



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ М.В. ЛОМОНОСОВА

Факультет вычислительной математики и кибернетики

Кафедра алгоритмических языков

Отчет о выполнении задания практикума

**ИГРА “TETRIS”**

Студенты 324 группы

Глуходед А.А., Гончаренко Е.Е.

Москва, 2022

## ПОСТАНОВКА ЗАДАЧИ

**Tetris** — компьютерная игра, цель которой — заполнить падающими фигурками всё пространство по горизонтали.

Случайные фигурки падают сверху в прямоугольный стакан шириной 10 и высотой 20 клеток. В полёте игрок может поворачивать фигурку на  $90^\circ$  и двигать её по горизонтали. Также можно «сбрасывать» фигурку, то есть ускорять её падение, когда уже решено, куда фигурка должна упасть. Фигурка летит до тех пор, пока не наткнётся на другую фигурку либо на дно стакана. Если при этом заполнился горизонтальный ряд из 10 клеток, он пропадает и всё, что выше него, опускается на одну клетку.

### Базовые требования

1. Интерфейс для игры в Тетрис
2. Подсчет игровых очков
3. Сохранение результатов игры

### Индивидуальные части

1. Расширенный интерфейс (Гончаренко Е.Е.)
2. База данных для сохранения результатов игры (Глуходед А.А.)

## МОДУЛИ ПРОЕКТА

Проект состоит из следующих модулей:

- Constants.hs – константы, используемые в проекте;
- DBase.hs – реализация базы данных;
- Logic.hs – модуль, реализующий логику изменения состояния игрового поля, в зависимости от нажатой клавиши и текущего состояния игры;
- Renderer.hs – модуль, реализующий графический интерфейс;
- State.hs – логика взаимодействия с пользователем до запуска игры;
- Piece.hs – геометрические фигуры, используемые в игре;
- Playfield.hs – игровое поле игры;
- Main.hs – запуск всей программы;

В **Constants.hs** описаны следующие константные значения:

- windowName – название игры “Tetris”;
- windowWidth – ширина окна;
- windowHeight – высота окна;
- windowPosition – координаты окна;
- backgroundColor – цвет окна;
- windowScale – масштаб окна;
- fps – кадровая частота;
- cellSize – размер одного квадрата;
- pieceVelocity – скорость падения фигурки;
- mainColor – цвет игрового поля;
- padding – отступ;
- wellWidth – ширина «стакана»;
- wellHeight – высота «стакана»;
- wallWidth – ширина рамки «стакана»;
- wallHeight – высота рамки «стакана»;
- wellColor – цвет «стакана»;
- wallColor – цвет рамки «стакана».

В программе описаны следующие типы:

- data Cell – ячейка - это прямоугольник на игровом поле, он может быть как полным, так и пустым;
- data Row – ряд ячеек;

- data Well – «стакан»;
- type Record – результат игры;
- type Records – все результаты игры;
- data Screen – структура состояний игры;
- data State – основная структура.

В **DBase.hs** описаны следующие функции:

- createNewUser – создает пользователя, если его нет, возвращает id;
- getIDFromResult – преобразует типы;
- getConnection – создает таблицы, записывает результаты;
- someFunc – создает/разрывает соединение;
- writeResultToTable – пишет результат в таблицу рекордов;
- getListFromQuery – разбирает массив, возвращает его таблицей;
- getTop10 – получает первые 10 результатов из таблицы рекордов.

В **Logic.hs** реализованы следующие функции:

- effectivePiecePeriod – время ожидания перед сбросом следующей фигуры;
- handleEvent – инициализация четырех состояний: NameField, Menu, Table и Game;
- handleText – изменяет имя пользователя и создает нового;
- deleteLast – удаляет предыдущее имя пользователя;
- handleMenu – переключает пункты меню;
- handleTable – переходит из таблицы результатов в меню;
- handleGame – перемещает/поворачивает фигуры;
- movePiece – перемещает падающую фигуру по горизонтали;
- transformPiece – перемещает фигуру по игровому полю;
- rotateCW – вращает падающую фигуру по часовой стрелке;
- rotateCCW – вращает падающую фигуру против часовой стрелки;
- updateGameState – переключает между состояниями игры;
- canPieceBeAt – проверяет возможность перемещения фигуры;
- applyMove – перемещает текущую фигуру на одну ячейку вниз;
- fixPiece – фиксирует падающую фигуру в ее текущую позицию и сбрасывает новую;
- handleFullRows – удаляет заполненные строки и соответствующим образом изменяет оценку;
- linesToScore – система подсчета очков.

В **Renderer.hs** реализованы следующие функции:

- `playfieldToScreen` – преобразует координаты игрового поля в координаты экрана;
- `renderCell` – отображает один квадратик;
- `renderWell` – визуализация игрового поля в изображение;
- `render` – визуализация текущего состояния игры;
- `drawNameField` – визуализация заполнения имени пользователя;
- `drawTextField` – заполнение имени пользователя;
- `renderTable` – визуализация таблицы результатов;
- `renderMenu` – визуализация меню;
- `drawTableElements` – заполнение таблицы результатов;
- `tableTiles` – прорисовка таблицы результатов;
- `drawMenuElements` – прорисовка меню;
- `renderGame` – визуализация всей игры с состояниями.

В **State.hs** реализованы следующие функции:

- `initialGameState` – начальное состояние игры;
- `resetGameState` – сброс состояния игры.

В **Piece.hs** реализованы следующие функции:

- `tetrominoI`, `tetrominoO`, `tetrominoS`, `tetrominoZ`, `tetrominoT`, `tetrominoJ`, `tetrominoL` – формируют падающие фигуры;
- `pieceContains` – проверяет, содержит ли фигура заданную координату;
- `pieceToAA` – преобразует произведение в строку `ascii-art`;
- `pieceCW` – вращает фигуру по часовой стрелке;
- `pieceCCW` – вращает фигуру против часовой стрелки;
- `validPos` – проверяет, допустима ли позиция для фигуры относительно границ «стакана»;
- `randomPiece` – преобразует случайное число в фигуру;
- `pieceColor` – возвращает цвет конкретной фигуры.

В **Playfield.hs** реализованы следующие функции:

- `cellColor` – возвращает цвет ячейки или черный, если она пуста;
- `emptyRow` – создание пустой строки;
- `emptyWell` – создание пустого «стакана»;
- `renderPiece` – визуализирует фигуру в «стакане»;

- `pieceCollides` – визуализирует внутренность «стакана»;
- `clearAndCountFilledRows` – очищает «стакан» от заполненных строк, накапливает очки.

В **Main.hs** реализованы следующие функции:

- `main` – запускает всю программу.

## **ИСПОЛЬЗУЕМЫЕ БИБЛИОТЕКИ**

При реализации программы были использованы следующие библиотеки:

- gloss – для создания графического интерфейса и взаимодействия с пользователем;
- random – генерация случайных чисел;
- Database.HDBC.Sqlite3 – подключение SQLite3.

## СЦЕНАРИИ РАБОТЫ С ПРИЛОЖЕНИЕМ

При запуске приложения пользователь может выбрать в меню пункт «Play» и начать игру или сменить имя игрока, выбрав в меню пункт «Change your name». По окончании игры можно просмотреть лучшие результаты, выбрав в меню пункт «Top 10 players».

Все взаимодействия с пользователем осуществляются с помощью интерфейса клавиатуры.

### Скриншоты игры:

