

тельского интерфейса для *web*-сайтов и мобильных приложений, особенно для разработки одностраничных приложений, в которых данные отображаются в режиме реального времени [1]. В настоящее время это одна из самых популярных интерфейсных библиотек *Javascript* на рынке. *React* представляется идеальным инструментом для создания масштабируемых *web*-приложений, особенно в тех ситуациях, когда приложение представляет *SPA*.

*Node.js* представляет среду выполнения кода на *JavaScript*, которая построена на основе движка *JavaScript Chrome V8*, который позволяет транслировать вызовы на языке *JavaScript* в машинный код. *Node.js* прежде всего предназначен для создания серверных приложений на языке *JavaScript*.

*Angular.js* представляет собой *opensource JavaScript*-фреймворк, использующий шаблон *MVC*. Для описания интерфейса используется декларативное программирование, а бизнес-аналитика отделена от кода интерфейса, что позволяет улучшить тестируемость и расширяемость приложений. Кроме того, *Angular.js* поддерживает такие функциональности, как *AJAX*, управление структурой *DOM*, анимация, шаблоны, маршрутизация и так далее.

Каждая из технологий имеет свои возможности и ограничения в индивидуальном порядке, что предоставляет разработчику широкий выбор инструментов разработки. Используя перечисленные технологии, повышается рейтинг веб-приложений и открываются новые возможности для привлечения новых клиентов.

## Литература

1 Бэнкс, А. А. *React и Redux. Функциональная веб-разработка* / А. А. Бэнкс, Е. К. Порселло. – М.: Бином-Пресс; СПб.: Питер принт, 2019. – 329 с.

**Е. В. Синицин, Е. В. Комракова**  
(ГГТУ им. П. О. Сухого, Гомель)

## РОЛЬ МАТЕМАТИКИ В БАЛАНСЕ ИГР

Очень часто разработчики игр случайным образом балансируют игровые сущности. В большинстве случаев такой подход ведёт к неприятным проблемам на этапе контролирования и прогнозирования

изменений. Поэтому хороший баланс должен иметь прочную основу, подкреплённую логикой и математикой. Это позволит: системно вводить новый игровой контент; прогнозировать его потребление игроками; задавать правила ввода и вывода сущностей; заранее предвидеть проблемы до того момента, как они станут критичными; снизить вероятность возникновения ошибок.

Задача математической части баланса чаще всего сводится к методу базисных характеристик, который основывается на вводе некоторого дополнительного параметра мощности определённого предмета. Данный параметр позволяет разработчикам получить возможность сравнивать по одному числу различные пары предметов. У какого предмета значение параметра мощности больше, тот предмет и имеет большее преимущество. В формулу расчёта мощности определённого предмета обязательно должны входить все его характеристики. Для того чтобы ускорить вычисления и уменьшить количество ошибок в расчётах, чаще всего, создатели игр используют уже заранее разработанный калькулятор баланса. Его цель заключается в вычислении добавочного параметра мощности и сравнении игровых объектов между собой. Также в подобных калькуляторах нередко устанавливаются граничные условия для фильтрации плохо сбалансированных предметов. Например, самое мощное оружие не должно иметь урон в секунду, более чем в два раза превышающий урон в секунду самого слабого оружия.

Однако важно заметить, что баланс игры всегда требуется проверять в условиях реальной игры на случай возникновения исключительных, ошибочных ситуаций. Без последующей проверки сбалансированности игровых сущностей сложно определить, насколько стабильная получилась система, так как баланс игры чаще зависит не только от математики, но и от дизайна игровых уровней.

**Д. Ю. Смахтин**  
(ГГУ им. Ф. Скорины, Гомель)

## **РАЗРАБОТКА ИГРОВОГО ПРИЛОЖЕНИЯ 2048**

Создание видеоигр является одним из наиболее крупных сегментов индустрии развлечений. Масштабы игровой индустрии сопоставимы с киноиндустрией, а по скорости роста за последние пять лет индустрия видеоигр существенно ее опередила.