

УТВЕРЖДАЮ

Руководитель практики от Университета

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(должность, ФИО)*

МП \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(Подпись)*

Дата « \_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 202\_ год

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

ПМ.05 Проектирование и разработка информационных систем

|  |  |
| --- | --- |
| Коршунов Даниэль Александрович | |
| *(Ф.И.О. обучающегося)* | |
| 09.02.07 Информационные системы и программирование | |
| *(специальность)* | |
|  | |
| Учебная группа | ИСПк-202-52-00 |
|  |  |
| Место прохождения практики | ФГБОУ ВО «Вятский государственный университет», |
| Колледж ВятГУ | *(наименование организации, структурного подразделения организации)* |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Итоговая оценка: |  | | | | |
| Руководитель  практики от университета |  |  |  |  |  |
|  | *(дата)* |  | *(подпись)* |  | *(Ф.И.О.)* |

Киров, 2023 г.



ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ф.И.О. обучающегося | | | Коршунов Даниэль Александрович | | | | | | |
| Специальность | | | 09.02.07 Информационные системы и программирование | | | | | | |
| Учебная группа | | | ИСПк-202-52-00 | | | | | | |
| Вид практики | | | учебная практика | | | | | | |
| Сроки прохождения практики с | | | | 11.01.2023 | по | 16.06.2023 | | | |
| Место прохождения практики | | | | ФГБОУ ВО «Вятский государственный университет»,  Колледж ВятГУ | | | | | |
|  | | (наименование организации, структурного подразделения организации) | | | | | | |
| № | Виды работ, выполняемых обучающимися во время практики | | | | | | Объем работ (час) | Формируемые компетенции | | |
| 1 | Пройти инструктаж по ознакомлению с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте | | | | | | 2 | - | | |
| 2 | Постановка задачи | | | | | | 8 | ОК2, ОК6, ОК7, ОК9 | | |
| 3 | Настройка рабочего окружения | | | | | | 8 | ОК1, ОК8, ОК9 | | |
| 4 | Анализ предметной области, постановка расширенного технического задания | | | | | | 10 | ПК5.1, ОК3, ОК4 | | |
| 5 | Проектирование | | | | | | 35 | ПК5.2, ПК5.3, | | |
| 6 | Реализация программного кода | | | | | | 35 | ПК5.4, ПК5.8, | | |
| 7 | Тестирование и отладка полученного кода | | | | | | 22 | ПК5.5, ПК6.4, ОК2, ОК9 | | |
| 8 | Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов | | | | | | 10 | ПК5.6, ПК5.7, ОК9, ОК10, ОК11 | | |
| 9 | Подготовка к сдаче и сдача промежуточной аттестации | | | | | | 2 | ОК4, ОК5 | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Индивидуальное задание на практику разработано в соответствии с рабочей программой практики. | | | | | |
| Руководитель  практики от университета |  |  |  |  |  |
|  | (дата) |  | (подпись) |  | (Ф.И.О.) |

|  |  |
| --- | --- |
| С индивидуальным заданием ознакомлен(а) |  |
|  | (дата, подпись обучающегося) |

**ОТЗЫВ РУКОВОДИТЕЛЯ ПРАКТИКИ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ф.И.О. обучающегося | | Коршунов Даниэль Александрович | | | |
| Специальность | | 09.02.07 Информационные системы и программирование | | | |
| Учебная группа | | ИСПк-202-52-00 | | | |
| Вид практики | | учебная практика | | | |
| Сроки прохождения практики с | | | 16.01.2023 | по | 16.06.2023 |
| Место прохождения практики | | | ФГБОУ ВО «Вятский государственный университет», Колледж ВятГУ | | |
|  | (наименование организации, структурного подразделения организации) | | | | |

ВИДЫ И КАЧЕСТВО ВЫПОЛНЕНИЯ РАБОТ

|  |  |  |  |
| --- | --- | --- | --- |
| Вид работ | Критерий выполнения работ | | |
| Выполнены полностью самостоятельно | Выполнены с незначительной помощью наставника | Выполнены с помощью наставника |
| Постановка задачи | V |  |  |
| Настройка рабочего окружения | V |  |  |
| Анализ предметной области, постановка расширенного технического задания | V |  |  |
| Проектирование | V |  |  |
| Реализация программного кода | V |  |  |
| Тестирование и отладка полученного кода | V |  |  |
| Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов | V |  |  |
| Подготовка к сдаче и сдача промежуточной аттестации | V |  |  |

Обучающийся ознакомлен с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также прошел вводный инструктаж и инструктаж на рабочем месте.

Во время прохождения учебной практики обучающимся освоены следующие профессиональные и общие компетенции:

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование компетенции | Показатели оценки | Оценка | |
| Освоена | Не освоена |
| ПК 5.1. Собирать исходные данные для разработки проектной документации на информационную систему. | Способен выполнять анализ предметной области, выявляя существенные элементы, оказывающие влияние на проектируемую систему | V |  |
| ПК 5.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика. | Способен выполнять декомпозицию сущностей с целью получения наиболее полной картины о целесообразной структуре разработки | V |  |
| ПК 5.3. Разрабатывать подсистемы безопасной информационной системы в соответствии с техническим заданием. | Способен прогнозировать потенциально некорректные действия пользователя и предусматривать соответствующие реакции со стороны системы | V |  |
| ПК 5.4. Производить разработку модулей информационной системы в соответствии с техническим заданием. | Способен создавать программный код, отвечающий предъявляемым требованиям | V |  |
| ПК 5.5. Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы. | Способен выполнять оценку корректности функционирования системы | V |  |
| ПК 5.6. Разрабатывать техническую документацию на эксплуатацию информационной системы. | Способен разрабатывать проектную, техническую и пользовательскую документации | V |  |
| ПК 5.7. Производить оценку информационной системы для выявления возможности ее модернизации. | Способен оценивать перспективы дальнейшего развития программной системы | V |  |
| ПК 5.8. Осуществлять интеграцию информационной системы с другими программными продуктами. | Способен повторно использовать готовые шаблонные решения при разработке программного продукта | V |  |
| ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. | Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности | V |  |
| ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. | Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности | V |  |
| ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие. | Способен находить пути улучшения имеющихся решений, позволяющих повысить их общий качественный уровень | V |  |
| ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. | Способен эффективно использовать компьютерное время, а также материальные ресурсы, необходимые для решения поставленных задач | V |  |
| ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. | Способен соблюдать требования внутреннего трудового распорядка организации, охраны труда и техники безопасности в целях сохранения собственного здоровья | V |  |
| ОК 09. Использовать информационные технологии в профессиональной деятельности. | Способен применять стек современных средств разработки ПО для решения задач профессиональной деятельности в заданном контексте | V |  |
| ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке. | Способен использовать в своей работе специализированную документацию | V |  |
| ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере. | Способен проектировать алгоритмические решения, принимая во внимание имеющиеся ресурсные ограничения | V |  |

**Краткая характеристика работы обучающегося**

|  |
| --- |
| Программа практики выполнена в полном объеме. Все виды работ выполнялись в срок, |
| без существенных замечаний. В достаточной степени была проявлена самостоятельность |
| и умение грамотно пользоваться источниками информации, находящимися в свободном |
| доступе. |

|  |  |  |
| --- | --- | --- |
|  |  | Руководитель практики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Подпись ФИО  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность)  Дата «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_\_ год |

**СОДЕРЖАНИЕ**

**[ВВЕДЕНИЕ](#_Toc137027817)** [7](#_Toc137027817)

[**1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ** 8](#_Toc137027818)

[**2. НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ** 9](#_Toc137027819)

[**3. ФОРМУЛИРОВКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ** 11](#_Toc137027820)

[**4. ОПИСАНИЕ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ** 12](#_Toc137027821)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 35](#_Toc137027822)

[**ПРИЛОЖЕНИЕ А** 36](#_Toc137027823)

[**ПРИЛОЖЕНИЕ Б** 48](#_Toc137027824)

[**ПРИЛОЖЕНИЕ В** 49](#_Toc137027825)

[**ПРИЛОЖЕНИЕ Г** 50](#_Toc137027826)

# **ВВЕДЕНИЕ**

Учебная практика ПМ.05 проходила на базе Колледжа ФГБОУ ВО «Вятский государственный университет» в период с 11.01.2023 г. по 16.06.2023 г. по субботам.

Цель практики: сформировать у обучающихся представление об алгоритмах обработки данных, а также современных библиотечных реализациях алгоритмов и структур данных и их использовании при решении прикладных задач.

Задачи практики:

– дать представление о различных структурах данных;

– показать различные методы решения алгоритмических задач;

– дать представление о сложных алгоритмах, используемых при решении алгоритмических задач.

# **1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ**

В период с 11.01.2023 по 16.06.2023 при прохождении учебной практики ПМ.05 на базе Колледжа ФГОБУ ВО «Вятский государственный университет» был выполнен следующий перечень работ, представленный в таблице 1.

Таблица 1 – Сведения о работе, выполненной в период практики

| Дата | Краткое содержание выполненных работ |
| --- | --- |
| 11.01.2023 | Ознакомление с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте |
| 12.01.2023-25.01.2023 | Постановка задачи |
| 26.01.2023-08.02.2023 | Настройка рабочего окружения |
| 09.02.2023-22.02.2023 | Анализ предметной области, постановка расширенного технического задания |
| 23.02.2023-29.03.2023 | Проектирование |
| 30.03.2023-03.05.2023 | Реализация программного кода |
| 04.05.2023-24.05.2023 | Тестирование и отладка полученного кода |
| 25.05.2023-13.06.2023 | Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов |
| 14.06.2023-16.06.2023 | Подготовка к сдаче и сдача промежуточной аттестации |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата) (подпись)

# **2. НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ**

В ходе прохождения учебной практики использовался язык программирования Python, среда разработки (IDE) PyCharm.

1) Python ­— это высокоуровневый язык программирования с открытым исходным кодом. Он легко читается и пишется, имеет множество встроенных библиотек и дополнительных библиотек доступных через PyPI. Однако, он имеет низкую производительность и может выдавать неожиданные ошибки из-за динамической типизации.

