Текущую работу реализовала команда «Красивых девочек»

Оглавление

[Роли 2](#_Toc131975076)

[Введение 2](#_Toc131975077)

[Задание 3](#_Toc131975078)

[Обоснование выбора платформы и языка для приложения 3](#_Toc131975079)

[Реализация элементов приложения 5](#_Toc131975080)

[Создание иконки приложения 5](#_Toc131975081)

[Цветовая схема и фоновое изображение 5](#_Toc131975082)

[Начальный экран 6](#_Toc131975083)

[Ввод значений в матрицу 7](#_Toc131975084)

[Вывод результата 9](#_Toc131975085)

[Алгоритмы, применяемые в матричных и биматричных играх 10](#_Toc131975086)

[Инструкция по применению 17](#_Toc131975087)

[Дополнительные элементы 17](#_Toc131975088)

[Тестирование программного продукта 18](#_Toc131975089)

# Роли

(Роли распределялись формально)

Власова Катечка – разработчик-аналитик по Kotlin;

Гуляева Юлечка – разработчик-аналитик по Android Studio;

Пензина Леночка – тестировщик-документатор;

Черкашина Анечка – дизайнер-тестировщик.

# Введение

На первой нашей встрече было проведено обсуждение того, что и как мы хотим реализовать в нашей лабораторной работе. Протокол этого собрания приведен на рисунке 1.

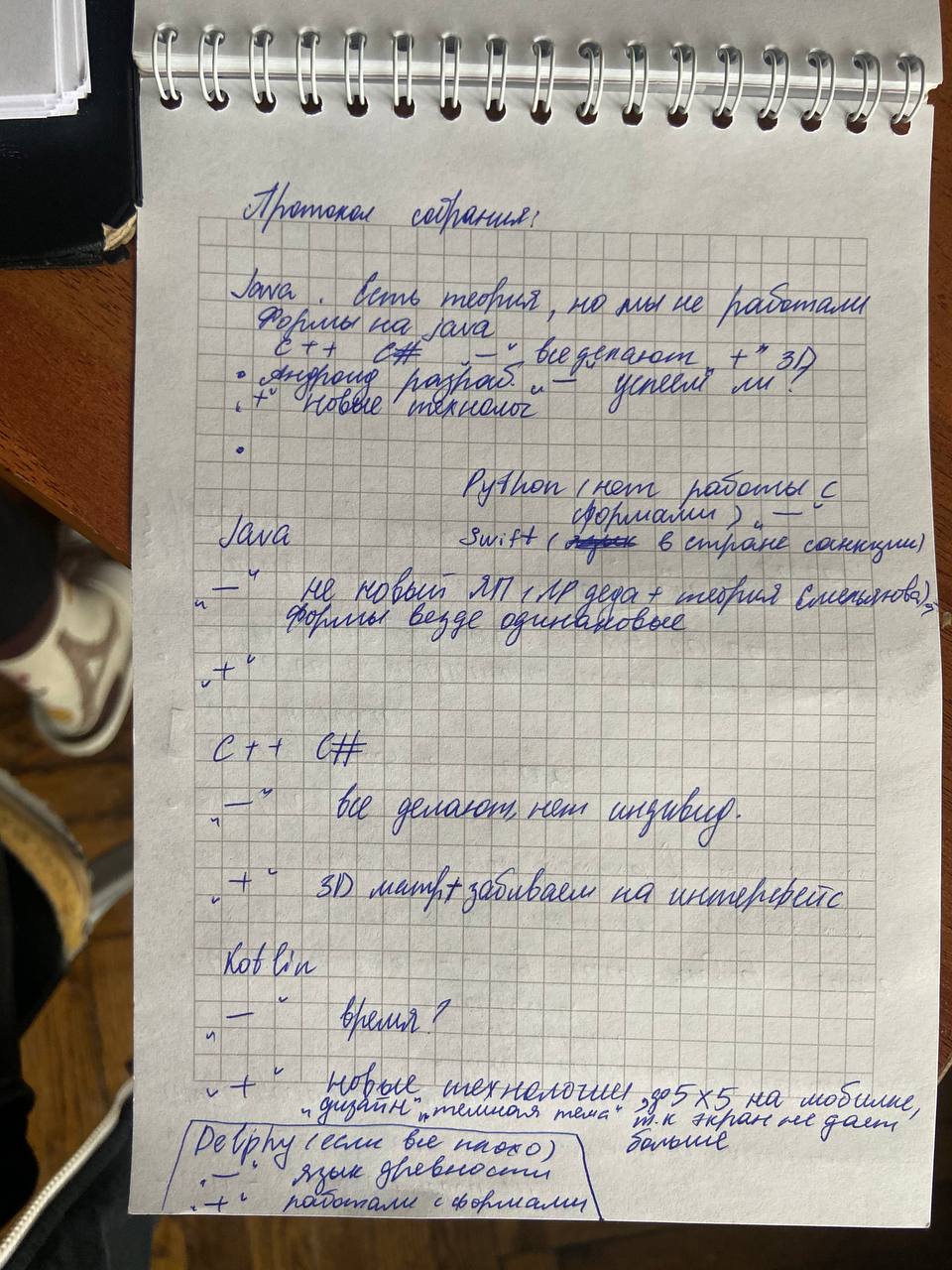


Рисунок 1. Протокол собрания

Дальше в документации может встречаться зачеркнутый текст, что свидетельствует о том, что после анализа масштабов работы, человеческого фактора и времени были реализованы не все запланированные нами пункты.

# Задание

Командная проектная работа по:

1. программной реализации матричной и биматричной теоретико-игровой модели, а именнохранение матриц для дальнейшей их обработки;
2. реализации ввода, вывода и представления матриц;
3. реализация алгоритмов рационализации и поиска решений, а именнообработка введенных матриц.

# Обоснование выбора платформы и языка для приложения

Для реализации поставленной задачи нами была выбрана платформа Android, для работы – среда разработки AndroidStudio, язык программирования –Kotlin.

Для отладки алгоритмов и их тестирования в консольном/файловом режиме использовалась среда IntelliJ IDEA.

Такой выбор был обоснован тем, что:

1. работа с этим языком и средой разработки для нас новая, т.е. изучение новой информации;
2. исходя из предыдущего пункта, нами была проведена работа над сложностями, возникающими во время работы с данной средой. Получен дополнительный опыт, и мы можем выделить плюсы и минусы выбранных средств реализации:

Android Studio:

* удобный редактор кода;
* совмещение режимов Code и Design, т.е. при создании окон приложения можно использовать не только код, но и устанавливать объекты напрямую из меню, что экономит время;
* встроенный эмулятор системы Android, который позволяет проводить тестирование, не загружая приложение на телефон каждый раз;
* занимает много памяти;
* необходимо иметь достаточно внушительную по производительности аппаратную основу ПК, в чем мы убедились на личном опыте, а именно: долгая загрузка первоначальных настроек (на некоторых устройствах это длилось больше часа, а на других довольно быстро).

Kotlin:

* покрывает все возможности Java, получается компактный и легко читаемый код, что удешевляет разработку и поддержку проектов;
* постоянно обновляет базу ошибок и помогает редактировать код до момента исполнения программы;
* низкая скорость срабатывания по сравнению с Java, поскольку Kotlin работает на JVM, что была разработкой для Java, следовательно, скорость компилирования получается низкой;
* малая база примеров применения и решения ошибок в интернете из-за не сильной популярности языка.

IntelliJ IDEA:

* Хорошая поддержка Kotlin;
* Множество инструментов для анализа кода;
* Требовательна к ресурсам, нужно иметь сильную аппаратную основу ПК.

1. приложением, которое получилось в итоге, может пользоваться любой человек, имеющий смартфон с ОС Android и ознакомившийся с инструкцией (и хорошем зрением).

Также нами была выбрана для разработки версия Android 5.0, потому что она совместима с 99.3% устройств на Android, что дает большему количеству людей использовать наше приложение, но во время тестирования на реальных моделях было обнаружено, что в версиях ниже 8.0 возникают системные ошибки, из-за чего приложение рекомендуем использовать пользователям с Android 8.0 и выше.

# Реализация элементов приложения

## Создание иконки приложения

Для создания иконки приложения был использован сайт: https://www.renderforest.com/logo-maker/editor

Там мы выбрали 4 подходящие под тематику нашего приложения иконки и создали их. Наша тематика: теория игр и красивые девочки. Полученные результаты представлены на рисунке 2. Поскольку приложение может использоваться и дальше, то мы отказались от белого фона и сделали его ярче, а именно, розовым, чтобы оно сразу бросалось в глаза.



Рисунок 2. Выбор иконки



Рисунок 3. Итоговая иконка

## Цветовая схема и фоновое изображение

Нами была выбранная пастельная цветовая схема в серо-розовых оттенках для лучшего восприятия человекам текста (и подходящая под название нашей бригады), она представлена на рисунке 2.

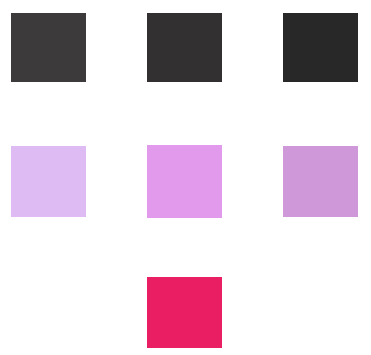


Рисунок 4. Цветовая схема

К выбору фонового изображения мы подошли очень серьезно. Из трех пар изображений мы выбрали ту, которая не вызывает дискомфорта при длительном просмотре, на которой можно четко рассмотреть вводимые/введенные данные и полученный результат, а именно это правая пара на рисунке 3.

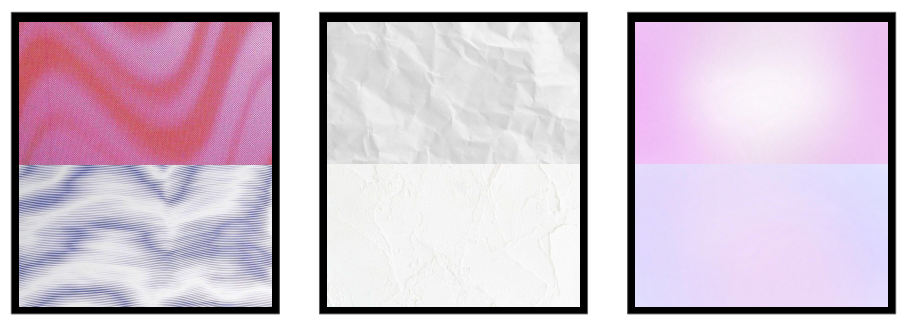


Рисунок 5. Фоновое изображение

## Начальный экран

Приложение запускается в горизонтальном режиме, и имеется функция автоматического поворота экрана на 180 градусов для лучшего видения общей картины. Поворот осуществляется в OnResume() в MainActivity.kt, чтобы при перекрытии приложения и повороте все равно оставался горизонтальный экран (Пы.Сы. опираемся на жизненный цикл активити).

От двух игроков на данном этапе требуется ввести названия своих стратегий через разделитель - запятую «,» (это является обязательным требованием). Количество стратегий не должно превышать 5.

После выбора стратегий необходимо выбрать в каком режиме будут выводиться данные: матричная игра или биматричная игра. Для этого нужно нажать на кнопку ONEMATRIX или TWOMATRIX.

