

# 中国矿业大学计算机学院

## 2019级本科生课程报告

课程名称 游戏设计与开发

报告时间 2022-6-30

学生姓名 王杰永

学 号 03190886

专 业 计算机科学与技术

任课教师 毛磊

评 分 表

毕业要求	指标点	课程教学目标	占比	得分
1.工程知识	1.2 具备扎实的计算机工程基础知识，了解通过计算机解决复杂工程问题的基本方法，并遵循复杂系统开发的工程化基本要求。	<b>目标 1：</b> 掌握一门合适的游戏开发的计算机高级语言，并能使用软件集成开发工具，设计、开发、调试及运行基于特定应用背景的软件系统。	10%	
2.问题分析	2.2 能够针对计算机领域实际工程案例进行需求分析和描述。	<b>目标 2：</b> 针对游戏设计的要求，进行功能需求分析，确定设计目标，并能撰写相关游戏策划书。	40%	
4.研究	4.2 针对计算机领域复杂工程问题，具有根据解决方案进行工程设计与实施的能力，具有系统的工程研究与实践经历。	<b>目标 3：</b> 在进行需求分析的基础上，设计游戏场景，编写代码，调试并正确运行满足需求的电脑游戏。	50%	
总分				

# 目 录

1 游戏概述.....	1
1.1 选题背景.....	1
1.2 五子棋简介.....	1
2 需求分析.....	2
2.1 功能需求.....	2
2.2 性能需求.....	2
2.3 环境需求.....	2
3 概要设计.....	2
3.1 总体功能流程.....	2
3.2 软件结构设计.....	3
3.3 系统接口设计.....	4
4 详细设计.....	5
4.1 界面设计.....	5
4.2 AI 落子逻辑.....	6
4.3 AI 休眠时间.....	7
4.4 系统主要逻辑.....	8
5 课程总结.....	9
6 参考文献.....	10

## 1 游戏概述

本款游戏名为 ThetaGobang，是一款单机的人机对战五子棋小游戏。操作简单、易于上手、界面友好是本项目的特点。受 AlphaGo 命名的启发，最终本产品被命名为 ThetaGobang。

### 1.1 选题背景

棋类游戏是人类日常生活中休闲益智的主要方式。随着时代的发展，在 PC 机、移动终端设备上的“虚拟”棋类游戏得到广泛的传播。这种电子棋类的游戏方式，开局迅速且不占用资源，受到广大玩家的喜爱。

在人工智能领域内，博弈是很重要的一个研究分支，很多实际问题都可以在博弈的研究中得到解决。1997 年，IBM 超级计算机 Deeper Blue 击败国际象棋世界冠军，成为了人工智能挑战人类智能发展的一个重要里程碑。

如何开发一款好的 AI，是广大博弈爱好者深入思索的问题。

### 1.2 五子棋简介

五子棋是起源于中国古代的传统黑白棋种之一。现代五子棋日文称之为“连珠”，英译为“Renju”，英文称之为“Gobang”或“FIR” (Five in a Row 的缩写)，亦有“连五子”、“五子连”、“串珠”、“五目”、“五目碰”、“五格”等多种称谓。在五子棋游戏中，有如下常见术语：

天元：棋盘最中间的点

阳线：棋盘上可见的横纵直线。

阴线：棋盘上无实线的隐形斜线。

连：在棋盘任意一条阳线或阴线上形成的有 5 个或 5 个以上的同色棋子不间隔的紧紧相连。

五连：在棋盘上形成的 5 个同色棋子的连。

活四：在棋盘的某一条阳线或阴线上有同色 4 子不间隔相连，且在此 4 子两端各有一个无子的落子点。

## 2 需求分析

ThetaGobang 是一款基于 Unity2D 的人机对战五子棋游戏，操作简单、易于上手、界面友好是本项目的特点。该项目的目标群体为各类热爱五子棋游戏的玩家。

