

# 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: **Software Design II**      任课教师: 张子臻

年级	15 级	专业 (方向)	软件工程
学号	15352155	姓名	赖贤城
电话	13727024851	Email	754578682@qq.com
开始日期	2016/4/30	完成日期	2016/5/2

## 1. 实验题目

运用面向对象编程技术实现井字棋游戏, 并且该游戏应有 3 种游戏模式—— 1. 人机模式 2. 双人模式 3. 电脑对战; 而且, 还应有让玩家选择先后手的功能。

## 2. 实验目的

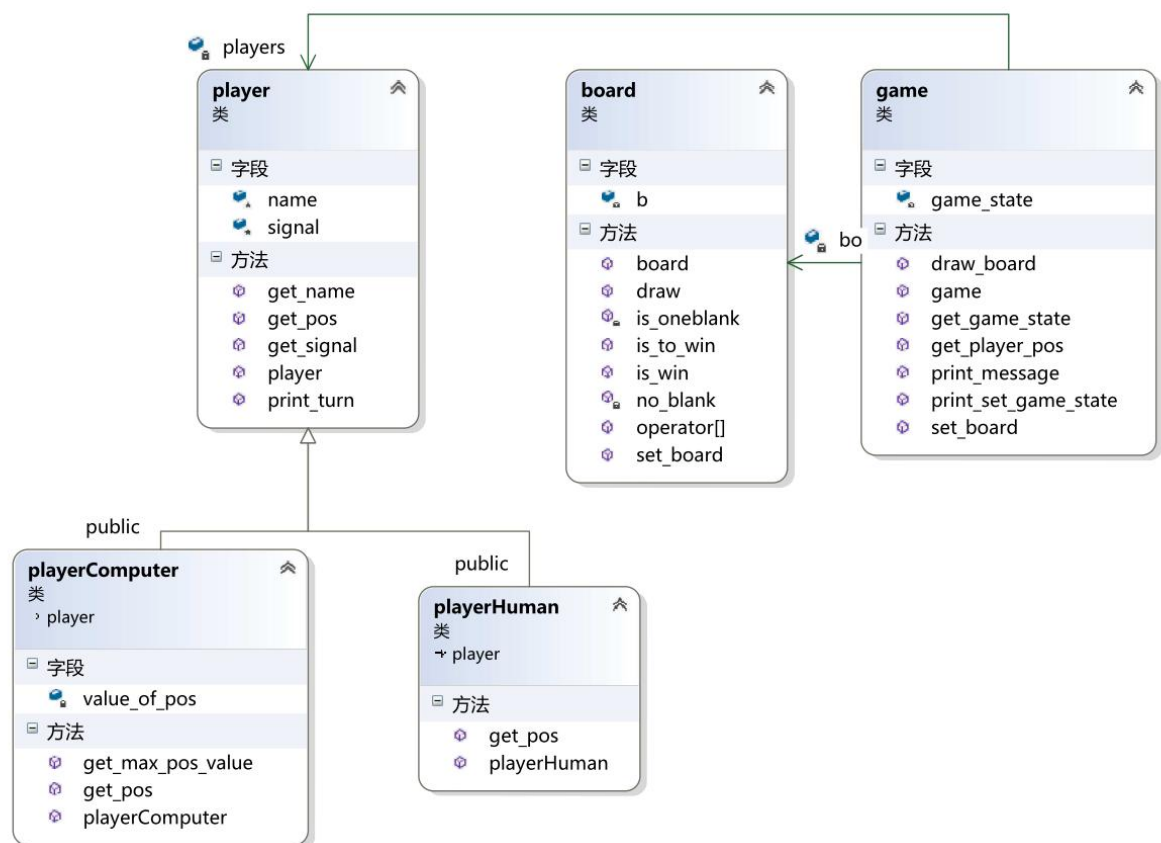
- (1) 提高对面向对象编程技术的综合运用能力
- (2) 理解多态的优点, 并能将其熟练地运用
- (3) 熟练运用继承, 体会继承为开发者带来的便利
- (4) 用基类指针管理派生类对象, 体会继承与多态的结合给对象管理带来的好处
- (5) 学习博弈算法, 本例可用极大极小的深度优先搜索算法

(6) 学会根据需求自己设计类的内容以及类与类之间的相互关系

(7) (对于图形界面版本) 学会使用 **QT** 控件，摸索体会做图形界面与做控制台界面时候思路的异同，总结自己的图形界面设计经验

### 3. 程序设计

#### (1) UML 图



## (2) AI 算法说明

AI 采用极大极小的深度优先搜索算法（或者准确的说是回溯算法），具体解释如下：

`pair<int, int> get_max_pos_value(int step, const board& bo);` //人工智能关键函数

`step` 表示从当前开始，以 AI 为第一步（因为只有 AI 才会调用此函数）紧接着第二步 是人下 ……

搜索时如果检测到当前是电脑下子，而且出现明显必胜局面（如：1 位置为电脑的棋子，3 位置为电脑的棋子，而 2 位置为空）那么就直接返回 `pair (2 (空的那个位置), 1)`；同理，假如当前是人下子且明显必胜 那么就直接返回 `pair (空的那个位置, -1)`；如果棋盘满了还没明显必胜局面，则返回 0（表示平局）。

