**Пояснительная записка**

«Anorak's Quest»

**Обоснование выбора темы**

Разработка игровых приложений для досуга пользователей является сложной задачей, на освоение которой требуется много усилий и времени. Это включает в себя связку логической составляющей с графической. Для этого существует множество библиотек, в частности PyGame. Разработка программного обеспечения Anorak's Quest поможет лучше освоить работу с данной библиотекой, а также создаст полезный инструмент для досуга пользователей.

**Цели**

- Освоить разработку игровых приложений;

- Изучить библиотеку «PyGame»;

- Создать полезный инструмент для досуга пользователей.

**Выбор и обоснование инструментов для реализации**

Приложение было разработано на языке программирования «Python», который используют во многих областях IT-индустрии. Он обладает такими свойствами, как: простота освоения; лаконичность; интерпретируемость; широкая сфера применения; динамическая типизация; обширная поддержка библиотек; большое количество документации и обучающих материалов; мультиплатформенность.

Для создания данного приложения была выбрана среда разработки «PyCharm», находящаяся в свободном доступе. Она обладает следующими преимуществами, помогая создавать высококачественное программное обеспечение быстро и эффективно:

* Мощные инструменты подсказок и автодополнения, которые позволяют увеличить производительность и качество кода;
* Интегрированная система рефакторинга позволяет легко изменять структуру кода, делая его более читаемым и понятным;
* Отличная интеграция с инструментами управления версиями, такими как Git, что упрощает работу в команде;
* Интуитивно понятный интерфейс, который упрощает навигацию по проекту;
* Широкий выбор плагинов для настройки среды под свои потребности;
* Интеллектуальный анализ кода поможет выявить потенциальные проблемы и предложить варианты их решения;
* Возможность использования отладчика для пошагового анализа выполнения кода;
* Поддержка различных фреймворков, библиотек и инструментов для анализа кода.

Весь интерфейс был разработан на основе библиотеки «PyGame», игрового движка для создания мультимедийных приложений. Данная библиотека предназначена для создания двумерных графических и физических проектов, обладает высокой производительностью, имеет мощный набор инструментов, а также по ней доступна в открытом доступе подробная документация, позволяющая в кратчайшие сроки выполнить поставленную задачу.

**Этапы и их реализация продукта**

1. Разработать стартовое окно приложения, содержащую кнопки: «начать игру», «результаты» и «выход»;

Было разработано стартовое окно приложения. Для этого было подготовлено фоновое изображение для окна в виде пиксельного лесного пейзажа. Также был разработан класс кнопок «Button», отвечающий за интерфейс приложения. Данный класс при инициализации запрашивает координаты его отображения в окне, ширину и высоту кнопки, текст, а также функцию, выполняемую при нажатии на кнопку. С помощью данного класса в стартовое окно добавлены кнопки «начать игру», «результаты» и «выход». Кроме того, была создана функция «ochistka», которая отвечает за удаление всех элементов из приложения, функция «sozdanie\_bd» для создания базы данных в случае ее отсутствия, и функции «vihod», «statistika» и «vibor\_urovna», отвечающие за нажатия кнопок «выход», «результаты» и «начать игру» соответственно. Также был разработан отдельный класс для отображения фоновых изображений окон «fonoviy\_sloy». Для его инициализации требуется указать путь к файлу, загружаемому изображению. Для загрузки изображений для класса была разработана функция «load\_image», позволяющая загружать картинки формата «.png» в удобный для приложения вид. Для объединения всех элементов была разработана функция «startovoe\_okno», отвечающая за генерацию стартового окна приложения. Окно загружается автоматически при запуске программы

1. Разработать окно со списком уровней игры в виде кнопок и кнопки «вернуться»;

Было разработано окно со списком из шести уровней игры и кнопки, возвращающей пользователя приложения на стартовое окно. Для этого было подготовлено фоновое изображение для окна в виде пиксельного лесного пейзажа. Также была разработана функция «vibor\_urovna», которая генерирует данное окно. В него запрограммированы функции очистки данных «ochistka» и созданы кнопки в количестве семи штук с помощью разработанного класса «Button». Также была разработана функция «vibor\_personazha», отвечающая за окно с выбором скина персонажа. Данная функция принимает один аргумент k, который указывает на выбранный уровень. К каждой кнопке присвоено свое значение k в соответствии с уровнем, расположенным в ней, а также функция «vibor\_personazha». Кнопке возврата присвоена функция возврата на стартовое окно «startovoe\_okno».

1. Разработать окно с выбором скина (внешнего вида) персонажа с возможностью ввода его никнейма;

Было разработано окно с выбором скина (внешнего вида) персонажа с возможностью ввода его никнейма. Для этого было подготовлено фоновое изображение для окна в виде пиксельного лесного пейзажа. Также была разработана функция «vibor\_personazha», отвечающая за генерацию данного окна. Также были разработаны классы «AnimatedSprite» и «svoi\_text» для анимации персонажа и вывода текста соответственно. Класс «AnimatedSprite» позволяет приводить действия персонажа в анимацию путем разделения одного изображения на несколько частей и переключением активной частив зависимости от таймера внутри класса. Для этого были созданы две анимированные модели персонажа. Класс «svoi\_text» позволяет выводить любую текстовую информацию на игровое поле. Для его инициализации необходимо указать отображаемый текст и его координаты. Также по желанию можно разрешить ему редактироваться от нажатий клавиатуры и изменить его цвет. Для разработанного окна с выбором скина было добавлено два экземпляра класса «svoi\_text». Один является статичным текстом, отображающим пояснительные подсказки о том, как поменять никнейм персонажа. Второй экземпляр класса является динамичным текстом, которому дано разрешение на изменения от нажатий кнопок клавиатуры. Они расположены в нижнем левом углу окна друг под другом. Также были добавлены 2 анимированных модели персонажа с помощью класса «AnimatedSprite» и 2 кнопки выбора персонажа под ними с помощью класса «Button». Также были разработаны функции, отвечающие за загрузку уровней: «uroven\_1», «uroven\_2», «uroven\_3», «uroven\_4», «uroven\_5» и «uroven\_6». В зависимости от переданного параметра кнопка персонажа будет активировать соответствующий уровень. Также была добавлена кнопка возврата с помощью класса «Button» с ссылкой на функцию «vibor\_urovna». Перед загрузкой окна происходит очистка данных с помощью функции «ochistka».

