Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Вычислительная техника

УТВЕРЖДАЮ
Заведующий кафедрой ВТ
_____ О.В. Непомнящий
подпись
«_____» _____2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Android-приложение для запоминания картин в игровой форме

Руководитель	_____ подпись, дата	ст. преподаватель	В.С. Васильев
Выпускник	_____ подпись, дата		У.А. Брюханова
Консультант	_____ подпись, дата	доцент, канд. техн. наук	Л. И. Покидышева
Нормоконтролер	_____ подпись, дата	доцент, канд. техн. наук	В. И. Иванов

Красноярск 2019

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Вычислительная техника

УТВЕРЖДАЮ
Заведующий кафедрой ВТ
_____ О.В. Непомнящий
подпись
«_____» _____2019 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

в форме бакалаврской работы

Студенту Брюхановой Ульяне Алексеевне

Группа 15-07Б Направление (специальность) 09.03.01

Информатика и вычислительная техника

Тема выпускной квалификационной работы Android-приложение для запоминания картин в игровой форме

Утверждена приказом по университету № _____ от _____

Руководитель ВКР В.С. Васильев, старший преподаватель кафедры «Информатика и вычислительная техника» ИКИТ СФУ. Консультант Л.И. Покидышева, профессор кафедры «Информатика и вычислительная техника» ИКИТ СФУ, кандидат технических наук»

Исходные данные для ВКР Требования и пожелания к приложению от преподавателей Сибирского государственного института искусств имени Дмитрия Хворостовского, рекомендации руководителя

Перечень разделов ВКР Введение, анализ задания на ВКР, проектирование, реализация и документация, заключение, список использованных источников

Перечень графического материала нет

Руководитель ВКР _____ В.С. Васильев
подпись, дата

Задание принял к исполнению _____ У.А. Брюханова
подпись, дата

« ___ » _____ 2019 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Android-приложение для запоминания картин в игровой форме» содержит 47 страниц текстового документа, 2 приложения, 31 использованный источник, 30 рисунков, 1 таблицу.

ПАЗЛЫ, ANDROID, QT, ИСКУССТВОВЕДЕНИЕ, ЖИВОПИСЬ, ОБУЧЕНИЕ.

Цель работы: разработка мобильного приложения «Пазлы-Живопись», для использования студентами Сибирского государственного института искусств имени Дмитрия Хворостовского в рамках дисциплин «История отечественного искусства и культуры», «История зарубежного искусства и культуры».

В результате выполнения ВКР было разработано приложение «Пазлы-Живопись».

В выпускную квалификационную работу входит введение, 3 главы и заключение.

Во введении ставится цель, и выполняется ее декомпозиция на задачи.

В первой главе рассматриваются аналоги, формулируются основные требования к разрабатываемому приложению и выбираются инструменты разработки.

Во второй главе разрабатываются: архитектура системы, структура базы данных, структуры данных и алгоритм формирования элементов пазла требуемой формы.

В третьей главе описана реализация основных элементов приложения. Составлена инструкция разработчика, проведено тестирование разработанного приложения «Пазлы-Живопись» и модуля обновления.

В заключении подводятся итоги по выполненной работе.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 4
1 АНАЛИЗ ЗАДАНИЯ НА ВКР 5
1.1 Анализ существующих аналогов 5
1.2 Разработка технического задания на игру «Пазлы» 8
1.2.1 Интерфейс программы 8
1.2.2 Функциональные требования к программе 9
1.2.3 Нефункциональные требования 11
1.3 Разработка технического задания на игру «Пазлы-Живопись» 11
1.3.1 Интерфейс программы 11
1.3.2 Функциональные требования 12
1.3.3 Нефункциональные требования 13
1.4 Выбор инструментов 14
1.5 Выводы по главе 14
2 ПРОЕКТИРОВАНИЕ 15
2.1 Динамическая модель системы 16
2.1.1 Диаграммы последовательности 16
2.1.2 Алгоритм построения пазлов 18
2.1.3 База данных 24
2.1.4 Диаграммы классов 26
2.2 Выводы по главе 31
3 РЕАЛИЗАЦИЯ И ДОКУМЕНТАЦИЯ 32
3.1 Реализация 32
3.1.1 Игровой процесс 32
3.1.2 Меню 34
3.1.3 Справка 34
3.1.4 База данных 35
3.1.5 Интерфейс 36
3.2 Инструкция разработчика 36
3.3 Тестирование 37
3.4 Выводы по главе 37

ЗАКЛЮЧЕНИЕ 38
СПИСОК СОКРАЩЕНИЙ..... 39
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 40
ПРИЛОЖЕНИЕ А Установка QT для платформы Android..... 43
ПРИЛОЖЕНИЕ Б Сертификат участника в конференции..... 47

ВВЕДЕНИЕ

Пазлы являются классической игрой, направленной на развитие логического и пространственного мышления у детей. В настоящее время существует тенденция реализации такого рода игр в виде приложений для мобильных платформ, так как это позволяет родителям экономить деньги, а также обеспечивает сохранность игры (элементы виртуального пазла нельзя порвать или потерять). Однако, имеющиеся в Google Play [1] приложения, либо являются платными, либо перегружены рекламой.

Студенты Сибирского государственного института искусств имени Дмитрия Хворостовского в рамках дисциплин «История отечественного искусства и культуры», «История зарубежного искусства и культуры» заучивают огромное количество картин, при этом ряд изображений очень похожи – например иконы 12-13 веков. Преподавателями СГИИ было предложено разработать приложение, напоминающее игру Пазлы, но обладающее рядом отличий.

Целью работы является разработка мобильного приложения «Пазлы-Живопись», для использования студентами СГИИ в рамках дисциплин «История отечественного искусства и культуры», «История зарубежного искусства и культуры». Для достижения цели в работе решаются следующие задачи:

- Составить техническое задание на разработку приложения «Пазлы», основываясь на прецедентах. Выполнить его проектирование и разработку. Продемонстрировать «Пазлы» преподавателям СГИИ, собрать их предложения по доработке в виде пользовательских историй (user stories).
- Доработать «Пазлы» с учетом предложений – «Пазлы-Живопись», разместить их на сервере Google Play, выполнить тестирование.
- Разработать инструкции пользователя и программиста.

1 Анализ задания на ВКР

В соответствии с заданием на ВКР необходимо разработать два приложения «Пазлы» и «Пазлы-Живопись» – одно реализует классическую игру, а второе обладает рядом особенностей. Для обеспечения конкурентоспособности приложения, выполнен анализ сильных и слабых сторон аналогичных приложений. С их учетом составлено техническое задание на разработку. Также, проанализирован ряд проектов с открытым исходным кодом на предмет возможности их доработки.

Проектирование решено вести на основе процесса ICONIX [2], в основе которого лежит язык моделирования UML [3, 4]. Процесс ICONIX базируется на вариантах использования, которые выполняют роль функциональной части технического задания. Варианты использования описывают диалог системы с внешней средой, которой в нашем случае является пользователь. Пользователь работает с программой посредством интерфейса, поэтому сначала, в виде диаграмм последовательности экранов [5] разработаны макеты интерфейса, а затем, на их основе, - описаны варианты использования.

1.1 Анализ существующих аналогов

На сервисе github.com по запросу «Jigsaw Puzzle» найдено 549 проектов, написанных на разных языках программирования, из которых 31 – на C++.

В связи с тем, что приложение разрабатывается под ОС Android, а на github в настройках поиска нет возможности выбора типа операционной системы – к запросу добавлено ключевое слово «Android». В результате было найдено 36 проектов, из которых большая часть написана на языке JavaScript и ни одного на C++. Часть найденных проектов не имеет никакого отношения к «Jigsaw Puzzle».

Из 549 проектов на github.com выделено 5, которые подходят под задачи ВКР. Результаты их анализа приведены в таблице.

Таблица – Основные показатели приложений на github.com

Название	Язык	Кол.-во строчек кода (шт.)	Лицензия	Удалось запустить	Документация
JigSawPuzzle2-Android	Java Script	1116	Не указано	нет	нет
Canvas Jigsaw	Java Script	900	MIT	да	да*
JigsawPuzzle	Swift	281	Не указано	нет	нет
Tetzle	C++	8428	GPL-3.0	да	нет
puzzle	C++	277	Не указано	да	да

– JigSawPuzzle2-Android [6]. К описанию игры приложена gif анимация, демонстрирующая процесс игры. Игровое поле представляет собой *правильно собранное прозрачное изображение*, на которое должны помещаться Фрагменты. Все Фрагменты изначально располагаются в правой части формы. При правильной установке становятся неактивными.

– Canvas Jigsaw [7]. Приложение запускается в браузере. При перетаскивании Фрагмента игровое поле и остальные Фрагменты становятся неактивными, а также прозрачными. Присутствует окно настройки, позволяющее выбирать количество разбиений и загружать собственные изображения.

  *В документации описана инструкция по сборке приложения, а также есть ссылка на источник, где можно подробнее прочитать о нем, но ссылка является недоступной.

– JigsawPuzzle [8]. Позволяет использовать только прямоугольную форму Фрагментов.

– Tetzle [9]. Позволяет разбивать собственное загруженное изображение в виде геометрических фигур тетрамино с возможностью поворота. При выделении, Фрагмент начинает подсвечиваться желтым цветом.

Присутствует функция выделения большого числа фигур, что иногда мешает при сборе.

– puzzle [10]. Фрагменты разбиты на квадраты, нет возможности передвигать их мышкой, только стрелками клавиатуры. Не интуитивно понятен механизм выбора изображения, окно приложения нельзя уменьшить или увеличить, а также оно не вписывается в расширение экрана на запускаемой машине.

