

ментов не только построчно, но и поколонно т.е. работать с областями.

Основная идея Grid состоит в том, чтобы представить веб-страницу в виде сетки, т.е. разбить страницу на определенный набор строк и столбцов, а затем, указывая области на сконфигурированной сетке, разместить необходимый контент, используя CSS.

Таким образом, с появлением Grid стало возможным точно указывать расположение элементов в сетке, не меняя структуру HTML-документа. Следует отметить, в спецификации CSS Grid появились новые функции, единицы и свойства для более удобной работы с сеткой (fr, repeat(), fit-content(), min-content и др.).

Еще одним достоинством Grid является удобство работы над адаптивностью страниц – можно полностью менять размещение элементов изменяя только лишь конфигурацию областей сетки при помощи свойства grid-template-areas. А создание резиновых макетов значительно упрощает использование новых единиц измерения – fr.

История инструмента начинается с апреля 2011 года. Именно тогда компания Microsoft отправила предложение группе разработчиков CSS с описанием структуры CSS Grid, его возможностей и предназначения.

С декабря 2017 это предложение находится в стадии рекомендации. Это практически финальная стадия спецификации, поэтому разработчики всех современных браузеров уже реализовали её.

В заключение хотелось бы отметить, что Grid разработан в качестве дополнения к уже существующим техникам размещения элементов, а не для их замещения: комбинирование Grid и Flexbox приветствуется. И это отдельно отмечено в спецификации.

**А.А. Зубов** (ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

## **ПЕРЕНОС ВЫЧИСЛЕНИЙ НА ГРАФИЧЕСКИЙ ПРОЦЕССОР**

В последние годы индустрия выпуска графических процессоров пережила небывалый стресс. Виной тому стало применение этих процессоров в вычислениях для получения криптовалют с помощью специальных ферм видеокарт. Видеокарты, с их графическими процессорами, стали «серебряной пулей», которая позволила снять вычислительную нагрузку с центрального

процессора (CPU) и перенести большую часть на графический процессор (GPU), который не так активно задействуется компьютером в повседневных задачах.

В рамках выполняемой работы была разработана библиотека классов на основе платформы OpenCL (Open Computing Language), которая используется для непосредственного взаимодействия с API (Application Programming Interface) графического процессора. Полученная библиотека предоставляет набор различных инструментов для оптимизации вычислений различными способами и является совместимой с платформой .NET.

Выбор пал на OpenCL после анализа существующих платформ и их подробного сравнения. Решающими факторами выбора OpenCL стали аппаратная и программная независимость этой платформы, а также количество существующих решений и библиотек, которые сильно упрощают разработку.

Библиотека разрабатывалась на языке C# с использованием дополнительной библиотеки классов Cloo, которая является обёрткой над OpenCL. Эта библиотека предоставляет удобные абстракции для низкоуровневых сущностей. К таким абстракциям можно отнести две основные сущности: KernelMemory и ComputeKernel, реализующие классическую парадигму вычислений: Есть сущность, которая производит сами вычисления (ComputeKernel), но для этого процесса ей необходимы системные ресурсы, память. Ресурсы выделяются с помощью сущности KernelMemory. Эта абстракция имеет доступ к механизмам резервирования областей в доступном адресном пространстве памяти. Поверх выделенных ресурсов и будет работать сущность ComputeKernel, производящая вычисления.

Для демонстрации были разработаны тестовые приложения, у которых была увеличена производительность путём внедрения разработанной библиотеки.

**Ю.А. Иванова** (ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Е.А. Ружицкая**, канд. физ.-мат. наук, доцент

## **РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ «SHEEPS»**

Приложение «Sheeps» было разработано на языке программирования C# с использованием возможностей среды разработки Unity.