**Software Requirements**

**Specification for**

**DOODLE JUMP**

**Version 1.0 approved**

**Prepared by Svetlov Pavel**

**20.03.2025**

**СОДЕРЖАНИЕ**

**[1. Вступление 3](#_Toc32138)**

[1.1.Цель 3](#_Toc1405)

[1.2 Правила оформления документов 3](#_Toc2391)

[1.2.1. Сокращения 3](#_Toc4527)

[1.3 Целевая аудитория 3](#_Toc30437)

[1.3.1. Разработчик 3](#_Toc1405)

[1.3.2 Тестировщик 3](#_Toc2391)

[1.3.3 Тех.лид 3](#_Toc4527)

[1.4 Краткое содержание проекта и его цель 3](#_Toc30437)

[1.5 Ссылки на источники 4](#_Toc24245)

**[2. Общее описание проекта 4](#_Toc19410)**

[2.1. Функция продукта 4](#_Toc1405)

[2.1.1 Что присутствует в игре 4](#_Toc2391)

[2.1.2 Игровой процесс 4](#_Toc4527)

[2.2 Операционная среда 5](#_Toc30437)

[2.3 Ограничения при проектировании и реализации 5](#_Toc1405)

[2.4 Документация пользователя 5](#_Toc2391)

[2.5 Допущения и зависимости 5](#_Toc4527)

**[3. Функциональные требования к проекту 5](#_Toc1028)**

[3.1. Главное меню 5](#_Toc1405)

[3.2. Игровой процесс 6](#_Toc2391)

[3.3. Поражение 1](#_Toc4527)0

[3.4. Быстрый рестарт игры 1](#_Toc30437)1

[3.5. Остальные функции 1](#_Toc24245)1

**[4. Требования к внешнему интерфейсу 1](#_Toc27227)2**

[4.1. Пользовательский интерфейс 1](#_Toc8664)2

[4.2. Аппаратный интерфейс 1](#_Toc16640)3

[4.3. Программный интерфейс 1](#_Toc21469)3

**[5. Другие нефункциональные требования 1](#_Toc18106)4**

[5.1. Требования к производительности 1](#_Toc1405)4

[5.1.1 Стабильность работы 1](#_Toc2391)4

[5.1.2 Быстрота загрузки экранов 1](#_Toc4527)4

[5.1.3 Обработка музыки 1](#_Toc30437)4

[5.2 Требования к безопасности 1](#_Toc1405)4

[5.2.1 Соглашение о конфиденциальности 1](#_Toc2391)4

[5.2.2 Сбор данных 1](#_Toc4527)4

[5.3 Атрибуты качества программного обеспечения 1](#_Toc30437)4

[5.3.1 Расширяемость 1](#_Toc1405)4

5.[3.2. Совместимость 1](#_Toc2391)5

5.[3.3 Системные требования 1](#_Toc4527)5

5.[3.4. Интерфйес 1](#_Toc30437)5

[5.4. Бизнес-правила 1](#_Toc24245)5

[5.4.1. Сбор и аналитика данных 1](#_Toc1405)5

5.[4.2. Маркетинг 1](#_Toc2391)5

[5.4.3. Обратная связь с сообществом 1](#_Toc4527)5

**[6. Приложения 1](#_Toc13945)6**

[6.1. Приложение A: Словарь 1](#_Toc28198)6

[6.2. Приложение B: Диаграмма 1](#_Toc21137)7

**1. Вступление**

1.1. Цель

Цель этой SRS представить подробное описание игры Doodle Jump для Web-Pc устройств. Документ предназначен для обеспечения ясного понимания ожидаемых результатов и руководства для разработчиков, содействуя успешному созданию игрового продукта. Здесь будет описано особенности геймплея, интерфейс, функциональные и нефункциональные возможности, а также ограничения для разработки игры.

1.2. Правила оформления документов

1.2.1. Сокращения

*ПИ* - персонаж игрока

1.3. Целевая аудитория и рекомендации по прочтению

Каждому читателю, независимо от их зоны ответственности, нужно прочитать

“**Вступление**”, “**Общее описание проекта**”

1.3.1. Разработчик

В первую очередь следует ознакомиться с разделом “**Функциональные требования к проекту**” для понимания полного функционала проекта, а также “**Требования к внешнему интерфейсу**” для понимания как пользователь взаимодействует с интерфейсом.

