Economy, strategy, conquest

Coins
Team #2

Каплун Виктория
Бучинский Иван

План

1. Кратко об игре
2. AIBot
3. SmartBot

Об игре

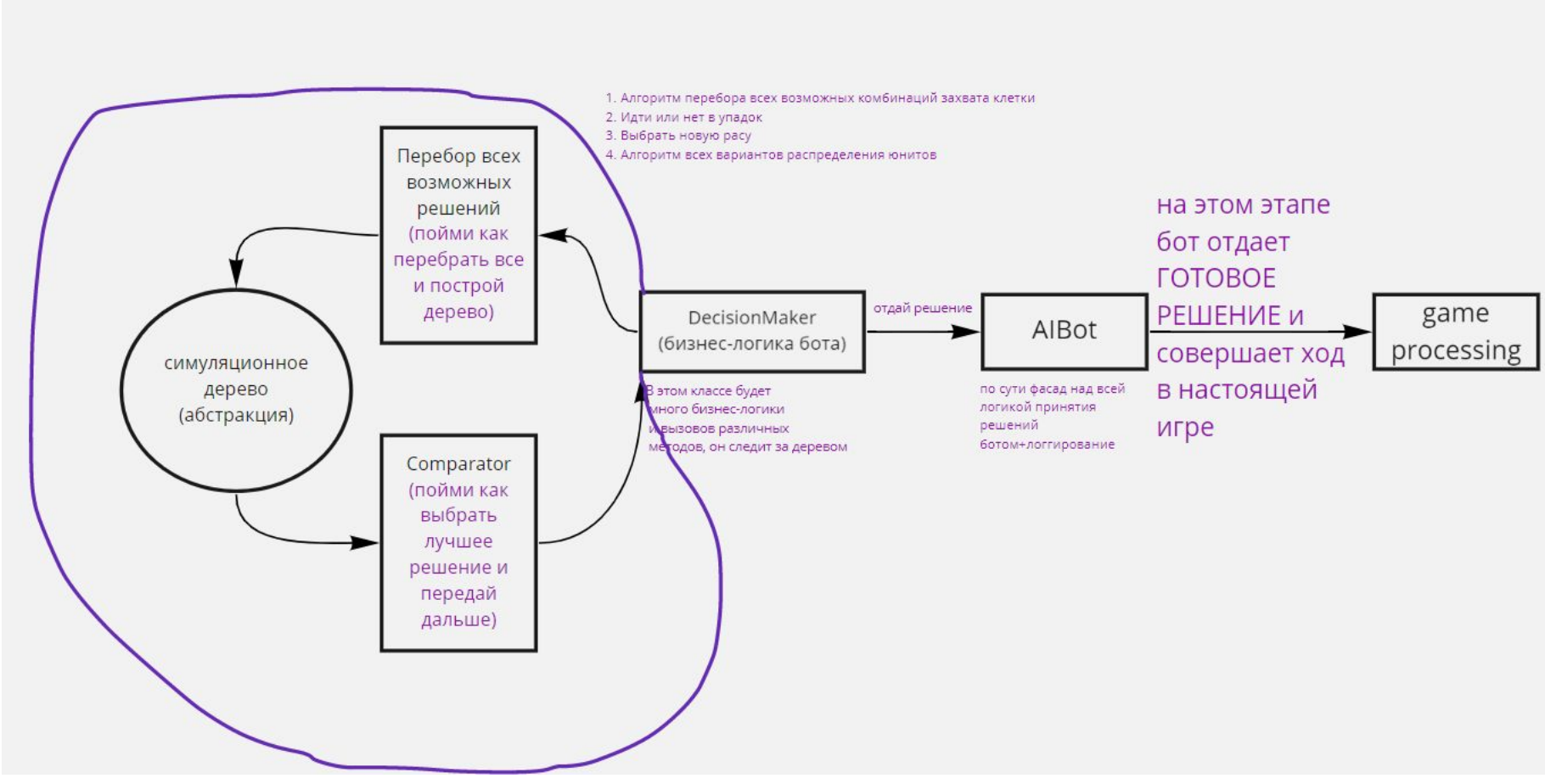

Настольная игра Филиппа Кейджита
SMALLWORLD
Маленький мир
DAYS OF WONDER
HOBBY WORLD
ПРАВИЛА ИГРЫ
SMALLWORLD
Маленький мир
Состав игры

