**Национальный исследовательский ядерный университет «МИФИ»**

Проект по дисциплине: Информационные системы и технологии специального назначения.

На тему: Создание GUI приложения с использованием библиотеки SWT.

Пояснительная записка.

Преподаватель: Ибрагимов А.М.

Cтуденты: Меркель А.В., Порошкина В.В. гр. К6-362

Москва, 2016

Приложение «Tetris x 2» является аналогом общеизвестной компьютерной игры «Тетрис» с некоторыми модификациями. Использование данного приложения подразумевает участие 2-х пользователей.

Описание функционала

Приложение можно условно разделить на две области: игровое поле и поле выбора фигуры. Игровое поле имеет размер 15х24 клеток, включая границы. Всего имеется 7 разных классических фигур тетрамино, каждая из которых окрашена в определенный цвет.

**1-й игрок** (активная область – игровое поле):

- может производить манипуляции с поданной ему фигурой на игровом поле, а именно: вращать фигуру, передвигать фигуру по горизонтали вправо/влево.

Фигура движется вниз, пока не наткнётся на любую другую фигуру или не достигнет нижней линии игрового поля. Такое состояние фигуры будем называть «фигура установлена». Если при этом заполнился горизонтальный ряд из 13-и клеток, он удаляется и все линии, что были выше него, опускаются на одну клетку вниз.

**2-й игрок** (активная область – поле выбора фигуры):

- может видеть игровое поле, на котором в реальном времени отображаются действия 1-го игрока;

- имеет возможность выбирать среди предоставленного ему набора фигур (любые различные три фигуры из имеющихся семи) одну, которая впоследствии будет передана 1-му игроку (если выбор не был осуществлен до момента установки 1-м пользователем предыдущей фигуры, приложение само выбирает любую фигуру);

**Сервер**: устанавливает соединение между 1-м и 2-м игроками; пересылает нажатия 1-го игрока 2-му для эмуляции у последнего его действий; пересылает выбранные 2-м игроком фигуры (элементы перечисления) 1-му.

Игра заканчивается, когда новая фигура не может поместиться в игровое поле (осталось менее 3-х свободных горизонтальных полос в центре поля). Игроки получают очки за каждый заполненный ряд, поэтому цель игры — заполнять ряды, не заполняя само игровое поле (по вертикали) как можно дольше, чтобы таким образом получить как можно больше очков.

Чем больше набрано очков в процессе текущей игры, тем больше скорость падения фигур.

Разработка

При создании приложения в качестве среды разработки используется ИСР Eclipse; так же используется библиотека SWT.

Структура приложения подразумевает проект, содержащий несколько классов, каждый из которых специализируется на выполнении определенных задач. Логически проект делится на три части, которые содержатся в разных пакетах:

* ru.game.client – клиентская часть приложения;
* ru.game.tetris.server – серверная часть приложения, реализует сетевое взаимодействие между клиентами;
* ru.game.message – вспомогательная часть приложения, посредством которой происходит сетевое взаимодействие.

В пакете ru.game.client содержатся классы:

* Block – класс одного блока (клетки) поля.

Содержит поля Color color – цвет блока и boolean filled – обозначающее, заполнен блок или пуст.

Содержит методы:

* getFilled – получить значение заполненности блока;
* setFilled – установить значение заполненности блока;
* getColor – получить цвет блока
* setColor – установить цвет блока
* Tetramino – класс цельной фигуры тетрамино.

Содержит поля int x, y – координаты фигуры, Color color – цвет фигуры и Composite figure, служащий для применения к фигуре определенных методов библиотеки SWT, Shape shape – форма фигуры (элемент перечисления);

Содержит методы:

* generateTetramino – создание элемента перечисления рандомной фигуры;
* createTetramino - создание новой фигуры в зависимости от поданного элемента перечисления;
* rotate - вращение фигуры.
* TetrisGame – основной класс клиентской части приложения, наследник Shell.

