

2. Исполнитель при первоначальном входе в приложение регистрируется, как пользователь и дальше входит, так же как исполнитель. И получает функционал в виде:

- просмотра количества прослушиваний его аудиофайлов;
- предлагать добавить его аудиофайл в приложение;
- удаление альбомов и аудиофайлов исполнителя;
- создание альбомов.

3. Пользователь при первоначальном входе в приложение регистрируется, как пользователь и так же входит, как пользователь, получив функционал в виде:

- просмотра аудиофайлов, плейлистов, альбомов;
- прослушивание аудиофайлов, плейлистов, альбомов;
- создание и удаление плейлистов;
- добавления аудиофайлов в плейлист или же удаления из плейлиста;
- добавление и удаление аудиофайлов или плейлистов в «Моя медиатека»;
- управление воспроизведение аудиофайлов.

Кроме выбора функционал данного мобильного приложения мы использовали:

– IntelliJ IDEA – это интерактивная среда разработки (IDE) от JetBrains, предназначенная для разработки программного обеспечения на различных языках программирования, включая Java, Kotlin, Scala и другие;

– Firebase Realtime Database – облачная база данных, которая обеспечивает синхронизацию данных в реальном времени между мобильными приложениями и сервисом Firebase;

– Figma – онлайн-инструмент или же приложение для персонального компьютера или мобильного устройства для дизайна интерфейсов;

– Kotlin – мощный, современный и многофункциональный язык программирования, который представляет простой и элегантный синтаксис, а также поддержку для различных платформ, включая JVM Android и JavaScript.

Мобильное приложение работает на операционной Android.

В итоге мобильное приложение успешно используется для воспроизведения аудиофайлов.

Л. Д. Вергунов

(ГГТУ имени П. О. Сухого, Гомель)

Науч. рук. **Л. К. Титова**, ст. преподаватель

РАЗРАБОТКА ИГРОВЫХ МЕХАНИК ШУТЕРА “BULLETS OF DOOM”

В истории игровой индустрии жанр шутеров по праву считается одним из самых значимых и весомых. Это обусловлено тем, что жанр традиционно входит в число самых популярных из-за отлаженной методологии производства и простоте в освоении игрового процесса. Жанр шутеров насчитывает историю длиной не менее чем в полвека. За все это время он, разумеется, претерпевал значительные изменения, пройдя путь от классических аркадных концепций к глубоким и многогранным игровым мирам. Разработчики по всему миру не боялись экспериментировать с новыми механиками, расширяя возможности игрока, создавая новые поджанры и комбинируя уже существующие. На сегодняшний день существует бесчисленное количество проектов, относящихся к этому обширному жанру, включающих в себя как претенциозные разработки крупных студий с многомиллионными бюджетами, так и небольшие, едва заметные игры от инди-разработчиков. Общим для данных проектов остается присутствие возможности стрельбы, а также некие сущности, выступающие в роли противников.

Об актуальности разработки приложений в данном сегменте игрового рынка лучше всего говорят данные о бюджетах крупных игровых компаний, а также об их годовой

выручке. Так, к примеру, на разработку последней на данный момент игры в культовой серии *Grand Theft Auto* было потрачено, по разным оценкам, от 265 до 365 миллионов долларов США. Эта сумма была окуплена в полном объеме уже в первый день продаж копий игры. Не менее впечатляющие цифры может предоставить также компания *Activision Blizzard* со своей серией классических сюжетных шутеров *Call of Duty*. Так, по данным компании, количество проданных за все время цифровых копий всех частей серии к 2023 году превысило 425 миллионов при суммарной чистой прибыли от их продаж в 30 миллиардов долларов. Это говорит о стабильном росте спроса на игры-шутеры [1].

К наиболее популярным, а также наиболее специфичным поджанрам шутеров относятся так называемые зомби-шутеры. Их отличает сеттинг, включающий уже сложившиеся паттерны, такие как большое количество противников с не самым проработанным поведением и интеллектом, возможность использовать как холодное, так и огнестрельное оружие, механика игры в кооперативе, а также общая гнетущая атмосфера и реалистичная демонстрация насилия и жестокости. Наиболее известными представителями являются такие культовые игры “*Left 4 Dead*”, “*Left 4 Dead 2*”, “*Dead Island*”, “*Minigore Zombies 2*” и многие другие. Данные проекты взяты за основу при разработке игрового приложения “*Bullets of Doom*”.

Основной платформой для разработки выбран игровой движок “*Unity*”, представляющий широкий функционал для создания трехмерных шутеров, позволяющий портировать игры для различных игровых платформ. В случае игры “*Bullets of Doom*” целевыми платформами являются персональные компьютеры и мобильные “*Android*” устройства. Портирование игры под “*Android*” подразумевает необходимость использовать игровые модели с низким числом полигонов, чтобы избежать чрезмерной нагрузки на графические адаптеры устройств и оперативную память. Игровой процесс также заточен под изометрическое трехмерное пространство. Это позволит реализовать передвижение игрового персонажа через простой контроллер в виде джойстика.

