

ЭКРАН ЗА НАСТРОЙКИ В ИГРА С ОТКРИТИ УСЛОВИЯ

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

При създаването на логически игри за мобилни устройства един от съществените компоненти за задържане на вниманието на играчите е способността на компютърния опонент да оказва достатъчно сериозна съпротива.



В играта DiceOverflow двама играчи се надпреварват за завземането на цялото игрално поле. В клетките на игралното поле се разполагат зарове с шест стени и стойности на всяка страна от 1 до 6.



4:04



LTE



Dice Overflow



Преливането може да предизвика лавинообразно препълване и на съседните клетки, като това от своя страна води до ново преливане. Целта на всеки играч е да завземе цялата налична територия и да отстрани изцяло противника.



4:07

LTE



Dice Overflow



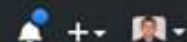
От областта на изкуствения интелект могат да се привлекат множество методи за създаване на компютърен опонент при такава антагонистична игра с открити условия. Най-лесният за реализация вариант е, ако компютърният опонент избира клетки на случаен принцип.





Search or jump to...

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



[VelbazhdSoftwareLLC](#) / [DiceOverflow](#)

[Unwatch](#) 1

[Unstar](#) 3

[Fork](#) 1

[Code](#)

[Issues](#) 0

[Pull requests](#) 0

[Actions](#)

[Projects](#) 0

[Wiki](#)

[Security](#)

[Insights](#)

[Settings](#)

Branch: [master](#)

[Create new file](#)

[Upload files](#)

[Find file](#)

[History](#)

[DiceOverflow](#) / [DiceOverflow](#) / [src](#) / [main](#) / [java](#) / [eu](#) / [veldsoft](#) / [dice](#) / [overflow](#) / [model](#) / [ai](#) /



TodorBalabanov Some thing ware removed from the list of the tasks.

✖ Latest commit 66c2c7b 4 days ago

..

AbstractArtificialIntelligence.java	Migration of Eclipse project to Android Studio project.	14 months ago
ArtificialIntelligence.java	Migration of Eclipse project to Android Studio project.	14 months ago
ImpossibleMoveException.java	Migration of Eclipse project to Android Studio project.	14 months ago
MonteCarloArtificialIntelligence.java	Migration of Eclipse project to Android Studio project.	14 months ago
NeuralNetworkArtificialIntelligence.java	Some thing ware removed from the list of the tasks.	4 days ago
RandomArtificialIntelligence.java	Migration of Eclipse project to Android Studio project.	14 months ago



Случайният избор на валиден ход е метод наречен „случайно търсене“. При този метод по никакъв начин не се преценява до колко по-перспективен е един ход спрямо друг.





Random search

From Wikipedia, the free encyclopedia

Random search (RS) is a family of numerical [optimization](#) methods that [do not require the gradient](#) of the problem to be optimized, and RS can hence be used on functions that are not [continuous](#) or [differentiable](#). Such optimization methods are also known as direct-search, derivative-free, or black-box methods.

The name "random search" is attributed to Rastrigin^[1] who made an early presentation of RS along with basic mathematical analysis. RS works by iteratively moving to better positions in the search-space, which are sampled from a [hypersphere](#) surrounding the current position.

Contents [hide]

- [Algorithm](#)
- [Variants](#)
- [See also](#)
- [References](#)

Algorithm [[edit source](#)]

Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be the fitness or cost function which must be minimized. Let $\mathbf{x} \in \mathbb{R}^n$ designate a position or candidate solution in the search-space. The basic RS algorithm can then be described as:

- Initialize \mathbf{x} with a random position in the search-space.
- Until a termination criterion is met (e.g. number of iterations performed, or adequate fitness reached), repeat the following:
 - Sample a new position \mathbf{y} from the [hypersphere](#) of a given radius surrounding the current position \mathbf{x} (see e.g. [Marsaglia's technique](#) for sampling a hypersphere.)
 - If $f(\mathbf{y}) < f(\mathbf{x})$ then move to the new position by setting $\mathbf{x} = \mathbf{y}$

Variants [[edit source](#)]

A number of RS variants have been introduced in the literature:

- Fixed Step Size Random Search (FSSRS) is Rastrigin's ^[1] basic algorithm which samples from a hypersphere of fixed radius.
- Optimum Step Size Random Search (OSSRS) by Schumer and Steiglitz ^[2] is primarily a theoretical study on how to optimally adjust the radius of the hypersphere so as to allow for speedy convergence to the optimum. An actual implementation of the OSSRS needs to approximate this optimal radius by repeated sampling and is therefore expensive to execute.
- Adaptive Step Size Random Search (ASSRS) by Schumer and Steiglitz ^[2] attempts to heuristically adapt the hypersphere's radius: two new candidate solutions are generated, one with the current nominal step size and one with a larger step-size. The larger step size becomes the new nominal step size if and only if it leads to a larger improvement. If for several iterations neither of the steps leads to an improvement, the nominal step size is reduced.
- Optimized Relative Step Size Random Search (ORSSRS) by Schrack and Choit ^[3] approximate the optimal step size by a simple exponential decrease. However, the formula for computing the decrease-factor is somewhat complicated.

See also [[edit source](#)]

- [Random optimization](#) is a closely related family of optimization methods which sample from a [normal distribution](#) instead of a hypersphere.
- [Luus–Jaakola](#) is a closely related optimization method using a [uniform distribution](#) in its sampling and a simple formula for exponentially decreasing the sampling

Методът за случайния избор може значително да се подобри, ако едно игрално табло бъде разиграно с множество случайно избрани ходове (както за единия опонент, така и за другия). Този вид стохастично изследване на дървото на състоянията позволява да се преброи кой ход е довел до най-голям брой крайни печеливши ситуации.





Search or jump to...



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



[VelbazhdSoftwareLLC](#) / [DiceOverflow](#)

Unwatch ▾

1

★ Unstar

3

Fork

1

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

Branch: master ▾

Create new file

Upload files

Find file

History

[DiceOverflow](#) / [DiceOverflow](#) / [src](#) / [main](#) / [java](#) / [eu](#) / [veldsoft](#) / [dice](#) / [overflow](#) / [model](#) / [ai](#) /



TodorBalabanov Some thing ware removed from the list of the tasks.

✖ Latest commit 66c2c7b 4 days ago

..


AbstractArtificialIntelligence.java	Migration of Eclipse project to Android Studio project.	14 months ago
ArtificialIntelligence.java	Migration of Eclipse project to Android Studio project.	14 months ago
ImpossibleMoveException.java	Migration of Eclipse project to Android Studio project.	14 months ago
MonteCarloArtificialIntelligence.java	Migration of Eclipse project to Android Studio project.	14 months ago
NeuralNetworkArtificialIntelligence.java	Some thing ware removed from the list of the tasks.	4 days ago
RandomArtificialIntelligence.java	Migration of Eclipse project to Android Studio project.	14 months ago



Тъй като ходовете се избират с помощта на псевдо-случайни числа, то този метод спада към групата Монте Карло методи. А тъй като игрите с открити условия се описват чрез дърво на състоянията, избирането на ходове е своеобразно спускане по дървото на състоянията. От там идва и названието на този метод, а именно „Монте Карло търсене в дърво“.





Photograph your local culture, help Wikipedia and win! 

Monte Carlo tree search

From Wikipedia, the free encyclopedia

In [computer science](#), **Monte Carlo tree search** (**MCTS**) is a [heuristic search algorithm](#) for some kinds of [decision processes](#), most notably those employed in game play. MCTS was introduced in 2006 for [computer Go](#).^[1] It has been used in other board games like [chess](#) and [shogi](#),^[2] games with incomplete information such as [bridge](#)^[3] and [poker](#),^[4] as well as in turn-based-strategy video games (such as [Total War: Rome II](#)'s implementation in the high level campaign AI^[5]).

Contents [hide]

- History
 - 1.1 Monte Carlo Method
 - 1.2 Monte Carlo Tree Search (MCTS)
- Principle of operation
- Pure Monte Carlo game search
- Exploration and exploitation
- Advantages and disadvantages
- Improvements
- See also
- References
- Bibliography

History [\[edit source \]](#)

Monte Carlo Method [\[edit source \]](#)

The [Monte Carlo method](#), which uses randomness for deterministic problems difficult or impossible to solve using other approaches, dates back to the 1940s. In his 1987 PhD thesis, Bruce Abramson combined [minimax search](#) with an *expected-outcome model* based on random game playouts to the end, instead of the usual [static evaluation function](#). Abramson said the expected-outcome model "is shown to be precise, accurate, easily estimable, efficiently calculable, and domain-independent."^[6] He experimented in-depth with [Tic-tac-toe](#) and then with machine-generated evaluation functions for [Othello](#) and [Chess](#).

Such methods were then explored and successfully applied to heuristic search in the field of [automated theorem proving](#) by W. Ertel, J. Schumann and C. Suttner in 1989,^{[7][8][9]} thus improving the exponential search times of uninformed search algorithms such as e.g. breadth-first search, depth-first search or [iterative deepening](#).

In 1992, B. Brügmann employed it for the first time in a [Go-playing program](#).^[10] Chang et al.^[11] proposed the idea of "recursive rolling out and backtracking" with "adaptive" sampling choices in their Adaptive Multi-stage Sampling (AMS) algorithm for the model of Markov decision processes. AMS was the first work to explore the idea of UCB-based exploration and exploitation in constructing sampled/simulated (Monte Carlo) trees and was the main seed for UCT (Upper Confidence Trees).^[12]

Monte Carlo Tree Search (MCTS) [\[edit source \]](#)

In 2006, inspired by these predecessors,^[14] [Rémi Coulom](#) described the application of the Monte Carlo method to game-tree search and coined the name Monte Carlo tree search.^[15] [Kocsis and Szepesvári](#) developed the UCT (Upper Confidence Trees) algorithm.

