



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
М. В. ЛОМОНОСОВА

Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Отчёт о выполнении задания практикума

Wordle

Студента 324 группы
И. Ю. Грибова

Москва, 2023

1 Постановка задачи

Wordle – это игра, где нужно угадать слово из пяти букв за шесть попыток. В начале игры необходимо вписать любое слово в верхний ряд игрового поля, представляющего собой 6 рядов по 5 пустых ячеек в каждом. Если хотя бы одна буква в этом слове соответствует загаданному, то она подсветится зелёным, когда стоит в нужном месте, и жёлтым, когда стоит не на месте. Если все ячейки вашего слова остаются серыми, значит, в искомом слове вообще нет этих букв.

Игрок побеждает в том случае, если сможет за 6 попыток отгадать это слова, и проигрывает в противном случае. В случае проигрыша игроку высвечивается загаданное слово.

1.1 Базовые требования

Базовые требования реализации:

1. Загрузка необходимых данных и обработка ошибок.
2. Интерфейс и графика для самой игры.
3. Определение проигрыша и выигрыша в игре.

1.2 Индивидуальные части

1. Описание правил для игры (чтобы пользователь не запутался, как вводить или удалять что-то).
2. Вывод правильного слова в случае проигрыша в игре или поздравления в случае выигрыша.
3. Вступление для игры при запуске программы.

2 Модули проекта

Проект состоит из следующих модулей:

- **Constants.hs** — Константы для настроек игры;
- **Game.hs** — Инициализация самой игры;
- **Gameplay.hs** — Реализация графического представления и обработка действия пользователя;

- **Preprocessing.hs** — Предобработка ввода пользователя и базы данных для игры;
- **Types.hs** — объявление основных типов программы;

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

- `cellHeight`, `cellWidth`, `step`, `stepLetHeight`, `maxWidth` и другие — задающие размеры отрисовки игрового поля;
- `minLenDict` — минимальный размер базы данных для игры;
- `wordLength` — длина отгадываемого слова;
- `greenCode`, `yellowCode`, `grayCode` — цвета для закрашивания введенных пользователем букв;
- `alphabet` — допустимый для пользователя ввод;
- `rightWordCode` — корректная последовательность кода, введенного пользователем слова, для победы в игре;
- и так далее...

В модуле **Types.hs** описаны следующие типы:

- `GameState` — описывает текущее состояние игры;
- `History` — описывает введенные пользователем слова и буквы;
- `Dictionary` — описывает словарь возможных для отгадывания слов;
- `FileName`, `CodeAnswer`, `UserWord` — синонимы `String` для понимания кода и логики функций.

В модуле **Game.hs** реализованы следующие функции:

- `startGame` и `continueGame` — Для инициализации и запуска игры, а также обработки возможных ошибок.

В модуле **Gameplay.hs** реализованы следующие функции:

- `drawBoard` — для отрисовки игрового поля.
- `drawLetters` — для отрисовки введенных пользователем букв.
- `drawState` — для отрисовки текущего состояния игры.

- `upgradeStateLet` — для обновления Истории ввода, когда пользователь ввел букву.
- `delStateLet` — для обновления Истории ввода, когда пользователь удалил букву.
- `compareWordInput` — для сравнения введенного пользователем слова и слова для отгадывания.
- `playGame` — для запуска игры.
- `handleInput` — для обработки ввода пользователя.
- `checkWordState` — для обработки введенного пользователем слова в рамках одной попытки.

В модуле **Preprocessing.hs** реализованы следующие функции:

- `checkReadFile` — проверка на наличие файла.
- `readDictionary` — чтение и обработка введенной базы данных со словами.
- `getRandomIndex` — получение случайного индекса из базы данных для загадывания слова.
- `randomWord` — получение слова по индексу.
- `checkGuessWord` — обработчик ошибок связанных с полученными данными.

3 Используемые библиотеки

При реализации использовались следующие библиотеки:

- `gloss` — графическое поле для игры и интерфейс для прогресса, а также обработка введенных пользователем букв и описание правил для ввода текста.
- `random` — генерация позиции слова из базы для отгадывания.
- `Data.List.Split` и `Data.Char` — для обработки полученных данных.
- `System.IO` — для взаимодействия с пользователем.
- `Control.Exception.Base` — для обработки ошибок.

4 Сценарии работы с приложением

Как же все-таки поиграть в эту замечательную игру (руководство):

1. Нужно скачать репозиторий к себе на устройство при помощи команды **git clone** (рекомендуется по SSH, но для этого надо сначала настроить подключение по SSH к GitHub).
2. Перейдите к каталогу, куда вы скачали репозиторий и введите команду для сборки игры **stack build** и команду для запуска игры **stack exec wordle-exe**
3. Далее введите имя файла, где собрана ваша база данных для игры, после этого запустится само игровое поле с описанием, как вводить слова. При любых ошибках (нет такого файла или недостаточно слов для игры (надо как минимум 7 слов длины 5)), вам придется запустить игру заново с помощью **stack exec wordle-exe**.
4. Have a nice time while playing!