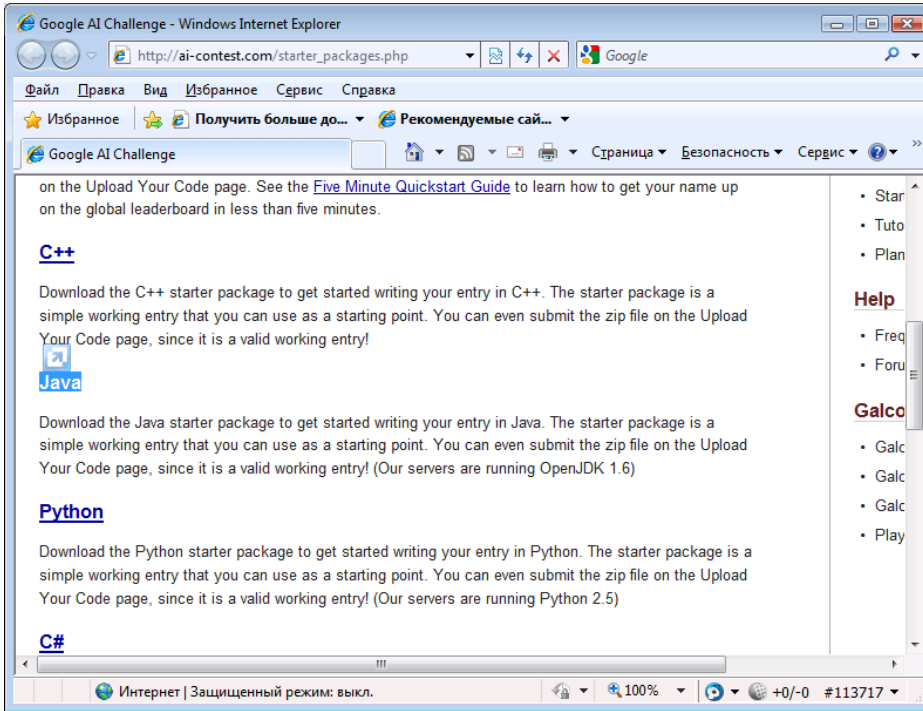
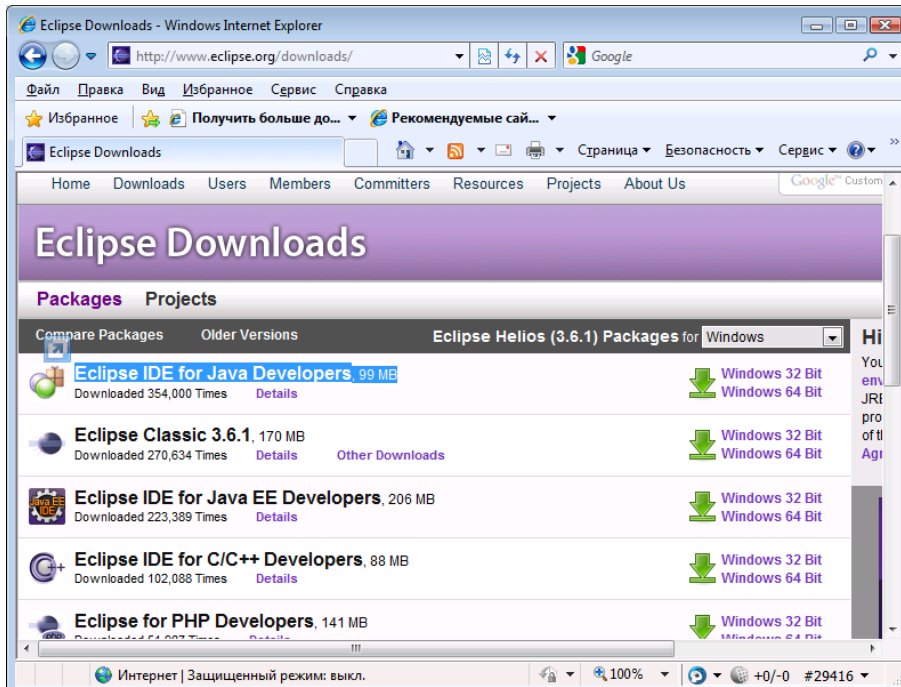


Google AI Challenge – Java.