### 2.1 功能需求

- (1) 游戏模式选择：允许玩家选择电脑是否先手。
- (2) 玩家落子：玩家点击棋盘时，在合理的位置出现玩家棋子。
- (3) AI 算法：AI 根据当前场面态势，计算出合适的落子位置并落子。
- (4) AI 思考时间：为了考虑玩家的游戏感受，AI 的“思考”时间会随游戏时间的累积而增长。

### 2.2 性能需求

AI 的对当前场面态势的计算速度应低于其随游戏时间增长的“思考”速度。

### 2.3 环境需求

本项目采用 Unity3D-2021.3.4f1c1 开发，打包 windows 系统的可执行文件后，其运行环境要求如下。

#### 硬件环境：

- (1) Inter i5 CPU
- (2) 内存不低于 4GB
- (3) 硬盘空间不低于 20GB

#### 软件环境：

- (1) Windows10/11 操作系统

## 3 概要设计

### 3.1 总体功能流程

本项目的基本逻辑较为简单，根据 Unity 环境的开发特点以及项目需要，本项目主体功能的流程图如下：

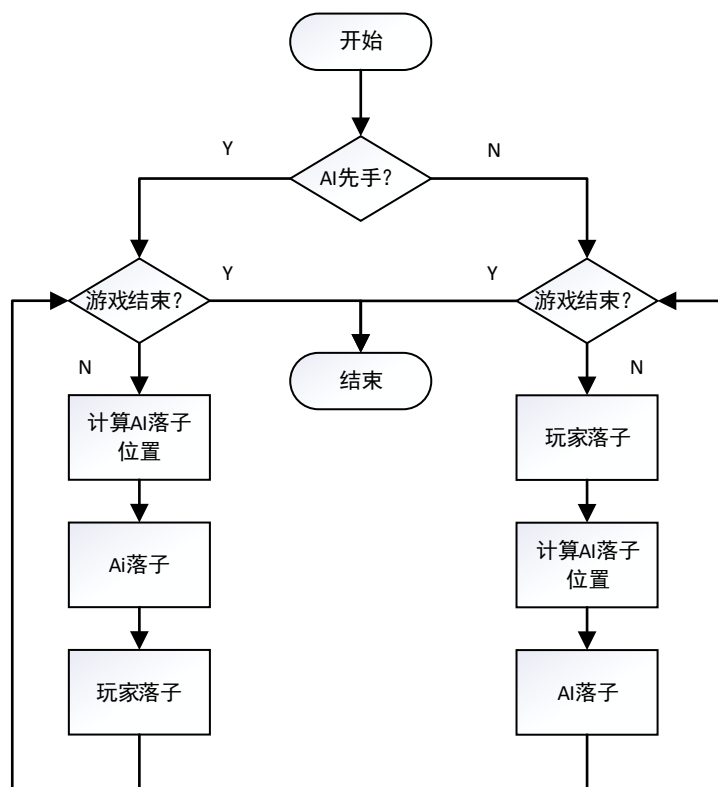


图 1 主体流程图

### 3.2 软件结构设计

根据 ThetaGobang 系统的逻辑结构，得到的软件结构图如图所示：

