Soltoon: Bokosh Bokosh

Technical documentation

V1.1.2

December, 2017

Authors:

Payam Mohammadi (payam.int@gmail.com)
Amirkasra Jalaldoust (amirkasraj@gmail.com)

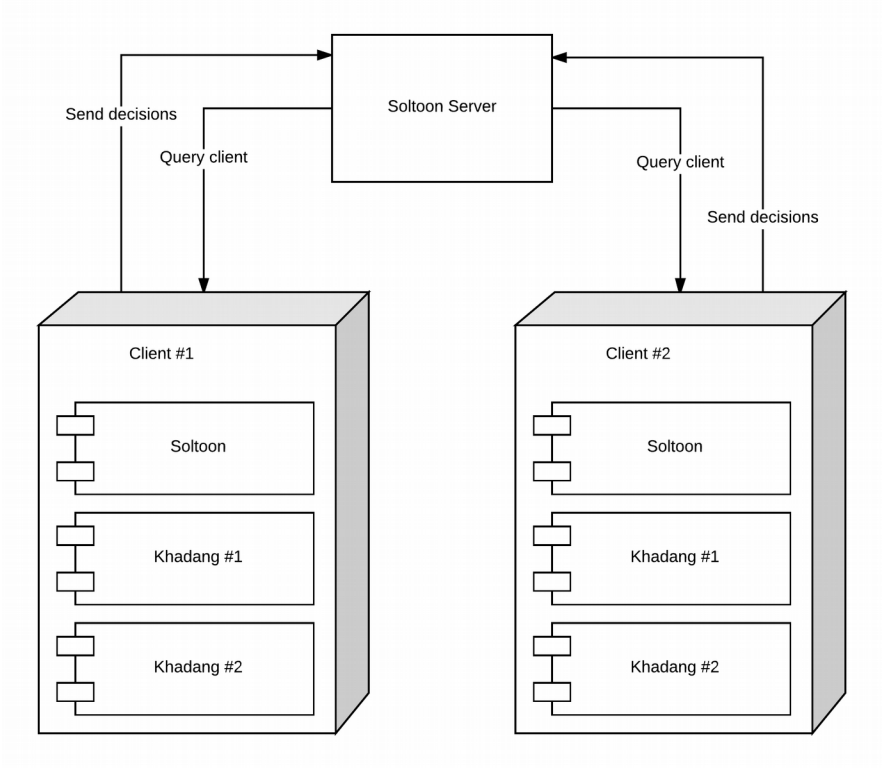
https://soltoon.net

بازی سلطون

بازی سلطون چگونه انجام می‌شود ؟

هر بازی سلطون یک یا چند شرکت کننده دارد. هر شرکت‌کننده برنامه‌ای دارد که تصمیمات سلطون خود و خدنگ‌های خود را مشخص می‌کند.

برنامه هر شرکت کننده به برنامه‌ی بازی متصل می‌شود و برنامه‌ی بازی یک رقابت میان آن‌ها برگزار می‌کند.



بازی در چند دور برگزار می‌شود. در هر دور برنامه‌ی بازی ابتدا از سلطون‌ها می‌خواهد که خدنگ‌هایشان را ایجاد کنند، سپس از هر خدنگ در بازی می‌خواهد که تصمیمش را اعلام کند.

وظیفه شما

وظیفه شما پیاده سازی یک برنامه است که در مورد حرکات سلطون و خدنگ‌هایتان در یک رقابت تصمیم بگیرد.

برای آشنایی شما با چالش‌هایی که با آن‌ها روبرو هستید چند سناریوی آموزشی نظر گرفته شده است.

اجرای بازی

دانلود آخرین نسخه بازی

ابتدا باید آخرین نسخه بازی سلطون را دانلود کنید و به پروژه‌تان اضافه کنید. برای دانلود به لینک زیر بروید و آخرین نسخه سلطون را دانلود کنید:

https://soltoon.github.io/

اضافه کردن فایل jar به پروژه

1. در محل پروژه‌تان دایرکتوری lib را ایجاد کنید.
2. فایل soltoon-game-1.1.0-jar-with-dependencies.jar را در پوشه lib قرار دهید.
3. در نرم افزار Intellij Idea در پنجره Project، پوشه lib، روی **فایل jar** راست کلیک کرده گزینه Add as library را انتخاب کنید.