Monte Carlo tree search

Class [Search algorithm](#)

Graph and tree search algorithms

[α-β](#) · [A*](#) · [B*](#) · [Backtracking](#) · [Beam](#) · [Bellman–Ford](#) · [Best-first](#) · [Bidirectional](#) · [Borůvka](#) · [Branch & bound](#) · [BFS](#) · [British Museum](#) · [D*](#) · [DFS](#) · [Dijkstra](#) · [Edmonds](#) · [Floyd–Warshall](#) · [Fringe search](#) · [Hill climbing](#) · [IDA*](#) · [Iterative deepening](#) · [Johnson](#) · [Jump point](#) · [Kruskal](#) · [Lexicographic BFS](#) · [LPA*](#) · [Prim](#) · [SMA*](#) · [SPFA](#)

Listings

[Graph algorithms](#) · [Search algorithms](#) · [List of graph algorithms](#)

Related topics

[Dynamic programming](#) · [Graph traversal](#) · [Tree traversal](#) · [Search games](#)

V · T · E

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

Interaction

[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools

[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)

Print/export

[Download as PDF](#)
[Printable version](#)

[Languages](#) 

[Español](#)
[فارسی](#)
[Français](#)
[한국어](#)
[日本語](#)
[Polski](#)
[Português](#)
[中文](#)

 [Edit links](#)

При игрите на дъска с открити условия винаги има краен брой ходове от които играчът може да избира. Всеки един от тези ходове може да получи оценка за перспективност. Ходът получил най-висока оценка би трябвало да е изборът на компютърния опонент в играта му срещу човек.



4:23



LTE



Dice Overflow



При Монте Карло изследването на дървото на състоянията се брои колко пъти всеки от възможните ходове е довел до победа. От съществено значение е да се отбележи, че това изследване винаги протича по различен начин, тъй като се базира на случайно генерирани числа, което пък води до разнообразни стратегии от страна на компютърния опонент.



4:26



LTE



Dice Overflow



При всички стохастични алгоритми от голямо значение е времетраенето на статистическото изследване. Колкото повече време се използва за изследване на дървото на състоянията, толкова по-печеливши ходове биха били оценени от компютърния опонент.



```

13 import android.widget.ImageView;
14 import android.widget.Toast;
15 import eu.veldsoft.dice.overflow.model.Board;
16 import eu.veldsoft.dice.overflow.model.Cell;
17 import eu.veldsoft.dice.overflow.model.ai.ArtificialIntelligence;
18 import eu.veldsoft.dice.overflow.model.ai.MonteCarloArtificialIntelligence;
19
20 /**
21  * Game screen.
22  *
23  * @author Todor Balabanov
24  */
25 public class GameActivity extends Activity {
26     /**
27      *
28      */
29     private final Handler handler = new Handler();
30
31     /**
32      * Sounds pool.
33      */
34     private SoundPool sounds = null;
35
36     /**
37      * Click sound identifier.
38      */
39     private int clickId = -1;
40
41     /**
42      * Finish sound identifier.
43      */
44     private int finishId = -1;
45
46     /**
47      * Computer opponent thread.
48      */
49     private Runnable ai = new Runnable() {
50         /**
51          * Computer opponent object.
52          */
53         private ArtificialIntelligence bot = new MonteCarloArtificialIntelligence(1000);
54         // TODO Use shared preferences to select calculations time.
55
56         /**
57          * {@inheritDoc}
58          */
59         @Override
60         public void run() {
61             /**
62              * If the game is over there is no need to play.
63              */

```

При програмирането на игри за Android е от съществено значение играта да не завзема твърде дълго и твърде агресивно ресурсите на операционната система. В същото време, не бива да се злоупотребява с търпението на играчите. Поради тази причина може да се определи време в което компютърният опонент да направи ход и това да се обвърже със сложността от страна на противника.

eu | veldsoft | dice | overflow | **GameActivity** | DiceOverflow-DiceOverflow | Nexus 10 API 28 | Git: ✓ | Event Log

GameActivity.java

24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71

```
*/
public class GameActivity extends Activity {
    /**
     *
     */
    private final Handler handler = new Handler();

    /**
     * Sounds pool.
     */
    private SoundPool sounds = null;

    /**
     * Click sound identifier.
     */
    private int clickId = -1;

    /**
     * Finish sound identifier.
     */
    private int finishId = -1;

    /**
     * Computer opponent thread.
     */
    private Runnable ai = new Runnable() {
        /**
         * Computer opponent object.
         */
        private ArtificialIntelligence bot = new MonteCarloArtificialIntelligence(1000);
        // TODO Use shared preferences to select calculations time.

        /**
         * {@inheritDoc}
         */
        @Override
        public void run() {
            /**
             * If the game is over there is no need to play.
             */
            if (board.isGameOver() == true) {
                return;
            }

            /**
             * Valid turn check.
             */
        }
    }
}
```

GameActivity > ai > new Runnable

Run | Logcat | TODO | Terminal | Version Control | Build | Profiler

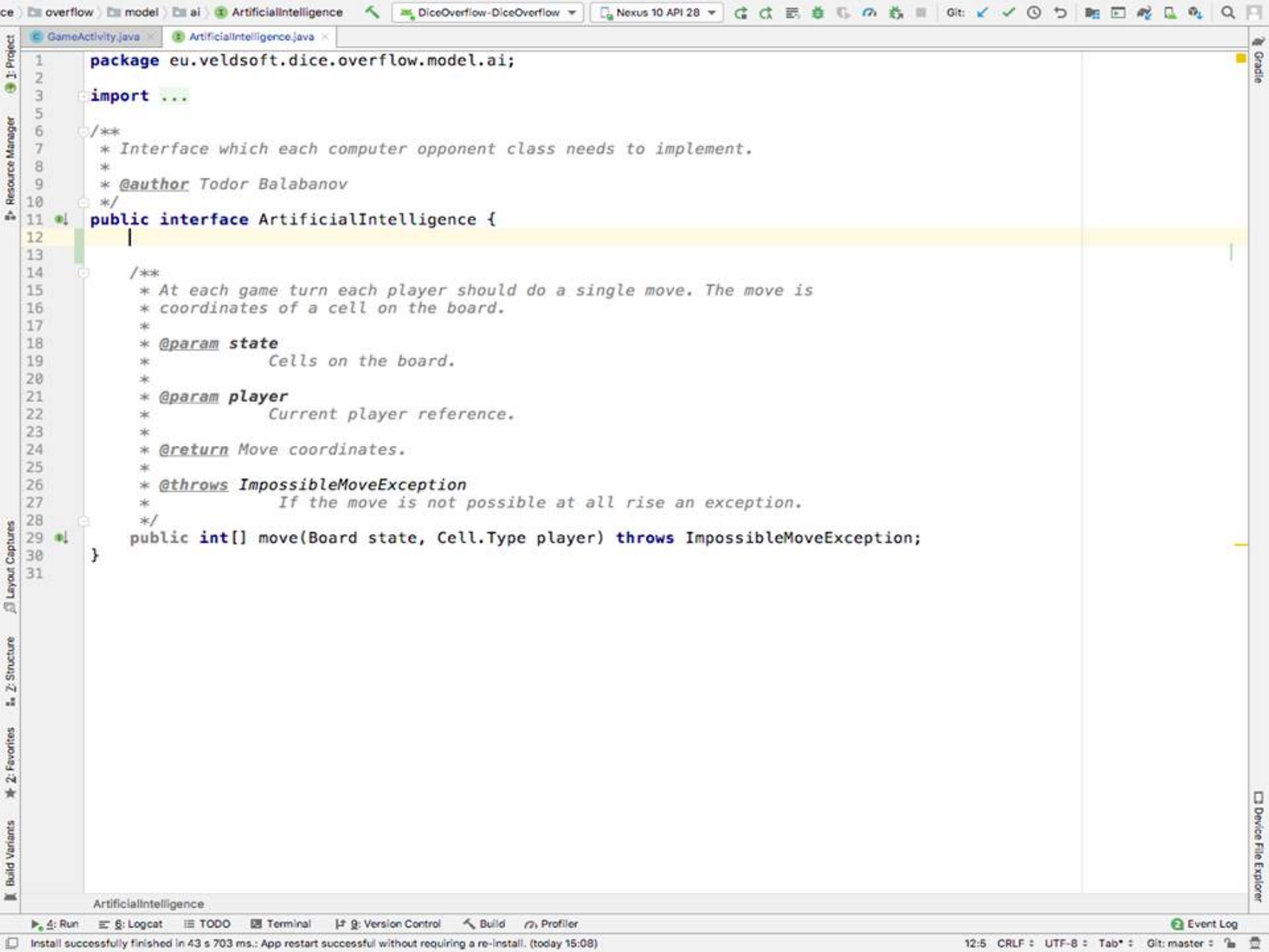
Install successfully finished in 43 s 703 ms.: App restart successful without requiring a re-install. (today 15:08)

145 chars, 2 line breaks | 55:1 | CRLF | UTF-8 | Tab* | Git: master

Project | Resource Manager | Layout Captures | Z: Structure | Z: Favorites | Build Variants | Device File Explorer