В Google Play по запросу «Пазлы» найдено 250 приложений, практически в каждом присутствует реклама и больше половины поддерживают покупки. Кроме того, приложения однотипны и отличаются, в основном, тематикой и количеством картинок. Из всех найденных подробно рассмотрены два:

– Jigsaw Puzzles Epic позволяет собирать пазлы как из заранее заготовленных изображений, так и пользовательских. Объем – 42 Мб.
– «Чудо Пазлы» позволяют выбрать уровень сложности, влияющий на количество Фрагментов пазла, а также задавать возможность их вращения. Объем – 82 Мб.

По результатам просмотра аналогов, принято решение добавить возможность поворота фрагментов пазла, подсвечивание Фрагмента при взаимодействии с ним.

Наиболее подходящими приложениями для заимствования кода в процессе разработки стали: Canvas Jigsaw и Tetzle. В случае использования Canvas Jigsaw с лицензий MIT, можно использовать код, указав авторские права и условия лицензионного соглашения. В случае использования Tetzle с лицензией GPL-3.0, необходимо будет предоставлять другим свободный доступ к программе и ее исходному код.

1.2 Разработка технического задания на игру «Пазлы»

1.2.1 Интерфейс программы

На рисунке 1 представлена диаграмма потока экранов для приложения «Пазлы», демонстрирующая переходы между окнами.

На каждом экране можно выбрать сложность сбора пазла и установить опцию поворота Фрагмента. При переходе от окна к окну установленная конфигурация сохраняется.

«Главное меню» содержит список всех категорий с возможностью выбора одной из них. «Выбор картинки» отображает список картинок выбранной категории. После выбора «Случайная картинка» запустится «Игровое поле» со случайной картинкой, выбранной из списка имеющихся в БД.

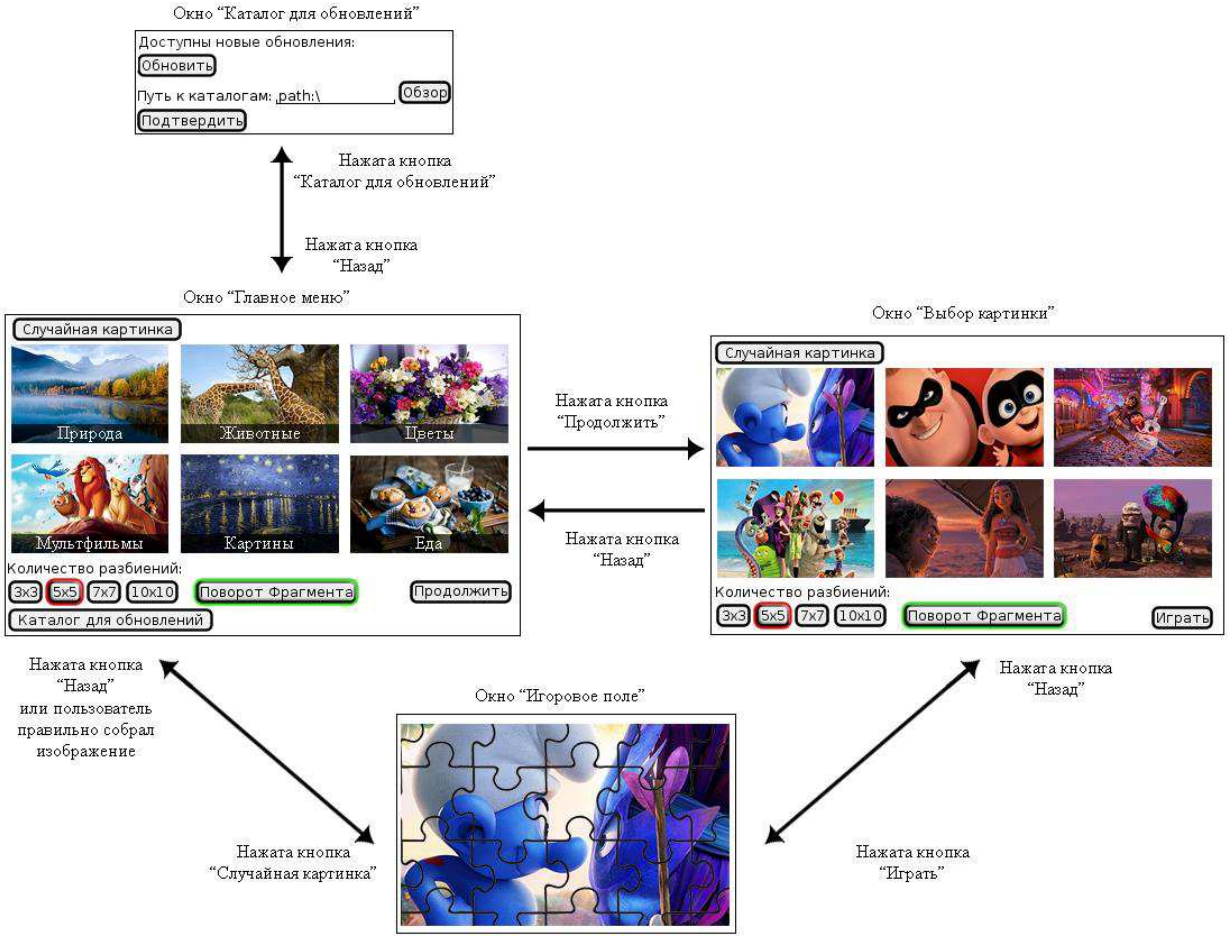


Рисунок 1 - Диаграмма потока экранов приложения «Пазлы»

1.2.2 Функциональные требования к программе

Функциональные требования выражены в виде диаграмм вариантов использования (рисунок 2).

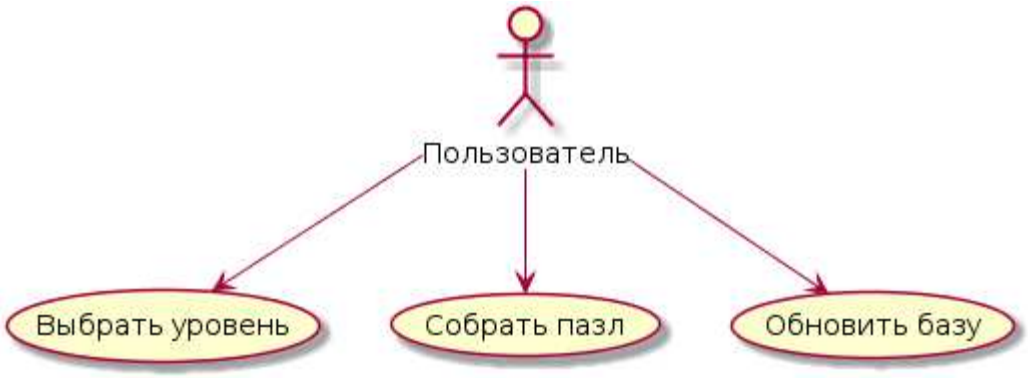


Рисунок 2 – Диаграмма вариантов использования приложения «Пазлы»

Текстовое описание прецедентов:

Название прецедента: Обновить базу.

Цель сценария: загрузить дополнительные изображения с сервера при их наличии.

Предусловия: пользователь находится в окне «Каталог для обновлений», нажал кнопку «Обновить».

Основной сценарий:

A. С сервера загружается файл с «индексом» (в формате JSON), в котором хранится список изображений, доступных для скачивания, а также указана дата их добавления.

B. При наличии обновлений – с сервера скачиваются в Каталог изображений все *новые* картинки, информация о них добавляется в базу данных.

Постусловия: Новые изображения появляются в меню.

Условие ввода в действие альтернативных сценариев

Условие 1. Изначально не доступен сервер.

A. Приложение выдает сообщение о невозможности загрузки.

Условие 2. Связь с сервером пропадает в момент загрузки изображения.

A. Изображения, которые удалось загрузить — добавляются в базу и отображаются в меню.

B. Приложение выдает сообщение о невозможности загрузки.

C. При повторном установлении связи с сервером, оставшиеся картинки докачиваются в Каталог.

Условие 3. Связь с сервером пропадает в момент загрузки файла с индексом.

A. Приложение выдает сообщение о невозможности загрузки.

Название прецедента: Собрать пазл.

Цель сценария: собрать изображение, разбитое на Фрагменты.

Предусловия: пользователь выполнил прецедент «Выбрать уровень».

Основной сценарий:

A. Программа делит изображение на Фрагменты.

B. Пользователь собирает изображение до тех пор, пока все Фрагменты не установятся в свои ячейки.

Постусловия: открывается окно меню, пройденный уровень отмечается.

Условие ввода в действие альтернативных сценариев

Условие 1. Неполное завершение сбора пазла.

A. Переход в «Главное меню» без сохранения прогресса прохождения уровня.

Название прецедента: Выбрать уровень.

Предусловия: пользователь находится в окне «Главное меню».

Основной сценарий:

A. В окне отображаются категории картинок, пользователь выбирает одну из них.

B. Отображается окно «Выбор картинки», содержащее изображения выбранной категории. Пользователь выбирает требуемое число

разбиений картинки и возможность поворота Фрагмента, а также картинку.

Постусловия: выполняется прецедент «Собрать пазл» с выбранными параметрами.

Условие ввода в действие альтернативных сценариев

Условие 1. «Случайная картинка»

Предусловия: пользователь выбрал категорию, отображается список ее картинок.

А. Пользователь выбирает «Случайная картинка».