Содержит методы:

* main – создает корневой объект display и окно приложения, реализующий цикл обработки событий;
* createСontentsofWindow – задает расположение, размеры, название окна приложения.
* TetrisMenu – класс, отвечающий за создание меню приложения и реализацию действий при нажатии на кнопки меню.

Содержит метод:

* createMenu, реализующий вышеописанные функции.
* Client – класс, отвечающий за сетевое взаимодействие с сервером.

Содержит методы:

* createSession – создание новой игровой сессии на двух игроков;
* joinSession – присоединение к одной из существующих игровых сессий;
* closeSession – закрытие текущей игровой сессии;
* sendMessage – отправка сообщений серверу;
* readMessage – чтение сообщений от сервера;
* hasMessage – проверка на наличие необработанных сообщений от сервера.
* TetrisCanvas – класс, отвечающий за отрисовку приложения, наследник Canvas, реализующий интерфейсы PaintListener, Runnable, KeyListener, DisposeListener.

Содержит методы:

* drawWaiting – создание новой игровой сессии (посредством метода createSession класса Client) и ожидание подключения второго игрока, после чего вызывается метод initializeGame для запуска игры;
* drawSessionList – подключение к существующей игровой сессии (посредством метода joinSession класса Client) и запуск игры с помощью метода joinGame;
* drawBlocks – создание блоков поля;
* initializeGame – вызывается из метода drawWaiting, запускает игровой цикл для 1-го игрока и обработку сообщений от сервера;
* joinGame – вызывается из метода drawSessionList, запускает игровой цикл для 2-го игрока и обработку сообщений от сервера;
* gameOver –завершении игры, остановка игрового цикла;
* isGridEnabled – управление видимостью сетки поля;
* createNullTetramino – создание первой рандомной фигуры (вызывает конструктор из класса Tetramino);
* createTetramino – создание фигуры в зависимости от элемента перечисления (вызывает конструктор из класса Tetramino);
* createTetraminoSelection – создание 3-х различных фигур на выбор для 2-го игрока;
* clearTetraminoSelection – очищение полей фигур на выбор;
* transformTetraminoToBlocks – преобразование фигуры в неактивные блоки (т.е. превращает фигуру в фон поля);
* isTouching – проверка, достигла ли фигура границ игрового поля/линий «старых» фигур;
* stepDown/ stepLeft/stepRight – движение фигуры вниз/влево/вправо;
* checkingLine – проверка, заполнена ли горизонтальная линия из 13-ти блоков;
* clearFullLine – удаление заполненной линии (вызывает метода scoreUp) ;
* scoreUp – увеличение текущего счета игры (вызывает метод speedUp );
* speedUp – увеличение скорости падения фигур;
* processingClient – обработка сообщений, пришедших от сервера;
* run – запуск игрового цикла;
* iteration – реализация игрового цикла обоих игроков;
* drawTetramino – рисование фигур;
* keyP – метод 2-го игрока для эмуляции действий 1-го игрока на основе пришедших от сервера сообщений;
* keyPressed – обработка событий, связанных с нажатием на клавиши;
* widgetDisposed – освобождение ресурсов;
* paintControl – отрисоква компонентов окна (фон, фигуры и блоки игрового поля).

Пакет ru.game.message состоит из классов:

* Message – является контейнером различных видов сообщений, посредством которых взаимодействуют клиентская и серверная части приложения.

Содержит перечисление всех возможных типов сообщений.

* Tetramino – класс цельной фигуры тетрамино, для ее правильной передачи посредством Message.

Пакет ru.game.tetris.server состоит из классов:

* Player – класс, отвечающий за сетевое взаимодействие с клиентом.