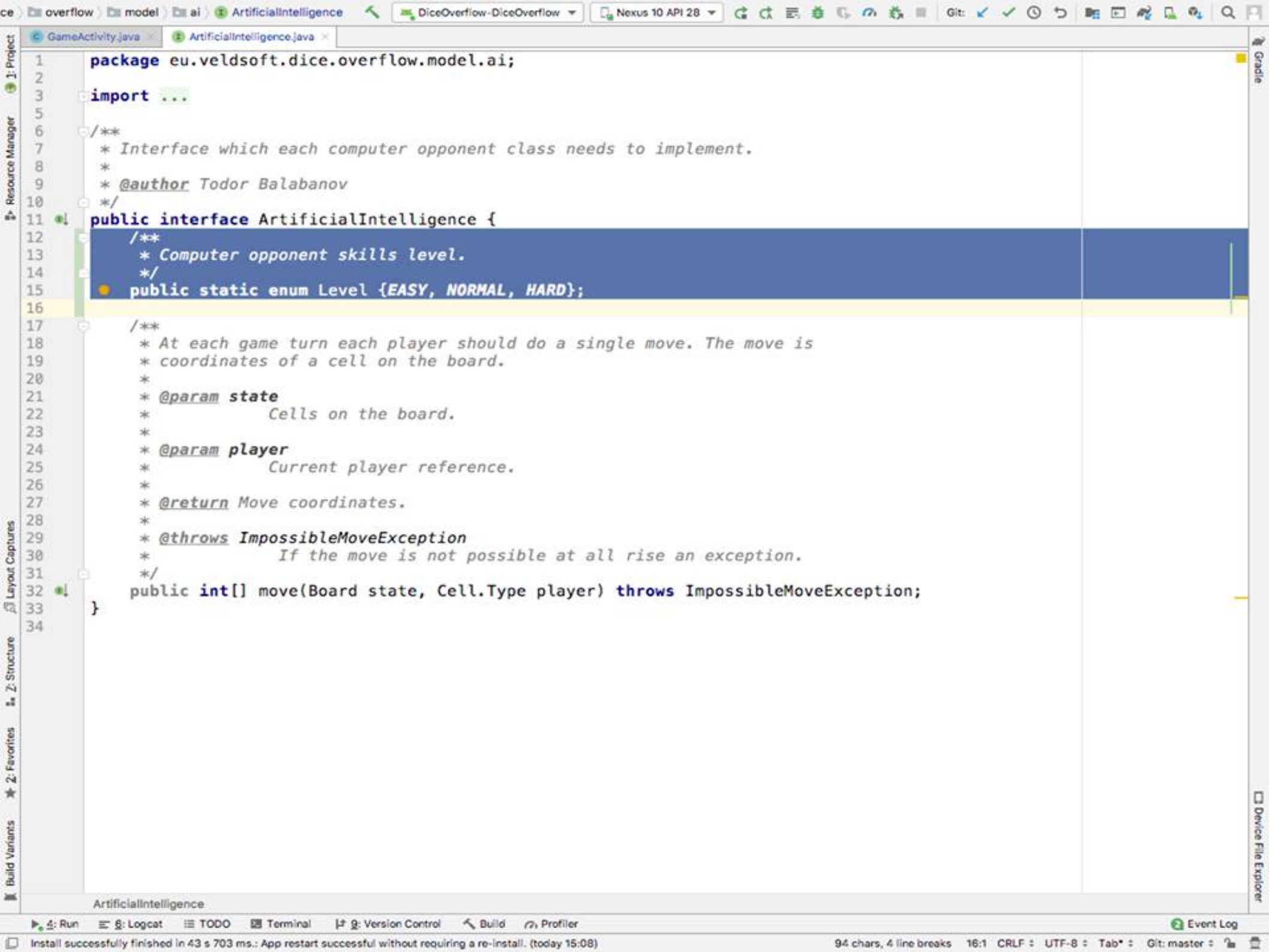
Удачно е за различните нива на опитност на компютърния опонент да се въведе подходящ изброим тип, който в последствие да се ползва за настройка на различните алгоритми, които изчисляват хода на компютъра.





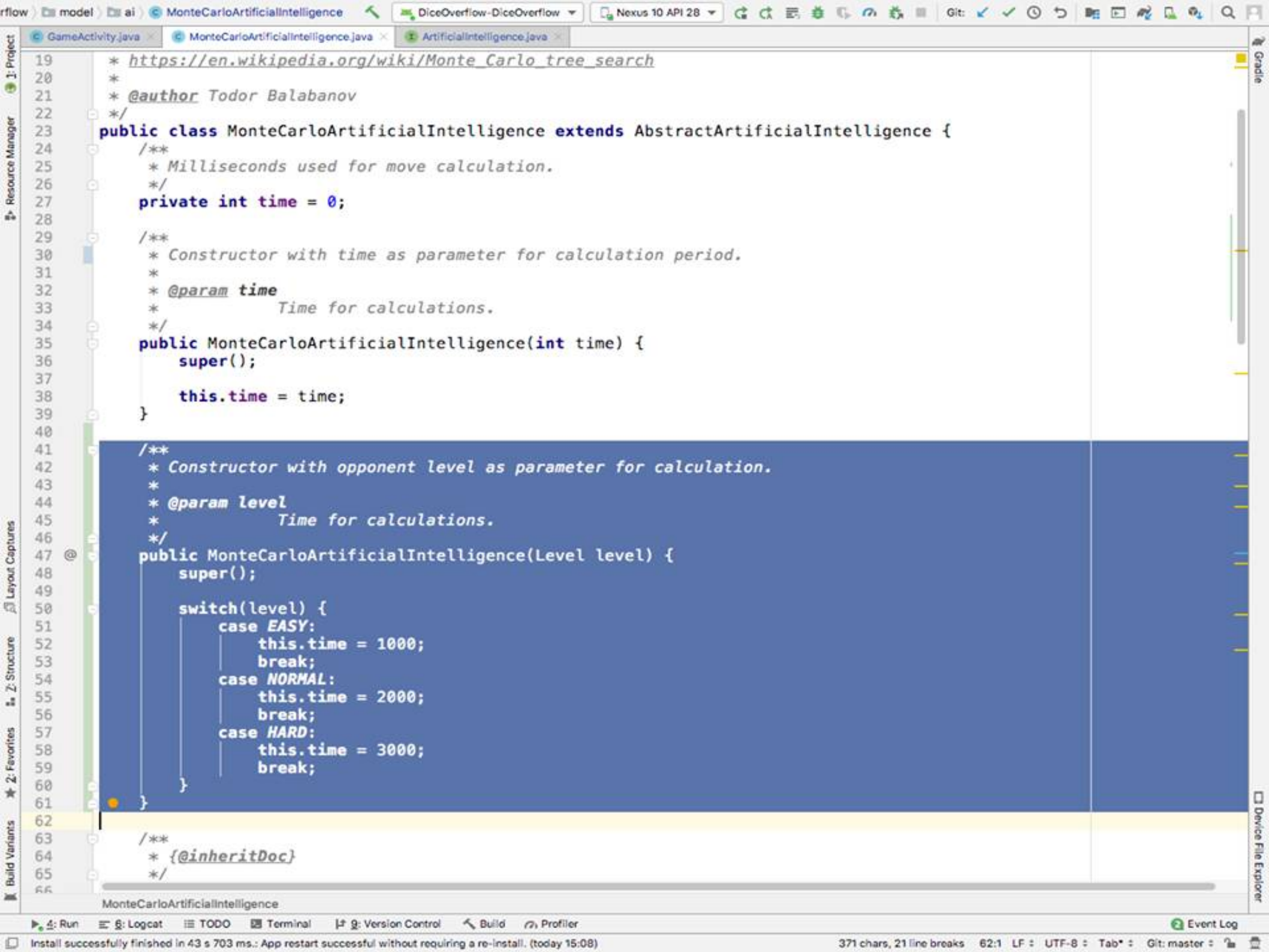
Първоначално три нива са напълно достатъчни: лесно, средно и трудно.





Тъй като се добавят възможности за настройка на уменията, които компютърният опонент би трябвало да демонстрира, то е удачно да се направи още един конструктор, който да определя по-вече време за извършване на пресмятанията при стохастичното изследване.





Потребителят трябва да може да избира нивото на сложност на компютърния опонент и поради тази причина е важно да се добави екран с настройки.



eu | veldsoft | dice | overflow | **GameActivity** | DiceOverflow-DiceOverflow | Nexus 10 API 28 | Git: ✓ ✓ ↺

GameActivity.java | MonteCarloArtificialIntelligence.java | ArtificialIntelligence.java

16 import eu.veldsoft.dice.overflow.model.Cell;
17 import eu.veldsoft.dice.overflow.model.ai.ArtificialIntelligence;
18 import eu.veldsoft.dice.overflow.model.ai.MonteCarloArtificialIntelligence;
19
20 /**
21 * Game screen.
22 *
23 * @author Todor Balabanov
24 */
25 public class GameActivity extends Activity {
26 /**
27 *
28 */
29 private final Handler handler = new Handler();
30
31 /**
32 * Sounds pool.
33 */
34 private SoundPool sounds = null;
35
36 /**
37 * Click sound identifier.
38 */
39 private int clickId = -1;
40
41 /**
42 * Finish sound identifier.
43 */
44 private int finishId = -1;
45
46 /**
47 * Computer opponent thread.
48 */
49 private Runnable ai = new Runnable() {
50 /**
51 * Computer opponent object.
52 */
53 private ArtificialIntelligence bot =
54 new MonteCarloArtificialIntelligence(ArtificialIntelligence.Level.EASY);
55
56 /**
57 * {@inheritDoc}
58 */
59 @Override
60 public void run() {
61 /**
62 * If the game is over there is no need to play.
63