Постусловия: выполняется прецедент «Собрать пазл» с выбранными параметрами сложности и случайной картинкой.

1.2.3 Нефункциональные требования

Приложение работает на платформе Android, начиная с минимальной версии 4.3 и уровнем API 18.

Приложение обладает удобным пользовательским интерфейсом:

- Не напрягающие глаза цвета фона и кнопок.
- Наличие справки с информацией для управления приложением.
- Обработка жестов пальцами (масштабирование пазла), и стандартной кнопки Android back button.

1.3 Разработка технического задания на игру «Пазлы-Живопись»

1.3.1 Интерфейс программы

На рисунке 3 представлена диаграмма потока экранов приложения «Пазлы-Живопись», которая имеет схожий интерфейс и функционал с «Пазлы». Но включает дополнительный экран для просмотра информации о собранной картине и экран отображения автора определенной эпохи.

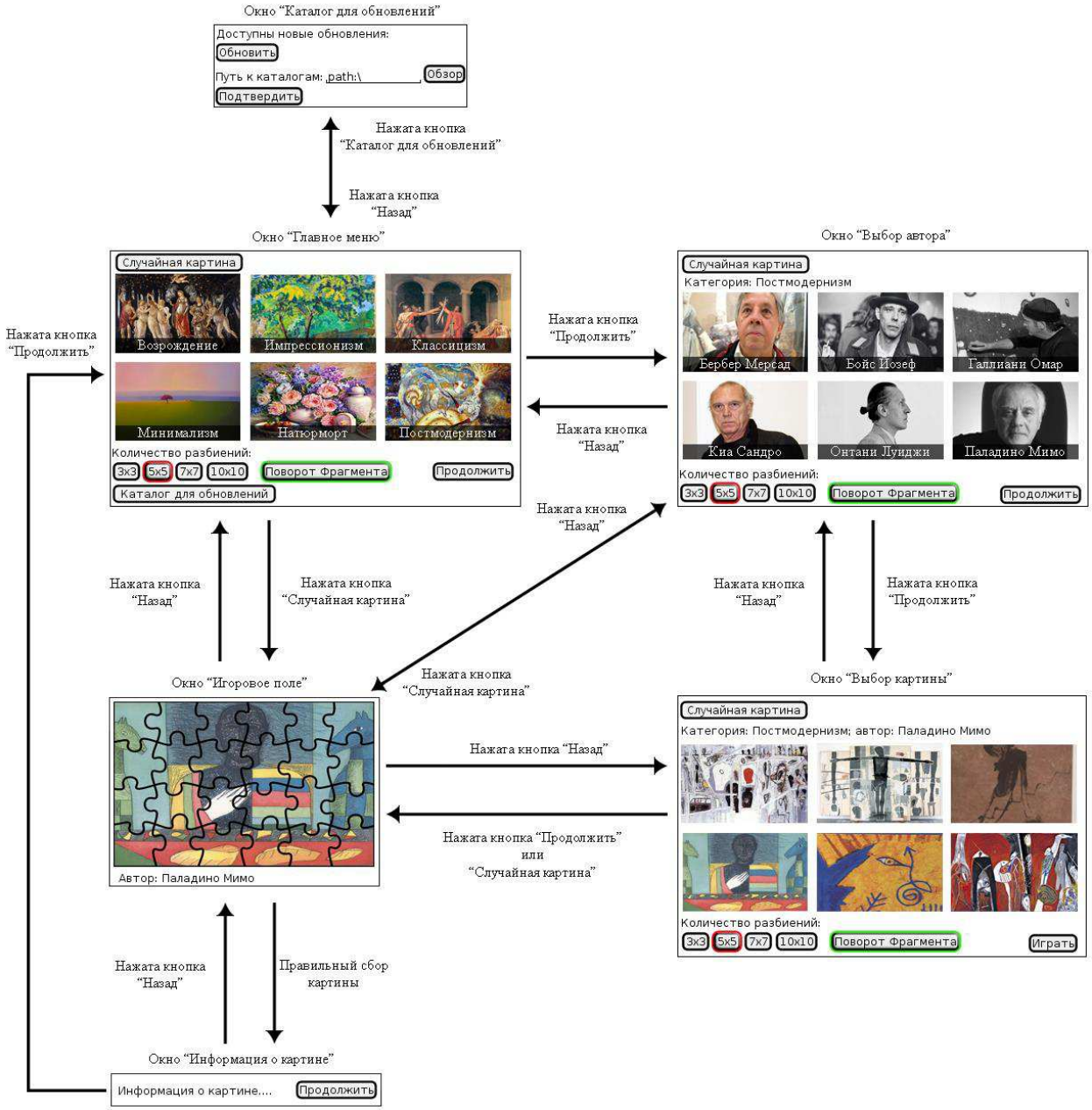


Рисунок 3 - Диаграмма потока экранов приложения «Пазлы-Живопись»

1.3.2 Функциональные требования

Приложение для студентов СГИИ имеет аналогичную диаграмму вариантов использования, как в пункте 1.2.2, но с включением некоторых дополнений.

В прецеденте «Собрать пазл», после правильного сбора изображения будет выведена информация о картине, только после этого пользователь может перейти в окно «Главное меню». Таким образом появляется новый прецедент

«Отобразить информацию о картине». Кроме того, изменится прецедент «Выбрать уровень».

Название прецедента: Отобразить информацию о картине

Цель сценария: вывести информацию о собранной картине.

Предусловия: пользователь успешно собрал картину и находится на экране «Отображение информации о картине».

Основной сценарий:

A. Приложение находит в БД информацию о собранной картине.

B. Приложение выводит информацию на экран о собранной картине.

Постусловия: прочтение информации о картине пользователем.

Название прецедента: Выбрать уровень.

Предусловия: пользователь находится в окне «Главное меню».

Основной сценарий:

A. В окне отображаются категории направлений живописи, пользователь выбирает одну из них.

B. Отображается окно «Выбор автора», содержащее список художников выбранной категории. Пользователь выбирает автора.

C. Отображается окно «Выбор картины», содержащее список картин выбранного художника. Пользователь выбирает картину, требуемое число разбиений картинки и возможность поворота Фрагмента.

Постусловия: выполняется прецедент «Собрать пазл» с выбранными параметрами.

1.3.3 Нефункциональные требования

Совпадают с требованиями для приложения «Пазлы».

1.4 Выбор инструментов

Приложению необходимы библиотеки для реализации графического интерфейса на платформе Android, а также работы с сетью, базами данных и обработки JSON. Кроме того, существенной проблемой является необходимость эффективной обработки большого количества графических элементов (элементов пазла). Наиболее сложной алгоритмической задачей является разрезание изображения на фигурные части.

Наиболее полно необходимые библиотеки представлены в фреймворке Qt. На этот фреймворк, в частности, был перенесен мобильный клиент 2gis [11], при этом в открытый доступ была выложена большая кодовая база [12]. Кроме того, он использовался для разработки мобильной версии Telegram [13] и множества других приложений, что говорит о его высокой стабильности и эффективности.

Основным языком разработки для фреймворка Qt является C++.

1.5 Выводы по главе

На основе анализа аналогов, сделаны выводы:

- В Google Play нет приложения, полностью удовлетворяющего требованиям задания. «Обычные» пазлы в Google Play либо содержат платный контент, либо перегружены рекламой.
- Дорабатывать под задание проекты, выложенные на github нецелесообразно. Однако, имеет смысл обращаться к некоторым из них чтобы посмотреть удачные программные решения.
- Сформулированы требования к разрабатываемым приложениям в виде диаграмм прецедентов и их текстового описания.
- Выбраны инструменты разработки.

2 Проектирование

Прецеденты описывают поведение пользователя в системе. Для визуализации взаимодействия компонентов между собой, а также с пользователем используется диаграмма последовательностей, отражающая динамическую модель системы. Результатом проектирования является статическая модель системы – диаграмма классов, на основе которой может быть выполнена генерация кода.

На рисунке 4 приведена архитектура разработанной системы. Android-приложение использует базу данных с информацией об уровнях и файловую систему телефона для хранения изображений.

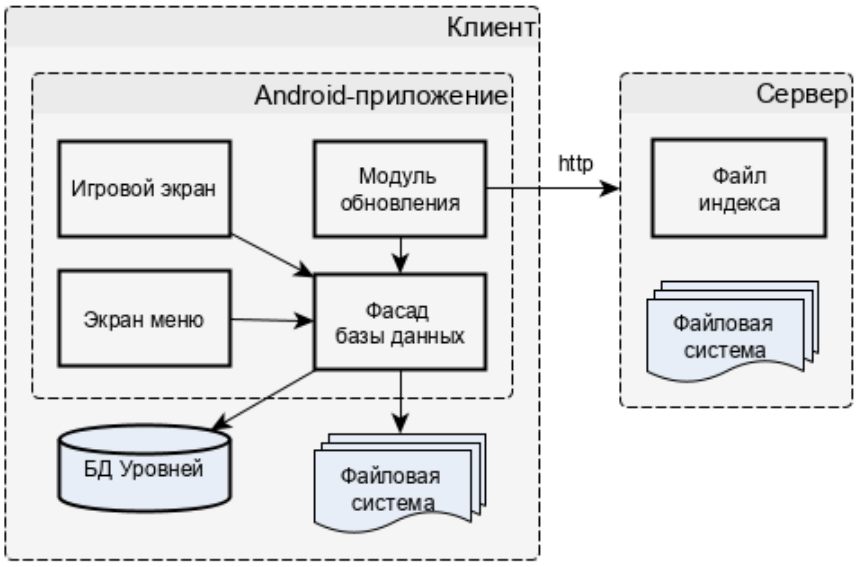


Рисунок 4 – Архитектура системы

Приложение состоит из ряда экранов, на рисунке показаны лишь основные. Экран меню загружает список категорий и уровней, отображает их иконки. Игровой экран отрисовывает изображение, разбитое на пазлы, соответствующее выбранному в меню уровню. Модуль обновления также взаимодействует с базой данных и файловой системой телефона. Взаимодействие осуществляется через объект-фасад [14], инкапсулирующий работу с базой данных и файловой системой.

