|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практическим работам №9-10**

по дисциплине «Системная и программная инженерия»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-13-19 | Изотина А. А.  Малютина В. А.  Назаркин В. И.  Тузов И. В. |
| **Проверил:** | доцент Туманова М.Б. |

МОСКВА 2022 г.

**Содержание**

[1 Практическая работа №9 3](#_Toc101174004)

[1.1 Ход работы: 3](#_Toc101174005)

[1.1.1 Выбор модели управления процессом разработки 3](#_Toc101174006)

[1.1.2 Организация процесса разработки 4](#_Toc101174007)

[1.2 Вывод: 4](#_Toc101174008)

[2 Практическая работа №10 5](#_Toc101174009)

[2.1 Ход работы: 5](#_Toc101174010)

[2.1.1 Документация разработчика 5](#_Toc101174011)

[2.1.2 Документация пользователя 6](#_Toc101174012)

[2.2 Вывод: 9](#_Toc101174013)

[3 Заключение 10](#_Toc101174014)

# Практическая работа №9

**Цель:**

Получить навыки по организации процесса командной разработки с применением различных методологий управления процессом разработки, а также технологий командной разработки.

## Ход работы:

### Выбор модели управления процессом разработки

На данный момент в сфере IT существует много различных методологий и технологий для работы над командным проектом. Все они разнятся в зависимости от требований заказчика и обеспеченности ресурсами команды разработчиков. Среди жизненных циклов ПО есть основные: каскадная, итерационная, инкрементная и спиральная. Каждый из этих циклов позволяет удовлетворить заданным целям разработки, однако у каждого из них различные временные ограничения.

При разработке компьютерной игры-платформера “Jumpkin” команда следовала спиральному жизненному циклу ПО. Сначала разрабатывались требования к этапу, потом создавались концепты и идеи, затем они внедрялись в игру, далее проводился бета-тест игры с внедрёнными обновлениями и игра дальше забиралась на доработку. Далее цикл повторялся, и тем самым в короткие сроки достигались цели, заданные командой в начале этапа, и можно было приступать к новому.

Кроме того, для успешного завершения работы над проектом необходимо уметь грамотно планировать время, ресурсы и быть постоянно в контакте с клиентом. В этой части IT-сфера уже имеет большой опыт и готова предложить решения по управлению процессом разработки. Среди моделей управления процессом разработки, подходящих под критерии команды, есть такие как Kanban и RAD. Они гибкие и позволяют работать с сжатыми сроками, ограниченными ресурсами благодаря продуманному этапу планирования.

Также имея сжатые сроки для разработки проекта (4 месяца), команда следовала гибкой модели RAD (Rapid Application Development). Аналитиком и менеджером использовались CASE-средства для моделирования и проектирования разрабатываемого проекта, далее созданные прототипы передавались разработчику, который в своё очередь создавал на их основе рабочий прототип и демонстрировал затем его команде. Данная методология действительно позволяет быстро разрабатывать продукт и получать ожидаемый результат в краткие сроки, работая небольшой командой.

### Организация процесса разработки

Для комфортной и быстрой разработки было выбрано следующее программное обеспечение:

1. Visual Studio Code
2. Visual Studio 2019
3. Unity
4. Blender
5. Photoshop

Для централизованного хранения и контроля за состоянием и изменением проекта был создан удалённый гит-репозиторий, ссылка приведена ниже: <https://github.com/VanoPremudriy/jumpkin-project>

## Вывод:

В ходе выполнения практической работы была определена модель управления процессом разработки, жизненный цикл проекта и организация процесса разработки.

# **Практическая работа №10**

**Цель:**

Получить навыки по формированию документации разработчика, формирование документации пользователя.

## Ход работы:

### Документация разработчика

В ходе работы над проектом была создана документация разработчика. Разработка игры велась на языке C# и на движке Unity. Документация кода была сгенерирована таким программным обеспечением, как Doxygen. В документацию разработчика были включены публичные методы и поля классов, приватные поля и методы были проигнорированы. К каждому классу было дано краткое пояснение и иерархическое дерево зависимостей. Внутри каждого класса были кратко описаны методы, отвечающие за логику игры, а также использующиеся в них поля. Документация была сгенерирована в формате HTML-файлов. Каждому классу отведена отдельная страница, на которую можно перейти через навигационное меню. Главной страницей документации с общей информацией о проекте является страница index.html. Примеры сгенерированной документации приведены на Рисунке 2.1, 2.2 и 2.3.