PyCharm — это интегрированная среда разработки (IDE) для Python. Она предлагает множество инструментов, таких как автоматическое завершение кода, отладку, инспектирование, анализ кода и интеграцию с VCS. Это позволяет ускорить разработку и улучшить качество кода.

2) Для разработки игры была взята библиотека PyQt5, которая используется для объектно-ориентированного программирования. Также используется библиотека random для случайной генерации чисел.

Скриншоты работы в выбранной среде и библиотеках приведены на рисунках 2.1 и 2.2.

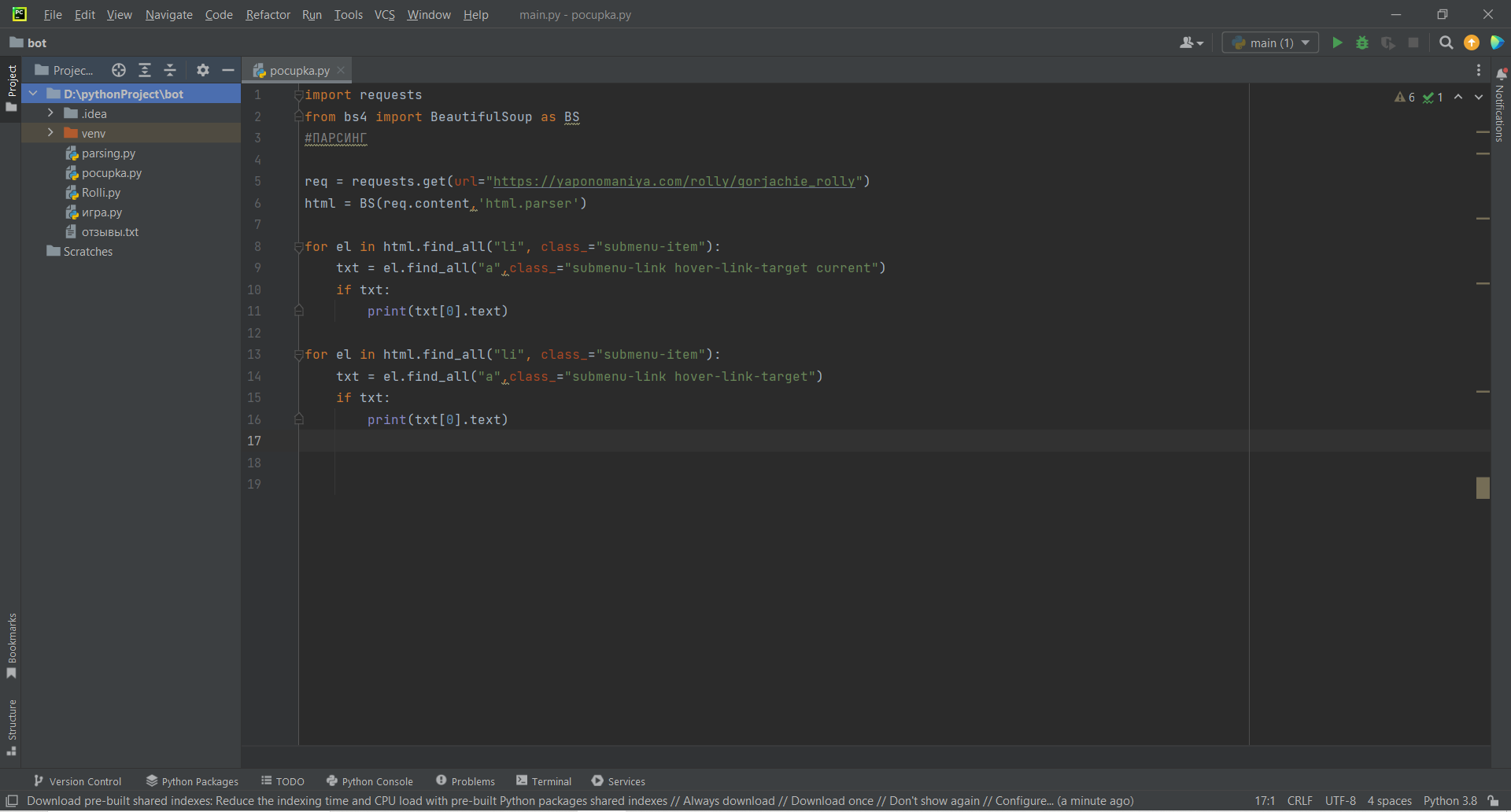


Рисунок 2.1 – PyCharm

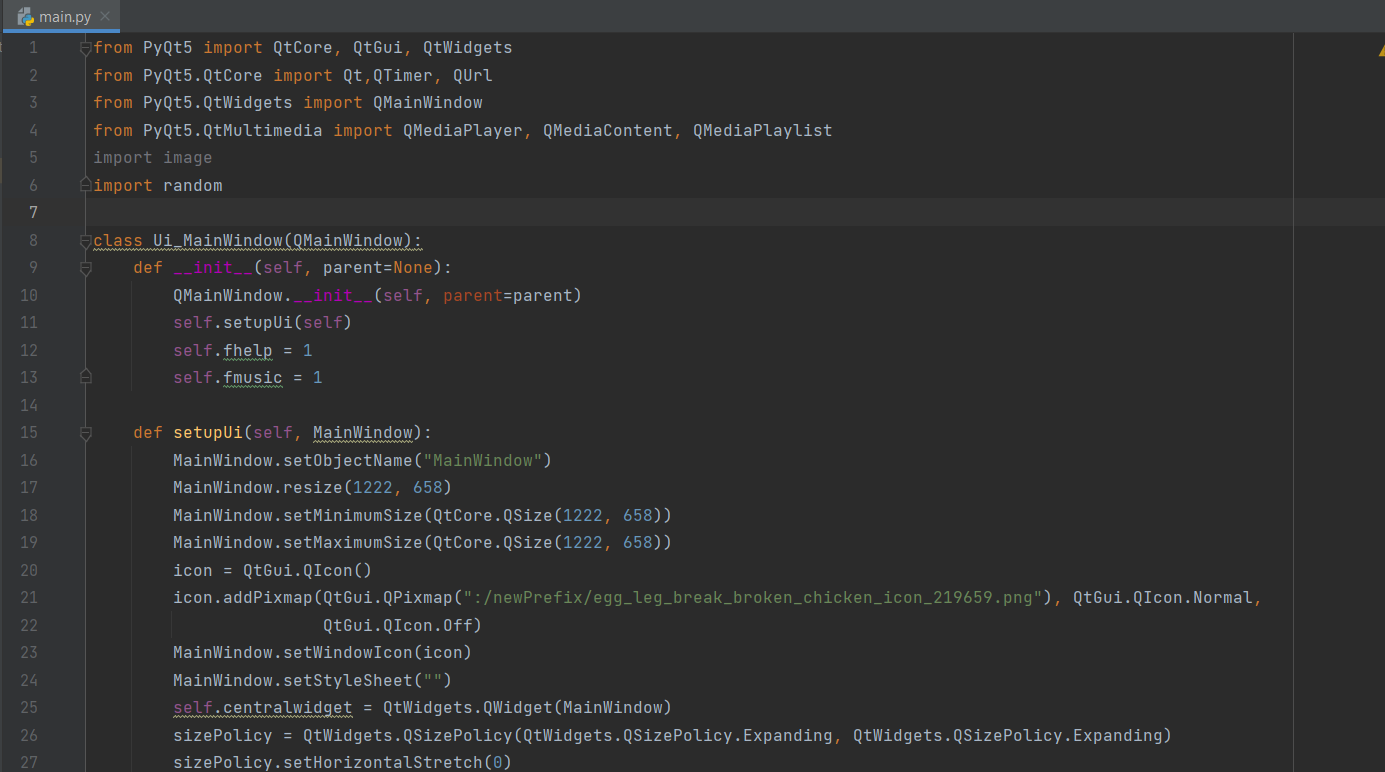


Рисунок 2.2 – Работа с библиотекой PyQt5

Для системы контроля версий в соответствии с заданием использована система GitHub.

Ссылка на аккаунт: [https://github.com/kkordan](https://github.com/kkordan%20) (см. Рисунок 2.3)