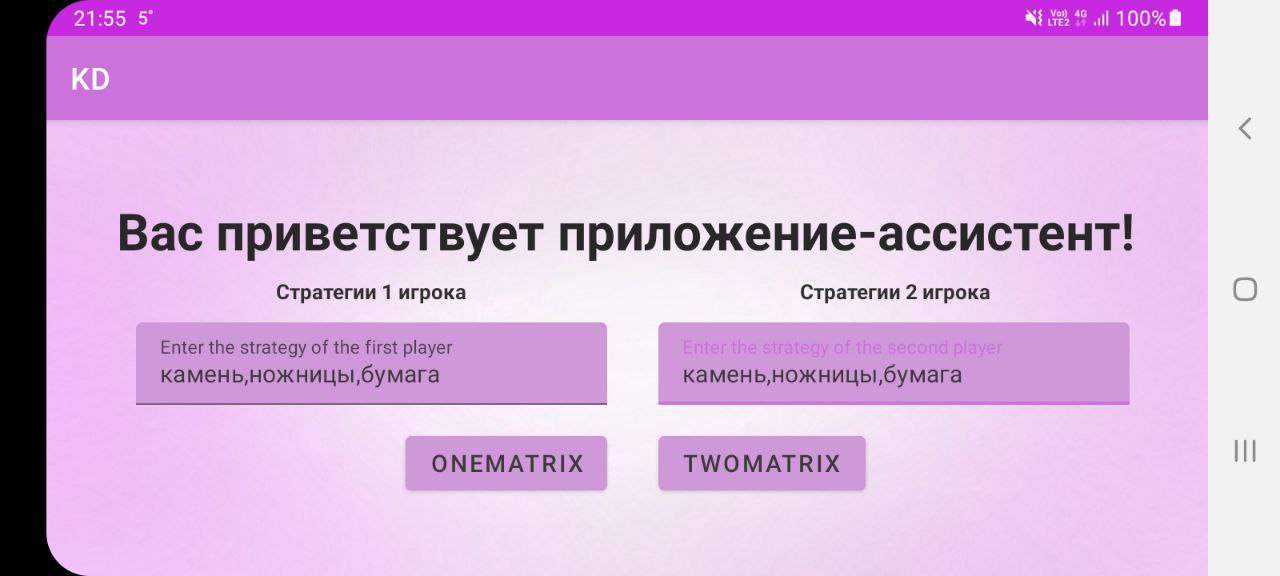


Рисунок 6. Начальный экран

## Ввод значений в матрицу

На втором экране происходит ввод числовых значений в таблицу. Для этого необходимо выбрать один из вариантов:

1. нажать кнопку GENERATE, после чего все поля заполнятся рандомно;
2. игроки записывают значения функций полезности в ячейки (для биматричной игры обязательное условие – ввести через «;»).

Чтобы изменения внеслись, нужно нажать кнопку ADD.

Для хранения функций полезности была задействована матрица, количество строк и столбцов которой может быть от 1 до 5. При этом ввод матрицы 1х1 не поддерживается. Ограниченное количество строк и столбцов в нем объясняется тем, что один экран телефона не может вместить в себя большее количество ячеек для просмотра данных, даже в горизонтальном режиме (если быть точнее – может, но нужно реализовать все так, чтобы мы могли понимать к какой строке и столбцу относится данное значение, поиск и освоение данного функционала требовал больше временных или людских ресурсов). Тут же на втором экране пользователю будет дана возможность выбора алгоритма из выпадающего списка, на который он хочет узнать ответ.

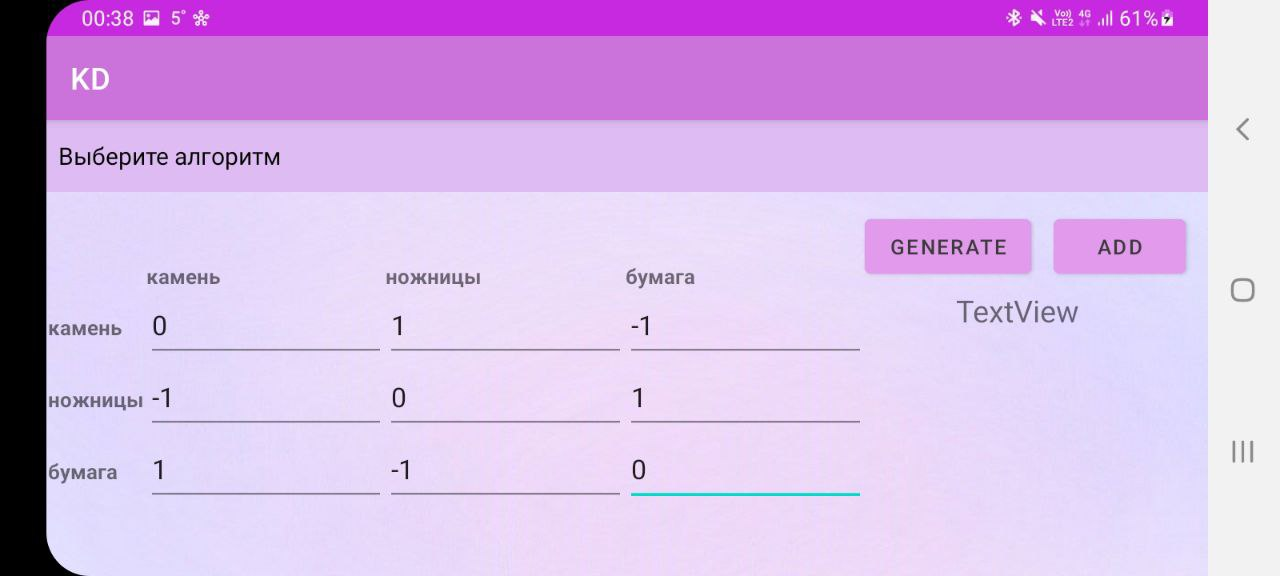


Рисунок 7. Матричная игра с введенными игроками значениями

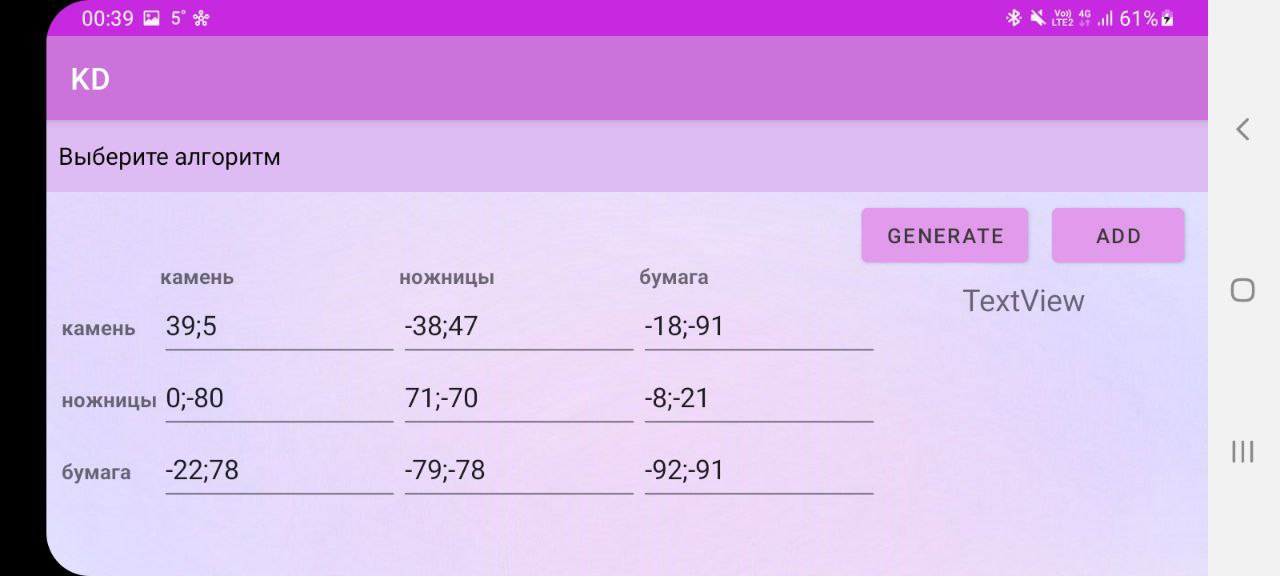


Рисунок 8. Биматричная игра со сгенерированными значениями

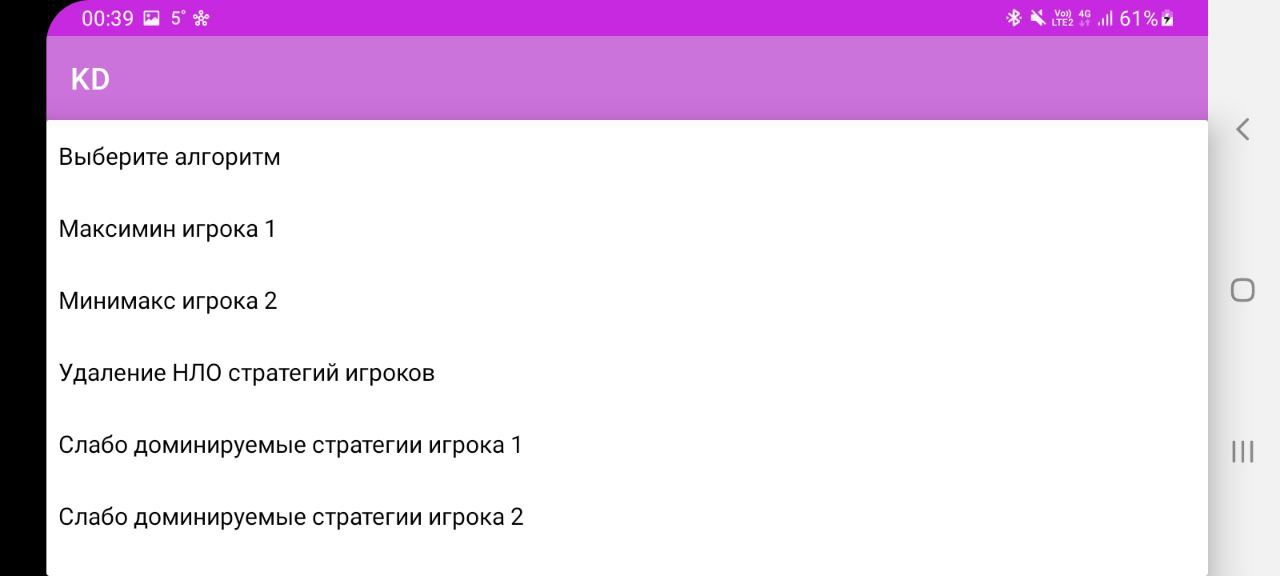


Рисунок 9. Выбор алгоритма

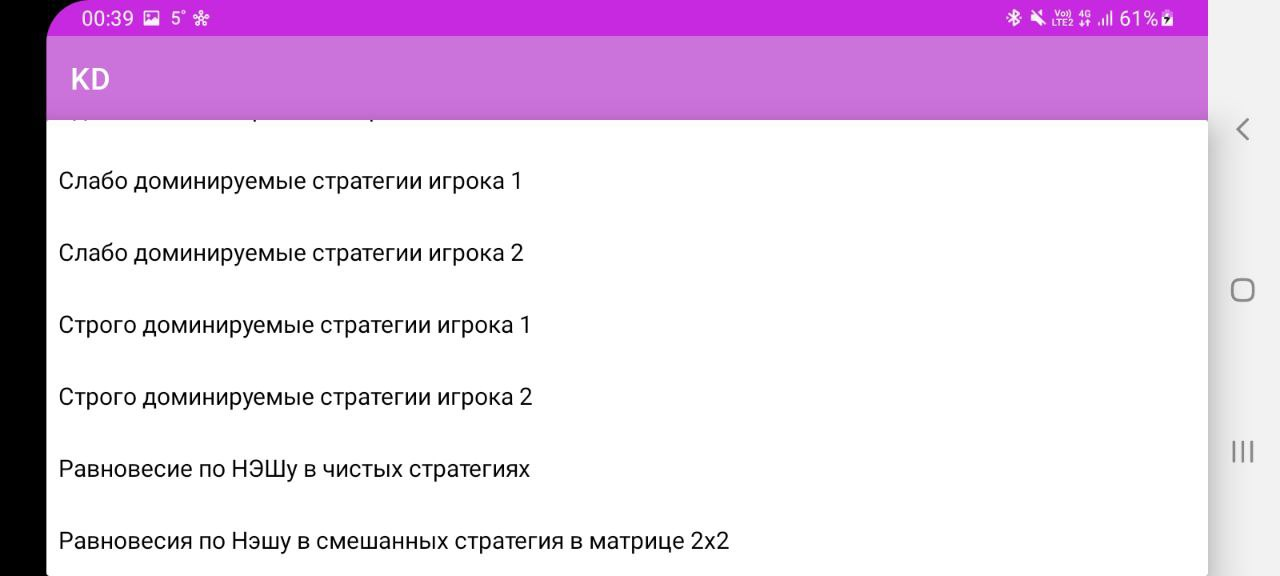


Рисунок 10. Выбор алгоритма

Также была внедрена история, для облегчения ввода пользователем, но она поддерживается только в консольном режиме.

