

Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Программирование

Курсовой проект по программированию
Игра Шахматы

Работу выполнил:

Крылов И.С.

Группа: 13501/4

Преподаватель:

Вылегжанина К.Д.

Санкт-Петербург
2016

Содержание

Глава 1. Шахматы	2
Задание	2
Концепция	2
Правила игры	2
Доска и начальная позиция	2
Ходы	2
Шах, мат и пат	3
Итог игры	4
MVP - минимально работоспособный продукт	4
Диаграмма прецедентов	4
Вывод	5
Глава 2. Проектирование приложения, реализующего игру шахматы	5
chessEngine	6
chessEngine API	6
Тесты	6
Стандарты языка	6
Компилятор	7
Вывод	7
Глава 3. Реализация игры шахматы	7
Среда разработки	7
chessEngine	7
chessCUI	8
0.1 Листинг	8
1 Выводы	8

Шахматы

Шахматы - всемирно известная логическая стратегическая игра, впервые возникшая в Индии полторы тысячи лет назад. Данная игра развивает логическое мышление, официально признана видом спорта и является одной из самых популярных настольных игр.

Задание

Графическое представление шахматного поля с возможностью перемещать фигуры в соответствии с правилами, предусмотренными игрой.

Реализация игры шахматы с графическим интерфейсом и предусмотренным искусственным интеллектом. Возможность изменять начальные параметры игры.

Концепция

Реализация игры шахматы: графическое представление поля, фигур, возможность перемещать шахматные фигуры по полю в соответствии с правилами, предусмотренными игрой. Несколько режимов игры на выбор: игра со вторым пользователем и игра с искусственным интеллектом. Реализация искусственного интеллекта на уровне случайных неанализируемых ходов, не противоречащим правилам игры, и на уровне анализирующего механизма, целью которого является нахождение наиболее выгодной стратегии для победы. Возможность изменять предустановки игры: выбор цвета фигур, опциональная возможность играть с/без счётчика времени. Тем самым пользователю предоставляется виртуальная реализация настольной игры шахматы.

Правила игры

Доска и начальная позиция

Игра происходит на доске, поделенной на равные квадратные клетки, или поля. Размер доски — 8×8 клеток. Вертикальные ряды полей (вертикали) обозначаются латинскими буквами от a до h слева направо, горизонтальные ряды (горизонталы) — цифрами от 1 до 8 снизу вверх; каждое поле обозначается сочетанием соответствующих буквы и цифры. Поля раскрашены в тёмный и светлый цвета (и называются, соответственно, чёрными и белыми) так, что соседние по вертикали и горизонтали поля раскрашены в разные цвета. Доска располагается так, чтобы ближнее угловое поле справа от игрока было белым (для белых это поле h1, для черных — поле a8). У игроков в начале игры имеется по одинаковому набору фигур. Фигуры одного из игроков условно называются «белыми», другого — «чёрными». Белые фигуры окрашены в светлый цвет, чёрные в тёмный. Сами игроки называются «белые» и «чёрные» по цвету своих фигур. В каждый комплект фигур входят: король, ферзь, две ладьи, два слона, два коня и восемь пешек. В начальной позиции фигуры обеих сторон размещаются так, как показано на диаграмме. Белые занимают первую и вторую горизонтали, чёрные — седьмую и восьмую. Пешки расположены на второй и седьмой горизонталях соответственно.