اگر از IntelliJ Idea استفاده نمی‌کنید به (https://soltoon.github.io) مراجعه کنید.

مرحله سوم: رقابت

برای برگزاری یک رقابت در اولین قدم باید برنامه شرکت‌کنندگان بازی اجرا شود.

برای اجرای هر کلاینت باید یکی از متد های زیر را صدا کنید:

ClientRunner	
`static void run(Class<? extends Soltoon> soltoonClass)`	اجرای کلاینت بعنوان بازیکن اول
`static void runPlayerTwo(Class<? extends Soltoon> soltoonClass)`	اجرای کلاینت بعنوان بازیکن دوم
`static void run(Class<? extends Soltoon> soltoonClass, ComRemoteConfig remoteConfig)`	اجرای کلاینت

- پارامتر `soltoonClass` کلاس سلطونی است که باید اجرا شود.

مثال:

اجرای یک کلاینت بعنوان بازیکن اول

Client1.java

```java
import ir.pint.soltoon.soltoongame.client.ClientRunner;
import ir.pint.soltoon.soltoongame.client.implementations.SoltoonSakht;
public class Client1 {
  public static void main(String[] args) {
    ClientRunner.run(SoltoonSakht.class);
  }
}
```

اجرای یک کلاینت بعنوان بازیکن دوم

Client2.java

```java
import ir.pint.soltoon.soltoongame.client.ClientRunner;
import ir.pint.soltoon.soltoongame.client.implementations.SoltoonSakht;
public class Client2 {
  public static void main(String[] args) {
    ClientRunner.runPlayerTwo(SoltoonSibl.class);
  }
}
```

بعد از اجرای کلاینت‌ها باید سرور را اجرا کنید. برای اجرای سرور از متدهای زیر استفاده کنید:

ServerRunner	
`static void run()`	اجرای بازی با دو بازیکن
`static void runHelloWorld()`	اجرای سرور با سناریوی آزمایشی «سلام‌دنیا!»
`static void runName(int width, int height)`	اجرای سرور با سناریوی آزمایشی «نام»
`static void runWarrior()`	اجرای سرور با سناریوی آزمایشی «جنگجو»
`static void runWarrior2()`	اجرای سرور با سناریوی آزمایشی «جنگجو» این متد یک پیاده‌سازی متفاوت از سناریوی جنگجو را اجرا می‌کند. توضیحات این پیاده‌سازی: https://github.com/Soltoon/soltoon-game/pull/2

مثال:

Server.java

```java
import ir.pint.soltoon.soltoongame.server.ServerRunner;
public class Server {
  public static void main(String[] args) {
    ServerRunner.run();
  }
}
```

اجرای بازی آزمایشی

برای اجرای بازی آزمایشی می‌توانید از سلطون‌های آزمایشی بازی استفاده کنید و کلاس مربوط به آن‌ها را بعنوان پارامتر به ClientRunner ورودی بدهید.

سلطون‌های آزمایشی:

SoltoonSakht	این سلطون یک حریف معمولی است.
SoltoonSibl	این سلطون فقط خدنگ‌هایی ایجاد می‌کند و خدنگ‌ها بطور تصادفی حرکت‌می‌کنند. این سلطون به شما حمله نخواهد کرد.

مثال:

Client2.java

```java
import ir.pint.soltoon.soltoongame.client.ClientRunner;
import ir.pint.soltoon.soltoongame.client.implementations.SoltoonSakht;
public class Client2 {
  public static void main(String[] args) {
    ClientRunner.runPlayerTwo(SoltoonSakht.class);
  }
}
```

اگر می‌خواهید با محیط بازی سلطون آشنا شوید میتوانید یک رقابت بین SoltoonSakht و SoltoonSibl برگزار کنید.

پیاده‌سازی برنامه

برای پیاده‌سازی سلطون و خدنگ‌هایتان باید کلاس‌های Soltoon و Khadang را پیاده‌سازی کنید. درهر دوی این کلاس‌ها باید متد های زیر را پیاده‌سازی کنید.