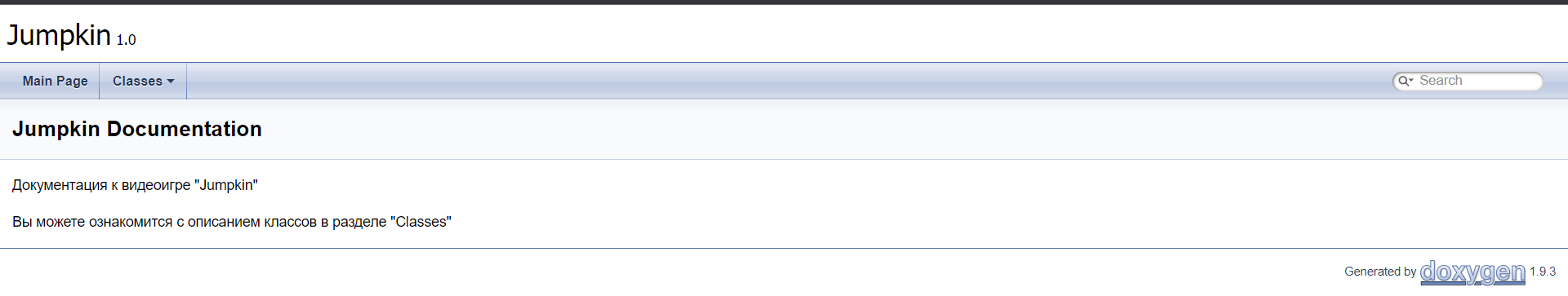


Рисунок 2.1 - Главная страница документации

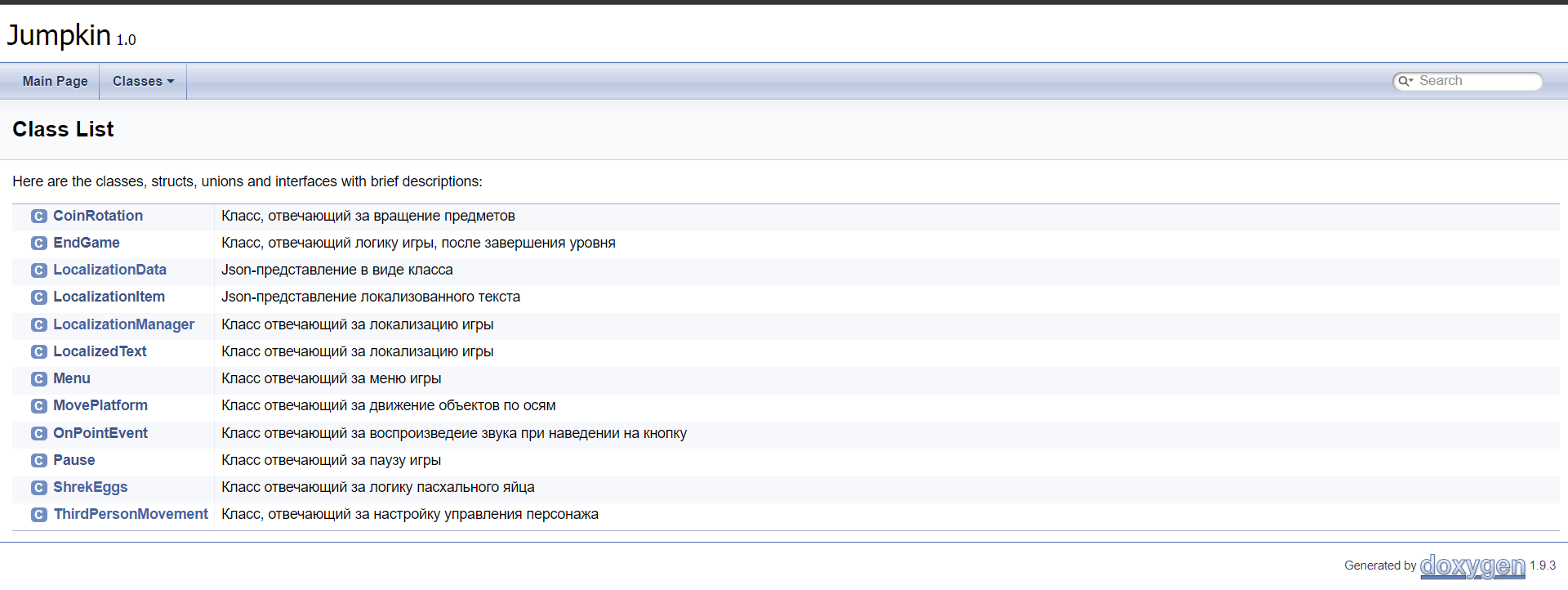


Рисунок 2.2 - Страница с классами