AIBot

Задачи:

- создать бота, который обыгрывает “рандомного” бота

Архитектура

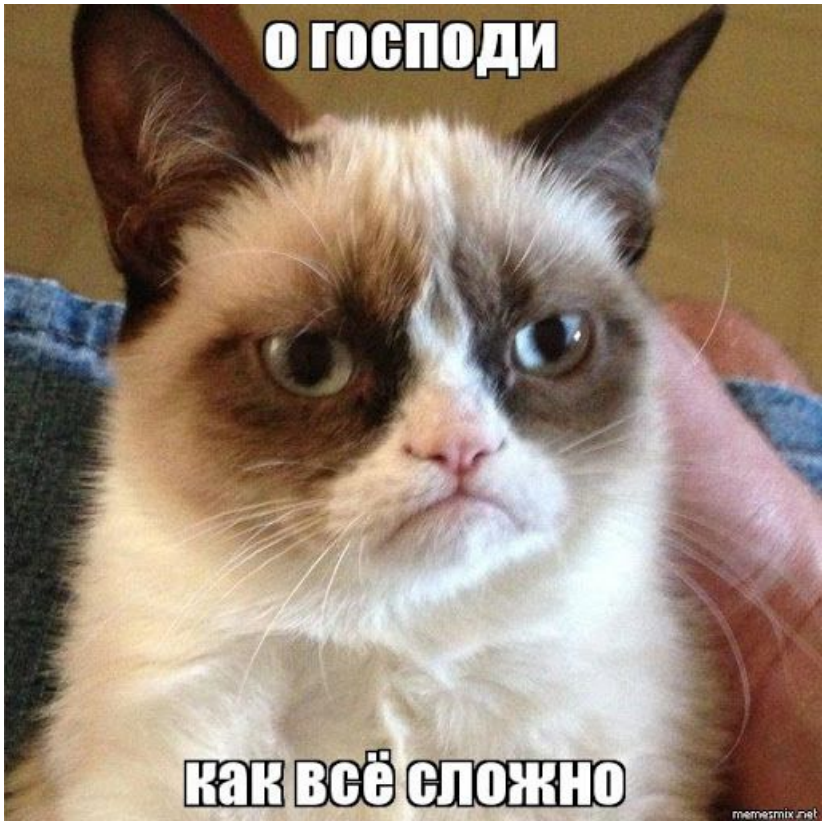


Изменения в архитектуре

много потраченного времени

В чем сложность

- ход состоит из нескольких частей



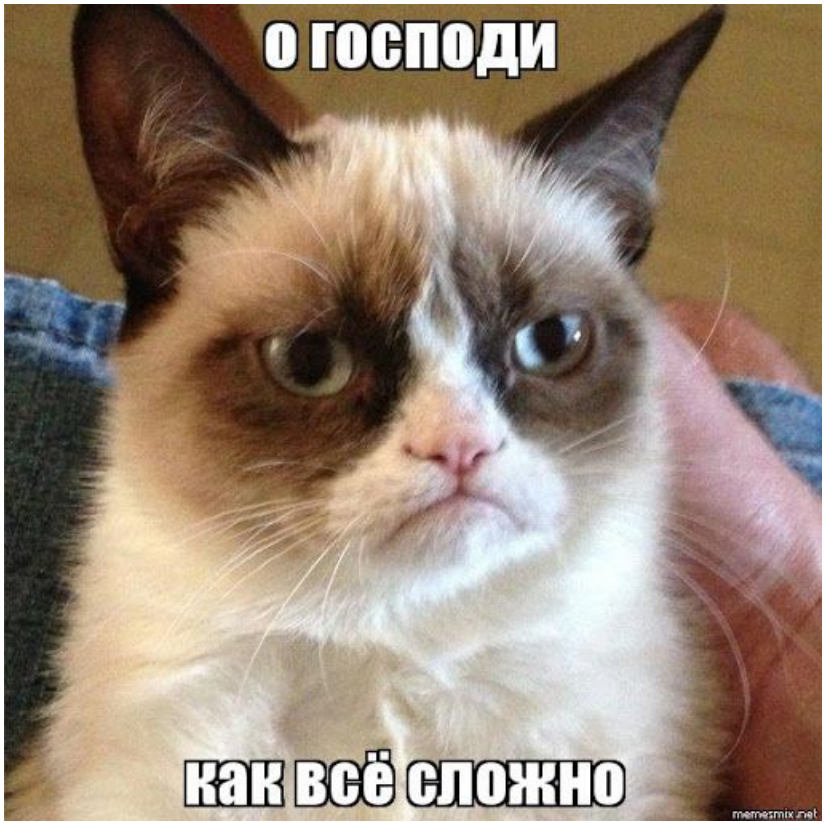
Решение

- ход состоит из нескольких частей

```
private static DecisionAndWin getBestDecisionByGameTree(final Player player, final IGame game,
                                                        @NotNull final DecisionType decisionType,
                                                        final int currentDepth) throws AIBotException {
    switch (decisionType) {