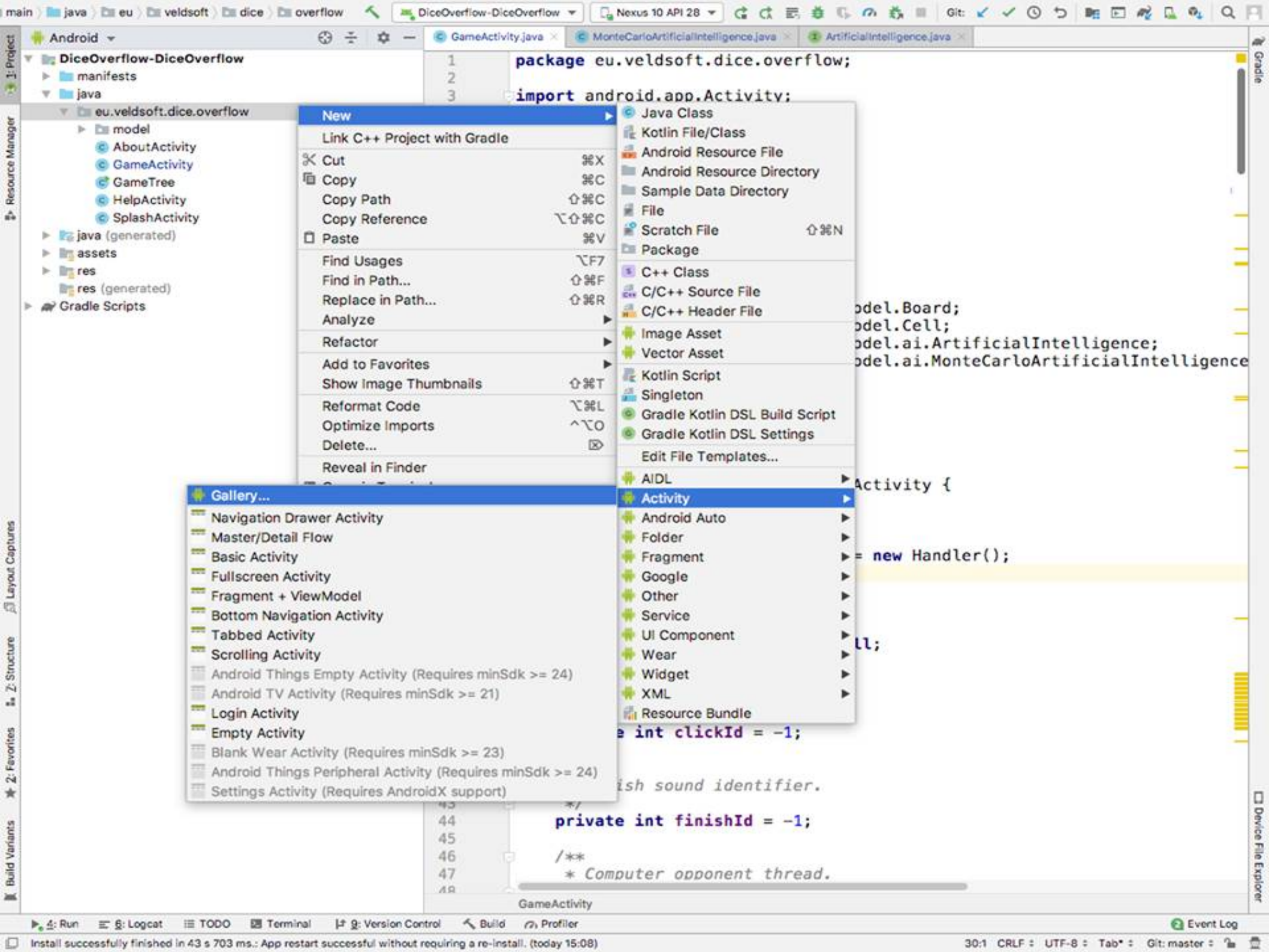
GameActivity > ai > new Runnable

Run | Logcat | TODO | Terminal | Version Control | Build | Profiler | Event Log

Install successfully finished in 43 s 703 ms.: App restart successful without requiring a re-install. (today 15:08) 117 chars, 2 line breaks 55:1 CRLF : UTF-8 : Tab* : Git: master :

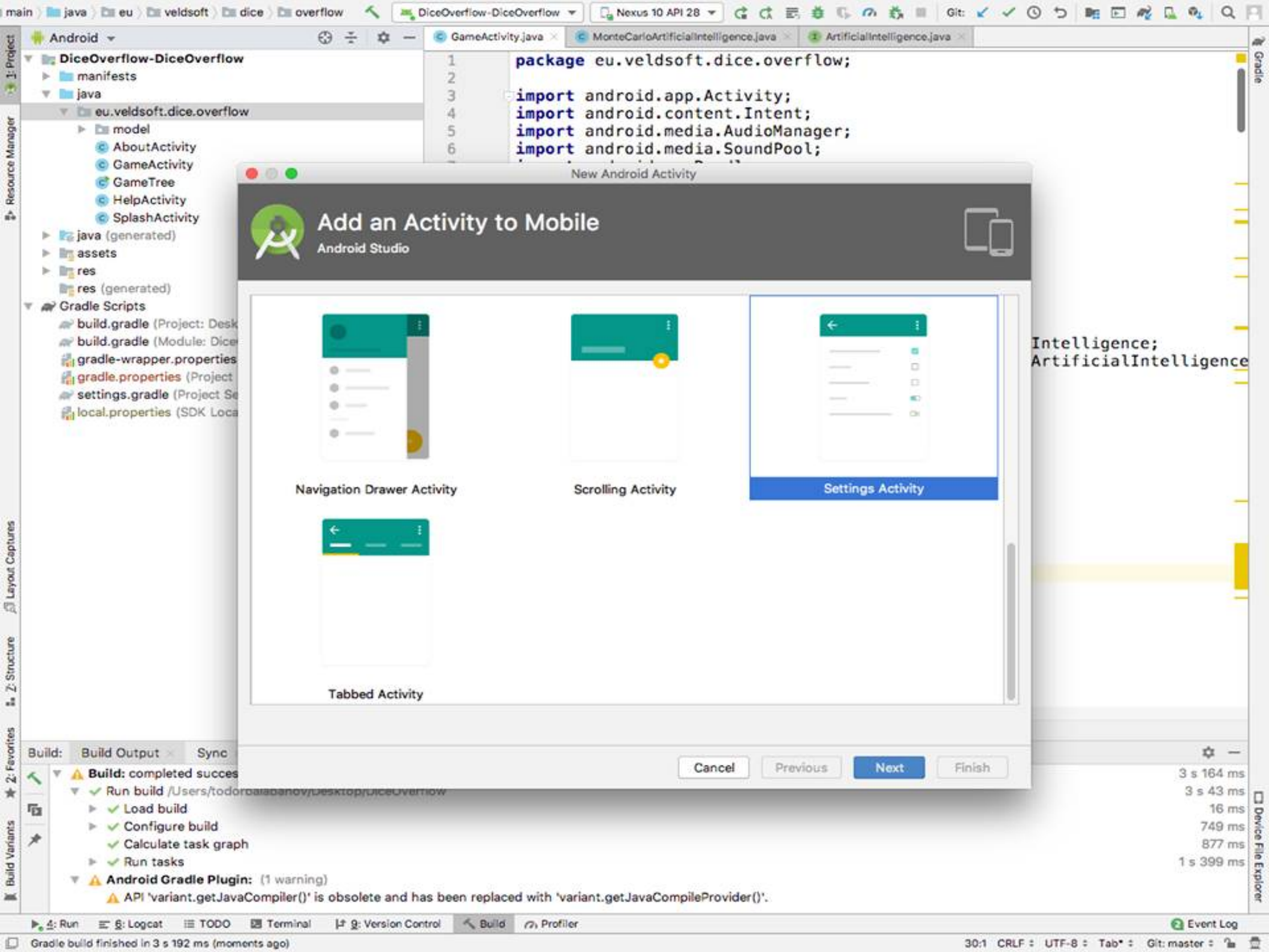
Екраните за настройки са лесни за създаване в Android, тъй като има конкретно предназначена заготовка на такъв екран. Също така са разработени серия визуални контроли, които да съхраняват избраните стойности в механизъм наречен Shared Preferences.





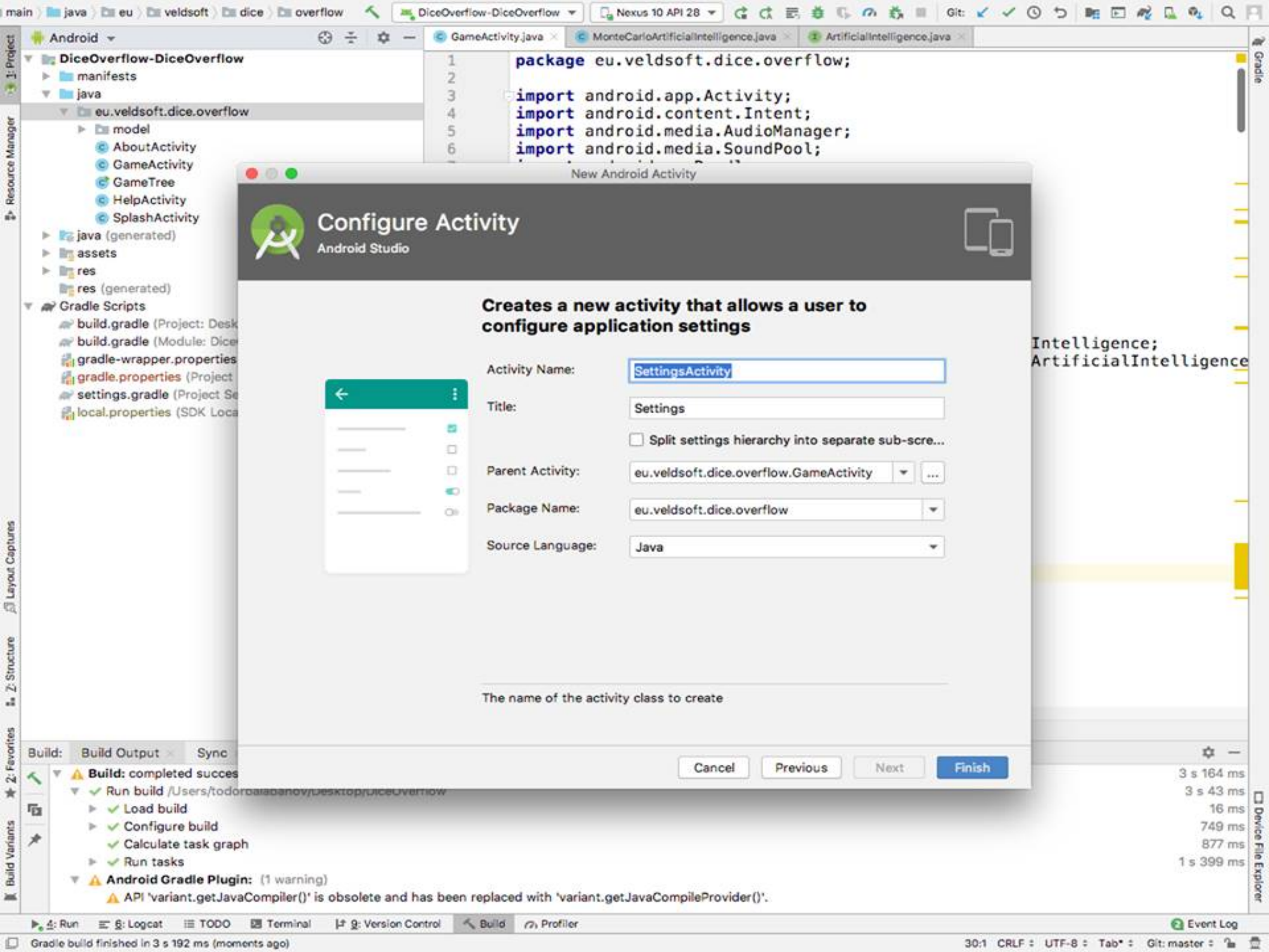
Екранът с настройки е екран, който ще се извика от главния екран с помощта на опция в менюто.