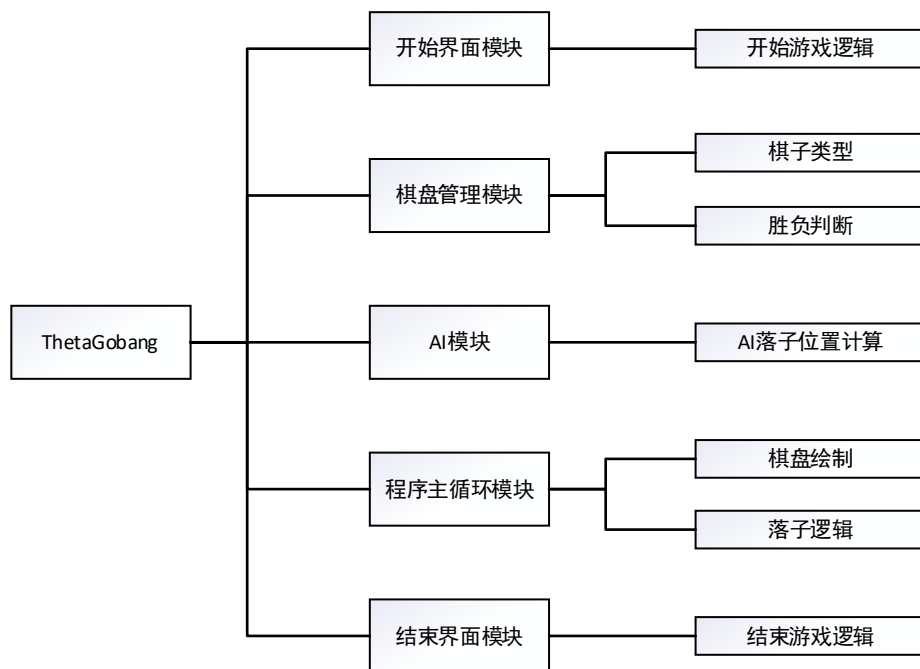


图 2 软件结构图

### 3.2.1 开始界面模块

包括系统背景音乐、开始游戏按钮、系统背景图；玩家点击开始游戏按钮，即可进入游戏。

### 3.2.2 棋盘管理系统

在该模块，设定了棋子的类型，同时需实现胜负判断功能——检测棋盘任意位置周边的最大连接情况。

### 3.2.3 AI 模块

AI 可以根据当前棋盘态势，通过特定算法，计算当前态势下的最优落子位置。由于系统的开发遵循软件工程开发模式，因此对 AI 落子算法的修改并不会引起整个系统的不可运行。体现系统良好的可拓展性。

### 3.2.4 程序主循环模块

该模块负责棋盘的绘制、落子的逻辑等功能；同时，该模块将其余各个模块整合到一起，形成完整的游戏架构。

### 3.2.5 结束界面模块

包括游戏结束时的背景、弹窗等功能。

## 3.3 系统接口设计

### 3.3.1 主要游戏界面

游戏界面较为简单，仅包含一张  $15 \times 15$  的游戏棋盘，玩家可以通过开始游戏界面的开始游戏按钮或游戏结束界面的重新开局按钮跳转至该界面。系统的主要功能如玩家下棋逻辑、AI 落子位置计算、AI 落子逻辑等均在该界面有所体现。该界面总共 225 个落子点作为系统对玩家的操作接口，均可被玩家点击。

