

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

ОТЧЕТ ПО КУРСОВОМУ ПРОЕКТУ

По дисциплину

«Конструирование программного обеспечения»

Выполнили студенты:

Данг Тхи Чанг гр. 3530904/80104

Нгуен Уи Динь гр. 3530904/80106

Нгуен Тхи Тху Чанг гр. 3530202/80201

Руководитель

Иванов Александр Сергеевич

Санкт-Петербург
2020

Содержание

Техническое задание.....	3
Разработка архитектуры и детальное проектирование.....	3
Метод решения	4
Скриншоты тестирования программы	6
Список использованных источников	9

Техническое задание

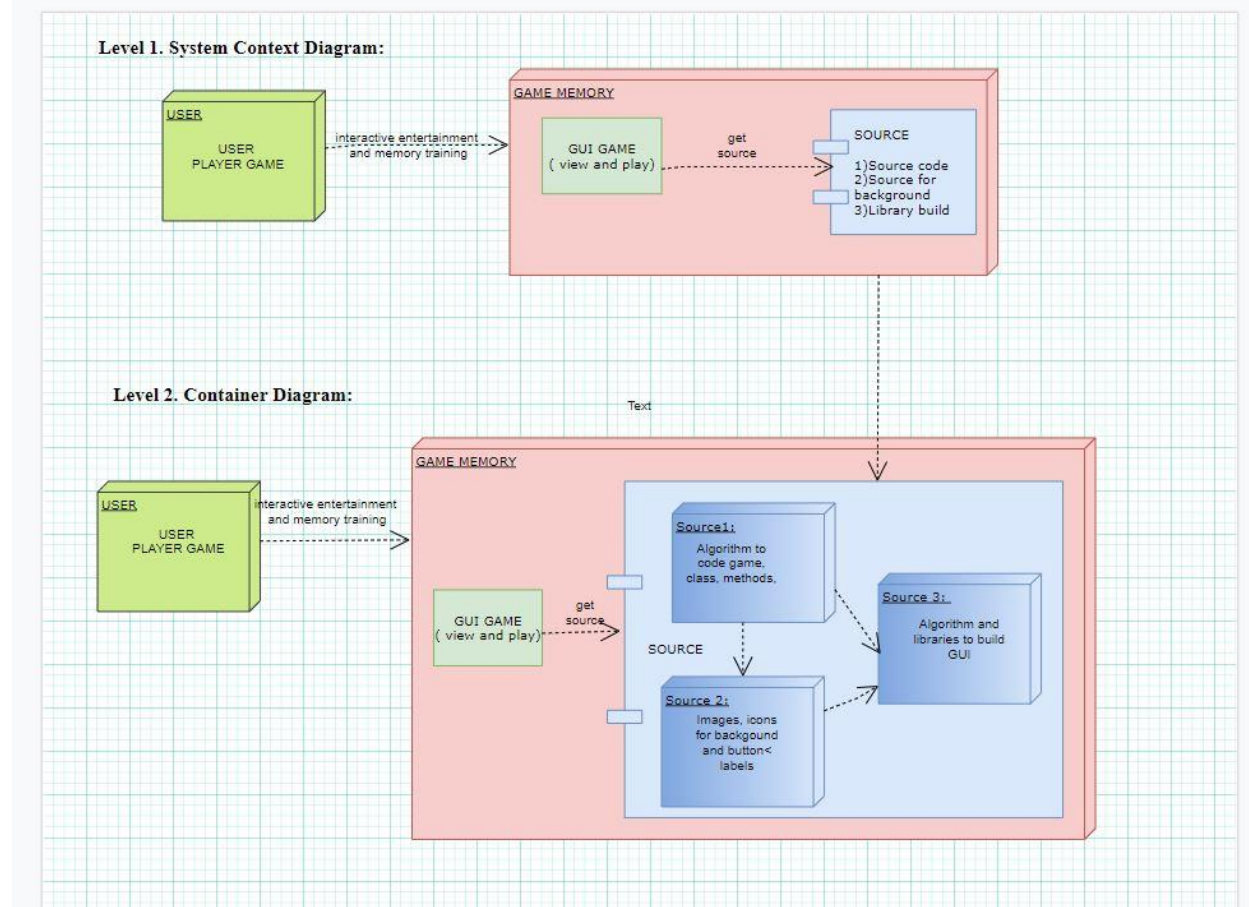
Цель работы: освоить создание GUI приложения, получить навыки работы с программой Java. Разработать на основе полученных навыков игру на запоминание на языке Java.