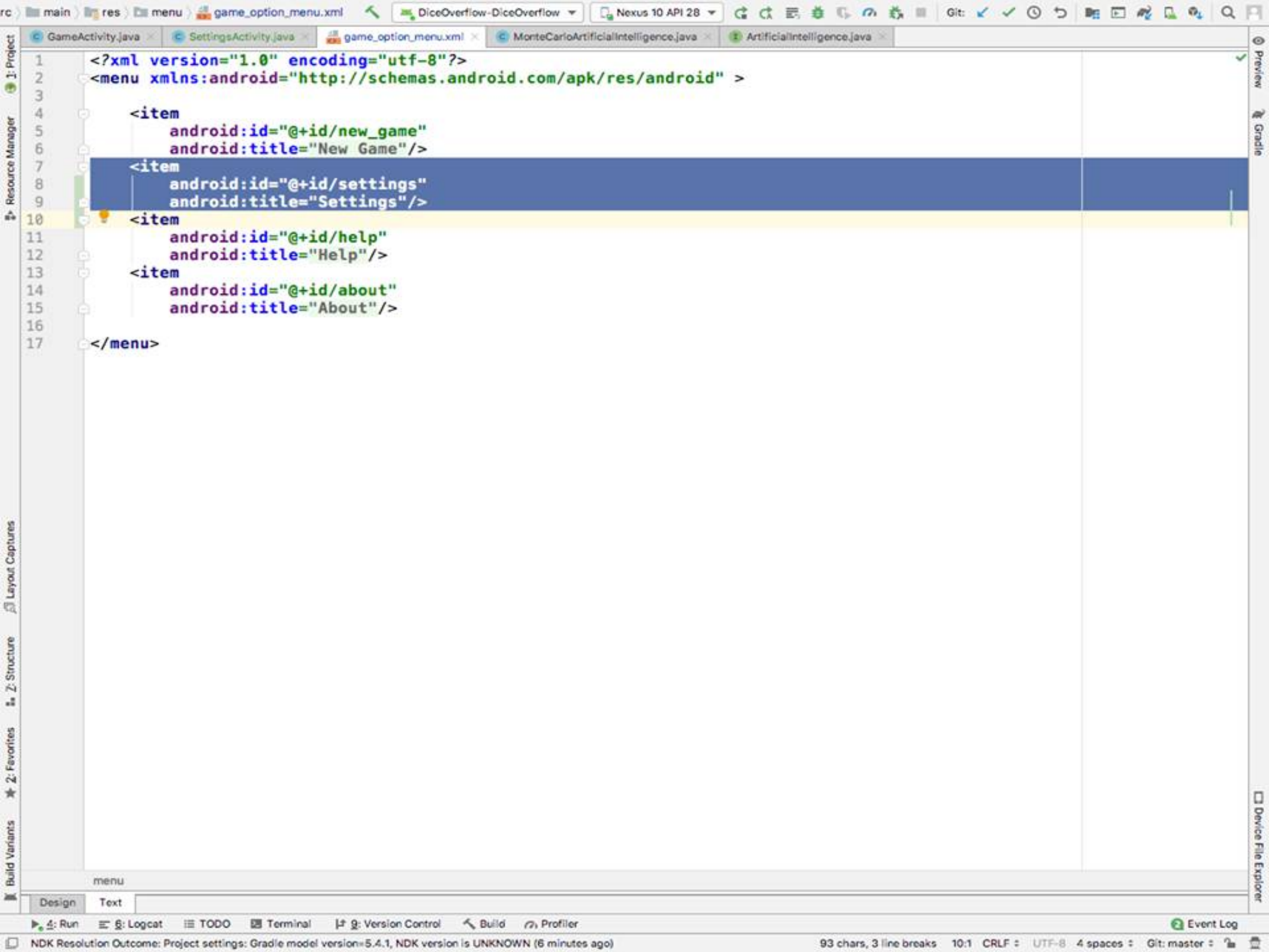
Главният екран е основният игрови екран, докато екрана за опциите е негов под екран. Тъй като програмният език на целия проект е Java, то и за екрана с опциите ще се използва Java.





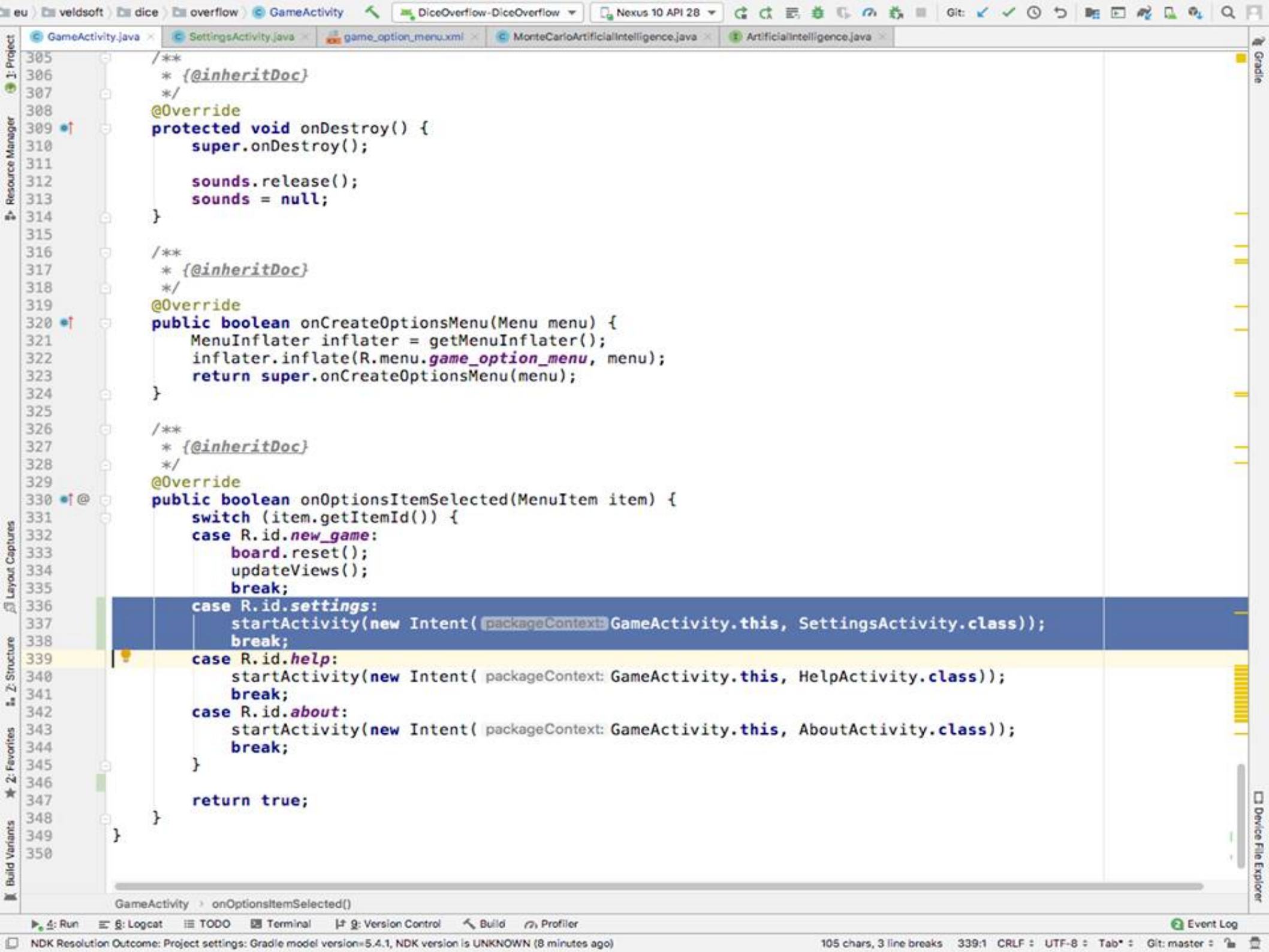
За да се извика екрана с опциите е нужно да се добави допълнителен ред в менюто на основния екран.