双方都采取最优策略，也就是说电脑将选择所有可能的下子点里面权重最大的，而人将选择所有可能的下子点里面权重最小的。

这就是所谓极大极小算法。

## 4. 程序运行与测试

### (1) 控制台版本

```
F:\C++\我的井字棋.exe
欢迎来到 井字棋 游戏!
该游戏以 1~9 代表坐标 (例如 第2行第3列 为 6) , 且先手为'X',后手为'O'

游戏开始: 请选择游戏模式—— 1. 人机模式 2. 双人模式 3. 电脑对战:
请输入 "1" 或者 "2" 或者 "3" 或者 "0"退出游戏
1
您是否要先走—— 1. 先走 2. 后走 请输入 "1" 或者 "2"
2
现在该Computer下子了
它下在了 1 的位置

X _ _
_ _ _
_ _ _

现在该Human下子了
请落子 (输入坐标 1~9):
3
X _ 0
_ _ _
_ _ _

现在该Computer下子了
它下在了 4 的位置

X _ 0
X _ _
_ _ _

搜狗拼音输入法 全 :7
```

```
F:\C++\我的井字棋.exe
现在该Computer下子了
它下在了 4 的位置

X _ 0
X _ _
_ _ _

现在该Human下子了
请落子 (输入坐标 1~9):
7
X _ 0
X _ _
0 _ _

现在该Computer下子了
它下在了 5 的位置

X _ 0
X X _
0 _ _

现在该Human下子了
请落子 (输入坐标 1~9):
6
X _ 0
X X _
0 _ _

搜狗拼音输入法 全 :7
```

```
F:\C++\我的井字棋.exe
现在该Computer下子了
它下在了 5 的位置

X _ 0
X X _
0 _ _

现在该Human下子了
请落子（输入坐标 1`9）：
6

X _ 0
X X 0
0 _ _

现在该Computer下子了
它下在了 9 的位置

X _ 0
X X 0
0 _ X

-----Computer赢了!!-----

游戏开始： 请选择游戏模式—— 1. 人机模式 2. 双人模式 3. 电脑对战；
请输入 “1” 或者 “2” 或者 “3” 或者 “0”退出游戏

搜狗拼音输入法 全 :7
```

```
F:\C++\我的井字棋.exe
欢迎来到 井字棋 游戏！
该游戏以 1`9 代表坐标（例如 第2行第3列 为 6）， 且先手为'X',后手为'O'

游戏开始： 请选择游戏模式—— 1. 人机模式 2. 双人模式 3. 电脑对战；
请输入 “1” 或者 “2” 或者 “3” 或者 “0”退出游戏
2
是否 玩家A 要先走—— 1. 先走 2. 后走 请输入 “1” 或者 “2”
2
现在该HumanB下子了
请落子（输入坐标 1`9）：
1

X _ _
_ _ _
_ _ _

现在该HumanA下子了
请落子（输入坐标 1`9）：
2

X 0 _
_ _ _
_ _ _

现在该HumanB下子了
请落子（输入坐标 1`9）：
3

X 0 X
_ _ _
_ _ _

搜狗拼音输入法 全 :
```

```
F:\C++\我的井字棋.exe
现在该HumanB下子了
请落子（输入坐标 1~9）：
3

X O X
_ _ _
_ _ _

现在该HumanA下子了
请落子（输入坐标 1~9）：
7

X O X
_ _ _
O _ _

现在该HumanB下子了
请落子（输入坐标 1~9）：
6

X O X
_ _ X
O _ _

现在该HumanA下子了
请落子（输入坐标 1~9）：
5
搜狗拼音输入法 全：
16:19 2016/5/8
```

```
F:\C++\我的井字棋.exe
现在该HumanA下子了
请落子（输入坐标 1~9）：
5

X O X
_ O X
O _ _

现在该HumanB下子了
请落子（输入坐标 1~9）：
4

X O X
X O X
O _ _

现在该HumanA下子了
请落子（输入坐标 1~9）：
8

X O X
X O X
O O _

-----HumanA赢了！！-----
游戏开始： 请选择游戏模式—— 1. 人机模式 2. 双人模式 3. 电脑对战：
搜狗拼音输入法 全：
16:19 2016/5/8
```

## (2) 图形界面版本

见提交文件里的“[图形界面视频.avi](#)”

### 5. 实验总结与心得

1) 面向对象编程技术的综合运用能力得到了提高

(2) 能熟练地运用多态，在该游戏中为 `playerHuman` 和 `playerComputer` 类的 `get_pos` 函数不同

(3) 能熟练运用继承，`playerHuman` 和 `playerComputer` 类继承自 `player` 类，省去了再添加 `name`，`get_name` 等变量或函数

(4) 基类指针管理派生类对象，无需再管该指针指向的对象类型，只需按部就班调用所需函数即可，省去了繁琐的判断

(5) 学习了博弈算法，本例为极大极小的深度优先搜索算法，加深了对回溯（深搜）的理解

(6) 学会了根据需求自己设计类的内容以及类与类之间的相互关系，如本例我用 `game` 类统一管理 `board` 类以及 `player` 类及其派生类

附录、提交文件清单

[我的井字棋.cpp](#)

[我的井字棋.exe](#)

[图形界面视频.avi](#)