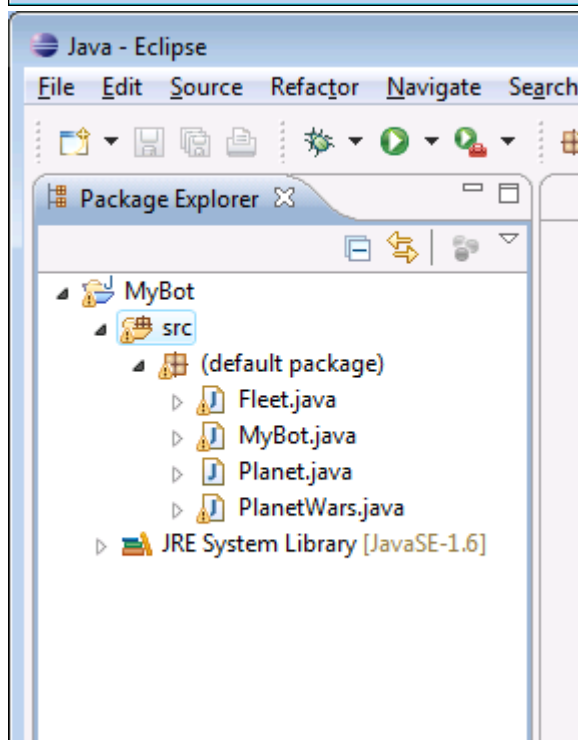
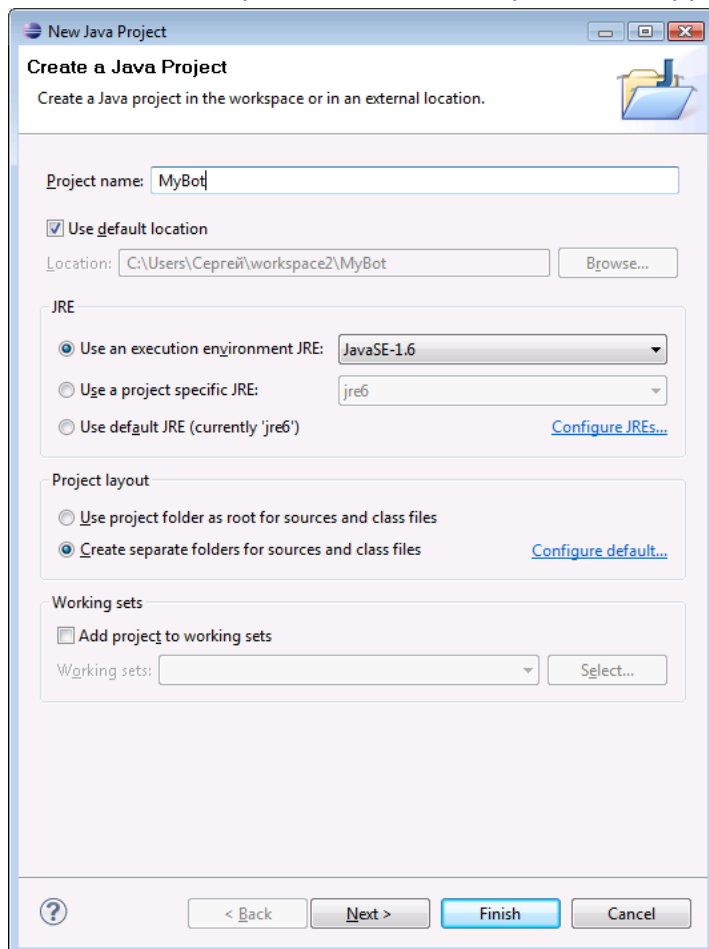
1. Качаем стартовый пакет для Java.