2.1 Динамическая модель системы

2.1.1 Диаграммы последовательности

Значительная часть логики системы связана с меню и обновлением. В разделе приведены диаграммы последовательностей, для наиболее значимых прецедентов, связанных со взаимодействием различных компонентов системы.

На рисунке 5 изображена диаграмма последовательности для варианта использования «Обновление системы», наглядно представляющая процесс взаимодействия программы с сервером для получения новых уровней и их добавление в меню.

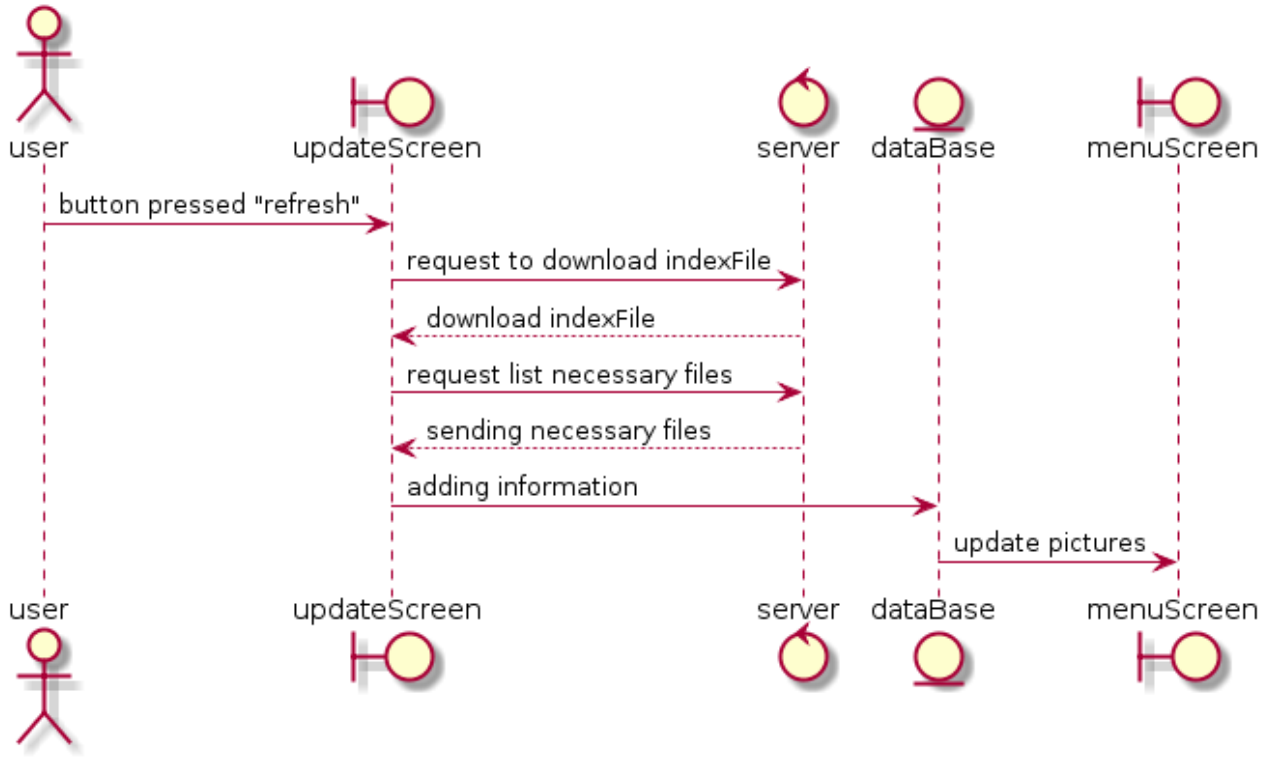


Рисунок 5 - Диаграмма последовательности «Обновление системы»

На рисунке 6 изображена диаграмма последовательности для варианта использования «Игра», представляющая процесс взаимодействия пользователя с игровым экраном при сборе пазла. В результате выделен класс ScreenController, переключающий экраны приложения.

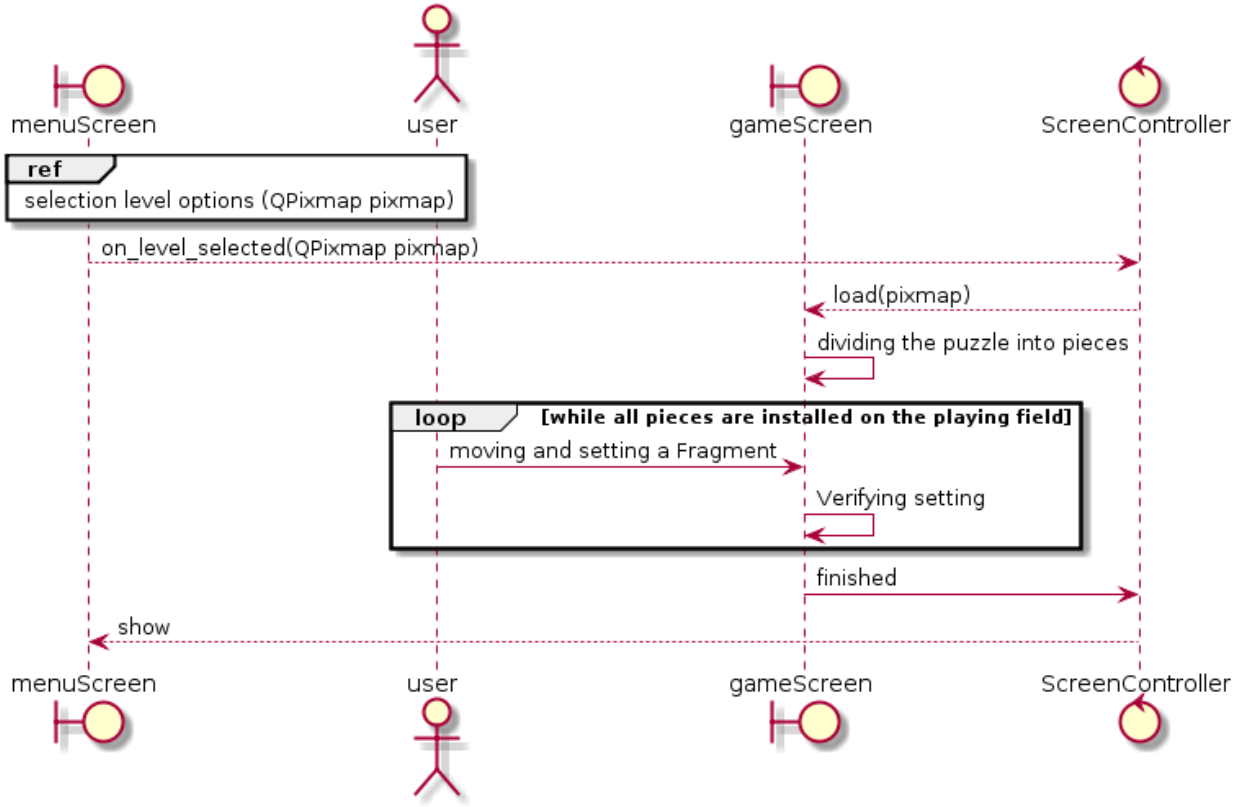


Рисунок 6 - Диаграмма последовательности «Игра»

На рисунке 7 изображена диаграмма последовательности для варианта использования «Выбор случайной картинки», представляющий процесс взаимоействия ScreenController с БД для выбора случайной картинки, экраном главного меню, игры и вывода информации о собранной картины.