## Вывод результата

Вывод данных будет осуществляться после выбора игроком алгоритма. При нажатии на него, ~~будут подсвечиваться строчки с правильным ответом~~ будет выводиться ответ или ~~измененная матрица после удаления некоторых стратегий~~ после рационализации будут скрываться строки и столбцы. ~~Также предусмотрен вывод объяснения, почему именно это является ответом.~~

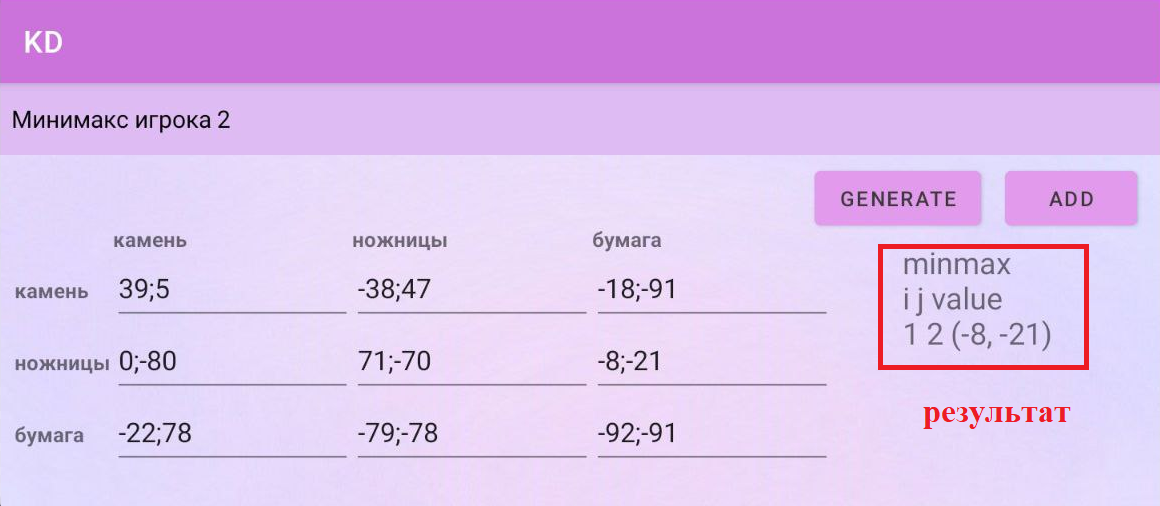


Рисунок 11. Вывод результата



Рисунок 12. Вывод результата и скрытие стратегии

## Алгоритмы, применяемые в матричных и биматричных играх

1. Поиск максимина/минимакса

а) максимин для первого игрока

Проходя в цикле по каждому столбцу, ищем минимум в нем и запоминаем значение. Проходим повторно, чтобы добавить все одинаковые минимумы. Значения заносим в map, ключом является пара из номера строки и столбца. После из полученных минимумов ищем максимум функцией maxof, которую поддерживает map. Проходим по всем элементам map, чтобы найти ключи, значения которых максимально. Функция возвращает массив пар. При выводе также выводим значения по полученным индексам.

б) минимакс для второго игрока

Проходя в цикле по каждой строке, ищем максимум в ней и запоминаем значение. Проходим повторно, чтобы добавить все одинаковые максимумы. Значения заносим в map, ключом является пара из номера строки и столбца. После из полученных максимумов ищем минимум функцией minof, которую поддерживает map. Проходим по всем элементам map, чтобы найти ключи, значения которых минимально. Функция возвращает массив пар. При выводе также выводим значения по полученным индексам.

Пользователь видит ответ для максимина и минимакса в текстовом поле внизу справа, как показано на рисунке 13.

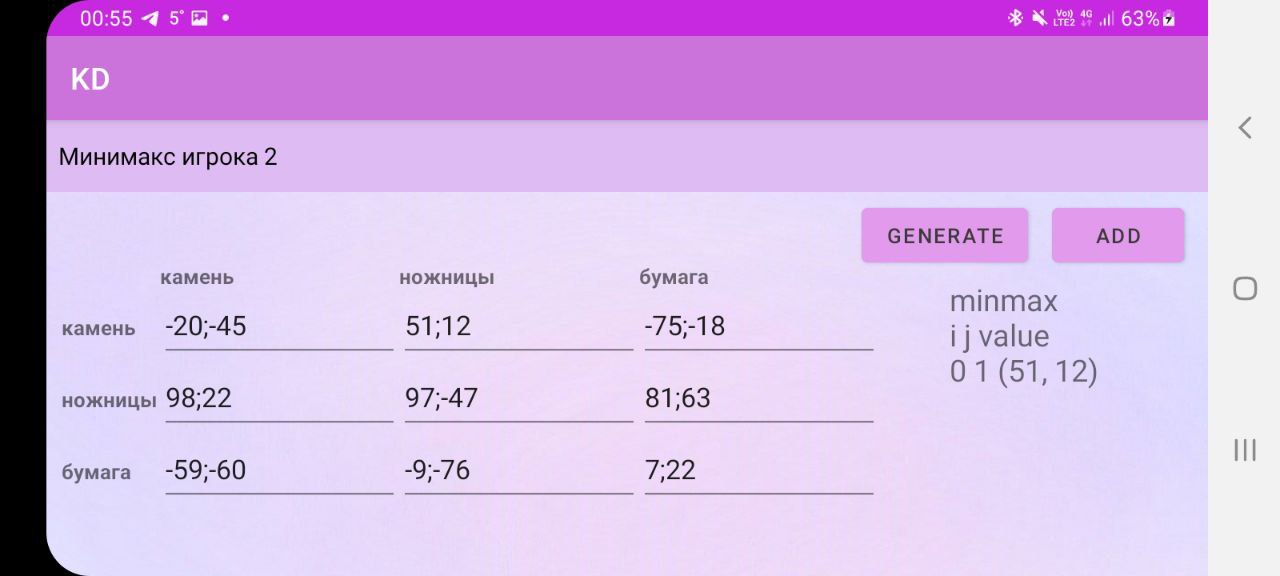


Рисунок 13. Вывод результата

1. Удаление строго доминируемых стратегий

Для первого игрока:

В сравнении участвует полезность первого игрока. Сравниваем стратегии, т.е. сравниваем строки матрицы. Сначала происходит сравнение первой строки с остальными, потом второй строки со всеми, начиная с третьей и т.д.. Сравнение строк происходит в массиве, изначально введены два флага, при равенстве значений в столбце сбрасываются оба флага; если значение полезности меньшего по индексу больше, то она не является доминируемой и сбрасывается флаг, отвечающий за это; аналогично для полезности меньшего по индексу.

Сравнив две строки, проверяем значения флагов. Если значение истинно, то добавляем стратегию, за доминируемость которой отвечал этот флаг.

Для второго игрока:

В сравнении участвует полезность первого игрока. Сравниваем стратегии, т.е. сравниваем столбцы матрицы. Сначала происходит сравнение первого столбца с остальными, потом второго столбца со всеми, начиная с третьего и т.д.. Сравнение столбцов происходит в массиве, изначально введены два флага, при равенстве значений в строке сбрасываются оба флага; если значение полезности меньшего по индексу больше, то она не является доминируемой и сбрасывается флаг, отвечающий за это; аналогично для полезности меньшего по индексу.

Сравнив два столбца, проверяем значения флагов. Если значение истинно, то добавляем стратегию, за доминируемость которой отвечал этот флаг.

Вывод результата для пользователя происходит в текстовое поле справа внизу, после пользователь нажимает на кнопку DELETE для удаления стратегий, после строчка/столбец со стратегией скрывается из поля видимости, как показано на рисунке 14.

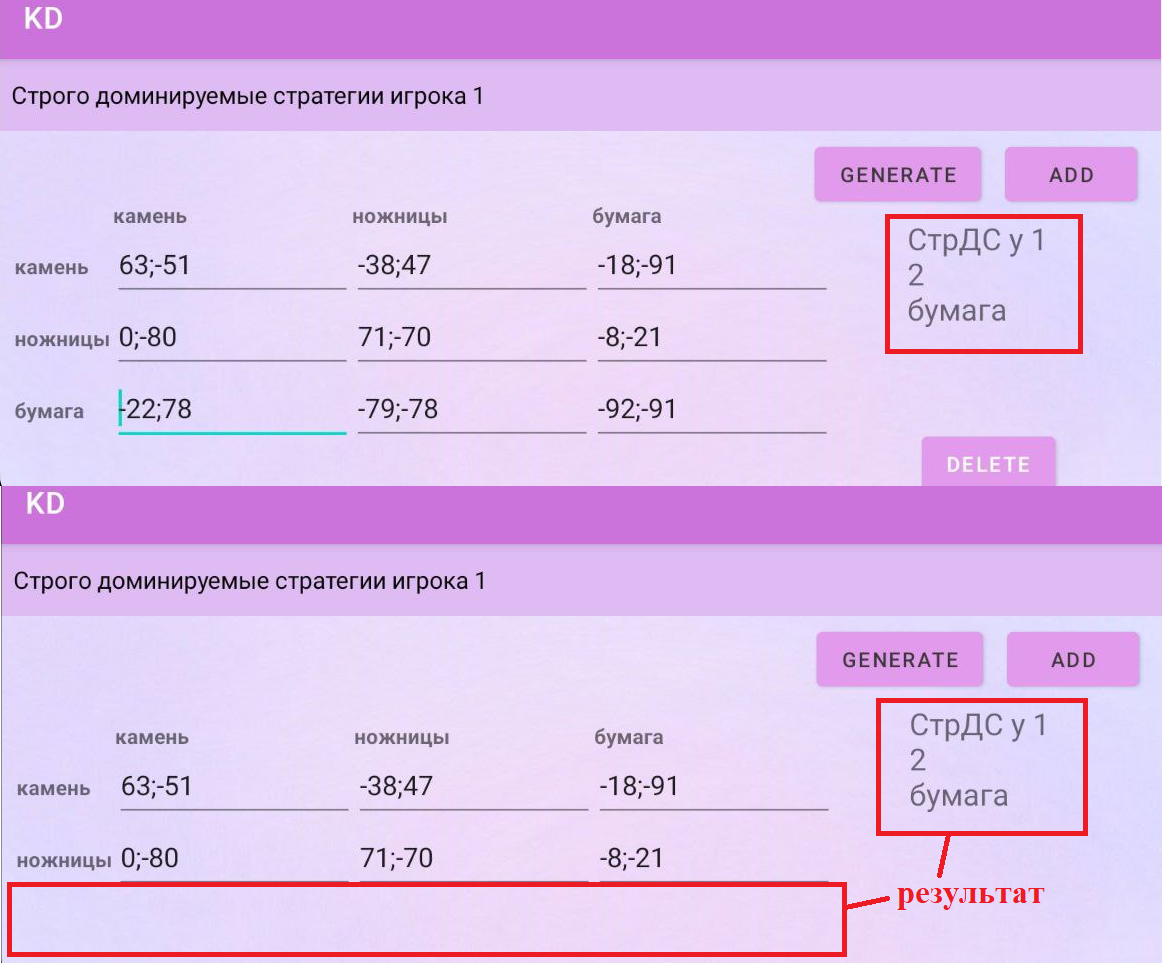


Рисунок 14. Вывод результата

1. Удаление слабо доминируемых стратегий

Для первого игрока:

В сравнении участвует полезность первого игрока. Сравниваем стратегии, т.е. сравниваем строки матрицы. Сначала происходит сравнение первой строки с остальными, потом второй строки со всеми, начиная с третьей и т.д.. Сравнение строк происходит в массиве, изначально введены три флага, при равенстве значений в столбце поднимается флаг, отвечающий за равенство хотя бы одного; если значение полезности меньшего по индексу больше, то она не является доминируемой и сбрасывается флаг, отвечающий за это; аналогично для полезности меньшего по индексу.

