Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ЭВМ

Отчёт

по лабораторной работе

**игра «Hell guardians»**

Выполнил: Проверил:

ст. гр. 450502 Кухарчук И. В.

Стаховский А.В.

Минск, 2016

**Скриншоты игры**

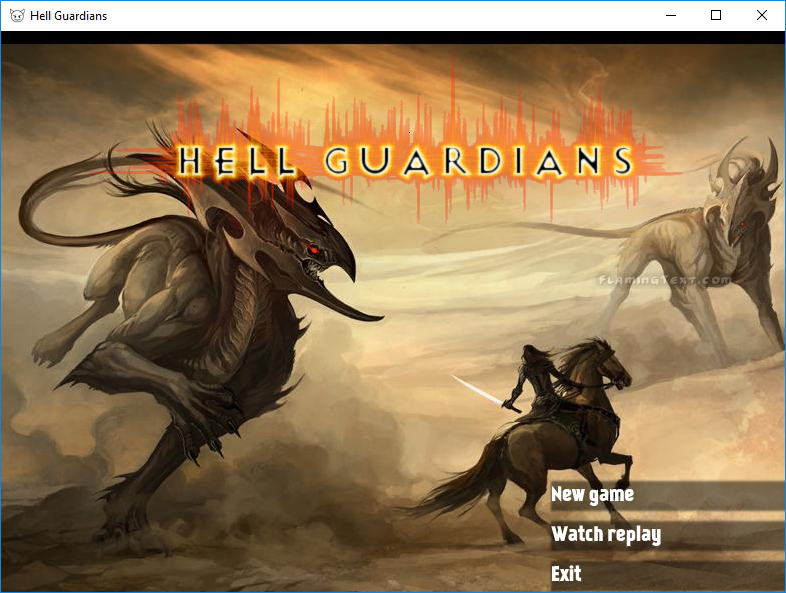
****

Рисунок 1 – Главное меню на операционных Windows