1. Разработать уровни;

Для взаимодействия игрока с персонажем было разработано три основных уровня различной сложности: простой (первый), средний (второй) и сложный (третий). Дополнительные уровни, которые должны являться усложненной версией основных, в представленной версии игры не добавлены, требуется дополнительная проработка.

Для простого уровня было подготовлено два класса sprite “MainHero” и “Slime”, для среднего уровня было подготовлено два класса sprite “MainHero2” и “FireBall”, для сложного уровня было подготовлено три класса sprite “MainHero3”, “FireBall2”и “Staff”.

Для всех уровней были подготовлены общие фоны для окон в виде рамки с пиксельным изображение лесного пейзажа “forest”, поверхности земли “grass”, голубого неба “sky”, для класса “Slime” было подготовлено изображение светло-синей слизи “slime”.

Для классов “MainHero”, “MainHero2” и “MainHero3” были подготовлены соответствующие пиксельные изображения героя “Hero\_p1”, “Hero\_p1\_2”, “Hero\_p2”, “Hero\_p2\_2”.

Для класса “FireBall” было подготовлено изображение “FireBall”, для класса “FireBall2” было подготовлено изображение “FireBall2”.

Для класса “Staff” было подготовлено изображение “Staff”.

Для простого уровня была разработана функция “uroven\_1”, для среднего уровня была разработана функция “uroven\_2”, для сложного уровня была разработана функция “uroven\_3”. Данные функции отвечают за запуск окна для игры на соответствующем уровне. В эти функции были добавлены элементы очистки в виде функции “ochistka”.

Для каждого уровня были заданы размеры фоновых окон и объектов. Для каждого объекта были вычислены координаты для расположения в фоновых окнах.

На всех трех уровнях взаимодействие игрока с персонажем происходит через нажатие стрелок. При нажатии стрелки “вверх” персонаж прыгает, “вправо” - передвигается в правую сторону, “влево” - передвигается в левую сторону.

При касании персонажем моба (slime или FireBall), персонаж перемещается на начальные координаты.

На представленной версии игры не добавлено поле для текстовых сообщений с информацией об уровне, требуется дополнительная проработка.

1. Разработать загрузку «победного» изображения конца уровня с результатом его прохождения;

Было разработано окно завершения уровня с результатом его прохождения. Для этого было подготовлено фоновое изображение для окна в виде пиксельного лесного пейзажа. Также была разработана функция «pamat». В нее были добавлены элементы очистки в виде функции «ochistka». Разработанная функция загружает подготовленное фоновое изображения и по центру выводит введенный пользователем никнейм и время прохождения уровня.

1. Разработать окно списка результатов по пройденным уровням пользователями;

Было разработано окно с выводом результатов по пройденный уровням пользователями. Для этого было подготовлено фоновое изображение для окна в виде рамки с пиксельным лесным пейзажем. Также разработана функция «statistika», отвечающая за генерацию данного окна. В нее входят функции очистки данных «ochistka», а также создания текста с помощью класса «svoi\_text» и выбора фонового изображения с помощью класса «fonoviy\_sloy». Для каждого результата создается свой экземпляр класса «svoi\_text», расположенных друг под другом.

1. Разработать базу данных;

Была разработана база данных, основанная на языке программирования «SQL» и состоящая из одной таблицы «history». «history» - главная таблица с историей всех прохождений уровней. В нее входят следующие данные:

• id отвечает за номер прохождения,

• name – введенный ник пользователя,

• time – время прохождения уровня,

• level – выбранный уровень,

• skin – номер выбранного скина,

**Процесс работы продукта**

При запуске программы открывается стартовое окно приложения. На нем отображены следующие кнопки: «начать игру», «результаты» и «выход».

При нажатии на кнопку «начать игру» перед пользователем открывается список всех уровней. Каждый уровень состоит из фона, персонажа, площадки для его перемещения, текстового поля для сообщений, а также дополнительных предметов (платформы, двери, рычаги, кнопки и т.д.). Персонаж является физическим объектом и не проваливается насквозь площадки. Он может передвигаться в левую и правую стороны, совершать прыжки, а также взаимодействовать с окружающими предметами (например, нажатие на кнопку).

В основные уровни входит следующий ряд целей: добраться до пункта назначения, преодолеть препятствия (паркур) и логические задачи (испытания на память).

После выбора уровня загружается окно с выбором скина (внешнего вида) персонажа, а также поля ввода его имени. На данный момент выбор состоит из двух вариантов внешнего вида.

После выбора персонажа загружается выбранный уровень и пользователю дается контроль над персонажем. При старте каждого уровня, в диалоговое окно выводится информация по нему, а также подсказки, как его пройти.

После завершения уровня пользователю загружается «победное» изображение, а также его результат, а именно, за сколько уровень был пройден.

При нажатии на кнопку «результаты» перед пользователем открывается список всех результатов по пройденным уровням всеми пользователями.

При нажатии на кнопку «выход» приложение закрывается.