Сравнив две строки, проверяем значения флагов. Если ни одно из значений попарно не равно, ничего не будет добавлено. Далее проверяется случай, когда стратегии идентичны, в таком случае ничего не добавляется. Если значение истинно, то добавляем стратегию, за доминируемость которой отвечал этот флаг.

Для второго игрока:

В сравнении участвует полезность первого игрока. Сравниваем стратегии, т.е. сравниваем столбцы матрицы. Сначала происходит сравнение первого стролбца с остальными, потом второго стролбца со всеми, начиная с третьего и т.д.. Сравнение столбцов происходит в массиве, изначально введены три флага, при равенстве значений в строке поднимается флаг, отвечающий за равенство хотя бы одного; если значение полезности меньшего по индексу больше, то она не является доминируемой и сбрасывается флаг, отвечающий за это; аналогично для полезности меньшего по индексу.

Сравнив два столбца, проверяем значения флагов. Если ни одно из значений попарно не равно, ничего не будет добавлено. Далее проверяется случай, когда стратегии идентичны, в таком случае ничего не добавляется. Если значение истинно, то добавляем стратегию, за доминируемость которой отвечал этот флаг.

Для пользователя результирующая стратегия выводится в текстовом поле внизу справа, также он имеет право выбрать, какую стратегию удалить (если их несколько), и нажать на кнопку удалить для скрытия этой стратегии. Результат приведен на рисунке 15 и рисунке 16.

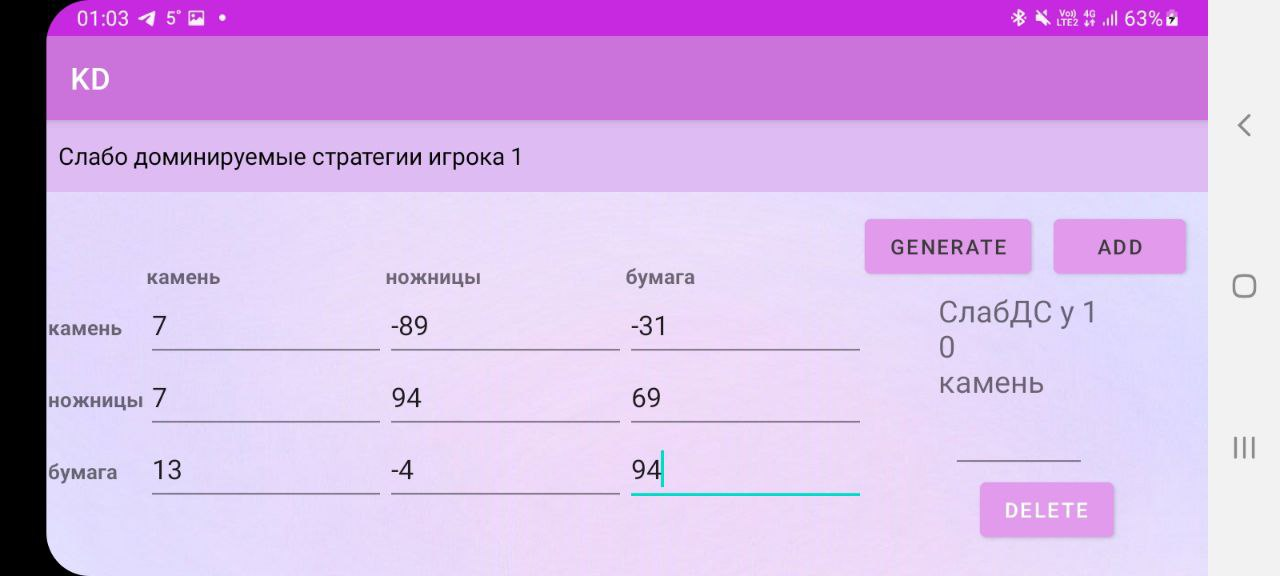


Рисунок 15. Вывод результата

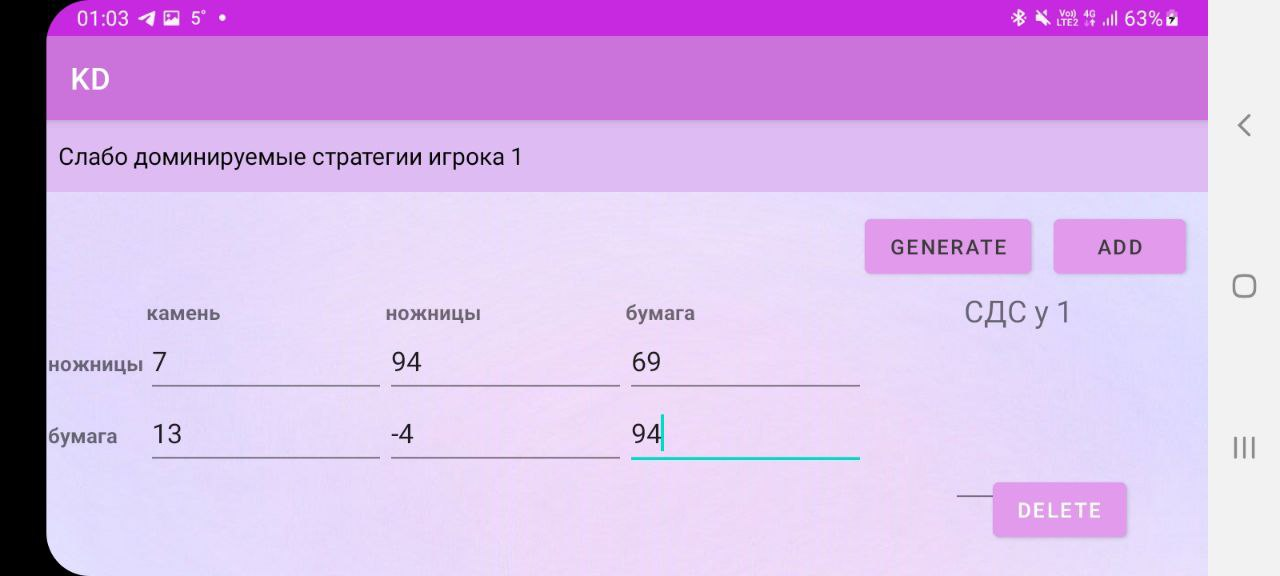


Рисунок 16. Вывод результата

1. Удаление НЛО стратегий

Для первого игрока:

Сначала ищем наилучший ответ в каждом столбце. Данная стратегия заносится во множество. По итогу проверки стратегии, которые не вошли в это множество, являются никогда не лучшим ответом.

Для второго игрока:

Сначала ищем наилучший ответ в каждой строке. Данная стратегия заносится во множество. По итогу проверки стратегии, которые не вошли в это множество, являются никогда не лучшим ответом.

Для пользователя результат выводится в текстовое поле в правом нижнем углу, после нажатия кнопки DELETE скрываются стратегии, которые являются НЛО. Результат показан на рисунках 17 и 18.

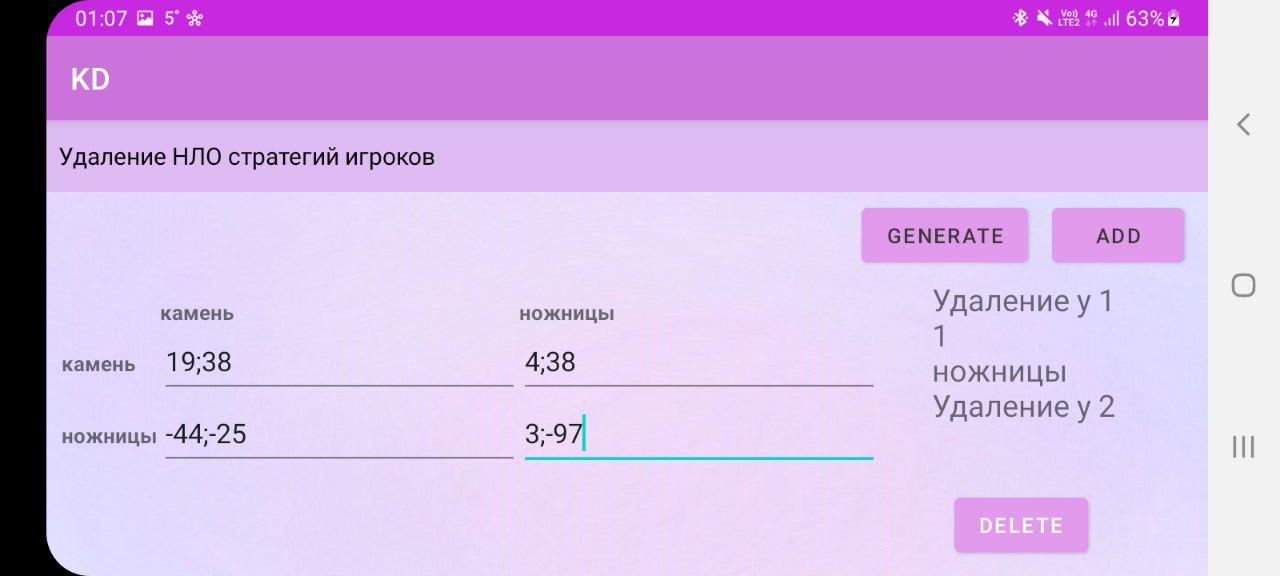


Рисунок 17. Вывод результата

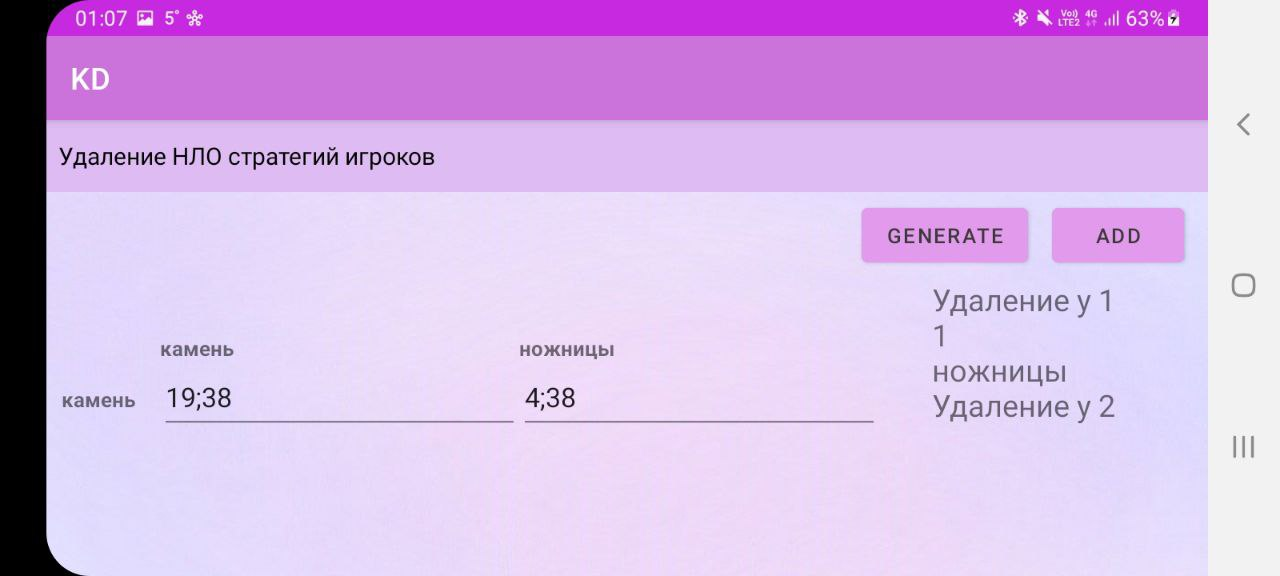


Рисунок 18. Вывод результата

1. Поиск равновесий по Нэшу в чистых стратегиях

Ищутся стратегии, которые являются наилучшим ответом для первого игрока и для второго. Ответом является пересечение множеств.