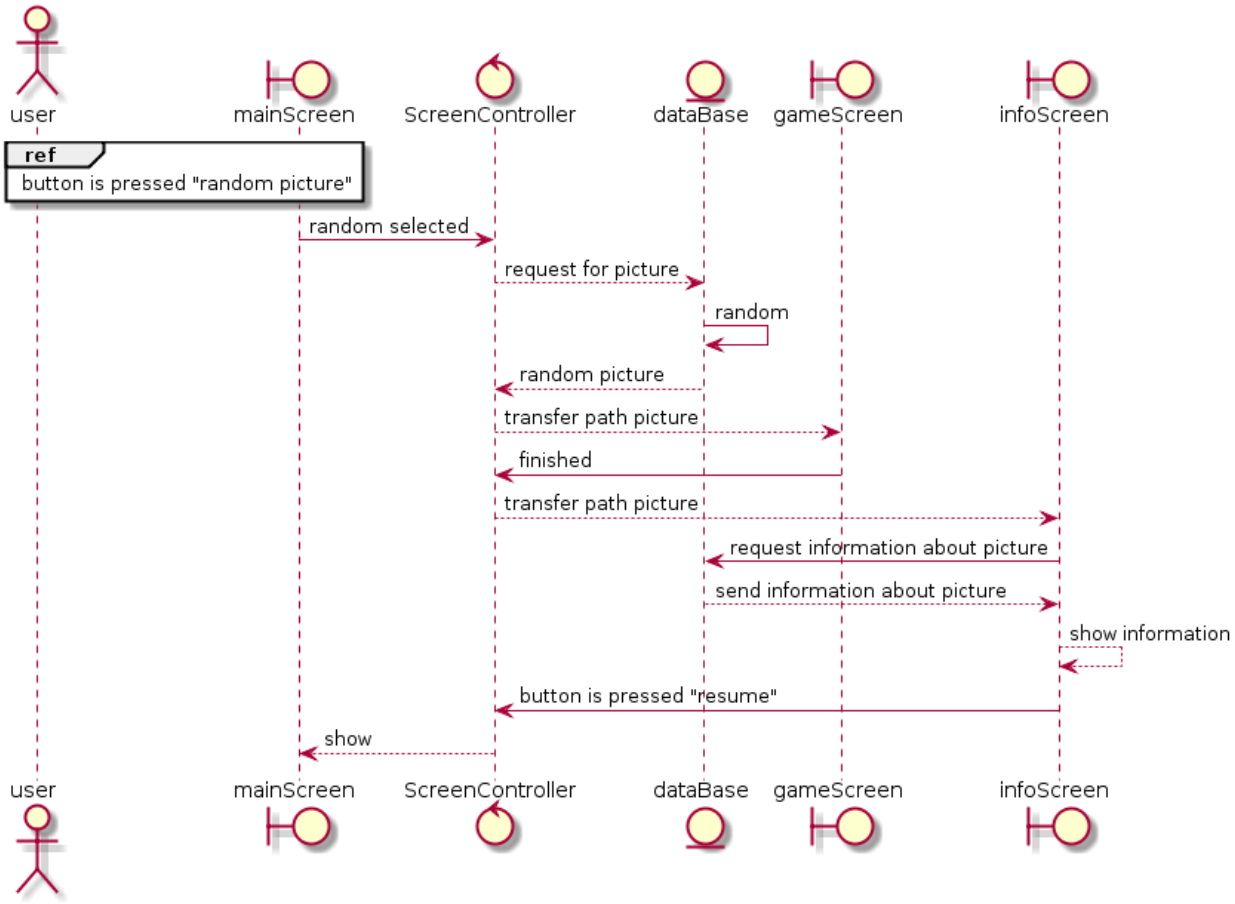


Рисунок 7 - Диаграмма последовательности «Выбор случайной картинки»

2.1.2 Алгоритм построения пазлов

Пазл состоит из Фрагментов различной формы (рисунок 8), которые «вырезаются» из изображения. *Фрагмент представляет собой* прямоугольник, с крючками, заходящими на области соседних Фрагментов. Их количество зависит от числа разбиений, вводимых пользователем.

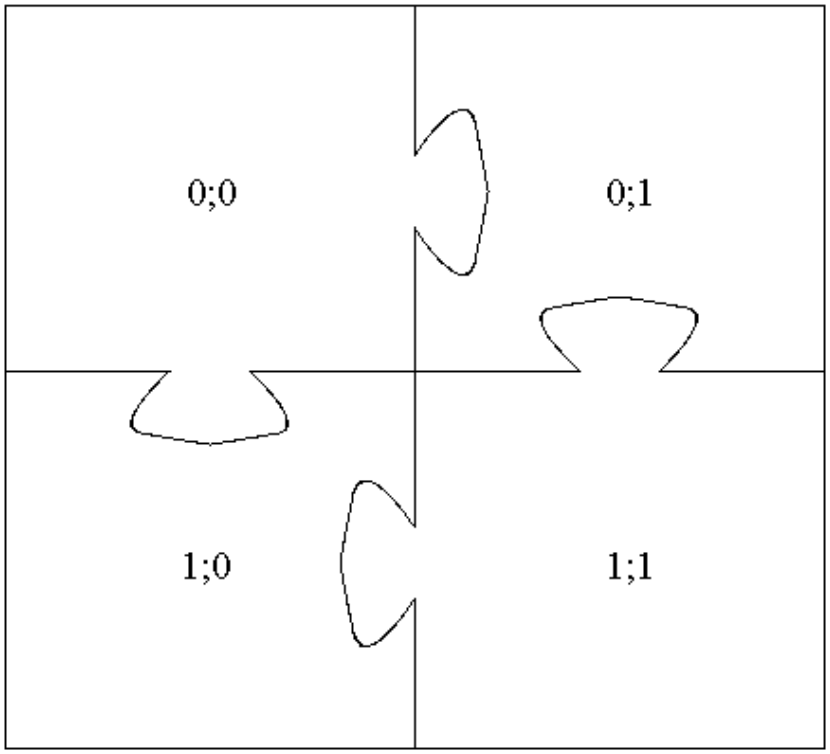


Рисунок 8 – Разбиение области на Фрагменты

Фрагмент строится из сторон (рисунок 9). Стороны могут быть вертикальными или горизонтальными, а также прямыми и Крючками. Крючок, также может быть двух типов: выпуклый и вогнутый. Если одна из сторон имеет вогнутый Крючок, то величина фрагмента не меняется, а если выгнутый – то соответствующая величина увеличивается на высоту или ширину крючка.

Рисунок 9 – Фрагмент пазла

Генерировать Крючок можно с помощью интерполяции. Интерполяция - способ нахождения промежуточных значений по имеющемуся дискретному набору известных значений. Таким образом, для задания требуемой формы крючков *может быть сформирован набор опорных точек, а затем с помощью интерполяции – получены промежуточные*.

Стороны строятся по опорным точкам: Крючок из 11, прямая линия из 2.

Чтобы получать Крючки разных форм, координаты точек b – j перерассчитываются в радиусе d (рисунок 10).

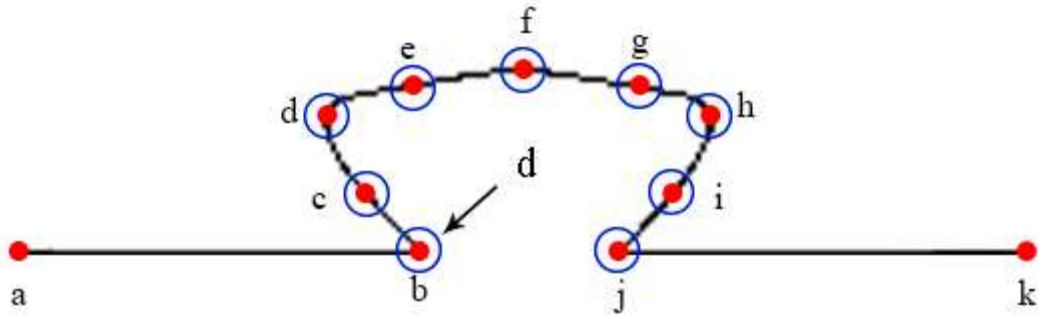


Рисунок 10 – Изображение точек и расстояний, на которое им можно сместиться

Для простоты реализации используется 4 типа сторон (горизонтальный/вертикальный крючки, горизонтальная/вертикальная прямые линии).

Для формирования Фрагментов пазла *используются двумерные массивы* из вертикальных (рисунок 11.a) и горизонтальных (рисунок 11.b) сторон. Данный двумерный массив соответствует рисунку 8.

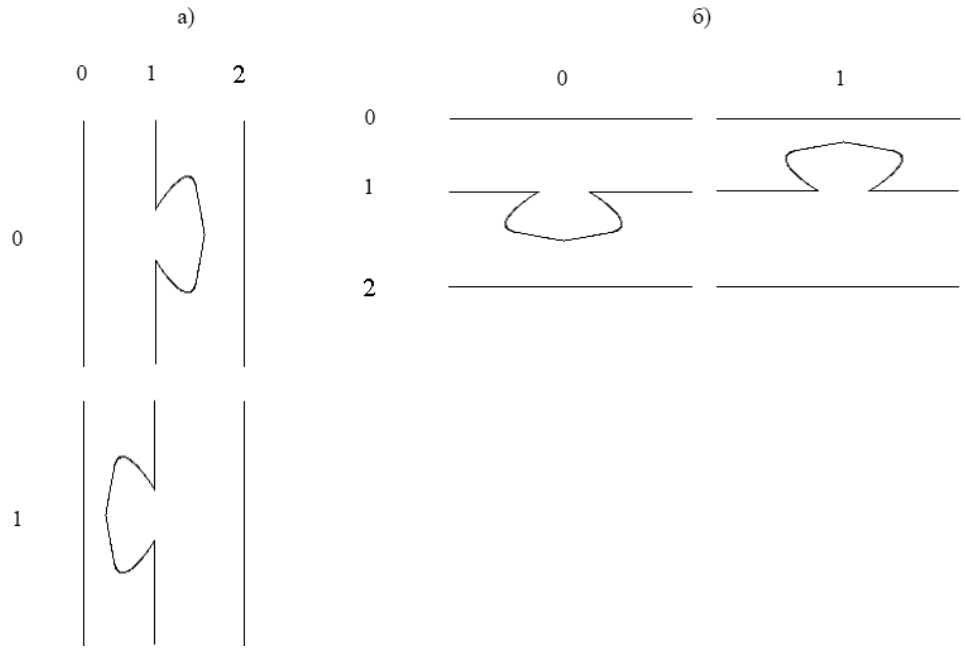


Рисунок 11 – Двумерный массив из а) вертикальных и б) горизонтальных строн для построения Фрагмента 2x2

Следует заметить, что *«внутренние» стороны являются общими* для нескольких фрагментов, при этом, сторона, являющаяся вогнутой для одного, является выгнутой для другого. На рисунке 12 показано, что сторона b является общей для первого и второго Фрагментов.

