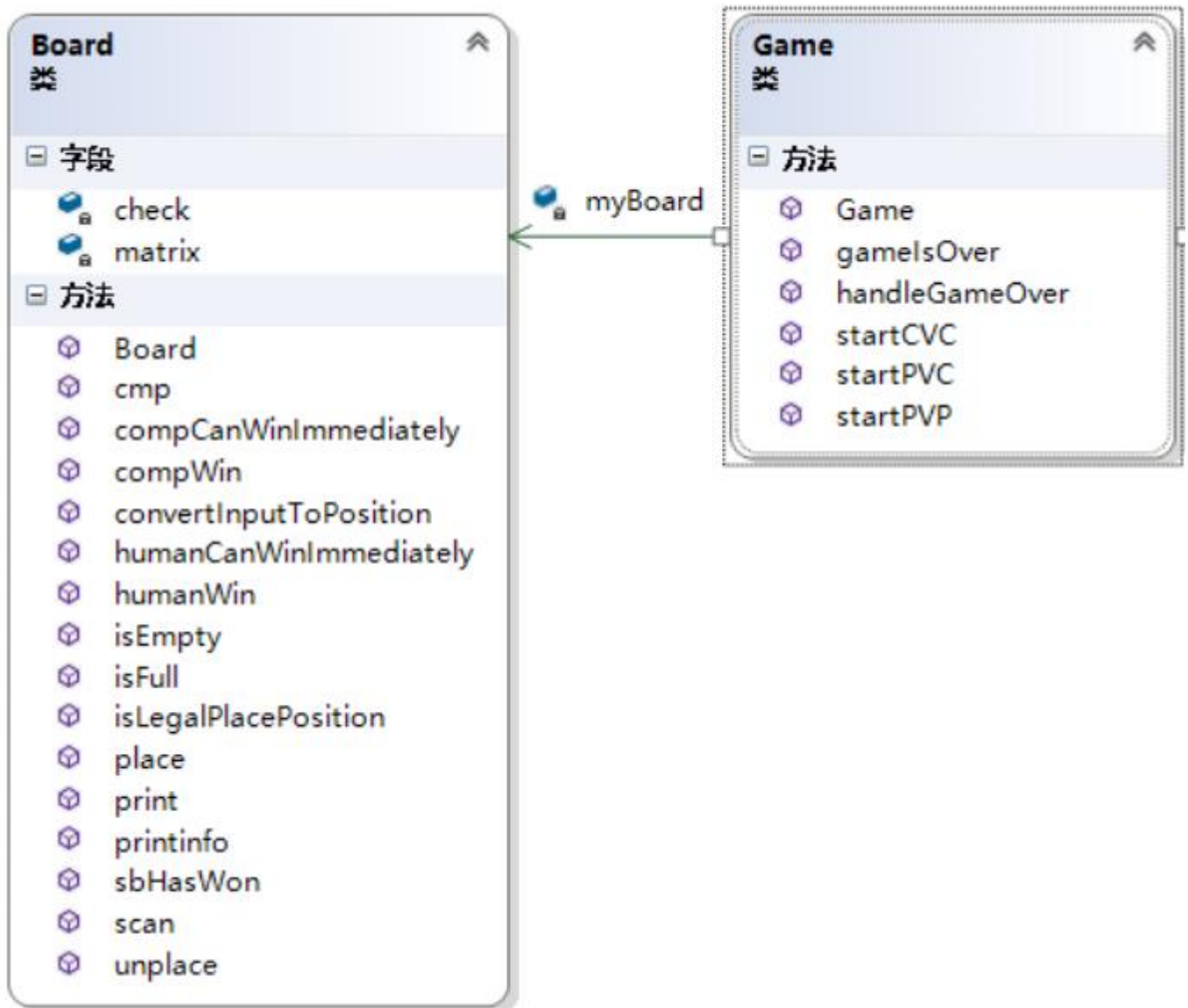
以下是 UML 图。



左半部分大图如下



右半部分大图如下



4.程序运行与测试

```
  1   2   3
1  |   |
--+---+--
2  |   |
--+---+--
3  |   |
```

```
Please input your choice. 1. 双人对战 2. 人机对战 3. 电脑对战
1
```

```
Who will place the first piece?
0->Human first
1->Computer first
```

界面概览如下

```
The computerA's choice:
```

```
   1   2   3
1  |   |  0
---+---+---
2  |   |
---+---+---
3  |   |
```

```
The computerB's choice:
```

```
   1   2   3
1  |   |  0
---+---+---
2  | X |
---+---+---
3  |   |
```

```
The computerA's choice:
```

```
   1   2   3
1 0 |   |  0
---+---+---
2  | X |
---+---+---
3  |   |
```

5.实验总结与心得

一开始没有思路做 AI，后来查到可以用极大极小搜索算法做，于是在图书馆查了一天，想了一个晚上，终于把极大极小算法大致弄懂了，说实话，挺高兴的。

算法挺神奇的，感觉第一个想出来的人好厉害！

总的来说，算法方面，了解到了：

- ① 深度搜索
- ② 极大极小搜索

③ Alpha-beta 剪枝算法

然后，在做游戏的过程中，对游戏的理解也深了。

网上有一种说法是占角是胜利的优先选择，其实不然，第一步有三种选择：第一步下在角；第一步下在中间；第一步下在边上（非角）。

如果双方都走最优的一步，最终结果一定是平局。而第一步走角（中间），对方如果不走中间（角），就一定输，也就是说，第一步走角或中间，对手理论上失误的概率，比第一步走边对手失误的概率高，所以说，第一步走角或边，是理论上胜利几率较高的选择。

附录、提交文件清单

由于程序是在 VS2015 下编写的，所以提交文件如下：

我的井字棋 project 文件夹



（程序 exe 在 release 文件夹里

，源代码在“我的井字棋”文件夹里。）