



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
М. В. ЛОМОНОСОВА

Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Отчёт о выполнении задания практикума

«Игра Реверси»

Студент 324 группы
А. А. Яндиев

Москва, 2022

1 Постановка задачи

В данной программе реализована известная настольная игра Реверси.

Правила игры:

- Первый ход принадлежит игроку, которому достались белые фишки. Следующую игру начинает игрок с черными фишками и т. д.;
- Всего используемых фишек 64. Фишки меняют цвет. Изначально на поле находятся 4 фишки (2 черных и 2 белых) - остальные 60 фишек находятся за пределами поля и выставляются поочередно;
- Сделать ход на поле означает взятой вне поля фишкой закрыть проход сопернику, выставляя ее таким образом, чтобы между выставленной и какой-либо другой фишкой игрока образовался туннель, заполненный дисками противника (за один ход можно выставить на поле только один диск);
- Все побитые диски с поля боя забирает ходивший игрок, т. е. меняет фишки противника на свои. При этом меняются диски соперника как по прямой, так по диагонали;
- Если у игроков есть возможность ходить фишками, то пропуск хода невозможен. Если же все ходы перекрыты, то участник пропускает ход;
- Игра продолжается до тех пор, пока все диски противника не будут побиты, или на доске не останется свободных клеток.

В программе реализованы все пункты правил игры.

1.1 Базовые требования и основной функционал

1. Предоставлена возможность играть в игру в режиме «человек против человека», используя графический интерфейс;
2. Не допускаются невозможные по правилам игры ходы игроков;
3. Ведется счет игровых очков на всем протяжении игры;
4. Определяется и отображается очередность ходов игроков (и в том случае, когда у одного из игроков нет доступных ходов);
5. Автоматически определяется момент окончания игры: победы одного из игроков или ничьей. Результат демонстрируется пользователям.
6. В конце игры пользователям предлагается начать новую партию, нажав на клавишу пробела.

2 Модули проекта

Проект состоит из следующих модулей:

- `Consts.hs` - константы для настроек игры;
- `Types.hs` - объявление основных типов;
- `Board.hs` - работа с игровым полем;
- `Moves.hs` - обработка ходов игроков;
- `Draw.hs` - реализация графического интерфейса;
- `Events.hs` - реализация обработчиков внешних событий;
- `InitGame.hs` - начальное состояние игры;
- `Game.hs` - запуск игры;
- `Main.hs` - главный модуль приложения.

В модуле `Consts.hs` описаны следующие константные значения:

- `gridSize` - размер игрового поля;
- `windowSizeX`, `windowSizeY` - размеры игрового окна;
- `startScore` - начальный счет игры;
- `boardColor` - цвет игрового поля;
- и т. д.

В модуле `Types.hs` описаны следующие типы:

- `Coordinate` - описывает координаты сетки игрового поля;
- `Score` - описывает текущий счет игры;
- `Player` - описывает цвет дисков игрока;
- `Board` - описывает игровое поле;
- `GameWorld` - описывает игровой мир.

В модуле `Board.hs` реализованы следующие функции:

- `initBoard` - начальное состояние игрового поля;
- `mapBoardSquares` - применяет заданную функцию к каждой клетке игрового поля;
- `coordinateInBounds` - проверяет, находится ли заданная координата клетки в сетке игрового поля;
- `diskAtCoordinate` - возвращает цвет диска заданной клетки поля;
- `opposingPlayer` - возвращает оппонента заданного игрока.

В модуле `Moves.hs` реализованы следующие функции:

- `getMovesOnClick` - определяет список координат с новыми фишками игрока после хода;
- `movesAvailableForPlayer` - определяет список всех доступных ходов игрока;
- `noMovesAvailableForBoth` - проверяет, доступен ли ход хотя бы для одного из игроков;
- `applyMove` - применяет ход игрока.

В модуле `Draw.hs` реализованы следующие функции:

- `drawGame` - объединяет все элементы графического интерфейса;
- `drawGrid` - отрисовывает сетку игрового поля;
- `drawDisks` - отрисовывает текущие диски игроков на поле;
- `drawScore` - выводит текущий счет игры;
- `drawTurnText` - выводит текст об очередности ходов;
- `drawGameOver` - выводит текст об окончании игры.