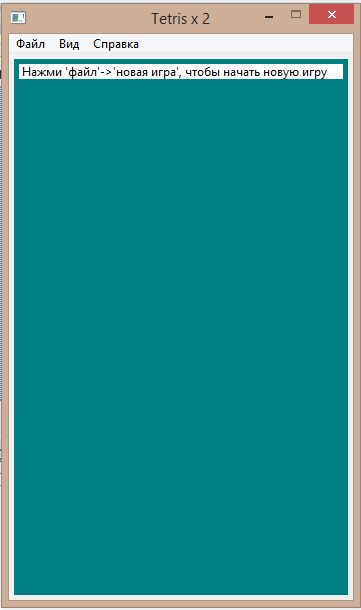
Содержит методы:

* closeConnection – закрытие текущего соединения;
* sendMessage – отправка сообщений клиенту;
* getMessage – чтение сообщений от клиента;
* hasNewMessages – проверка на наличие необработанных сообщений от клиента.
* Server – основной класс серверной части приложения. Содержит:
* основной метод main, запускающий сервер, на котором будет храниться набор сессий для двух игроков.
* start – метод, в котором реализована логика подключения игроков к серверу. Создаёт новую сессию в случае сообщения от игрока, нажавшего «Новая игра», подключает игрока к новой сессии в случае сообщения от игрока, нажавшего «Присоединиться к игре».
* Session – класс отдельной сессии на двух игроков. Осуществляет пересылку сообщений между игроками внутри одной сессии.

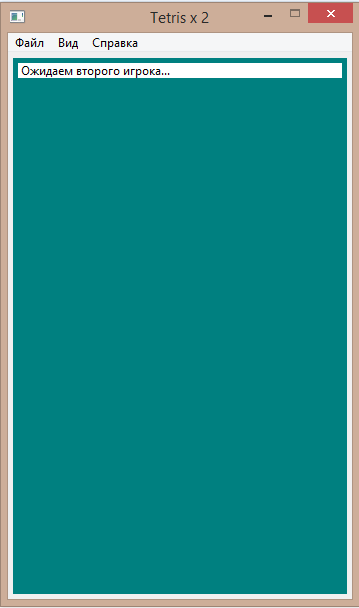
Содержит методы:

* addPlayer – добавляет игрока к текущей сессии.
* readyToStart – возвращает тип boolean, проверяет, можно ли запустить игру на текущей сессии.
* run – запуск игрового цикла. Запускает текущую сессию, отсылает игрокам сообщение о начале игры, реализует обмен сообщениями.

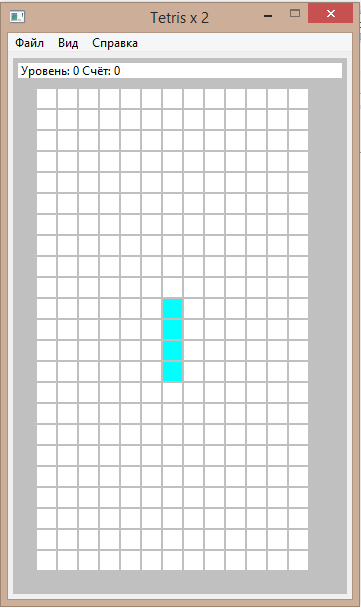
Скриншоты приложения:



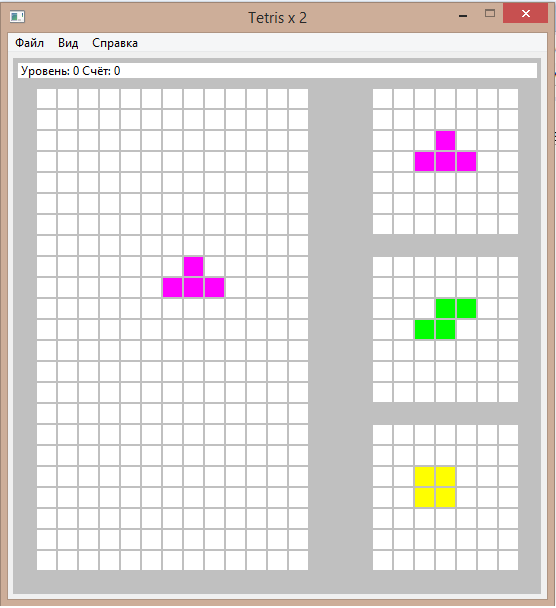
Приложение запущено, игра остановлена.



1-й игрок запустил игру, ожидает 2-го игрока



Идет игра, вид окна 1-го игрока



Идет игра, вид окна 2-го игрока