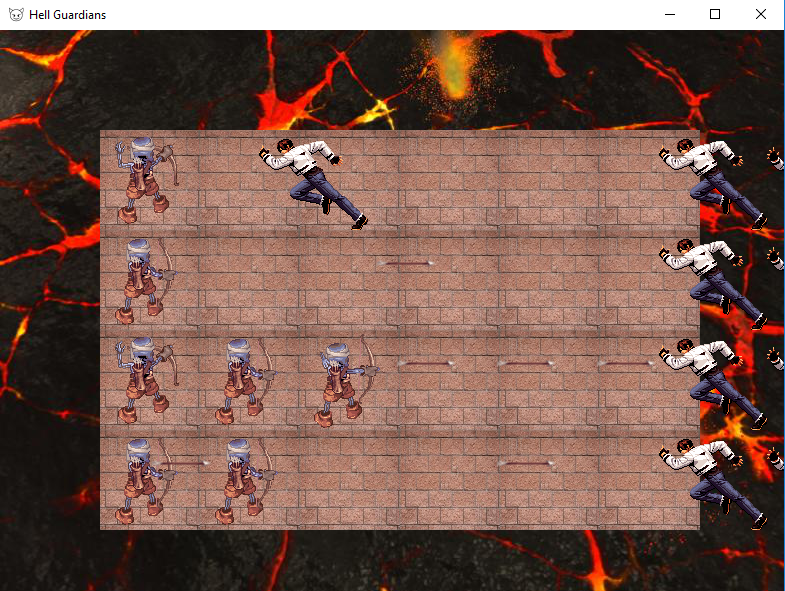


Рисунок 2 – Вид игрового экрана на операционных системах Windows



Рисунок 3 – Главное меню на операционных системах Linux

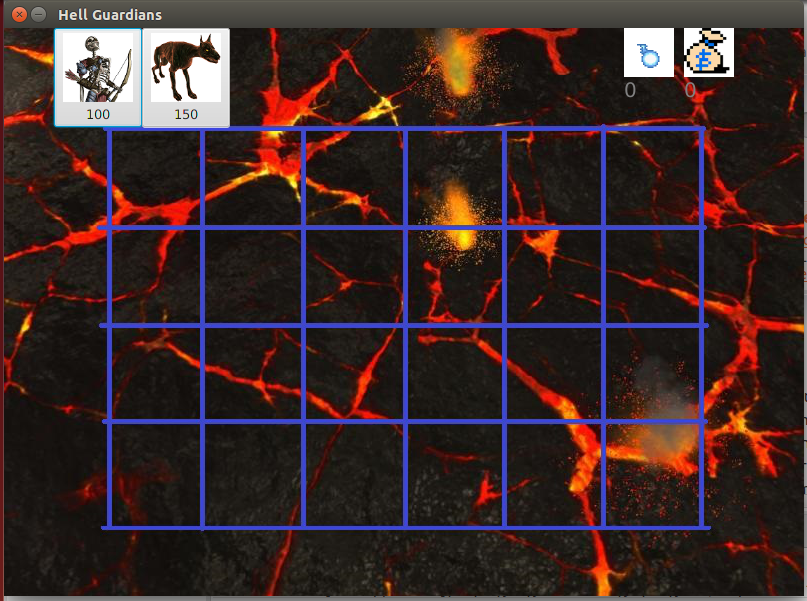


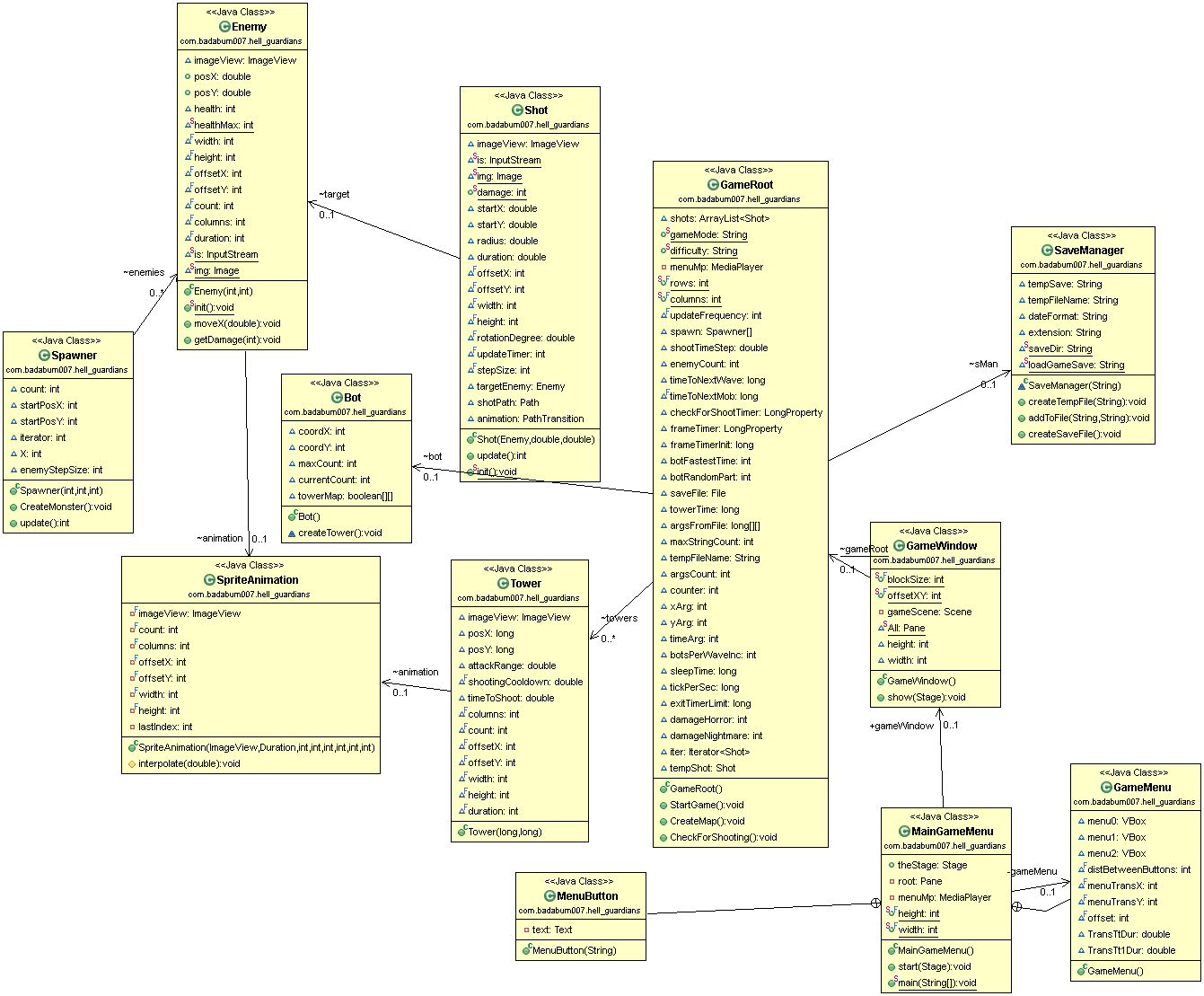
Рисунок 4 – Вид игрового экрана на операционных системах Linux