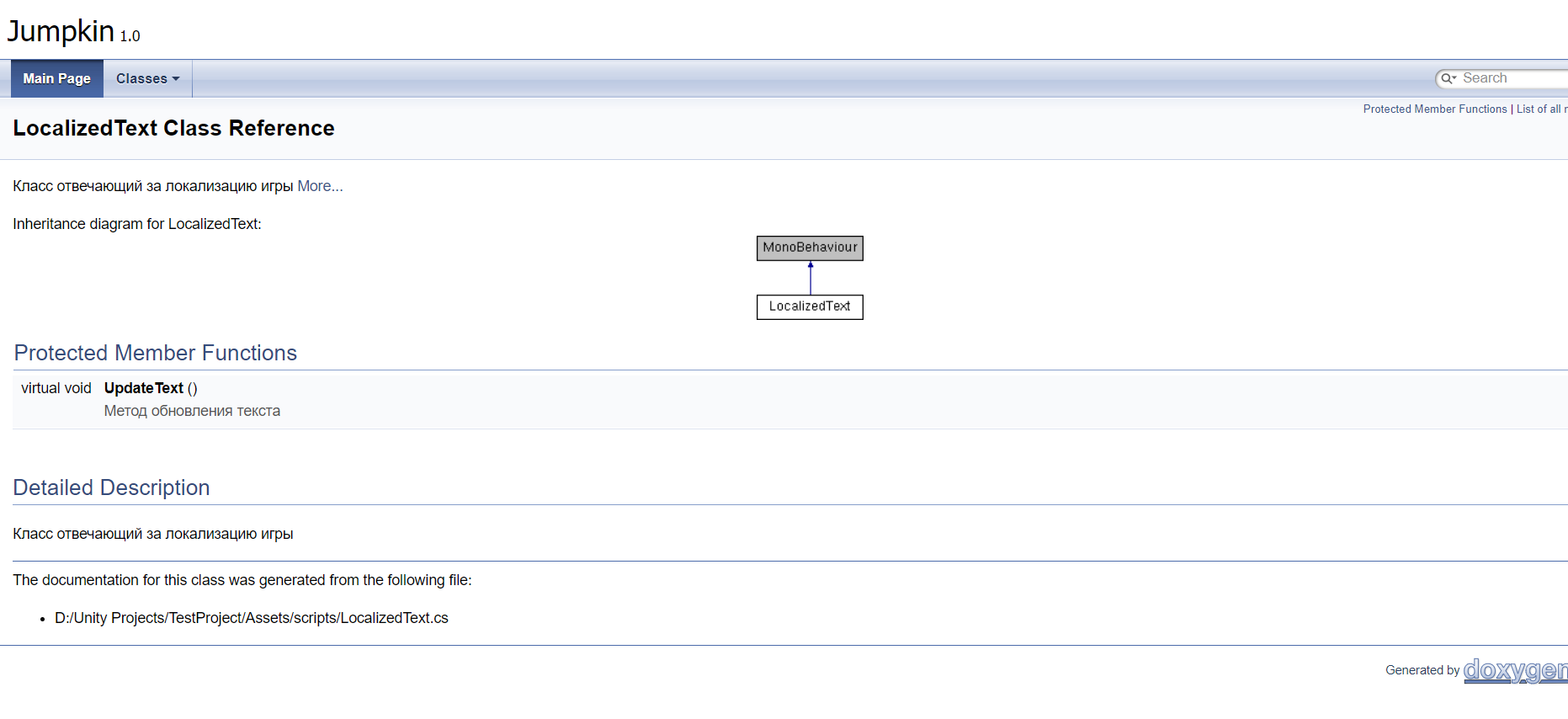


Рисунок 2.3 - Страница описания класса

Полная версия документации прикреплена к отчету.