Ходы

Игра заключается в том, что игроки поочерёдно делают ходы. Первый ход делают белые. За исключением взятия на проходе и рокировки, описанных ниже, ход заключается в том, что игрок перемещает одну из своих фигур на другое поле по следующим правилам:

- Фигуры, кроме коня, во время хода считаются передвигающимися по прямой линии в плоскости доски, то есть «проходящими» все поля между начальным и конечным, поэтому все эти поля должны быть свободны. Если на пути фигуры находится другая фигура, то переместить фигуру на поле за ней невозможно. Исключением является ход коня (см. ниже)
- Ход на поле, занятое своей фигурой, невозможен
- При ходе на поле, занятое чужой фигурой, она снимается с доски (взятие)

- Король ходит на расстояние 1 по вертикали, горизонтали или диагонали
- Ферзь ходит на любое расстояние по вертикали, горизонтали или диагонали
- Ладья ходит на любое расстояние по вертикали или горизонтали
- Слон ходит на любое расстояние по диагонали
- Конь ходит на поле, находящееся на расстоянии 2 по вертикали и 1 по горизонтали или 1 по вертикали и 2 по горизонтали от текущего положения. В отличие от всех остальных шахматных фигур, ход коня делается вне плоскости доски, то есть конь непосредственно перемещается («перепрыгивает») с начального поля на конечное и никакие фигуры, стоящие на других полях, ходу коня помешать не могут. В частности, конь может ходить на поле, даже если оно полностью окружено своими или чужими фигурами
- Пешка ходит со взятием по диагонали на одно поле вперёд-вправо или вперёд-влево, а без взятия — по вертикали на одно поле вперёд. Если пешка в данной партии ещё не делала ходов, она может сделать ход без взятия на два поля вперёд. Направлением «вперёд» называется направление к восьмой горизонтали для белых или к первой для чёрных. Когда пешка ходит на последнюю горизонталь (для белых — на восьмую, для чёрных — на первую), ходящий должен заменить её на любую другую фигуру того же цвета, кроме короля (превращение пешки). Превращение пешки является частью того хода, которым она перемещается на последнюю горизонталь. Таким образом, если, например, превращённая из пешки фигура угрожает королю противника, то этот король в результате хода пешкой на последнюю горизонталь немедленно оказывается под шахом

Кроме того, есть два специальных хода:

- Рокировка — если король и одна из ладей того же цвета не двигались с начала игры, то король и эта ладья могут в один ход одновременно сменить положение (рокироваться). При рокировке король сдвигается на 2 клетки по направлению к ладье, а ладья ставится на поле между начальной и конечной позицией короля. Рокировка невозможна, если король или соответствующая ладья уже ходили. Рокировка временно невозможна, если поле, на котором стоит король, или поле, которое он должен пересечь, или поле, которое он должен занять, находится под ударом одной из фигур противника, или если между рокируемыми королём и ладьёй находится какая-либо фигура. Рокировка считается ходом короля, а не ладьи, поэтому рокировку следует начинать с перестановки короля, а не ладьи
- Взятие на проходе — когда пешка совершает ход на две клетки через битое поле, находящееся под ударом пешки противника, то ответным ходом она может быть взята этой пешкой противника. При этом пешка противника перемещается на битое поле, а сбитая пешка снимается с доски. Взятие на проходе возможно только непосредственно в ответ на ход пешки через битое поле, на следующих ходах оно уже не разрешено

Шах, мат и пат

- > Король, находящийся на битом поле, называется стоящим под шахом. Сделать ход, после которого король противника оказывается под шахом, значит дать шах королю (или объявить шах). Ходы, после которых король сделавшего ход остаётся или оказывается под шахом, запрещены; игрок, король которого находится под шахом, обязан немедленно его устранить
- > Если король игрока находится под шахом и игрок не имеет ни одного хода, позволяющего устранить этот шах, этот игрок называется получившим мат, а его противник поставившим мат. Цель игры и состоит в том, чтобы поставить мат королю противника
- > Если игрок при своей очереди хода не имеет возможности сделать ни одного хода по правилам, но король игрока не находится под шахом, такая ситуация называется пат

Итог игры

Игра завершается выигрышем одной из сторон или ничьей.