2. Качаем Eclipse IDE for Java Developers



3. Устанавливаем Eclipse, создаем новый проект и копируем исходники на java



4. Открываем MyBot.java и меняем оценочную функцию

Было:

Java - MyBot/src/MyBot.java - Eclipse

```

// (2) Find my strongest planet.
Planet source = null;
double sourceScore = Double.MIN_VALUE;
for (Planet p : pw.MyPlanets()) {
    double score = (double)p.NumShips();
    if (score > sourceScore) {
        sourceScore = score;
        source = p;
    }
}
// (3) Find the weakest enemy or neutral planet.
Planet dest = null;
double destScore = Double.MIN_VALUE;
for (Planet p : pw.NotMyPlanets()) {
    double score = 1.0 / (1 + p.NumShips());
    if (score > destScore) {
        destScore = score;
        dest = p;
    }
}
// (4) Send half the ships from my strongest planet to the weakest
// planet that I do not own.
if (source != null && dest != null) {
    int numShips = source.NumShips() / 2;
    pw.IssueOrder(source, dest, numShips);
}

```

Problems @ Javadoc Declaration
0 errors, 3 warnings, 0 others

MyBot.java - MyBot/src

Стало:

Java - MyBot/src/MyBot.java - Eclipse

```

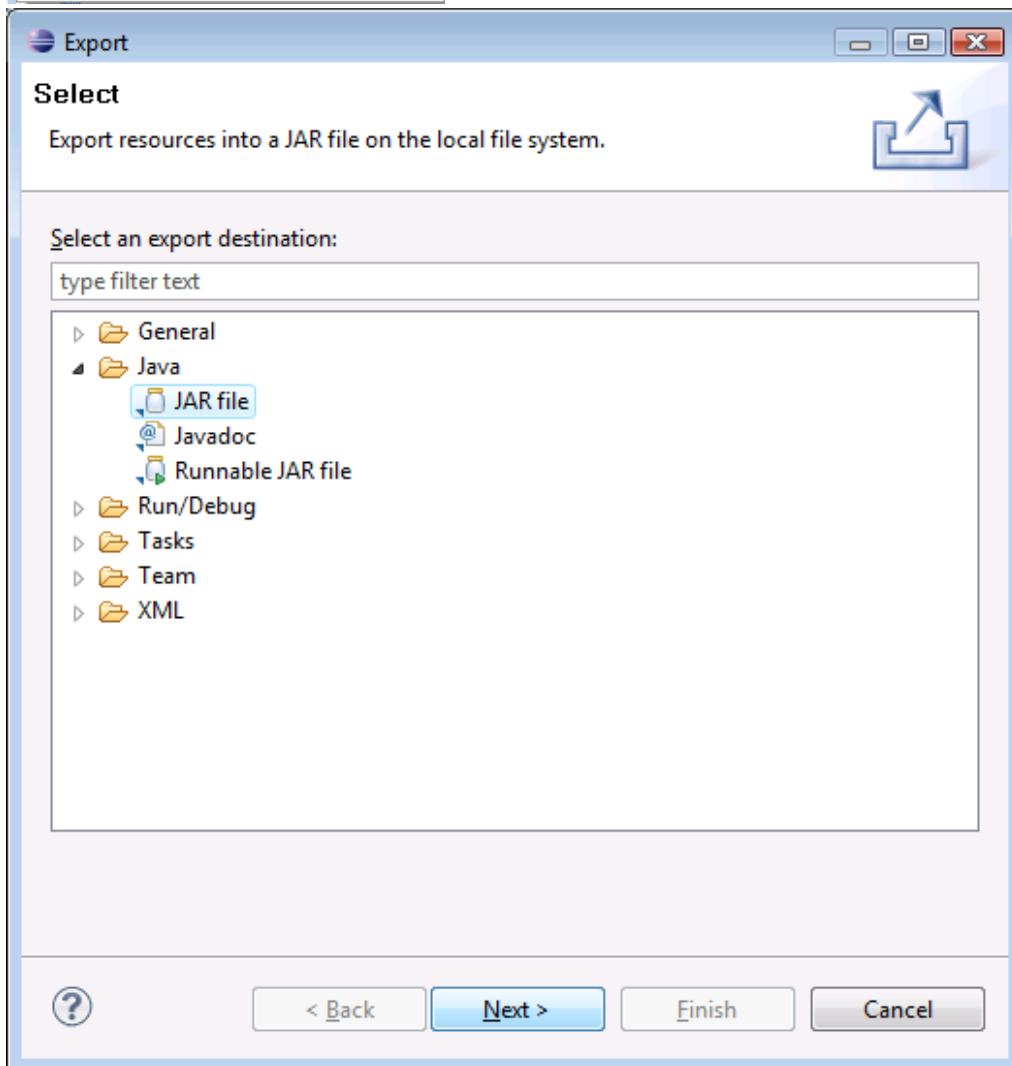
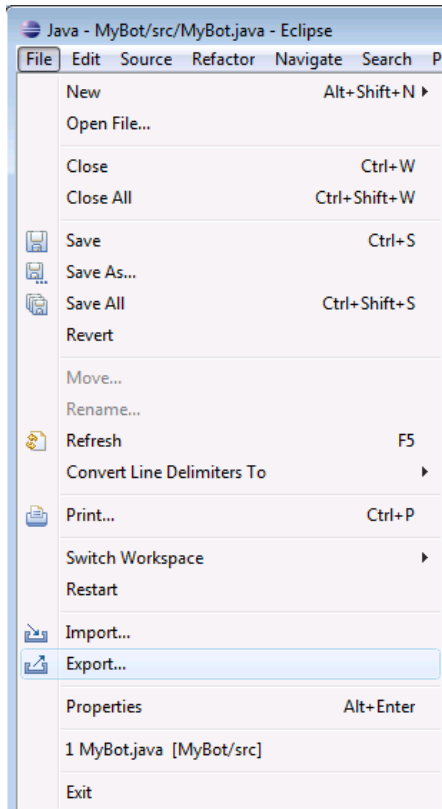
double score = (double)p.NumShips();
if (score > sourceScore) {
    sourceScore = score;
    source = p;
}
}
// (3) Find the weakest enemy or neutral planet.
Planet dest = null;
double destScore = Double.MIN_VALUE;
int destNumShips = 0;
for (Planet p : pw.NotMyPlanets()) {
    int distan = pw.Distance(source.PlanetID(), p.PlanetID());
    double score = (70 - distan)*p.GrowthRate();
    int pNumShips = p.NumShips();
    if (p.Owner() == 0) {score -= p.NumShips();}
    if (p.Owner() == 2) {pNumShips += distan*p.GrowthRate();}
    if (source.NumShips() < pNumShips + 1) {continue;}
    if (score > destScore) {
        destScore = score;
        dest = p;
        destNumShips = pNumShips + 1;
    }
}
// (4) Send half the ships from my strongest planet to the weakest
// planet that I do not own.
if (source != null && dest != null) {
    pw.IssueOrder(source, dest, destNumShips);
}
}
}