### Документация пользователя

Для удобства пользованием разрабатываемой компьютерной игрой было создано руководство пользователя на основе вики-сайтов. В качестве основы был взят портал Fandom. Данный сервис хорошо подходит для документирования функций игр, так как на нём размещена не одна компьютерная игра, следовательно, можно брать пример по оформлению руководства пользователя с других вики-пособий.

На вики-сайте по игре Jumpkin представлены 3 основные на данный момент страницы: страница тыквы, страница фермы и страница геймплея. На каждую из них пользователь может перейти и получить необходимую информацию для прохождения игры.

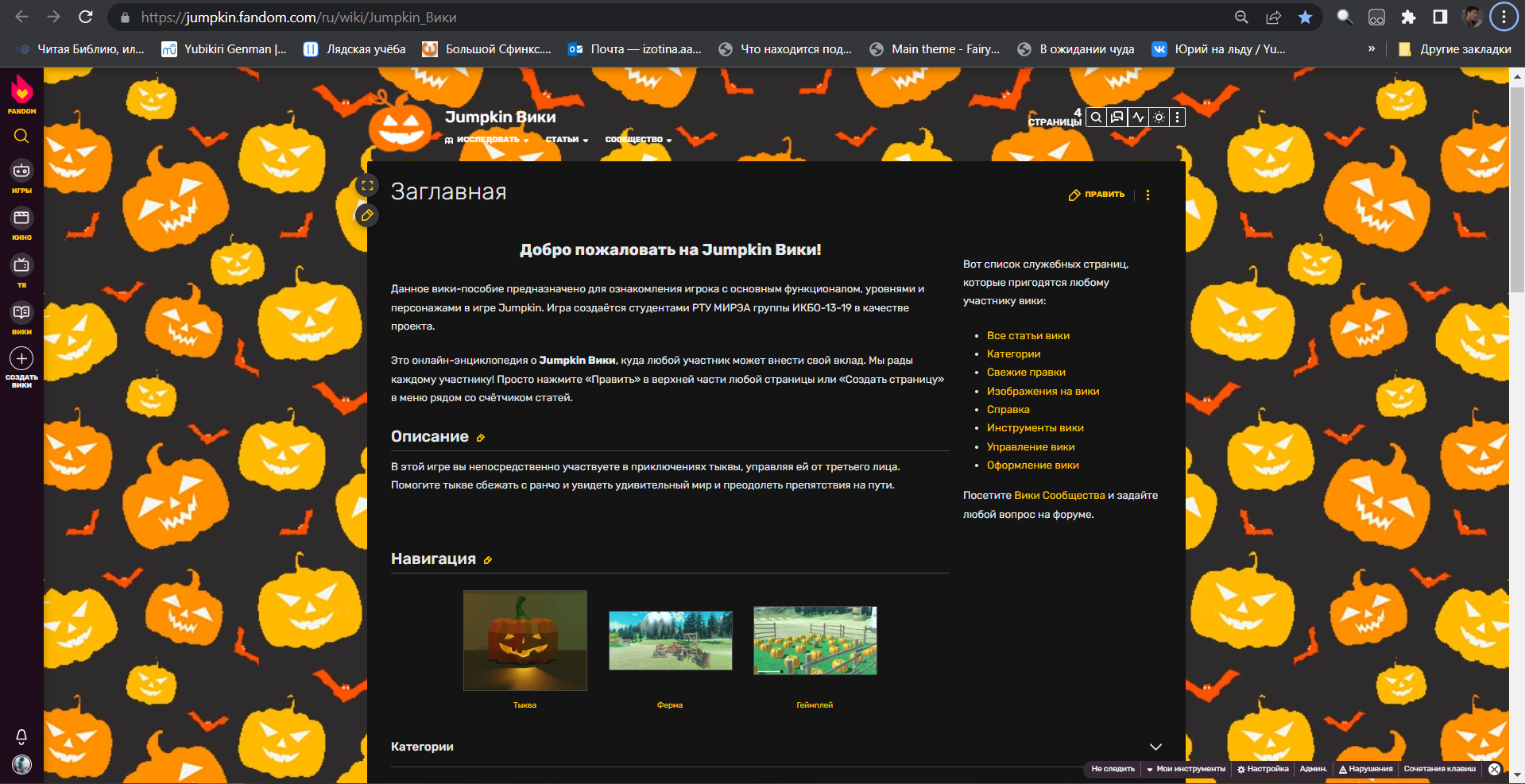


Рисунок 2.4 – Заглавная страница

На странице Тыквы предоставлена основная информация о тыкве, о её предыстории, а также есть информация по управлению персонажем.

