

Шахматоподобные игры

Ежков Марк
Поляков Денис

23 мая 2014 г.

Основные идеи и принципы

- Возможность играть в произвольные шахматоподобные игры в одном приложении.
- Правила игры описываются на XML.
- Приложение должно обладать графическим интерфейсом и искусственным интеллектом.

Параметры игрового поля. Начальная расстановка.

- Наиболее простой и естественной частью языка является описание игрового поля и начальной расстановки фигур.

```
1  <?xml version="1.0"?>
2  <game>
3
4      <board xsize = "8" ysize = "8">
5          <rect x = "1" y = "1" xsize = "8" ysize = "8"/>
6
7      </board>
8
9
10     <player id = "1" color = "black" name = "Blacks" orientation = "down" turn = "2">
11         <start id = "6" x = "1" y = "7"/>
12         ...
13         <start id = "2" x = "4" y = "8"/>
14
15     </player>
16
17     <player id = "2" color = "white" name = "Whites" orientation = "up" turn = "1">
18         <start id = "6" x = "1" y = "2"/>
19         ...
20         <start id = "2" x = "4" y = "1"/>
21
22     </player>
23
24 </game>
```

Описание игровых фигур.

- Фигура обладает набором атрибутов: id, weight, image.

```
1 <piece id = "2" name = "Queen" short = "Qn" weight = "9">
2
3   <moves>
4       <slide x = "0" y = "1"/>
5       <slide x = "1" y = "0"/>
6       <slide x = "0" y = "-1"/>
7       <slide x = "-1" y = "0"/>
8       <slide x = "1" y = "1"/>
9       <slide x = "1" y = "-1"/>
10      <slide x = "-1" y = "1"/>
11      <slide x = "-1" y = "-1"/>
12   </moves>
13
14   <captures diff = "0"/>
15
16 </piece>
```

Описание игровых фигур.

- Простые перемещения и взятия описываются тегами `<jump/>` и `<slide/>`.

```
1  <?xml version="1.0"?>
2  <game>
3    <piece id = "5" name = "Knight" short = "Kt" weight = "3">
4      <moves>
5        <jump x = "2" y = "1"/>
6        ...
7        <jump x = "-2" y = "-1"/>
8      </moves>
9
10     <captures diff = "0"/>
11
12   </piece>
13
14   <piece id = "3" name = "Rook" short = "Rk" weight = "5">
15     <moves>
16       <slide x = "0" y = "1"/>
17       ..
18       <slide x = "-1" y = "0"/>
19     </moves>
20
21     <captures diff = "0"/>
22
23   </piece>
24
25 </game>
```

Описание игровых фигур.

- Сложные ходы, например взятие на проходе, описываются тегом `<special/>`.

```
1  <?xml version="1.0"?>
2  <game>
3
4      <piece id = "6" name = "Pawn" short = "Pn" weight = "1">
5
6          ...
7
8          <special id = "1">
9              <self moved = "0" x = "0" y = "2"/>
10             <free x = "0" y = "1"/>
11             <free x = "0" y = "2"/>
12         </special>
13
14         <special>
15             <self x = "1" y = "1"/>
16             <prey id = "6" moved = "1" x = "1" y = "0"/>
17             <free x = "1" y = "1"/>
18         </special>
19
20         <special>
21             <self x = "-1" y = "1"/>
22             <prey id = "6" moved = "1" x = "-1" y = "0"/>
23             <free x = "-1" y = "1"/>
24         </special>
25
26     </piece>
```

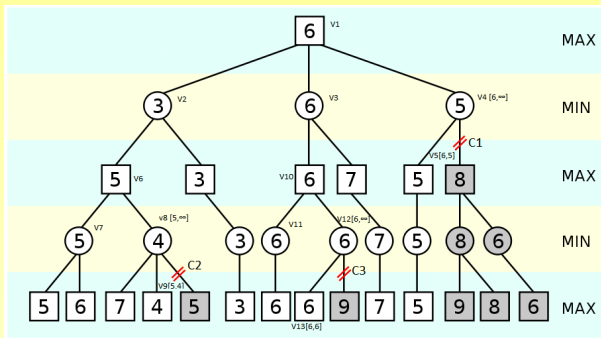
Модель

Тесты

- Для тестирования мы использовали unit testing.
- Главной задачей тестирования была проверка корректности выполнения ходов.
- Модель позволяет в теле теста автоматически разыгрывать различные шахматные партии и их части.

Искусственный интеллект.

- Интеллект реализуется алгоритмом минимакса с модификацией: альфа-бета отсечение.



Искусственный интеллект.

- Для работы интеллекта требуется оценочная функция позиции и список всех возможных ходов, также необходимо создавать копии объекта Game.

```
1 public class Game implements Cloneable {
2
3     ...
4
5     public ArrayList<Route> allDestinations() {
6         ArrayList<Route> destinations = new ArrayList<Route>();
7         ...
8         return destinations;
9     }
10
11     public int estimation() {
12         int ownEstimation = 0;
13         ...
14         return ownEstimation;
15     }
16
17     public Game clone() throws CloneNotSupportedException {
18         Game newGame = (Game) super.clone();
19         newGame.board = (Board) board.clone();
20         return newGame;
21     }
22 }
```