

Муниципальное Автономное Общеобразовательное Учреждение
«Гимназия №14» г. Улан-Удэ

ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

Создание движка монгольских шахмат «Шатар»

Автор-разработчик: обучающийся 11 класса «М»

Лубсанов Д. А.

Руководитель: учитель информатики

Кононова О. В.

Работа допущена к защите «_____» марта 2023 г.

Подпись руководителя проекта _____ (Кононова О. В.)

г. Улан-Удэ

2023 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ОБЗОР И ТЕОРИЯ.....	5
1.1. Используемые технологии.....	5
1.2. Обзор игры.....	5
1.3. Теория шахматных движков.....	6
1.3.1. Шахматная доска.....	6
1.3.2. Оценка.....	8
1.3.3. Алгоритм.....	9
ГЛАВА 2. РЕАЛИЗАЦИЯ.....	10
ЗАКЛЮЧЕНИЕ.....	11
СПИСОК ИСТОЧНИКОВ.....	12
ПРИЛОЖЕНИЯ.....	13

ВВЕДЕНИЕ

Шахматы издавна были популярны во всём мире. Очень часто разные народы вносили свои собственные изменения в правила (собственно, современные шахматы — сильно изменившаяся древнеиндийская игра чатуранга).

Одним из таких вариантов является **«Шатар»** — монгольские шахматы. Само название — видоизменённое персидское «шатранг», а то, в свою очередь, происходит из вышеупомянутого «чатуранга».

Актуальность обусловлена вновь возросшим интересом к этому национальному виду шахмат. Открываются кружки, проводятся соревнования, турниры — например, в рамках фестивалей «Алтаргана» и «Сурхарбан». Они были впервые включены в их программу относительно недавно — в 2008 году.

Мною, к сожалению, не были найдены другие программные реализации этой игры.

2022 год объявлен Годом культурного наследия народов России; также многие направления определены в «Концепции сохранения и развития нематериального культурного наследия народов Российской Федерации...». Проводятся мероприятия по сохранению культур народов. Развивается и культура Бурятии; потому также важны усилия по популяризации народных игр, в том числе и «шатар».

Объект: монгольский вариант шахмат шатар.

Предмет: игровой движок шахмат шатар.

Цель проекта: написать на языке программирования C++ программу шахмат шатар.

Задачи:

1. Изучить информацию о шатар.

2. Изучить теорию шахматных движков
3. Написать шахматный движок на языке C++

ГЛАВА 1. ОБЗОР И ТЕОРИЯ

1.1. Используемые технологии

Создание движка будет вестись на языке C++. Это мощный и популярный язык для разработки настольных приложений.

Разработка будет вестись в среде CLion от компании JetBrains. Она также является популярной, имеет широкий набор возможностей. Кроме того, производитель предоставляет бесплатные лицензии учащимся.

Компилятором является GNU G++ 12.2, а системой сборки — CMake 3.26. Они оба имеют открытый исходный код и бесплатны. Проект создан при помощи дополнения cmake-init.

Для управления терминальным выводом используется библиотека rang[6].

1.2. Обзор игры

Как уже было сказано, шатар — это разновидность шахмат, и большинство правил схожи с европейскими шахматами. Так что имеет смысл описать различия. На самом деле, степень отличия может варьироваться, ведь среди шатар тоже встречаются разные варианты.

Возьмём такую версию[2][7]:

- Отсутствует рокировка;
- Пешка не может совершать двойное перемещение, кроме случаев начала игры. Таким образом, отсутствует взятие на проходе;
- В начале игры обязательно совершаются ходы d2d4 и d7d5 (можно считать это начальным состоянием игры);
- Ферзь может ходить только на одно поле по диагонали.

1.3. Теория шахматных движков

Шахматный движок — программа, переносящая игру в шахматы в форму компьютерной программы. Движки также обычно содержат в себе некий алгоритм, способный играть в шахматы.

Тогда шахматный движок как бы состоит из двух частей: шахматной доски и алгоритма игры (условно — искусственного интеллекта, ИИ), который, в свою очередь, состоит из механизмов *оценки* и *поиска*.

1.3.1. Шахматная доска

Существует множество способов представления шахматной доски[5] — например, можно было бы использовать «наивный»: создадим массив размером 8×8 , в каждой ячейке которого находится некая фигура. Ещё лучше будет создать доску размером 10×12 , тогда удобнее будет обработать фигуры, которые могут выйти за пределы поля.

Однако, уже давно существует более удобный (в первую очередь для компьютера) способ, использующий т.н. «битовые доски» (англ. *bitboards*)[4]. Его достоинства в том, что требуется меньше памяти, а также увеличивается скорость (за счёт использования встроенных операций). Так, можно сгенерировать ходы для пешек просто произведя побитовый сдвиг. Многие операции можно очень быстро выполнять — за $\mathcal{O}(1)$.

Назовём *битовой доской* двоичное число длины 64, i -й бит которого обозначает, находится ли в позиции i какая-либо фигура.

Для удобства отметим, что нумерация битов начинается со старшего (most significant) бита. Тогда получится такое соответствие квадратам доски:

Теперь шахматную доску можно представить как набор таких битовых досок — по одной на каждый тип фигуры, ещё две, показывающие, какой сто-

Таблица 1.1: Соответствие

	A	B	C	D	E	F	G	H
8	0	8	16	24	32	40	48	56
7	1	9	17	25	33	41	49	57
6	2	10	18	26	34	42	50	58
5	3	11	19	27	35	43	51	59
4	4	12	20	28	36	44	52	60
3	5	13	21	29	37	45	53	61
2	6	14	22	30	38	46	54	62
1	7	15	23	31	39	47	55	63

роне принадлежит фигура, а также — общая, т.е. показывающая наличие или отсутствие фигуры на доске в принципе. К ним понадобятся ещё несколько вспомогательных (инвертированных).