**Игровой процесс**

Игра основывается на Tower Defense механике, использованной в игре «Plants vs Zombies». Игра «Hell guardians» проходит в режиме «игры на выживание». Игрок строит «башни» для защиты от бесконечно прибывающих врагов. За каждого убитого врага начисляются очки и ресурсы для постройки новых башен. С течением времени количество врагов растет. Для успешной игры игроку требуется грамотно организовать оборону. Цель игрока — набрать как можно больше очков. При достижении врагами башен происходит нанесение урона сооружениям игрока. Если количество полученного урона равно или превышает лимит прочности башни, то сооружение разрушается. На месте разрушенной башни можно построить новую.

Игровое поле представляет из себя сетку. Игрок может строить башни в свободных ячейках сетки. Враги передвигаются по прямой через центр клеток. По достижению врагами конца игрового поля, наступает конец игры.

**Диаграмма классов**



Описание классов:

* MainGameMenu – класс, отвечающий за отрисовку стартового меню программы
* GameMenu – класс, являющийся контейнером для кнопок. Управляет поведением кнопок
* MenuButton – обеспечивает создание одиночной кнопки
* GameWindow – создает и настраивает игровую сцену, передает дальнейшее управление игрой объекту класса GameRoot
* GameRoot – основной класс. Управляет процессом игры: генерация врагов, обновление их позиции, управление ботом, управление поведением башен
* Spawner – отвечает за генерацию врагов в определенной точке
* Enemy – описание врага и его поведения
* Tower – класс, описывающий башню
* Block – параметры клетки игрового поля. Блоки являются площадками для строения башен
* SpriteAnimation – управление спрайтовой анимацией башен и врагов
* Bot – логика бота, который может играть вместо игрока
* Shot – параметры выстрела, генерируемого башней

**Блок-схема основного алгоритма**



**Сохранение игры**

Для сохранения игры использовался следующий формат записи:

количество\_урона\_наносимого\_выстрелом

позиция\_башни\_по\_Х позиция\_башни\_по\_У время\_постройки\_башни

.

.

.

позиция\_башни\_по\_Х позиция\_башни\_по\_У время\_постройки\_башни

Пример сохранения:

20

400 400 276

400 300 318

100 400 703

100 300 744

Для хранения данных о сохранениях используется папка saves. Во время игры данные сохраняются во временный файл, из которого потом записываются в отформатированном виде в сохранение. Имя файла сохранения имеет следующий формат: ггггММдд\_ччммсс (дата и время постройки) + количество построенных башен. Сохранения сохраняются с расширением .sav и содержат все необходимые данные для точного воспроизведения игры. Сохранение происходит автоматически после каждой игры. Выбрать сохранение для воспроизведения можно перейдя в диалоговое окно через кнопку “Watch replay” из главного меню.