### 3.3.2 开始游戏界面

在游戏开始时，该界面向玩家展示。玩家可以自由选择进入或是退出游戏。以下是开始游戏界面向玩家开放的按钮接口的说明：

- (1) 开始游戏按钮：当用户点击开始游戏按钮后，系统跳转至主要游戏界面。
- (2) 退出游戏按钮：当用户点击退出游戏按钮后，系统退出。

### 3.3.3 结束游戏界面

在玩家获胜或是 AI 获胜后，该界面向玩家展示。玩家可以通过单击重新开局按钮，重新进入主要游戏界面。

## 4 详细设计

### 4.1 界面设计

系统的 UI 界面主要采用 Unity 的画布 Canvas 以及预定义的外部图片绘制，由主要游戏界面、开始游戏界面以及结束游戏界面组成。



图 3 开始游戏界面

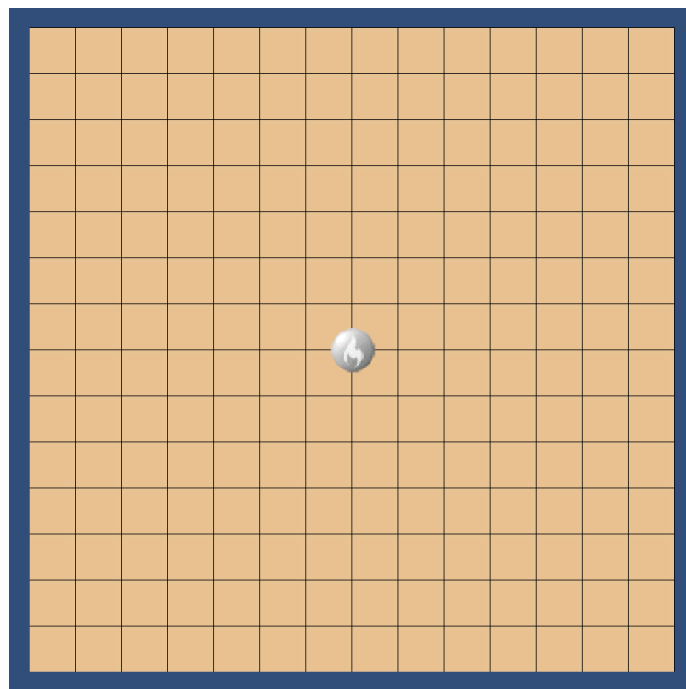


图 4 主要游戏界面





图 5 游戏结束界面

#### 4.2 AI 落子逻辑

在  $15 \times 15$  的标准五子棋棋盘上，共有 225 个落子点，以五连珠作为获胜条件，共有 572 种不同的获胜位置。通过遍历当前棋盘这 572 种获胜位置的落子情况，分别给予落子数量 1、2、3、4 不同的分值，计算当前棋盘的启发式得分，从而找到 AI 的最佳落子位置。

每个落子位置得分的启发式计算公式如下：

$$AIScores(x, y) = \begin{cases} 5 & \text{if 该方案达成的五连数 } 1/5 \\ 50 & \text{if 该方案达成的五连数 } 2/5 \\ 500 & \text{if 该方案达成的五连数 } 3/5 \\ 5000 & \text{if 该方案达成的五连数 } 4/5 \end{cases} \quad (1)$$

$$PlayerScores(x, y) = \begin{cases} 5 & \text{if 该方案达成的五连数 } 1/5 \\ 55 & \text{if 该方案达成的五连数 } 2/5 \\ 495 & \text{if 该方案达成的五连数 } 3/5 \\ 5500 & \text{if 该方案达成的五连数 } 4/5 \end{cases} \quad (2)$$

其中， $AIScores(x, y)$ 表示 AI 在位置 $(x, y)$ 处的得分计算公式， $PlayerScores(x, y)$ 为玩家位置 $(x, y)$ 处的得分计算公式。可以通过设置不同的值来改变 AI 的游戏策略——以三连珠 AI 与玩家的得分取值为例，在式(1)与式(2)的取值方案中，玩家达成三连珠的位置得分低于 AI 得分，因此 AI 会优先将自己的三连珠扩展为四连珠，以此达成进攻

策略：如果将式(2)中三连珠得分由 495 改为 505，即玩家达成三连珠的位置得分高于 AI 得分，此时 AI 会优先对玩家的三连珠进行“围堵”，以此达成防守策略。

AI 落子位置的计算算法如下：

---

算法：AI 落子位置计算

---

- 1) 计算每个落子点的得分.
  - 2) 对于棋盘上的每个落子点：
    - a) 计算黑子的最大得分，其坐标作为黑子的最佳落子点.
    - b) 计算白子的最大得分，其坐标作为白子的最佳落子点.
  - 2) 若白子的最佳落子点得分大于黑子最佳落子点的得分，则 AI 的最佳落子点为白子的最佳落子点；反之，AI 的最佳落子点为黑子的最佳落子点.
  - 4) 返回 AI 的最佳落子点.
- 

其中，步骤 1) 计算每个落子点得分的算法如下：

---

算法：落子点得分计算

---

- 1) 遍历全部的落子点.
  - 2) 对每个落子点涉及到的全部可能获胜方案.
  - 3) 根据黑子的连珠数量以及式 2) 计算黑子得分.
  - 4) 根据白子的连珠数量以及式 3) 计算白子得分.
  - 5) 返回全部落子点的得分.
- 