Agent (Soltoon, Khadang)	
`abstract void init(Game g)`	~~این متد بعد از اینکه سلطون/خدنگ ایجاد شدند صدا زده می‌شود. صدا زده شدن این متد به این معناست که ایجاد این سلطون/خدنگ سمت سرور ایجاد شده است.~~ این متد قبل از اینکه خدنگ در سمت سرور ایجاد شود صدا زده می‌شود.
`abstract void lastThingsToDo(Game g)`	این متد هنگام کشته شدن خدنگ صدا زده می‌شود.
`abstract Action getAction(Game g)`	در هر نوبت تصمیم هر سلطون/خدنگ با این متد دریافت می‌شود. برای خدنگ در هر دور این متد یک‌بار صدا زده می‌شود ولی برای سلطون ممکن است بیشتر از یک‌بار صدا زده شود. این متد یک شیء از نوع Action باز می‌گرداند. در صورتی که null برگرداند هیچ تصمیمی گرفته نمی‌شود. برای سلطون Actionها AddKhadang و برای خدنگ‌ها Move و Shoot مجاز است.

در ادامه یک نمونه از این کلاس‌ها پیاده سازی می‌کنیم. هدف‌ما در این بخش پیاده‌سازی یک سلطون است که در نقطه (1,1) یک خدنگ ایجاد می‌کند که این خدنگ دائما به نقطه (0,0) شلیک میکند.

قدم اول: نوشتن کلاس مربوط به خدنگی که به نقطه (0,0) شلیک می‌کند

همه خدنگ‌ها باید کلاس Khadang را Extend کنند. خدنگ‌ما هنگام ایجاد و هنگام کشته‌شدن کاری برای انجام دادن ندارد پس فقط متد getAction را پیاده‌سازی می‌کنیم.

ShootZeroZero.java

```java
import ir.pint.soltoon.soltoongame.shared.actions.Action;
import ir.pint.soltoon.soltoongame.shared.actions.Shoot;
import ir.pint.soltoon.soltoongame.shared.agents.Khadang;
import ir.pint.soltoon.soltoongame.shared.map.Game;
import ir.pint.soltoon.soltoongame.shared.map.KhadangType;
public class ShootZeroZero extends Khadang {
  public ShootZeroZero(KhadangType type) {
    super(type);
  }
  @Override
  public void init(Game g) {
    // do nothing
  }
  @Override
  public void lastThingsToDo(Game g) {
    // do nothing
  }
  @Override
  public Action getAction(Game g) {
    return new Shoot(0, 0);
  }
}
```

قدم دوم: نوشتن کلاس مربوط به سلطونی که در نقطه (1,1) یک خدنگ ایجاد می‌کند.

حال می‌خواهیم سلطونی بنویسیم که در نقطه (1,1) یک خدنگ ایجاد کند منتها اگر پول کافی داشت در این نقطه یک قلعه بسازد درغیراینصورت یک تیرانداز بسازد. برای این کار سلطون باید از اطلاعاتی که پارامتر game دارد استفاده کند.

پارامتر game شامل تمام اطلاعات لازم در مورد زمین بازی است.

SoltoonOneOne.java

```java
import ir.pint.soltoon.soltoongame.shared.actions.Action;
import ir.pint.soltoon.soltoongame.shared.actions.AddKhadang;
import ir.pint.soltoon.soltoongame.shared.agents.Soltoon;
import ir.pint.soltoon.soltoongame.shared.map.Game;
import ir.pint.soltoon.soltoongame.shared.map.GameSoltoon;
import ir.pint.soltoon.soltoongame.shared.map.KhadangType;

public class SoltoonOneOne extends Soltoon {
  @Override
  public void init(Game g) {
    // do nothing
  }
  @Override
  public void lastThingsToDo(Game g) {
```

```
    // do nothing
  }
  @Override
  public Action getAction(Game g) {
    GameSoltoon me = g.getSoltoon(getId());
    if (me.getMoney() > KhadangType.CASTLE.getCost()) {
      return new AddKhadang(new ShootZeroZero(KhadangType.CASTLE), 0, 0);
    } else {
      return new AddKhadang(new ShootZeroZero(KhadangType.MUSKETEER), 0, 0);
    }
  }
}
```

کلاس Game

این کلاس اطلاعات مربوط به زمین بازی را به ما می‌دهد.