        case DECLINE_RACE: {
            return createDeclineRaceDecision(game, player, currentDepth);
        }
        case CHANGE_RACE: {
            return createChangeRaceDecision(game, player, currentDepth);
        }
        case CATCH_CELL: {
            return createCatchCellDecision(game, player, currentDepth);
        }
        case DISTRIBUTION_UNITS: {
            return createDistributionUnitsDecision(game, player, currentDepth);
        }
        default:
            throw new AIBotException(AIBotExceptionErrorCode.DECISION_NOT_EXISTS);
    }
}
```

В чем сложность

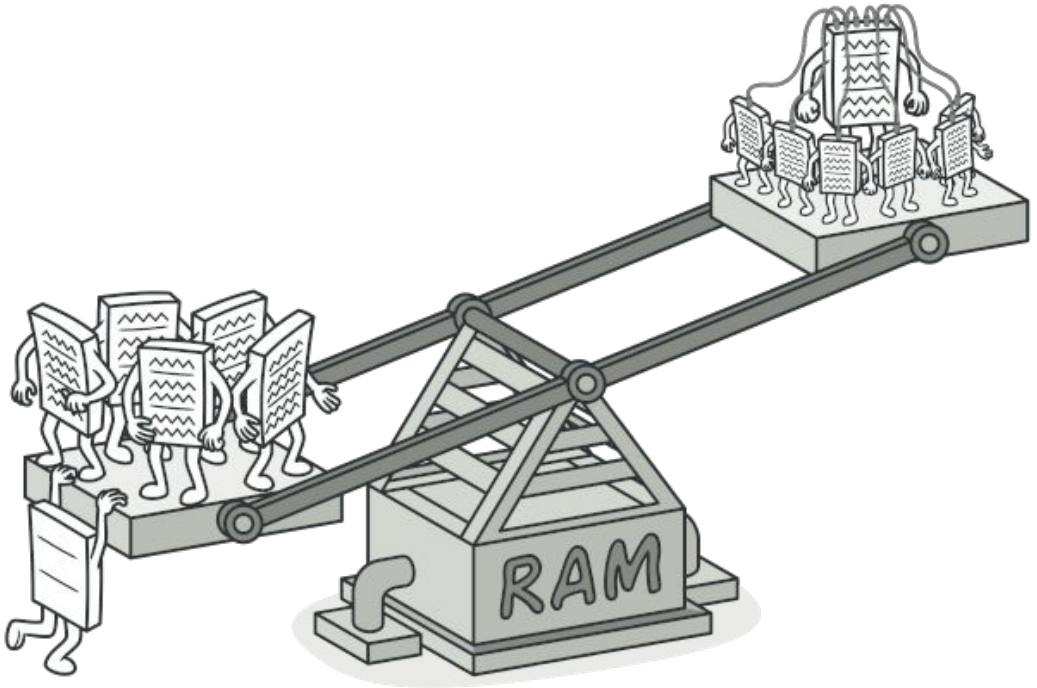
- большие затраты по времени и по памяти