В “*Bullets of Doom*” предусмотрена как одиночная, так и кооперативная игра в локальной сети. Количество игроков в одном лобби ограничено четырьмя. Сетевое взаимодействие реализовано с применением технологии *Unity Networking*, реализуя одноуровневую модель peer-to-peer, где каждый из участников игрового лобби может выступать как в роли клиента, так и в роли хоста для связывания сети. Самым полезным для разработки приложения инструментом в *Unity Networking* является возможность использования автоматической синхронизации игроков в игровом времени и пространстве без необходимости разрабатывать данные алгоритмы самостоятельно.

Каждый из игроков предварительно выбирает игрового персонажа, которых также четыре. Помимо визуальных отличий, игровые аватары имеют свои собственные индивидуальности, что можно временно активировать (на 10 секунд), собрав три специальных объекта на игровом поле. Собрав три таких объекта, аватар 1 получает неуязвимость, аватар 2 – повышенный урон, аватар 3 – повышенную скорость движения, а аватар 4 – повышенную скорострельность оружия. Временное улучшение игрока реализовано через структурный паттерн проектирования под названием «декоратор». Он позволяет динамически подключать к объекту дополнительную функциональность через систему наследования и открытые модификаторы доступа. Данная функциональность может быть отключена в любой момент при удалении объекта-декоратора или изменении его характеристик [2].

Игра не может являться полноценным шутером без наличия в ней непосредственно оружия, возможности его использовать. В “*Bullets of Doom*” имеется боевая система, включающая как холодные, так и огнестрельные виды вооружений. Причем имеет несколько их видов, различающихся по скорострельности, силе наносимого урона и его площади. Игрок может использовать подходящие под его предпочтения виды вооружений. Получить то или иное оружие можно из появляющихся время от времени на игровой карте ящиков. Однако использовать игрок может лишь разблокированные при помощи внутриигровых предметов

и систему крафта мечи, топоры, ружья и автоматы. Получить данные предметы можно случайным образом от поверженных противников или приобрести за игровую валюту. Система создания расходных материалов открывается между боевыми сессиями. Каждый составной предмет в ней имеет свой собственный рецепт создания.

Неотъемлемой частью основной игровой механики шутера является наличие неких игровых сущностей, мешающих игроку в выполнении его задачи, проще говоря, противников. Обладающие развитым искусственным интеллектом враги способны подарить пользователю более запоминающийся опыт. Жанр зомби-шутера не накладывает необходимость в продвинутом интеллекте: противникам хватит базовых алгоритмов поиска пути к игроку через поиск по дереву, алгоритмов атаки. Активным сенсором выступает конус зрения [3]. Внедрен алгоритм, размещающий случайным образом на карте противников, ящики с оружием, расходные материалы.

Литература

1. Новостной портал ГК РосБизнесКонсалтинг. [Электронный ресурс]. – Режим доступа: <https://www.rbc.ru/>. Дата доступа: 17.03.2024.

2. Мартин, Р. С. Чистый код. Создание, анализ и рефакторинг / Р. С. Мартин. СПб : Питер, 2019. – 464 с.

3. Overholtzer, C. A. Adding Smart Opponents to a First-Person Shooter Video Game through Evolutionary Design / C. A. Overholtzer. – Computer Science Department Washington and Lee University : Lexington, 2016. – 156 p.

М. А. Винокуров

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. В. Васькевич**, ст. преподаватель

СПОСОБЫ ПОИСКА И УСТРАНЕНИЯ УЯЗВИМОСТИ OS COMMAND INJECTION

В списке OWASP Top 10 2017 года инъекции занимали первую позицию как самые опасные уязвимости. В 2021 году они сместились на третью строчку списка [1]. Стоит отметить, что нельзя недооценивать уязвимости, связанные с инъекциями, т. к. проверку приложения на защищённость от инъекций можно автоматизировать. Инъекции – это группа уязвимостей, связанных с внедрением («инъекцией») вредоносного кода в целевую систему.

В данной работе будут разобраны способы поиска и устранения одного из типов инъекций, а именно OS Command Injection. OS Command Injection (уязвимость командной строки) – это уязвимость, которая позволяет злоумышленнику выполнять команды операционной системы на сервере, где работает приложение, и обычно полностью компрометирует приложение и его данные.

Поиск данной уязвимости может производиться следующими способами: ручное тестирование, использование специализированных инструментов, использование сканеров уязвимостей, статический и динамический анализ.

При ручном тестировании специалист своими силами ищет места в приложении, где возможна данная уязвимость, и проверяет их. Проверка может производиться как обычным введением полезной нагрузки (вредоносного кода) в форму в приложении, так и отправкой GET и POST запросов через командную строку, или с использованием ПО Burp Suite.

Ручное тестирование можно автоматизировать использованием специальных инструментов. Для OS Command Injection подходит инструмент commix. Так же можно использовать самописные скрипты, которые путём перебора словаря с полезной нагрузкой отправляют запросы на сервер, тестируемый на уязвимость. Сканеры, помимо инъекций, могут найти множество других уязвимостей. Для поиска OS Command Injection подходят: Burp Suite Pro, OWASP ZAP и Arachni.