Бюджетное общеобразовательное учреждение

Города Омска «Лицей № 149»

Проект

Java Dungeon

Подготовил ученик:

Арефьев Вениамин Андреевич

Научный руководитель:

Преподаватель БОУ г. Омска «Лицей № 149»

Лапенко Марина Васильевна

Оглавление

[**Вступление** 3](#_Toc509518070)

[**Задачи** 4](#_Toc509518071)

[**Планы выполнения проекта** 5](#_Toc509518072)

[**ООП и его 4 основных столба** 7](#_Toc509518073)

[**Идейная составляющая игры** 8](#_Toc509518074)

[**Структура игры** 11](#_Toc509518075)

[**Заключение** 12](#_Toc509518076)

[**Приложения** 13](#_Toc509518077)

[**Список использованной литературы** 21](#_Toc509518078)

# **Вступление**

Цель проекта: изучить язык ООП(Java), написать roguelike игру и сделать презентацию.

Предполагаемые продукты проекта: исполняемый файл программы (с расширением .jar), презентация.

# **Задачи**

1. Понять, что такое ООП и изучить 4 основных столба
2. Изучить Java
3. Составить примерную схему взаимодействия классов
4. Реализовать игру
5. Упаковать всю игру в архив с расширением jar
6. Подготовить презентацию к проекту

# **Планы выполнения проекта**

1. Понять, что такое ООП и изучить 4 основных столба
2. Изучить Java
   1. Понять, что такое классы, поля, методы и локальные переменные
   2. Узнать, что такое интерфейсы, анонимные классы, вложенные классы
   3. Узнать о модификаторах приватности классов, методов и полей, так же узнать о геттерах и сеттерах
3. Составить примерную схему взаимодействия классов
4. Реализовать игру
   1. Реализовать стартовое окно и добавить в него 2 кнопки (играть и настройки)
   2. Реализовать окно настроек
      1. Сделать 3 сложно с помощью флажков
      2. Сделать окно с возможностью задать сид
      3. Добавить Debug mod
      4. Реализовать секретную кнопку (открывающую диалоговое окно для ввода пароля)
         1. Реализовать диалоговое окно и проверку пароля на правильность
      5. Реализовать логгер и специфическую настройку для моей игры
   3. Реализовать окно игры и логику
      1. Реализовать панель с игрой
         1. Взять из класса карту и перерисовать её (20 раз в секунду
         2. Нарисовать текстуры для стен, дорог, игрока, врагов (3 типа и каждому типу по 3 картинки в зависимости от оставшихся жизней)
      2. Реализовать панель с информацией
         1. Сделать панель с сидом(который можно выделить и скопировать)
         2. Сделать панель с информацией о текущем этаже
         3. Сделать кнопочку Debug, с помощью которой можно изменять жизни, урон и защиту игрока
         4. Сделать панель с информацией о характеристиках игрока
         5. Сделать панель с динамично изменяющимся ключиком
         6. Сделать панель с информацией о прогрессе повышения уровня, количестве доступных повышений уровня и кнопку, которая открывает окно для повышения уровня
         7. Сделать суперсекретную кнопку, которая, будет открывать диалоговое окно для ввода пароля и разблокирования кнопочки Debug
      3. Реализовать класс с картой игры(матрицей) и всей логикой игры
         1. Реализовать карту и список врагов
         2. Реализовать сидовую генерацию карты и врагов
         3. Реализовать систему PVE
         4. Реализовать передвижение врагов
         5. Реализовать прогрессию игрока (опыт и передвижение на следующие этажи)
         6. Реализовать прогрессию врагов относительно уровня и сложности игры
         7. Реализовать систему с ключом и замком на люке
         8. Реализовать битву с боссом на 11 уровне
   4. Почистить код от Debug информации, и подготовить его к релизу
5. Упаковать всю игру в архив с расширением jar
6. Подготовить презентацию к проекту

# **ООП и его 4 основных столба**

Объектно-ориентированное программирование (ООП) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Существует 4 основных столба ООП:

1. **Абстракция** – это придание [объекту](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) характеристик, которые чётко определяют его концептуальные границы, отличая от всех других объектов.
2. **Инкапсуляция** – механизм языка, позволяющий ограничить доступ одних компонентов программы к другим.
3. **Наследование** — свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником, дочерним или производным классом.
4. **Полиморфизм** – свойство системы, позволяющее использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта

# **Идейная составляющая игры**

Интерфейс

Как только игрок нажимает кнопку Play(играть), открывается окно Game. В этом окне игрок сразу же видит первый уровень игры слева и информационную панель справа.(Приложения № 4)

На информационной панели можно увидеть:

1. Сид генерации
2. Текущий этаж
3. Характеристики игрока (здоровье (текущее и максимальное), атаку и защиту)
4. Прогресс сбора ключа
5. Прогресс получения нового уровня
6. 2 кнопки, с помощью которых можно изменять характеристики персонажа

О характеристиках персонажа и врагов

У каждого существа (персонажа, врага или босса) есть как минимум 3 характеристики: жизни, урон и защита. Когда какое-либо существо пытается атаковать другое урон вычисляется по формуле: нанесённый урон = атака атакующего – защита атакуемого, но нанесённый урон — это всегда целое число, которое больше или равно 1. Но у главного героя есть ещё одна характеристика (максимальный запас здоровья). Каждый раз, когда наш главный герой переходит на новый уровень, его фактический запас здоровья восполняется до максимального. (Приложение № 4)

Игровой процесс

В начале игры игрок появляется на первом уровне. Чтобы пройти на второй и последующие уровни, он должен убить некоторое количество врагов и тем самым собрать ключ. На уровне всегда есть 8 врагов, у которых всегда есть по кусочку ключа. Как только игрок соберёт ключ полностью (из 8 кусочков), люк (переход на следующий уровень) откроется. (приложения № 9-18)

Как только игрок спрыгнет в люк, он попадёт на следующий уровень и восстановит свои жизни. (Приложение № 5) Но не всё так просто, как кажется на первый взгляд, у врагов тоже есть прогрессия. С каждым уровнем они становятся всё сильнее, так же как игрок. Существует всего 3 типа врагов:

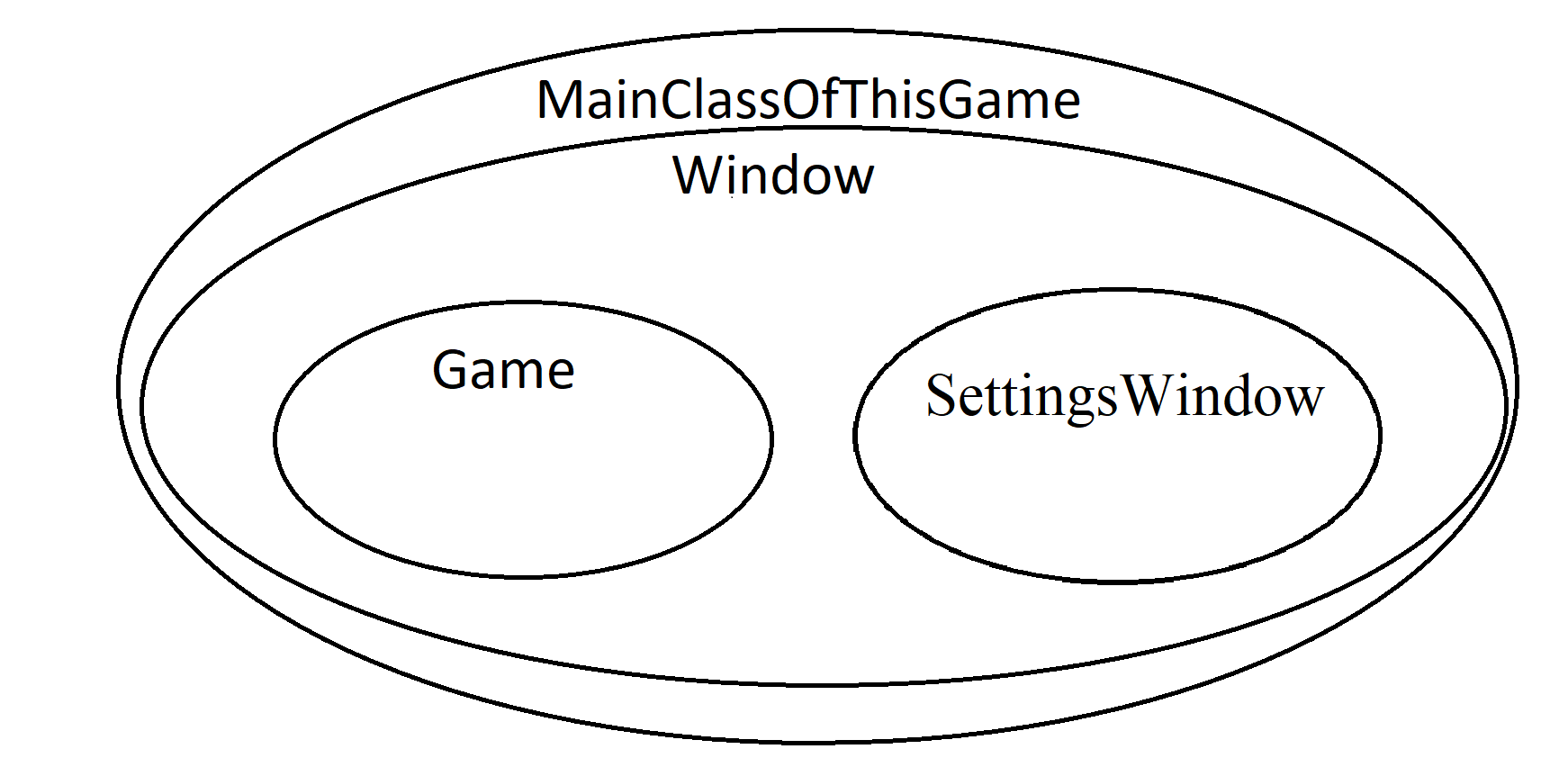
1. Лёгкий тип – зелёные враги
2. Нормальный тип – фиолетовые враги
3. Сложный тип – синие враги