Решение

- большие затраты по времени и по памяти

1. Паттерн Легковес



Решение

2. Умное копирование

```
@Contract(pure = true)
@JsonIgnore
@Override
public @NotNull Game getCopy() {
    final IBoard board = this.board.getCopy();
    final List<Player> players = new LinkedList<>();
    this.players.forEach(player -> players.add(player.getCopy()));
    final Map<Player, Set<Cell>> feudalToCells =
            getCopyPlayerToCellsSet(this.feudalToCells, isFeudal: true, this.board, board, players);
    final Map<Player, List<Cell>> ownToCells =
            getCopyPlayerToCellsList(this.ownToCells, isOwn: true, this.board, board, players);
    final Map<Player, List<Cell>> playerToTransitCells =
            getCopyPlayerToCellsList(this.playerToTransitCells, isOwn: false, this.board, board, players);
    final Map<Player, Set<Cell>> playerToAchievableCells =
            getCopyPlayerToCellsSet(this.playerToAchievableCells, isFeudal: false, this.board, board, players);

    return new Game(board, this.currentRound, feudalToCells, ownToCells, playerToTransitCells,
            playerToAchievableCells, this.gameFeatures, new LinkedList<>(this.racesPool), players);
}
```

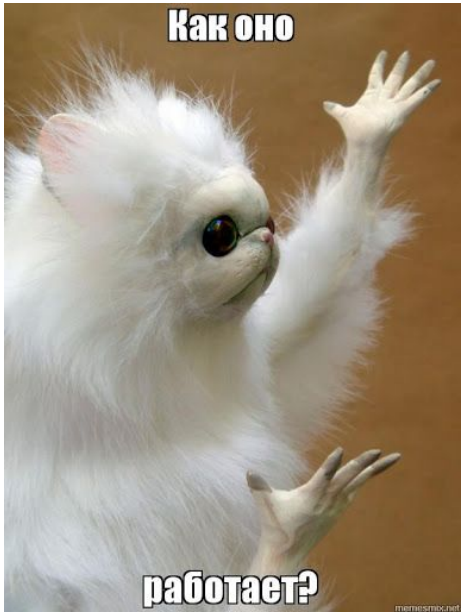
Решение

3. **Многопоточность**

Конфигурация бота

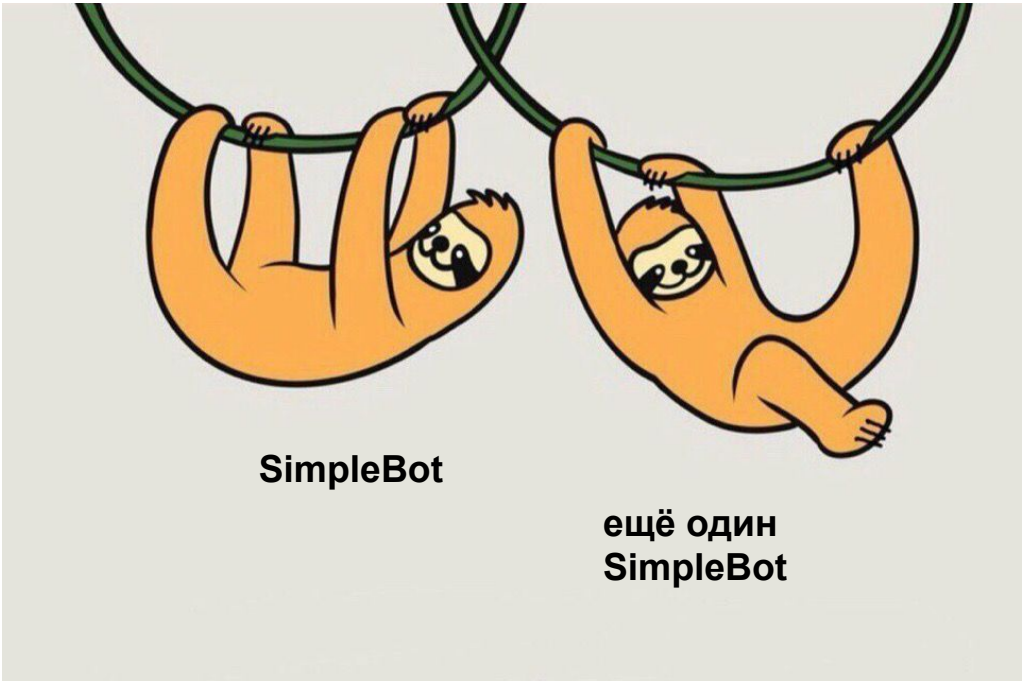
- глубина построения дерева - 1
- без построения хода оппонента
- бот захватывает одну клетку и переходит к этапу перераспределения юнитов
- функция оценки **sum(coins) -> max (жадный алгоритм)**