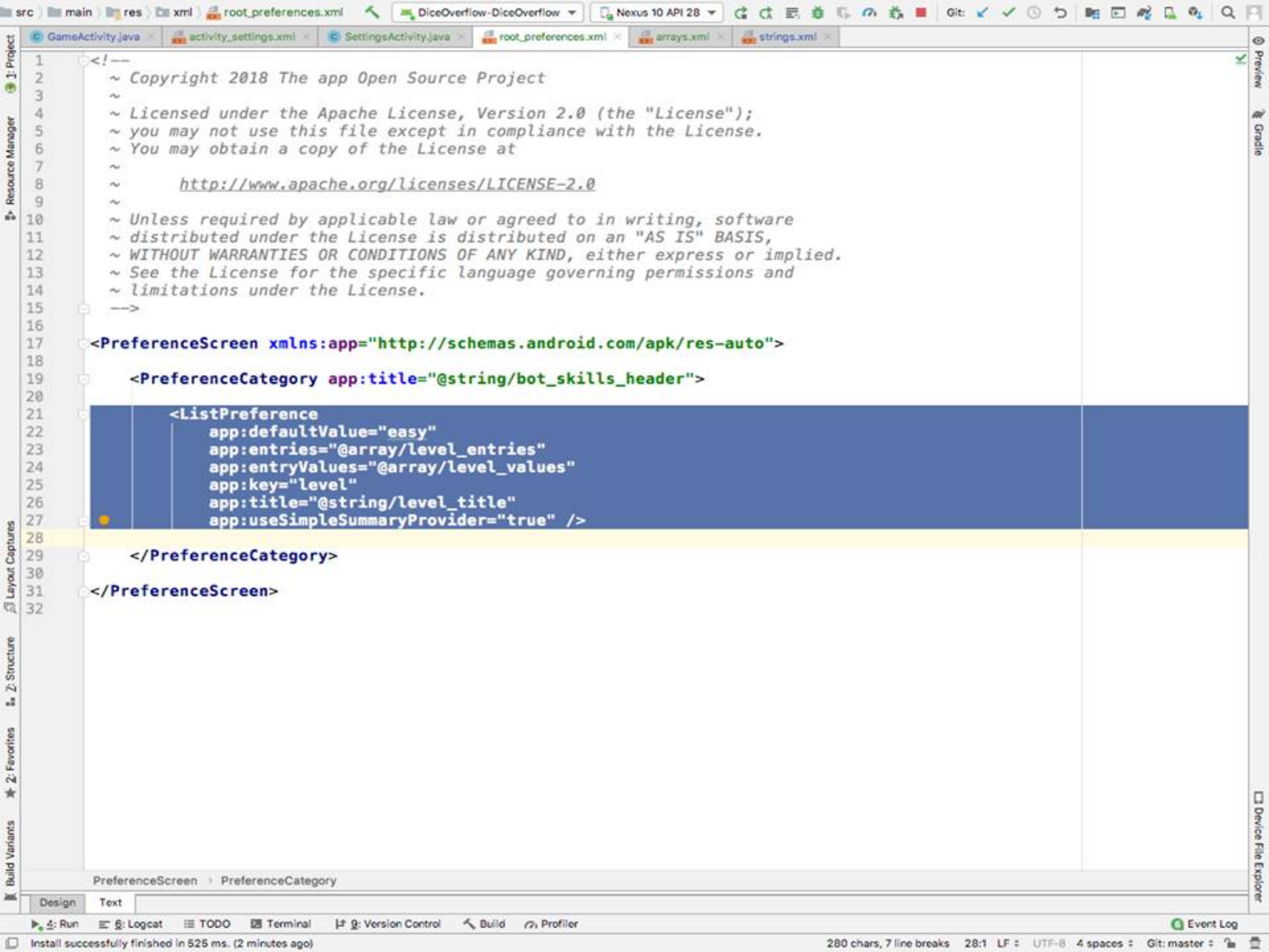
Допълнително е нужен програмен код, който да изпълни самото извикване на екрана с опциите, когато бъде избрана тази възможност от менюто.





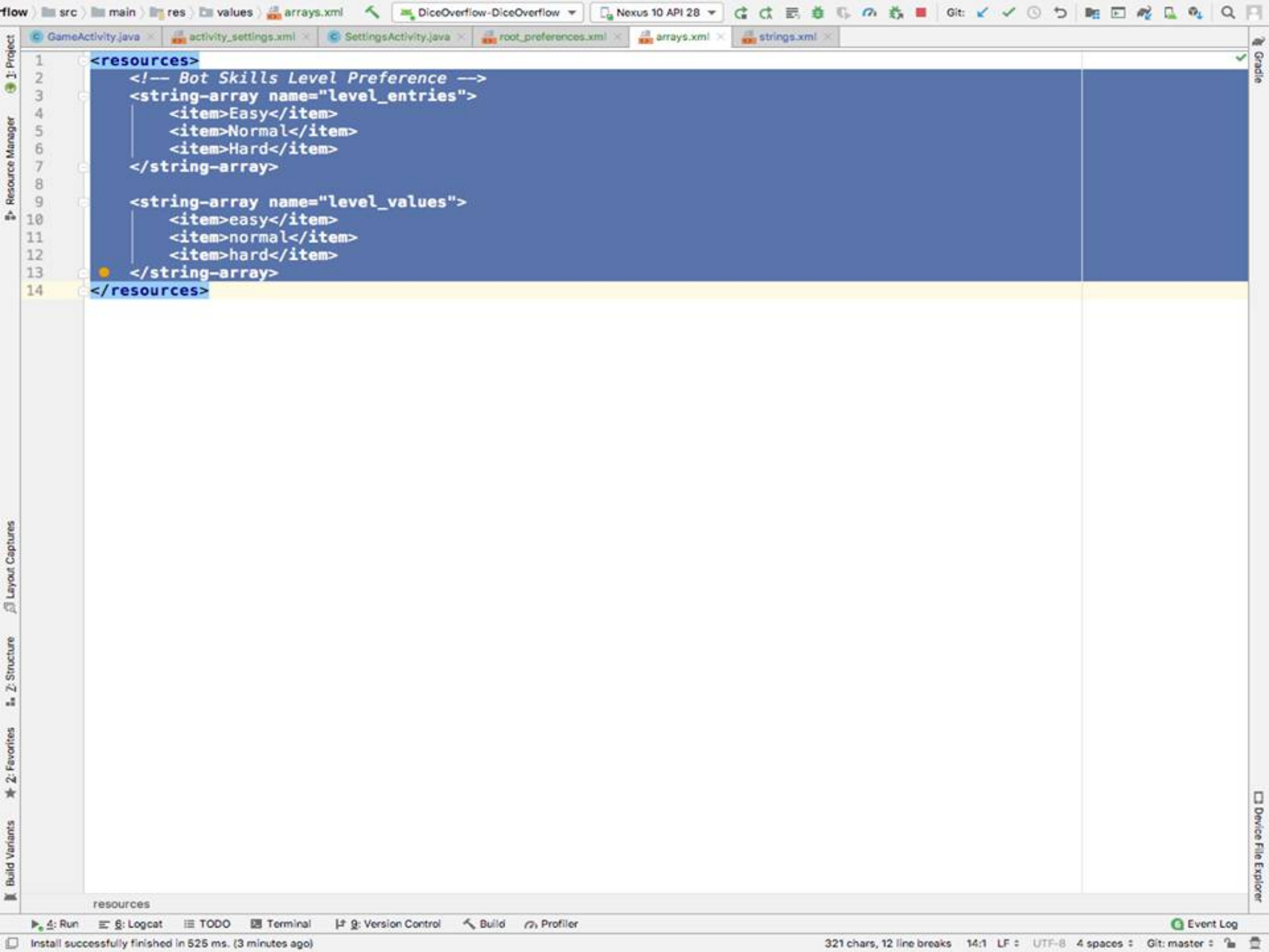
Първоначално екранът с опциите ще съдържа единствено секция за определяне сложността на компютърния опонент.





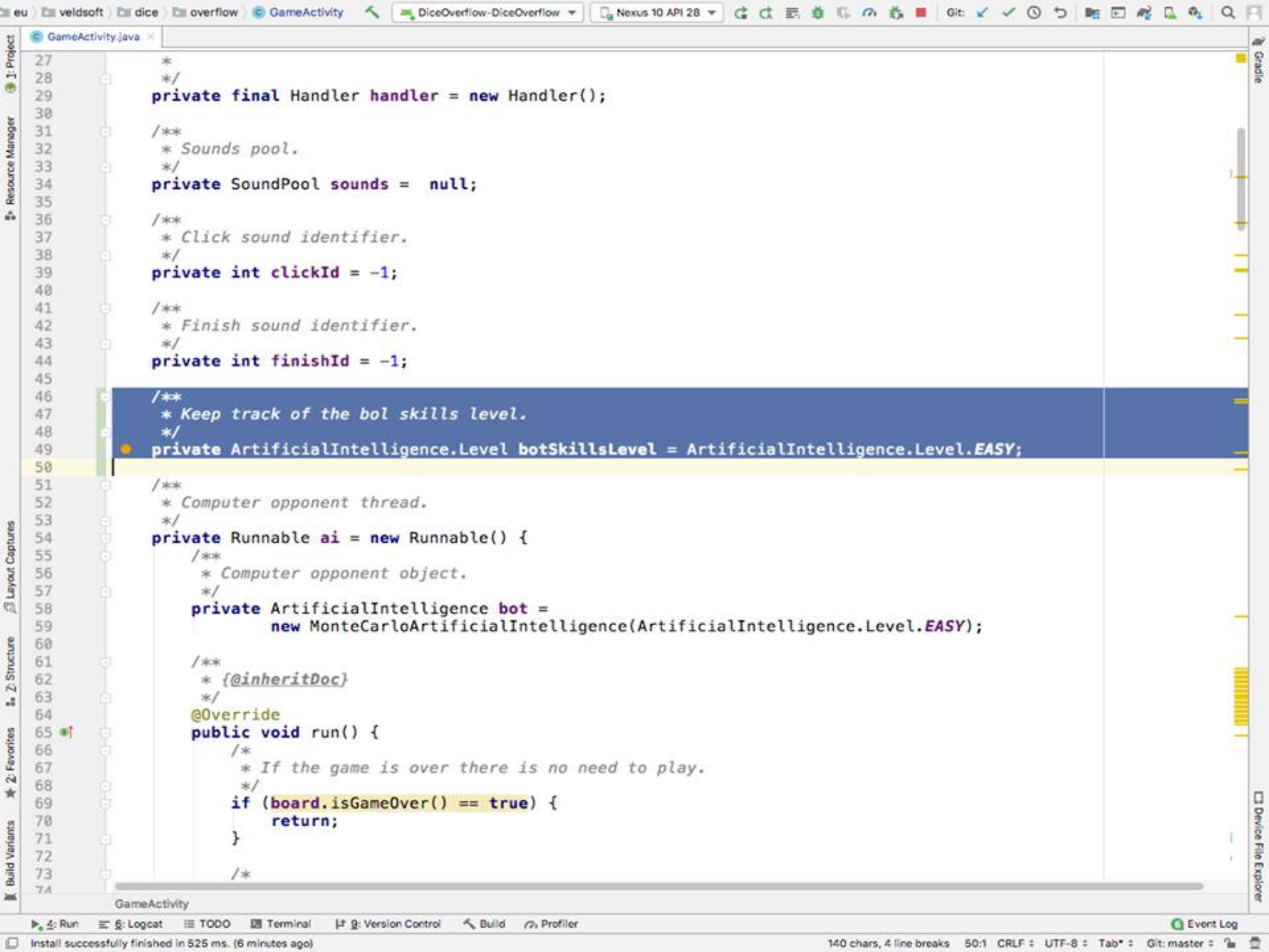
За тази цел се дефинират два масива в ресурсите. Единият масив определя стойностите, а другия масив определя визуализацията на отделните опции.





