ГУАП
КАФЕДРА № 51

ПРЕПОДАВАТЕЛЬ

доцент, к.т.н.		Линский Е. М.
должность , уч. степень, звание	подпись, дата	инициалы, фамилия

ПРАКТИКА
Игра «Шашки»

по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	5612		В.С.Марковский
		подпись, дата	**инициалы, фамилия**

Санкт-Петербург 2018

Задание

Написать программу, позволяющую играть в шашки.
В данной версии программы реализованы следующие возможности:

- Режим для двух игроков
- Режим игры с ИИ
- Выбор сложности и цвета своих шашек при игре с ИИ
- Возможность продолжения последней сохраненной партии
- Возможность отмены хода
- Сохранение истории ходов с возможностью возвращения на любое состояние доски

Разделение обязанностей:

- Недошивин Павел: создание внутренней модели работы программы, разработка алгоритма игры компьютера с пользователем, тестирование модулей
- Марковский Владимир: работа с графическим интерфейсом программы, тестирование модулей

Руководство пользователя

После запуска программы появляется главное меню.

Рис. 1. Главное меню

Пользователю доступны кнопки:

1. New game – выбор режима новой игры
2. Continue – продолжение последней сохраненной партии (если такая отсутствует – появляется диалоговое окно)

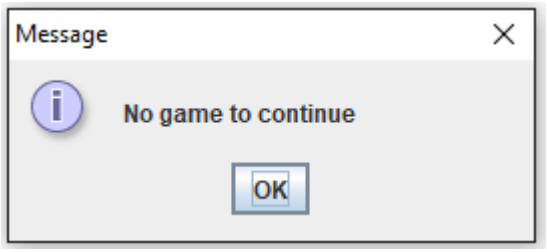

Рис. 2. Предупреждение: продолжить невозможно!

3. Rules – правила игры

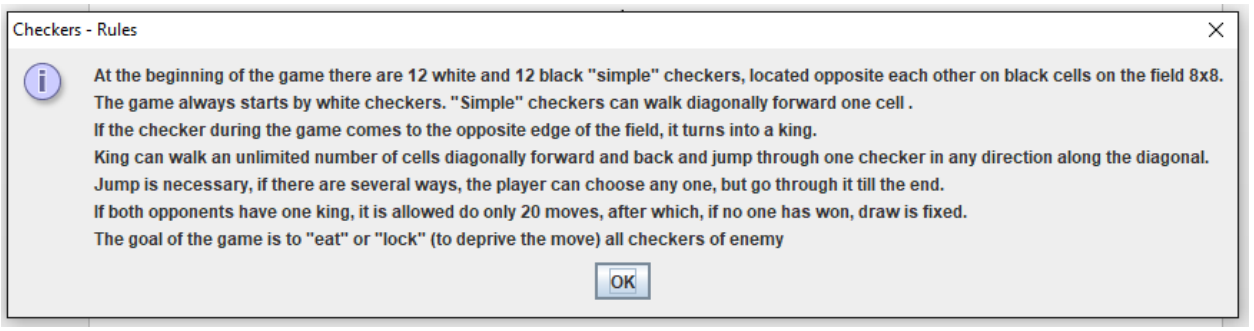

Рис. 3. Правила

4. Authors – сведения о разработчиках

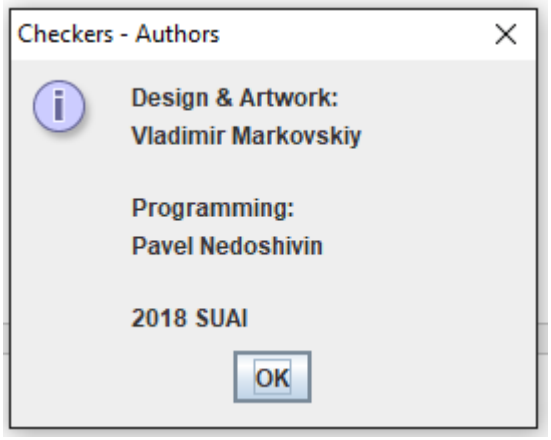

Рис. 4. Об авторах

5. Exit – выход из программы

Пользователю доступны следующие режимы:

Рис. 5. Выбор режимов

1. Player vs computer – режим игры с ИИ
2. Player vs player – режим игры для двух игроков

Если выбрана игра с компьютером, то появляется опция выбора сложности. Доступны три уровня: easy, medium, hard.

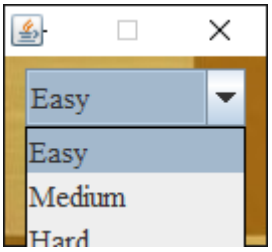

Рис. 6. Выбор сложности

Затем появляется выбор цвета шашек, за которые будет играть пользователь.