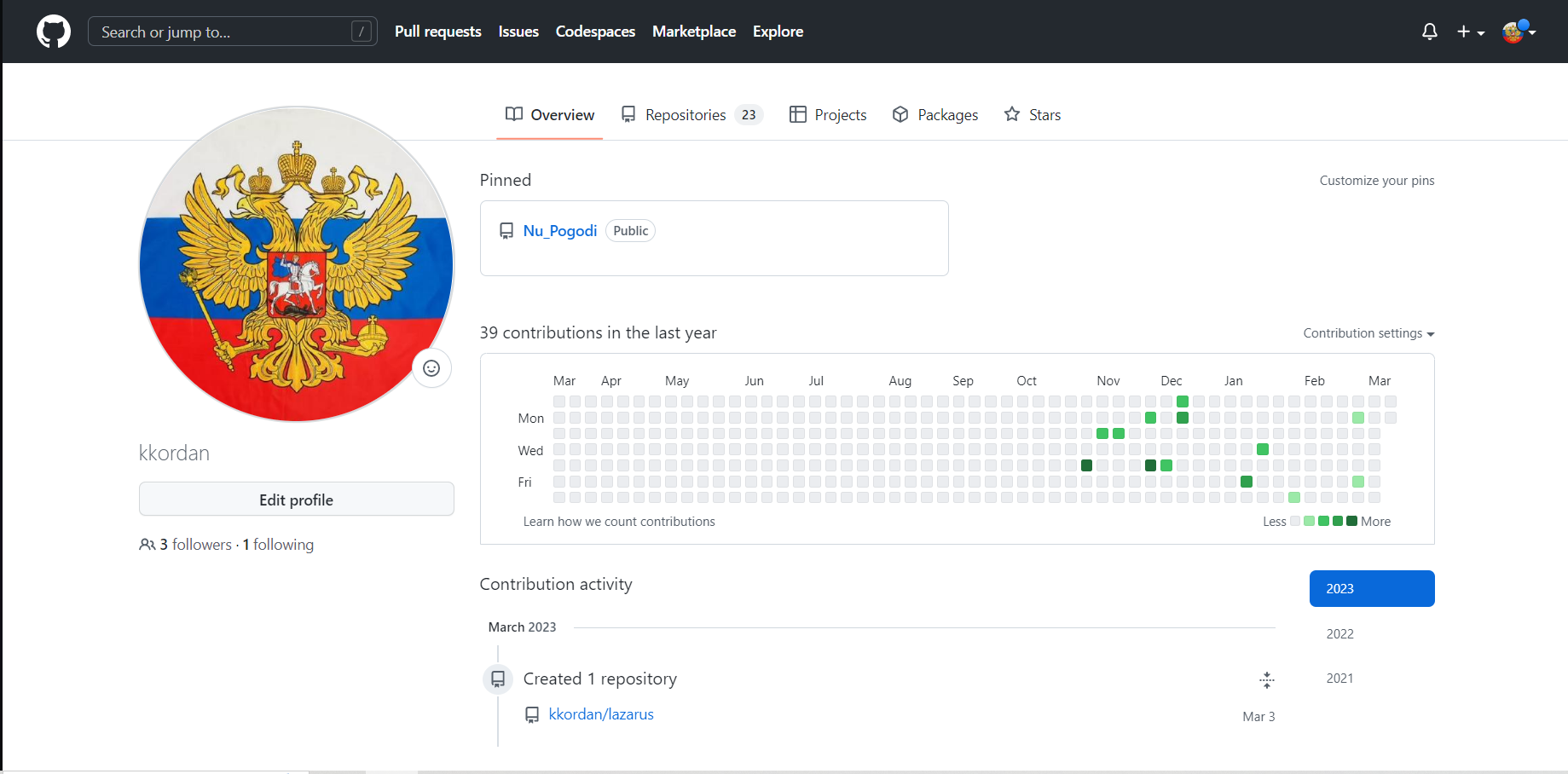


Рисунок 2.3 ⎯ аккаунт на Github

# **3. ФОРМУЛИРОВКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

В соответствии с индивидуальным заданием на учебную практику необходимо:

* Разработать игру «Ну, Погоди!»;
* Подготовить всю необходимую документацию для разработанной игры;

# **4. ОПИСАНИЕ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

**4.1 Анализ предметной области и обзор аналогов**

**4.1.1 Анализ предметной области**

Игры — это развлекательная деятельность, которая включает в себя участие в некоторой форме активности с целью развлечения, удовлетворения любопытства, общения с другими людьми или развития навыков.

Игры могут быть физическими (например, спортивные игры), настольными (например, шахматы, монополия), компьютерными или мобильными (например, видеоигры), ролевыми, карточными (например, покер) и многими другими.

В играх обычно есть определенные правила и цели, которые нужно достигнуть, чтобы победить. Игроки обычно действуют в конкуренции друг с другом, хотя в некоторых играх может быть и кооперативный режим. В любом случае, игры часто позволяют игрокам учиться, развиваться и получать удовольствие в процессе.

Игры могут быть как соревновательными, так и развивающими. Некоторые игры помогают учиться новым навыкам, например, улучшению математических навыков, расширению словарного запаса или улучшению памяти. Другие игры просто служат для развлечения и отдыха.

Игры являются частью культуры любой нации и могут иметь важное значение для развития личности, социальной адаптации и взаимоотношений между людьми.

Игра «Ну, Погоди!» заключалась в том, что волк должен был успеть поймать яйца, которые катятся из курятников, поэтому в народе игру прозвали «Яйцеловка».

«Ну, погоди!» можно считать своего рода первой игровой приставкой. Она является аналогом игры EG-26 Egg, выпущенной компанией Nintendo, о которой в СССР никто ничего не слышал.

Проблема предметной области:

Многие считают, что компьютерные игры используются только для развлечения и никакой познавательной или образовательной информации не несут. Но нужно понимать, что игры могут быть полезными не только для детей, но и взрослых. С помощью игр можно развивать стратегические навыки, моторику рук и ловкость.

**4.1.2 Обзор аналогов**

Первой игрой-аналогом, взятой для анализа, стала электронная игра "Ну, погоди!" («Электроника ИМ-02») (Рисунок 4.1.1).

Четыре маленьких птенца сидели на своих гнездах и несли яйца, которые постепенно скатывались по лоткам. В игре, где вы играете за Волка из мультфильма «Ну, погоди! », вам необходимо поймать как можно больше яиц, которые могут упасть в один из четырех лотков. Каждое пойманное яйцо приносит вам одно очко, но если оно упадет на землю, вы получите штрафное очко, обозначенное изображением цыпленка, вылупляющегося из яйца. Если вы пропустите яйцо в присутствии кролика, высовывающегося из окна домика в левом углу, из разбитого яйца вылезет птенец, а вы получите половину штрафного очка. Если вы соберете 200 и 500 яиц, штрафные очки будут аннулированы.

Если вы получите три штрафных очка (от трех до шести выпавших яиц), игра завершится. Когда вы наберете 999 очков, игра продолжится с 0, но штрафные очки не будут сброшены, а скорость игры немного замедлится. Затем скорость игры снова начнет увеличиваться. Ваше наивысшее достижение в 999 баллов будет отображаться в таблице рекордов.

К достоинствам игры можно отнести:

* Портативность игры
* Простота управления

Недостатки игры:

* Старая игра
* Устаревший дизайн
* Тяжело найти подобное устройство с игрой в продаже



Рисунок 4.1.1 – Скриншот «Ну, Погоди!»

Следующей для анализа игрой была выбрана онлайн играВолк Ловит Яйца (Рисунок 4.1.2).

Это полная копия игры «Ну, Погоди!», которая реализована на сайте и в которую можно поиграть с только с интернетом. Здесь реализовано все то же самое что и в оригинальной игре, но есть возможность играть на компьютере или смартфоне с доступом к интернету. Так же есть возможность развернуть игру в полноэкранный режим.

Достоинства игры:

* Современная реализация игры в интернете, которая позволяет запустить игру на компьютерах и смартфонах с низкой производительностю
* Простота управления

Недостатки игры:

* Игра доступна только онлайн.

Рисунок 4.1.2 –Скриншот из «Волк Ловит Яйца»

Третьей игрой – аналогом является игра «Игра Волк Ловит Яйца: Ну погоди» (Рисунок 4.1.3).

Это онлайн игра с сайта в которой управление реализовано мышкой, что не удобно для игры. Также недостатком является интерфейс низкого качества. Игра очень похожа на те аналоги, которые были рассмотрены выше.

Достоинства игры:

* Современная реализация игры в интернете, которая позволяет запустить игру на компьютерах и смартфонах с низкой производительностю

Недостатки игры:

* Игра доступна только онлайн.
* Низкое качество графики.



Рисунок 4.1.3 – Скриншот «Игра Волк Ловит Яйца: Ну погоди»

Вывод:

Было рассмотрено три игр-аналогов и проанализированы достоинства и недостатки этих игр. И был выявлен ряд проблем. Основными проблемами рассмотренных аналогов является требование наличия интернета, или возможность игры только на специальных устройствах. Эти недостатки будут исправлены в новой игре, которая будет разработана в процессе учебной практики.

**4.2 Разработка алгоритмов функционирования и структуры программы**

Алгоритм работы игры “Ну, Погоди!” представлен на рисунке 4.2.1.

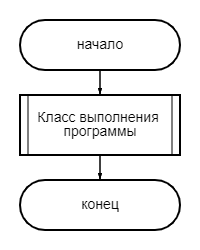


Рисунок 4.2.1 – Алгоритм работы игры “Ну, Погоди!”

Программа состоит из класса выполнения программы в котором прописаны все объекты и функции для выполнения программы. Алгоритм этого класса представлен на рисунке 4.2.2.

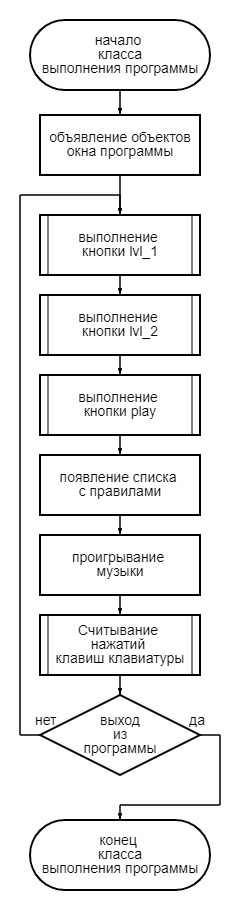


Рисунок 4.2.2 – Алгоритм класса выполнения программы

В классе происходит объявление и описание всех объектов программы и описание окна в котором выполняется программа. В классе циклично происходит обработка нажатий кнопок lvl\_1, lvl\_2, play, появление правил игры, проигрывание музыки и считывание нажатий клавиш клавиатуры. Каждая функция описана ниже.

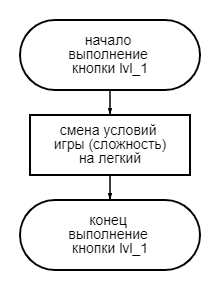


Рисунок 4.2.3 – Выполнение кнопки lvl\_1

При нажатии кнопки lvl\_1 происходит смена сложности игры на легкий.

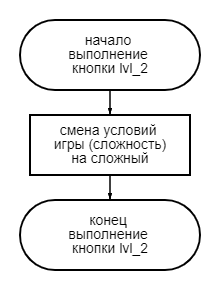


Рисунок 4.2.4 – Выполнение кнопки lvl\_2

При нажатии кнопки lvl\_2 происходит смена сложности игры на легкий.

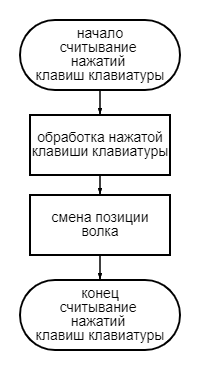


Рисунок 4.2.5 – Считывание нажатий клавиш клавиатуры

В этой функции происходит обработка нажатых клавиш клавиатуры и дальнейшее исполнение нажатых клавиш (смена позиции волка в игре).