Доска и счётчик ходов составляют *позицию*.

Нам потребуется следить за повторяющимися позициями. Можно, конечно, хранить каждую позицию по отдельности, но это достаточно затратно. Поэтому можно каждую позицию *хешировать*, т.е. присвоить *хеш*. Такой хеш называют Zobrist-хешем[5].

Для того, чтобы совершать ходы, нужна некая структура, описывающая ход. Она содержит в себе собственно ход, то, какой фигурой сходили, съеденную фигуру (если таковая была), продвижение (если оно было).

Требуется реализовать правила. Сделаем так: для каждой позиции сгенерируем все возможные легальные ходы (т.е. те, которые не нарушают правил игры). Теперь можно легко проверить ход на правильность, найдя его в списке возможных ходов.

Позиция может быть инициализирована при помощи FEN-строки (строки в формате нотации Форсайта-Эдвардса — нотации, описывающей шахматную позицию).

1.3.2. Оценка

Оценка — это некоторое число, описывающее текущее состояние игры — насколько близко тот или иной игрок к победе (или проигрышу). Очевидно, это требуется для того, чтобы алгоритм поиска понимал, какие ходы выгоднее, а какие — нет.

Зачастую в данный момент времени невозможно сказать, кто выиграет. Но мы можем приблизительно понять, кто сейчас «ведёт». Скажем, что если число *меньше*, то игра идёт в пользу белых, а если оно *больше* — чёрных.

Из чего складывается эта оценка? Тут есть множество факторов, основные из них — это

- Материал — какие фигуры сейчас на доске,
- Мобильность — сколько они бьют полей,
- Расположение фигур.

Материал — это сумма цен всех фигур на доске. Вот они [5]:

Таблица 1.2: Цена фигуры

Фигура	Цена
Пешка	100
Конь	300
Слон	350
Ладья	500
Ферзь	900

Считается, что Король имеет очень большое значение; здесь же это просто не учитывается.

Мобильность рассчитывается из количества полей, которые бьют фигуры. За каждое поле даётся:

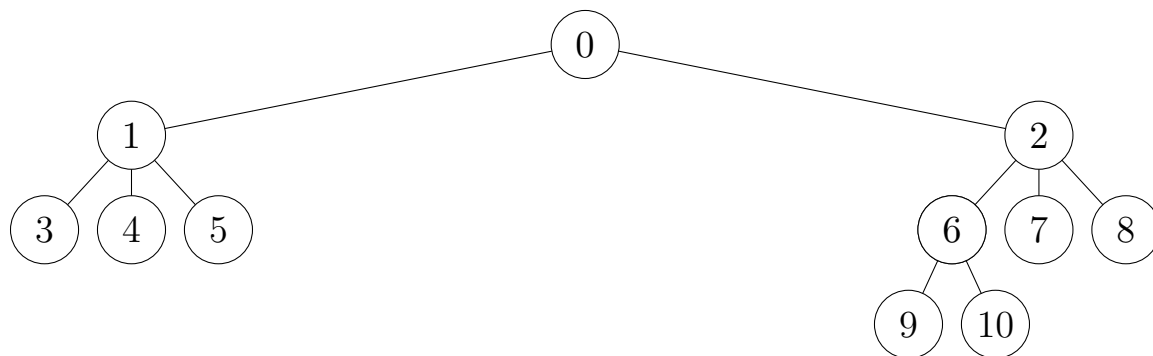
В положениях фигур учитываются только сдвоенные пешки (отнимает 25) и два слона (даёт 50).

Таблица 1.3: Цена фигуры

Фигура	Цена
Конь	9
Слон	4
Ладья	3
Ферзь	3

Имеется особая оценка для эндшпиля: единица дистанции между королями и королём и центром даёт 10 очков.

1.3.3. Алгоритм



ГЛАВА 2. РЕАЛИЗАЦИЯ

ЗАКЛЮЧЕНИЕ

СПИСОК ИСТОЧНИКОВ

1. Адельсон-Вельский, Г. М. Машина Играет в шахматы / Г. М. Адельсон-Вельский, В. Л. Арлазаров, А. Р. Битман, М. В. Донской; отв. ред. А. Ф. Волков. — М.:Наука, 1983. — 208 с.
2. Байнов, Е. В. Шатар — бурятские шахматы. / Е. В. Байнов, М. Е. Байнова. — Улан-Удэ: Детско-юношеская спортивная школа №8, 2015. — 26 с.
3. Корнилов, Е. Н. Программирование шахмат и других логических игр. — СПб.:БХВ-Петербург, 2005. — 272 с.
4. Browne, Cameron. Bitboard Methods for Games // ICGA Journal. — 2014. — Vol. 37. — No. 2. — pp. 65-84.
5. Chess Programming Wiki : [Electronic Resource]. — URL: <https://chessprogrammingwiki.org>.
6. rang: A Minimal, Header only Modern c++ library for terminal goodies // GitHub — The world's leading software development platform. — Access mode: <https://github.com/agauniyal/rang> (access date: 17.03.2023).
7. Shatar // Wikipedia. The Free Encyclopedia. — Access mode: <https://en.wikipedia.org/wiki/Shatar> (access date: 17.03.2023).

ПРИЛОЖЕНИЯ