Для пользователя результат выводится в текстовое окно в правом нижнем углу, как показано на рисунке 19.

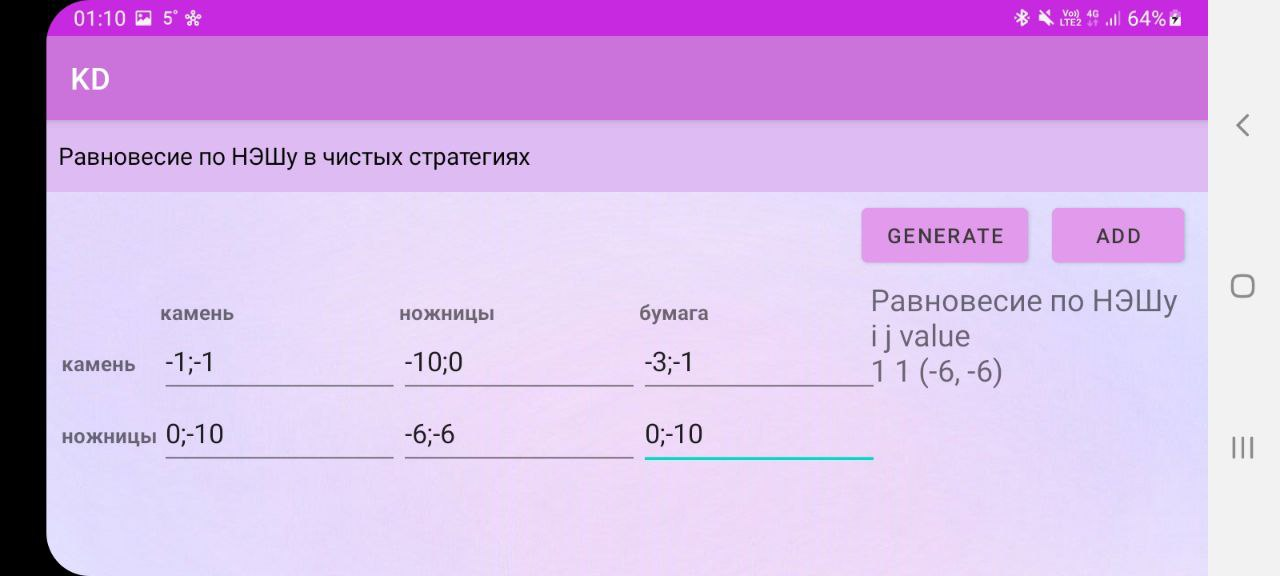


Рисунок 19. Вывод результата

1. Поиск равновесия по Нэшу в смешанных стратегия в матрице 2х2

Сначала ищется равновесие Нэша в чистых стратегиях. Проверяется их количество. При наличии 4-ех равновесий Нэша. Смешанным равновесием Нэша является любое распределение вероятностей, поэтому мы выводим частное значение P=(0.5; 0.5), Q=(0.5; 0.5). Таким же образом находится одно равновесие в случае 3-ех равновесий Нэша в чистых стратегиях. В случае 2-ух, если они находятся рядом друг с другом в одном значении комбинация нуля и единицы, в другом две половины. Если располагаются по диагонали, то вычисляется через приравнивание полезности первого игрока для первой стратегии и для второй стратегии. По уравнению вычисляется Q. Через приравнивание полезности второго игрока, вычисляется P. Если одно равновесие Нэша в чистых стратегиях, то P и Q для этих стратегий будет равно единице, а для оставшихся нулю.

Для пользователя результат выводится в текстовое окно в правом нижнем углу, как показано на рисунке 20.

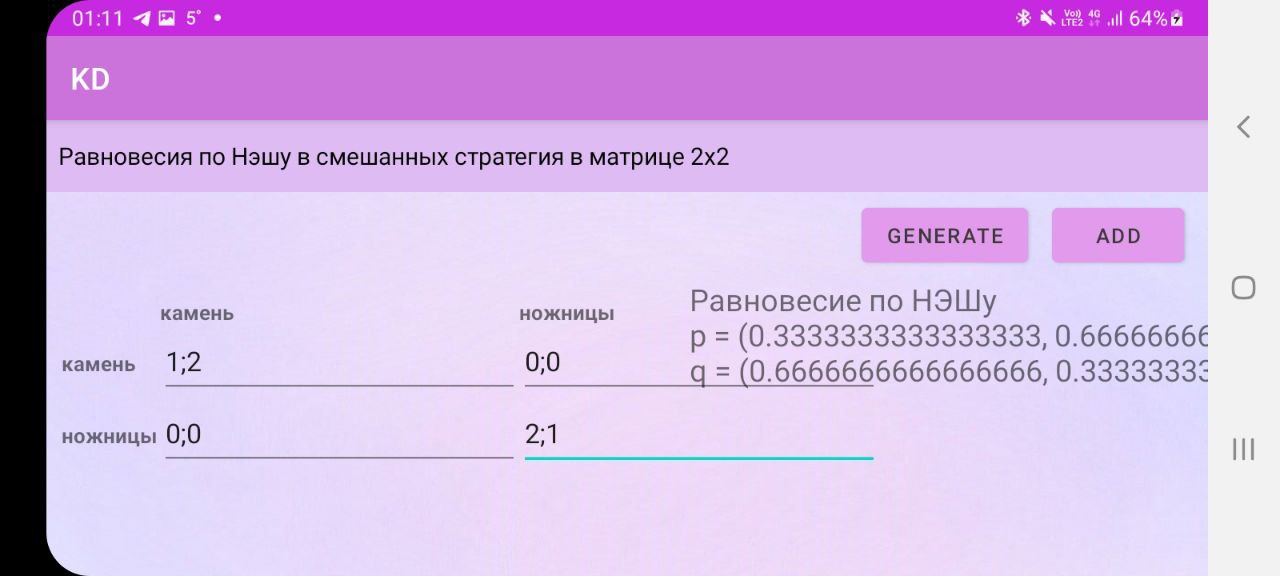


Рисунок 20. Вывод результата

# Инструкция по применению

На экране приветствия два игрока вводят для себя названия стратегий обязательно через запятую «,».

Далее им необходимо выбрать матричную (кнопка ONEMATRIX) или биматричную игру (кнопка TWOMATRIX), что будет влиять на вид вводимых данных в матрице: одно число - для матричной и два числа через точку с запятой «;» - для биматричной.

После нажатия соответствующей кнопки, появится второй экран с таблицей нужного размера, куда необходимо вписать значения функции полезности для каждого игрока. Либо с помощью кнопки GENERATE сгенерировать рандомную матрицу. После этого необходимо применить изменения, нажав на кнопку ADD.

Далее игроку следует выбрать из выпадающего списка сверху алгоритм, на который он хочет получить ответ, и нажать на него.

Ознакомиться с выданным результатом.

~~С помощью истории вернуться к нужной матрице или~~ ввести данные заново.

# Дополнительные элементы

Для облегчения тестирования и отладки нами были сделаны версии проекта для ввода/вывода в консольном режиме и ввод/вывод из файла, помимо графического режима. Тестирование проводилось во всех режимах, при личной встрече мы сможем продемонстрировать это.

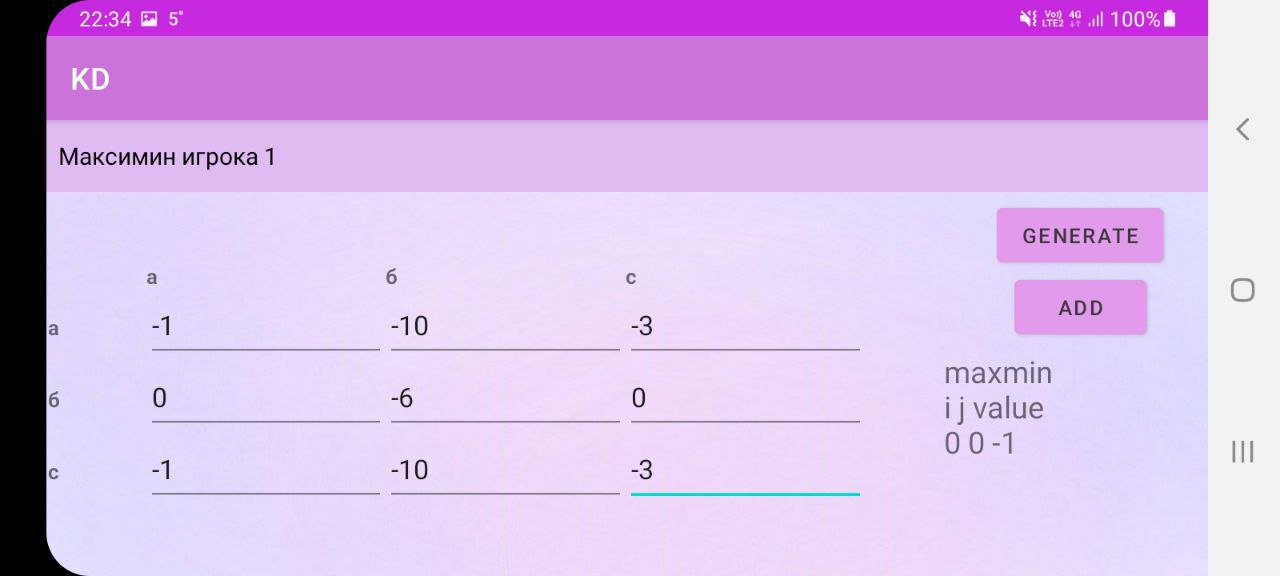
Также в консольном/файловом режиме существует возможность возвращения к матрице путем ее запоминания и последующего вывода (внедрена история).

# Тестирование программного продукта

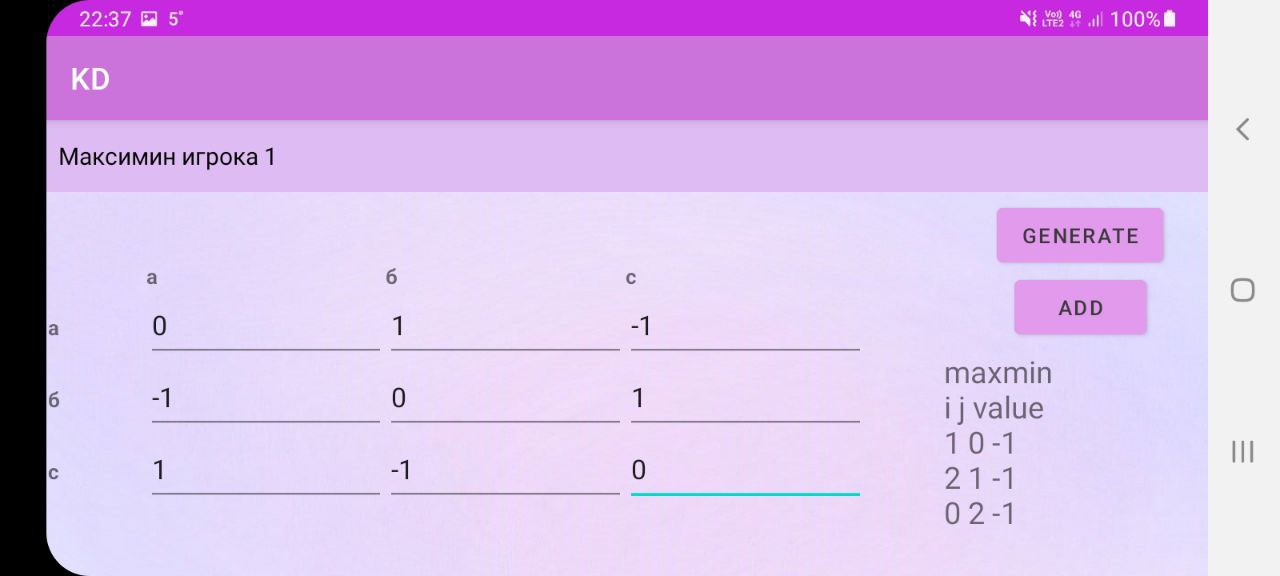
**Поиск максимина:**

Матричная игра:

Когда есть один максимин



Когда есть несколько максиминов

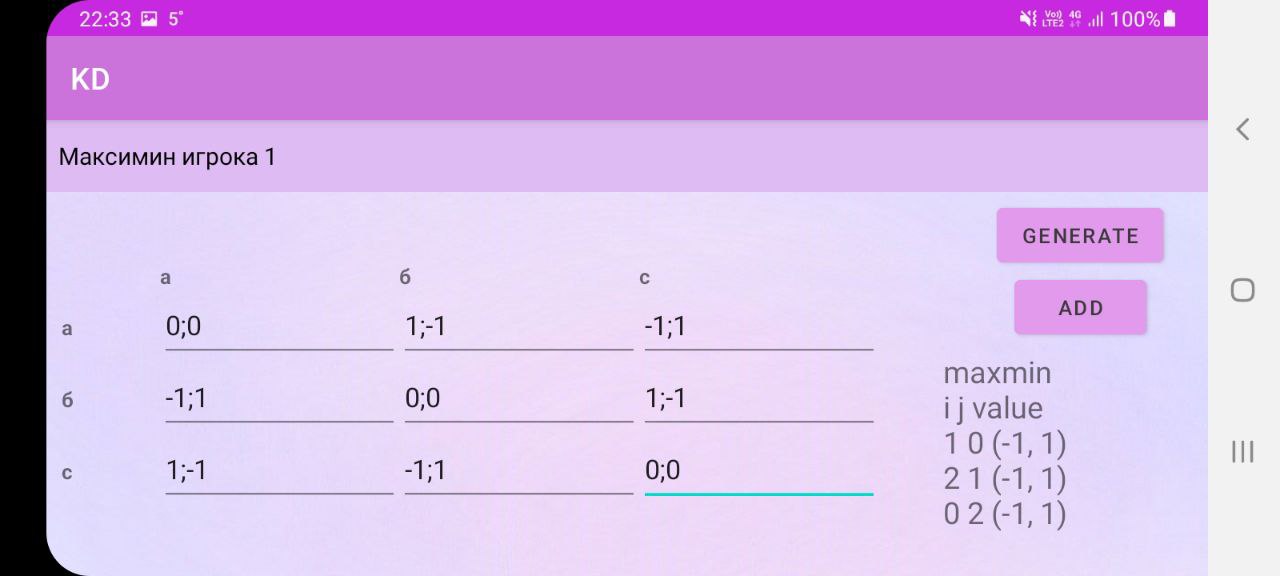


Биматричная игра:

Когда есть один максимин (выводится значение целой ячейки, берется в ответ только первая цифра)



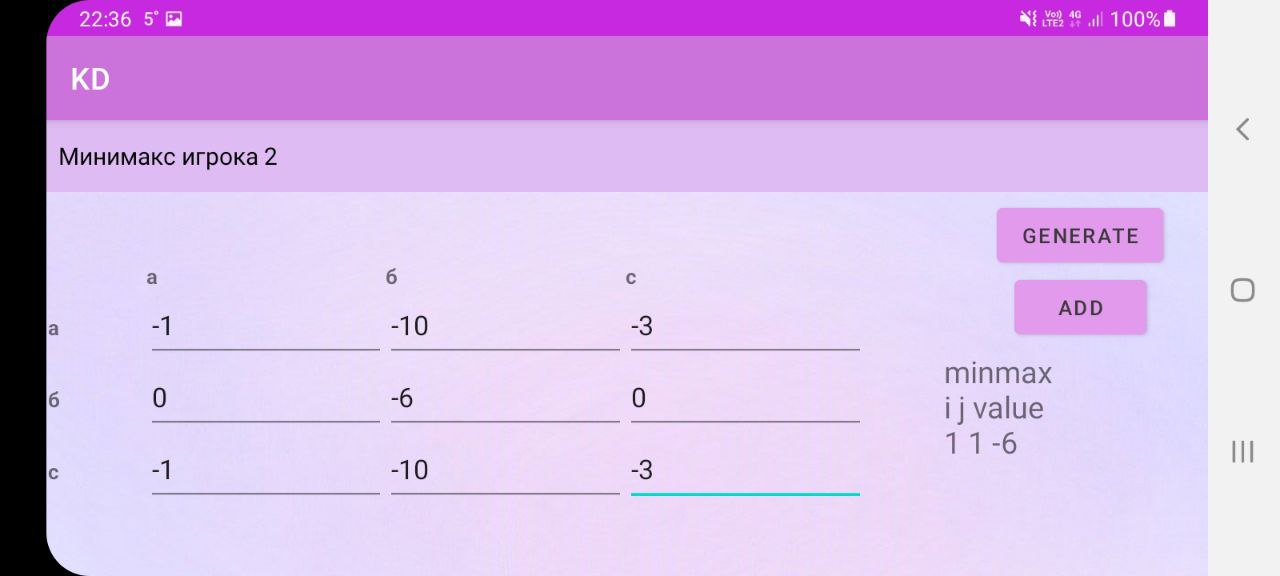
Когда есть несколько максиминов (выводится значение целой ячейки, берется в ответ только первая цифра)



**Поиск минимакса:**

Матричная игра:

Когда есть один минимакс



Когда есть несколько минимаксов (значение выводится по матрице домноженной на -1, поэтому результат будет 1 1 1)

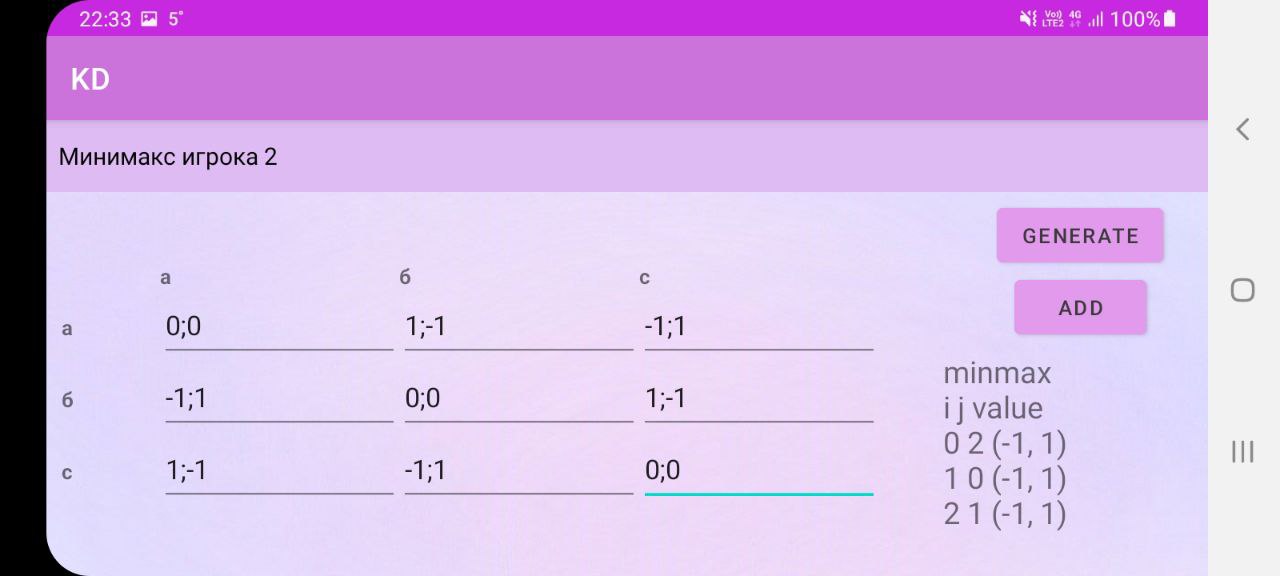


Биматричная игра:

Когда есть один минимакс (выводится значение целой ячейки, берется в ответ только вторая цифра)



Когда есть несколько минимаксов (выводится значение целой ячейки, берется в ответ только вторая цифра)

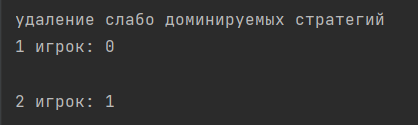
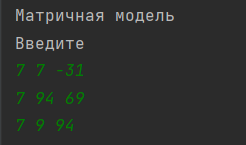


**Слабо доминируемые стратегии**

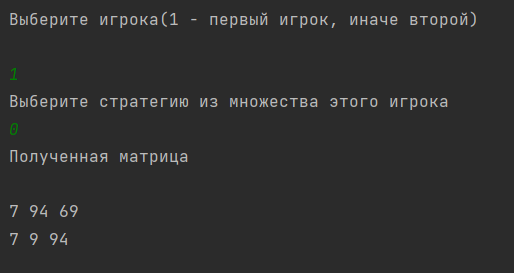
Матричная игра:

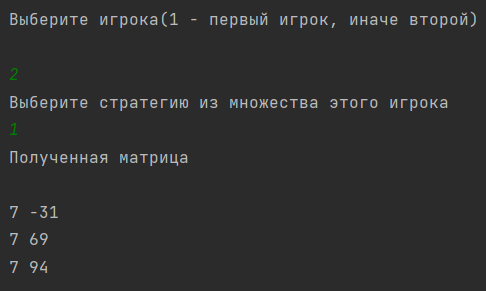
Есть одна стратегия (приведены различные нахождения этой стратегии в матрице):

1)

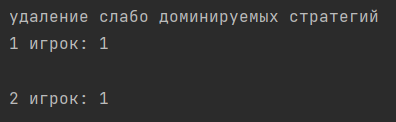
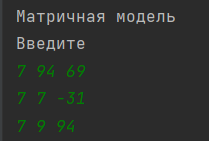


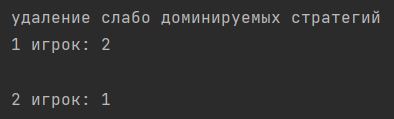
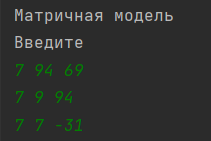
После удаления:

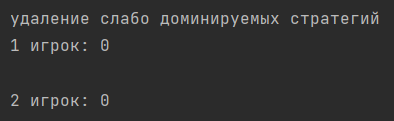
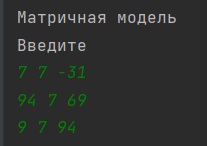


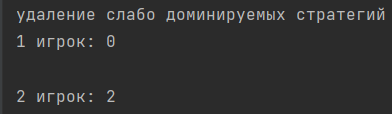
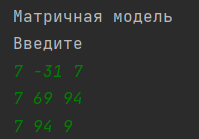


2)



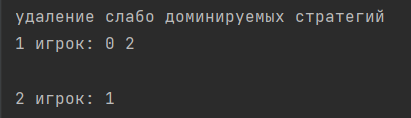
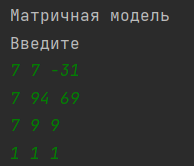




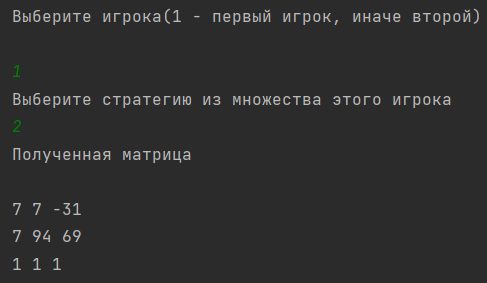


Есть несколько стратегий:

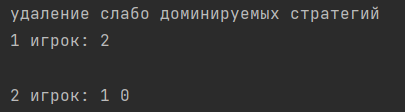
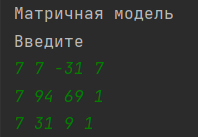
1)



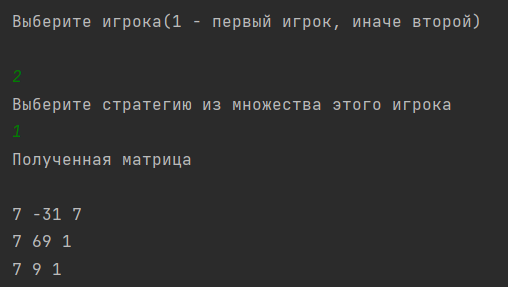
После удаления:



