

ИНСТРУМЕНТИ ЗА АНАЛИЗ НА ПРОГРАМЕН КОД

Research described in this tutorial was partially supported by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)", financed by the Ministry of Education and Science.

Developed by Velbazhd Software LLC

В професията на софтуерните инженери има множество трудности, но една от най-значимите е свързана с навлизането в нов проект. С тази трудност се сблъскват както специалистите с опит, така и младите специалисти.



На практика, очакванията са специалистите с опит да навлязат по-бързо в един нов проект отколкото това могат да сторят младите специалисти. Реалността е, че и за двете групи един нов проект в началото е нещо непознат с което трябва да се сблъскат.



Съществуват множество различни инструменти, които могат да спомогнат за изучаването на кода при навлизането в нов проект. За демонстрация на някои от тях ще бъде използван проектът JSkat:





Search or jump to...

Pull requests Issues Marketplace Explore



b0n541 / jskat-multimodule

Watch

1

Unstar

10

Fork

9

Code

Issues 22

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

JSkat is a free software implementation of the game Skat in Java. <http://www.jskat.org>

1,708 commits

7 branches

0 packages

10 releases

7 contributors

Branch: develop

New pull request

Create new file

Upload files

Find file

Clone or download

b0n541 Merge pull request #73 from b0n541/dependabot/gradle/org.reflections-...

Latest commit fee7b07 23 days ago

gradle/wrapper	Update Gradle wrapper	29 days ago
info	new splash screen	5 years ago
jskat-base	Bump reflections from 0.9.11 to 0.9.12	23 days ago
jskat-javafx-gui	Bump junit from 4.12 to 4.13	29 days ago
jskat-swing-gui	Fix license header	5 months ago
.gitignore	Add IntelliJ's out dir to .gitignore	13 months ago
.travis.yml	Make JSkat compatible with JDK 11	14 months ago
README.md	Update Gradle tasks	5 months ago
build.gradle	Move on towards version 0.18.0	5 months ago
gradlew	Update Gradle wrapper	29 days ago
gradlew.bat	Update Gradle wrapper	29 days ago
settings.gradle	issue #38: first step towards Gradle builds	4 years ago

README.md

This is the multi-module of JSkat and JSkat on Android.



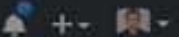
За да се работи по кода на проекта първо се прави разклонение на хранилището (fork):





Search or jump to...

Pull requests Issues Marketplace Explore



b0n541 / jskat-multimodule

Watch

1

★ Unstar

10

Fork

9

<> Code

Issues 22

Pull requests

JSkat is a free software implementation of the game Skat.

1,708 commits

Branch: develop

New pull request

b0n541 Merge pull request #73 from b0n541

gradle/wrapper

info

jskat-base

jskat-javafx-gui

jskat-swing-gui

.gitignore

.travis.yml

README.md

build.gradle

gradlew

gradlew.bat

settings.gradle

Fork jskat-multimodule

×

Where should we fork jskat-multimodule?



TodorBalabanov



Coding-Sunday-Sofia



Will be created as Coding-Sunday-Sofia/jskat-multimodule.



VelbazhdSoftwareLLC



Academy-Soft-Intellect

Insights

7 contributors

Find file

Clone or download

Latest commit fee7b07 23 days ago

29 days ago

5 years ago

23 days ago

29 days ago

5 months ago

13 months ago

14 months ago

5 months ago

5 months ago

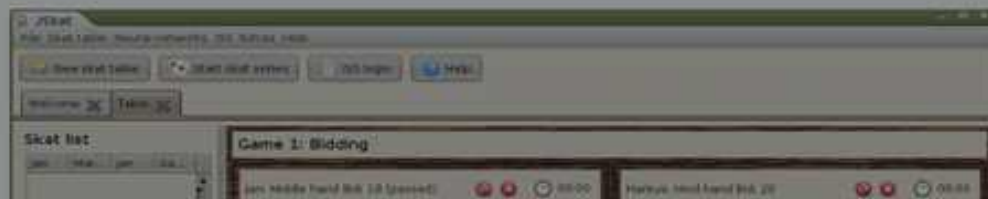
29 days ago

29 days ago

4 years ago

README.md

This is the multi-module of JSkat and JSkat on Android.





Search or jump to...



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



[Coding-Sunday-Sofia](#) / [jskat-multimodule](#)

forked from [b0n541/jskat-multimodule](#)

Watch ▾

0

★ Star

0

Fork

10

Code

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

Forking b0n541/jskat-multimodule

It should only take a few seconds.

Refresh



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

Coding-Sunday-Sofia / jskat-multimodule

forked from b0n541/jskat-multimodule

Watch 0

Star 0

Fork 10

<> Code

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

JSkat is a free software implementation of the game Skat in Java. <http://www.jskat.org>

[Edit](#)[Manage topics](#)

1,708 commits

7 branches

0 packages

10 releases

7 contributors

Branch: develop ▾

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download ▾](#)

This branch is even with b0n541:develop.

[Pull request](#)

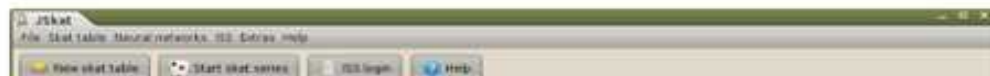
[Compare](#)

b0n541	Merge pull request b0n541#73 from b0n541/dependabot/gradle/org.reflec...	Latest commit fee7b07 23 days ago
gradle/wrapper	Update Gradle wrapper	29 days ago
info	new splash screen	5 years ago
jskat-base	Bump reflections from 0.9.11 to 0.9.12	23 days ago
jskat-javafx-gui	Bump junit from 4.12 to 4.13	29 days ago
jskat-swing-gui	Fix license header	5 months ago
.gitignore	Add IntelliJ's out dir to .gitignore	13 months ago
.travis.yml	Make JSkat compatible with JDK 11	14 months ago
README.md	Update Gradle tasks	5 months ago
build.gradle	Move on towards version 0.18.0	5 months ago
gradlew	Update Gradle wrapper	29 days ago
gradlew.bat	Update Gradle wrapper	29 days ago
settings.gradle	issue b0n541#38: first step towards Gradle builds	4 years ago

README.md



This is the multi-module of JSkat and JSkat on Android.



Code Scene е един от свободно достъпните инструменти, които могат да се използват при първоначалния анализ на кода в проекта:





Welcome to the



Watch the movie

CodeScene Cloud

The powerful visualization tool using Predictive Analytics to find hidden risks and social patterns in your code.

Start today. No installation. Just log in with your Github.

(To run CodeScene in your datacenter or on your laptop, check out CodeScene **On-Prem.**)

Log in with
GitHub

CodeScene
On-premise

Demo
samples

First time user? Read "[How It Works](#)" or send us an [email](#).

Click to configure



-25 percent monthly output decrease

Last month

Age Group	Percentage
18-24	10%
25-34	15%
35-44	20%
45-54	25%
55-64	30%
65-74	35%
75-84	40%
85+	45%

■■■■■■■■■■

78%

Red Hotspots

За да бъде направен анализ на проекта първо той трябва да бъде добавен в списъка за анализиране:



Select Repositories

Select the repositories you want to analyze in your project.

Want to analyze private repositories? [Upgrade your payment plan.](#)

❗ Your current plan has a limit of max 10 repositories.

Filter visible repositories by regular expression

Coding-Sunday-Sofia/jskat-r



Filter

Select at least one repository

☐ Coding-Sunday-Sofia/jskat-multimodule



Select at least one repository

Select Repositories

Select the repositories you want to analyze in your project.

Want to analyze private repositories? [Upgrade your payment plan.](#)

❗ Your current plan has a limit of max 10 repositories.

Filter visible repositories by regular expression

Coding-Sunday-Sofia/jskat-multimodule



Filter

Continue

☒ Coding-Sunday-Sofia/jskat-multimodule



Continue

Create New Project

Project Name:

Projects names can consist of alpha-numeric characters, dashes, underscores, and dots. They must be 1-100 characters long.

Project Repositories:



Project Successfully Created!

You are ready to go!

[Run Initial Analysis](#)

Първоначалният анализ показва интересните области с потенциал за подобрене в проекта:



ANALYSIS RESULTS
JSKAT-MULTIMODULE

- Dashboard
- Scope
- Technical Debt
- Architecture
- Social Analyses
- Project Management

Hotspot Health

	Status now	Last month
JSkatViewImpl.java jskat-multimodule/jskat-swing-gui/src/.../jskat/gui/swing		
SkatGame.java jskat-multimodule/jskat-base/src/main/java/org/jskat/control		
IssController.java jskat-multimodule/jskat-base/src/.../jskat/control/iss		
JSkatOptions.java jskat-multimodule/jskat-base/src/main/java/org/jskat/data		
SkatTablePanel.java jskat-multimodule/jskat-swing-gui/src/.../swing/table		

More Hotspots →
Interactive Hotspot Map →

Interactive
Hotspots Map

13%

Red Hotspots

29%

Development Effort in Red
Hotspots

53%

Of Estimated Bugfixes in Red
Hotspots

Hotspots →

Hotspots Code
Health

Status Now



Branch Delivery Risk

This project does not have multiple branches, or the branch analysis is disabled.

