

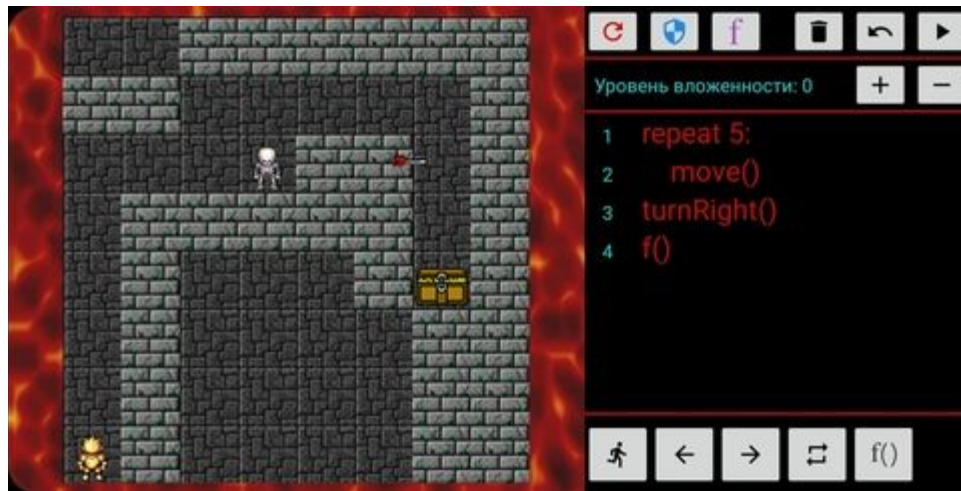
Создание мобильной игры, обучающей основам программирования

Выполнена студентом группы ИУ9-52Б
Пичугиным Владиславом

Москва, 2019

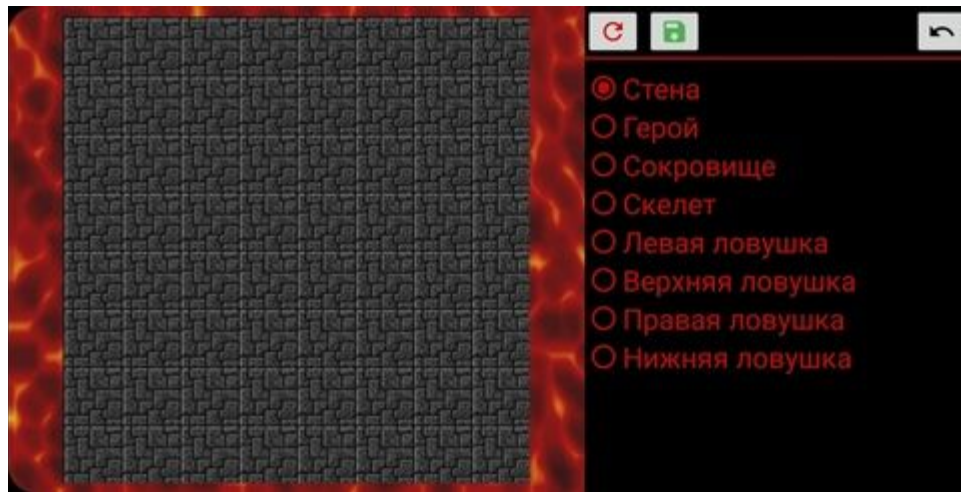
Основные идеи

- Игровой режим
 - Визуализация игрового поля с помощью конфигурационного JSON-файла;
 - Анимированное перемещение некоторых игровых объектов;
 - Редактор кода для игрового языка программирования;
 - Интерпретатор, приводящий в движение игрового персонажа.



Основные идеи

- Режим конструктора уровней
 - Возможность касанием добавить объект на игровое поле;
 - Редактор для управления созданием уровня;
 - Возможность сохранения конфигурации созданного уровня в формате JSON.



Обзор инструментов Android для создания игры

- Activity и Intent
- Fragment (в том числе и DialogFragment)
- View
- Bitmap
- RecyclerView
- Gson
- Assets и Internal Storage
- Toast

