

Unreal Engine также предлагает различные режимы физики, которые можно использовать в игре. Например, физика может быть отключена для некоторых объектов, что позволяет создавать анимации и спецэффекты. Режимы физики также могут быть настроены на разных уровнях детализации, в зависимости от требуемой производительности игры.

Кроме того, в Unreal Engine имеются инструменты для создания динамических эффектов, таких как разрушение и деформация объектов. Это позволяет создавать реалистичные эффекты взрывов, столкновений, а также моделировать различные физические явления, такие как гравитация, ветер, давление и т.д.

Одним из главных преимуществ физики в Unreal Engine является поддержка виртуальной реальности. Виртуальная реальность позволяет создавать игровые миры, где игрок может ощущать реалистичные эффекты и взаимодействовать с объектами более натуральным образом. Физика в Unreal Engine позволяет создавать реалистичные эффекты столкновений, разрушений и деформаций, что существенно улучшает впечатление от игры.

Таким образом, физика в Unreal Engine играет важную роль в создании реалистичных и интерактивных игровых миров. Система физики в Unreal Engine позволяет создавать различные эффекты и взаимодействия между объектами, что способствует улучшению качества и реалистичности игры.

А. В. Коваленко

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **О. М. Дерюжкова**, канд. физ.-мат. наук, доцент

ИСПОЛЬЗОВАНИЕ ФРЕЙМВОРКА SPRING В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ JAVA

Долгое время для создания web-приложений программисты пользовались JavaBeans. Это такие классы в языке java, которые должны удовлетворять определённым условиям. Однако данные классы были сильно ограничены в работе с транзакциями, безопасностью и многопоточностью. А перечисленные выше три пункта до сих пор являются одними из самых важных для больших приложений.

И уже в июне 2003 года появилось решение этой проблемы, когда Род Джонсон опубликовал первую версию Spring [1].

На сегодняшний день spring это самый популярный фреймворк для разработки web-приложений на таком языке программирования как

java. Само понятие фреймворк можно сравнить с обычной библиотекой, однако есть значимая разница. Говоря простым языком подключая библиотеку, вы создаёте объекты классов, которые содержатся в этой библиотеке, а вызывая нужные методы, получаете требуемый результат. Но для работы с фреймворками нужен немного другой подход. Классы каждый пользователь пишет сам и так же сам придумывает логику для их работы, а вот объекты этих классов создаёт фреймворк. Довольно частое явление, если ваши классы будут имплементировать какие-либо интерфейсы из фреймворка или даже являться наследниками различных классов из него, тем самым получая дополнительную функциональность, которая уже написана за вас. Но не всегда это работает только так [2].

Кто же использует Spring? Это:

- Java developers, которые работают над созданием мобильных или web-приложений, Spring используется во многих направлениях. Но в основном речь идёт о масштабных приложениях;

- Kotlin developers, тут причина заключается в том что этот язык был разработан на java и участвовал в написания spring;

- программисты, главным требованием для которых является проектирование. Spring используется как шаблон для приложений;

- в редких случаях Spring является инструментом для backend developers, в качестве написания серверной части на java [3].

Рассмотрим структуру Spring Framework (рисунок 1).