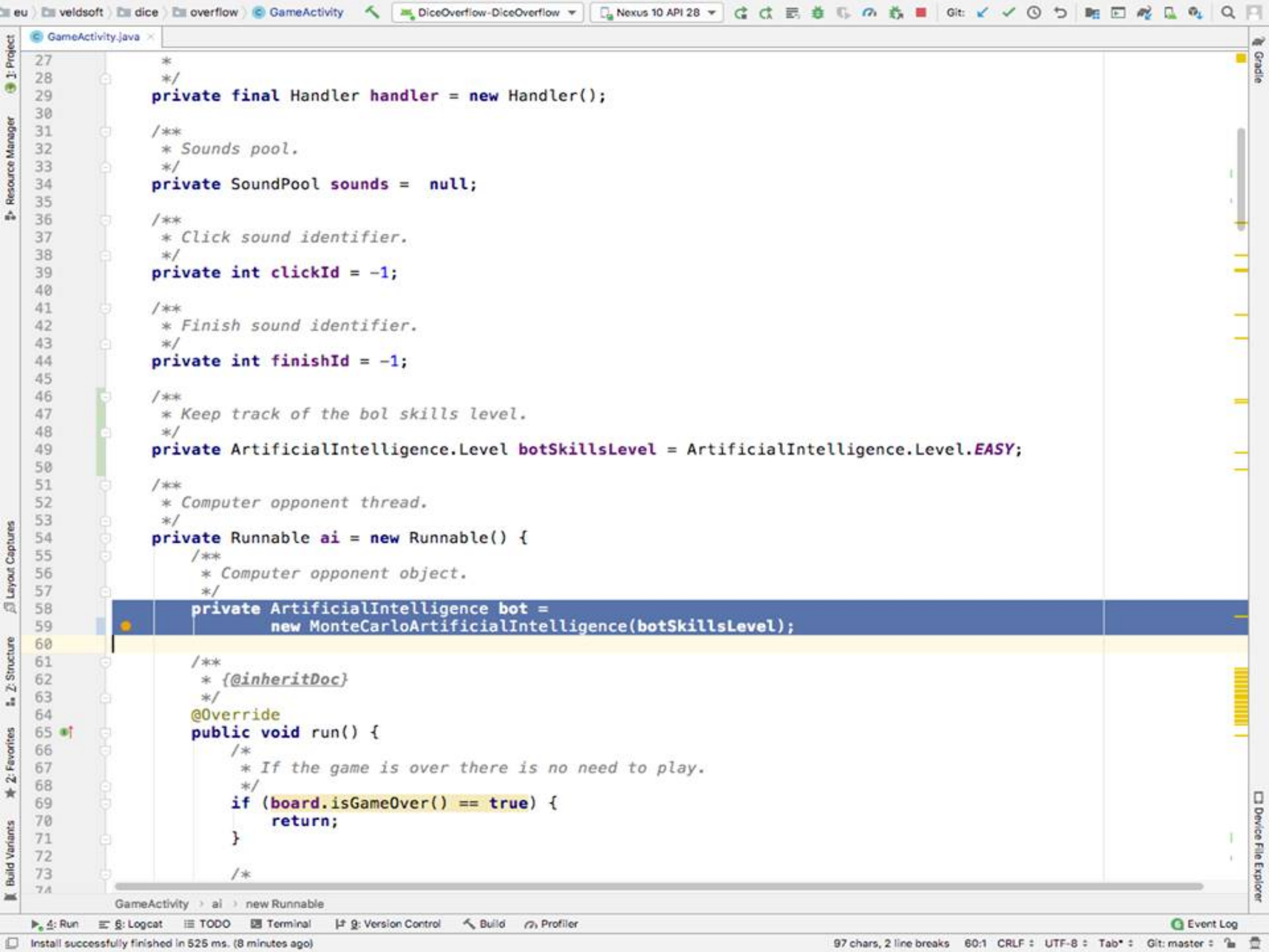
В основния екран се добавя вътрешна променлива, която да следи за промяна в нивото на опитност за компютърния опонент.





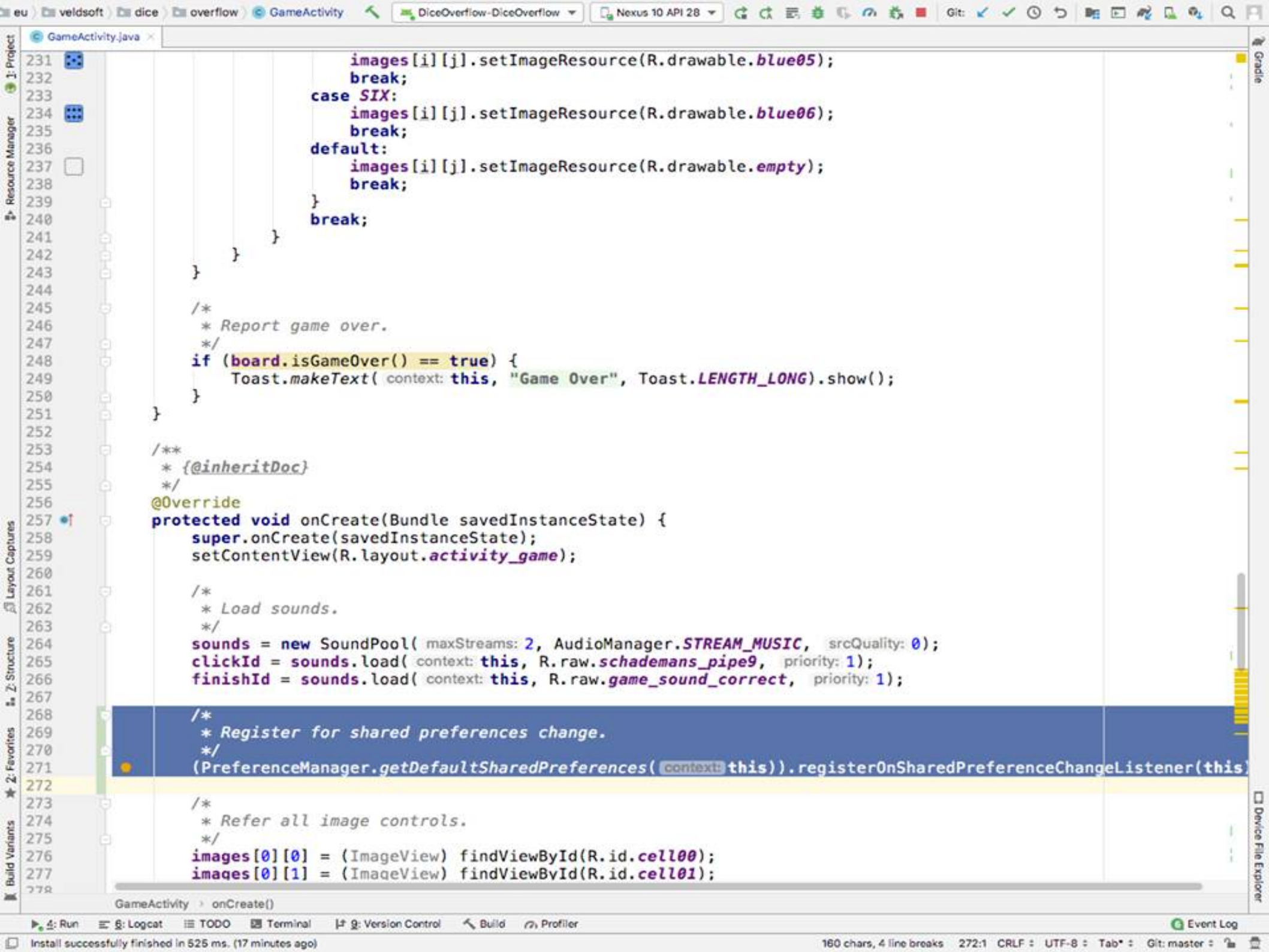
По този начин при инициализация на компютърния опонент винаги ще е налично актуално избраното от потребителя състояние.





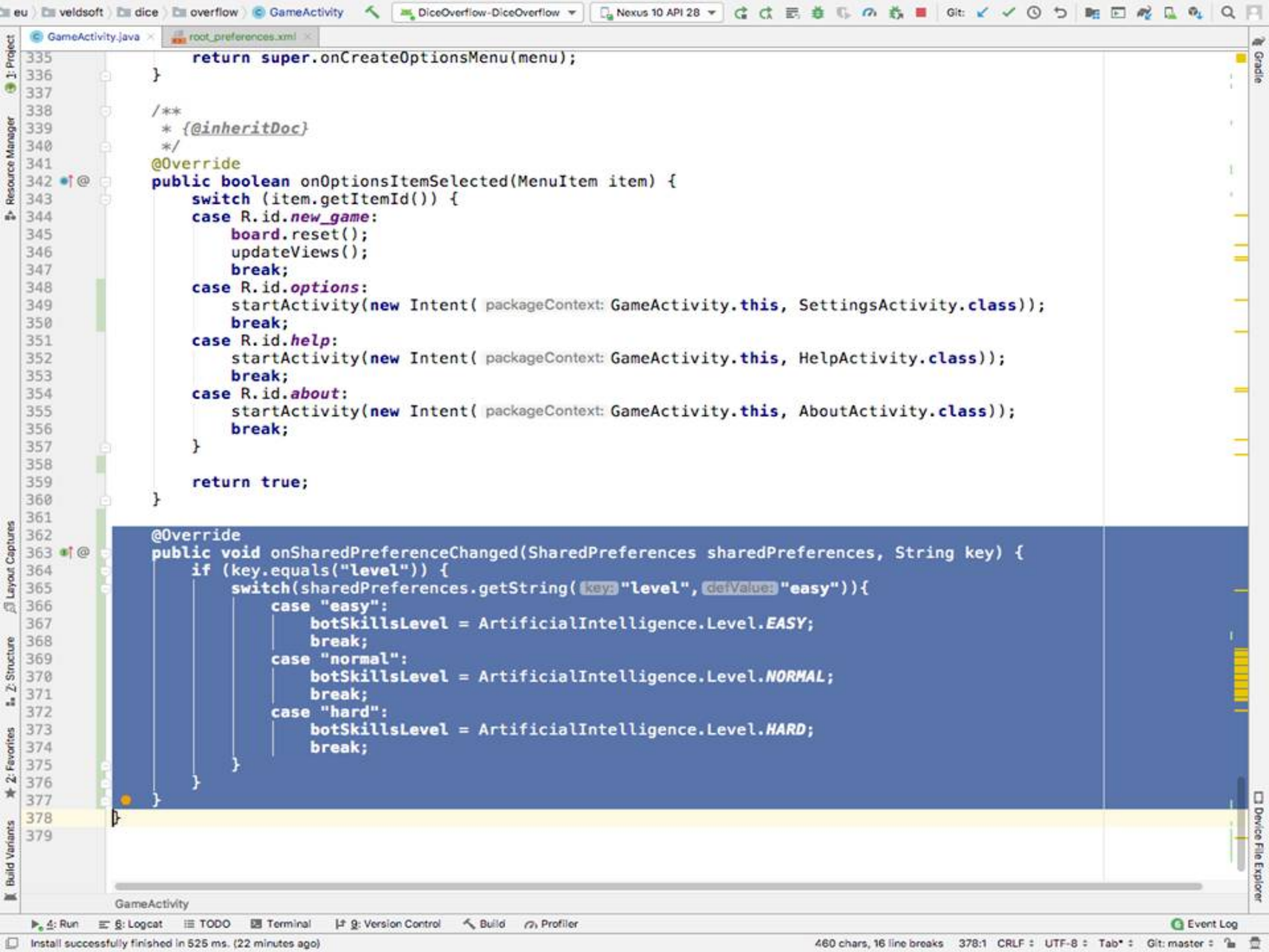
За да се следят промените в опциите текущият екран става слушател на събития за промяна и следва регистрация от мениджъра за промени в опциите.





