



Рис. 1. Структура системы

В перспективе планируется расширение функциональности системы за счет интеграции с геоинформационными системами для визуализации перемещения отходов и добавления модуля прогнозирования образования отходов. Дополнительно рассматривается возможность создания мобильного приложения для оперативного ввода данных непосредственно с мест хранения отходов.

Таким образом, разработанный программный комплекс не только автоматизирует процесс учета отходов, но и предоставляет комплексный инструмент для контроля соблюдения экологических нормативов. Это существенно сокращает трудозатраты экологов, снижает риски штрафных санкций и делает процесс экологического контроля более прозрачным и эффективным.

Литература

1. EcoMan: Система экологического мониторинга. – URL: <https://ecoman.ru/> (дата обращения: 10.02.2025).
2. 1С: Предприятие 8.3. Документация разработчика. – URL: <https://v8.1c.ru/> (дата обращения: 10.02.2025).
3. Федеральный классификационный каталог отходов. – URL: <http://rpn.gov.ru/> (дата обращения: 10.02.2025).

АРКАДНАЯ ИГРА «РАЗРУШИТЕЛЬ БЛОКОВ» С РАЗРАБОТКОЙ ИГРОВОГО БОТА, ИМИТИРУЮЩЕГО ПОВЕДЕНИЕ ИГРОКА ПОД ОС ANDROID

И. В. Громыко

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель Л. К. Титова

Рассмотрена разработка игрового приложения под ОС Android. Отмечено, что особенностью игрового приложения является разработка игрового бота, который реализует

ся с помощью подхода *Utility AI*. Выделена возможность настройки игровых механик приложения с помощью инструментов движка *Unity*.

Ключевые слова: *Unity*, игровое приложение, искусственный интеллект.

Сегодня существует масса примеров различных игровых приложений, которые написаны на разных языках программирования и созданы для различных платформ и целевых аудиторий.

Теперь игры используются не только для развлечения, но и в массе других сфер: в обучении, торговле и здравоохранении. Количество времени, которое проводят люди за игрой, с каждым годом растет, в связи с чем растет и потребность в создании новых приложений, так как старые игры интересуют их уже намного меньше.

К тому же сейчас индустрия нацелена, в первую очередь, на развитие многопользовательских игр с соединением по сети для компьютеров с высокой вычислительной мощностью, в связи с чем сохраняется актуальность однопользовательского игрового приложения для мобильных устройств.

Также в настоящее время популярным направлением при разработке игровых приложений является создание игрового искусственного интеллекта (ИИ).

Так как *Unity* – более простой в освоении игровой движок, во многом из-за того, что в качестве языка программирования он применяет *C#*, который оставляет большое количество работы с аппаратным обеспечением на себя, в отличие от *C++*, использующих игровой движок *Unreal Engine*, а также предоставляющего удобный функционал для разработки двумерных игр на мобильные устройства, то для приложения, реализующего игровое приложение «Разрушитель блоков» в жанре «Аркада» с разработкой игрового бота, имитирующего поведение игрока, используется игровой движок *Unity*.

Unity – это профессиональный игровой движок, с помощью которого можно создавать видеоигры для различных платформ: от мобильных устройств до различных игровых консолей и персональных компьютеров. На *Unity* за свою историю существования было написано множество игровых приложений различных величин: от небольших проектов до одних из самых крупных в индустрии [1].

Приложение «Разрушитель блоков» является двухмерной игрой в жанре «Аркада», где игроку необходимо управлять платформой, которая отбивает шарики, с помощью которых разрушаются блоки.

В приложении «Разрушитель блоков» кроме обыкновенных блоков также реализованы различные бонусные блоки, которые могут помогать пользователю быстрее пройти уровень, или мешать ему и замедлять прохождение. Одним из таких бонусов является бонус, при активации которого игрок не может управлять платформой, так как управление платформой происходит с помощью игрового бота, который старается отбивать шарик и подбирать положительные бонусы.

При разработке ИИ для создания модели поведения в игровых приложениях используется масса подходов. Примерами могут служить конечные автоматы, деревья поведения, *Utility AI* и др.

Utility AI является приоритетным вариантом для задач, где необходимо создание неигровых персонажей, поведение которых должно быть чувствительным к изменяющимся условиям игровой среды.

При разработке искусственного интеллекта с помощью *Utility AI* необходимо указать действия, которые искусственный интеллект может выполнять в определенный момент времени.

При создании действия нужно указать список условий, которые будут влиять на выполнение данного действия. Например, платформа может передвигаться в сторону бонусного блока, и на данное действие могут влиять различные условия: расстояние до шарика, расстояние до выпадающего бонуса и др.

Для каждого условия задается определенная формула, которая будет высчитывать очки для оценки выгоды определенного действия. Очки могут высчитываться различными способами: с помощью задания кривой, дискретными значениями или другим алгоритмом таким образом, чтобы диапазон значений был от 0 до 1. Для действия можно указать приоритет, повышая или понижая значения с помощью функций.

При определении действия, которое должен выполнить интеллект, производится подсчет очков всех условий для каждого возможного действия, по результату которого *Utility AI* определяет, какое действие получило наивысшую оценку, и выполняет его.