В ходе летней практики планировалось создание GUI приложения для операционной системы Windows – игры на запоминание – с использованием фреймворка AWT и SWING, которые предоставляет широкие возможности для кроссплатформенной разработки, графики и пользовательского интерфейсы.

GitHub-репозиторий: <https://github.com/Trang68/GameMemory>

Разработка архитектуры и детальное проектирование.

Первые две диаграммы по модели C4:



Метод решения

Разрабатываемая игра должна быть похожа на оригинальную игру на запоминание: Каждый игровой экран будет иметь $2 * k$ (четные числа), картинка была скрыта, картинки не меняют положение на каждом игровом экране. Игрок может открыть две фотографии подряд (игрок увидит эти две фотографии), если они одинаковые, они будут удалены, в противном случае две фотографии останутся скрытыми.

Когда игрок откроет все одинаковые пары картинок, игра закончится, чтобы сделать игру более привлекательной, на каждом экране появится давление на время, требующее от игрока очень спокойного запоминания положения фотографий.

На главном экране игры находится игровая площадка, описанная выше. Кроме того, в верхней части экрана расположена временная шкала, показывающая время воспроизведения каждого уровня. Внизу экрана находится окно, показывающее баллы, которые игрок получил, и окно, которое помогает игроку во время игры. Если игрок открыл правильные 2 фотографии, баллы будут добавлены и если неправильно, баллы будут вычтены. В конце каждого уровня, баллы будут добавляться в зависимости от времени, когда игрок завершает этот уровень. Чем раньше игрок завершит уровень, тем больше баллов он получит и наоборот. Функции помощи для игроков, чтобы начать использовать на 1-м уровне. Эта функция справки поможет игрокам удалить 1 пару изображений. Через каждый уровень будет плюс одна помощь.

Рассмотрим методы, приведенные в классе MemoryGame отвечающем за отрисовку игрового поля и обработку операции игры:

1. `public MemoryGame()` – инициализация класса MemoryGame
 - a. `timer 1` - для проверки изображений и создания эффекта задержки при проверке фотографий.
 - b. `timer2` - для завершения игры, когда время истекло.
2. `public void open()` – для проверки два одинаковые изображения. Если они одинаковые, они будут удалены, в противном случае две фотографии останутся скрытыми.
3. `public Container init()` – для построения графического интерфейса пользователя включают игровые рамки, кнопки отображения баллов,

функциональные кнопки, панель отображения времени игрока, кнопки отображения изображений. Выбранные изображения будут открыты и сравнено.

+) `public void actionPerformed()` – прослушает и выполнит команду функции кнопки справки, чтобы удалить любые два идентичных изображения.

4. `private Icon getIcon()` – извлекать изображения из заданного индекса.
5. `public void newGame()` – создание новую игру.
6. `public void nextGame()` – перейти на более высокий уровень.
7. `public void showDialognewGame()` – показать новый диалог игры, когда время истекло. Хотел бы начать сначала или выйти.
8. `public void actionPerformed()` – это метод переопределения, чтобы выполнить команду, чтобы щелкнуть, чтобы перевернуть изображение.
9. `public JLabel setBackground()` – установит фон уровней во время игры.
10. `public BufferedImage cropImageIcon()` – обрезает изображение в соответствии с заданным соотношением.
11. `public void RemoveTwoIcon()` – удаление двух изображений одинаково.
12. `public int CountFunctionImage()` – для подсчета количества функциональных изображений (+100 или +300).

Рассмотрим методы, приведенные в классе отвечает за настройку матрицы случайных чисел для символизации изображений.

`CreatMatrixNumber()` – настроить цифровую матрицу.

- +) установит все иконки не выбран.
 - +) выбирает половину общей матрицы ($n * m / 2$).
 - +) дублирует выбранные значки.
 - +) установит тяжелый значок для сортировки.
 - +) стирает выбранные значки.
 - +) расположит матрицу
1. `public int[][] getMatrix()` – вернуть созданную матрицу.
 2. `public void bubbleSort()` – расположить элементы в массиве.
 3. `public swapElement()` – переместит двух элемента в массиве.

Полный код программы с подробными комментариями приведен в Приложении.