Рис. 7. Выбор цвета шашек

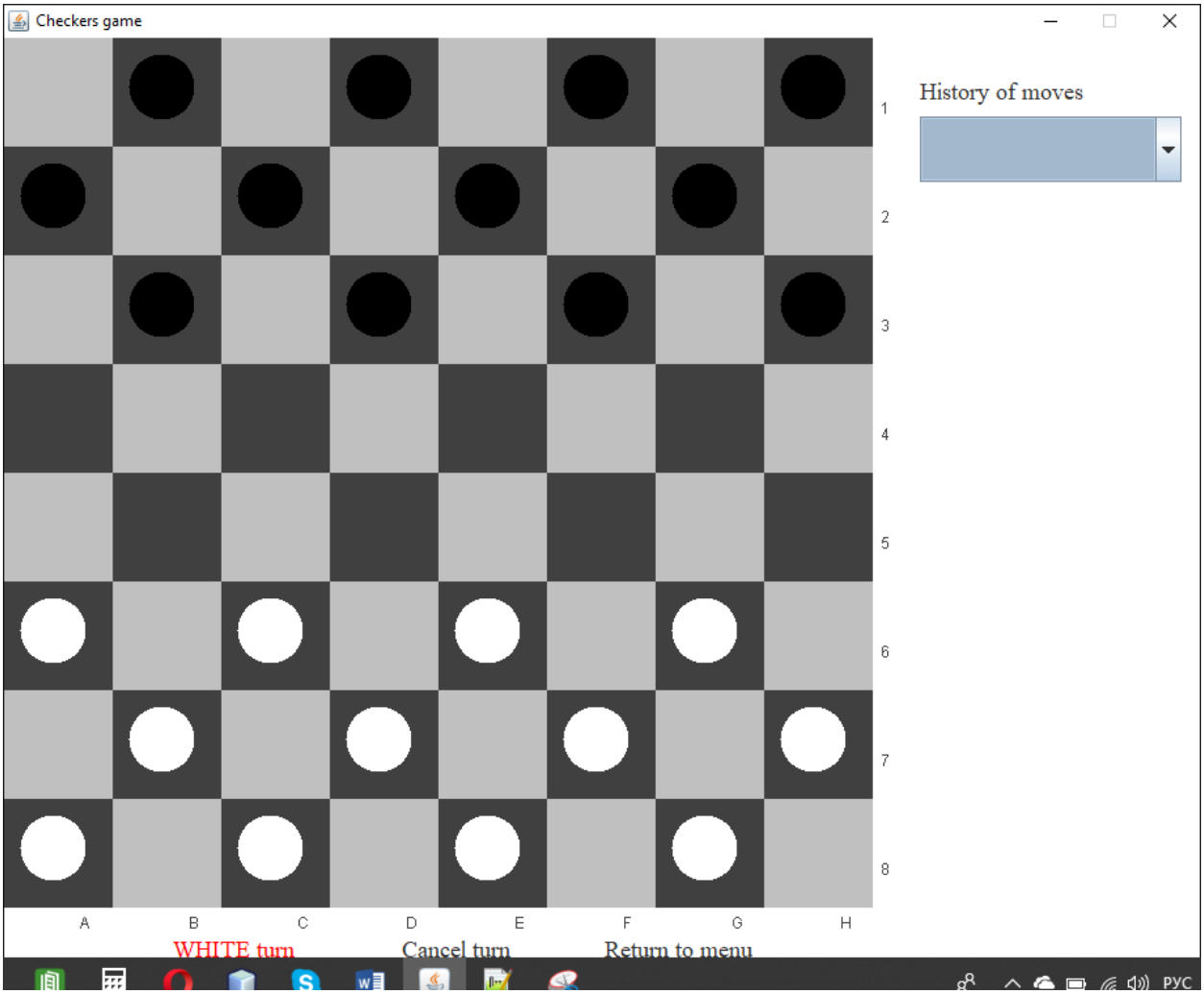

Рис. 8. Доска с шашками

Для хода необходимо нажать на одну из своих шашек, а затем нажать на одну из зеленых (красных) клеток. Для отмены хода необходимо нажать на кнопку Cancel turn. Справа отображается история ходов. При нажатии на один из пунктов можно вернуться к состоянию доски после этого хода, но при этом все последующие ходы удаляются.

Описание публичных классов и методов

Класс Main

Здесь запускается программа.

1. public static void main(String[] args) throws IOException – точка входа в программу

Пакет model:

Класс CheckerCell

Хранение одной клетки доски.

1. public CheckerCell(int cod, int r, int c) – конструктор. Параметры: код, номер строки, номер столбца
2. public CheckerCell(CheckerCell o) – копирующий конструктор
3. public void changeCode(int value) – изменить код клетки
4. public int getRow() – возвращает номер строки
5. public int getColumn() – возвращает номер столбца
6. public int getCode() – возвращает код
7. public String toString() – возвращает строковое представление клетки: ее код.

Класс CheckerNode

Узел дерева при оценке хода компьютером.