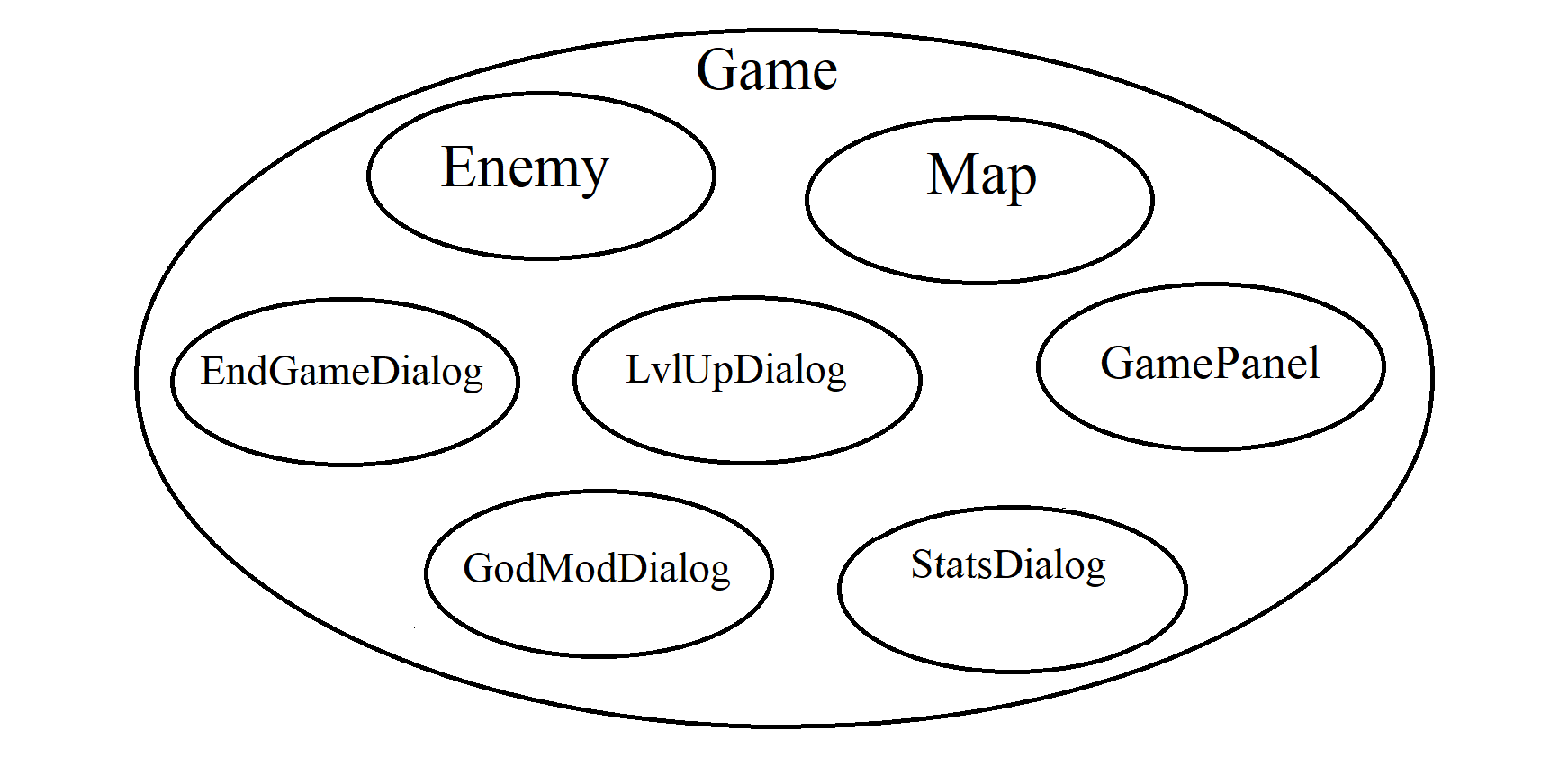
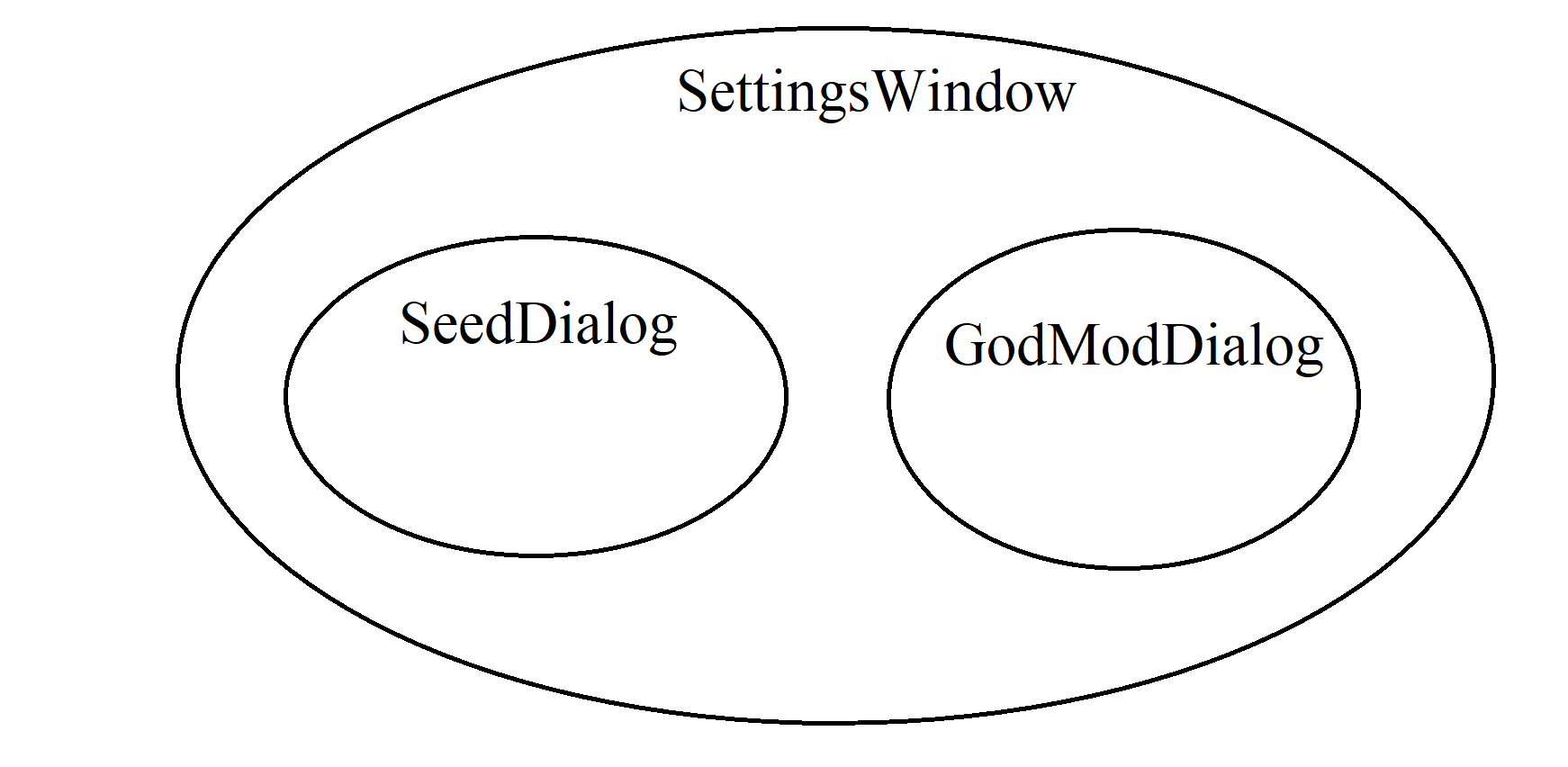
Чем сложнее враг, тем больше у него жизней, защиты и атаки. Существует особый тип врагов – босс. У него очень много жизней, большой запас атаки и здоровья.