Скриншоты тестирования программы

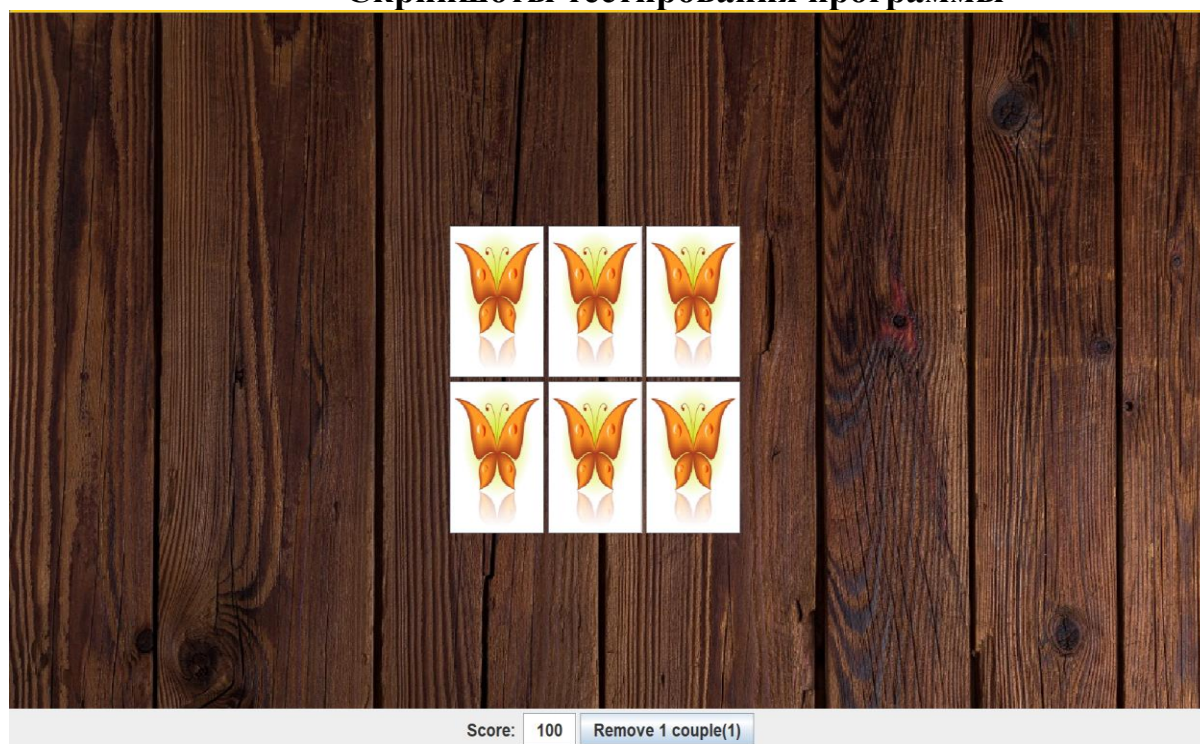


Рис. 1. Начальный экран приложения.