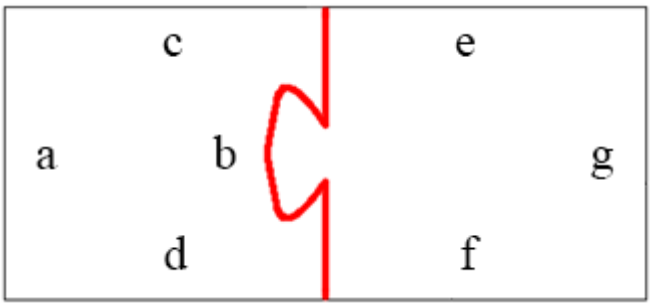


Рисунок 12 – Смежные стороны Фрагментов

После создания массивов происходит соединение сторон в Фрагменты (рисунок 13). Так, например, для построения левого Фрагмента (рисунок 12) соединяются стороны c, b, d и a.

Также, важны точки, которыми сторона присоединяется к предыдущим – стороны d и a перед добавлением в Фрагмент – отражаются относительно одного из своих концов.

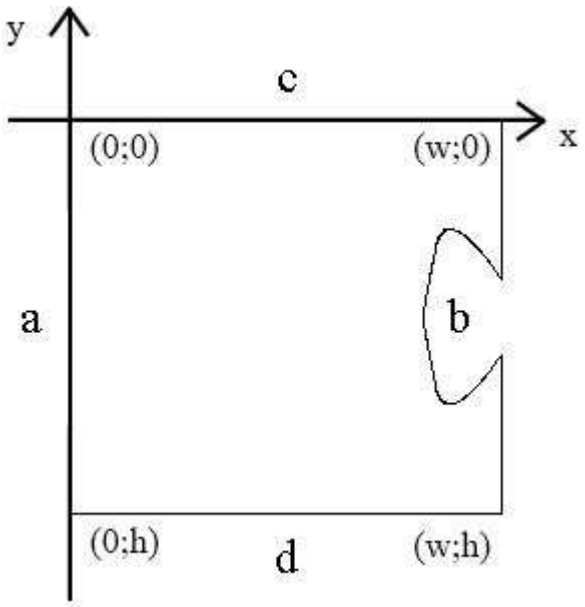


Рисунок 13 – Построение Фрагмента пазла из сторон в локальных координатах

Алгоритм осложняется и тем фактом, что точка, являющая началом стороны для одного Фрагмента будет являться концом для другого (рисунок 14.а, рисунок 14.б). Т.е. для Фрагмента 1 нижнее ребро будет начинаться в точке «d» и заканчиваться в «c». А для Фрагмента 2, верхнее ребро, являющееся нижним ребром Фрагмента 1 начинается в точке «c» и заканчивается в «d».

Зеркальное отражение отрезка cd не решит проблему, т. к. Крючок является несимметричным. Таким образом это приведет к ситуации, как показано на рисунке 14.в. Крючки буду смотреть в разные стороны.

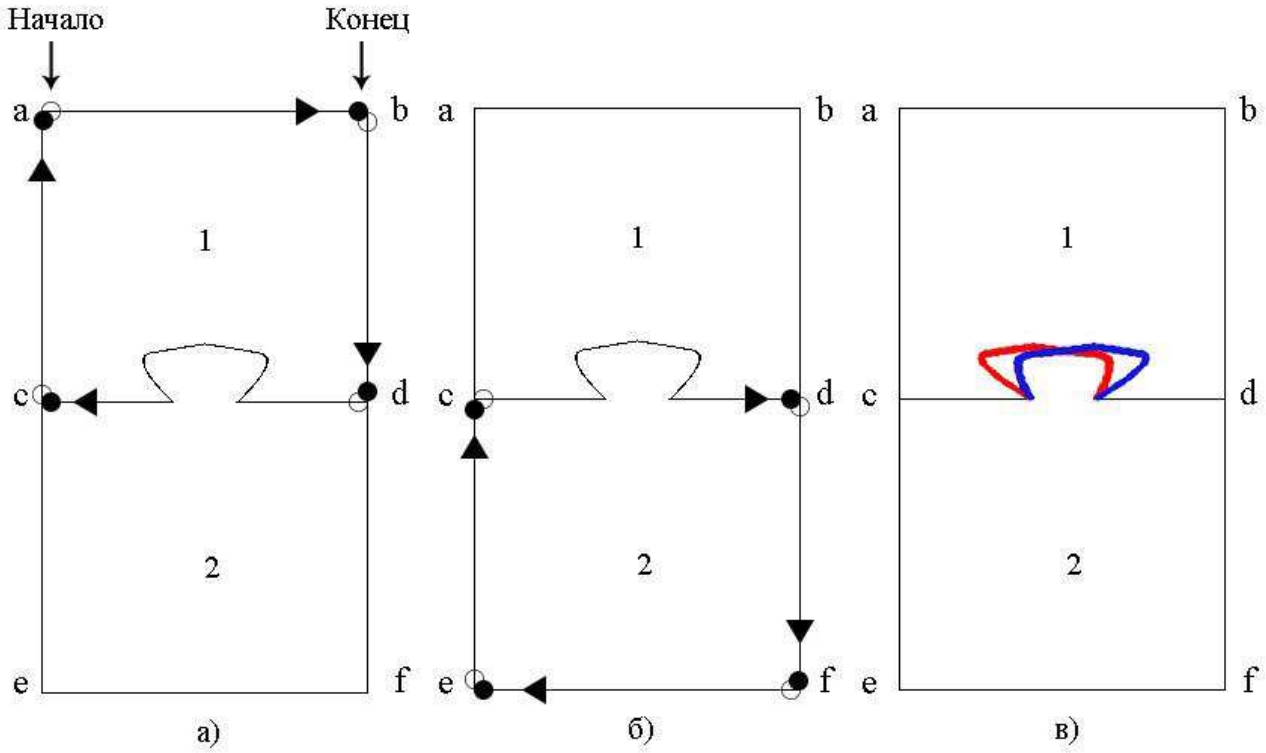


Рисунок 14 – Демонстрация соединения смежного крючка за разные концы относительно а) Фрагмента 1, б) Фрагмента 2; в) соединение за одинаковые концы

После построения Фрагмента, на него необходимо отразить соответствующую часть изображения, а также обеспечить возможность его перемещения и вращения. Реализация всех этих возможностей зависит от используемых библиотек и описана в третьей главе.

Для **формирования кривых требуемой формы** (*формирование_кривой*) используется следующий алгоритм:

Входные данные: ориентация линии (горизонтальная/вертикальная), тип линии (крючок, прямая).

Выходные данные: линия с заданными параметрами.

Последовательность действий:

1. В зависимости от сочетания типа и ориентации линии, выбирается один из заранее подготовленных наборов опорных точек (массив точек).
2. Каждая опорная точка смещается случайным образом в заданной окрестности.

3. Выполняется интерполяция массива опорных точек для получения промежуточных значений (обеспечения «плавности» формы кривой).

Для **построения Фрагмента пазла**, используется следующий алгоритм:

Входные данные: картинка и количество разбиений, выбранные пользователем.

Выходные данные: матрица Фрагментов пазла.

Последовательность действий:

1. С помощью функции *формирование_кривой* заполняются двумерные массивы горизонтальных и вертикальных сторон (структура данных показана на рисунке 11).
2. Заполняется матрица Фрагметов пазла, для этого для каждого Фрагмента выбираются соответствующие кривые из двумерных массивов и соединяются в контур, контур заливается Цветом.
3. Изображение нарезается на Прямоугольники по количеству Фрагментов, с учетом их размера (Прямоугольники могут «накладываться» друг на друга).
4. На каждый прямоугольник накладывается маска в виде закрашенного контура Фрагмента. В результате Цвет заменяется данными исходного изображения (Прямоугольника). Статическая модель системы

2.1.3 База данных

При установке приложения, информация о картинах, авторах и эпохах считывается из файла json, а затем сохраняется в БД. При этом, размещаясь по трем таблицам: pictures, authors, category. Взаимосвязь между которыми представлена на рисунке 18 в виде диаграммы.

Структура файла в формате JSON приведена на рисунках 15 - 17.