Интерактивна карта на проекта маркира
горещите точки и потенциалните места
изискващи подобрения:



ANALYSIS RESULTS
JSKAT-MULTIMODULEHotspots identify the modules with most development activity -- often technical debt. [+](#)

Hotspots

Refactoring
Targets

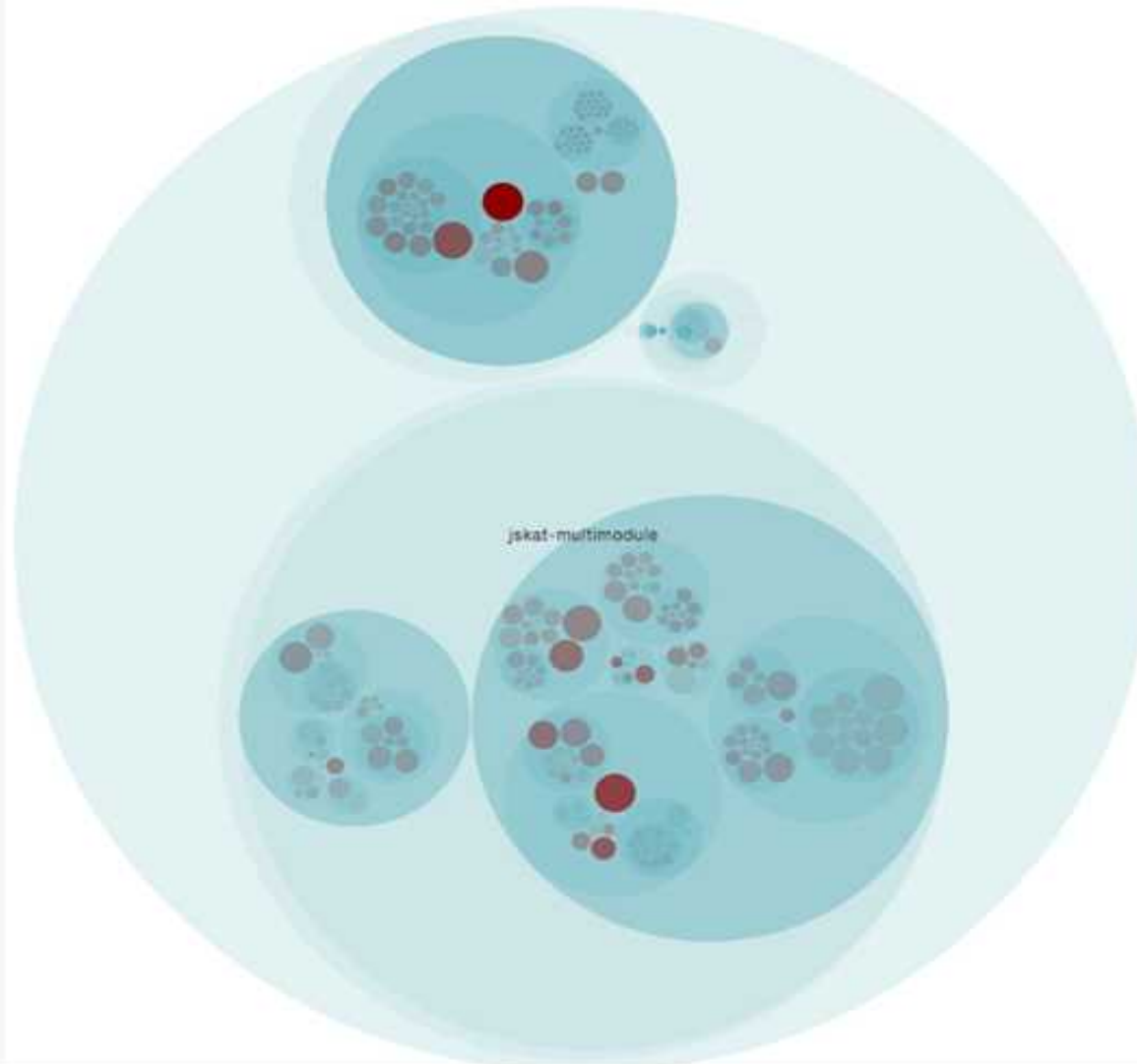
Costs

Defects

Programming
Language

System

jskat-multimodule



SkatGame.java се откроява със значителна сложност спрямо останалите файлове в проекта:



ANALYSIS RESULTS
JSKAT-MULTIMODULEHotspots identify the modules with most development activity -- often technical debt. [+](#)

Hotspots

Refactoring
Targets

Costs

Defects

Programming
Language

Dashboard

Scope

Technical Debt

Hotspots

Hotspot Code Health

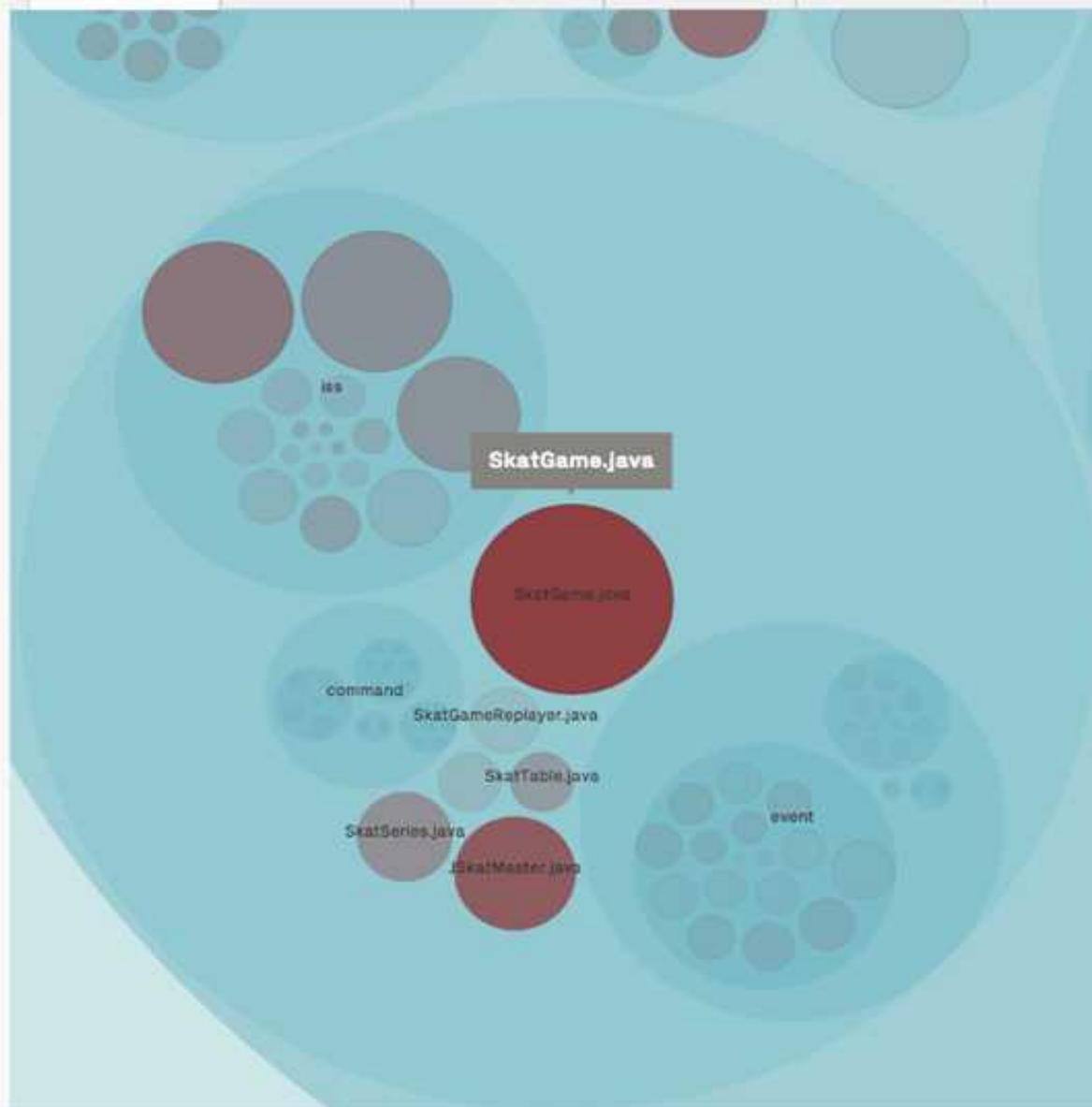
Refactoring Targets

Temporal Coupling

Architecture

Social Analyses

Project Management



System / jskat-multimodule
/ jskat-base / src / main
/ java / org / jskat / control
/ SkatGame.java

Size 768 Lines of
Code

Code
Health
6

Change
Frequency 158 Commits

Main
Author b0n541 (83 %)

Knowledge
Loss 0 %
Abandoned
Code

Defect
Commits 30 (18 % of
Total
Commits)

Last
Modified 11 months ago

Actions

View Code

X-Ray

Trends

Authors ▾

Complexity Trend



SkatGame.java е един от основните файлове
препоръчан за преработка:





Refactoring Targets



Prioritize improvements to these files since they have the highest technical debt interest rate.