```

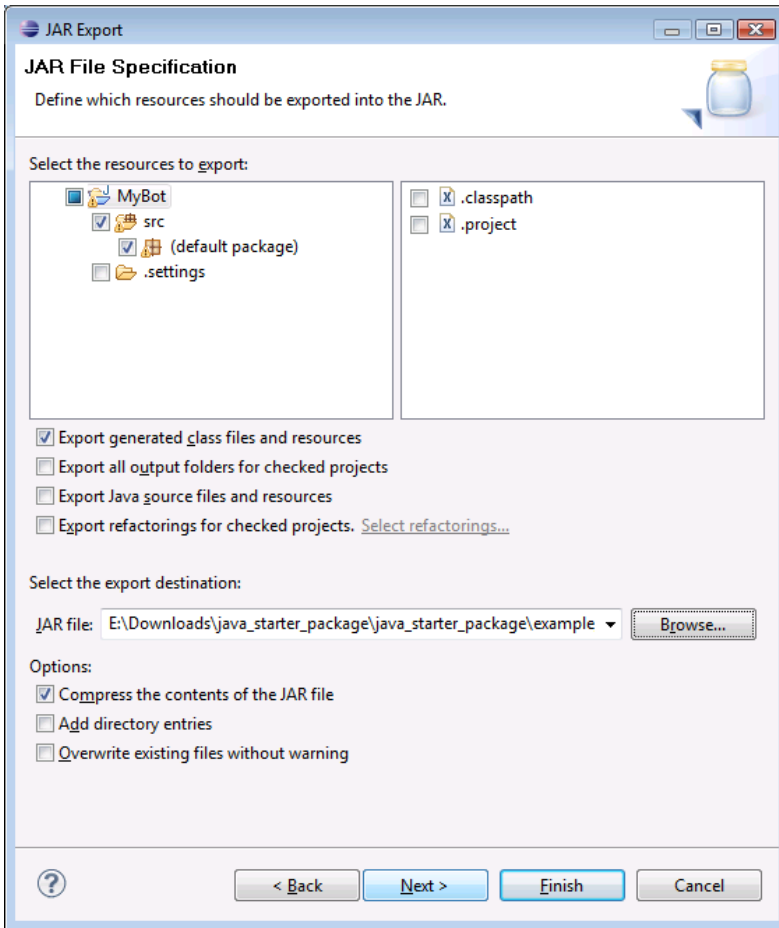
Problems @ Javadoc Declaration
0 errors, 3 warnings, 0 others