В модуле `Events.hs` реализованы следующие функции:

- `handleEvent` - обрабатывает все внешние события;
- `checkPlayerTurn` - проверяет допустимость совершенного хода и применяет его.

В модуле `InitGame.hs` реализована следующая функция:

- `initGame` - определяет начальное состояние игрового поля.

В модуле `Game.hs` реализована следующая функция:

- `runGame` - запускает игру.

В модуле `Main.hs` реализована следующая функция:

- `main` - основная функция, с которой начинается работа всего приложения.

3 Используемые библиотеки

При реализации использовались следующие библиотеки:

- Gloss - графический интерфейс и обработка внешних событий;
- Array - работа с массивами.

4 Сценарии работы с приложением

Для запуска приложения достаточно перейти в корень папки проекта и запустить приложение в консоли с помощью команды **stack run**.

В начале игры пользователям отображается начальное состояние игрового поля, установленное классическими правилами:

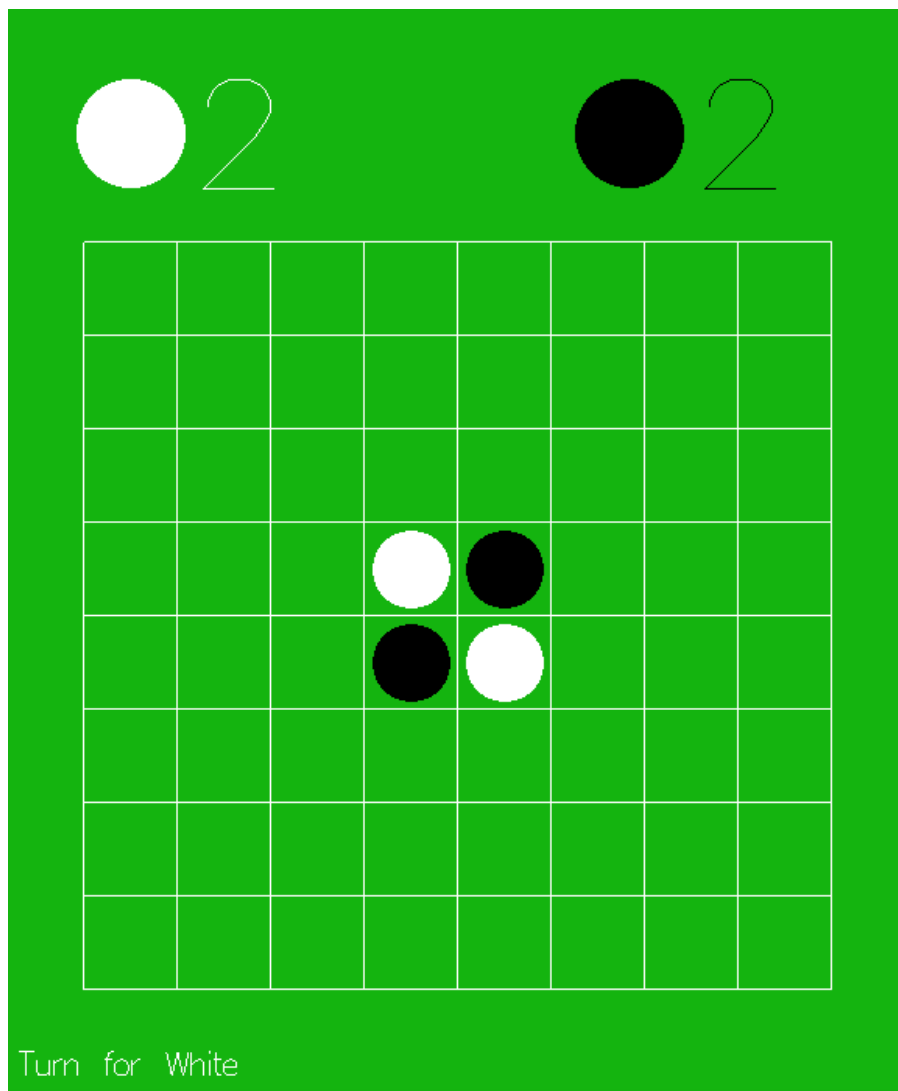


Рис. 1: Начальное состояние игры

Для совершения хода игроку необходимо нажать на соответствующую клетку игрового поля левой кнопкой мыши. Если ход доступен, то он будет выполнен, иначе ничего не произойдет. Таким образом, состояние игрового поля не изменится до тех пор, пока текущий игрок не совершит корректный ход.