[Dashboard](#)[Scope](#)[Technical Debt](#)[Hotspots](#)[Hotspot Code Health](#)[Refactoring Targets](#)[Temporal Coupling](#)[Architecture](#)[Social Analyses](#)[Project Management](#)[jskat-multimodule/jskat-swing-gui/src/main/java/org/jskat/gui/swing/JSkatViewImpl.java](#)

X-Ray

[jskat-multimodule/jskat-base/src/main/java/org/jskat/control/SkatGame.java](#)

X-Ray

[jskat-multimodule/jskat-swing-gui/src/main/java/org/jskat/gui/swing/table/SkatTablePanel.java](#)

X-Ray

[jskat-multimodule/jskat-base/src/main/java/org/jskat/data/SkatGameData.java](#)

X-Ray

[jskat-multimodule/jskat-base/src/main/java/org/jskat/control/iss/IssController.java](#)

X-Ray

[jskat-multimodule/jskat-base/src/main/java/org/jskat/data/JSkatOptions.java](#)

X-Ray

[jskat-multimodule/jskat-swing-gui/src/main/java/org/jskat/gui/swing/JSkatOptionsDialog.java](#)

X-Ray

[jskat-multimodule/jskat-base/src/main/java/org/jskat/util/CardList.java](#)

X-Ray

[jskat-multimodule/jskat-base/src/main/java/org/jskat/ai/mjl/OpponentPlayer.java](#)

X-Ray

[jskat-multimodule/jskat-base/src/main/java/org/jskat/ai/algorithmic/AlgorithmicOpponentPlayer.java](#)

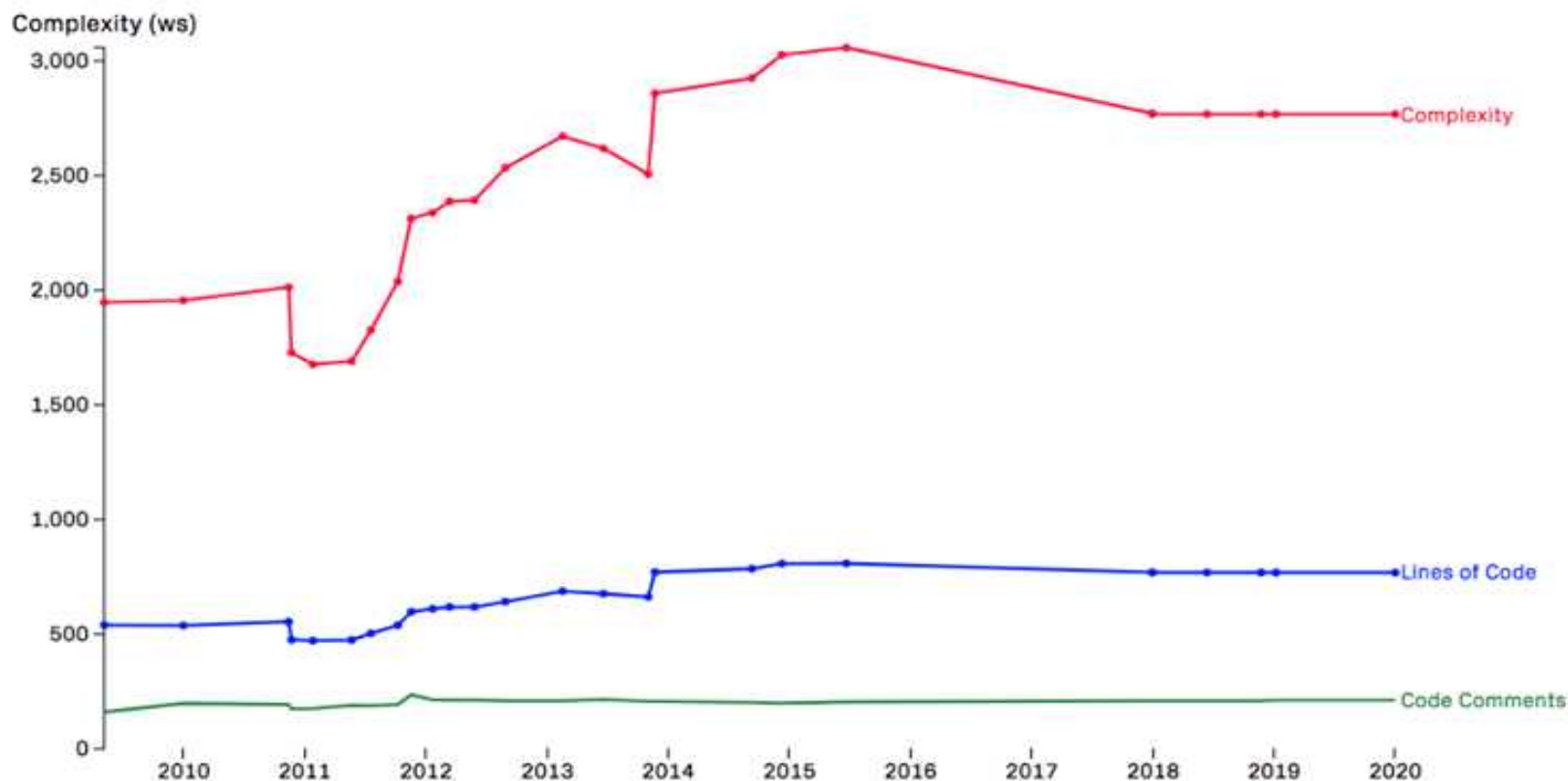
X-Ray

Съществен индикатор е сложността на кода
съпоставена спрямо броя редове във файла:



ANALYSIS RESULTS
JSKAT-MULTIMODULE

Complexity Trend

[Hotspots](#) / [jskat-multimodule/jskat-base/src/main/java/org/jskat/control/SkatGame.java](#) / Complexity Trend

Descriptive Statistics

Complexity vs. Lines of Code

] tio

При допълнителен анализ на сложността в отделните методи се открояват два подходящи кандидата за преработка playTricks и run:



ANALYSIS RESULTS
JSKAT-MULTIMODULE

X-Ray File Results

[Projects](#) / [jskat-multimodule](#) / [Files](#) / [jskat-multimodule/jskat-base/src/main/java/org/jskat/control/SkatGame.java](#)

Hotspots

[Internal Temporal Coupling](#)[Structural Recommendations](#)

Function

Change Frequency

Complexity/Size

Cyclomatic Complexity

playTricks

49

106

20

run

40

126

26

playCard

33

59

12

bidding

26

74

7

discarding

26

35

3

SkatGame

18

21

2

dealCards

17

54

6

setGameAnnouncement

16

15

2

calculateGameValue

14

22

2

twoPlayerBidding

13

60

6

setGameState

12

17

3

contraRe

11

11

3

Сложността в метода `run` може да бъде намалена чрез раздробяване на кода в няколко помощни функции:



run

```
public SkatGameResult run() {
    // FIXME jan 11.07.2013: this method is too long, break it down to smaller
    // methods or implement it in another way
    view.setGameState(tableName, data.getGameState());

    do {
        log.debug("Game state: " + data.getGameState());

        switch (data.getGameState()) {
            case GAME_START:
                setGameState(GameState.DEALING);
                break;
            case DEALING:
                dealCards();
                setGameState(GameState.BIDDING);
                break;
            case BIDDING:
                setActivePlayer(Player.MIDDLEHAND);

                if (variant == GameVariant.FORCED_RAMSCH) {
                    // ramsch games are enforced
                    final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
                    gaf.setGameType(GameType.RAMSCH);
                    setGameAnnouncement(gaf.getAnnouncement());
                } else {
                    // "normal" game (i.e. no ramsch)
                    bidding();
                }

                if (GameType.PASSED_IN.equals(data.getGameType())) {
                    setGameState(GameState.PRELIMINARY_GAME_END);
                } else if (GameType.RAMSCH.equals(data.getGameType())) {
                    setGameState(GameState.RAMSCH_GRAND_HAND_ANNOUNCING);
                } else {
                    view.setDeclarer(tableName, data.getDeclarer());
                    setGameState(GameState.PICKING_UP_SKAT);
                }
                break;
            case RAMSCH_GRAND_HAND_ANNOUNCING:
                final boolean grandHandAnnounced = grandHand();

                if (grandHandAnnounced) {
                    log.debug(data.getDeclarer() + " is playing grand hand"); //NON-NLS-1$
                    final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
                    gaf.setGameType(GameType.GRAND);
                    gaf.setHand(Boolean.TRUE);
                }
            }
        }
    } while (true);
}
```


Собственото хранилище на разклонения проект трябва да се клонира на локалната машина, така че проектът да се работи локално:



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

Coding-Sunday-Sofia / jskat-multimodule

forked from b0n541/jskat-multimodule

Watch 0

Star 0

Fork 10

<> Code

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

JSkat is a free software implementation of the game Skat in Java. <http://www.jskat.org>

[Edit](#)[Manage topics](#)

1,708 commits

7 branches

0 packages

10 releases

7 contributors

Branch: develop

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download](#)

This branch is even with b0n541:develop.

b0n541 Merge pull request b0n541#73 from b0n541/dependabot/gradle/org.reflec...

gradle/wrapper	Update Gradle wrapper	
info	new splash screen	
jskat-base	Bump reflections from 0.9.11 to 0.9.12	23 days ago
jskat-javafx-gui	Bump junit from 4.12 to 4.13	29 days ago
jskat-swing-gui	Fix license header	5 months ago
.gitignore	Add IntelliJ's out dir to .gitignore	13 months ago
.travis.yml	Make JSkat compatible with JDK 11	14 months ago
README.md	Update Gradle tasks	5 months ago
build.gradle	Move on towards version 0.18.0	5 months ago
gradlew	Update Gradle wrapper	29 days ago
gradlew.bat	Update Gradle wrapper	29 days ago
settings.gradle	issue b0n541#38: first step towards Gradle builds	4 years ago

Clone with HTTPS

[Use SSH](#)

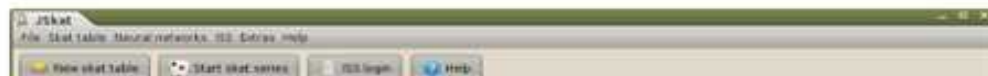
Use Git or checkout with SVN using the web URL.

<https://github.com/Coding-Sunday-Sofia/jskat-multimodule>

[Open in Desktop](#)[Download ZIP](#)

README.md

This is the multi-module of JSkat and JSkat on Android.



На локалния компютър първоначално има само един клон (branch) на проекта:




```
MACMINI:Desktop todorbalabanov$ git clone https://github.com/Coding-Sunday-Sofia/jskat-multimodule.git
Cloning into 'jskat-multimodule'...
remote: Enumerating objects: 742, done.
remote: Counting objects: 100% (742/742), done.
remote: Compressing objects: 100% (519/519), done.
remote: Total 33766 (delta 377), reused 370 (delta 156), pack-reused 33024
Receiving objects: 100% (33766/33766), 81.21 MiB | 3.25 MiB/s, done.
Resolving deltas: 100% (18734/18734), done.
MACMINI:Desktop todorbalabanov$ cd jskat-multimodule/
MACMINI:jskat-multimodule todorbalabanov$ git branch
* develop
MACMINI:jskat-multimodule todorbalabanov$
```


В проекта JSkat има установени правила за работа с Git и разклоненията в проекта, но поради учебният характер на настоящото изложение тези правила няма да бъдат спазени и ще бъде създаден клон извън заложената в проекта схема:




```
MACMINI:jskat-multimodule todorbalabanov$ git checkout -b small-fix/run-method-split develop
Switched to a new branch 'small-fix/run-method-split'
MACMINI:jskat-multimodule todorbalabanov$ git branch
  develop
* small-fix/run-method-split
MACMINI:jskat-multimodule todorbalabanov$
```


Проектът използва Gradle скрипт за изграждане и преди да бъде променян кодът е добре проектът да бъде изграден:



MACMINI:jsskat-multimodule todorbalabanov\$./gradlew clean assembleDist
Starting a Gradle Daemon (subsequent builds will be faster)

> Task :jsskat-base:compileJava

Note: Some input files use or override a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

> Task :jsskat-swing-gui:compileJava

Note: Some input files use or override a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

Note: Some input files use unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.

BUILD SUCCESSFUL in 1m 39s

17 actionable tasks: 13 executed, 4 up-to-date

MACMINI:jsskat-multimodule todorbalabanov\$ █

Проектът е организиран да поддържа множество платформи, но в случая ще бъде стартиран Swing вариантът:



Play on local table

START SKAT SERIES

Play on ISS

Replay game

Welcome



Welcome to JSkat



Play on ISS

Games against human players and the best AI players.
An internet connection is needed.

Play on local table

Games against local AI players.
No internet connection is needed.

Preferences

Set program options and skat rules.



Exit JSkat

Close JSkat

```

MACMINI:jskat-multimodule todorbalabanov$ java -jar bui
ld/libs/jskat-multimodule-0.18.0-SNAPSHOT.jar
5948 [Thread-4] INFO  org.reflections.Reflections
    (239 ) - Reflections took 1525 ms to scan 1 urls, pr
oducing 2252 keys and 9361 values
5993 [Thread-4] WARN  org.reflections.Reflections
    (318 ) - could not get type for name javax.servlet.F
ilter from any class loader
org.reflections.ReflectionsException: could not get typ
e for name javax.servlet.Filter
    at org.reflections.ReflectionUtils.forName(Refle
ctionUtils.java:312)
    at org.reflections.Reflections.expandSuperTypes
(Reflections.java:382)
    at org.reflections.Reflections.<init>(Reflectio
ns.java:140)
    at org.jskat.player.JSkatPlayerResolver.getAllI
mplementations(JSkatPlayerResolver.java:80)
    at org.jskat.player.JSkatPlayerResolver.getAllA
IPlayerImplementations(JSkatPlayerResolver.java:66)
    at org.jskat.gui.swing.table.SkatSeriesStartDia
log.initGUI(SkatSeriesStartDialog.java:116)
    at org.jskat.gui.swing.table.SkatSeriesStartDia
log.<init>(SkatSeriesStartDialog.java:101)
    at org.jskat.gui.swing.JSkatViewImpl.<init>(JSk
atViewImpl.java:198)
    at org.jskat.JSkatFX$1.call(JSkatFX.java:111)
    at org.jskat.JSkatFX$1.call(JSkatFX.java:94)
    at javafx.concurrent.Task$TaskCallable.call(Tas
k.java:1425)
    at java.base/java.util.concurrent.FutureTask.ru
n(FutureTask.java:264)
    at java.base/java.lang.Thread.run(Thread.java:8
34)
Caused by: java.lang.ClassNotFoundException: javax.serv
let.Filter
    at java.base/jdk.internal.loader.BuiltinClassLo
ader.loadClass(BuiltinClassLoader.java:583)
    at java.base/jdk.internal.loader.ClassLoaders$A
ppClassLoader.loadClass(ClassLoaders.java:178)
    at java.base/java.lang.ClassLoader.loadClass(Cl
assLoader.java:521)
    at org.reflections.ReflectionUtils.forName(Refle
ctionUtils.java:310)
    ... 12 common frames omitted
6007 [Thread-4] WARN  org.reflections.Reflections

```


Най-удачно подобряване на кода в този случай е чрез група помощни методи:




```

    // inform all players about the starting of the new game
    for (final Player pos : player.keySet()) {
        getPlayerInstance(pos).newGame(pos);
    }

    setGameState(GameState.GAME_START);
}

/**
 * Runs the skat game.
 *
 * @return Result of the skat game
 */
public SkatGameResult run() {
    // FIXME jan 11.07.2013: this method is too long, break it down to smaller
    // methods or implement it in another way
    view.setGameState(tableName, data.getGameState());

    do {
        log.debug("Game state: " + data.getGameState());

        switch (data.getGameState()) {
            case GAME_START:
                setGameState(GameState.DEALING);
                break;
            case DEALING:
                dealCards();
                setGameState(GameState.BIDDING);
                break;
            case BIDDING:
                setActivePlayer(Player.MIDDLEHAND);

                if (variant == GameVariant.FORCED_RAMSCH) {
                    // ramsch games are enforced
                    final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
                    gaf.setGameType(GameType.RAMSCH);
                    setGameAnnouncement(gaf.getAnnouncement());
                } else {
                    // "normal" game (i.e. no ramsch)
                    bidding();
                }

                if (GameType.PASSED_IN.equals(data.getGameType())) {
                    setGameState(GameState.PRELIMINARY_GAME_END);
                } else if (GameType.RAMSCH.equals(data.getGameType())) {

```


От case конструкцията първо се извежда в собствен метод процедурата по наддаване:




```

switch (data.getState()) {
case GAME_START:
    setGameState(GameState.DEALING);
    break;
case DEALING:
    dealCards();
    setGameState(GameState.BIDDING);
    break;
case BIDDING:
    setActivePlayer(Player.MIDDLEHAND);

    if (variant == GameVariant.FORCED_RAMSCH) {
        // ramsch games are enforced
        final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
        gaf.setGameType(GameType.RAMSCH);
        setGameAnnouncement(gaf.getAnnouncement());
    } else {
        // "normal" game (i.e. no ramsch)
        bidding();
    }

    if (GameType.PASSED_IN.equals(data.getGameType())) {
        setGameState(GameState.PRELIMINARY_GAME_END);
    } else if (GameType.RAMSCH.equals(data.getGameType())) {
        setGameState(GameState.RAMSCH_GRAND_HAND_ANNOUNCING);
    } else {
        view.setDeclarer(tableName, data.getDeclarer());
        setGameState(GameState.PICKING_UP_SKAT);
    }
    break;
case RAMSCH_GRAND_HAND_ANNOUNCING:
    final boolean grandHandAnnounced = grandHand();

    if (grandHandAnnounced) {
        log.debug(data.getDeclarer() + " is playing grand hand"); //$NON-NLS-1$
        final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
        gaf.setGameType(GameType.GRAND);
        gaf.setHand(Boolean.TRUE);
        setGameAnnouncement(gaf.getAnnouncement());
        setGameState(GameState.TRICK_PLAYING);
        log.debug("grand hand game started"); //$NON-NLS-1$
        break;
    } else {
        if (JJSkatOptions.instance().isSchieberamsch(true)) {
            log.debug("no grand hand - initiating schieberamsch"); //$NON-NLS-1$
            setGameState(GameState.SCHIEBERAMSCH);
        }
    }
}

```


Помощната функция е с private обхват на
видимост:




```

log.debug(data.getGameState().name());
log.debug("Game moves:");
for (final SkatGameEvent event : data.getGameMoves()) {
    log.debug(event.toString());
}

return getResult();
}

```

```

private void runBidding() {
    setActivePlayer(Player.MIDDLEHAND);

    if (variant == GameVariant.FORCED_RAMSCH) {
        // ramsch games are enforced
        final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
        gaf.setGameType(GameType.RAMSCH);
        setGameAnnouncement(gaf.getAnnouncement());
    } else {
        // "normal" game (i.e. no ramsch)
        bidding();
    }

    if (GameType.PASSED_IN.equals(data.getGameType())) {
        setGameState(GameState.PRELIMINARY_GAME_END);
    } else if (GameType.RAMSCH.equals(data.getGameType())) {
        setGameState(GameState.RAMSCH_GRAND_HAND_ANNOUNCING);
    } else {
        view.setDeclarer(tableName, data.getDeclarer());
        setGameState(GameState.PICKING_UP_SKAT);
    }
}

```

```

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
        setActivePlayer(activePlayerBeforeContraRe);
    }
}