Выигрыш фиксируется в следующих случаях:

1. Мат. Игрок, поставивший мат, выигрывает
2. Один из игроков сдался. Игрок, решивший, что дальнейшее сопротивление бессмысленно, может сдаться в любой момент. Его противник объявляется победителем

Ничья фиксируется в следующих случаях:

1. Пат
2. Ни у одной из сторон нет минимально необходимого для мата количества фигур (например, на доске остались только короли и одна лёгкая фигура)
3. Троекратное повторение позиции
4. Правило 50 ходов. Обе стороны сделали 50 последних ходов без взятия и без хода пешкой. Как и в случае троекратного повторения позиции. Партия также считается закончившейся вничью, если обе стороны сделали 75 ходов без взятия и без хода пешкой

MVP - минимально работоспособный продукт

Реализовать консольное приложение с возможностью перемещать фигуры согласно правилам игры и закончить игру в случае взятия короля.

Диаграмма прецедентов

На рис. 1 изображена UML Use Case диаграмма организации игрового процесса приложения

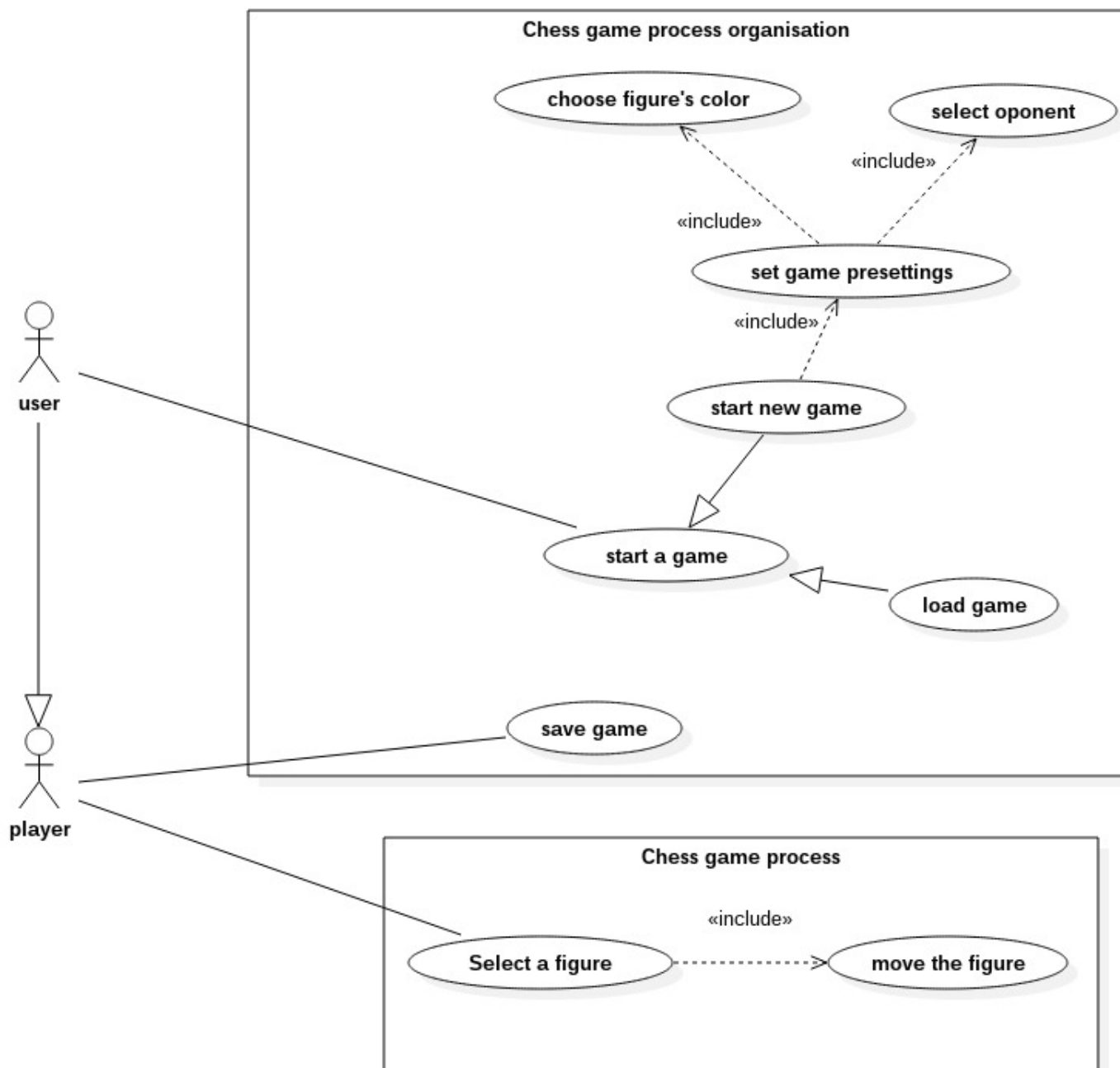


Рис. 1: UML-диаграмма вариантов использования приложения

Вывод

Был получен навык работы с UML диаграммами, что позволило лучшее структуризировать проект.

