

Рисунок 4 — Работа программы по демонстрации модели собирающей и рассеивающей линз

Заключение. Математическая модель собирающей и рассеивающей линз работает корректно и соответствует следующим требованиям к математической модели: адекватности, универсальности и экономичности [2].

Разработанную программу можно улучшать путем добавления большего количества настроек. Такая модель удобна для использования в учебных заведениях для демонстрации свойств линз.

Список цитируемых источников

1. Методы математического моделирования [Электронный ресурс]. — Режим доступа: <https://mipt.ru/education/chair/mathematics/upload/2f8/Umnov-Metoditska-MatModel-arph0ds4sb2.pdf>. — Дата доступа : 27.09.2021.
2. Основы математического моделирования [Электронный ресурс]. — Режим доступа: https://elar.urfu.ru/bitstream/10995/68494/1/978-5-7996-2576-4_2019.pdf. — Дата доступа : 27.09.2021.

УДК 004.92

А. А. Малышев, О. Д. Кравчук

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

РЕАЛИЗАЦИЯ ИГРОВЫХ ОБЪЕКТОВ НА ПЛАТФОРМЕ UNITY 3D

Введение. Игры, будучи самым молодым развлекательным форматом, придуманным человечеством, развиваются быстрее и зарабатывают денег больше, чем любой другой сегмент развлекательной индустрии. В 2020 году всемирный рынок видеоигр приблизился к цифре в 175 миллиардов долларов — на 12 % больше, чем годом раньше. Игры зарабатывают больше, чем суммарно кино-, книжная и музыкальная индустрии [1].

Доля «инди» — любительской разработки без больших бюджетов и организации — около 68 % в среднем по отрасли и ещё больше среди мобильных платформ. Низкие требования к входу на рынок предъявляемые к любительской разработке могут представлять интерес для разработчика.

Кроме того, этот вид разработки может быть крайне увлекательным возможностью создавать собственные миры, живущие и развивающиеся по придуманным правилам, которые, к тому же, можно менять в любой момент как захочется, наблюдая изменения, которые не предусматривали, но являющиеся следствием вмешательства.

Таким образом, актуальной является задача разработки игрового приложения под мобильные устройства.

Целью работы является исследование компонентов и методов для проектирования игровых объектов и реализации их активности.

Объектом исследования выступает разрабатываемая компьютерная игра в жанре RPG.

Основная часть. При разработке мобильных таких игр выделяют такие плюсы как:

1. Полная свобода. Для творческих личностей этот плюс — основной и перекрывающий даже минусы.
2. Нулевая репутация. На первый взгляд это может показаться минусом, но на самом деле выпуск игр под неизвестным ранее именем, не имеющим отношения к брендам, имеет свои плюсы. Репутация бренда не

всегда может быть безупречной (былые провалы, неудачный пиар), тогда как репутация независимой никому не известна, а имя привлекает своей новизной.

Минусы:

1. Мало людей. Чаще всего независимые разработчики работают либо в одиночку, либо в совсем небольшой команде. По этой причине многозадачность неизбежна, а ни для кого не является секретом, что чем больше дел делается одновременно, тем хуже результат для каждого из них.

2. Нулевая репутация. Обратной стороной популярности является недоверие аудитории.

В качестве инструмента для создания графических приложений использован Unity 3D. В качестве мультиплатформенной среды разработки выбрана MonoDevelop IDE, разработка программы велась на платформе .Net Framework. Для написания программного продукта был выбран объектно-ориентированный язык программирования высокого уровня C#.

В качестве архитектурного паттерна для создания надстройки, был выбран Model-View-Controller (MVC) — концепция разделения данных приложения на три отдельных компонента: модель, представление, контроллер (рисунок 1).

1. Модель — хранит данные и реагирует на команды контроллера, изменяя хранимые данные.

2. Представление — пользовательский интерфейс и отображение модели данных.

3. Контроллер — управляющая логика приложения.

Объект Player отвечает за поведение и логику главного персонажа игрока на игровом поле. К нему подключены компоненты Player Control, Character Controller, Audio Source.

Компонент Character Controller в основном используется для управления от третьего или первого лица, где не требуется физика. Компонент Audio Source является источником звука. Компонент Player Control является скриптом, который и реализует все игровые механики персонажа, передвижение, стрельбу, анимацию, визуальные эффекты. Для реализации движения персонажа использовались двумерные и трехмерные вектора. Первым этапом следует получить координаты, по которым пользователь нажал на стик на экране устройства. В компонент Character Controller передается вектор, который хранит в себе координаты направления, тот в последующем двигает персонажа в нужном направлении. В зависимости от того, как близко находится стик к границам джойстика, зависит скорость передвижения. Получение вектора направления отображено на рисунке 2.