Результаты




```
[main] DEBUG i.n.i.coins.utils.LoggerFile - * Logging in file game
[main] INFO i.n.i.c.s.s.GameStatisticLogger - PLAYER: F1
[main] INFO i.n.i.c.s.s.GameStatisticLogger - WIN: 99
[main] INFO i.n.i.c.s.s.GameStatisticLogger - % : 100.0
[main] INFO i.n.i.c.s.s.GameStatisticLogger - PLAYER: F2
[main] INFO i.n.i.c.s.s.GameStatisticLogger - WIN: 0
[main] INFO i.n.i.c.s.s.GameStatisticLogger - % : 0.0
[main] DEBUG i.n.i.coins.utils.LoggerFile - * Logging in file game
```

Почему это работает



А что дальше?

- экспертиза
- увеличение глубины построения дерева
- какие результаты дают известные алгоритмы для этой игры

Вопросы?

Кратко о том, чем я занимался

Monetki2, Бучинский Иван
28.08.2020

Как всё работает

Как всё работает

- Дерево строится и обходится с помощью рекурсии;

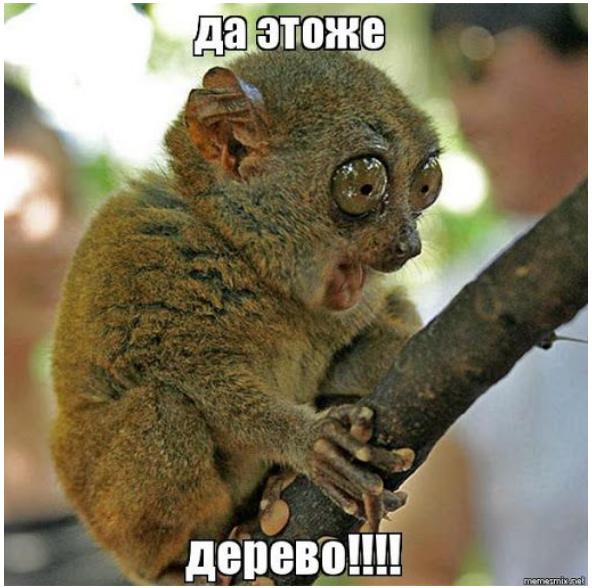
Как всё работает

- Дерево строится и обходится с помощью рекурсии;
- Бот выбирает лучшее для себя действие в зависимости от своего типа;

Кратко о структуре

Кратко о структуре

1. Классы для “дерева”: NodeTree, Edge, Action…



Кратко о структуре

1. Классы для “дерева”: NodeTree, Edge, Action…
2. Наборы функций для построения дерева обёрнутые в классы TreeCreator, AIProcessor;

Кратко о структуре

1. Классы для “дерева”: NodeTree, Edge, Action…
2. Наборы функций для построения дерева обёрнутые в классы TreeCreator, AIProcessor;
3. SmartBot, SmartClient.

