графических элементов интерфейса. Основным отличием от своих конкурентов это то, что на стороне клиента или сервера вся логика написана на объектно-ориентированном языке C#, вместо JavaScript. Графические элементы, составляющие конечный продукт, отображаются благодаря проверенным временем технологиям – HTML и CSS.

Данный фреймворк является частью платформы .NET с открытым исходным кодом. Он поддерживается активным сообществом участников из более чем 3700 компаний. Поскольку .NET является бесплатным, так же, как и Blazor, это не связывает разработчиков какими-либо лицензиями или подписками для возможности пользоваться им, в том числе для коммерческого использования.

Основные плюсы разработки с Blazor: разработка логики веб-приложений с помощью C# вместо JavaScript; использование наработанной разноплановой базы библиотек платформы .NET и её скорости работы с данными; интеграция с современными платформами размещения, такими как Docker; применение средств разработки от Microsoft во много раз увеличивают скорость и удобство разработки благодаря шаблонам.

Все приложения Blazor состоят из компонентов. Компонент – это любой элемент UI-интерфейса: кнопка, выпадающий список, форма ввода. Фактически сам фреймворк состоит из двух продуктов: Blazor Server, Blazor WebAssembly.

Blazor Server приложения работает на стороне сервера. Обновление элементов UI-интерфейса, обработка событий, работа JavaScript на клиентской стороне осуществляются благодаря совместной работе сервера и клиента через SignalR.

Blazor WebAssembly позволяет создавать интерактивные одностраничные приложения, которые запускаются на стороне клиента с помощью технологии WebAssembly.

Б. А. Тесёлкин, Е. В. Комракова
(*ГГТУ им. П. О. Сухого, Гомель*)

РАЗГОН ФИЗИЧЕСКОГО ДВИЖКА ПРИ СОЗДАНИИ ИГРЫ В UNITY

Разработчик, опираясь на понимание наиболее значительных возможностей физического движка *Unity*, может использовать методы оптимизации производительности физической системы игрового приложения.

Существует ряд методов, применение которых к сцене позволяет снизить вероятность нестабильной работы физического движка: масштаб всех физических объектов игрового мира следует удерживать как можно ближе к соотношению 1:1:1, аналогично расположение всех объектов вблизи точки (0,0,0) приведет к увеличению точности операций с плавающей запятой и улучшению согласованности при моделировании.

Важной частью физической системы является контроль взаимодействия её объектов, за что отвечает матрица столкновений. Она определяет объекты в слоях, которые могут сталкиваться с объектами в других слоях. Все прочие пары слоев или объектов просто игнорируются физическим движком, что делает этот механизм важным средством уменьшения нагрузки, минимизирующим количество проверок, выполняемых на каждом шаге. Также для большинства объектов следует использовать способ определения столкновений по умолчанию *Discrete*. Одноразовая телепортация объектов и выявление совмещения пар близко расположенных друг к другу объектов не составляют особого труда. Но важнейшей частью физического объекта является его коллайдер. Большинство объектов можно аппроксимировать одним из четырех примитивных коллайдеров. Можно использовать несколько коллайдеров, подходящих для определения столкновений объектов сложной формы, путем присоединения дополнительных дочерних игровых объектов с их собственными коллайдерами. Это практически всегда менее затратно, чем применение одного меш-коллайдера, и этому менее сложному решению следует отдавать предпочтение.

Все методы, описанные выше, использовались при разработке игрового приложения и доказали высокую степень эффективности оптимизации физического движка *Unity*.

В. С. Устименко, Н. Б. Осипенко
(*ГГУ им. Ф. Скорины, Гомель*)

РАЗРАБОТКА ВЕБ-САЙТА О ЕДИНОБОРСТВАХ

Сайт может быть поисковой системой, форумом, социальной сетью, блогом или чем-то еще. Разработанный веб-сайт содержит информацию о единоборствах, предстоящих спортивных событиях, биографию бойцов.