### 4.3 AI 休眠时间

由于 AI 落子计算的算法计算速度很快，此时玩家落子后 AI 便会在极短的时间内完成决策并落子。AI 极快的思考速度对玩家的游戏体验很不友好，且与真实的人人对战并不相同。因此，ThetaGobang 中加入了 AI 的“思考”时间，从而让玩家的游戏感受更贴近真实感。

对于“思考”时间，可以通过系统休眠随机时间来模拟，即：

$$sleepTime = Random(_sleepDown, _sleepUp) \quad (3)$$

其中， $sleepTime$  为休眠时间， $_sleepDown$  为休眠时间下限， $_sleepUp$  为休眠时间上限， $Random(x, y)$  为区间  $[x, y)$  中的随机数。

式(3)的方式固然简单，但却将 AI 的“思考”时间局限于固定的区间内，尽管“思考”时间是随机的，但固定的随机区间会产生非随机感。因此，ThetaGobang 采用了更

为复杂的随机区间线性随机增长策略：

$$\_sleepUp = \_sleepUp + 50 + Random(0,5) \times 25 \quad (4)$$

$$\_sleepDown = \_sleepDown + 25 + Random(0,5) \times 12 \quad (5)$$

式(4)和式(5)描述了休眠时间上限/下限随游戏的时间而变化的趋势。AI 在完成每次落子的同时，通过式(4)和式(5)提高休眠时间的上限/下限，从而模拟出 AI 的“思考”时间随游戏时间的进行愈来愈高的模式，更贴近真实场景下的人人对局。

同时，为了防止游戏时间过长导致的 AI “思考”时间过长，本系统还设置的思考时间上限/下限的阈值  $\_upThreshold$ 、 $\_downThreshold$ ，一旦上限/下限超过了阈值，则强制设置为阈值，从而消除了时间增长过快的隐患。即

$$\_sleepUp = max(\_sleepUp, \_upThreshold) \quad (6)$$

$$\_sleepDown = max(\_sleepDown, \_downThreshold) \quad (7)$$

其中， $max(x,y)$ 为求 $x$ 与 $y$ 最大值的函数。

#### 4.4 系统主要逻辑

本项目采用 Unity3D-2021.3.4f1c1 开发。Unity 游戏开发引擎是一款上手简单、开发迅速的工具，基于面向对象的思想，很容易便可完成开发。

在 Unity 中，使用 C#语言开发，每个类一旦继承自父类 `MonoBehaviour`，便可以获得 Unity 对象的特性。最直接的表现是两个方法——`Start()`方法以及 `Update()`方法。

`Start()`方法会在脚本挂载并运行后自动执行，类似于一个类的构造方法。通常会在 `Start()`方法中编写某些初始化语句。

`Update()`方法会在系统运行时，每间隔一帧执行一次。这是 Unity 中最为重要的特性之一。

基于 Unity 中 `Update` 方法的特性，本系统的主要逻辑如下。

在 `Start()`方法中，进行 UI 的绘制、棋盘的创建等；设置全局变量以标识当前的执子方；在主循环的 `Update()`方法中，不断判断上述变量，若当前轮到 AI 落子，则调用 AI 落子的接口方法完成落子，并修改上述变量为“轮到玩家落子”；对棋盘的每个交叉点设置“单击”监听方法，若当前轮到玩家落子且交叉点无棋子放置，则完成玩家的本次落子，并修改变量为“轮到 AI 落子”。