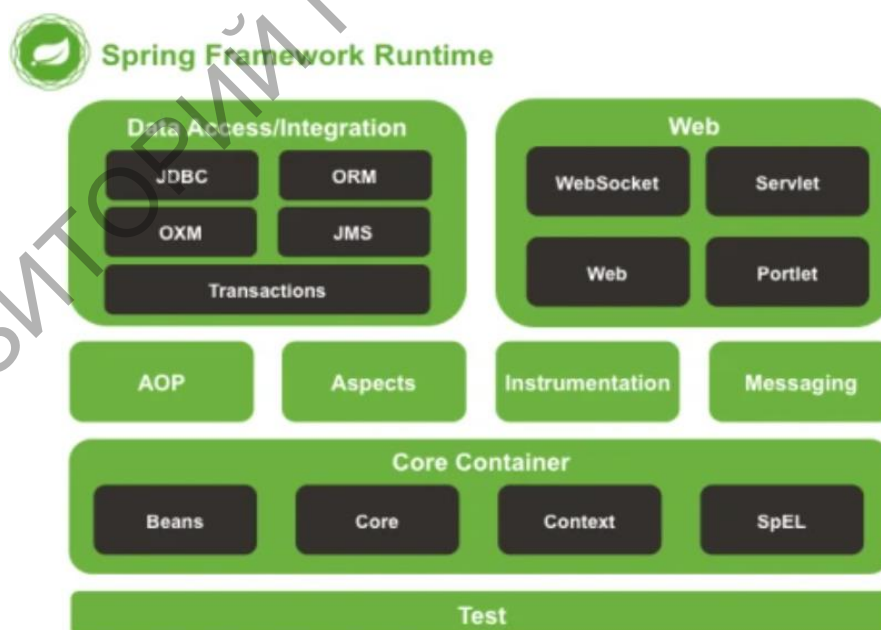


Рисунок 1 – Структура программы Spring Framework

Spring использует исключительно те модули, которые необходимы программисту, и игнорирует другие модули. Модули из которых состоит Spring:

- data access;
- web;
- core;
- и другие.

Data access благодаря инструментам содержащимся в нём позволяет работать с базами данных. В случае web работать с сетью. В этом фреймворке есть такие понятия как DI/IoC. Рассмотрим их подробнее.

DI имеет два определения. Dependency Inversion (не допущение такой ситуации когда один класс напрямую завязан на другом) и Dependency Injection (например, это когда Spring создаёт объекты наших классов, а от нас требуется просто «попросить» об этом).

IoC (Inversion of Control) переводится на русский как инверсия управления. В начале статьи уже говорилось о разнице библиотеки и фреймворка. Так вот в случае использования Spring программа сама вызывает в необходимый момент код, который в свою очередь написан пользователем. То есть, работой программы управляет уже не программист.

Главные способы конфигурации Spring приложения:

- используя xml файлы;
- используя java-конфиги;
- автоматическая конфигурация.

Среди программистов самый востребованный способ – третий. В случае невозможности использования автоматической конфигурации предпочтение отдается языку java. Наименьший приоритет из всех конфигураций имеют xml файлы. Spring позволяет комбинировать эти способы между собой, например, если программа способна настроить что-то самостоятельно, то она это выполнит. В случае необходимости добавить особые настройки программист воспользуется java-конфигами [2].

Таким образом, java программист имеет гибкий и полезный инструмент, который помогает создавать web-приложения.

Литература

1. Фреймворк Spring: зачем он нужен, как устроен и как работает [Электронный ресурс] / Официальный сайт «ЧОУ ДПО «Образовательные технологии «Скилбокс (Коробка навыков)». – Режим доступа: <https://skillbox.ru/media/code/freymvork-spring-zachem-on-nuzhen-kak-ustroen-i-kak-rabotaet/>. – Дата доступа: 06.03.2023.

2. Spring для ленивых. Основы, базовые концепции и примеры с кодом [Электронный ресурс]. / Интерактивный онлайн-курс по изучению Java-программирования с нуля. – Режим доступа: <https://javarush.com/groups/posts/spring-framework-java-1>. – Дата доступа: 06.03.2023.

3. Spring [Электронный ресурс] / Честные истории о карьере в IT. – Режим доступа: <https://blog.skillfactory.ru/glossary/spring/>. – Дата доступа: 06.03.2023.

К. А. Кошара

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. С. А. Лукашевич, ст. преподаватель

СОЗДАНИЕ 3D-ГРАФИКИ В БРАУЗЕРЕ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ THREE.JS

Three.js это революционная библиотека JavaScript, которая полностью изменила способ, которым разработчики создают 3D-компьютерную графику в Интернете. Впервые он был разработан в 2010 году блестящим умом Рикардо Кабельо, также известного под своим онлайн-псевдонимом «Mr. Doob». С момента своего создания Three.js стала одним из самых популярных и широко используемых инструментов для разработки 3D-графики, что делает ее важным компонентом экосистемы веб-разработки [1–4].

История создания Three.js началась с того, что Рикардо Кабельо работал над проектом, который требовал создания 3D-анимации на JavaScript. Он быстро осознал сложность создания 3D-графики в веб-браузере и необходимость в библиотеке, которая упростила бы этот процесс. Это побудило Кабельо разработать Three.js, которую он выпустил в апреле 2010 года.

Изначально разработанная для работы с WebGL, библиотека Three.js теперь совместима с несколькими другими технологиями рендеринга, включая Canvas и SVG. Библиотека предоставляет широкий спектр функций, таких как освещение, материалы и камеры, которые позволяют без особых усилий создавать потрясающую 3D-графику.

Кроме того, Three.js предлагает несколько инструментов для работы с 3D-моделями и позволяет импортировать модели из различных форматов файлов. Она постоянно развивается и совершенствуется, с