

# Документация к проекту "Тетрис"

kimikodo

10 сентября 2024 г.

## 1 Введение

Введение в документацию к проекту "Тетрис".

## 2 Структура проекта

Структура проекта "Тетрис" следующая:

- `src` - каталог с исходным кодом
- `src/include` - каталог с заголовочными файлами
- `src/brick_game` - каталог с реализацией игры
- `src/brick_game/tetris` - каталог с реализацией логики игры
- `src/gui` - каталог с реализацией интерфейса
- `src/gui/cli` - каталог с реализацией интерфейса командной строки
- `test_tetris` - каталог с тестами
- `doc` - каталог с документацией

## 3 Компиляция проекта

`make` - сборка проекта "Тетрис". Установка будет вестись в каталог `brick_game`, который будет создан вне папки `src`

Компиляция проекта "Тетрис" происходит с помощью `Makefile`. Компиляция производится с помощью следующих шагов:

- `all` - основная цель, собирает и устанавливает проект.

- `clean` - удаляет все объектные файлы, исполняемые файлы и другие временные файлы.
- `install` - устанавливает проект в указанную директорию.
- `uninstall` - удаляет установленный проект.
- `format-check` - проверяет форматирование кода.
- `format` - форматирует код в соответствии с заданными правилами.
- `cppcheck` - проверяет код на ошибки с помощью утилиты `cppcheck`.
- `valgrind` - запускает тесты с использованием утилиты `valgrind`.
- `leaks` - проверяет наличие утечек памяти.
- `coverage_flag` - добавляет флаги для сбора информации о покрытии кода.
- `sanitize` - запускает проект с включенными флагами санитизации.
- `sanitize_flag` - добавляет флаги санитизации.
- `gcov_report` - собирает отчет о покрытии кода с помощью утилиты `gcovr`.
- `test` - запускает тесты проекта.
- `s21_tetris.a` - собирает статическую библиотеку `s21_tetris.a`.
- `debug` - добавляет флаги для сборки в режиме отладки.
- `dist` - создает архив проекта.

## 4 Интерфейс

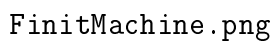
Интерфейс проекта "Тетрис" реализован с помощью библиотеки `ncurses`. Интерфейс состоит из следующих компонентов:

- `StartMenu` - меню, которое отображается при запуске программы
- `GameMenu` - интерфейс игры, который отображается, когда пользователь начинает играть
- `EndMenu` - интерфейс, который отображается, когда пользователь заканчивает играть
-

## 5 Логика игры

Логика игры "Тетрис" реализована с помощью конечного автомата. Конечный автомат имеет следующие состояния:

- **NewGame** - состояние, в котором игра ожидает, пока пользователь не нажмет кнопку "Enter"
- **SpawnFigure** - состояние, в котором игра генерирует новую фигуру
- **Refresh** - состояние, в котором игра отображает фигуру и ожидает, пока пользователь не нажмет кнопку
- **Paused** - состояние, в котором игра приостановлена пользователем
- **SureEndGame** - состояние, в котором игра выводит окно с подтверждением выхода из игры. Нужно для предотвращения случайного нажатия
- 
- **EndGame** - состояние, в котором игра заканчивается



The diagram area is mostly blank, with the text 'FinitMachine.png' centered. It likely contains a state transition diagram for the Tetris game logic, showing states like NewGame, SpawnFigure, Refresh, Paused, SureEndGame, and EndGame and the transitions between them based on user input.

FinitMachine.png

Рис. 1: Конечный автомат

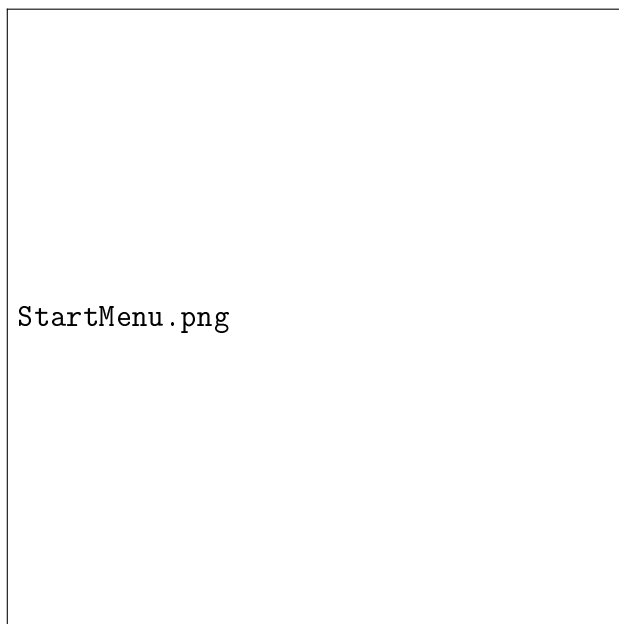


Рис. 2: Начальный интерфейс

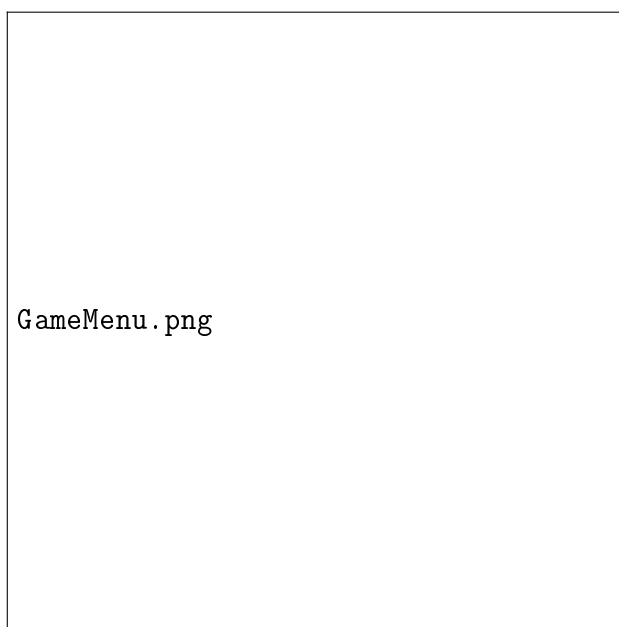
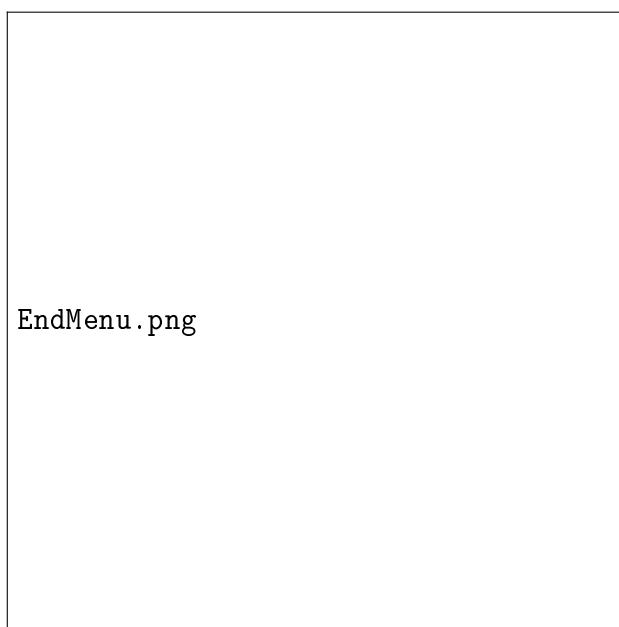


Рис. 3: Интерфейс игры



EndMenu.png

Рис. 4: Завершающий интерфейс