Проектирование приложения, реализующего игру шахматы

В проекте было выделено 5 подпроектов:

- Ядро - chessEngine
- chessCUI - консольное приложение
- chessGUI - графическое приложение
- chessAI - искусственный интеллект

- Тесты

На рис. 2 изображена UML - диаграмма компонентов, отображающая общую структуру проекта, разделение на подпроекты

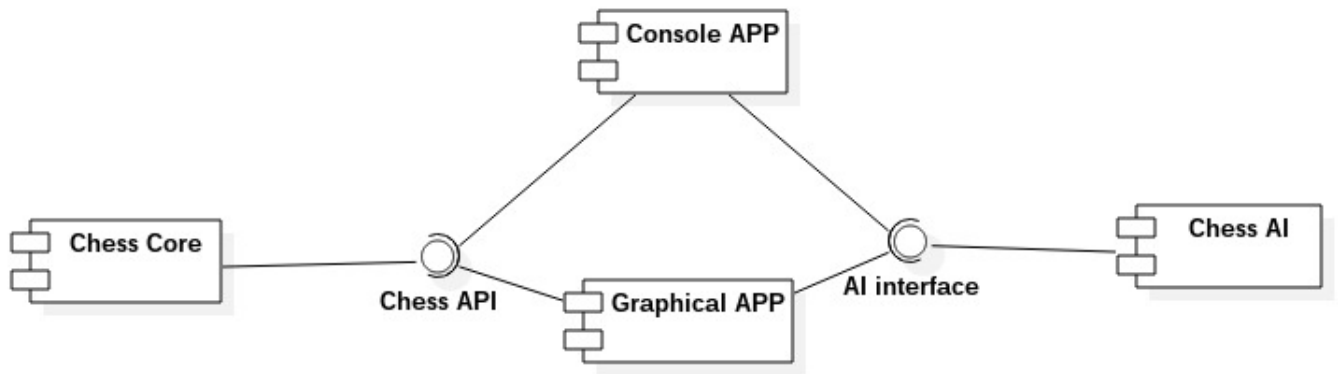


Рис. 2: UML-диаграмма компонентов

chessEngine

Ядро приложения, реализующее игровую логику.

chessEngine API

Интерфейс ядра предоставляет следующие возможности использования:

- **void chooseMyFigure(int x, int y);**
Позволяет выбрать по передаваемым координатам фигуру, которой впоследствии будет выполнен ход
- **void moveMyFigure(int x, int y);**
Позволяет переместить выбранную фигуру на указанную позицию
- **myFigure* getMyFigure(int x, int y);**
Возвращает указатель на содержимое игровой доски на указанной позиции
- **bool isFigureSelected();**
Возвращает TRUE если фигура уже выбрана и FALSE, если не выбрана

Тесты

Будут реализованы позже

Стандарты языка

При создании приложения использовались:

- **c++98**
- **c++11**

Компилятор

GCC 4.9.1

Вывод

Был получен опыт работы с проектированием объектно-ориентированного приложения.