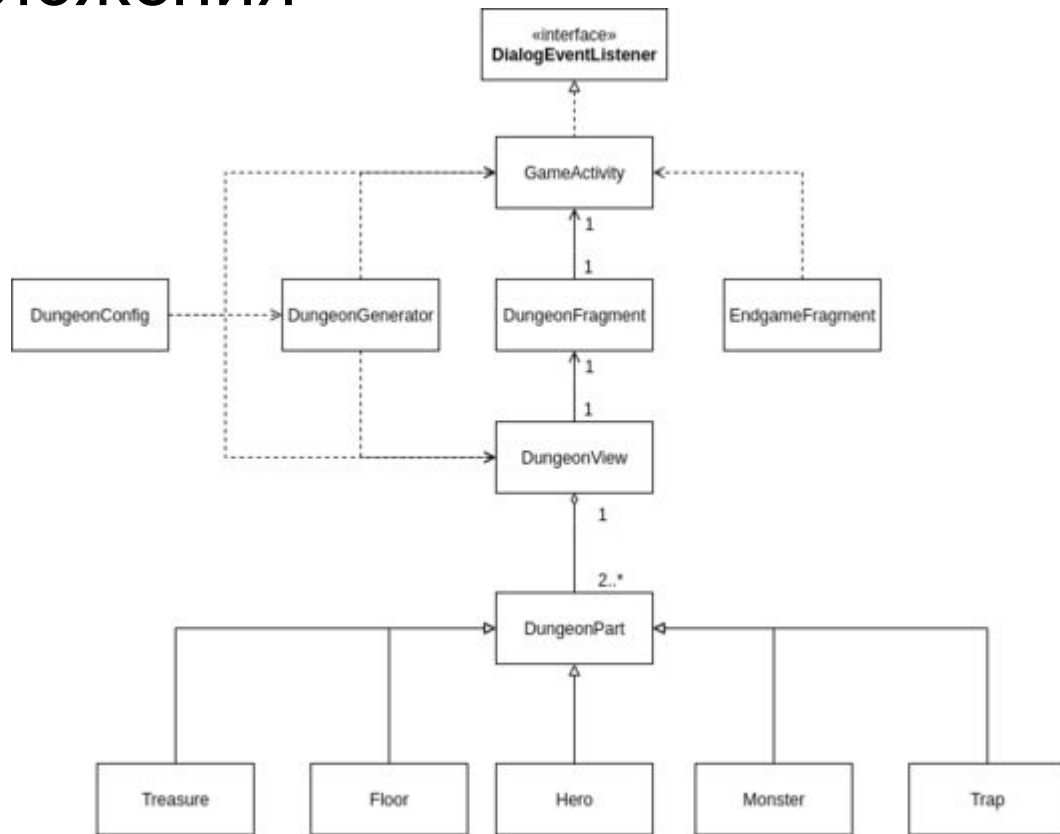


android

Проектирование приложения

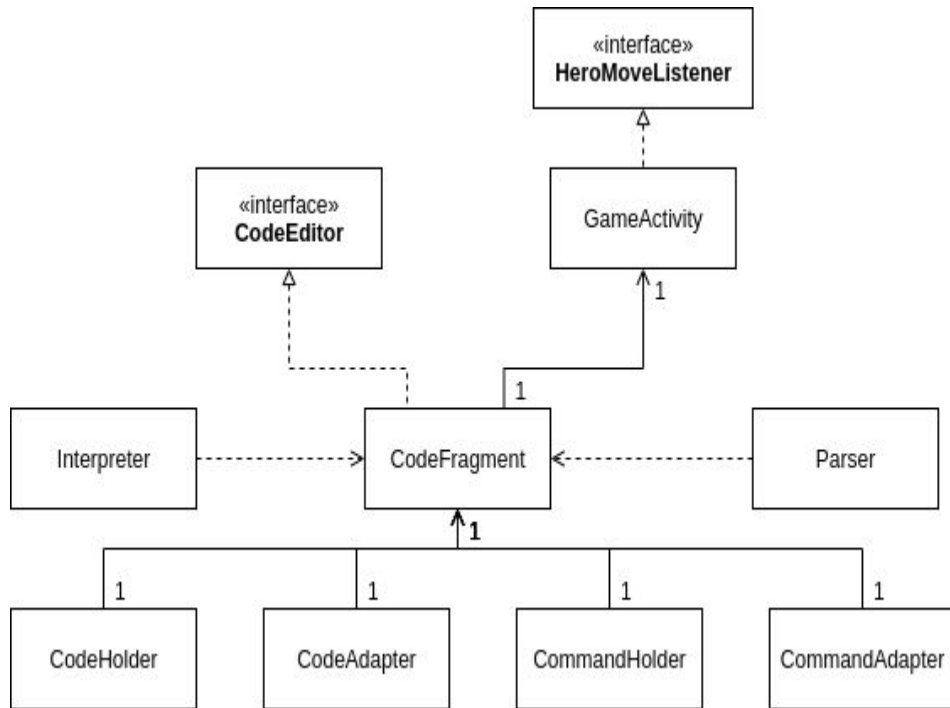
- Основные классы игрового режима, отвечающие за игровое поле:

- `GameActivity`
- `DungeonFragment`
- `DungeonView`
- `DungeonPart` и его наследники
- `DungeonGenerator`
- `DungeonConfig`



Проектирование приложения

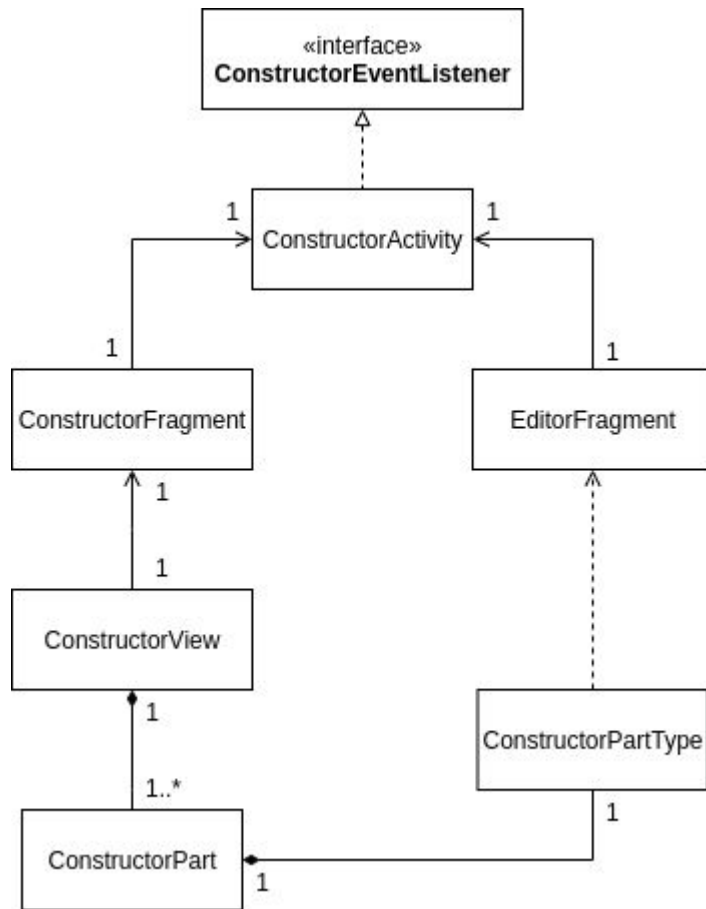
- Основные классы игрового режима, отвечающие за редактор кода:
 - HeroMoveListener
 - GameActivity
 - CodeEditor
 - CodeFragment
 - CodeHolder и CodeAdapter
 - CommandHolder и CommandAdapter
 - Parser
 - Interpreter



Проектирование приложения

- Основные классы режима конструктора уровней:

- ConstructorEventListener
- ConstructorActivity
- ConstructorFragment
- ConstructorView
- ConstructorPart
- EditorFragment



Проектирование приложения

- Не менее важные для работы приложения классы:
 - StartActivity
 - LevelsActivity
 - JsonRepo
 - DncApplication