`int getMapHeight()`	عرض زمین بازی
`int getMapWidth()`	طول زمین بازی
`int getCurrentRound()`	دور
`Cell getCell(Cell center, Direction direction)`	دریافت یک خانه زمین بازی با توجه به یک خانه دیگر و جهت
`Cell getCell(Integer x, Integer y)`	دریافت یک خانه زمین بازی با x و y
`GameSoltoon getOwner(Long id)`	دریافت سلطون صاحب خدنگ
`GameKhadang getKhadang(Long id)`	دریافت خدنگ
`Map<Long, GameKhadang> getKhadangs()`	خدنگ‌ها براساس شماره‌شان
`GameSoltoon getSoltoon(Long id)`	دریافت سلطون ها
`Map<Long, GameSoltoon> getSoltoons()`	سلطون‌ها براساس شماره‌شان

- تغییر در این کلاس تغییری در بازی به‌وجود نمی‌آورد.

محدودیت زمانی اجرای متدهای getAction, init, lastThingsToDo

این متدها محدودیت زمان اجرا دارند. یعنی اگر تصمیم گیری شما از مدت زمانی بیشتر طول بکشد برنامه‌ی بازی اجرای متد را متوقف می‌کند. برای مدیریت زمان کلاس‌های Khadang و Soltoon متدهایی دارند:

دریافت زمان باقی‌مانده (میلی‌ثانیه)	`int getRemainingTime()`
خروجی موقتی این متد موقتا مقدار return متد را مشخص می‌کند که اگر زمان اجرای متد بیش از حد مجاز بود این آبجکت بعنوان خروجی استفاده شود.	`void returnTemporary(Object returnObject)`

متدهای کلاس Khadang و Soltoon

این متد شناسه خدنگ/سلطون را می‌دهد. بعنوان مثال اگر در متد getAction یک خدنگ این متد را صدا بزنید id خدنگ مورد نظر را برمی‌گرداند.	`Long getId()`

سناریوهای آموزشی

سناریوی اول: سلام دنیا!

هدف از این سناریو آشنایی شما با متدهای مربوط به زمین بازی است.

یک سلطون طراحی کنید که هنگام صدا زدن متد getAction اطلاعات زیر را به ازای هر خدنگ روی زمین بازی در کنسول بنویسد:

- شناسه خدنگ (id)
- شناسه سلطون صاحب خدنگ
- مکان خدنگ
- نوع خدنگ
- شناسه خدنگ های در محدوده شلیک این خدنگ

معیار عملکرد شما

- درستی توصیف شما از بازی

سناریوی دوم: نام

هدف از این سناریو آشنایی شما با چگونگی ایجاد خدنگ‌هاست.

در این سناریو شما برای ساختن خدنگ‌ها محدودیتی ندارید. وظیفه شما این است که با پر کردن خانه های بازی از خدنگ ها نام خودتان را روی صفحه بازی رسم کنید. برای این کار باید سلطونی طراحی کنید که این کار را انجام دهد.

در یک بازی عادی سلطون‌ها باید پول کافی برای ایجاد خدنگ‌ها را داشته باشند. همچنین خدنگ جدید نباید در محدوده شلیک باقی خدنگ‌های زمین باشد. در این سناریو این دو محدودیت وجود ندارد.

معیار عملکرد شما

- صحت و زیبایی کارتان

سناریوی سوم: جنگجو

هدف از این سناریو آشنایی شما با چگونگی حرکت و شلیک خدنگ‌هاست.
در شروع سناریو شما باید یک غول بسازید و منتظر باشید تا یک خدنگ بی‌دفاع حریف روی یک نقطه‌ی تصادفی زمین بازی ظاهر شود، بعد باید هرچه سریع‌تر به او نزدیک شوید و او را بکشید و همین اتفاق به طور پیوسته تکرار می‌شود. تضمین می‌شود که در هر لحظه حداکثر یک خدنگ حریف روی زمین است.