Как только игрок сможет преодолеть 10 уровней, он попадёт на 11 уровень с боссом. Как только игрок победит босса, игра становиться пройденной. (Приложение № 6) На протяжении всей игры у игрока есть возможность проиграть, это может произойти тогда, когда запас здоровья игрока достигнет 0. (Приложение № 7).

Каждый раз, убивая какого-либо врага, игрок получает определённое количество опыта. Собирая определённое количество опыта, игрок получает возможность повысить уровень. Об этом свидетельствует активация кнопки Level UP. Как только игрок её нажимает, открывается новое окно. (Приложение № 8). В данном окне игрок может выбрать один из трёх тупей развития: +1 к урону, +1 к защите, + 50 к максимальному здоровью. Как только игрок выбирает что-либо, окошко сразу же закрывается, и применяется выбранный путь развития.

# **Структура игры**





# **Заключение**

Я успешно изучил основные принципы объектно-ориентированного программирования и основы Java. Пользуясь полученными знаниями, я написал свою игру Java Dungeon. В которой есть 3 уровня сложности, процедурно псевдослучайно генерируемая карта с возможностью задания сида, система прокачки персонажа, прогрессия врагов, система PVE (Player Versus Environment), на последней этаже (11), есть битва с боссом.

Продукты проекта: исполняемый файл программы (с расширением .jar), презентация.

# **Приложения**

Приложения № 1. Эскизы ключей

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |

Приложения № 2. Эскизы врагов

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Приложения № 3. Эскизы игрока, босса, дороги и стен

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

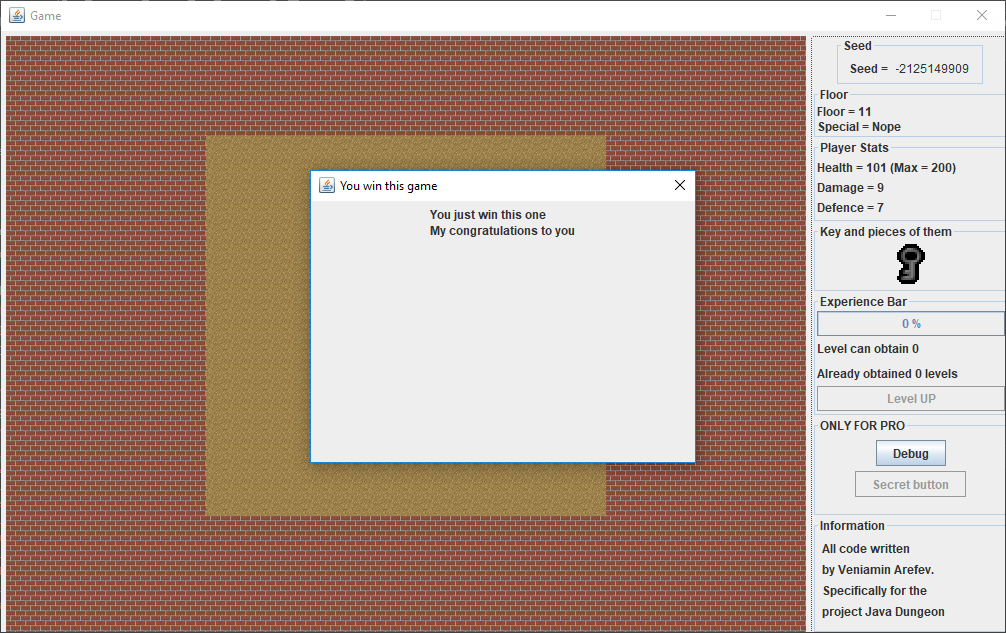
Приложения № 4.