Типы ботов

Типы ботов

1. MAX_PERCENT (максимизация вероятности выигрыша);

Типы ботов

1. MAX_PERCENT (максимизация вероятности выигрыша);
2. MIN_PERCENT (минимизация вероятности выигрыша оппонента);

Типы ботов

1. MAX_PERCENT (максимизация вероятности выигрыша);
2. MIN_PERCENT (минимизация вероятности выигрыша оппонента);
3. MIN_MAX_PERCENT (минимакс);

Типы ботов

1. MAX_PERCENT (максимизация вероятности выигрыша);
2. MIN_PERCENT (минимизация вероятности выигрыша оппонента);
3. MIN_MAX_PERCENT (минимакс);
4. MAX_VALUE (максимизация числа монет);

Типы ботов

1. MAX_PERCENT (максимизация вероятности выигрыша);
2. MIN_PERCENT (минимизация вероятности выигрыша оппонента);
3. MIN_MAX_PERCENT (минимакс);
4. MAX_VALUE (максимизация числа монет);
5. MIN_VALUE (минимизация числа монет оппонента);

Типы ботов

1. MAX_PERCENT (максимизация вероятности выигрыша);
2. MIN_PERCENT (минимизация вероятности выигрыша оппонента);
3. MIN_MAX_PERCENT (минимакс);
4. MAX_VALUE (максимизация числа монет);
5. MIN_VALUE (минимизация числа монет оппонента);
6. MIN_MAX_VALUE (минимакс);

Типы ботов

1. MAX_PERCENT (максимизация вероятности выигрыша);
2. MIN_PERCENT (минимизация вероятности выигрыша оппонента);
3. MIN_MAX_PERCENT (минимакс);
4. MAX_VALUE (максимизация числа монет);
5. MIN_VALUE (минимизация числа монет оппонента);
6. MIN_MAX_VALUE (минимакс);
7. MAX_VALUE_DIFFERENCE (максимизация отрыва от оппонента в числе монет);

Типы ботов

1. MAX_PERCENT (максимизация вероятности выигрыша);
2. MIN_PERCENT (минимизация вероятности выигрыша оппонента);
3. MIN_MAX_PERCENT (минимакс);
4. MAX_VALUE (максимизация числа монет);
5. MIN_VALUE (минимизация числа монет оппонента);
6. MIN_MAX_VALUE (минимакс);
7. MAX_VALUE_DIFFERENCE (максимизация отрыва от оппонента в числе монет);
8. MIN_MAX_VALUE_DIFFERENCE (минимакс).

Основная проблема и её решение

- Проблема оптимизации:

Основная проблема и её решение

- Проблема оптимизации:

    1. Обрезание с различной вероятностью поддеревьев, не особо ценных для нас. Их ценность определяется заведомо.

Основная проблема и её решение

- Проблема оптимизации:

    1. Обрезание с различной вероятностью поддеревьев, не особо ценных для нас. Их ценность определяется заведомо;

    2. ForkJoinPool как спасение от огромного числа потоков и прочих проблем многопоточности;

Основная проблема и её решение

- Проблема оптимизации:

    1. Обрезание с различной вероятностью поддеревьев, не особо ценных для нас. Их ценность определяется заведомо;

    2. ForkJoinPool как спасение от огромного числа потоков и прочих проблем многопоточности;

    3. Профилирование с помощью JVisualVM.

Основная проблема и её решение

- Проблема оптимизации:

    1. Обрезание с различной вероятностью поддеревьев, не особо ценных для нас. Их ценность определяется заведомо;

    2. ForkJoinPool как спасение от огромного числа потоков и прочих проблем многопоточности;

    3. Профилирование с помощью JVisualVM.

    4. Умное копирование

Про исследования

Про исследования

- Сервер может запускать любое конечное число игр подряд для подключившихся клиентов. Нужно лишь выбрать типы для ботов этих клиентов.

Про исследования

- Сервер может запускать любое конечное число игр подряд для подключившихся клиентов. Нужно лишь выбрать типы для ботов этих клиентов.
- Класс GameStatistic, запускающий SelfPlay столько, сколько нам нужно, с нужными нам числом ботов и их типами.

Основные исследовательские результаты

Основные исследовательские результаты

1. Гномы не “тащат”;

Основные исследовательские результаты

1. Гномы не “тащат”;
2. Под конец выгоднее брать те расы, которые дают доп. монеты: эльфы, грибы;

Основные исследовательские результаты

1. Гномы не “тащат”;
2. Под конец выгоднее брать те расы, которые дают доп. монеты: эльфы, грибы;
3. Грибы - для грибов, вода - для амфибий.

Спасибо за внимание!

Вопросы?