1.3.2. Тестировщик

Стоит ознакомиться с разделом “**Функциональные требования к проекту**”, “**Требования к внешнему интерфейсу**” и “**Требования к производительности**” для понимания функционала проекта и моментов где стоит провести тесты

.

1.3.3. Менеджер проекта/Тех. лид.

Этому классу стоит ознакомиться **со всеми разделами** данного документа для выбора стратегии и реализации проекта.

**1.4. Краткое содержание проекта и его цель**

Проект представляет собой 2D-игру в жанре аркада. Главная цель игры - помочь «Дудлеру» (Doodle) добраться как можно выше по платформам и набрать как можно большее количество очков. Героем можно управлять с помощью всего двух кнопок на ПК (стрелка влево, стрелка вправо).

1.5. Ссылки на источники

Doodle Jump wiki:

<https://ru.wikipedia.org/wiki/Doodle_Jump>

Шаблон SRS:

https://docs.google.com/document/d/11VijWKtIiPqDzkZWk\_7Mv9ddiC7hujqL/edit?usp

=drive\_link&ouid=103322450875369337478&rtpof=true&sd=tru

**2. Общее описание проекта**

2.1. Функции продукта

2.1.1. Что присутствует в игре:

**Бесконечный геймплей; простое управление;** п**латформы разных типов.**

**2.1.2.**

**Игровой процесс**

● **ПИ:**

○ **Управление:** игрок может сменить только направление движения с помощью стрелок на клавиатуре.

○ **Движение:** *ПИ* автоматически движется в одну из сторон вверх, или он может двигаться влево или вправо, в зависимости от того, на какую кнопку нажмет игрок.

○ **Смерть:** *ПИ* может умереть, упав с платформы вниз.

**2.2. Операционная среда**

Проект должен быть совместим с PС

Память - до 100 МБ

Оперативная память: 1ГБ ОЗУ

**Требования к PC:** наличие клавиатуры, монитора и мыши(по возможности), Windows XP и более.

**2.3. Ограничения при проектировании и реализации**

Игра должна создаваться в среде PyCharm Community при помощи языка Python. Код должен быть чистым и понятным для чтения.

Проект на должен превышать 100 МБ память.

**2.4. Документация пользователя**

В игре не будет какой либо справки или подсказок, т.к. в ней должно быть все интуитивно понятно. Игра предназначена для любых возрастов.

**2.5. Допущения и зависимости**

**Производительность устройств:** хоть проект и не является требовательным, при тестировании стоит проверить на каких устройствах проект будет корректно работать.

1. **Функциональные требования к проекту**

**3.1. Главное меню**

**Название функции:** Загрузка уровня

**Актер:** Пользователь

**Приоритет:** имеет приоритет выше среднего

**Стимул:** Пользователь нажимает на кнопку “Играть**”**

**Реакция:** Запуск игрового процесса

**Последовательность действий:** 1) Игрок нажал на кнопку “Играть”; 2) Игра началась

**Примечания к выполнению:** Должен появиться экран, где игроку нужно будет нажимать стрелки влево или вправо, чтобы прыгать по платформам

**3.2. Игровой процесс**

**Название функции:** Начало геймплея

**Актер:** Игра

**Приоритет:** Имеет высокий приоритет

**Стимул:** Пользователь взаимодействует с игрой

**Реакция:** Запуск игрового процесса

**Последовательность действий:** 1) Игрок нажал на кнопку “Играть”; 2) Начало игрового процесса

**Примечания к выполнению:** Через несколько секунд начинается игровой процесс

**Название функции:** Движение

**Актер:** Пользователь

**Приоритет:** Имеет самый высокий приоритет

**Стимул:** Пользователь нажимает на клавиши управления

**Реакция:** Персонаж начал движение

**Последовательность действий:** 1) Игрок нажал на клавишу управления персонажем; 2) ПИ начинает движение; 3) Игрок взаимодействует с окружающей средой

**Примечания к выполнению:** Персонаж двигается с одной скоростью в одно направление

**Название функции:** Смена направления движения

**Актер:** Пользователь

**Приоритет:** Имеет самый высокий приоритет

**Стимул:** Игрок нажал на кнопки для смены направления движения

**Реакция:** Персонаж начал движение в выбранном направлении

**Последовательность действий:** 1) Игрок нажал на кнопки смены направления движения; 2) Персонаж сменил направление движения

