

УДК 004.422

## ХРАНЕНИЕ СОСТОЯНИЙ ИГРЫ И КОНТРОЛЬ ПОТОКА ИЗМЕНЕНИЯ

Жумиков. Е. О. (Университет ИТМО)

Научный руководитель – Малышева. Т. А., к.т.н., доцент  
(Университет ИТМО)

В данной работе представлен архитектурный подход для организации хранения состояний игры и контроля потока их изменений. Также в работе рассматривается пример реализации этой архитектуры для многопользовательской игры в мессенджере. Данные наработки позволяют определить возможность и необходимость использования предложенной архитектуры в сторонних разработках, а также принципы, по которым следует внедрять представленную разработку.

Так как игровая разработка является довольно узкоспециализированной нишей в программной инженерии, а большинство решений являются коммерческими, с закрытым исходным кодом, в индустрии довольно мало общепринятых архитектурных подходов. Одной из важных проблем в данном контексте является организация доступа к состоянию игры из множества программных компонентов. Часто доступ к состоянию возможно получить в результате вызова цепочки методов. Существует архитектурный подход, решающий эту проблему, называемый «Entity-Component-System», но важно отметить, что он не предполагает обработки событий, что требует дополнительных изменений в архитектуре приложения.

Данная работа основана на Flux-архитектуре веб-приложений и заключается выделении нескольких компонентов: Action, Executor, Data и Dispatcher.

- Data представляет любые данные игровой сессии, которые могут быть модифицированы в ответ на действия игрока или внутриигровое событие.
- Action является декларативным описанием модификации состояния.
- Executor – это сервис, который позволяет изменять состояние на основе декларативных описаний.
- Dispatcher – сервис, выполняющий диспетчеризацию процедур изменения состояний.

Описанные компоненты архитектуры позволяют отслеживать и отлаживать изменения в игровом состоянии, разделить ответственность приложения и даже хранить эти модификации для передачи между, например, клиентами или гибкой системы вещей и игровых эффектов.

Результатом данной работы является архитектурный подход, позволяющий организовать хранение и изменения состояния в играх, а также его программная реализация. Для апробации подхода в качестве игрового приложения использовалась многопользовательская асинхронная игра для мессенджеров, где данный подход позволил гибко описывать игровые компоненты и упростить доступ к состоянию игровой системы.

Жумиков. Е. О. (автор)

Подпись

Малышева Т.А. (научный руководитель)

Подпись