Приложения № 5.



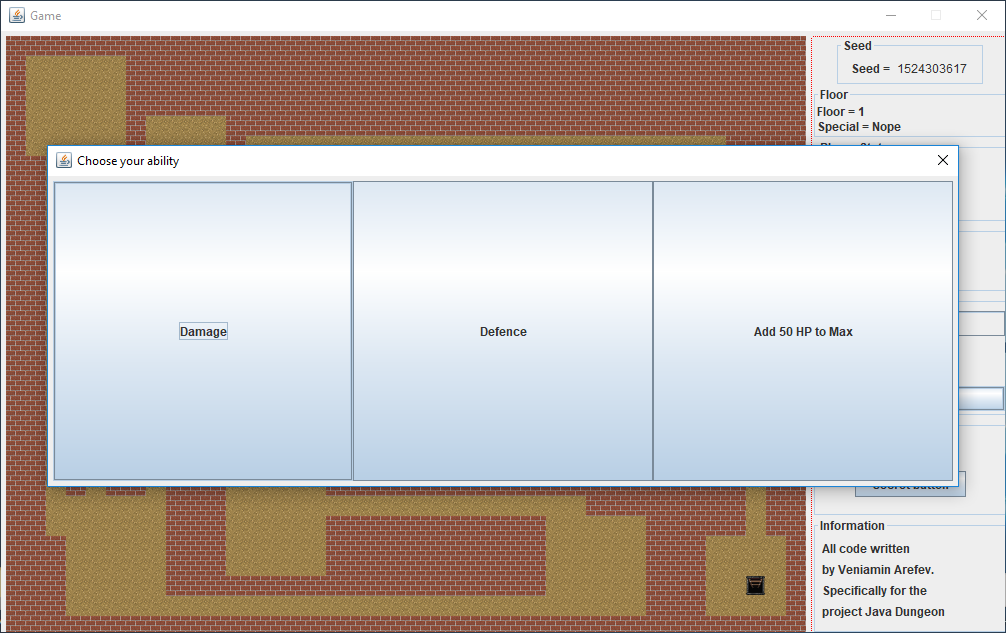
Приложения № 6.



Приложения № 7.



Приложения № 8.



Приложения № 9.



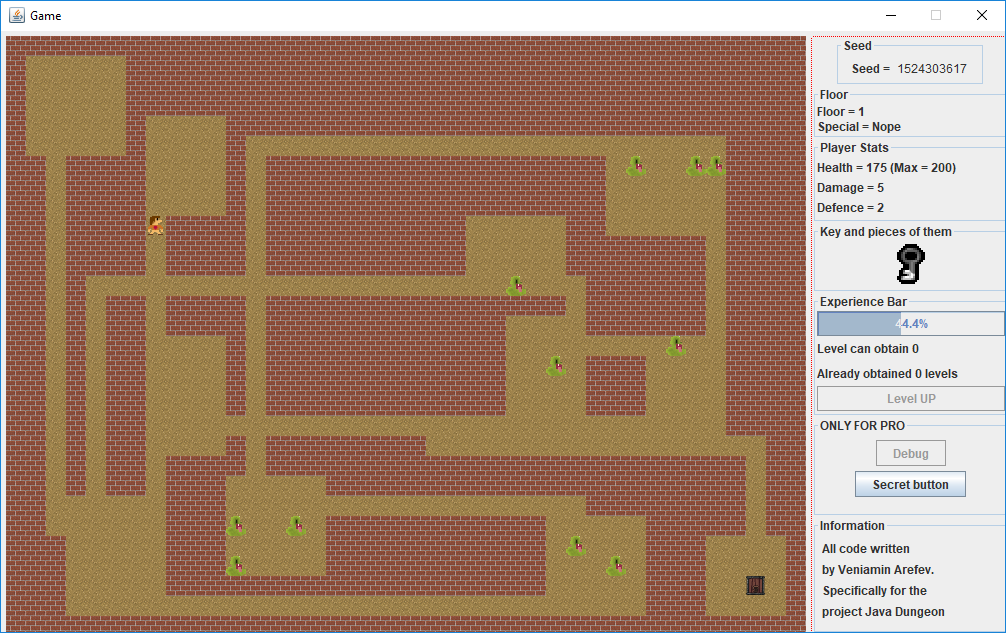
Приложения № 10.