В нижнем левом углу игрового окна после каждого хода обновляется информация об очередности ходов игроков:



Turn for White

Рис. 2: Ход белого игрока



No moves available for Black, turn for White

Рис. 3: Пропуск хода черным игроком

На протяжении всей игры пользователям отображается текущий счет противостояния:



○42 ●21

Рис. 4: Счет игры

В конце игры пользователям отображается информация о победителе. А также предлагается начать новую игру: для этого следует нажать клавишу пробела:

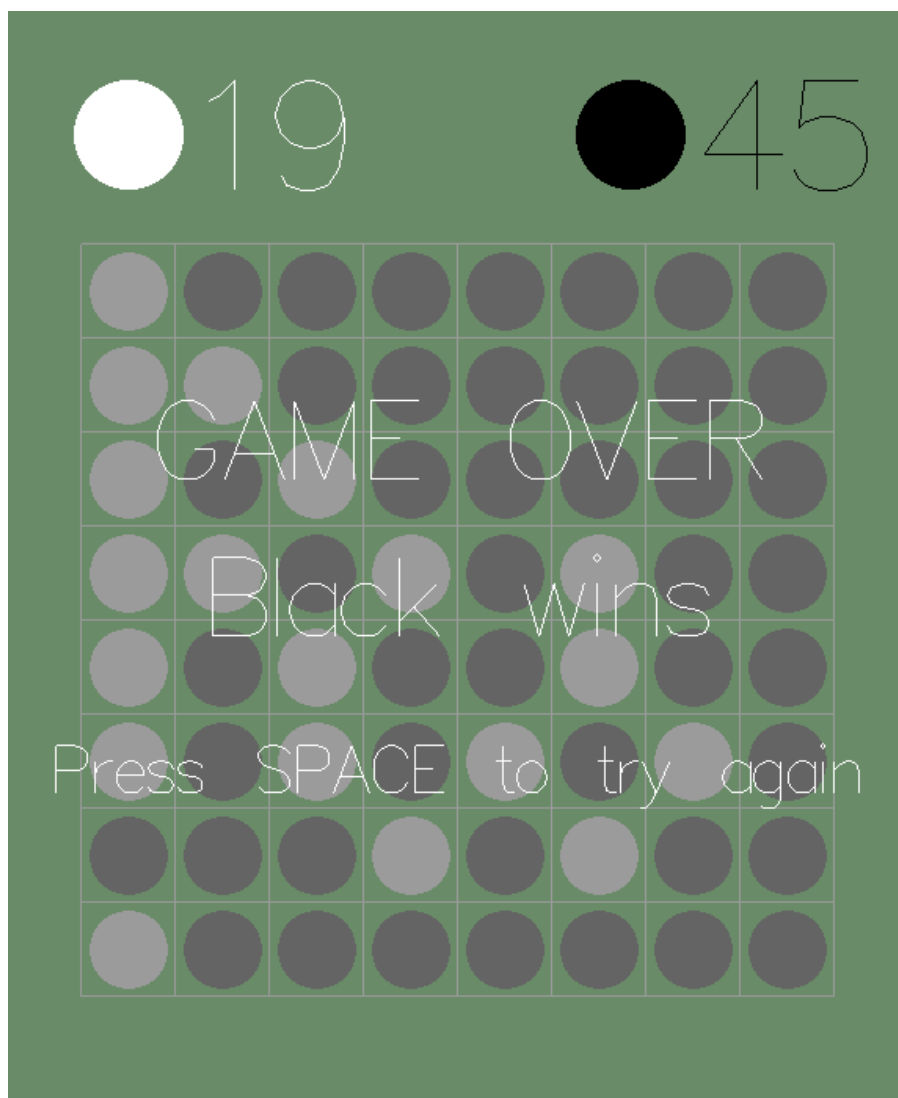


Рис. 5: Конец игры