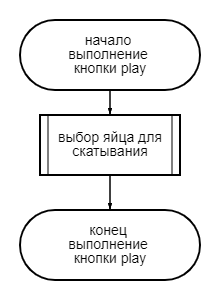


Рисунок 4.2.6 – Выполнение кнопки play

При нажатии кнопки play происходит запуск процесса игры в виде выполнения функции по выбору яйца для скатывания. Описание алгоритма выбора яйца изображено на рисунке 4.2.7.

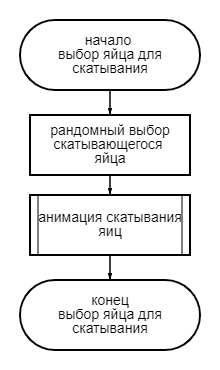


Рисунок 4.2.7 – Выбор яйца для скатывания

В этой функции случайно выбирается яйцо, которое будет катиться и последующая анимация этого катящегося яйца. Алгоритм анимации описан на рисунке 4.2.8.

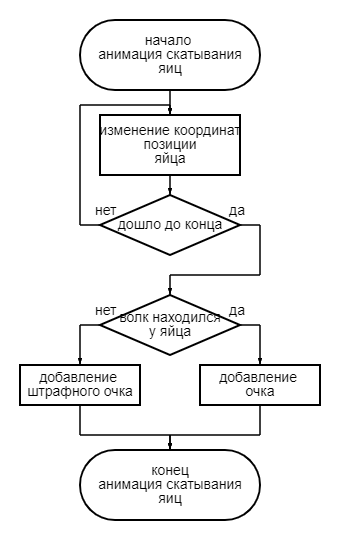


Рисунок 4.2.8 – Анимация скатывания яиц

Анимация происходит по следующему алгоритму: координаты позиции яйца изменяются, дальше идет проверка дошло ли яйцо до конца, если нет, то процесс изменения координат продолжается до тех пор пока яйцо не дойдет до конца, после происходит проверка поймал ли волк яйцо или нет и уже из этих условий происходит назначение очков игроку.



Рисунок 4.2.9 – Остановка игры

Остановка игры происходит следующим образом: останавливаются таймеры отвечающие за анимацию скатывания яиц и останавливается таймер анимации появления яиц.

Информационная структура игры приведена на рисунке 4.2.10.

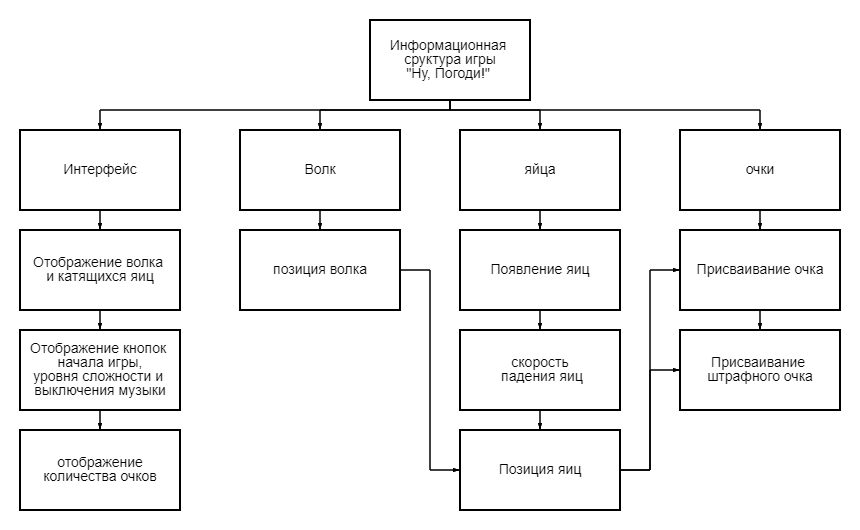


Рисунок 4.2.10 – Информационная структура приложения

Информационная структура игры «Ну, Погоди!» стостоит из:

- Интерфейса в котором есть отображение самого волка, яиц, кнопок (начало игры, уровня сложности и выключения музыки), отображение очков;

- Волка у которого определяется его позиция;

- Яиц, с их появлением, скоростью падения и позицией;

- Очков;

Вывод:

В процессе выполнения учебной практики была разработан алгоритм игры Ну, Погоди!», этот алгоритм разморен в разделе 4.3.

В данном разделе была рассмотрен и разработан алгоритм работы игры «Ну, Погоди!», и её информационная структура, которая включает в себя: интерфейс, волка, яйца, очки.

**4.3 Реализация программы. Разработка пользовательского интерфейса**

В данном разделе будут произведены разработки алгоритмов функционирования программы, разработан интерфейс программы, реализованы основные функции программы.

**4.3.1 Разработка программного обеспечения**

Программное обеспечение разработано в среде PyCharm на языке программирования Python.

Программное обеспечение состоит из одного класса Ui\_MainWindow, в котором описаны все объекты игры и все функции.

Исходный код программного обеспечения представлен в приложении А.

Функция setupUi(self, MainWindow) содержит код описания объектов игры, их местоположение, размер, имена, стили и т.д. Исходный код функции setupUi(self, MainWindow) представлен на рисунке 4.3.1.

def setupUi(self, MainWindow):  
 MainWindow.setObjectName("MainWindow")  
 MainWindow.resize(1222, 658)  
 MainWindow.setMinimumSize(QtCore.QSize(1222, 658))  
 MainWindow.setMaximumSize(QtCore.QSize(1222, 658))  
 icon = QtGui.QIcon()  
 icon.addPixmap(QtGui.QPixmap(":/newPrefix/egg\_leg\_break\_broken\_chicken\_icon\_219659.png"), QtGui.QIcon.Normal,  
 QtGui.QIcon.Off)  
 MainWindow.setWindowIcon(icon)  
 MainWindow.setStyleSheet("")  
 self.centralwidget = QtWidgets.QWidget(MainWindow)  
 sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Expanding)  
 sizePolicy.setHorizontalStretch(0)

Рисунок 4.3.1 – Исходный код функции setupUi

Функция closeWindow(self) это функция которая закрывает программу и состоит из метода quit. Исходный код функции closeWindow(self) представлен на рисунке 4.3.2.

def closeWindow(self):  
 QtCore.QCoreApplication.quit()

Рисунок 4.3.2 – Исходный код функции closeWindow

Функция play\_lvl1(self) используется для смены режима игры на легкий путем изменения переменной, отвечающей уровень игры, так же она останавливает запущенный процесс скатывания яиц. Исходный код функции play\_lvl1(self) представлен на рисунке 4.3.3.

def play\_lvl1(self):  
 self.labelgame1.setVisible(True)  
 self.labelgame2.setVisible(False)  
 self.game\_over = True  
 self.lv = True  
 self.timer.stop()  
 self.play.setText('Играть')  
 self.badshet = 3

Рисунок 4.3.3 – Исходный код функции play\_lvl1

Функция play\_lvl2(self) используется для смены режима игры на сложный путем изменения переменной, отвечающей уровень игры, так же она останавливает запущенный процесс скатывания яиц. Исходный код функции play\_lvl2(self) представлен на рисунке 4.3.4.

def play\_lvl2(self):  
 self.labelgame2.setVisible(True)  
 self.labelgame1.setVisible(False)  
 self.game\_over = True  
 self.lv = False  
 self.timer.stop()  
 self.play.setText('Играть')  
 self.badshet = 3

Рисунок 4.3.4 – Исходный код функции play\_lvl2

Функция play\_Game(self) используется для запуска игры. Функция запускает qtimer который отвечает за появление яиц, также в этой функции определяется скорость появления яиц. Исходный код функции play\_Game(self) представлен на рисунке 4.3.5.

def play\_Game(self):  
 if self.game\_over == True:  
 self.game\_over = False  
 self.play.setText('Закончить')  
 self.labelPomer.setVisible(False)  
 else:  
 self.game\_over = True  
 self.play.setText('Играть')  
  
 if ((self.game\_over == False) and (self.lv == True)):  
 self.timer.setInterval(2000)  
 self.timer.timeout.connect(self.egg\_time)  
 self.timer.start()

Рисунок 4.3.5 – Исходный код функции play\_Game

Функция vopros (self) используется для визуализации (с помощью метода setVisible) правил игры при нажатии на кнопку правил игры. Исходный код функции vopros(self) представлен на рисунке 4.3.6.

def vopros (self):  
 self.fhelp += 1  
 if self.fhelp % 2 == 0:  
 self.pravila.setVisible(True)  
 self.fhelp = 0  
 else:  
 self.pravila.setVisible(False)

Рисунок 4.3.6 – Исходный код функции vopros

Функция playAudio(self) используется для проигрывания музыки. Исходный код функции playAudio (self) представлен на рисунке 4.3.7.

def playAudio (self):

self.fmusic += 1

if self.fmusic % 2 == 0:

self.player.pause()

self.fmusic = 0

self.icon1.addPixmap(QtGui.QPixmap(":/newPrefix/mute.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

self.volume.setIcon(self.icon1)

else:

self.player.play()

self.icon1.addPixmap(QtGui.QPixmap(":/newPrefix/volume.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

self.volume.setIcon(self.icon1)

Рисунок 4.3.7 – Исходный код функции playAudio

Функция retranslateUi(self, MainWindow) используется для присваивания названий кнопок, надписей, спискам. Исходный код функции retranslateUi(self, MainWindow) представлен на рисунке 4.3.8.