```

```

private boolean grandHand() {
    // ...
}

```


Което позволява безопасно извикване от публичната функция:




```

switch (data.getState()) {
case GAME_START:
    setGameState(GameState.DEALING);
    break;
case DEALING:
    dealCards();
    setGameState(GameState.BIDDING);
    break;
case BIDDING:
    runBidding();
    break;
case RAMSCH_GRAND_HAND_ANNOUNCING:
    final boolean grandHandAnnounced = grandHand();

    if (grandHandAnnounced) {
        log.debug(data.getDeclarer() + " is playing grand hand"); //$NON-NLS-1$
        final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
        gaf.setGameType(GameType.GRAND);
        gaf.setHand(Boolean.TRUE);
        setGameAnnouncement(gaf.getAnnouncement());
        setGameState(GameState.TRICK_PLAYING);
        log.debug("grand hand game started"); //$NON-NLS-1$
        break;
    } else {
        if (JSkatOptions.instance().isSchieberamsch(true)) {
            log.debug("no grand hand - initiating schieberamsch"); //$NON-NLS-1$
            setGameState(GameState.SCHIEBERAMSCH);
        } else {
            log.debug("no grand hand and no schieberamsch - play ramsch"); //$NON-NLS-1$
            setGameState(GameState.TRICK_PLAYING);
        }
    }
    break;
case SCHIEBERAMSCH:
    schieberamsch();
    final GameAnnouncementFactory factory = GameAnnouncement.getFactory();
    factory.setGameType(GameType.RAMSCH);
    setGameAnnouncement(factory.getAnnouncement());
    setGameState(GameState.TRICK_PLAYING);
    break;
case PICKING_UP_SKAT:
    setActivePlayer(data.getDeclarer());
    if (pickUpSkat()) {
        setGameState(GameState.DISCARDING);
    } else {
        setGameState(GameState.DECLARING);
    }
}

```


След това се извежда фрагмента код за анонсиране на голяма ръка:




```

switch (data.getState()) {
case GAME_START:
    setGameState(GameState.DEALING);
    break;
case DEALING:
    dealCards();
    setGameState(GameState.BIDDING);
    break;
case BIDDING:
    runBidding();
    break;
case RAMSCH_GRAND_HAND_ANNOUNCING:
    final boolean grandHandAnnounced = grandHand();

    if (grandHandAnnounced) {
        log.debug(data.getDeclarer() + " is playing grand hand"); //$NON-NLS-1$
        final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
        gaf.setGameType(GameType.GRAND);
        gaf.setHand(Boolean.TRUE);
        setGameAnnouncement(gaf.getAnnouncement());
        setGameState(GameState.TRICK_PLAYING);
        log.debug("grand hand game started"); //$NON-NLS-1$
        break;
    } else {
        if (JSkatOptions.instance().isSchieberamsch(true)) {
            log.debug("no grand hand - initiating schieberamsch"); //$NON-NLS-1$
            setGameState(GameState.SCHIEBERAMSCH);
        } else {
            log.debug("no grand hand and no schieberamsch - play ramsch"); //$NON-NLS-1$
            setGameState(GameState.TRICK_PLAYING);
        }
    }
    break;
case SCHIEBERAMSCH:
    schieberamsch();
    final GameAnnouncementFactory factory = GameAnnouncement.getFactory();
    factory.setGameType(GameType.RAMSCH);
    setGameAnnouncement(factory.getAnnouncement());
    setGameState(GameState.TRICK_PLAYING);
    break;
case PICKING_UP_SKAT:
    setActivePlayer(data.getDeclarer());
    if (pickUpSkat()) {
        setGameState(GameState.DISCARDING);
    } else {
        setGameState(GameState.DECLARING);
    }
}

```



```

        bidding();
    }

    if (GameType.PASSED_IN.equals(data.getGameType())) {
        setGameState(GameState.PRELIMINARY_GAME_END);
    } else if (GameType.RAMSCH.equals(data.getGameType())) {
        setGameState(GameState.RAMSCH_GRAND_HAND_ANNOUNCING);
    } else {
        view.setDeclarer(tableName, data.getDeclarer());
        setGameState(GameState.PICKING_UP_SKAT);
    }
}

```

```

private void runRamschGrandHandAnnouncing(){
    final boolean grandHandAnnounced = grandHand();

    if (grandHandAnnounced) {
        log.debug(data.getDeclarer() + " is playing grand hand"); //$NON-NLS-1$
        final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
        gaf.setGameType(GameType.GRAND);
        gaf.setHand(Boolean.TRUE);
        setGameAnnouncement(gaf.getAnnouncement());
        setGameState(GameState.TRICK_PLAYING);
        log.debug("grand hand game started"); //$NON-NLS-1$
        break;
    } else {
        if (JSkatOptions.instance().isSchieberamsch(true)) {
            log.debug("no grand hand - initiating schieberamsch"); //$NON-NLS-1$
            setGameState(GameState.SCHIEBERAMSCH);
        } else {
            log.debug("no grand hand and no schieberamsch - play ramsch"); //$NON-NLS-1$
            setGameState(GameState.TRICK_PLAYING);
        }
    }
}
}

```

```

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
        setActivePlayer(activePlayerBeforeContraRe);
    }
}

```



```

switch (data.getGameState()) {
    case GAME_START:
        setGameState(GameState.DEALING);
        break;
    case DEALING:
        dealCards();
        setGameState(GameState.BIDDING);
        break;
    case BIDDING:
        runBidding();
        break;
    case RAMSCH_GRAND_HAND_ANNOUNCING:
        runRamschGrandHandAnnouncing();
        break;
    case SCHIEBERAMSCH:
        schieberamsch();
        final GameAnnouncementFactory factory = GameAnnouncement.getFactory();
        factory.setGameType(GameType.RAMSCH);
        setGameAnnouncement(factory.getAnnouncement());
        setGameState(GameState.TRICK_PLAYING);
        break;
    case PICKING_UP_SKAT:
        setActivePlayer(data.getDeclarer());
        if (pickUpSkat()) {
            setGameState(GameState.DISCARDING);
        } else {
            setGameState(GameState.DECLARING);
        }
        break;
    case DISCARDING:
        setActivePlayer(data.getDeclarer());
        discarding();
        if (!GameState.PRELIMINARY_GAME_END.equals(data.getGameState())) {
            setGameState(GameState.DECLARING);
        }
        break;
    case DECLARING:
        announceGame();
        if (isContraPlayEnabled(ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
            setGameState(GameState.CONTRA);
        } else {
            setGameState(GameState.TRICK_PLAYING);
        }
        break;
    case CONTRA:
        for (final Player player : Player.getOrderedList()) {
            if (!isContraEnabledForPlayer(player, ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {

```


При този case изнасянето на блока код в отделен помощен метод не може да стане механично, тъй като case-а съдържа break инструкцията, която управлява do-while цикъла:




```
do {
    log.debug("Game state: " + data.getGameState());

    switch (data.getGameState()) {
        case GAME_START:
            setGameState(GameState.DEALING);
            break;
        case DEALING:
            dealCards();
            setGameState(GameState.BIDDING);
            break;
        case BIDDING:
            runBidding();
            break;
        case RAMSCH_GRAND_HAND_ANNOUNCING:
            if (runRamschGrandHandAnnouncing() == true) {
                break;
            }
            break;
        case SCHIEBERAMSCH:
            schieberamsch();
            final GameAnnouncementFactory factory = GameAnnouncement.getFactory();
            factory.setGameType(GameType.RAMSCH);
            setGameAnnouncement(factory.getAnnouncement());
            setGameState(GameState.TRICK_PLAYING);
            break;
        case PICKING_UP_SKAT:
            setActivePlayer(data.getDeclarer());
            if (pickUpSkat()) {
                setGameState(GameState.DISCARDING);
            } else {
                setGameState(GameState.DECLARING);
            }
            break;
        case DISCARDING:
            setActivePlayer(data.getDeclarer());
            discarding();
            if (!GameState.PRELIMINARY_GAME_END.equals(data.getGameState())) {
                setGameState(GameState.DECLARING);
            }
            break;
        case DECLARING:
            announceGame();
            if (isContraPlayEnabled(ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
                setGameState(GameState.CONTRA);
            } else {
```



```

        bidding();
    }

    if (GameType.PASSED_IN.equals(data.getGameType())) {
        setGameState(GameState.PRELIMINARY_GAME_END);
    } else if (GameType.RAMSCH.equals(data.getGameType())) {
        setGameState(GameState.RAMSCH_GRAND_HAND_ANNOUNCING);
    } else {
        view.setDeclarer(tableName, data.getDeclarer());
        setGameState(GameState.PICKING_UP_SKAT);
    }
}

private void runRamschGrandHandAnnouncing(){
    final boolean grandHandAnnounced = grandHand();

    if (grandHandAnnounced) {
        log.debug(data.getDeclarer() + " is playing grand hand"); //$NON-NLS-1$
        final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
        gaf.setGameType(GameType.GRAND);
        gaf.setHand(Boolean.TRUE);
        setGameAnnouncement(gaf.getAnnouncement());
        setGameState(GameState.TRICK_PLAYING);
        log.debug("grand hand game started"); //$NON-NLS-1$
        break;
    } else {
        if (JSkatOptions.instance().isSchieberamsch(true)) {
            log.debug("no grand hand - initiating schieberamsch"); //$NON-NLS-1$
            setGameState(GameState.SCHIEBERAMSCH);
        } else {
            log.debug("no grand hand and no schieberamsch - play ramsch"); //$NON-NLS-1$
            setGameState(GameState.TRICK_PLAYING);
        }
    }
}

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
        setActivePlayer(activePlayerBeforeContraRe);
    }
}

```



```

        bidding();
    }

    if (GameType.PASSED_IN.equals(data.getGameType())) {
        setGameState(GameState.PRELIMINARY_GAME_END);
    } else if (GameType.RAMSCH.equals(data.getGameType())) {
        setGameState(GameState.RAMSCH_GRAND_HAND_ANNOUNCING);
    } else {
        view.setDeclarer(tableName, data.getDeclarer());
        setGameState(GameState.PICKING_UP_SKAT);
    }
}

private boolean runRamschGrandHandAnnouncing(){
    final boolean grandHandAnnounced = grandHand();

    if (grandHandAnnounced) {
        log.debug(data.getDeclarer() + " is playing grand hand"); //$NON-NLS-1$
        final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
        gaf.setGameType(GameType.GRAND);
        gaf.setHand(Boolean.TRUE);
        setGameAnnouncement(gaf.getAnnouncement());
        setGameState(GameState.TRICK_PLAYING);
        log.debug("grand hand game started"); //$NON-NLS-1$
        break;
    } else {
        if (JSkatOptions.instance().isSchieberamsch(true)) {
            log.debug("no grand hand - initiating schieberamsch"); //$NON-NLS-1$
            setGameState(GameState.SCHIEBERAMSCH);
        } else {
            log.debug("no grand hand and no schieberamsch - play ramsch"); //$NON-NLS-1$
            setGameState(GameState.TRICK_PLAYING);
        }
    }
}

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
        setActivePlayer(activePlayerBeforeContraRe);
    }
}