Приложения № 11.



Приложения № 12.



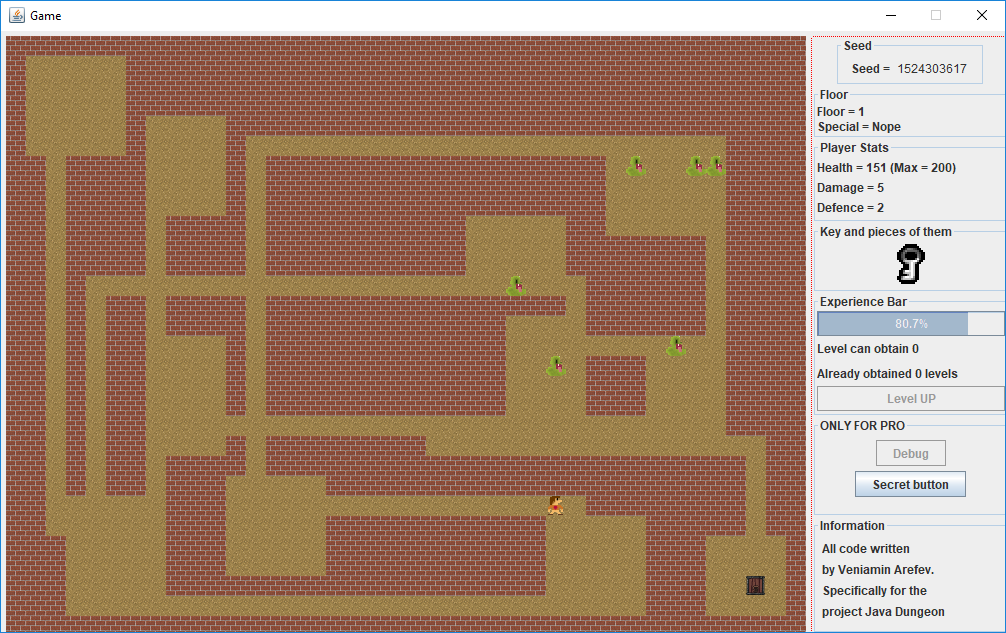
Приложения № 13.



Приложения № 14.



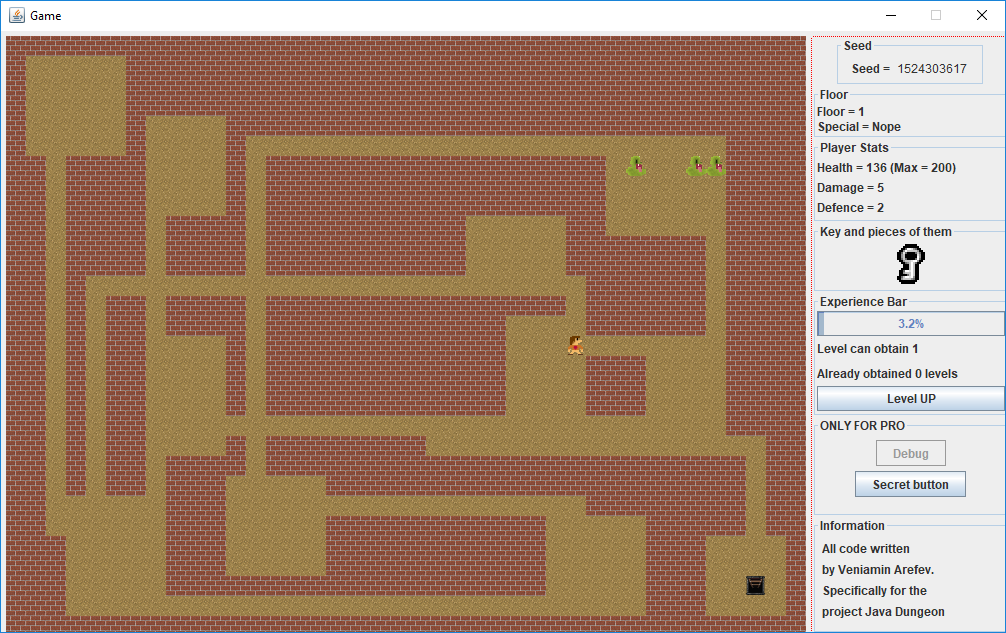
Приложения № 15.