**Примечания к выполнению:** Персонаж двигается с одной скоростью в выбранное направление, поднимаясь вверх по платформам

**Название функции:** ПИ умирает

**Актер:** ПИ

**Приоритет:** Имеет самый высокий приоритет

**Стимул:** ПИ упал с платформы вне игровую зону

**Реакция:** ПИ умер

**Последовательность действий:** 1) Игрок упал с платформы вне игровую зону; 2) Срабатывание проверки на смерть; 3) Игрок проиграл

**Примечания к выполнению:** Здесь ничего не предусмотрено

**Название функции:** Проверка на смерть

**Актер:** Система/программа

**Приоритет:** Имеет высокий приоритет

**Стимул:** Падение игрока за пределы игровой зоны

**Реакция:** Проверка на смерть. ПИ умирает. Остановка игрового процесса.

**Последовательность действий:** 1) Проверка условий смерти; 2) ПИ умирает; 3) Остановка игрового процесса

**Примечания к выполнению:** Определение границ игровой зоны

**Название функции:** Счетчик очков

**Актер:** Программа

**Приоритет:** Имеет средний приоритет

**Стимул:** Игрок получает очки

**Реакция:** Прибавляется 50 очков

**Последовательность действий:** *ПИ* выполняет прыжки на платформы, чем выше он поднимается, тем больше очков игрок получает

**Примечания к выполнению:** Прибавляемое к-во очков зависит от того, насколько игрок будет прыгать вверх по платформам. Идёт прибавление к предыдущему значению. Если игрок закончил играть с совпадающим рекордным к-вом очков, то рекорд остаётся тем же

**Название функции:** Изменение рекордного к-во очков

**Актер:** Программа

**Приоритет:** Имеет средний приоритет

**Стимул:** Игрок завершает уровень с новым к-вом очков

**Реакция:** Изменяет рекордное к-во очков

**Последовательность действий:** 1) Игрок проиграл; 2) Проверка на рекорд; 3) Считывание текущего к-ва очков

**Примечания к выполнению:** Рекордное количество очков

изменяется только если новое значение предыдущего.

**3.3. Поражение**

**Название функции:** Поражение

**Актер:** Игрок

**Приоритет:** Имеет высокий приоритет

**Стимул:** Игрок умирает

**Реакция:** Игрок может начать рестарт игры, нажав клавишу R, или вернуться в главное меню

**Последовательность действий:** 1) Игрок падает вне игровую зону; 2) Игрок проиграл

**Примечания к выполнению:** Через некоторое время после смерти появляется главное меню или нажмите клавишу R для быстрого рестарта

**3.4. Быстрый рестарт игры**

**Название функции:** Рестарт игрового процесса

**Актер:** Пользователь

**Приоритет:** Имеет низкий приоритет

**Стимул:** Игровой процесс начинается быстрее с клавиши R

**Реакция:** Запускается игровой процесс быстрее

**Последовательность действий:** 1) Пользователь нажал на клавишу R после поражения; 2) Игровой процесс начался

**Примечания к выполнению:** Через некоторое время после смерти появляется главное меню или нажмите клавишу R для быстрого рестарта

**3.5. Остальные функции**

**Название функции:** Появление платформ

**Актер:** Программа/система

**Приоритет:** Имеет высокий приоритет

**Стимул:** Игра запускается

**Реакция:** На игровом поле появляются платформы

**Последовательность действий:** 1) Генерация платформ

**Примечания к выполнению:** Платформы должны быть в любом случа

**4. Требования к внешнему интерфейсу**

4.1. Пользовательский интерфейс

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **Элемент интерфейса** | | |  | | --- | | Функция | |
| **ГЛАВНОЕ МЕНЮ** | |
| |  | | --- | | Кнопка “Play” | | |  | | --- | | Запускает игровой процесс | |
| |  | | --- | | Кнопка “Quit” | | |  | | --- | | Закрывает приложение | |
| **ЭКРАН УРОВНЯ** | |
| Кнопка “Back to main menu” | Выходит в главное меню, завершая игровой процесс |
| Счётчик очков | Показывает сколько очков заработал игрок |