Визуализация поля в игровом режиме

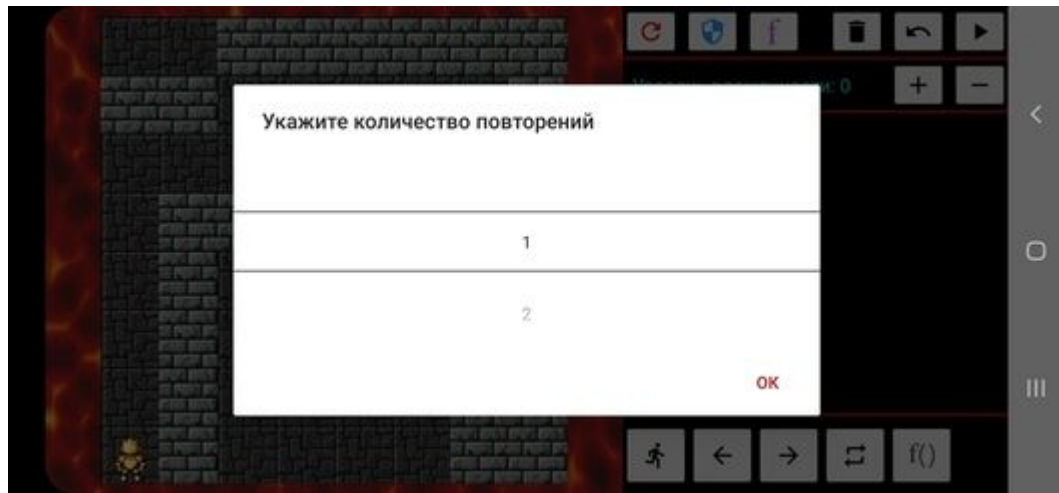
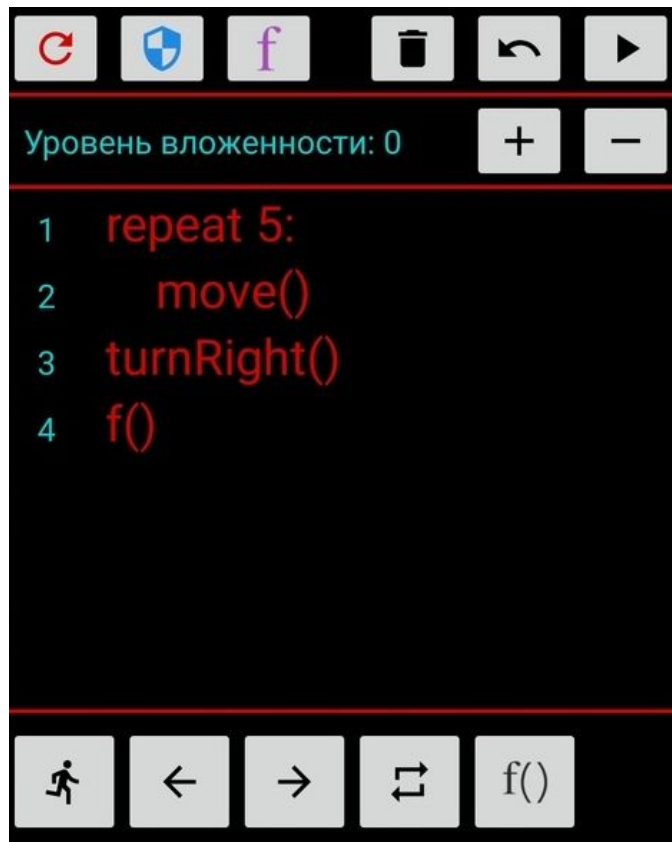
- Общая схема работы:
 - LevelsActivity открывает GameActivity, передавая объект Intent с информацией об уровне;
 - GameActivity с помощью JsonRepo получает конфигурационную JSON-строку;
 - С помощью DungeonGenerator JSON-строка преобразуется в объект DungeonConfig;
 - На основании размера экрана мобильного устройства создается матрица стен;
 - Создаются и позиционируются все остальные игровые объекты;
 - Происходит отрисовка игрового поля.



Игровой язык программирования

- Создан для демонстрации базовых возможностей программирования;
- Для выделения групп команд использует уровни вложенности;
- Поддерживает следующие команды:
 - `move()`
 - `turnLeft()`, `turnRight()`
 - `repeat`
 - `f()`
 - `if`, `elif`, `else`
 - `dodgeLeft()`, `dodgeTop()`, `dodgeRight()`, `dodgeBottom()`