Writable Smart Insert 34:1

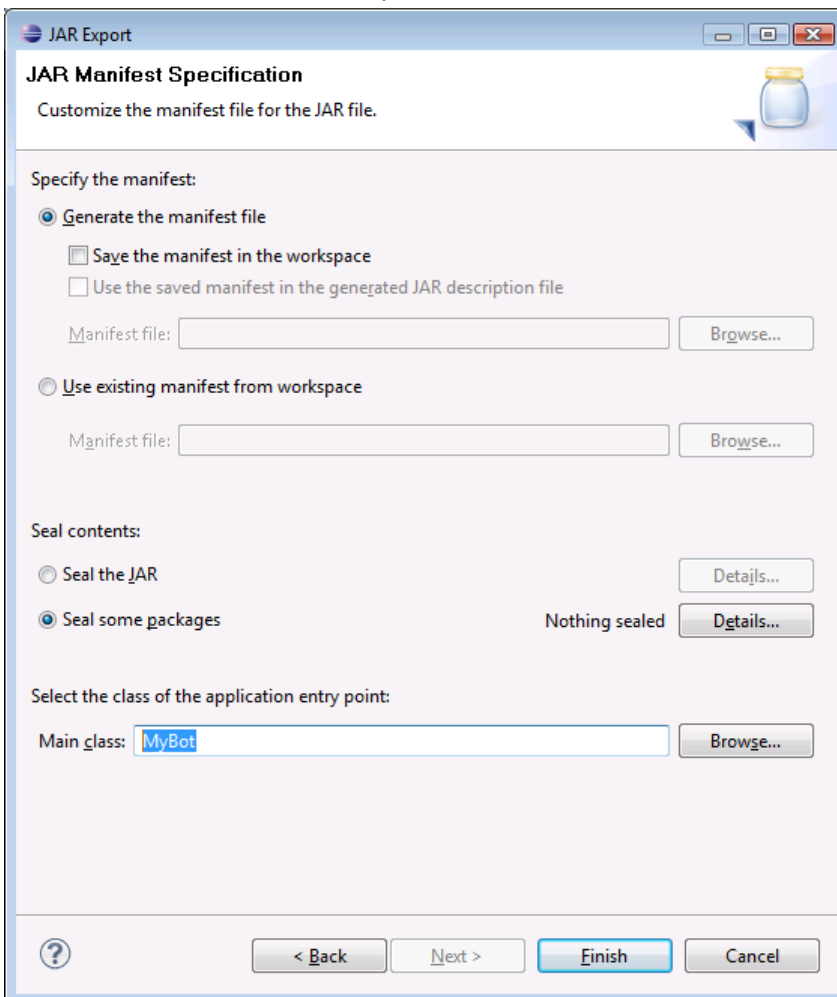
5. Экспортируем исходники в jar.



Выбираем наши исходники и задаем имя jar файла



Указываем основной класс (MyBot)



6. Запускаем тест нашего бота против любого из комплекта

```
java -Duser.language=en -jar tools/PlayGame.jar maps/map20.txt 1000 199 log.txt "java -jar example_bots/MyBotTest.jar" "java -jar example_bots/DualBot.jar" | java -jar tools/ShowGame.jar
```

Ура, все работает!

7. Проверяем результаты на 100 картах

```
set Bot1="java -jar example_bots/MyBotTest.jar"
set Bot2="java -jar example_bots/DualBot.jar"

echo off
del out1.txt
for /r maps %%i in (*.txt) do (
@echo %%i
java -Duser.language=en -jar tools/PlayGame.jar %%i 1000 199 log.txt %Bot1% %Bot2% > nul 2>> out1.txt
find /c "Player 1 Wins!" out1.txt
find /c "Player 2 Wins!" out1.txt
find /c "Draw!" out1.txt
find /c "WARN" out1.txt
)
Pause
```

С такими изменениями получаем счет 87:13 !

8. Смотрим результаты и возвращаемся к доработке бота.