```



```

        bidding();
    }

    if (GameType.PASSED_IN.equals(data.getGameType())) {
        setGameState(GameState.PRELIMINARY_GAME_END);
    } else if (GameType.RAMSCH.equals(data.getGameType())) {
        setGameState(GameState.RAMSCH_GRAND_HAND_ANNOUNCING);
    } else {
        view.setDeclarer(tableName, data.getDeclarer());
        setGameState(GameState.PICKING_UP_SKAT);
    }
}

private boolean runRamschGrandHandAnnouncing(){
    final boolean grandHandAnnounced = grandHand();

    if (grandHandAnnounced) {
        log.debug(data.getDeclarer() + " is playing grand hand"); //$NON-NLS-1$
        final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
        gaf.setGameType(GameType.GRAND);
        gaf.setHand(Boolean.TRUE);
        setGameAnnouncement(gaf.getAnnouncement());
        setGameState(GameState.TRICK_PLAYING);
        log.debug("grand hand game started"); //$NON-NLS-1$
        return true;
    } else {
        if (JSkatOptions.instance().isSchieberamsch(true)) {
            log.debug("no grand hand - initiating schieberamsch"); //$NON-NLS-1$
            setGameState(GameState.SCHIEBERAMSCH);
        } else {
            log.debug("no grand hand and no schieberamsch - play ramsch"); //$NON-NLS-1$
            setGameState(GameState.TRICK_PLAYING);
        }
    }
}

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
        setActivePlayer(activePlayerBeforeContraRe);
    }
}

```



```

        bidding();
    }

    if (GameType.PASSED_IN.equals(data.getGameType())) {
        setGameState(GameState.PRELIMINARY_GAME_END);
    } else if (GameType.RAMSCH.equals(data.getGameType())) {
        setGameState(GameState.RAMSCH_GRAND_HAND_ANNOUNCING);
    } else {
        view.setDeclarer(tableName, data.getDeclarer());
        setGameState(GameState.PICKING_UP_SKAT);
    }
}

private boolean runRamschGrandHandAnnouncing(){
    final boolean grandHandAnnounced = grandHand();

    if (grandHandAnnounced) {
        log.debug(data.getDeclarer() + " is playing grand hand"); //$NON-NLS-1$
        final GameAnnouncementFactory gaf = GameAnnouncement.getFactory();
        gaf.setGameType(GameType.GRAND);
        gaf.setHand(Boolean.TRUE);
        setGameAnnouncement(gaf.getAnnouncement());
        setGameState(GameState.TRICK_PLAYING);
        log.debug("grand hand game started"); //$NON-NLS-1$
        return true;
    } else {
        if (JSkatOptions.instance().isSchieberamsch(true)) {
            log.debug("no grand hand - initiating schieberamsch"); //$NON-NLS-1$
            setGameState(GameState.SCHIEBERAMSCH);
        } else {
            log.debug("no grand hand and no schieberamsch - play ramsch"); //$NON-NLS-1$
            setGameState(GameState.TRICK_PLAYING);
        }
    }
}

return false;
}

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
    }
}

```


Следва извеждането на още един блок от програмен код:




```

do {
    log.debug("Game state: " + data.getGameState());

    switch (data.getGameState()) {
    case GAME_START:
        setGameState(GameState.DEALING);
        break;
    case DEALING:
        dealCards();
        setGameState(GameState.BIDDING);
        break;
    case BIDDING:
        runBidding();
        break;
    case RAMSCH_GRAND_HAND_ANNOUNCING:
        if (runRamschGrandHandAnnouncing() == true) {
            break;
        }
        break;
    case SCHIEBERAMSCH:
        schieberamsch();
        final GameAnnouncementFactory factory = GameAnnouncement.getFactory();
        factory.setGameType(GameType.RAMSCH);
        setGameAnnouncement(factory.getAnnouncement());
        setGameState(GameState.TRICK_PLAYING);
        break;
    case PICKING_UP_SKAT:
        setActivePlayer(data.getDeclarer());
        if (pickUpSkat()) {
            setGameState(GameState.DISCARDING);
        } else {
            setGameState(GameState.DECLARING);
        }
        break;
    case DISCARDING:
        setActivePlayer(data.getDeclarer());
        discarding();
        if (!GameState.PRELIMINARY_GAME_END.equals(data.getGameState())) {
            setGameState(GameState.DECLARING);
        }
        break;
    case DECLARING:
        announceGame();
        if (isContraPlayEnabled(ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
            setGameState(GameState.CONTRA);
        }
    }
}

```



```

        setState(GameState.TRICK_PLAYING);
        log.debug("grand hand game started"); //$NON-NLS-1$
        return true;
    } else {
        if (JSkatOptions.instance().isSchieberamsch(true)) {
            log.debug("no grand hand - initiating schieberamsch"); //$NON-NLS-1$
            setGameState(GameState.SCHIEBERAMSCH);
        } else {
            log.debug("no grand hand and no schieberamsch - play ramsch"); //$NON-NLS-1$
            setGameState(GameState.TRICK_PLAYING);
        }
    }

    return false;
}

```

```

private void runSchieberamsch() {
    schieberamsch();
    final GameAnnouncementFactory factory = GameAnnouncement.getFactory();
    factory.setGameType(GameType.RAMSCH);
    setGameAnnouncement(factory.getAnnouncement());
    setGameState(GameState.TRICK_PLAYING);
}

```

```

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
        setActivePlayer(activePlayerBeforeContraRe);
    }
}

```

```

private boolean grandHand() {
    boolean grandHandAnnounced = false;

    for (final Player currPlayer : Player.getOrderedList()) {
        setActivePlayer(currPlayer);
        if (!grandHandAnnounced && playGrandHand()) {
            log.debug("Player " + activePlayer + " is playing grand hand.");
            setDeclarer(activePlayer);
            grandHandAnnounced = true;
        } else {
            log.debug("Player " + activePlayer + " doesn't want to play grand hand.");
        }
    }
}

```



```

log.debug("game state: " + data.getGameState());

switch (data.getGameState()) {
case GAME_START:
    setGameState(GameState.DEALING);
    break;
case DEALING:
    dealCards();
    setGameState(GameState.BIDDING);
    break;
case BIDDING:
    runBidding();
    break;
case RAMSCH_GRAND_HAND_ANNOUNCING:
    if (runRamschGrandHandAnnouncing() == true) {
        break;
    }
    break;
case SCHIEBERAMSCH:
    runSchieberamsch();
    break;
case PICKING_UP_SKAT:
    setActivePlayer(data.getDeclarer());
    if (pickUpSkat()) {
        setGameState(GameState.DISCARDING);
    } else {
        setGameState(GameState.DECLARING);
    }
    break;
case DISCARDING:
    setActivePlayer(data.getDeclarer());
    discarding();
    if (!GameState.PRELIMINARY_GAME_END.equals(data.getGameState())) {
        setGameState(GameState.DECLARING);
    }
    break;
case DECLARING:
    announceGame();
    if (isContraPlayEnabled(ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
        setGameState(GameState.CONTRA);
    } else {
        setGameState(GameState.TRICK_PLAYING);
    }
    break;
case CONTRA:
    for (final Player player : Player.getOrderedList()) {
        if (!isContraEnabledForPlayer(player, ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {

```


И следващ блок код:




```

    case DEALING:
        dealCards();
        setGameState(GameState.BIDDING);
        break;
    case BIDDING:
        runBidding();
        break;
    case RAMSCH_GRAND_HAND_ANNOUNCING:
        if (runRamschGrandHandAnnouncing() == true) {
            break;
        }
        break;
    case SCHIEBERAMSCH:
        runSchieberamsch();
        break;
    case PICKING_UP_SKAT:
        setActivePlayer(data.getDeclarer());
        if (pickUpSkat()) {
            setGameState(GameState.DISCARDING);
        } else {
            setGameState(GameState.DECLARING);
        }
        break;
    case DISCARDING:
        setActivePlayer(data.getDeclarer());
        discarding();
        if (!GameState.PRELIMINARY_GAME_END.equals(data.getGameState())) {
            setGameState(GameState.DECLARING);
        }
        break;
    case DECLARING:
        announceGame();
        if (isContraPlayEnabled(ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
            setGameState(GameState.CONTRA);
        } else {
            setGameState(GameState.TRICK_PLAYING);
        }
        break;
    case CONTRA:
        for (final Player player : Player.getOrderedList()) {
            if (isContraEnabledForPlayer(player, ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
                setActivePlayer(player);
                contraRe();
            }
        }
        setGameState(GameState.TRICK_PLAYING);

```



```

    return false;
}

private void runSchieberamsch() {
    schieberamsch();
    final GameAnnouncementFactory factory = GameAnnouncement.getFactory();
    factory.setGameType(GameType.RAMSCH);
    setGameAnnouncement(factory.getAnnouncement());
    setGameState(GameState.TRICK_PLAYING);
}

private void runPickingUpSkat() {
    setActivePlayer(data.getDeclarer());
    if (pickUpSkat()) {
        setGameState(GameState.DISCARDING);
    } else {
        setGameState(GameState.DECLARING);
    }
}

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
        setActivePlayer(activePlayerBeforeContraRe);
    }
}

private boolean grandHand() {
    boolean grandHandAnnounced = false;

    for (final Player currPlayer : Player.getOrderedList()) {
        setActivePlayer(currPlayer);
        if (!grandHandAnnounced && playGrandHand()) {
            log.debug("Player " + activePlayer + " is playing grand hand.");
            setDeclarer(activePlayer);
            grandHandAnnounced = true;
        } else {
            log.debug("Player " + activePlayer + " doesn't want to play grand hand.");
        }
    }
}