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "Ну, Погоди!"))

self.lvl1.setText(\_translate("MainWindow", "1"))

self.lvl2.setText(\_translate("MainWindow", "2"))

self.labelLVL.setText(\_translate("MainWindow", "Уровень сложности"))

self.play.setText(\_translate("MainWindow", "Играть"))

\_\_sortingEnabled = self.pravila.isSortingEnabled()

self.pravila.setSortingEnabled(False)

item = self.pravila.item(0)

item.setText(\_translate("MainWindow", "Правила игры"))

Рисунок 4.3.8 – Исходный код функции retranslateUi

Функция keyPressEvent(self, e) используется для отслеживания нажатий клавиш клавиатуры и перемещает волка, также она меняет логические переменные, которые отвечают за условие присваивания очка. Исходный код функции keyPressEvent(self, e) представлен на рисунке 4.3.9.

def keyPressEvent(self, e):  
 if e.key() == Qt.Key\_Up:  
 self.VUL.setVisible(True)  
 self.VDL.setVisible(False)  
 self.VDR.setVisible(False)  
 self.VUR.setVisible(False)  
 self.EVUl = True  
 self.EVDL = False  
 self.EVDR = False  
 self.EVUR = False

Рисунок 4.3.9 – Исходный код функции keyPressEvent

Функция egg\_time(self) используется для отслеживания нажатий клавиш клавиатуры и перемещает волка, также она меняет логические переменные, которые отвечают за условие присваивания очка. Исходный код функции egg\_time(self) представлен на рисунке 4.3.10.

def egg\_time(self):

self.count.setProperty("intValue", self.chet)

e = random.randint(0,1000000)

if 0 <= e <= 250000:

self.egg\_spawn = 1

elif 250001 <= e <= 500000:

self.egg\_spawn = 2

elif 500001 <= e <= 750000:

self.egg\_spawn = 3

else:

self.egg\_spawn = 4

if self.chet <= 10:

self.timer1.setInterval(60)

self.timer2.setInterval(60)

self.timer3.setInterval(60)

self.timer4.setInterval(60)

Рисунок 4.3.10 – Исходный код функции egg\_time

Функция updateEggPosition1(self) используется для анимации движения яиц для всех четырех яиц своя функция (updateEggPosition1, updateEggPosition2, updateEggPosition3, updateEggPosition4), которые аналогичны только поменяны имена объектов (эти функции можно посмотреть в исходном коде – Приложение А). Исходный код функции updateEggPosition1(self) представлен на рисунке 4.3.11.

def updateEggPosition1(self):

self.count.setProperty("intValue", self.chet)

self.ELU1.move(self.egg1\_x, self.egg1\_y)

self.ELU1.setVisible(True)

self.egg1\_x += self.egg\_speed\_x

self.egg1\_y += self.egg\_speed\_y # (390;297)

if self.egg1\_x >= 390 and self.egg1\_y >= 297 and self.EVUl == True :

self.chet += 1

self.egg1\_x = 310

self.egg1\_y = 257

self.ELU1.setVisible(False)

self.timer1.stop()

self.et1 = False

Рисунок 4.3.11 – Исходный код функции updateEggPosition1

Функция End(self) используется для завершения процесса игры (прекращение скатывания яиц и обнуляется счет). Исходный код функции End(self) представлен на рисунке 4.3.12.

def End(self):

self.timer.stop()

self.labelPomer.setVisible(True)

self.game\_over = True

self.play.setText('Играть')

self.timer1.stop()

self.timer2.stop()

self.timer3.stop()

self.timer4.stop()

self.ELU1.setVisible(False)

self.ELD1.setVisible(False)

self.ERU1.setVisible(False)

self.ERD1.setVisible(False)

Рисунок 4.3.12 – Исходный код функции End

Диаграмма класса представлена на рисунке 4.3.13

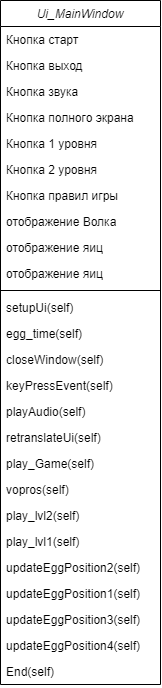


Рисунок 4.3.13 – Диаграмма класса

**4.3.2 Разработка пользовательского интерфейса**

Для разработки пользовательского интерфейса использовалась библиотека PyQt5 и приложение Qt Designer. Интерфейс состоит из одной экранной формы в которой происходят все действия игры и управления. Интерфейс игры изображен на рисунке 4.3.14



Рисунок 4.3.14 – Пользовательский интерфейс

На пользовательском интерфейсе расположен волк, скатывающиеся яйца, надпись уровня игры, надпись окончания игры. Для отображения этих объектов использовался label.

В интерфейсе с помощью push button сделаны следующие кнопки:

- кнопка «Играть» - запускает процесс скатывания яиц (запуск процесса игры);

- кнопки «1» и «2» - выбирают уровень сложности игры;

- кнопка «volum» (звук) – включает и выключает звук

- кнопка «Выход» ­- выходит из игры;

- кнопка «Правила» (знак вопроса) – выводит на экран краткие правила игры;

Также фон самой игры сделан с помощью label который расположен во весь экран.

Вывод: Сделан пользовательский интерфейс игры «Ну, Погоди!», который состоит из одной экранной формы в котором есть кнопки запуска и остановки игры, выхода из игры, кнопки выбора уровня игры, выключения звука и кнопка правил игры. В интерфейсе есть отображение волка, яиц, надписей и фона. В пункте 4.3 были описаны все функции исходного кода игры, составлена диаграмма класса игры и описан пользовательский интерфейс.

**4.4 Тестирование программы**

**Таблица 1 - Результаты испытаний**

| № | Шаг испытаний (проверок) | № пункта  Методики | Отметка о прохождении (да/нет) | Примечания |
| --- | --- | --- | --- | --- |
| 1 | Проверка состава и качества сопроводительной документации. | 1 | Да | — |
| 2 | Возможность запуска игры без использования интернета. | 2 | Да | — |
| 3 | Возможность запуска игры на любом ПК или ноутбуке, который доступен человеку. | 3 | Да | — |
| 4 | Анимация скатывания яиц с 4 насестов. | 4 | Да | — |
| 5 | Управление волком с помощью 4 клавиш, каждая клавиша направляет волка в сторону одного из 4 насестов, по которым катятся яйца. | 5 | Да | — |
| 6 | Наличие счетчика пойманных яиц волком. | 6 | Да | — |
| 7 | Ловля яиц волком, если он смотрит в направлении насеста, у которого яйцо докатилось до конца насеста. | 7 | Да | — |
| 8 | Выбор уровня сложности игры двумя кнопками (легкий, сложный). | 8 | Да | — |
| 9 | Автоматическое изменение скорости скатывания яиц. | 9 | Да | — |
| 10 | Настройка проигрывания музыки в игре с помощью кнопки, выключающей и включающей музыку | 10 | Да | — |
| 11 | Работа кнопки «Играть». | 11 | Да | — |
| 12 | Работа кнопки правил (знак вопроса). | 12 | Да | — |
| 13 | Работа кнопки «2» уровня игры. | 13 | Да | — |
| 14 | Работа кнопки «1» уровня игры. | 14 | Да | — |
| 15 | Работа кнопки «Выход» | 15 | Да | — |

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения учебной практики была написана полная документация игры «Ну, Погоди!» и разработана сама игра. Игра была протестирована и написана программа и методика испытаний. Так же во время учебной практики были получены навыки работы с библиотекой pyqt5 и программой Qt Designer. Игра написанная во время учебной практики является первой самостоятельно разработанной игрой разработчика.

Таким образом, программа учебной практики ПМ.05 в период с 11.01.2023 по 16.06.2023 была выполнена мной в полном объеме.

При выполнении работ, предусмотренных заданием на практику, в организации ФГБОУ ВО «Вятский государственный университет» мной, Коршуновым Даниэлем Александровичем, соблюдались правила внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологические правила и гигиенические нормативы. Пройден вводный инструктаж и инструктаж на рабочем месте. Замечаний не имел.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* Виджай, Б. Python GUI Programming Cookbook: Develop functional and responsive user interfaces with tkinter and PyQt5. Москва: ДМК Пресс, 2018. 408 с. ISBN 978-5-97060-657-7.
* Саммерфилд, М. Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming. Санкт-Петербург: БХВ-Петербург, 2016. 656 с. ISBN 978-5-9775-1697-7.
* Иванов, А. Б. Разработка игр с использованием библиотеки PyQt5. Игровая разработка и компьютерная графика. 2020, № 3, с. 45-50.
* The PyQt5 Documentation. Version 5.15.3. Дата доступа: 15 мая 2023. URL: <https://www.riverbankcomputing.com/static/Docs/PyQt5/>.
* Ламарр, М. Создание компьютерных игр с помощью Python и библиотеки PyQt5. Москва: Диалектика, 2019. 320 с. ISBN 978-5-907191-71-5.

# **ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД**

from PyQt5 import QtCore, QtGui, QtWidgets  
from PyQt5.QtCore import Qt,QTimer, QUrl  
from PyQt5.QtWidgets import QMainWindow  
from PyQt5.QtMultimedia import QMediaPlayer, QMediaContent, QMediaPlaylist  
import image  
import random  
  
class Ui\_MainWindow(QMainWindow):  
 def \_\_init\_\_(self, parent=None):  
 QMainWindow.\_\_init\_\_(self, parent=parent)  
 self.setupUi(self)  
 self.fhelp = 1  
 self.fmusic = 1  
  