Также при разработке приложения особое внимание было уделено настройке игровых механик с помощью редактора.

С помощью средств игрового движка *Unity* при разработке игрового приложения были созданы диалоговые окна внутри редактора, с помощью которых можно создавать игровые уровни, а также управлять переводом игры на различные языки.

Приложение реализовано таким образом, что изменение настроек множества механик производится с помощью конфигурационных файлов в редакторе *Unity*. Чтобы изменить настройки игры (например, скорость полета шарика), необходимо поменять соответствующее значение в конфигурационном файле без необходимости изменения кодовой базы. Аналогичным образом настраивается поведение различных блоков: для каждого блока есть определенный набор настроек его поведения.

На рис. 1 показан пример настройки блока бомбы с помощью редактора.

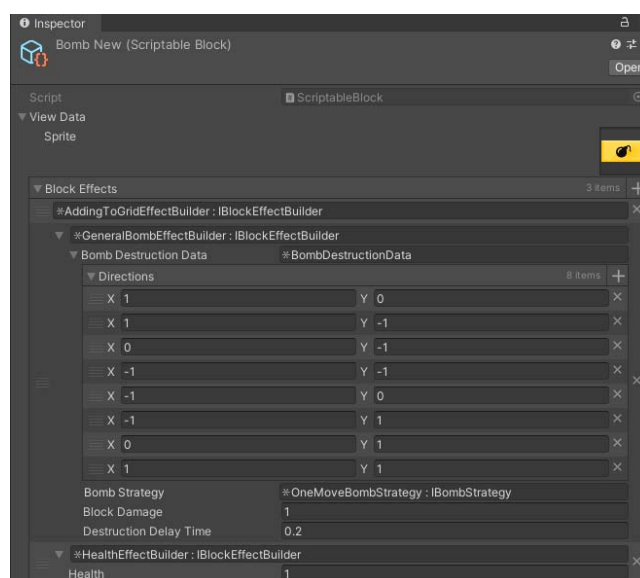


Рис. 1. Пример настройки блока «Бомба»

В результате выполнения работы было разработано игровое приложение на игровом движке *Unity*, в котором игроку необходимо управлять платформой, чтобы отбивать шарик, который уничтожает блоки. Цель игры – уничтожить все блоки на уровне, не потеряв все очки здоровья.

Преимуществом такого варианта игры является простой и интуитивный интерфейс, а также наличие различного вида блоков и бонусов, которые делают игровой процесс для пользователя более разнообразным. Одним из таких бонусов является бонус, при подборе которого управление платформой переходит игровому боту, который имитирует поведение игрока: подбирает положительные блоки, отбивает шарик, избегает негативных блоков. При разработке ИИ использовался подход *Utility AI*. При создании ИИ в игровых приложениях с помощью подхода *Utility AI* необходимо проработать функции и приоритеты для действий, так как от них будет зависеть то, каким образом неигровые персонажи будут реагировать на различные условия игровой среды.

Литература

1. Джейсон, Г. Игровой движок. Программирование и внутреннее устройство / Г. Джейсон. – 3-е изд. – СПб. : Питер, 2021. – 1136 с.

ОСОБЕННОСТИ РАЗРАБОТКИ БОЕВОЙ СИСТЕМЫ В 3D-ИГРОВOM ПРИЛОЖЕНИИ «SAGA OF VALOR» В ЖАНРЕ «ЭКШЕН-ПРИКЛЮЧЕНИЕ» НА ПЛАТФОРМЕ Unity

Е. А. Бурдов

*Учреждение образования «Гомельский Государственный Технический
университет имени П. О. Сухого», Республика Беларусь*

Научный руководитель Е. В. Комракова

Проведен всесторонний анализ боевой системы, используемой в современных 3D-играх. Рассмотрены основные принципы организации сражений, механизмы расчета урона, системы реакции на атаки и интеграция боевой логики с остальными подсистемами игры. Приведен обзор известных реализаций боевых систем в популярных играх, таких, как «Dark Souls», «Assassin's Creed» и «God of War», что позволяет выделить ключевые моменты для эффективного построения подобных систем. Описаны особенности реализации боевой системы в игровом приложении «Saga of Valor».

Ключевые слова: боевые механики, 3D-игра, сражения, экшен, приключение, игровая логика.

Разработка боевой системы является одним из важнейших направлений в создании современных трехмерных игр, поскольку от ее качества напрямую зависит динамика сражений и общее впечатление от игрового процесса. Современные игры стремятся обеспечить реалистичность боевых столкновений посредством сложной логики взаимодействия между персонажами, что предполагает учет множества факторов: физики, анимации, реакции на пользовательские команды и синхронизации с аудиовизуальными эффектами. При этом подход к реализации боевых систем характеризуется модульностью, когда каждая подсистема отвечает за отдельный аспект боя – от расчета урона до визуализации эффектов столкновений.

Известные игровые проекты демонстрируют разнообразие подходов к организации боевых механик. Например, в серии «Dark Souls» акцент делается на тактическом планировании и точном расчете параметров атаки, что позволяет игрокам оценивать каждое сражение как испытание навыков и терпения. На рис. 1 представлена игра «Dark Souls» III.