

## Ifrit chess game

26.09.25- 26.09м.25

Сегодня решил сделать Ифрит с которым можно играть в браузере. Просто загружаете страницу и играете. Что получится пока не понимаю.

08.11.25- 08.11м.25

Пока был две недели в отпуске(с 20.10м.25 по 02.11м.25) с утра до вечера занимался программой. Это примерно 15 часов в день и 210 часов за отпуск.

### Код программы:

До сих пор код написан как простые скрипты без модулей. Это позволяет запускать программу локально, но очень неудобно так как все ошибки ловятся на стадии исполнения. И если в начале программа была маленькой и это было нормально, то сейчас это уже проблема. Очень много времени стало уходить на отлавливание ошибок которые компилятор находит мгновенно при модульной системе. Вспоминаю как здорово это организовано для TypeScript.

### Представление доски:

Движок основан на представлении доски 0x88. У нас одномерный массив от 0 до 127 можно представить как две доски рядом. Пока мы в пределах игровой доски проверка SquareIndex & 0x88 дает 0. Таким образом контролируется выход за пределы доски при генерации ходов когда фигуры стоят на краю доски.

### Генератор ходов:

В генераторе есть особенность. Он генерирует список на основе типа хода. Каждому типу хода присваивается свое имя. Особенно много для превращений и превращений с взятием. К примеру взятие ладьей пешки это отдельный тип хода. Всего получилось 60 типов ходов, причем 0 это нет хода. Благодаря такой системе хорошие взятия легко поднимаются вверх по списку.

В начале генератор был в одном файле и считал все ходы подряд. Устройство очень простое- мы идем по доске, считаем ходы фигур и записываем в список ходов. Потом дошел до того, что нужно считать только взятия для быстрого поиска. В связи с этим разделил генератор на два. Один считает взятия, а другой все остальные ходы. Генератор проверен на тестовых позициях. Далеко не сразу исправил все ошибки. Некоторые настолько необычные, что без теста даже бы не подумал что такое может быть. Ходы считаются без проверки на шах. Поэтому список ходов избыточен. Делатель ходов проверяет на шах после хода и если он есть то откатывает ход назад.

### Отдельный поток(worker):

Сразу сделал счет в одном потоке, но это привело к зависанию интерфейса. Пришлось организовать счет в отдельном потоке. Общение с потоком через строки. Сейчас что то похожее на протокол uci, однако ход движок передает в виде fen-позиции. Надо будет написать версию максимально приближенную к uci-протоколу. Локально потоки не работают. Приходится запускать сервер.

### Сделал несколько типов сортировки списка ходов:

Сортировка по типу хода позволяет поднять вверх хорошие взятия и вообще взятия. Также позволяет оставить ходы пешек на потом.

Вставляем киллера после взятий сдвигая все ходы до бывшей позиции хода-киллера вниз.

Сортируем ходы после взятий по исторической эвристике. Взятия в начале списка

находятся автоматически потому что в начале генерируются все взятия.

В киллеры записываем ходы вызвавшие отсечку по альфе или бете. В хеш-таблицу также записываем ходы вызвавшие отсечку по альфе или бете. В оценке истории тоже отмечаем ходы вызвавшие отсечку по альфе или бете.

Что еще нужно сделать:

- 1 автоматически тест генератора позиций.
- 2 uci-протокол.
- 3 максимально прокомментировать код программы.
- 4 возврат хода и просмотр хода игры.
- 5 вывод информации по хеш таблице.
- 6 автоматическая игра

Глобально дальше переходить на игру людей через сервер.

14.12.25- 14.12м.25

Из планов не сделал ничего. Все это время тестировал код. Надо сказать, что просто написать код это ноль. Самое трудное и важное это протестировать код. Ошибки есть всегда и даже в самом маленьком и прозрачном коде. Во время всевозможных тестов выяснилось, что у меня все эвристики были с ошибками.

В целом код опять стал для меня непонятным. И в нем много неработающих частей. Это все особенность js который указывает на ошибку только во время выполнения в браузере. В очередной раз проявляется тенденция, что собственный проект в котором каждая строчка написана мной становится чужим и непонятным. Читаешь все как будто писал кто то другой.

На уровне функций все нормально. Подробные комментарии и описание логики делает код понятным. Плюс еще код как правило обозрим. На уровне модулей по разному. Есть простые понятные, а есть на которых даже я смотрю и не понимаю что это. А ведь это я написал. :)

Лучше всего помогает разделение программы на независимые модули с четко прописанным входом и выходом и логикой работы, а также сомнительных моментов.

Идеально когда все состояния модуля прописаны и протестированы. Одно из фундаментальных правил такое, что любой код, даже самый очевидный и короткий, должен быть протестирован. Тут не должно быть исключений.

Еще один момент который нужно отметить это то, что надо смелее убирать не используемый код. Все эти соображения, что код в будущем понадобится нужно отметить. В крайнем случае скинуть версию в архив. Потому что проект со временем растет и все эти куски кода «на потом» сильно мешают.

17.01.26 - 17.01м.26

Начал добавлять комментарии jsdoc. Без них код js это тихий ужас. То, что я до сих пор писал просто на js это подвиг. Тестирование компенсирует недостаток но не до конца. В итоге много ошибок искажающих игру но совершенно незаметных. Код без тестирования это сломанный код и код без jsdoc комментариев это кривой код. Пока программа маленькая жить так можно, но стоит ей увеличиться уже следить становится невозможно.