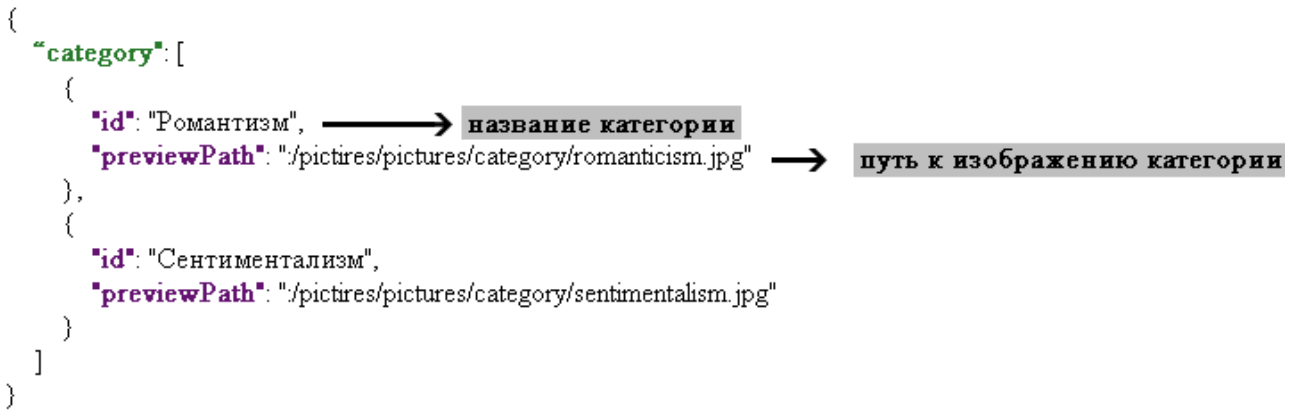


Рисунок 15 – Формат «category» в JSON – файле

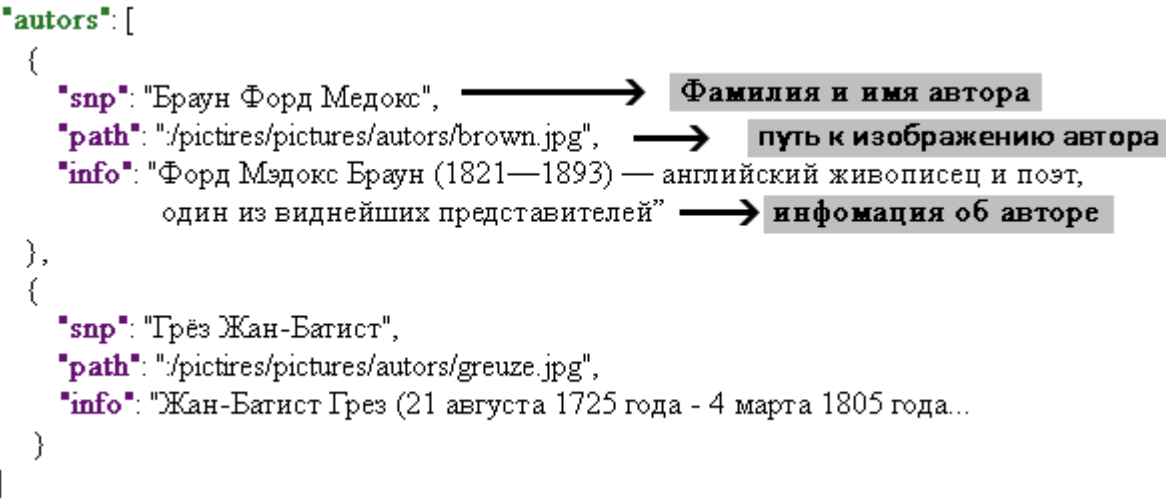


Рисунок 16 – Формат «category» в JSON – файле

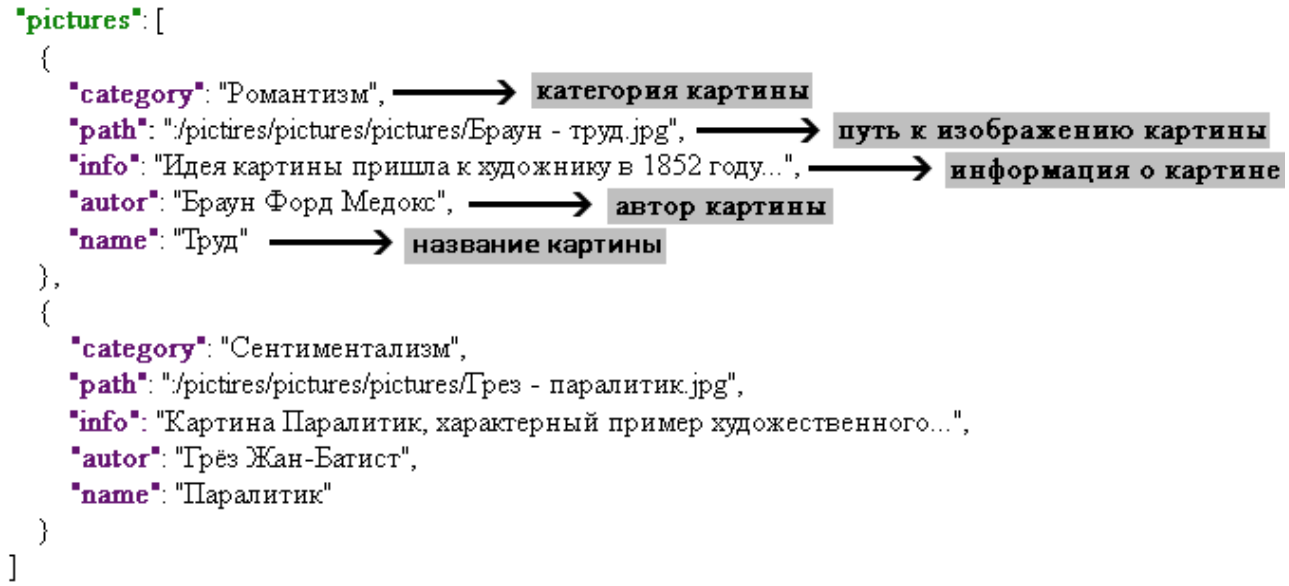


Рисунок 17 – Формат «pictures» в JSON – файле

Кроме того, для хранения и обновления настроек в игре используется таблица settings. Поле «property» задает тип свойства. Всего используется 3 разновидности свойств: «date» - дата последнего успешного обновления; «sound» - воспроизводимый звук при правильной установке Фрагмента в свою ячейку; «path» - путь к каталогам для хранения изображений. При изменении значения одного из свойств, оно перезаписывается в БД.

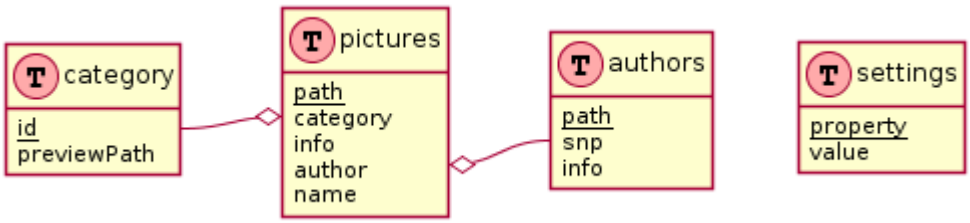


Рисунок 18 – Структура базы данных

2.1.4 Диаграммы классов

Исходное изображение нарезается на Фрагменты в соответствии с описанным выше алгоритмом. Фрагменты умеют перемещаться, привязываться к своей ячейки на игровом поле, подсвечиваться и вращаться. Все эти функции реализуются в классах, представленных на рисунке 19.

Игровое поле представляет собой сетку из ячеек, куда устанавливаются правильно расположенные Фрагменты. Ячейка - это прямоугольник, задающий «точки привязки» для Фрагмента (рисунок 20).

Каждый класс - наследник реализует свою часть функционала при этом обладает свойствами класса – родителя:

– Базовым для всех является PuzzleItem, в задачи которого является отрисовка, связанного с ним Фрагмента, а также контура черного цвета вокруг него.
– SelectableItem обрабатывает события нажатия и отпускания мыши. Рисует вокруг Фрагмента красную рамку двойной толщины, если объект «выбран», то есть имело место событие нажатия мыши, но не

было поступило отпускания. Изменяет слой объекта так, чтобы выделенный объект находился поверх всех остальных.

– Rotatable обрабатывает события нажатия, отпускания и перемещения мыши. Выполняет поворот Фрагмента в момент отпускания мыши, если не имело место перетаскивание Фрагмента более чем на 15 пикселей.
– MovableItem обрабатывает событие нажатие и перемещение мыши для реализации перетаскивания Фрагмента.
– SettableItem - при отпускании мыши, проверяет корректность установки текущей позиции элемента, в соответствии с необходимым положением в пазле. Если разница позиции меньше 15 пикселей, то элемент устанавливается в правильную позицию, после чего все поступающие события мыши для этого игнорируется.

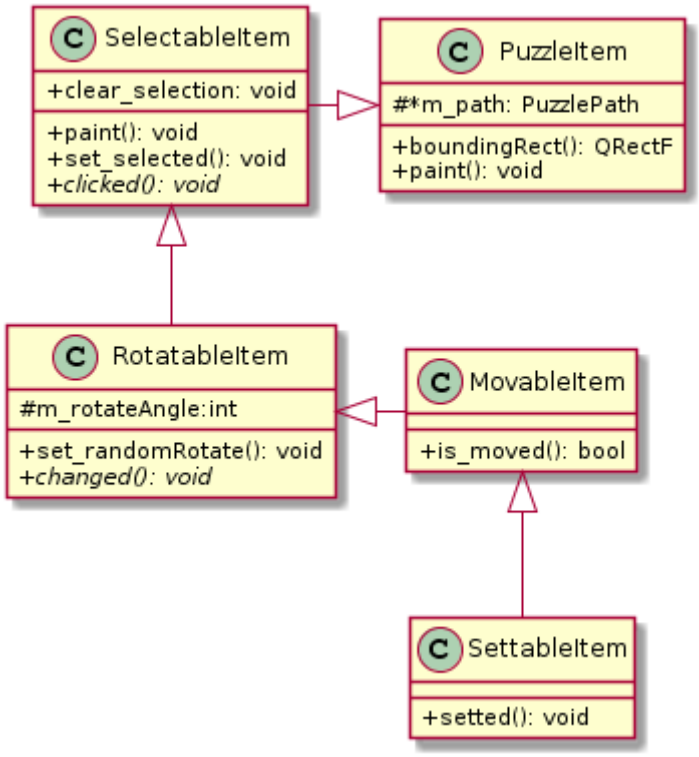


Рисунок 19 - Диаграмма классов Фрагмента пазла

Как показано на рисунке 20, PuzzleGame добавляет на игровое поле сетку и Фрагмент (SettableItem). Кроме того, устанавливает для них начальные номера слоев, так чтобы Фрагменты пазла находились поверх сетки. Также он

обрабатывает сигнал установки Фрагмента в свою ячейку и после сбора пазла вырабатывает сигнал о завершении игры.