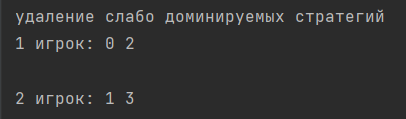
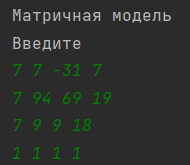
2)



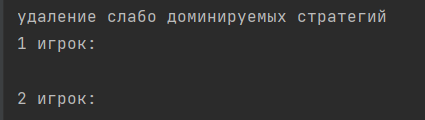
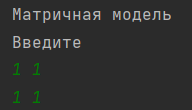
После удаления:

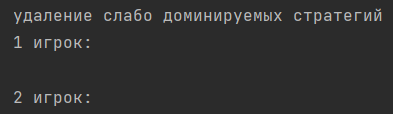
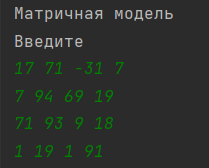


3)



Нет таких стратегий:

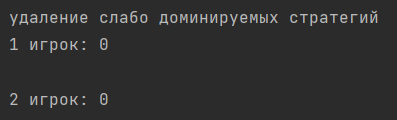
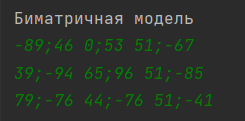




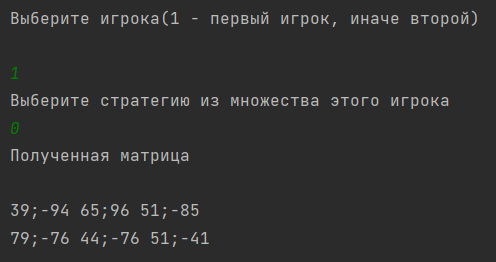
Биматричная игра:

Есть одна стратегия:

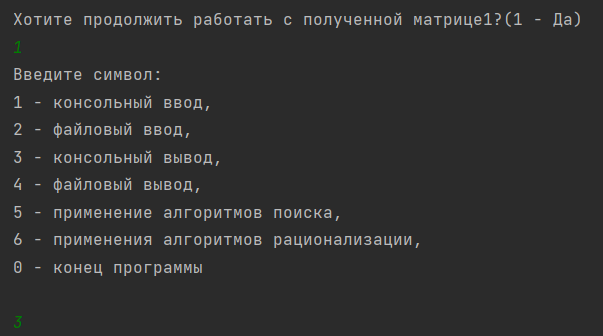
1)

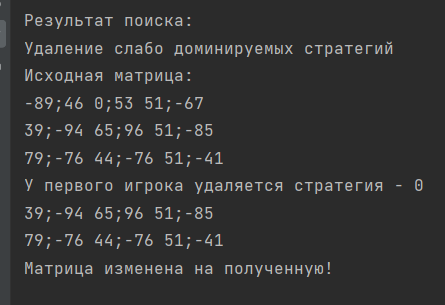


После удаления у первого игрока:

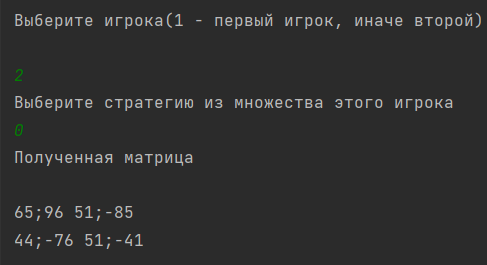


Тут срабатывает история:

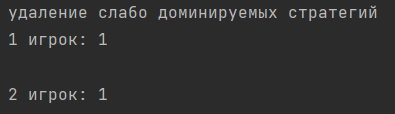
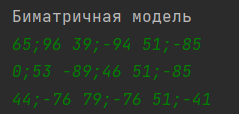




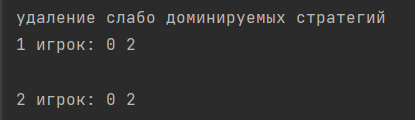
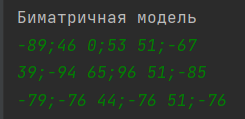
И удаляем стратегию второго игрока:



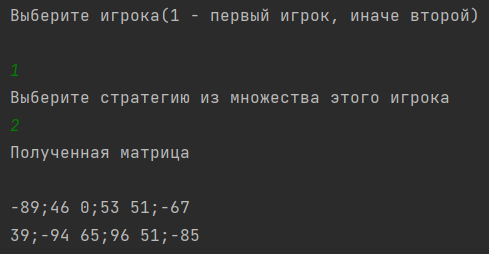
2)



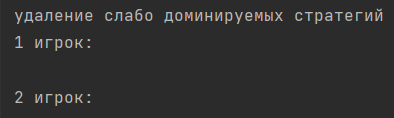
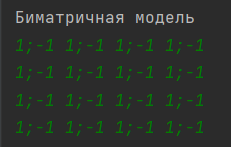
Есть несколько стратегий

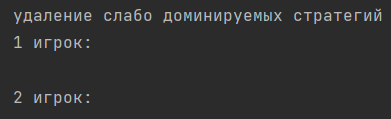
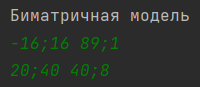


После удаления у первого игрока второй стратегии:



Нет таких стратегий

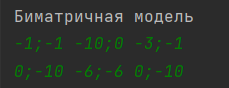




**Строго доминируемые стратегии**

Биматричная модель:

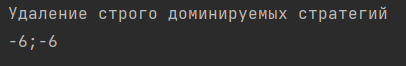
1)



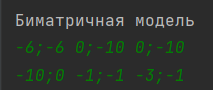
У 1-го игрока: 0 (одна стратегия)

У 2-го игрока: 0 2 (несколько стратегий)

После удаления:



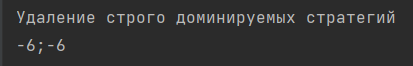
2)



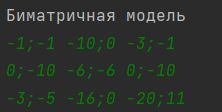
У 1-го игрока: 1 (одна стратегия)

У 2-го игрока: 1 2 (несколько стратегий)

После удаления:



3)после рационализации должно быть -6;-6 0;-10 или как у нас ответ (с последующей проверкой и удалением)



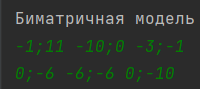
После удаления:

У 1-го игрока: 0 2 (несколько стратегий)

У 2-го игрока: 0 (одна стратегия)

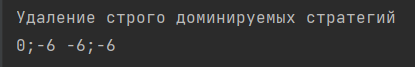


4)

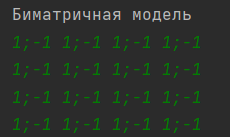


У 1-го игрока: 0 (одна стратегия)

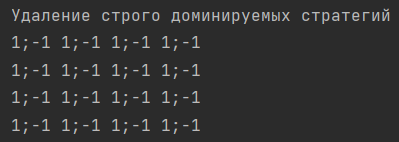
У 2-го игрока: 2 (одна стратегий)



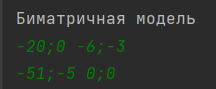
5)



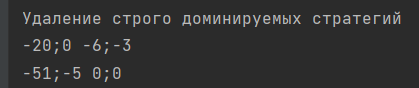
Нет таких стратегий, удаления не произошло:



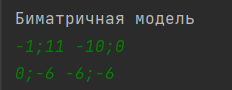
6)



Нет таких стратегий, удаления не произошло:



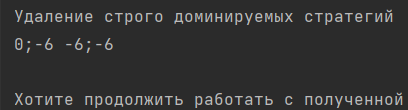
7)



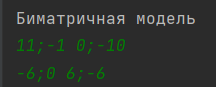
У 1-го игрока: 0 (одна стратегия)

У 2-го игрока: (нет стратегий)

После удаления:



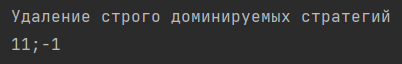
8) после рационализации должно быть 11;-1 -6;0 или как у нас ответ (с последующей проверкой и удалением)



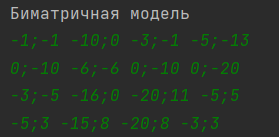
У 1-го игрока: (нет стратегий)

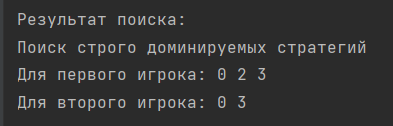
У 2-го игрока: 1 (одна стратегия)

После удаления:



9)



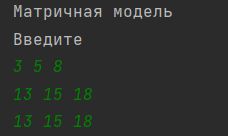


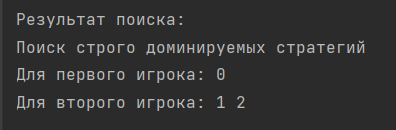
Удаление после всех рационализаций:

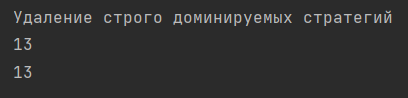


Матричная модель:

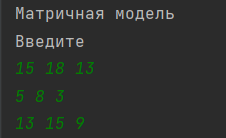
1)

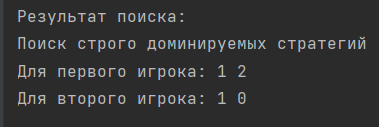


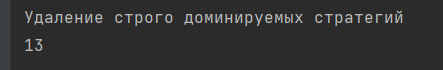




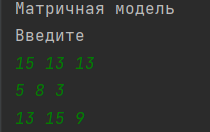
2)

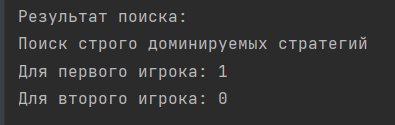


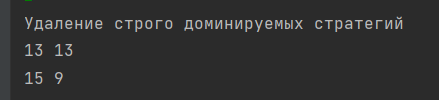




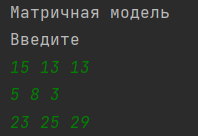
3)

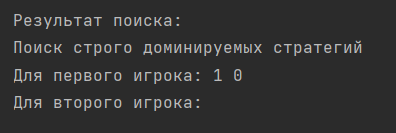






4)

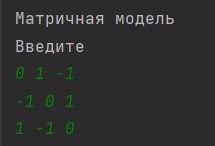




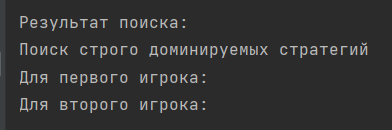
Здесь удаление стратегий было до полной рационализации



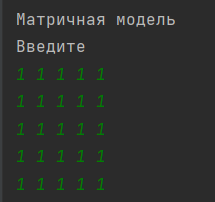
5)



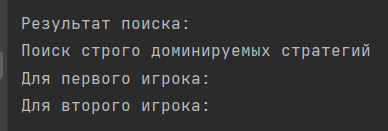
Таких стратегий нет:



6)



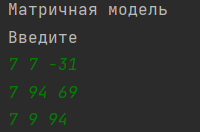
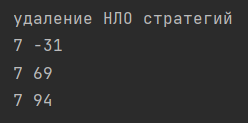
Таких стратегий нет:



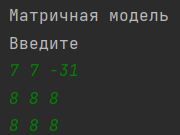
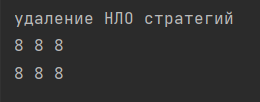
**НЛО стратегии**

Матричная игра:

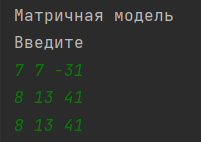
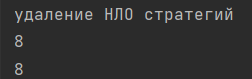
1)У второго игрока удалилась одна НЛО стратегия (1)

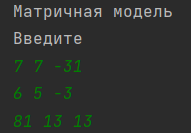
2)У первого удалилась одна НЛО стратегия (0)

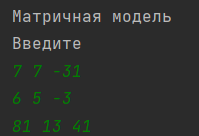
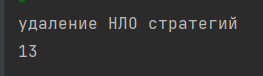
3)У первого удалилась одна НЛО стратегия (0), у второго удалилось две НЛО стратегии (1 2)