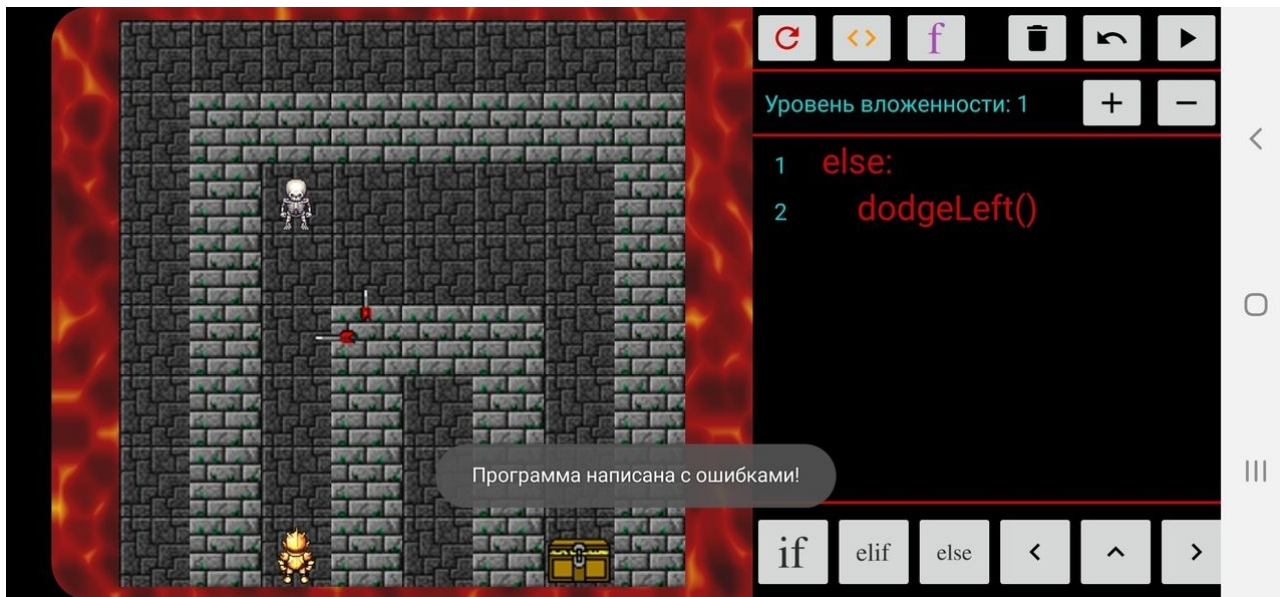
Редактор кода



Синтаксическая проверка игровой программы

- Грамматика для основного кода и для кода подпрограммы:
 - `<Program>::=<PrimCom><Program>|<Repeat><Program>|e`
 - `<PrimCom>::=move()|turnLeft()|turnRight()|f()`
 - `<Repeat>::=repeat <Program>`
- Грамматика для кода защитного скрипта:
 - `<Program>::=<PrimCom><Program>|<Cond><Program>|e`
 - `<PrimCom>::=dodgeLeft()|dodgeTop()|dodgeRight()|dodgeBottom()`
 - `<Cond>::=if <Program'><Cond'>`
 - `<Program'>::=<PrimCom><Program'>|e`
 - `<Cond'>::=elif <Program'><Cond'>|<Cond''>`
 - `<Cond''>::=else <Program'>|e`

Синтаксическая проверка игровой программы



Особенности интерпретации игрового кода

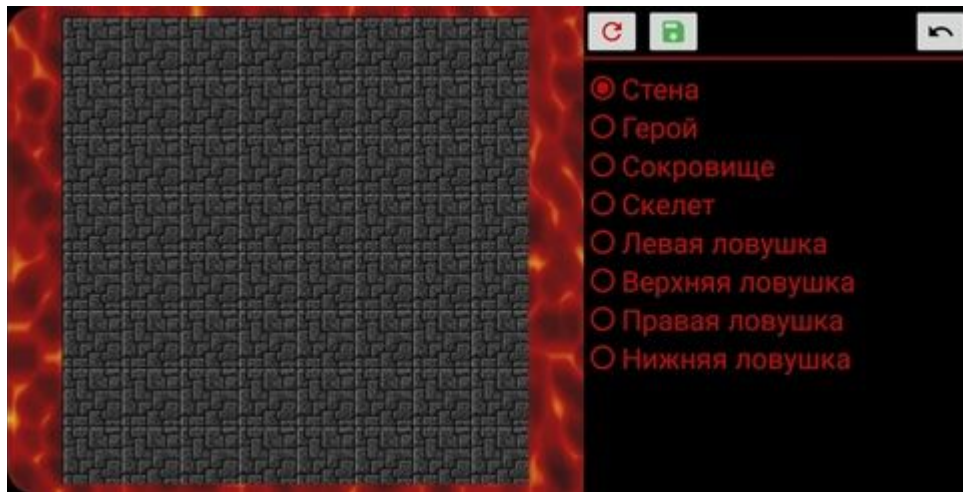
- Управление игровым полем через переменную типа HeroMoveListener;
- Построен на рекурсии;
- Ожидает завершения анимации перемещения игрового персонажа перед выполнением следующей инструкции;
- Корректно обрабатывает вложенные циклы, вызов подпрограммы и рекурсию;
- Анализирует защитный скрипт при попадании персонажа на одну клетку с ловушкой.

Реализация анимаций

- Перед началом анимации персонажа формируется объект HeroActions, содержащий набор колбэков для отдельных этапов перемещения;
- Используется класс ValueAnimator;
- Основные действия происходят в обработчике событий обновления значений объекта ValueAnimator.

Реализация конструктора уровней

- Добавление игровых объектов с помощью нажатия на игровое поле;
- стек с историей добавления элементов;
- Сохранение конфигурации созданного уровня в формате JSON.



Заключение

- Разработана игра под ОС Android, позволяющая пользователям познакомиться с некоторыми базовыми идеями программирования;
- В ходе выполнения работы были реализованы алгоритмы визуализации игрового поля и интерпретации игрового кода;
- Был создан пользовательский интерфейс, позволяющий:
 - редактировать игровой код;
 - создавать собственный игровой уровень.