Что нам нужно доработать:

- a. Пишем функцию анализа летящих кораблей - прогноз кол-ва кораблей и нового владельца
- b. Оставляем на планете столько кораблей, чтобы ее не захватили (актуально против RageBot)
- c. Оптимизируем оценочную функцию
- d. Переходим в мультизадачность – каждая планета должна на каждом ход уметь посылать нужное кол-во кораблей для захвата или подкрепления.
- e. Защищаем собственные планеты от захвата летящих кораблей противника
- f. Посылаем подкрепление с тыла к фронту боевых действий

Приведу несколько примеров работы с Java.

Условия

```
if (p1.FreeNumShips() > p2.FreeNumShips()) {
    return -1;
} else if (p1.FreeNumShips() < p2.FreeNumShips()) {
    return 1;
}
```

== - равно

!= - не равно

```
&& - И  
|| - ИЛИ
```

Циклы

```
for (Planet pd : Planets) {  
    ...  
    continue;  
}  
  
int i;  
for (i = 0; i < 3; i++) {  
    ...  
    break;  
}
```

Сортировка

```
Collections.sort(Fleets, new Comparator<Fleet>() {  
    public int compare(Fleet p1, Fleet p2) {  
        int p1TurnsRemaining = p1.TurnsRemaining();  
        int p2TurnsRemaining = p2.TurnsRemaining();  
        if (p1TurnsRemaining > p2TurnsRemaining) {  
            return 1;  
        } else if (p1TurnsRemaining < p2TurnsRemaining) {  
            return -1;  
        }  
        return 0;  
    }  
});
```

Массивы

```
static int[][] Dist;  
Dist = new int[Planets.size()][Planets.size()];  
Dist[p1.PlanetID()][p2.PlanetID()] = cache;  
int cache = Dist[p1.PlanetID()][p2.PlanetID()];
```

Возврат из функции множества значений

```
// В функции возвращаем  
HashMap<String, Object> ret = new HashMap<String, Object>();  
ret.put("Owner", Owner);  
ret.put("NumShips", NumShips);  
  
// Так вызываем функцию и получаем результаты  
HashMap<String, Object> stat = statistics(...);  
int newOwner = (Integer) stat.get("Owner");  
NumShips = (Integer) stat.get("NumShips");
```

Различные функции

```
int c = Math.min(a, b);  
int c = Math.max(a, b);  
double s = Math.abs(d);
```

Вывод лога

```
import java.io.*; // подключаем нужные классы в самом начале  
  
FileWriter writer = new FileWriter("logMyBot.txt");  
BufferedWriter buffered = new BufferedWriter(writer);  
buffered.write("START");  
buffered.newLine();
```

```
buffered.flush();
buffered.close();
writer.close();
```

При этом методы должны быть дополнены инструкцией **throws IOException**.
Например,
`public static void DoTurn(PlanetWars pw) throws IOException {`

Методы PlanetWars

Planets() - список всех планет
MyPlanets() - список моих планет (где Owner = 1)
EnemyPlanets() - список планет противника (где Owner = 2)
NotMyPlanets() - список чужих планет (где Owner != 1)
GetPlanet(planetID) - получить объект Планета по индексу
Fleets() - список всех летящих флотов
Distance(sourcePlanet, destinationPlanet) - возвращает расстояние между планетами.
IssueOrder(sourcePlanet, destinationPlanet, numShips) - запись команды на посыл кораблей
FinishTurn() - окончание хода (посылается "go")

Методы Planet

PlanetID() - возвращает номер планеты
Owner() - текущий владелец планеты (0 - нейтральная, 1 - наша, 2 - противника)
NumShips() - кол-во кораблей на планете
GrowthRate() - прирост на планете
AddShips(amount) - увеличить кол-во кораблей на планете
RemoveShips(amount) - уменьшить кол-во кораблей на планете

Методы Fleet

Owner() - текущий владелец флота (1 - наши войска, 2 - противника)
NumShips() - кол-во кораблей во флоте
SourcePlanet() - номер планеты источника, откуда отправился флот
DestinationPlanet() - номер планеты назначения, куда летит флот
TurnsRemaining() - кол-во ходов до прилета флота на планету назначения

Кроме этих конструкций и функций из стартового комплекта больше ничего не надо для создания умного бота для игры.