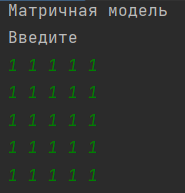
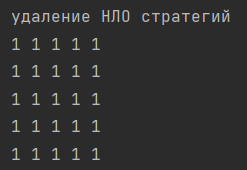
4)У первого удалилось две НЛО стратегии (0 1), у второго удалилась одна НЛО стратегия (0)

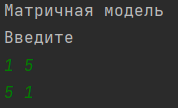
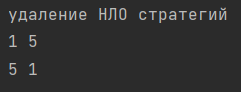
 

5) У первого удалилось две НЛО стратегии (0 1), у второго удалилось две НЛО стратегии (0 2)

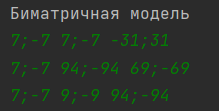
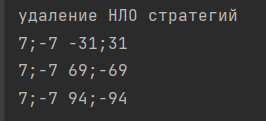
6) У первого нет таких стратегий, у второго нет таких стратегий (ничего не удалится)

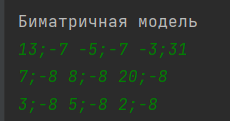
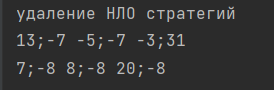
 

Биматричная игра:

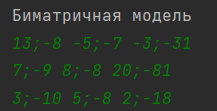
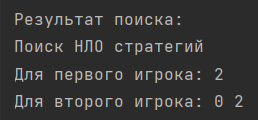
1)У второго игрока удалилась одна НЛО стратегия (1)

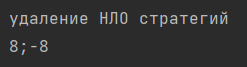
2)У первого удалилась одна НЛО стратегия (2)

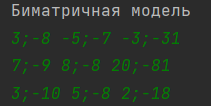
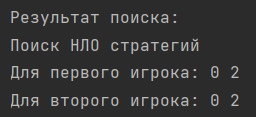
3)У первого удалилась одна НЛО стратегия, у второго удалилось две НЛО стратегии

(После прохождения всех этапов рационализации)

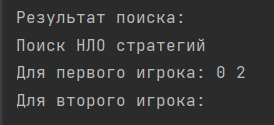
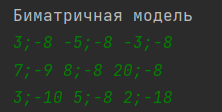


4) У первого удалилось две НЛО стратегии, у второго удалилось две НЛО стратегии



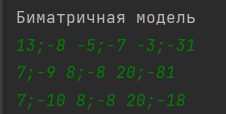
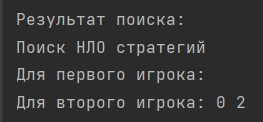
5) У первого удалилось две НЛО стратегии, у второго нет таких стратегий



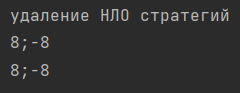
(После прохождения всех этапов рационализации)



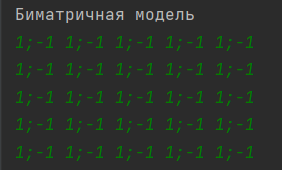
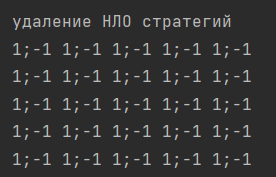
6)У первого нет таких стратегий, у второго удалилось две НЛО стратегии

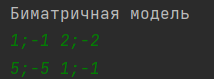
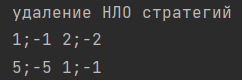
 

(После прохождения всех этапов рационализации)



7) У первого нет таких стратегий, у второго нет таких стратегий (ничего не удалится)

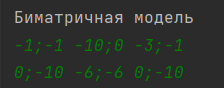
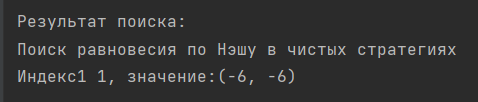
 

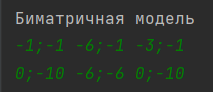
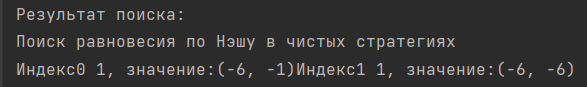
**Поиск равновесий по Нэшу в чистых стратегиях**

Биматричная игра:

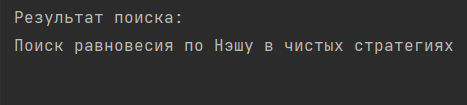
Одно равновесие:

Несколько равновесий:

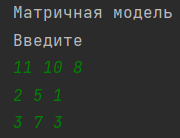
 

Нет равновесий:

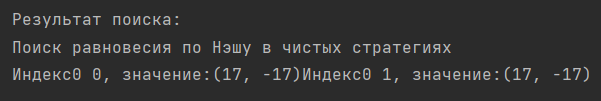
 

Матричная игра:

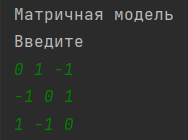
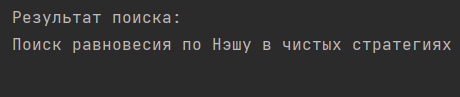
Одно равновесие:

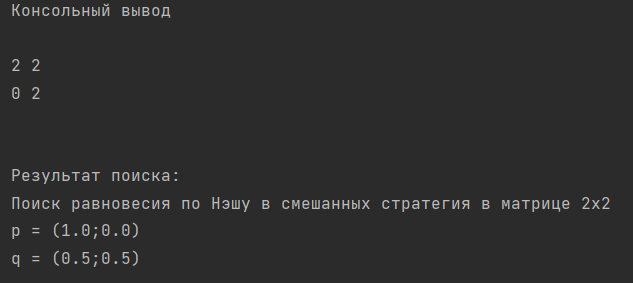
Несколько равновесий:

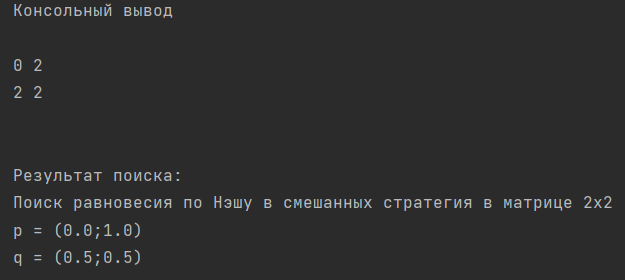
 

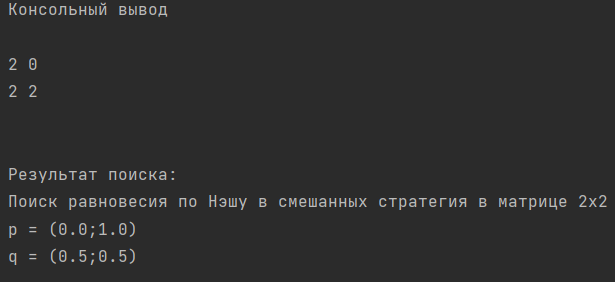
Нет равновесий:

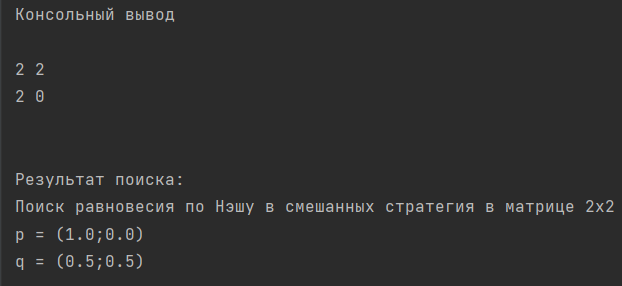
 

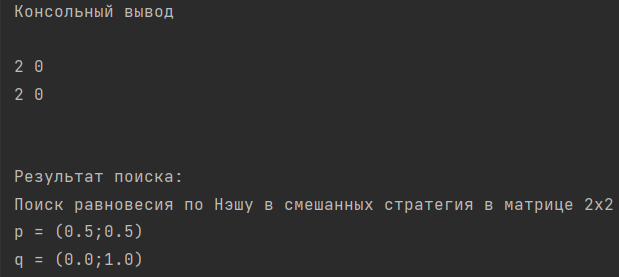
**Поиск равновесия по Нэшу в смешанных стратегия в матрице 2х2**

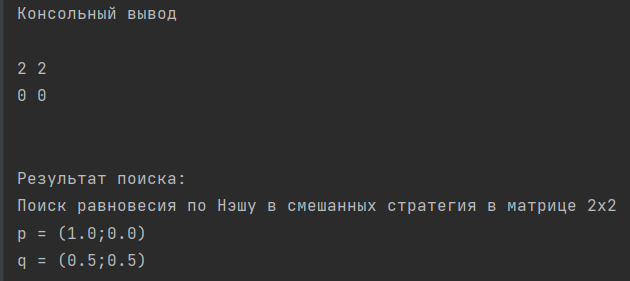


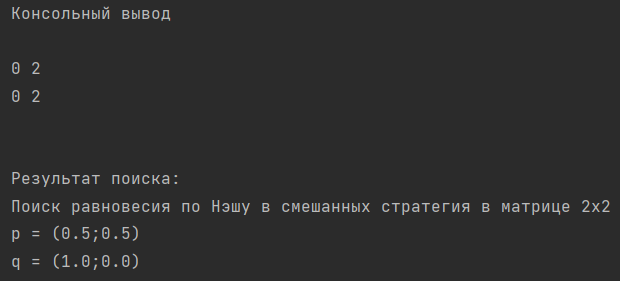


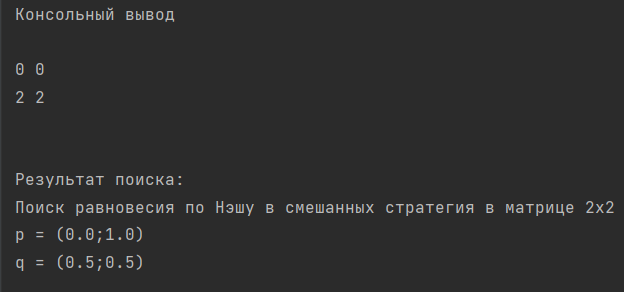


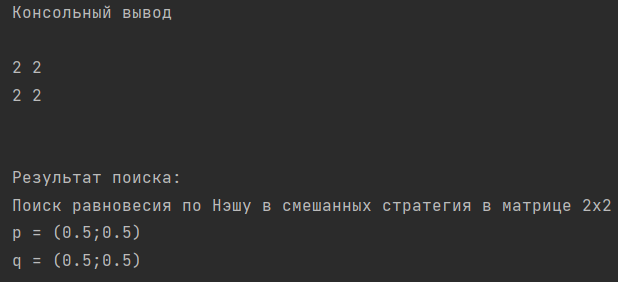




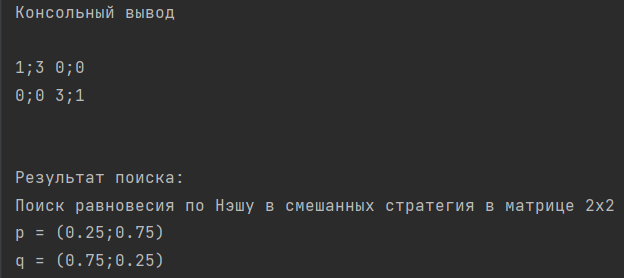


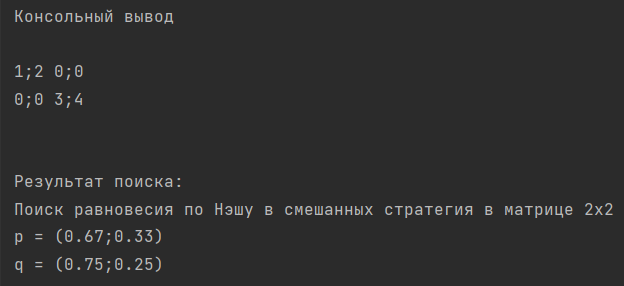












**Тестирование на экранах разных диагоналей**

Также наше приложение тестировалось на телефонах с различной диагональю экрана, например, 6.0”, 6.4”, 5.46”, 5.0”.