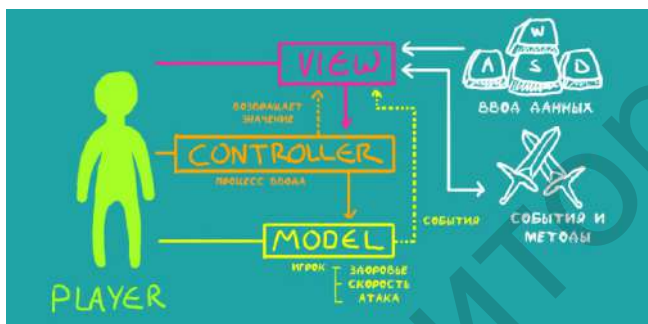


Рисунок 1 — Схема архитектурного паттерна Model-View-Controller

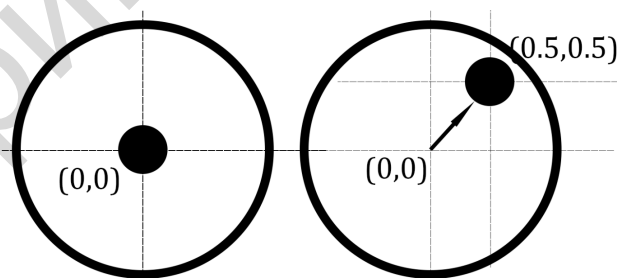


Рисунок 2 — Расчет вектора направления

При нахождении вектора направления вектор складывается с текущей позицией персонажа и умножается на скорость и переменную Time.DeltaTime хранящее в себе время обработки предыдущего кадра для того, чтобы предотвратить рывки в анимации движения.

Для стрельбы используется тот же принцип, определяется вектор направления, и в зависимости от того, как долго игрок держит стик активным зависит скорость снаряда и его мощность. После того стик отпущен, происходит создание нового объекта Projectile, который является префабом снаряда, имеющим свои компоненты и скрипт. В скрипт снаряда передается вектор направления и длительность зажатия стика.

Для анимации используется компонент Animator в который передается 3D модель персонажа, заранее анимированная в 3D редакторе. Переходы между анимацией реализуются с помощью дерева анимаций.

Для выполнения каких-либо действий в определенный кадр анимации, используется встроенная функция в Unity, которая позволяет привязать к конкретному кадру выполнение метода из скрипта. У игрока на последнем кадре анимации проигрыша, выполняется метод, который вызывает объект с сообщением о проигрыше и останавливает игровое время.

Для улучшения графики использовался шейдер — компьютерная программа, для создания подсветки поверхности объекта, построения теней, закрасивания поверхностей или сечений объемных объектов [2].

В Unity существует инструмент, позволяющий разрабатывать шейдеры с легкостью, перетаскивая блоки вместо написания кода. Создавая и объединяя узлы в графы, с возможностью следить за результатом в реальном времени благодаря открытой архитектуре и инструментам разработки графов в Shader Graph.

Сравнения стандартной обработки модели и с помощью шейдера на примере модели из игры, приводятся на рисунке 3. С помощью шейдера меняется интенсивность освещения, наличие и цвет теней, добавляется контурная обводка.



Рисунок 3 — Сравнение обработки 3D моделей

Заключение. По итогам исследования компонентов и методов для проектирования игровых объектов и реализации их активности были сделаны следующие выводы:

1. Первостепенное значение в проектировании игр играет выбор паттернов проектирования, так как они выступают строительными блоками, которые позволяют программистам использовать существующие знания вместо того, чтобы начинать с нуля строительство каждой системы. Кроме того, они имеют набор стандартных моделей, которые помогают другим разработчикам понять, как их проекты могут взаимодействовать с данной системой.

2. Выбор используемых компонент также играет важную роль, так как от них зависит будущий функционал игрового приложения.

3. Выбор графических средств обуславливает будущий привлекательный и интуитивно понятный интерфейс игрового приложения, который позволяет игроку глубже погрузиться в геймплей. От качества интерфейса зависит количество удовольствия, которое получит пользователь во время игрового процесса.

Список цитируемых источников

1. Подшибякин, А. М. Время игр! Отечественная игровая индустрия в лицах и мечтах : от Parkan до World of Tanks. / А. М. Подшибякин. — Бомбора, 2020. — 295 с.
2. Магдануров, Г. И. Visual Basic на практике / Г. И. Магдануров. — Спб. : БВХ-Петербург, 2008. — 480 с.

УДК 004.942

Н. С. Миколайчук, А. В. Шах

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

МОДЕЛИРОВАНИЕ УДАРНОГО ВОЗДЕЙСТВИЯ В САЕ-СИСТЕМЕ ANSYS AUTODYN

Введение. Краевые задачи (системы уравнений с начальными и граничными условиями), описывающие те или иные процессы в любой области науки и техники, в силу своей сложности, как правило, не позволяют получать решение аналитически. Речь здесь идет, конечно же, о математических моделях сложных процессов и объектов, а не о типовых ситуациях или частных случаях. В случае невозможности получить аналитическое решение на помощь исследователям приходят численные методы.

Современное представление о численных методах решения краевых задач может быть сформулировано рядом простых тезисов.

Во-первых, следует сказать о том, что все формализуемые процессы могут быть описаны математической моделью, которая в общем случае представляет собой систему дифференциальных уравнений, дополненную краевыми условиями. Во-вторых, большинство процессов и явлений, происходящих в природе, требуют описания сложными моделями, часто трехмерными и нелинейными. Поэтому исследователи вынуждены применять численные методы для получения решения поставленных задач. В-третьих, все численные методы условно можно разделить следующим образом: методы конечных разностей и методы конечных элементов. В основе метода конечных элементов лежит принцип деления исследуемой области на совокупность подобластей — отсюда метод и получил свое название [1].

Основная часть. В данном исследовании основной задачей было проведение моделирования ударного взаимодействия и анализ полученных результатов в среде ANSYS Workbench 14.5.7.

Для разработки и построения модели использовалась система проектирования SolidWorks 2018, построенная на базе платформы Dassault Systemes 3DEXPERIENCE. Она охватывает весь процесс разработки про-