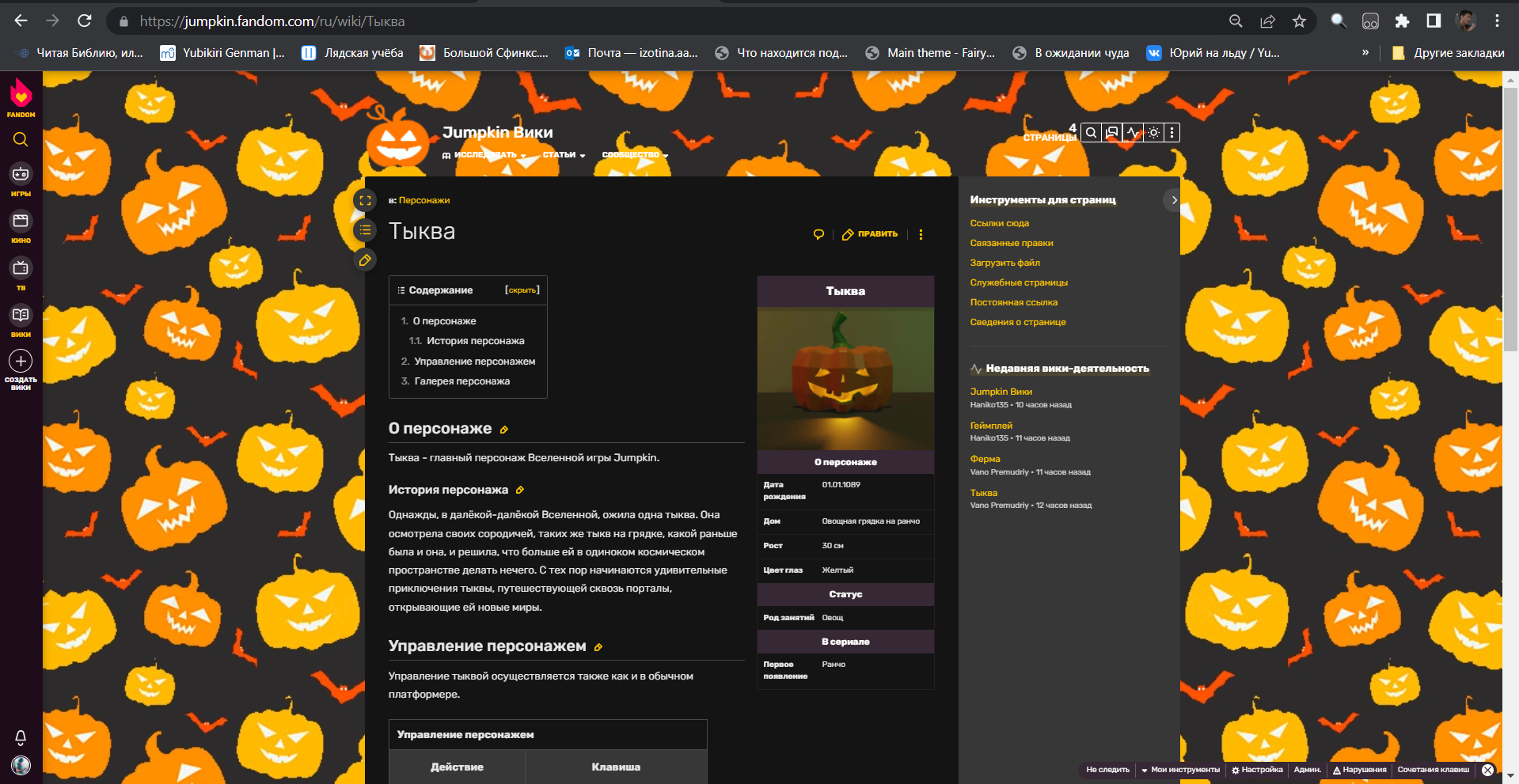


Рисунок 2.5 - Страница Тыквы

На странице Фермы дана информация о локации в целом, о её наполнении и представлена галерея со снимками этой локации.