Приложения № 16.



Приложения № 17.



# **Список использованной литературы**

1. Герберт Шилдт, «Java 8. Руководство для начинающих (6 издание)»,2015
2. Java 7: Ильдар Хабибуллин — Москва, БХВ-Петербург, 2012 г.- 768 с.
3. Java EE 6 и сервер приложений GlassFish 3: Дэвид Хеффельфингер — Москва, ДМК Пресс, 2013 г.- 416 с.
4. Java для чайников: Барри Берд — Санкт-Петербург, Вильямс, 2012 г.- 368 с.
5. Java. Библиотека профессионала. Том 1. Основы: Кей С. Хорстманн, Гари Корнелл — Санкт-Петербург, Вильямс, 2015 г.- 864 с.
6. Java. Библиотека профессионала. Том 2. Расширенные средства программирования: Кей С. Хорстманн, Гари Корнелл — Москва, Вильямс, 2014 г.- 1008 с.
7. WEB-программирование на Java и JavaScript: Андрей Гарнаев, Сергей Гарнаев — Санкт-Петербург, БХВ-Петербург, 2005 г.- 1040 с.
8. Интернет-программирование на Java: В.А. Будилов — Санкт-Петербург, Книга по Требованию, 2012 г.- 702 с.
9. Программирование на Java и С# для студента (+ CD-ROM): О. В. Герман, Ю. О Герман — Санкт-Петербург, БХВ-Петербург, 2012 г.- 512 с.
10. Разработка приложений Java EE 6 в NetBeans 7: Дэвид Хеффельфингер — Санкт-Петербург, ДМК Пресс, 2013 г.- 332 с.
11. Современные Java-технологии на практике (+ CD-ROM): Тимур Машнин — Санкт-Петербург, БХВ-Петербург, 2010 г.- 560 с.
12. Создание распределенных приложений на Java 2: И. Хабибуллин — Санкт-Петербург, Книга по Требованию, 2012 г.- 698 с.
13. Язык программирования Java: Создание интерактивных приложений для Internet: Петр Карабин — Санкт-Петербург, Познавательная книга Плю, 2001 г.- 224 с.
14. Герберт Шилдт, «Java 8. Полное руководство 9 издание»,2015
15. Джошуа Блох «Эффективное программирование» 3 издание, 2017
16. Кей Хорстманн, Гари Корнелл «Java. Библиотека профессионала. Том 1. Основы.» 10 издание, 2016
17. Кей Хорстманн, Гари Корнелл «Java. Библиотека профессионала. Том 2. Расширенные средства программирования.» 10 издание, 2016
18. Яков Файн, «Программирование на Java», 4 издание, 2016
19. Кэти Сьерра, Берт Бейтс «Изучаем Java»,2017
20. Кей С. Хорстманн «Java SEE8 вводный курс», 2015
21. Брюс Эккель «Философия Java» 6 издание, 2017
22. Роберт Лафоре «Структуры данных и алгоритмы Java» 3 издание, 2017
23. Роберт Мартин «Чистый код» 2 издание, 2015
24. Brian Goetz «Java Concurrency in Practice», 2017
25. Joshua Bloch and Neal Gafter Java Puzzlers, 2016
26. Роберт Седжвик Алгоритмы на Java, 2015
27. [https://.sun.com](http://www.sun.com)
28. [https:// java-online.ru](http://java-online.ru)
29. <https://ru.wikibooks.org/wiki/Java>
30. [https://habrahabr.ru](https://habrahabr.ru/)
31. <https://www.tutorialspoint.com/java/index.htm>
32. <https://www.oracle.com/index.html>
33. <http://www.datadisk.co.uk/html_docs/java/graphics_java2d.htm>
34. <http://tutorials.jenkov.com/java-logging/logger.html>
35. <http://developer.alexanderklimov.ru/android/java/java.php>
36. <http://spec-zone.ru/RU/Java/Docs/7/api/overview-summary.html>
37. <http://www.skipy.ru/index.html>