 def setupUi(self, MainWindow):  
 MainWindow.setObjectName("MainWindow")  
 MainWindow.resize(1222, 658)  
 MainWindow.setMinimumSize(QtCore.QSize(1222, 658))  
 MainWindow.setMaximumSize(QtCore.QSize(1222, 658))  
 icon = QtGui.QIcon()  
 icon.addPixmap(QtGui.QPixmap(":/newPrefix/egg\_leg\_break\_broken\_chicken\_icon\_219659.png"), QtGui.QIcon.Normal,  
 QtGui.QIcon.Off)  
 MainWindow.setWindowIcon(icon)  
 MainWindow.setStyleSheet("")  
 self.centralwidget = QtWidgets.QWidget(MainWindow)  
 sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Expanding)  
 sizePolicy.setHorizontalStretch(0)  
 sizePolicy.setVerticalStretch(0)  
 sizePolicy.setHeightForWidth(self.centralwidget.sizePolicy().hasHeightForWidth())  
 self.centralwidget.setSizePolicy(sizePolicy)  
 self.centralwidget.setMinimumSize(QtCore.QSize(1222, 658))  
 self.centralwidget.setMaximumSize(QtCore.QSize(1222, 658))  
 self.centralwidget.setObjectName("centralwidget")  
 self.VUL = QtWidgets.QLabel(self.centralwidget)  
 self.VUL.setGeometry(QtCore.QRect(310, 260, 291, 271))  
 self.VUL.setText("")  
 self.VUL.setPixmap(QtGui.QPixmap(":/newPrefix/VUL.png"))  
 self.VUL.setObjectName("VUL")  
 self.VDL = QtWidgets.QLabel(self.centralwidget)  
 self.VDL.setGeometry(QtCore.QRect(300, 300, 291, 271))  
 self.VDL.setText("")  
 self.VDL.setPixmap(QtGui.QPixmap(":/newPrefix/VDL.png"))  
 self.VDL.setObjectName("VDL")  
 self.VUR = QtWidgets.QLabel(self.centralwidget)  
 self.VUR.setGeometry(QtCore.QRect(530, 260, 291, 271))  
 self.VUR.setText("")  
 self.VUR.setPixmap(QtGui.QPixmap(":/newPrefix/VUR.png"))  
 self.VUR.setObjectName("VUR")  
 self.VDR = QtWidgets.QLabel(self.centralwidget)  
 self.VDR.setGeometry(QtCore.QRect(530, 300, 291, 271))  
 self.VDR.setText("")  
 self.VDR.setPixmap(QtGui.QPixmap(":/newPrefix/VDR.png"))  
 self.VDR.setObjectName("VDR")  
 self.ELU1 = QtWidgets.QLabel(self.centralwidget)  
 self.ELU1.setGeometry(QtCore.QRect(310, 257, 41, 31))  
 self.ELU1.setText("")  
 self.ELU1.setPixmap(QtGui.QPixmap(":/newPrefix/EGG.png"))  
 self.ELU1.setObjectName("ELU1")  
 self.ELD1 = QtWidgets.QLabel(self.centralwidget)  
 self.ELD1.setGeometry(QtCore.QRect(310, 357, 41, 31))  
 self.ELD1.setText("")  
 self.ELD1.setPixmap(QtGui.QPixmap(":/newPrefix/EGG.png"))  
 self.ELD1.setObjectName("ELD1")  
 self.ERU1 = QtWidgets.QLabel(self.centralwidget)  
 self.ERU1.setGeometry(QtCore.QRect(860, 257, 41, 31))  
 self.ERU1.setText("")  
 self.ERU1.setPixmap(QtGui.QPixmap(":/newPrefix/EGG.png"))  
 self.ERU1.setObjectName("ERU1")  
 self.ERD1 = QtWidgets.QLabel(self.centralwidget)  
 self.ERD1.setGeometry(QtCore.QRect(860, 357, 41, 31))  
 self.ERD1.setText("")  
 self.ERD1.setPixmap(QtGui.QPixmap(":/newPrefix/EGG.png"))  
 self.ERD1.setObjectName("ERD1")  
 self.background = QtWidgets.QLabel(self.centralwidget)  
 self.background.setGeometry(QtCore.QRect(0, 0, 1222, 658))  
 self.background.setToolTipDuration(0)  
 self.background.setAutoFillBackground(False)  
 self.background.setStyleSheet("")  
 self.background.setText("")  
 self.background.setPixmap(QtGui.QPixmap(":/newPrefix/fon2.png"))  
 self.background.setObjectName("background")  
 self.count = QtWidgets.QLCDNumber(self.centralwidget)  
 self.count.setGeometry(QtCore.QRect(530, 170, 141, 41))  
 self.count.setLineWidth(0)  
 self.count.setMode(QtWidgets.QLCDNumber.Dec)  
 self.count.setSegmentStyle(QtWidgets.QLCDNumber.Flat)  
 self.count.setProperty("intValue", 0)  
 self.count.setObjectName("count")  
 self.lvl1 = QtWidgets.QPushButton(self.centralwidget)  
 self.lvl1.setGeometry(QtCore.QRect(1030, 110, 81, 71))  
 font = QtGui.QFont()  
 font.setPointSize(14)  
 font.setBold(True)  
 font.setWeight(75)  
 self.lvl1.setFont(font)  
 self.lvl1.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))  
 self.lvl1.setCheckable(False)  
 self.lvl1.setChecked(False)  
 self.lvl1.setObjectName("lvl1")  
 self.lvl2 = QtWidgets.QPushButton(self.centralwidget)  
 self.lvl2.setGeometry(QtCore.QRect(1120, 110, 81, 71))  
 font = QtGui.QFont()  
 font.setPointSize(14)  
 font.setBold(True)  
 font.setWeight(75)  
 self.lvl2.setFont(font)  
 self.lvl2.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))  
 self.lvl2.setCheckable(False)  
 self.lvl2.setChecked(False)  
 self.lvl2.setObjectName("lvl2")  
 self.labelLVL = QtWidgets.QLabel(self.centralwidget)  
 self.labelLVL.setGeometry(QtCore.QRect(1030, 70, 181, 31))  
 font = QtGui.QFont()  
 font.setFamily("Agency FB")  
 font.setPointSize(10)  
 font.setBold(True)  
 font.setItalic(False)  
 font.setUnderline(False)  
 font.setWeight(75)  
 font.setStrikeOut(False)  
 font.setKerning(True)  
 self.labelLVL.setFont(font)  
 self.labelLVL.setCursor(QtGui.QCursor(QtCore.Qt.ArrowCursor))  
 self.labelLVL.setStyleSheet("color: rgb(144, 29, 9);")  
 self.labelLVL.setAlignment(QtCore.Qt.AlignCenter)  
 self.labelLVL.setObjectName("labelLVL")  
 self.play = QtWidgets.QPushButton(self.centralwidget)  
 self.play.setGeometry(QtCore.QRect(410, 590, 411, 41))  
 font = QtGui.QFont()  
 font.setPointSize(18)  
 font.setBold(True)  
 font.setWeight(75)  
 self.play.setFont(font)  
 self.play.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))  
 self.play.setStyleSheet("color: rgb(149, 11, 11);")  
 self.play.setCheckable(False)  
 self.play.setChecked(False)  
 self.play.setObjectName("play")  
 self.volume = QtWidgets.QPushButton(self.centralwidget)  
 self.volume.setGeometry(QtCore.QRect(40, 210, 81, 71))  
 self.volume.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))  
 self.volume.setStyleSheet("")  
 self.volume.setText("")  
 self.icon1 = QtGui.QIcon()  
 self.icon1.addPixmap(QtGui.QPixmap(":/newPrefix/volume.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)  
 self.volume.setIcon(self.icon1)  
 self.volume.setIconSize(QtCore.QSize(40, 40))  
 self.volume.setCheckable(False)  
 self.volume.setChecked(False)  
 self.volume.setObjectName("volume")  
 self.pravila = QtWidgets.QListWidget(self.centralwidget)  
 self.pravila.setGeometry(QtCore.QRect(1020, 260, 201, 281))  
 self.pravila.setObjectName("pravila")  
 item = QtWidgets.QListWidgetItem()  
 font = QtGui.QFont()  
 font.setBold(True)  
 font.setWeight(75)  
 item.setFont(font)  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 item = QtWidgets.QListWidgetItem()  
 self.pravila.addItem(item)  
 self.ButPravila = QtWidgets.QPushButton(self.centralwidget)  
 self.ButPravila.setGeometry(QtCore.QRect(1090, 570, 71, 71))  
 self.ButPravila.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))  
 self.ButPravila.setStyleSheet("")  
 self.ButPravila.setText("")  
 icon2 = QtGui.QIcon()  
 icon2.addPixmap(QtGui.QPixmap(":/newPrefix/icons8-question-mark-100.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)  
 self.ButPravila.setIcon(icon2)  
 self.ButPravila.setIconSize(QtCore.QSize(40, 40))  
 self.ButPravila.setCheckable(False)  
 self.ButPravila.setChecked(False)  
 self.ButPravila.setObjectName("ButPravila")  
 self.urlicon = QtWidgets.QLabel(self.centralwidget)  
 self.urlicon.setGeometry(QtCore.QRect(10000, 480, 55, 16))  
 self.urlicon.setObjectName("urlicon")  
 self.labelgame = QtWidgets.QLabel(self.centralwidget)  
 self.labelgame.setGeometry(QtCore.QRect(330, 160, 55, 16))  
 font = QtGui.QFont()  
 font.setPointSize(10)  
 font.setBold(True)  
 font.setWeight(75)  
 self.labelgame.setFont(font)  
 self.labelgame.setObjectName("labelgame")  
 self.labelgame1 = QtWidgets.QLabel(self.centralwidget)  
 self.labelgame1.setGeometry(QtCore.QRect(380, 160, 16, 16))  
 font = QtGui.QFont()  
 font.setPointSize(10)  
 font.setBold(True)  
 font.setWeight(75)  
 self.labelgame1.setFont(font)  
 self.labelgame1.setObjectName("labelgame1")  
 self.labelgame2 = QtWidgets.QLabel(self.centralwidget)  
 self.labelgame2.setGeometry(QtCore.QRect(390, 160, 21, 16))  
 font = QtGui.QFont()  
 font.setPointSize(10)  
 font.setBold(True)  
 font.setWeight(75)  
 self.labelgame2.setFont(font)  
 self.labelgame2.setObjectName("labelgame2")  
 palette = QtGui.QPalette()  
 brush = QtGui.QBrush(QtGui.QColor(139, 17, 14))  
 brush.setStyle(QtCore.Qt.SolidPattern)  
 palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)  
 brush = QtGui.QBrush(QtGui.QColor(139, 17, 14))  
 brush.setStyle(QtCore.Qt.SolidPattern)  
 palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Base, brush)  
 brush = QtGui.QBrush(QtGui.QColor(139, 17, 14))  
 brush.setStyle(QtCore.Qt.SolidPattern)  
 palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Window, brush)  
 brush = QtGui.QBrush(QtGui.QColor(139, 17, 14))  
 brush.setStyle(QtCore.Qt.SolidPattern)  
 palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Button, brush)  
 brush = QtGui.QBrush(QtGui.QColor(139, 17, 14))  
 brush.setStyle(QtCore.Qt.SolidPattern)  
 palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Base, brush)  
 brush = QtGui.QBrush(QtGui.QColor(139, 17, 14))  
 brush.setStyle(QtCore.Qt.SolidPattern)  
 palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Window, brush)  
 brush = QtGui.QBrush(QtGui.QColor(139, 17, 14))  
 brush.setStyle(QtCore.Qt.SolidPattern)  
 palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Button, brush)  
 brush = QtGui.QBrush(QtGui.QColor(139, 17, 14))  
 brush.setStyle(QtCore.Qt.SolidPattern)  
 palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Base, brush)  
 brush = QtGui.QBrush(QtGui.QColor(139, 17, 14))  
 brush.setStyle(QtCore.Qt.SolidPattern)  
 palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Window, brush)  
 self.labelPomer = QtWidgets.QLabel(self.centralwidget)  
 self.labelPomer.setGeometry(QtCore.QRect(420, 210, 391, 71))  
 font = QtGui.QFont()  
 font.setPointSize(26)  
 font.setBold(True)  
 font.setWeight(75)  
 self.labelPomer.setFont(font)  
 self.labelPomer.setObjectName("label")  
 # self.butFullScreen = QtWidgets.QPushButton(self.centralwidget)  
 # self.butFullScreen.setGeometry(QtCore.QRect(40, 130, 81, 71))  
 # self.butFullScreen.setText("")  
 # icon3 = QtGui.QIcon()  
 # icon3.addPixmap(QtGui.QPixmap(":/newPrefix/full-screen.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)  
 # self.butFullScreen.setIcon(icon3)  
 # self.butFullScreen.setIconSize(QtCore.QSize(50, 50))  
 # self.butFullScreen.setObjectName("butFullScreen")  
 self.butVihod = QtWidgets.QPushButton(self.centralwidget)  
 self.butVihod.setGeometry(QtCore.QRect(40, 50, 81, 71))  
 font = QtGui.QFont()  
 font.setPointSize(9)  
 font.setBold(True)  
 font.setWeight(75)  
 self.butVihod.setFont(font)  
 self.butVihod.setStyleSheet("color: rgb(165, 55, 66);")  
 self.butVihod.setObjectName("butVihod")  
 self.background.raise\_()  
 self.VUL.raise\_()  
 self.VUR.raise\_()  
 self.VDR.raise\_()  
 self.ELU1.raise\_()  
 self.ELD1.raise\_()  
 self.ERU1.raise\_()  
 self.ERD1.raise\_()  
 self.count.raise\_()  
 self.lvl1.raise\_()  
 self.lvl2.raise\_()  
 self.labelLVL.raise\_()  
 self.play.raise\_()  
 self.volume.raise\_()  
 self.pravila.raise\_()  
 self.ButPravila.raise\_()  
 self.urlicon.raise\_()  
 self.labelgame.raise\_()  
 self.labelgame1.raise\_()  
 self.labelgame2.raise\_()  
 self.VDL.raise\_()  
 self.labelPomer.raise\_()  
 # self.butFullScreen.raise\_()  
 self.butVihod.raise\_()  
 MainWindow.setCentralWidget(self.centralwidget)  
  