Реализация игры шахматы

Среда разработки

IDE: Qt Creator 3.5.0 (opensource)

OS: Debian 3.14.1

Документация: Doxygen 1.8.8

Компилятор: GCC 4.9.1

Анализатор: Valgrind 3.10.0

SVN: Git 2.1.4

chessEngine

Ядро содержит 18 классов, из которых 1 является интерфейсом ядра, 7 - исключения, 7 классов используется для представления фигур, один класс задаёт структуру клетки, и один класс задаёт структуру игровой доски.

Основные классы:

- **class myCell**
Класс, задающий структуру клетки на игровой доске
- **class myFigure**
Абстрактный класс, задающий общее представление об объектах фигур. От него наследуются все остальные классы фигур
- **class Pawn: public myFigure**
Класс для представления пешек. Содержит цвет и позицию фигуры. Наследуется от myFigure
- **class Knight: public myFigure**
Класс для представления фигуры "Конь". Содержит цвет и позицию фигуры. Наследуется от myFigure
- **class Rook: public myFigure**
Класс для представления Ладьи. Содержит цвет и позицию фигуры. Наследуется от myFigure
- **class Bishop: public myFigure**
Класс для представления фигуры "Слон". Содержит цвет и позицию фигуры. Наследуется от myFigure
- **class King: public myFigure**
Класс для представления фигуры "Король". Содержит цвет и позицию фигуры. Наследуется от myFigure
- **class Queen: public myFigure**
Класс для представления ферзя. Содержит цвет и позицию фигуры. Наследуется от myFigure
- **class BoardLogic**
Класс для представления игровой доски. Содержит двумерный массив указателей на фигуры

- **class OutOfBoardException**
Исключение, генерируемое при попытке походить за границы доски
- **class FigureNotSelectedException**
Исключение, генерируемое при попытке сделать ход, не выбрав фигуру
- **class SameColorFigureException**
Исключение, генерируемое при попытке атаковать собственную фигуру
- **class EmptyCellException**
Исключение, генерируемое при попытке выбрать пустую клетку
- **class AgainstTheRulesException**
Исключение, генерируемое при попытке переместить фигуру не по правилам игры
- **class WrongColorMoveException**
Исключение, генерируемое при попытке переместить фигуру не в свой ход
- **class KingKilledException**
Исключение, генерируемое при взятии короля

chessCUI

Содержит один класс **class Game**, реализующий консольное приложение.

chessGUI

Содержит 3 класса для представления и отрисовки различных окон графического приложения:

- **class MainWindow: public QMainWindow**
Класс для отображения нужного виджета в соответствии с запросом пользователя. С помощью QStackedWidget контролирует, чтобы одновременно отображалось только одно окно
- **class Dialog: public QDialog**
Класс для отображения начального меню
- **class Board: public Qidget**
Класс для графического представления игровой доски с фигурами

0.1 Листинг

Текст без отступа (следует за вставкой)

Новый параграф

Новый параграф с принудительно выключенным отступом

1 Выводы

Л^AT_EX удобен для создания отчётов, так как сам следит за нумерацией таблиц, рисунков, листингов и отсылок к ним (так, например, здесь всегда будет указан номер рисунка "sample" не зависимо от того, какой он (1,2 или другой) - это рисунок ??). Не менее важно что весь документ оформлен в едином стиле, а исходные материалы подключаются к отчёту, а не хранятся в нём. Всё это позволяет легко получить качественный отчёт без дополнительных трат на его оформление.

Исключения, пожалуй, составляют таблицы, так как их значительно сложнее создавать кодом, нежели в графическом редакторе. Но здесь никто не запрещает использовать визуальные средства создания таблиц для L^AT_EX .