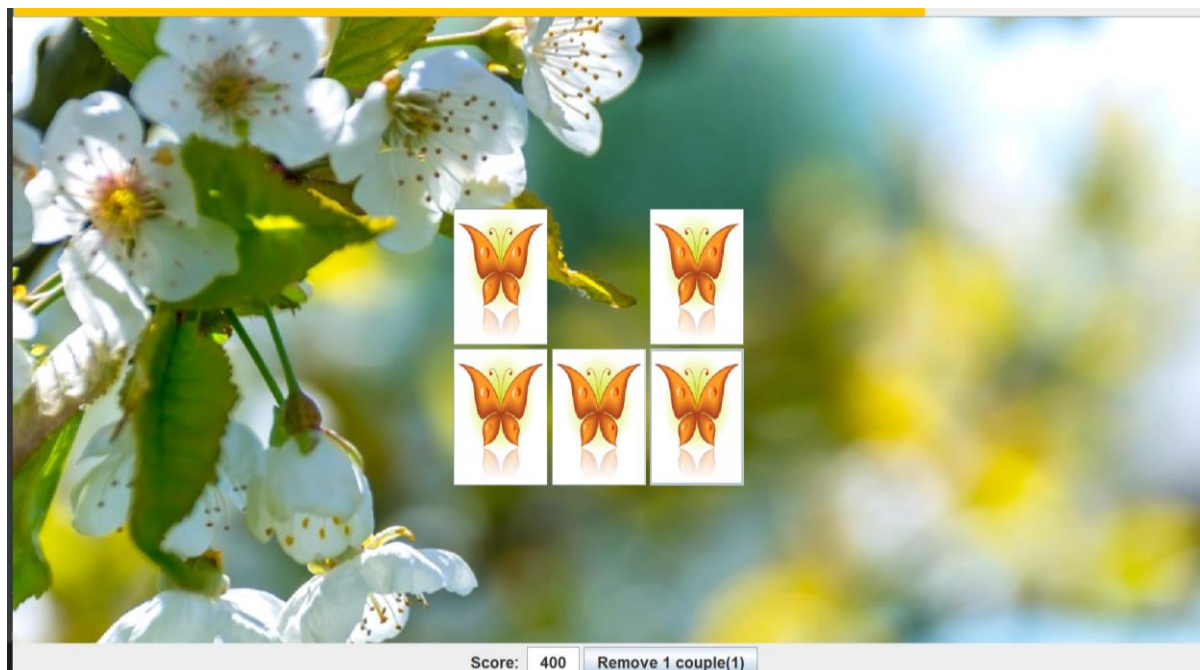


Рис. 2: Экран открытия изображения плюс 300 баллов

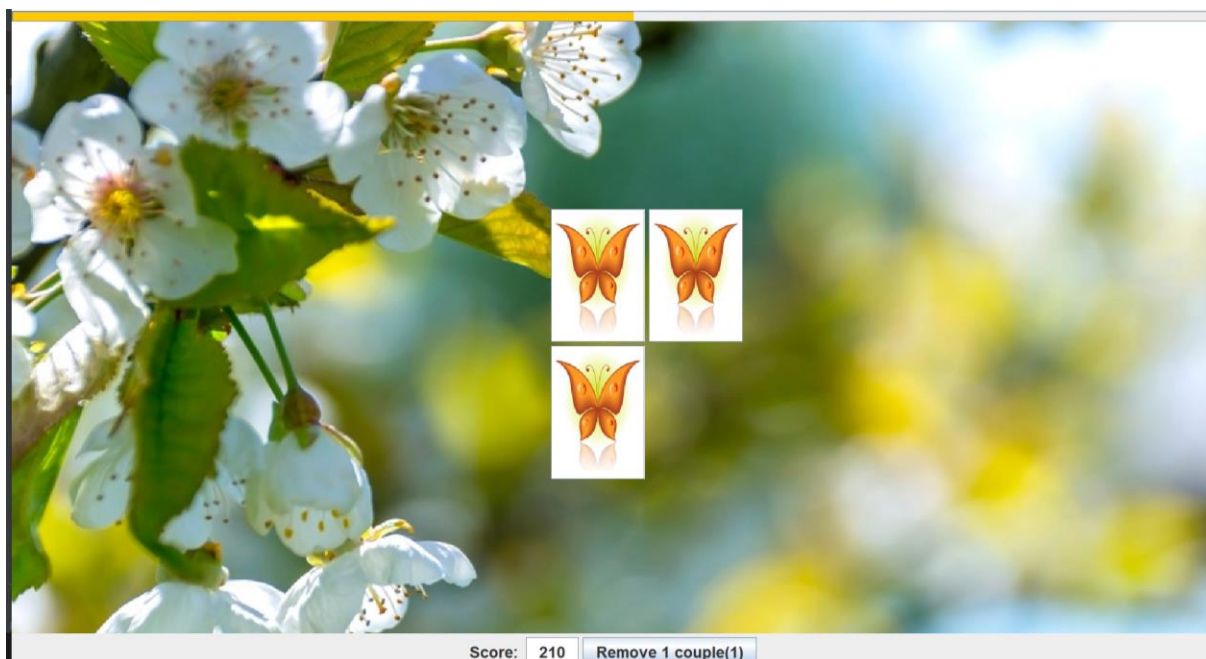


Рис. 3: Экран открытия изображения плюс 300 баллов и те же два изображения.