从本质上讲，`Update()`方法以及上述提到的监听方法分属于不同的线程，基于多线程技术才可实现上述的落子逻辑。因此，本项目的主体逻辑由两个线程构成。其流程图

如下：

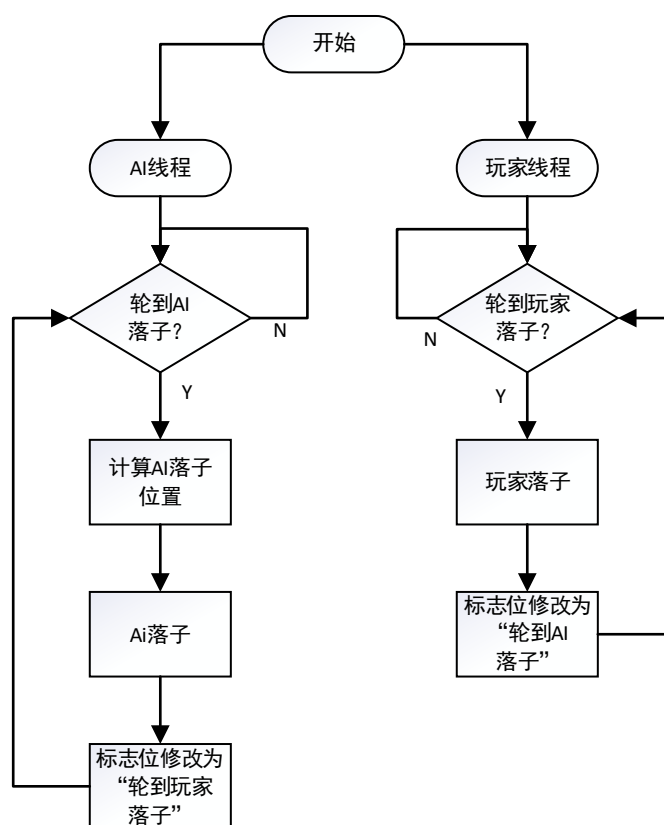


图 6 系统主要逻辑

## 5 课程总结

《游戏设计与开发》是计算机学院在游戏人才培养目标方面的专业选修课程，也是学生接触和进入游戏产业的基础课程。凭借对游戏的热爱，在大三下学期我选择了本门课程。

通过本门课程的学习，我收获了许多。在课程的一开始，课堂上展示的游戏策划考核真题让我感受到了震惊，原来一名游戏策划对数学、对概率的要求并不低，这也打破了我对游戏从业人员的传统认知；在后续的课程中，我对游戏产业的发展，对游戏剧情、数值等的设计，对游戏开发的一些简单基础有了了解，满足了我对游戏相关产业的好奇。

在最后的大作业中，我选择了 ThetaGobang——人机对战五子棋作为结课项目，并顺利完成。之所以选择人机五子棋作为考核，是因为本学期的前半段时间学习了《人工智能 A》这门课程，学习到的 A\*算法、 $\alpha - \beta$ 剪枝等理论知识正是棋类游戏人机博弈的基础，自己希望通过本次课程完成相关算法由理论到实践的转变。

尽管最后只是实现了一个较为简单的启发式算法，但课堂所学内容潜移默化之间影响了我。在开发过程中，站在玩家的角度为玩家思考游戏体验问题，最后设定了合理的数值来让玩家有更好的游戏体验。

此外，通过本门课程的学习，自己又了解了计算机科研或是从业人员的一个方向——计算机图形学，并产生了一定的兴趣。个人比较喜爱偏底层的技术，因此计算机图形方向对我来说或许是个不错的选择。

感谢本门课程。

## 6 参考文献

- [1]姜勇. 五子棋人机对战系统设计[D].电子科技大学,2010.
- [2]李昊. 五子棋人机博弈算法优化研究与实现[D].大连海事大学,2020.DOI:10.26989/d.cnki.gdlhu.2020.000523.
- [3]郑健磊,匡芳君.基于极小极大值搜索和 Alpha Beta 剪枝算法的五子棋智能博弈算法研究与实现[J].温州大学学报(自然科学版),2019,40(03):53-62.
- [4]陈桥. 基于 BP 神经网络的五子棋自学习系统的设计与实现[D].燕山大学,2016.