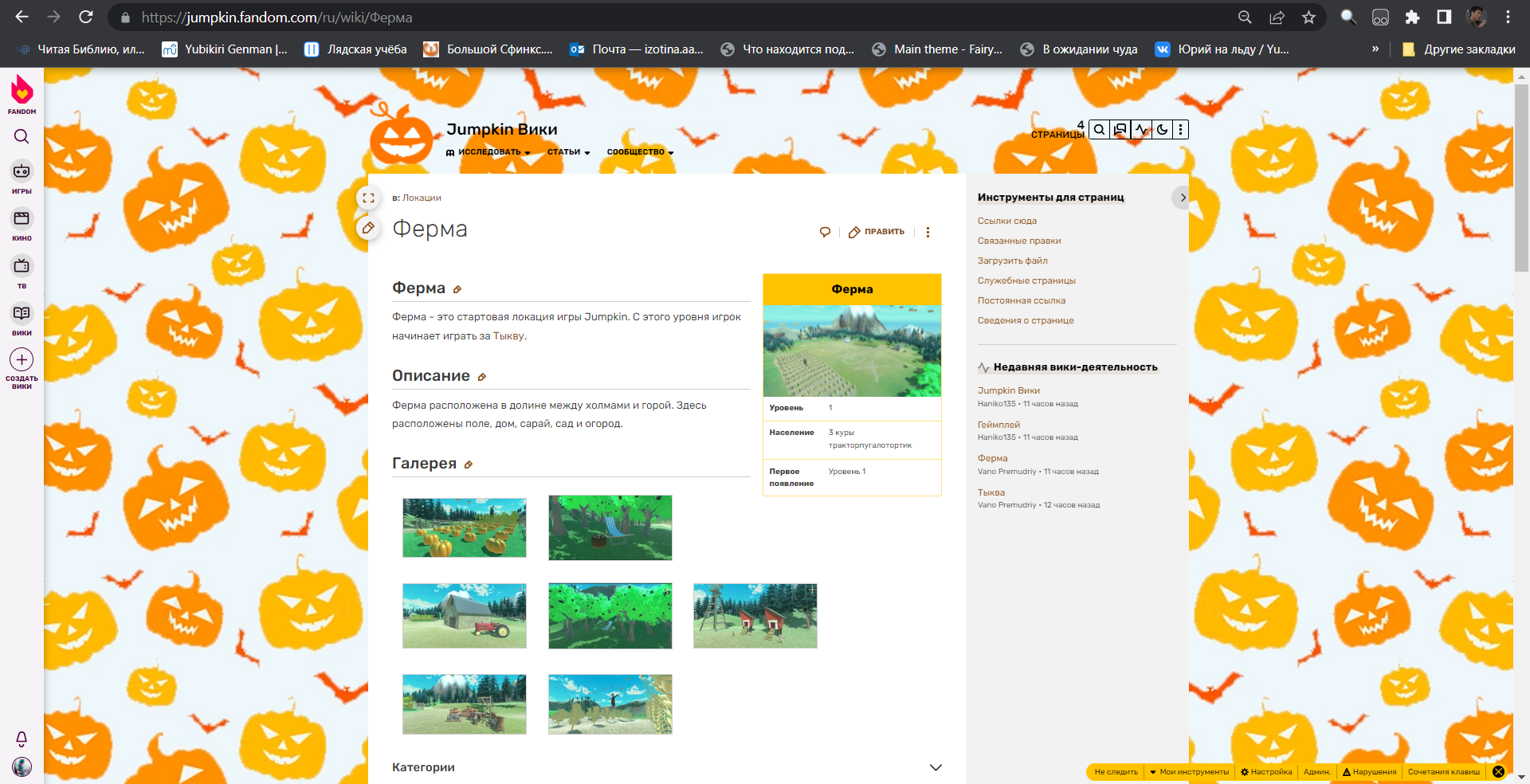


Рисунок 2.6 - Страница Фермы

На странице Геймплея написано об основных правилах игры и о игре в целом.