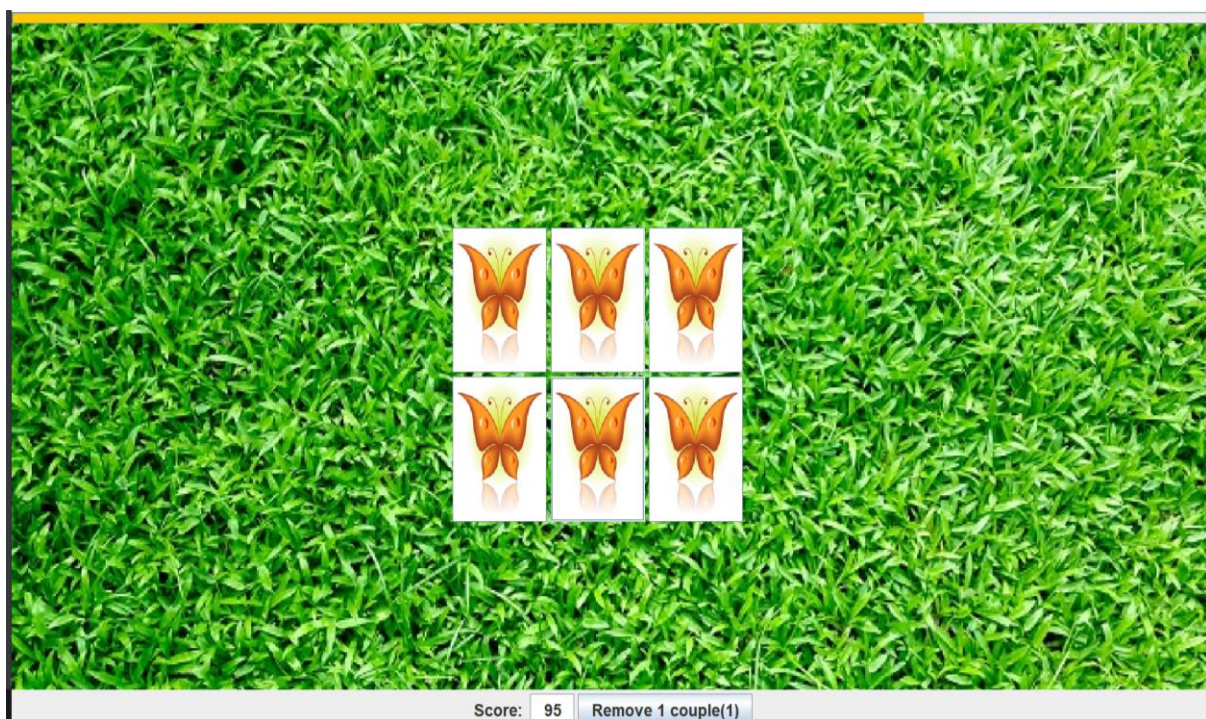


Рис. 4: Экран открытия двух разных изображений.