```



```

    }

    setGameState(GameState.GAME_START);
}

/**
 * Runs the skat game.
 *
 * @return Result of the skat game
 */
public SkatGameResult run() {
    // FIXME jan 11.07.2013: this method is too long, break it down to smaller
    // methods or implement it in another way
    view.setGameState(tableName, data.getGameState());

    do {
        log.debug("Game state: " + data.getGameState());

        switch (data.getGameState()) {
            case GAME_START:
                setGameState(GameState.DEALING);
                break;
            case DEALING:
                dealCards();
                setGameState(GameState.BIDDING);
                break;
            case BIDDING:
                runBidding();
                break;
            case RAMSCH_GRAND_HAND_ANNOUNCING:
                if (runRamschGrandHandAnnouncing() == true) {
                    break;
                }
                break;
            case SCHIEBERAMSCH:
                runSchieberamsch();
                break;
            case PICKING_UP_SKAT:
                runPickingUpSkat();
                break;
            case DISCARDING:
                setActivePlayer(data.getDeclarer());
                discarding();
                if (!GameState.PRELIMINARY_GAME_END.equals(data.getGameState())) {
                    setGameState(GameState.DECLARING);
                }
        }
    }
}

```


И следващ блок код:




```

    case DEALING:
        dealCards();
        setGameState(GameState.BIDDING);
        break;
    case BIDDING:
        runBidding();
        break;
    case RAMSCH_GRAND_HAND_ANNOUNCING:
        if (runRamschGrandHandAnnouncing() == true) {
            break;
        }
        break;
    case SCHIEBERAMSCH:
        runSchieberamsch();
        break;
    case PICKING_UP_SKAT:
        runPickingUpSkat();
        break;
    case DISCARDING:
        setActivePlayer(data.getDeclarer());
        discarding();
        if (!GameState.PRELIMINARY_GAME_END.equals(data.getGameState())) {
            setGameState(GameState.DECLARING);
        }
        break;
    case DECLARING:
        announceGame();
        if (isContraPlayEnabled(ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
            setGameState(GameState.CONTRA);
        } else {
            setGameState(GameState.TRICK_PLAYING);
        }
        break;
    case CONTRA:
        for (final Player player : Player.getOrderedList()) {
            if (isContraEnabledForPlayer(player, ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
                setActivePlayer(player);
                contraRe();
            }
        }
        setGameState(GameState.TRICK_PLAYING);
        break;
    case TRICK_PLAYING:
        playTricks();
        setGameState(GameState.CALCULATING_GAME_VALUE);
        break;
    case PRELIMINARY_GAME_END:

```



```

    return false;
}

private void runSchieberamsch() {
    schieberamsch();
    final GameAnnouncementFactory factory = GameAnnouncement.getFactory();
    factory.setGameType(GameType.RAMSCH);
    setGameAnnouncement(factory.getAnnouncement());
    setGameState(GameState.TRICK_PLAYING);
}

private void runPickingUpSkat() {
    setActivePlayer(data.getDeclarer());
    if (pickUpSkat()) {
        setGameState(GameState.DISCARDING);
    } else {
        setGameState(GameState.DECLARING);
    }
}

private void runDiscarding() {
    setActivePlayer(data.getDeclarer());
    discarding();
    if (!GameState.PRELIMINARY_GAME_END.equals(data.getGameState())) {
        setGameState(GameState.DECLARING);
    }
}

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
        setActivePlayer(activePlayerBeforeContraRe);
    }
}

private boolean grandHand() {
    boolean grandHandAnnounced = false;

    for (final Player currPlayer : Player.getOrderedList()) {
        setActivePlayer(currPlayer);
    }
}

```



```

case DEALING:
    dealCards();
    setGameState(GameState.BIDDING);
    break;
case BIDDING:
    runBidding();
    break;
case RAMSCH_GRAND_HAND_ANNOUNCING:
    if (runRamschGrandHandAnnouncing() == true) {
        break;
    }
    break;
case SCHIEBERAMSCH:
    runSchieberamsch();
    break;
case PICKING_UP_SKAT:
    runPickingUpSkat();
    break;
case DISCARDING:
    runDiscarding();
    break;
case DECLARING:
    announceGame();
    if (isContraPlayEnabled(ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
        setGameState(GameState.CONTRA);
    } else {
        setGameState(GameState.TRICK_PLAYING);
    }
    break;
case CONTRA:
    for (final Player player : Player.getOrderedList()) {
        if (isContraEnabledForPlayer(player, ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
            setActivePlayer(player);
            contraRe();
        }
    }
    setGameState(GameState.TRICK_PLAYING);
    break;
case TRICK_PLAYING:
    playTricks();
    setGameState(GameState.CALCULATING_GAME_VALUE);
    break;
case PRELIMINARY_GAME_END:
    setGameState(GameState.CALCULATING_GAME_VALUE);
    break;
case CALCULATING_GAME_VALUE:
    calculateGameValue();

```


И следващ блок код:




```

case RAMSCH_GRAND_HAND_ANNOUNCING:
    if (runRamschGrandHandAnnouncing() == true) {
        break;
    }
    break;
case SCHIEBERAMSCH:
    runSchieberamsch();
    break;
case PICKING_UP_SKAT:
    runPickingUpSkat();
    break;
case DISCARDING:
    runDiscarding();
    break;
case DECLARING:
    announceGame();
    if (isContraPlayEnabled(ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
        setGameState(GameState.CONTRA);
    } else {
        setGameState(GameState.TRICK_PLAYING);
    }
    break;
case CONTRA:
    for (final Player player : Player.getOrderedList()) {
        if (isContraEnabledForPlayer(player, ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
            setActivePlayer(player);
            contraRe();
        }
    }
    setGameState(GameState.TRICK_PLAYING);
    break;
case TRICK_PLAYING:
    playTricks();
    setGameState(GameState.CALCULATING_GAME_VALUE);
    break;
case PRELIMINARY_GAME_END:
    setGameState(GameState.CALCULATING_GAME_VALUE);
    break;
case CALCULATING_GAME_VALUE:
    calculateGameValue();
    setGameState(GameState.GAME_OVER);
    break;
case GAME_OVER:
    break;
}

```



```

    } else {
        setGameState(GameState.DECLARING);
    }
}

private void runDiscarding() {
    setActivePlayer(data.getDeclarer());
    discarding();
    if (!GameState.PRELIMINARY_GAME_END.equals(data.getGameState())) {
        setGameState(GameState.DECLARING);
    }
}

```

```

private void runDeclaring() {
    announceGame();
    if (isContraPlayEnabled(ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
        setGameState(GameState.CONTRA);
    } else {
        setGameState(GameState.TRICK_PLAYING);
    }
}

```

```

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
        setActivePlayer(activePlayerBeforeContraRe);
    }
}

```

```

private boolean grandHand() {
    boolean grandHandAnnounced = false;

    for (final Player currPlayer : Player.getOrderedList()) {
        setActivePlayer(currPlayer);
        if (!grandHandAnnounced && playGrandHand()) {
            log.debug("Player " + activePlayer + " is playing grand hand.");
            setDeclarer(activePlayer);
            grandHandAnnounced = true;
        } else {
            log.debug("Player " + activePlayer + " doesn't want to play grand hand.");
        }
    }
}

```



```

* Runs the game.
* @return Result of the skat game
*/
public SkatGameResult run() {
    // FIXME jan 11.07.2013: this method is too long, break it down to smaller
    // methods or implement it in another way
    view.setGameState(tableName, data.getGameState());

    do {
        log.debug("Game state: " + data.getGameState());

        switch (data.getGameState()) {
            case GAME_START:
                setGameState(GameState.DEALING);
                break;
            case DEALING:
                dealCards();
                setGameState(GameState.BIDDING);
                break;
            case BIDDING:
                runBidding();
                break;
            case RAMSCH_GRAND_HAND_ANNOUNCING:
                if (runRamschGrandHandAnnouncing() == true) {
                    break;
                }
                break;
            case SCHIEBERAMSCH:
                runSchieberamsch();
                break;
            case PICKING_UP_SKAT:
                runPickingUpSkat();
                break;
            case DISCARDING:
                runDiscarding();
                break;
            case DECLARING:
                runDeclaring();
                break;
            case CONTRA:
                for (final Player player : Player.getOrderedList()) {
                    if (isContraEnabledForPlayer(player, ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
                        setActivePlayer(player);
                        contraRe();
                    }
                }
        }
    }
}

```


И следващ блок код:




```

        break;
    case RAMSCH_GRAND_HAND_ANNOUNCING:
        if (runRamschGrandHandAnnouncing() == true) {
            break;
        }
        break;
    case SCHIEBERAMSCH:
        runSchieberamsch();
        break;
    case PICKING_UP_SKAT:
        runPickingUpSkat();
        break;
    case DISCARDING:
        runDiscarding();
        break;
    case DECLARING:
        runDeclaring();
        break;
    case CONTRA:
        for (final Player player : Player.getOrderedList()) {
            if (isContraEnabledForPlayer(player, ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
                setActivePlayer(player);
                contraRe();
            }
        }
        setGameState(GameState.TRICK_PLAYING);
        break;
    case TRICK_PLAYING:
        playTricks();
        setGameState(GameState.CALCULATING_GAME_VALUE);
        break;
    case PRELIMINARY_GAME_END:
        setGameState(GameState.CALCULATING_GAME_VALUE);
        break;
    case CALCULATING_GAME_VALUE:
        calculateGameValue();
        setGameState(GameState.GAME_OVER);
        break;
    case GAME_OVER:
        break;
}

} while (data.getGameState() != GameState.GAME_OVER);

JSkatEventBus.TABLE_EVENT_BUSSES.get(tableName).unregister(data);