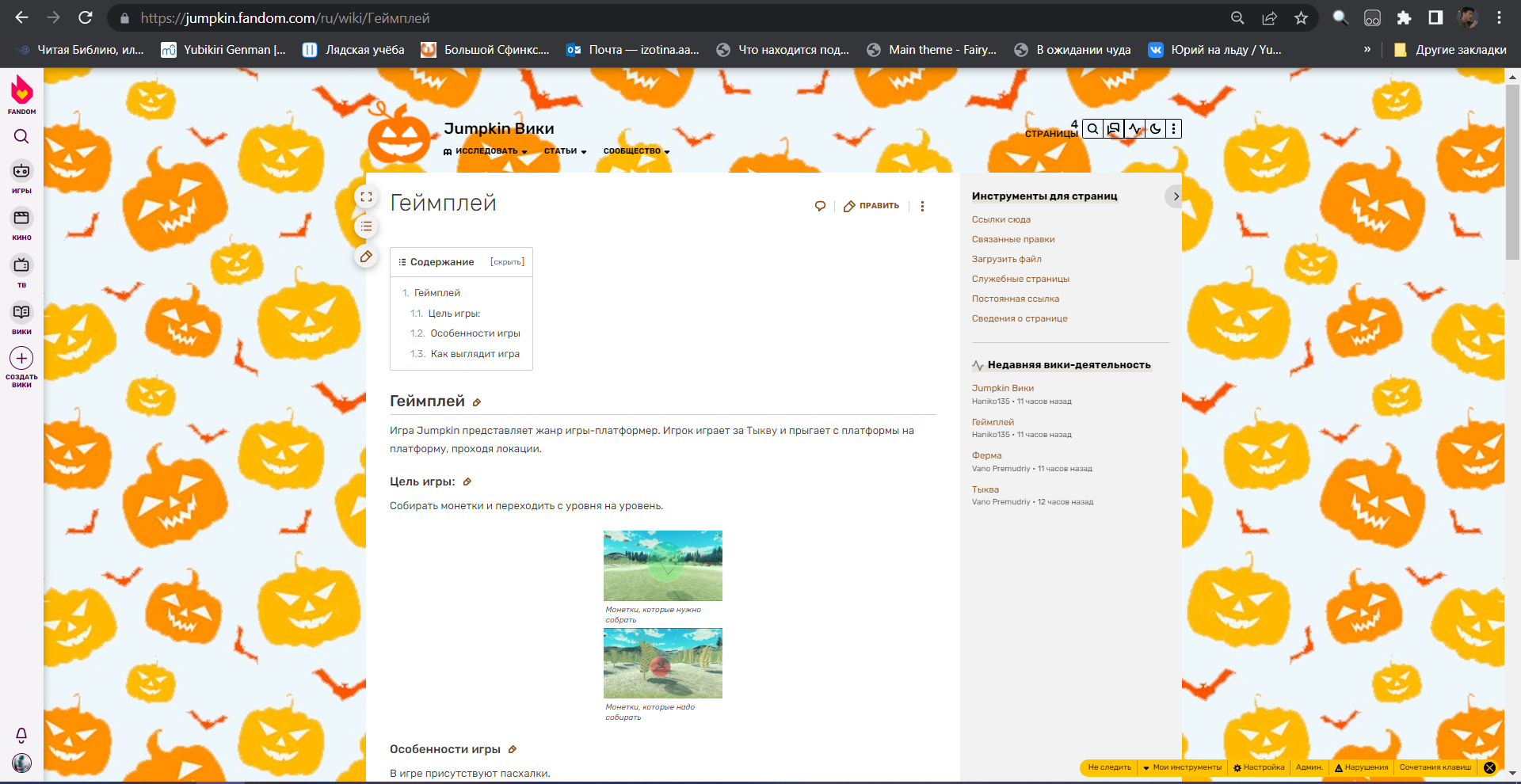


Рисунок 2.7 - Страница геймплея

Ссылка на Jumpkim Вики представлена далее: <https://jumpkin.fandom.com/ru/wiki/Jumpkin_%D0%92%D0%B8%D0%BA%D0%B8>

## Вывод:

В ходе выполнения практической работы была сгенерирована документация пользователя и документация разработчика. Документация разработчика приведена ниже.

# Заключение

В ходе практических работ были освоены технологии для планирования и управления процессом разработки, получены знания о жизненном цикле ПО, сгенерирована и оформлена документация разработчика, а также создано руководство пользователя.