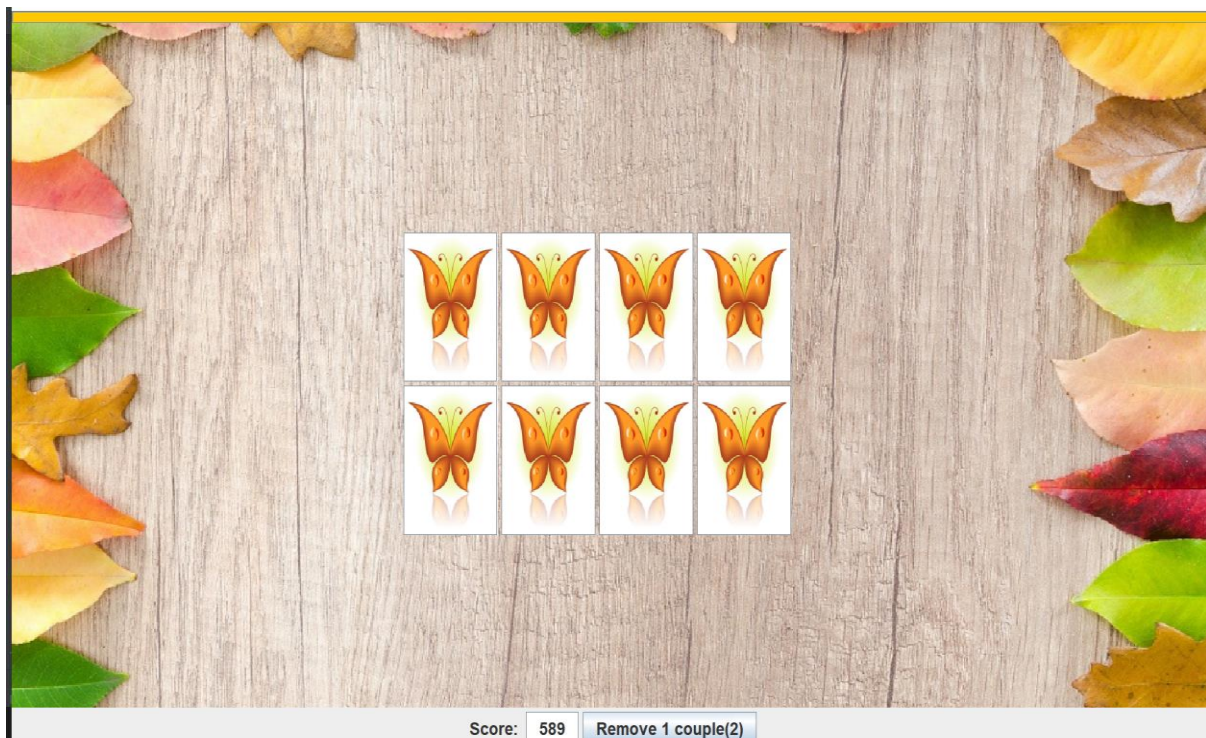


Рис. 5: Экран второго уровня.

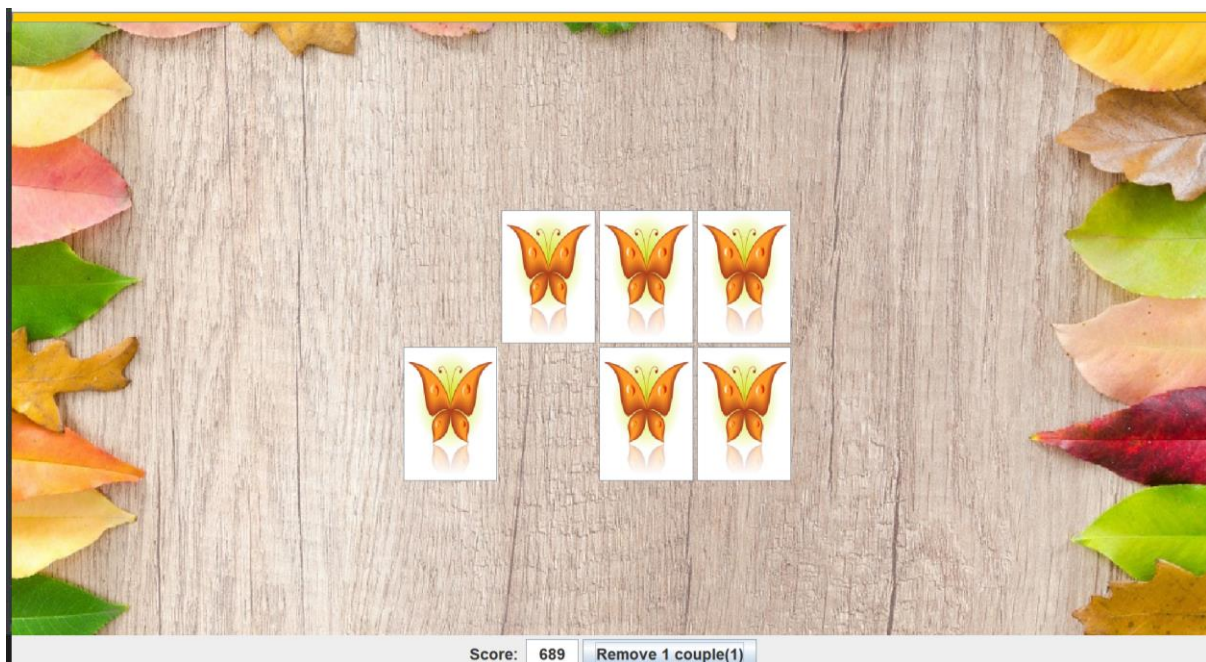


Рис. 6: Экран справки с помощью удаления пары изображений.

Список использованных источников

1. <https://www.javatpoint.com/java-swing> - часть Java Foundation Classes (JFC), которая используется для создания оконных приложений.
2. <https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html> - Содержит все классы для создания пользовательских интерфейсов и для рисования графики и изображений.