1. public CheckerNode(int v, CheckerCell f, CheckerCell t) – конструктор. Параметры: значение узла, клетка, откуда начат ход, клетка, где ход будет завершен
2. public CheckerNode(CheckerNode o) – копирующий конструктор
3. public void setChoice(CheckerNode v) – установить какой из узлов-наследников будет выбран
4. public CheckerNode getChoice() – вернуть узел-наследник, который будет выбран
5. public LinkedList getChain() – вернуть цепочку поедания шашек
6. public boolean hasChain() – есть ли цепочка поедания в этом узле
7. public void addToChain(CheckerCell v) – добавить клетку в цепочку поедания
8. public CheckerCell getFrom() – вернуть клетку, откуда будет начат ход
9. public void changeValue(int v) – изменить значение узла дерева
10. public CheckerCell getTo() – вернуть клетку, где будет завершен ход
11. public void addSuccessor(CheckerNode v) – добавить наследника

12. public LinkedList getSuccessors() – вернуть список наследников

Класс CheckersAI

Искусственный интеллект.

1. public CheckersAI(CheckersBoard v, int diff) – конструктор. Параметры: доска с шашками, уровень сложности
2. public void makeTurn() throws IOException – сделать ход

Класс CheckersBoard

Представление доски и операции с ней.

1. public CheckersBoard(boolean cont, String logname) throws IOException – конструктор. Параметры: есть ли продолжение последней партии, имя лога
2. public CheckersBoard(CheckersBoard o) – копирующий конструктор
3. public int getTurn() – вернуть номер хода
4. public String toString() – возвращает строковое представление доски
5. public void saveBoard(String logname) throws IOException – сохранить состояние доски в лог
6. public LinkedList getNodes() throws IOException – вернуть список возможных узлов дерева ходов (для ИИ)
7. public void cancelTurn() throws IOException – отменить ход
8. public void loadBoard(String filename) throws IOException – загрузить из файла состояние доски
9. public boolean couldEat() – проверка, возможно ли поедание шашек противника при данном состоянии доски
10. public boolean chooseChecker(int row, int column) – выбрать шашку для хода
11. public int makeTurn(int r, int c) – совершает перемещение выбранной шашки в указанную клетку. Возвращает: 0 – ход завершен, 1 – необходимо продолжить цепь поедания, 2 – указана неверная клетка для хода
12. public void deleteMark() – удалить особую маркировку клеток: 5 – в эту клетку можно сходить (выделена зеленым), 6 – в эту клетку нужно сходить (выделена красным).

Пакет io:

Класс CheckerReader

Чтение и обработка текстовых файлов.

1. public LinkedList loadHistory(String filename) throws IOException – загрузить историю ходов и вернуть список строк для панели истории
2. public Object[] loadSettings(String filename) throws IOException – загрузить настройки партии
3. public String loadBoard(String filename) throws IOException – загрузить состояние доски
4. public void cancelTurn(String logname) throws IOException – отменить последний ход
5. public int getTurn() – возвращает номер хода
6. public int getCountdown() – возвращает счетчик оставшихся до ничьей ходов

Класс CheckerWriter

Редактирование текстовых файлов, сохранение данных.

1. public void clearHistory() throws IOException – очистить историю ходов
2. public void saveHistory(String filename, String s) throws IOException – добавить в историю ходов строку s
3. public void saveSettings(String filename, boolean vsComp, int diff, boolean compTurn) throws IOException – сохранить настройки партии. Параметры: имя файла, игра с ИИ, уровень сложности, цвет шашек коспьютера
4. public void clearLog(String logname) throws IOException – очистить лог партии
5. public void saveBoard(String filename, int turn, int count, String s) throws IOException – сохранить состояние доски в лог. Также указывается номер хода и счетчик оставшихся ходов
6. public void cancelTurnHistory(String filename) throws IOException – удаление последней строки в истории ходов при отмене хода

Пакет view:

Класс Menu

Реализация меню игры.

1. public Menu() – конструктор без параметров.

2. public void actionPerformedl(ActionEvent evt) – обработка нажатия на кнопку

Класс Board

Шахматная доска, фигуры, история ходов.

1. public Board() – конструктор без параметров.
2. public void actionPerformedl(ActionEvent evt) – обработка нажатия на кнопку
3. public void setMode(int d, boolean who) – запуск игры с компьютером. Параметры: сложность, чей ход
4. public void actionPerformed(ActionEvent evt) – обработка нажатия на кнопку отмены хода и список с историей ходов
5. public void changeMessage() – вывод сообщения о том, кто ходит
6. public void doNewGame() – запуск партии с компьютером
7. public void doContinue() – продолжение последней партии
8. public boolean gameOver() – проверка на окончание партии, вывод сообщения о победителе
9. public void doChooseChecker(int row, int col) – выбор шашки. Параметры: координаты клеток
10. public void returnToMove(int num) – возвращение к заданному ходу. Параметр: номер хода
11. public String getMove() – получить координаты хода в строковом представлении
12. public void doMakeMove(int row, int col) – сделать ход. Параметры: координаты конца хода
13. public void paintComponent(Graphics g) – рисование доски, шашек, отметка выбранной шашки, возможных клеток для хода