```



```

discarding();
if (!GameState.PRELIMINARY_GAME_END.equals(data.getGameState())) {
    setGameState(GameState.DECLARING);
}
}

```

```

private void runDeclaring() {
    announceGame();
    if (isContraPlayEnabled(ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
        setGameState(GameState.CONTRA);
    } else {
        setGameState(GameState.TRICK_PLAYING);
    }
}

```

```

private void runContra() {
    for (final Player player : Player.getOrderedList()) {
        if (isContraEnabledForPlayer(player, ContraCallingTime.AFTER_GAME_ANNOUNCEMENT, 0)) {
            setActivePlayer(player);
            contraRe();
        }
    }
    setGameState(GameState.TRICK_PLAYING);
}

```

```

private void contraRe() {
    if (getActivePlayerInstance().callContra()) {
        JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ContraEvent(activePlayer)));
        final Player activePlayerBeforeContraRe = activePlayer;
        setActivePlayer(data.getDeclarer());
        if (getActivePlayerInstance().callRe()) {
            JSkatEventBus.INSTANCE.post(new TableGameMoveEvent(tableName, new ReEvent(activePlayer)));
        }
        setActivePlayer(activePlayerBeforeContraRe);
    }
}

```

```

private boolean grandHand() {
    boolean grandHandAnnounced = false;

    for (final Player currPlayer : Player.getOrderedList()) {
        setActivePlayer(currPlayer);
        if (!grandHandAnnounced && playGrandHand()) {
            log.debug("Player " + activePlayer + " is playing grand hand.");
            setDeclarer(activePlayer);
            grandHandAnnounced = true;
        }
    }
}

```



```

*/
public SkatGameResult run() {
    // FIXME jan 11.07.2013: this method is too long, break it down to smaller
    // methods or implement it in another way
    view.setGameState(tableName, data.getGameState());

    do {
        log.debug("Game state: " + data.getGameState());

        switch (data.getGameState()) {
            case GAME_START:
                setGameState(GameState.DEALING);
                break;
            case DEALING:
                dealCards();
                setGameState(GameState.BIDDING);
                break;
            case BIDDING:
                runBidding();
                break;
            case RAMSCH_GRAND_HAND_ANNOUNCING:
                if (runRamschGrandHandAnnouncing() == true) {
                    break;
                }
                break;
            case SCHIEBERAMSCH:
                runSchieberamsch();
                break;
            case PICKING_UP_SKAT:
                runPickingUpSkat();
                break;
            case DISCARDING:
                runDiscarding();
                break;
            case DECLARING:
                runDeclaring();
                break;
            case CONTRA:
                runContra();
                break;
            case TRICK_PLAYING:
                playTricks();
                setGameState(GameState.CALCULATING_GAME_VALUE);
                break;
            case PRELIMINARY_GAME_END:
                setGameState(GameState.CALCULATING_GAME_VALUE);
        }
    } while (true);
}

```


Не блоковете от всички case конструкции ще бъдат изнесени в собствени методи. Достатъчно е само да се демонстрира начина за подобряване на програмния код:




```
MACMINI:jskat-multimodule todorbalabanov$ git status
```

```
On branch small-fix/run-method-split
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   jskat-base/src/main/java/org/jskat/control/SkatGame.java
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
MACMINI:jskat-multimodule todorbalabanov$ █
```


Тъй като не се спазва стратегията за разклоняване, избрана в проекта за контрол на версиите, промените се добавят в локалното хранилище и се изпращат директно до отдалеченото хранилище:




```
MACMINI:jskat-multimodule todorbalabanov$ git status
```

```
On branch small-fix/run-method-split
```

```
Changes not staged for commit:
```

```
  (use "git add <file>..." to update what will be committed)
```

```
  (use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   jskat-base/src/main/java/org/jskat/control/SkatGame.java
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
MACMINI:jskat-multimodule todorbalabanov$ git add .
```

```
MACMINI:jskat-multimodule todorbalabanov$ git commit -m "Split of method run was done."
```

```
[small-fix/run-method-split 95fee23f] Split of method run was done.
```

```
 1 file changed, 99 insertions(+), 67 deletions(-)
```

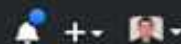
```
MACMINI:jskat-multimodule todorbalabanov$ █
```



```
MACMINI:jskat-multimodule todorbalabanov$ git push --set-upstream origin small-fix/run-method-split
Counting objects: 10, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 1.22 KiB | 626.00 KiB/s, done.
Total 10 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
remote:
remote: Create a pull request for 'small-fix/run-method-split' on GitHub by visiting:
remote:   https://github.com/Coding-Sunday-Sofia/jskat-multimodule/pull/new/small-fix/run-method-split
remote:
To https://github.com/Coding-Sunday-Sofia/jskat-multimodule.git
 * [new branch]      small-fix/run-method-split -> small-fix/run-method-split
Branch 'small-fix/run-method-split' set up to track remote branch 'small-fix/run-method-split' from 'origin'.
MACMINI:jskat-multimodule todorbalabanov$
```


В отдалеченото хранилище веднага се установява наличието на нов клон (branch) и се дава възможност за сравнение на промените и изпращане на Pull Request:



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

Coding-Sunday-Sofia / jskat-multimodule

forked from b0n541/jskat-multimodule

[Watch](#) 0[★ Star](#) 0[Fork](#) 10[Code](#)[Pull requests](#) 0[Actions](#)[Projects](#) 0[Wiki](#)[Security](#)[Insights](#)[Settings](#)

JSkat is a free software implementation of the game Skat in Java. <http://www.jskat.org>

[Edit](#)[Manage topics](#)[1,708](#) commits[8](#) branches[0](#) packages[10](#) releases[7](#) contributors

Your recently pushed branches:

[small-fix/run-method-split](#) (1 minute ago)

[Compare & pull request](#)

Branch: [develop](#)

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download](#)

This branch is even with b0n541:develop.

[Pull request](#) [Compare](#)

b0n541	Merge pull request b0n541#73 from b0n541/dependabot/gradle/org.reflec...	Latest commit fee7b07 23 days ago
gradle/wrapper	Update Gradle wrapper	29 days ago
info	new splash screen	5 years ago
jskat-base	Bump reflections from 0.9.11 to 0.9.12	23 days ago
jskat-javafx-gui	Bump junit from 4.12 to 4.13	29 days ago
jskat-swing-gui	Fix license header	5 months ago
.gitignore	Add IntelliJ's out dir to .gitignore	13 months ago
.travis.yml	Make JSkat compatible with JDK 11	14 months ago
README.md	Update Gradle tasks	5 months ago
build.gradle	Move on towards version 0.18.0	5 months ago
gradlew	Update Gradle wrapper	29 days ago
gradlew.bat	Update Gradle wrapper	29 days ago
settings.gradle	issue b0n541#38: first step towards Gradle builds	4 years ago

[README.md](#)

Вписват се малко допълнителни обяснения
за извършените промени:





Search or jump to...

Pull requests Issues Marketplace Explore



b0n541 / jskat-multimodule

Watch 1

Unstar 10

Fork 10

Code

Issues 22

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



base repository: b0n541/jskat-multimodule

base: develop



head repository: Coding-Sunday-Sofia/jskat-...

compare: small-fix/run-method-split

✓ Able to merge. These branches can be automatically merged.



Split of method run was done.

Write

Preview

AA B i “ <> 🔗 ☰ ☷ ↩

Method simplification was done by splitting the code into more helper methods.

Attach files by dragging & dropping, selecting or pasting them.



Create pull request

1 commit

1 file changed

0 commit comments

1 contributor



Commits on Jan 31, 2020



TodorBalabanov

Split of method run was done.

95fee23

Showing 1 changed file with 99 additions and 67 deletions.

Unified

Split

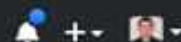
Следва проверка от автоматичната система
за изграждане на проекта:





Search or jump to...

Pull requests Issues Marketplace Explore



b0n541 / jskat-multimodule

Watch 1 Unstar 10 Fork 10

Code Issues 22 Pull requests 1 Actions Projects 0 Wiki Security Insights

Split of method run was done. #74

Edit

Open TodorBalabanov wants to merge 1 commit into b0n541:develop from Coding-Sunday-Sofia:small-fix/run-method-split

Conversation 0 Commits 1 Checks 0 Files changed 1 +99 -67



TodorBalabanov commented now



Method simplification was done by splitting the code into more helper methods.



Split of method run was done.

95fee23

Add more commits by pushing to the small-fix/run-method-split branch on Coding-Sunday-Sofia/jskat-multimodule.



Some checks haven't completed yet

Hide all checks

1 pending check



continuous-integration/travis-ci/pr Pending — The Travis CI build is in pr...

Details



This branch has no conflicts with the base branch

Only those with write access to this repository can merge pull requests.



Write Preview

AA B i “ < > @

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.



Close pull request

Comment

Reviewers



Suggestions

b0n541

Request

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Customize

Unsubscribe

You're receiving notifications because you authored the thread.

1 participant



С помощта на инструментите за обследване на програмен код могат лесно да се открият подходящи места в проектите, които да бъдат преработени. С помощта на повече участия в проекти с отворен код всеки специалист може да подобри портфолиото си, а от там и шансовете за намиране на по-добре платена работа.