 self.retranslateUi(MainWindow)  
 QtCore.QMetaObject.connectSlotsByName(MainWindow)  
  
 self.egg1\_x = 310  
 self.egg1\_y = 257  
  
 self.egg2\_x = 310  
 self.egg2\_y = 357  
  
 self.egg3\_x = 860  
 self.egg3\_y = 257  
  
 self.egg4\_x = 860  
 self.egg4\_y = 357  
  
 self.egg\_speed\_x = 2  
 self.egg\_speed\_y = 1  
  
 self.VDR.grabKeyboard() #перевод фокуса на волка для стрелок  
  
 # Визуализация виджетов  
 self.EVUl = False  
 self.EVDL = True  
 self.EVDR = False  
 self.EVUR = False  
 self.VDR.setVisible(False)  
 self.VUR.setVisible(False)  
 self.VUL.setVisible(False)  
 self.ELD1.setVisible(False)  
 self.ELU1.setVisible(False)  
 self.ERD1.setVisible(False)  
 self.ERU1.setVisible(False)  
 self.pravila.setVisible(False)  
 self.labelPomer.setVisible(False)  
  
 #Таймеры кнопок  
 self.et1 = False  
 self.et2 = False  
 self.et3 = False  
 self.et4 = False  
  
 self.labelgame1.setVisible(True)  
 self.labelgame2.setVisible(False)  
 self.game\_over = True  
 self.lv = True  
  
 # Исполение нажатий кнопок  
 self.ButPravila.clicked.connect(self.vopros)  
 self.play.clicked.connect(self.play\_Game)  
 self.lvl1.clicked.connect(self.play\_lvl1)  
 self.lvl2.clicked.connect(self.play\_lvl2)  
 self.volume.clicked.connect(self.playAudio)  
 self.butVihod.clicked.connect(self.closeWindow)  
  
 #Музыка  
 self.playlist = QMediaPlaylist()  
 self.playlist.addMedia(QMediaContent(QUrl.fromLocalFile("music\Atomic Heart OST — Trava u Doma (Geoffrey Day Remix, Земляне) (www.lightaudio.ru).mp3")))  
 self.playlist.addMedia(QMediaContent(QUrl.fromLocalFile("music\Mujuice\_Pesnyary\_Atomic\_Heart\_-\_Kosil\_Jas\_Konjushinu\_Mujuice\_Acid\_Rework\_75479749.mp3")))  
 self.playlist.addMedia(QMediaContent(QUrl.fromLocalFile("music\DVRST, Игорь Скляр, Atomic Heart — Комарово(Phonk Remix) (www.lightaudio.ru).mp3")))  
 self.playlist.setPlaybackMode(QMediaPlaylist.Loop)# устанавливаем режим воспроизведения на циклический  
 self.player = QMediaPlayer()  
 self.player.setPlaylist(self.playlist)  
 self.player.play()  
  
 #Объявление таймеров  
 self.timer1 = QTimer(self)  
 self.timer1.setInterval(60)  
 self.timer1.timeout.connect(self.updateEggPosition1)  
  
 self.timer2 = QTimer(self)  
 self.timer2.setInterval(60)  
 self.timer2.timeout.connect(self.updateEggPosition2)  
  
 self.timer3 = QTimer(self)  
 self.timer3.setInterval(60)  
 self.timer3.timeout.connect(self.updateEggPosition3)  
  
 self.timer4 = QTimer(self)  
 self.timer4.setInterval(60)  
 self.timer4.timeout.connect(self.updateEggPosition4)  
  
 self.timer = QTimer(self)  
 self.timer.setInterval(2000)  
 self.timer.timeout.connect(self.egg\_time)  
  
  
 def closeWindow(self):  
 QtCore.QCoreApplication.quit()  
  
 def play\_lvl1(self):  
 self.labelgame1.setVisible(True)  
 self.labelgame2.setVisible(False)  
 self.game\_over = True  
 self.lv = True  
 self.timer.stop()  
 self.play.setText('Играть')  
 self.badshet = 3  
  
 def play\_lvl2(self):  
 self.labelgame2.setVisible(True)  
 self.labelgame1.setVisible(False)  
 self.game\_over = True  
 self.lv = False  
 self.timer.stop()  
 self.play.setText('Играть')  
 self.badshet = 3  
  
 def play\_Game(self):  
 if self.game\_over == True:  
 self.game\_over = False  
 self.play.setText('Закончить')  
 self.labelPomer.setVisible(False)  
 else:  
 self.game\_over = True  
 self.play.setText('Играть')  
  
 if ((self.game\_over == False) and (self.lv == True)):  
 self.timer.setInterval(2000)  
 self.timer.start()  
 self.chet = 0  
 self.badshet = 0  
 elif ((self.game\_over == False) and (self.lv == False)):  
 self.timer.setInterval(1000)  
 self.timer.start()  
 self.chet = 0  
 self.badshet = 0  
 elif self.game\_over == True:  
 self.badshet = 3  
  
 def vopros (self):  
 self.fhelp += 1  
 if self.fhelp % 2 == 0:  
 self.pravila.setVisible(True)  
 self.fhelp = 0  
 else:  
 self.pravila.setVisible(False)  
  
 def playAudio (self):  
 self.fmusic += 1  
 if self.fmusic % 2 == 0:  
 self.player.pause()  
 self.fmusic = 0  
 self.icon1.addPixmap(QtGui.QPixmap(":/newPrefix/mute.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)  
 self.volume.setIcon(self.icon1)  
 else:  
 self.player.play()  
 self.icon1.addPixmap(QtGui.QPixmap(":/newPrefix/volume.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)  
 self.volume.setIcon(self.icon1)  
  
 def retranslateUi(self, MainWindow):  
 \_translate = QtCore.QCoreApplication.translate  
 MainWindow.setWindowTitle(\_translate("MainWindow", "Ну, Погоди!"))  
 self.lvl1.setText(\_translate("MainWindow", "1"))  
 self.lvl2.setText(\_translate("MainWindow", "2"))  
 self.labelLVL.setText(\_translate("MainWindow", "Уровень сложности"))  
 self.play.setText(\_translate("MainWindow", "Играть"))  
 \_\_sortingEnabled = self.pravila.isSortingEnabled()  
 self.pravila.setSortingEnabled(False)  
 item = self.pravila.item(0)  
 item.setText(\_translate("MainWindow", "Правила игры"))  
 item = self.pravila.item(2)  
 item.setText(\_translate("MainWindow", "Управление волком "))  
 item = self.pravila.item(3)  
 item.setText(\_translate("MainWindow", "производится с помощью "))  
 item = self.pravila.item(4)  
 item.setText(\_translate("MainWindow", "стрелок на клавиатуре."))  
 item = self.pravila.item(6)  
 item.setText(\_translate("MainWindow", "Цель: собрать как можно"))  
 item = self.pravila.item(7)  
 item.setText(\_translate("MainWindow", "больше яиц."))  
 item = self.pravila.item(9)  
 item.setText(\_translate("MainWindow", "Выберите уровень "))  
 item = self.pravila.item(10)  
 item.setText(\_translate("MainWindow", "(1 - легкий, 2 - сложный)"))  
 item = self.pravila.item(11)  
 item.setText(\_translate("MainWindow", "и нажмите кнопку \"Играть\""))  
 self.pravila.setSortingEnabled(\_\_sortingEnabled)  
 self.urlicon.setText(\_translate("MainWindow",  
 "<a target=\"\_blank\" href=\"https://icons8.com/icon/10568/question-mark\">Question Mark</a> icon by <a target=\"\_blank\" href=\"https://icons8.com\">Icons8</a>"))  
 self.labelgame.setText(\_translate("MainWindow", "ИГРА"))  
 self.labelgame1.setText(\_translate("MainWindow", "1"))  
 self.labelgame2.setText(\_translate("MainWindow", "2"))  
 self.labelPomer.setText(\_translate("MainWindow", "Игра окончена"))  
 self.butVihod.setText(\_translate("MainWindow", "Выход"))  
  