**4.2. Аппаратный интерфейс**

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **Характеристики** | | **Компьютер** |
| Логические | Отображение экрана |
| Физические | 1) Клавиатура/мышь  2) Монитор |

**4.3. Программный интерфейс**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Компонент** | | |  | | --- | | Связь | | Цель |
| |  | | --- | | Инструмент разработки | | PyCharm Community, в котором можно писать код на Python | Обеспечивает  средствами для  разработки продукта. |
| |  | | --- | | Язык программирования | | |  | | --- | | Python | | С его помощью реализуется вся игровая логика. |
| |  | | --- | | Операционные системы | | |  | | --- | | Windows, Linux, | | Обеспечивает разнообразие платформ, на которые можно выпустить проект. |
| |  | | --- | | Пользовательский интерфейс | | Библиотеки tkinter, customtkinter. | Обеспечивает пользователя взаимодействием с игрой. |
| Музыкальные ресурсы | Звуки/музыка  используемая в игре. | Интегрирование музыки и звуков в проект. |

1. **Другие нефункциональные требования**

**5.1. Требования к производительности**

**5.1.1. Стабильность работы**

Игра должна работать без сбоев, вылетов, зависаний.

**5.1.2. Быстрота загрузки экранов**

Загрузка не должна занимать большое количество времени(не >5c).

**5.1.3. Обработка музыки**

Музыка не должна искажаться при воспроизведении.

**5.2. Требования безопасности**

**5.2.1. Соглашение о конфиденциальности**

Игра должна иметь соглашение о конфиденциальности, которое предусматривает сбор и передачу личной информации с целью аналитики и маркетинга.

**5.2.2. Сбор данных**

Игра должна собирать данные об игровом процессе игрока, при этом

обеспечить полную конфиденциальность и предотвратить утечки данных.

**5.3. Атрибуты качества программного обеспечения**

**5.3.1. Расширяемость**

Игра должна иметь возможность к расширению, добавлению новых

механик без каких-либо проблем.

**5.3.2. Совместимость**

Игра должны быть совместима с различными версиями операционных систем PC устройств(например Windows 10-11).

**5.3.3. Системные требования**

Игра должна иметь минимальные системные требования, такие чтобы её могло запустить любое устройство.

**5.3.4. Интерфейс**

Интерфейс должен быть интуитивно понятным, чтобы пользователь не задумывался куда надо нажимать.

**5.4. Бизнес правила**

**5.4.1. Сбор и аналитика данных**

Игра должна анализировать данные игрового процессы для выявления сложных мест геймплея, ошибок во время игры, среднее время использования приложения и т.п. для улучшения качества геймплея.

**5.4.2. Маркетинг**

Для продвижения игры стоит распространять рекламу на разных частях экрана, но так, чтобы игроку это не мешало.

**5.4.3. Обратная связь с сообществом**

Должно быть активное взаимодействие с игроками через различные форумы (или другие средства связи) для выявления скучных, сложных мест геймплея или выявления ошибок.

**6. Приложения**

**6.1. Приложение A: Словарь**

***Игровое поле***: состоит из платформ по которым прыгает ПИ.

**Платформа неподвижная:** статический, неподвижный обьект. Из платформ состоит места, по которым игрок может прыгать.

**Платформа подвижная:** динамический, подвижный обьект. Из платформ состоит места, по которым игрок может прыгать.

**Стимул -** это внешнее событие, действие или воздействие, которое

инициирует или вызывает реакцию в системе или устройстве. В данном контексте стимул - это то, что пользователь делает на сенсорном экране или в игре, чтобы взаимодействовать с ней. Например, пользователь касается и перемещает палец по сенсорному экрану - это стимул.

**Реакция -** это ответ или действие, которое выполняется системой или устройством в ответ на стимул. В данном контексте реакция - это то, что система делает в ответ на действия пользователя или другие события. Например, система регистрирует движение пальца и преобразует его в движение персонажа - это реакция.

**UI (User Interface)** - Интерфейс пользователя, обеспечивающий взаимодействие пользователя с ПО: элементы управления, графические элементы.

**6.2. Приложение B: Диаграмма**

**6.2.2. Диаграмма классов**