معیار عملکرد شما

- بهینه بودن حرکت‌هایتان

سناریوی چهارم: مدافع با بودجه‌ی اولیه‌ی معلوم

در شروع سناریو به شما مقدار مشخصی پول داده شده است. شما باید در ابتدای بازی تعدادی خدنگ ثابت در نقاط مختلف زمین ایجاد کنید. سپس حریفتان که پولش هر لحظه زیادتر می‌شود، با ساختن سرباز و حمله به خدنگ‌های شما سعی در نابود کردن تمامی آنها دارد و خدنگ‌های ثابت شما باید از خودشان دفاع کنند.

جزئیات بیشتر این سناریو در نسخه‌های بعدی متنشر خواهد شد.

معیار عملکرد شما

- معیار عملکرد شما مدت زمانی است که دوام می‌آورید.

سناریوی پنجم: مهاجم با بودجه‌ی رو به رشد و مواضع مشخص حریف

در شروع سناریو حریفتان تعدادی برجک در نقاط از پیش مشخص شده ایجاد می‌کند. شما باید با ساختن سربازان و حمله به او استحکاماتش را نابود کنید. گفتنی است که در سناریوهای مهاجم با گذشت زمان پولتان زیاد می‌شود.

معیار عملکرد شما

- معیار عملکرد شما سرعت‌تان در تخریب استحکامات حریف است.

سناریوی ششم: مدافع با بودجه‌ی اولیه‌ی نامعلوم

تفاوت این سناریو با سناریوی چهارم این است که مقدار اولیه‌ی پولتان معلوم نیست و برنامه‌ی شما به طور خودکار باید بودجه‌بندی کند و استحکامات بسازد و آماده‌ی دفاع شود.

معیار عملکرد شما

- معیار عملکرد شما مدت زمانی است که دوام می‌آورید.

سناریوی هفتم: مهاجم با بودجه‌ی رو به رشد و مواضع نامشخص حریف

تفاوت این سناریو با سناریوی پنجم این است که شما از قبل نمی‌دانید استحکامات حریفتان به چه شکل خواهد بود.

معیار عملکرد شما

- معیار عملکرد شما سرعت‌تان در تخریب استحکامات حریف است.

جنگ جهانی سلطون

- شما و حریفتان با بودجه‌ی اولیه‌ی مساوی شروع میکنید و پولتان با گذشت زمان و با نرخ ثابتی زیاد میشود.
- هرکدامتان میتوانید در لحظات مختلف خدنگ‌هایی ایجاد کنید و گاهی دفاع کنید و گاهی حمله.
- در صورتی که بازی با درگیری ناچیز خدنگ‌ها همراه باشد برای هر دو سلطون یک عملکرد بد تلقی می‌شود.

جزئیات بیشتر این سناریو در بزودی مشخص خواهد شد.

معیار عملکرد شما

- معیار عملکرد شما در پایان بازی، امتیازتان است.

سوالات متداول

چطور می‌توانم در کلاس یک خدنگ جای آن خدنگ در زمین را پیدا کنم ؟

```
public class MyKhadang extends Khadang {

  // ...
  @Override
  public Action getAction(Game g) {
    GameKhadang me = g.getKhadang(getId());
    Cell myCell = me.getCell(); // <---- My cell

    return null;
  }
}
```

چطور می‌توانم مقدار پول سلطونم را بدانم ؟

```
public class MySoltoon extends Soltoon {
  // ...

  @Override
  public Action getAction(Game g) {
    GameSoltoon me = g.getSoltoon(getId());
    int money = me.getMoney(); // <---- My Money
    return null;
  }
}
```

چطور می‌توانم هزینه ساخت یک خدنگ را پیدا کنم ؟

```
public class MySoltoon extends Soltoon {
  // ...
  @Override
  public Action getAction(Game g) {
    Integer giantCost = KhadangType.GIANT.getCost(); // <--- Giant cost
```

```
    Integer castleCost = KhadangType.CASTLE.getCost(); // <--- Castle cost
    return null;
  }
}
```

چطور میتونم بفهمم در یک خانه از جدول چه نوع خدنگی وجود دارد ؟

```
public class MySoltoon extends Soltoon {
  // ...
  @Override
  public Action getAction(Game g) {
    Cell cell = g.getCell(2, 5);
    if (cell.hasKhadang()){
      GameKhadang khadang = cell.getKhadang();
      KhadangType type = khadang.getType(); // <--- Khadang Type
    }
    return null;
  }
}
```