При всяка промяна на стойностите за сила на компютърния опонент се активира събитие в което се отразява следващата актуална стойност за сложност на компютърния опонент.





Допълнителната опция в менюто се появява под опцията за стартиране на нова игра.



6:57

LTE



Dice Overflow



New Game

Settings

Help

About




От падащо меню се избира нивото за сложност на компютърния опонент.



6:58

LTE

 Settings

Bot Skills Level

Bot Level
Normal

Bot Level

- ☐ Easy
- ☒ Normal
- ☐ Hard

CANCEL

Макар и даващ повече време на компютърния опонент, методът Монте Карло за претърсване на дървото не дава достатъчно добри резултати и компютърният опонент не оказва нужното ниво на съпротива, така че играчът да бъде сериозно затруднен.



7:01



LTE



Dice Overflow



Game Over

Направените промени се добавят в локалното хранилище.




```
MACMINI:Desktop todorbalabanov$ cd DiceOverflow/
MACMINI:DiceOverflow todorbalabanov$ git add .
MACMINI:DiceOverflow todorbalabanov$ git commit -m "Bot skills settings were added."
[master bde5e4d] Bot skills settings were added.
51 files changed, 841 insertions(+), 83 deletions(-)
create mode 100644 .idea/libraries/Gradle__androidx_activity_activity_1_0_0_alpha08_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_annotation_annotation_1_1_0_rc01_jar.xml
create mode 100644 .idea/libraries/Gradle__androidx_appcompat_appcompat_1_1_0_alpha05_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_appcompat_appcompat_resources_1_1_0_alpha05_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_arch_core_core_common_2_1_0_beta01_jar.xml
create mode 100644 .idea/libraries/Gradle__androidx_arch_core_core_runtime_2_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_asynclayoutinflater_asynclayoutinflater_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_collection_collection_1_1_0_rc01_jar.xml
create mode 100644 .idea/libraries/Gradle__androidx_concurrent_concurrent_futures_1_0_0_beta01_jar.xml
create mode 100644 .idea/libraries/Gradle__androidx_coordinatorlayout_coordinatorlayout_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_core_core_1_1_0_beta01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_cursoradapter_cursoradapter_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_customview_customview_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_documentfile_documentfile_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_drawerlayout_drawerlayout_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_fragment_fragment_1_1_0_alpha08_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_interpolator_interpolator_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_legacy_legacy_support_core_ui_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_legacy_legacy_support_core_utils_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_lifecycle_lifecycle_common_2_2_0_alpha01_jar.xml
create mode 100644 .idea/libraries/Gradle__androidx_lifecycle_lifecycle_livedata_2_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_lifecycle_lifecycle_livedata_core_2_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_lifecycle_lifecycle_runtime_2_2_0_alpha01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_lifecycle_lifecycle_viewmodel_2_2_0_alpha01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_loader_loader_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_localbroadcastmanager_localbroadcastmanager_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_preference_preference_1_1_0_alpha05_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_print_print_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_recyclerview_recyclerview_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_savedstate_savedstate_1_0_0_beta01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_slidingpanelayout_slidingpanelayout_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_swiperefreshlayout_swiperefreshlayout_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_vectordrawable_vectordrawable_1_1_0_beta01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_vectordrawable_vectordrawable_animated_1_1_0_beta01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_versionedparcelable_versionedparcelable_1_1_0_beta01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_viewpager_viewpager_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__com_google_guava_listenablefuture_1_0_jar.xml
create mode 100644 DiceOverflow/src/main/java/eu/veldsoft/dice/overflow/SettingsActivity.java
create mode 100644 DiceOverflow/src/main/res/layout/activity_settings.xml
create mode 100644 DiceOverflow/src/main/res/values/arrays.xml
create mode 100644 DiceOverflow/src/main/res/xml/root_preferences.xml
```


След което се изпращат до отдалеченото хранилище.




```

create mode 100644 .idea/libraries/Gradle__androidx_asynclayoutinflater_asynclayoutinflater_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_collection_collection_1_1_0_rc01_jar.xml
create mode 100644 .idea/libraries/Gradle__androidx_concurrent_concurrent_futures_1_0_0_beta01_jar.xml
create mode 100644 .idea/libraries/Gradle__androidx_coordinatorlayout_coordinatorlayout_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_core_core_1_1_0_beta01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_cursoradapter_cursoradapter_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_customview_customview_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_documentfile_documentfile_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_drawerlayout_drawerlayout_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_fragment_fragment_1_1_0_alpha08_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_interpolator_interpolator_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_legacy_legacy_support_core_ui_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_legacy_legacy_support_core_utils_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_lifecycle_lifecycle_common_2_2_0_alpha01_jar.xml
create mode 100644 .idea/libraries/Gradle__androidx_lifecycle_lifecycle_livedata_2_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_lifecycle_lifecycle_livedata_core_2_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_lifecycle_lifecycle_runtime_2_2_0_alpha01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_lifecycle_lifecycle_viewmodel_2_2_0_alpha01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_loader_loader_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_localbroadcastmanager_localbroadcastmanager_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_preference_preference_1_1_0_alpha05_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_print_print_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_recyclerview_recyclerview_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_savedstate_savedstate_1_0_0_beta01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_slidingpanelayout_slidingpanelayout_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_swiperefreshlayout_swiperefreshlayout_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_vectordrawable_vectordrawable_1_1_0_beta01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_vectordrawable_vectordrawable_animated_1_1_0_beta01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_versionedparcelable_versionedparcelable_1_1_0_beta01_aar.xml
create mode 100644 .idea/libraries/Gradle__androidx_viewpager_viewpager_1_0_0_aar.xml
create mode 100644 .idea/libraries/Gradle__com_google_guava_listenablefuture_1_0_jar.xml
create mode 100644 DiceOverflow/src/main/java/eu/veldsoft/dice/overflow/SettingsActivity.java
create mode 100644 DiceOverflow/src/main/res/layout/activity_settings.xml
create mode 100644 DiceOverflow/src/main/res/values/arrays.xml
create mode 100644 DiceOverflow/src/main/res/xml/root_preferences.xml
create mode 100644 gradle.properties

```

MACMINI:DiceOverflow todorbalabanov\$ git push

Counting objects: 69, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (62/62), done.

Writing objects: 100% (69/69), 37.09 KiB | 2.65 MiB/s, done.

Total 69 (delta 44), reused 0 (delta 0)

remote: Resolving deltas: 100% (44/44), completed with 13 local objects.

To https://github.com/VelbazhdSoftwareLLC/DiceOverflow.git

9b12abb..bde5e4d master -> master

MACMINI:DiceOverflow todorbalabanov\$

Антагонистичните игри с открити условия дават множество възможности за реализация на алгоритми от областта на изкуствения интелект. При играта DiceOverflow би било интересно да се експериментира с изкуствени невронни мрежи.





Search or jump to...

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



[VelbazhdSoftwareLLC](#) / [DiceOverflow](#)

[Unwatch](#) 1

[★ Unstar](#) 3

[Fork](#) 1

[Code](#)

[Issues](#) 0

[Pull requests](#) 0

[Actions](#)

[Projects](#) 0

[Wiki](#)

[Security](#)

[Insights](#)

[Settings](#)

Dice Overflow is a simple board logical game developed as master thesis in New Bulgarian University.

[Edit](#)

[board-game](#)

[perfect-information](#)

[Manage topics](#)

[75 commits](#)

[1 branch](#)

[0 packages](#)

[1 release](#)

[1 contributor](#)

[GPL-3.0](#)

Branch: [master](#)

[New pull request](#)

[Create new file](#)

[Upload files](#)

[Find file](#)

[Clone or download](#)



TodorBalabanov Bot skills settings were added.

Latest commit [bde5e4d](#) 2 minutes ago

.idea	Bot skills settings were added.	2 minutes ago
DiceOverflow	Bot skills settings were added.	2 minutes ago
Documentation	Keep documentation in uncompressed files.	14 months ago
gradle/wrapper	Build scripts were updated.	3 days ago
.gitignore	Initial commit	4 years ago
.travis.yml	Level 22 of SDK setup.	14 months ago
Desktop-DiceOverflow.iml	Build scripts were updated.	3 days ago
LICENSE	Documentation was moved.	14 months ago
README.md	Update README.md	last month
build.gradle	Build scripts were updated.	3 days ago
gradle.properties	Bot skills settings were added.	2 minutes ago
gradlew	Migration of Eclipse project to Android Studio project.	14 months ago
gradlew.bat	Migration of Eclipse project to Android Studio project.	14 months ago
settings.gradle	Migration of Eclipse project to Android Studio project.	14 months ago

[README.md](#)



DiceOverflow