 def keyPressEvent(self, e):  
 if e.key() == Qt.Key\_Up:  
 self.VUL.setVisible(True)  
 self.VDL.setVisible(False)  
 self.VDR.setVisible(False)  
 self.VUR.setVisible(False)  
 self.EVUl = True  
 self.EVDL = False  
 self.EVDR = False  
 self.EVUR = False  
 if e.key() == Qt.Key\_Left:  
 self.VUL.setVisible(False)  
 self.VDL.setVisible(True)  
 self.VDR.setVisible(False)  
 self.VUR.setVisible(False)  
 self.EVUl = False  
 self.EVDL = True  
 self.EVDR = False  
 self.EVUR = False  
 if e.key() == Qt.Key\_Down:  
 self.VUL.setVisible(False)  
 self.VDL.setVisible(False)  
 self.VDR.setVisible(True)  
 self.VUR.setVisible(False)  
 self.EVUl = False  
 self.EVDL = False  
 self.EVDR = True  
 self.EVUR = False  
 if e.key() == Qt.Key\_Right:  
 self.VUL.setVisible(False)  
 self.VDL.setVisible(False)  
 self.VDR.setVisible(False)  
 self.VUR.setVisible(True)  
 self.EVUl = False  
 self.EVDL = False  
 self.EVDR = False  
 self.EVUR = True  
  
 def egg\_time(self):  
 self.count.setProperty("intValue", self.chet)  
 e = random.randint(0,1000000)  
 if 0 <= e <= 250000:  
 self.egg\_spawn = 1  
 elif 250001 <= e <= 500000:  
 self.egg\_spawn = 2  
 elif 500001 <= e <= 750000:  
 self.egg\_spawn = 3  
 else:  
 self.egg\_spawn = 4  
  
 if self.chet <= 10:  
 self.timer1.setInterval(60)  
 self.timer2.setInterval(60)  
 self.timer3.setInterval(60)  
 self.timer4.setInterval(60)  
 elif 10 < self.chet <= 30:  
 self.timer1.setInterval(45)  
 self.timer2.setInterval(45)  
 self.timer3.setInterval(45)  
 self.timer4.setInterval(45)  
 elif 30 < self.chet <= 40:  
 self.timer1.setInterval(30)  
 self.timer2.setInterval(30)  
 self.timer3.setInterval(30)  
 self.timer4.setInterval(30)  
 elif 40 < self.chet <= 50:  
 self.timer1.setInterval(15)  
 self.timer2.setInterval(15)  
 self.timer3.setInterval(15)  
 self.timer4.setInterval(15)  
 elif 50 < self.chet <= 60:  
 self.timer1.setInterval(60)  
 self.timer2.setInterval(60)  
 self.timer3.setInterval(60)  
 self.timer4.setInterval(60)  
 elif 60 < self.chet <= 70:  
 self.timer1.setInterval(30)  
 self.timer2.setInterval(30)  
 self.timer3.setInterval(30)  
 self.timer4.setInterval(30)  
 elif 70 < self.chet <= 80:  
 self.timer1.setInterval(15)  
 self.timer2.setInterval(15)  
 self.timer3.setInterval(15)  
 self.timer4.setInterval(15)  
 elif 80 < self.chet:  
 self.timer1.setInterval(20)  
 self.timer2.setInterval(20)  
 self.timer3.setInterval(20)  
 self.timer4.setInterval(20)  
  
 if self.egg\_spawn == 1:  
  
 self.et1 = True  
 self.timer1.start()  
  
 if self.egg\_spawn == 2:  
  
 self.et2 = True  
 self.timer2.start()  
  
 if self.egg\_spawn == 3:  
  
 self.et3 = True  
 self.timer3.start()  
  
 if self.egg\_spawn == 4:  
  
 self.et4 = True  
 self.timer4.start()  
  
 def updateEggPosition1(self):  
 self.count.setProperty("intValue", self.chet)  
 self.ELU1.move(self.egg1\_x, self.egg1\_y)  
 self.ELU1.setVisible(True)  
 self.egg1\_x += self.egg\_speed\_x  
 self.egg1\_y += self.egg\_speed\_y # (390;297)  
  
 if self.egg1\_x >= 390 and self.egg1\_y >= 297 and self.EVUl == True :  
 self.chet += 1  
 self.egg1\_x = 310  
 self.egg1\_y = 257  
 self.ELU1.setVisible(False)  
 self.timer1.stop()  
 self.et1 = False  
 elif self.egg1\_x >= 390 and self.egg1\_y >= 297 and self.EVUl == False:  
 self.badshet +=1  
 self.egg1\_x = 310  
 self.egg1\_y = 257  
 self.ELU1.move(self.egg1\_x, self.egg1\_y)  
 self.ELU1.setVisible(False)  
 self.timer1.stop()  
 self.et1 = False  
 else:  
 self.ELU1.move(self.egg1\_x, self.egg1\_y)  
  
 if self.badshet == 3:  
 self.End()  
  
 def updateEggPosition2(self):  
 self.count.setProperty("intValue", self.chet)  
 self.ELD1.move(self.egg2\_x, self.egg2\_y)  
 self.ELD1.setVisible(True)  
 self.egg2\_x += self.egg\_speed\_x  
 self.egg2\_y += self.egg\_speed\_y #(390;397)  
  
  
 if self.egg2\_x >= 390 and self.egg2\_y >= 397 and self.EVDL == True:  
 self.chet += 1  
 self.egg2\_x = 310  
 self.egg2\_y = 357  
 self.ELD1.setVisible(False)  
 self.timer2.stop()  
 self.et2 = False  
 elif self.egg2\_x >= 390 and self.egg2\_y >= 397 and self.EVDL == False:  
 self.badshet += 1  
 self.egg2\_x = 310  
 self.egg2\_y = 357  
 self.ELD1.setVisible(False)  
 self.timer2.stop()  
 self.et2 = False  
 else:  
 self.ELD1.move(self.egg2\_x, self.egg2\_y)  
  
 if self.badshet == 3:  
 self.End()  
  
 def updateEggPosition3(self):  
 self.count.setProperty("intValue", self.chet)  
 self.ERU1.move(self.egg3\_x, self.egg3\_y)  
 self.ERU1.setVisible(True)  
 self.egg3\_x -= self.egg\_speed\_x  
 self.egg3\_y += self.egg\_speed\_y #(780;297)  
  
 if self.egg3\_x <= 780 and self.egg3\_y >= 297 and self.EVUR == True:  
 self.chet += 1  
 self.egg3\_x = 860  
 self.egg3\_y = 257  
 self.ERU1.setVisible(False)  
 self.timer3.stop()  
 self.et3 = False  
 elif self.egg3\_x <= 780 and self.egg3\_y >= 297 and self.EVUR == False:  
 self.badshet += 1  
 self.egg3\_x = 860  
 self.egg3\_y = 257  
 self.ERU1.setVisible(False)  
 self.timer3.stop()  
 self.et3 = False  
 else:  
 self.ERU1.move(self.egg3\_x, self.egg3\_y)  
  
 if self.badshet == 3:  
 self.End()  
  
 def updateEggPosition4(self):  
 self.count.setProperty("intValue", self.chet)  
 self.ERD1.move(self.egg4\_x, self.egg4\_y)  
 self.ERD1.setVisible(True)  
 self.egg4\_x -= self.egg\_speed\_x  
 self.egg4\_y += self.egg\_speed\_y #(780;397)  
  
 if self.egg4\_x <= 780 and self.egg4\_y >= 397 and self.EVDR == True:  
 self.chet += 1  
 self.egg4\_x = 860  
 self.egg4\_y = 357  
 self.ERD1.setVisible(False)  
 self.timer4.stop()  
 self.et4 = False  
 elif self.egg4\_x <= 780 and self.egg4\_y >= 397 and self.EVDR == False:  
 self.badshet += 1  
 self.egg4\_x = 860  
 self.egg4\_y = 357  
 self.ERD1.setVisible(False)  
 self.timer4.stop()  
 self.et4 = False  
 else:  
 self.ERD1.move(self.egg4\_x, self.egg4\_y)  
  
 if self.badshet == 3:  
 self.End()  
  
 def End(self):  
 self.timer.stop()  
 self.labelPomer.setVisible(True)  
 self.game\_over = True  
 self.play.setText('Играть')  
 self.timer1.stop()  
 self.timer2.stop()  
 self.timer3.stop()  
 self.timer4.stop()  
 self.ELU1.setVisible(False)  
 self.ELD1.setVisible(False)  
 self.ERU1.setVisible(False)  
 self.ERD1.setVisible(False)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 import sys  
 app = QtWidgets.QApplication(sys.argv)  
  
 ui = Ui\_MainWindow()  
  
 ui.show()  
 sys.exit(app.exec\_())

# **ПРИЛОЖЕНИЕ Б**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

# **ПРИЛОЖЕНИЕ В**

**ПРОГРАММА И МЕТОДИКА ИССЛЕДОВАНИЯ**

# **ПРИЛОЖЕНИЕ Г**

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**