Сетка в виде ячеек прямоугольной формы реализована с использованием класса ItemGround. Масштабирование содержимого игры с помощью жестов пальцами осуществляется при помощи PuzzleView.

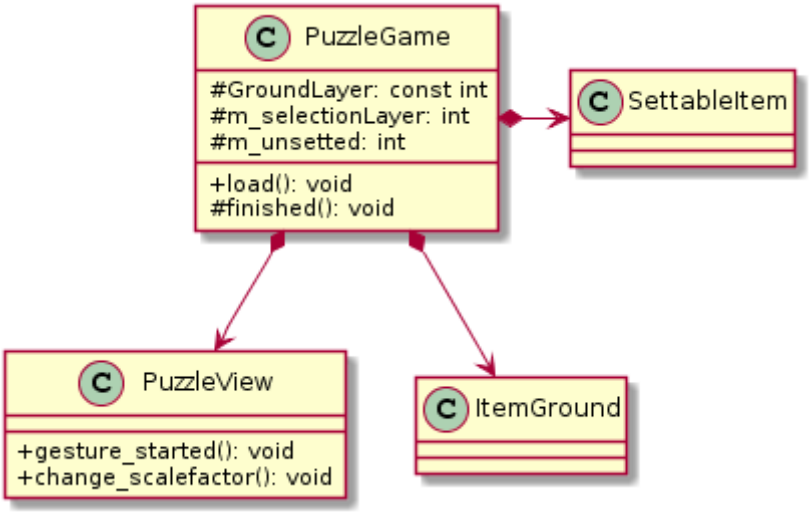


Рисунок 20 – Диаграмма классов игрового поля

Главное меню (рисунок 21) реализует класс MainMenuWidget, который представляет собой виджет с кнопками для перехода к другим окнам приложения. IconsMenuWidget - виджет для отображения различных реализаций InnerMenuVIew.

LevelModeWidget содержит кнопки установки сложности сбора пазла. Включает класс PickedButton - кнопку - переключатель, находящуюся в состоянии (выбрано/не выбрано) для поворота Фрагмента и ModeButton – кнопку выбора режима игры (установки количества разбиений).

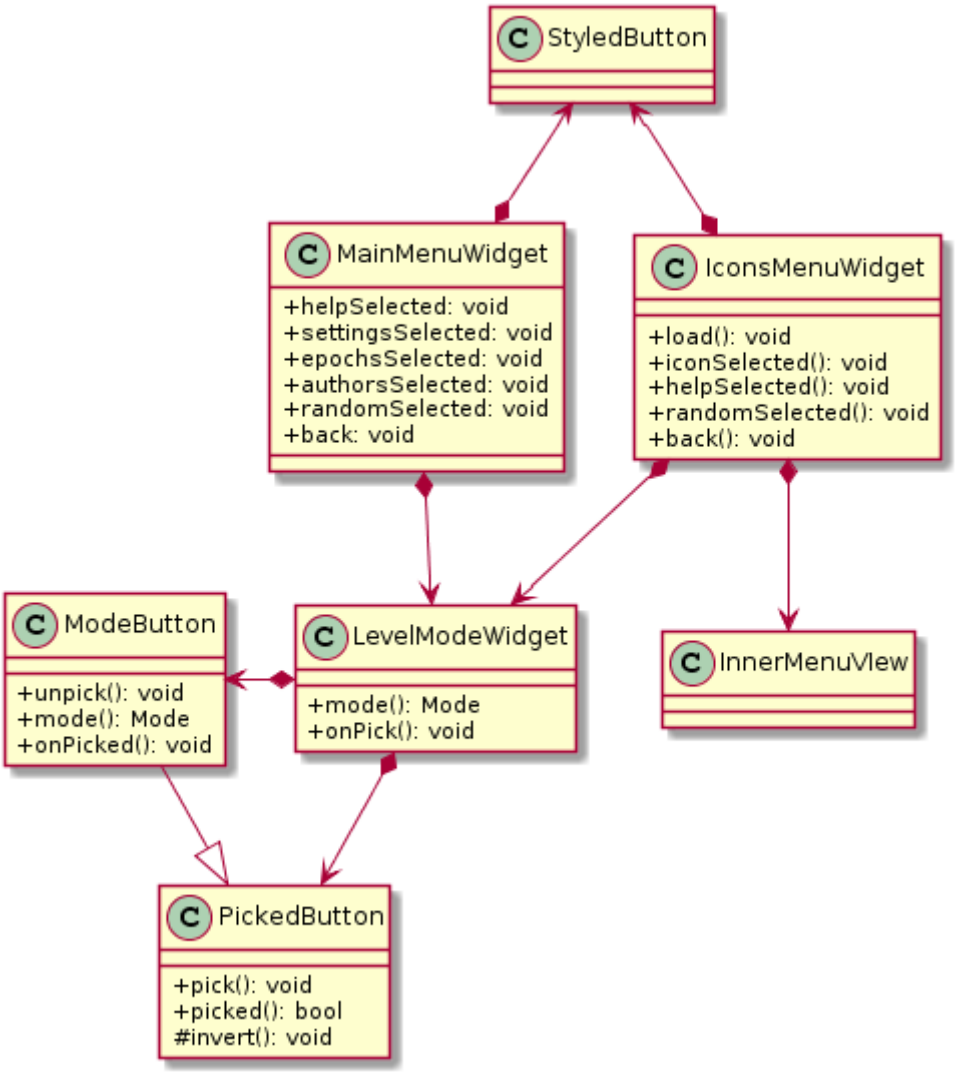


Рисунок 21 – Диаграмма классов меню

Иконки для отображения эпох, авторов и картин реализованы в классах, показанных на рисунке 22. InnerMenuVIew реализует отображение на сцене иконок уровней. От него наследуются три класса: AutorMenuView, CategoryMenuView и PictureMenuView с перегруженным методом load, отвечающий за загрузку своего типа картинок. TaskIcon реализует отображение иконки с уменьшенным изображением.

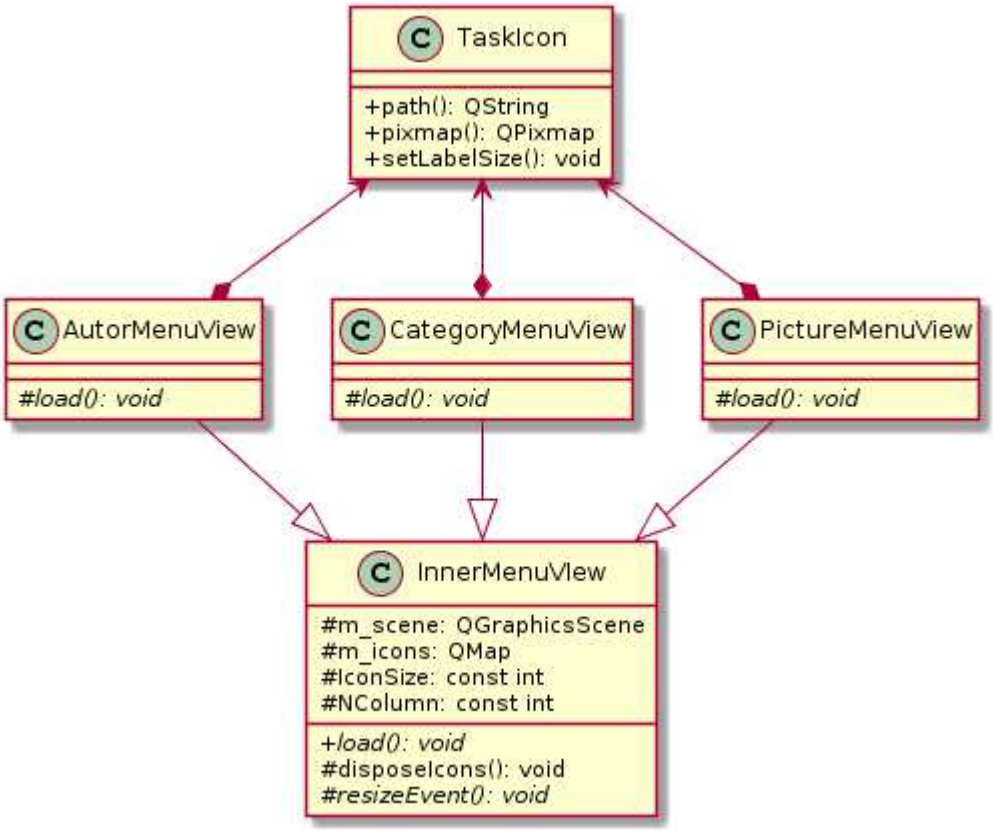


Рисунок 22 – Диаграмма классов иконок

Для изоляции логики приложения от логики взаимодействия с БД целесообразно использовать шаблон проектирования фасад [14]. Фасад БД должен представлять и реализовывать удобный интерфейс, обеспечивающий потребности приложения.

Несмотря на то, что применяется одна БД, для разделения логики между таблицами созданы два объекта – фасада SettingsDBFacade и LevelsDBFacade, реализующих создание таблиц и обеспечивающих удобный интерфейс для работы с информацией БД. Соответствующая диаграмма классов представлена на рисунке 23.

Класс DBFacade обеспечивает фунционал выполнения запросов и обработки ошибок.

LevelsDBFacade осуществляет создание трех таблиц: эпохи, авторы, картины; загрузку данных из файла json; занесение данных в таблицы БД; запросы для получения необходимых данных.

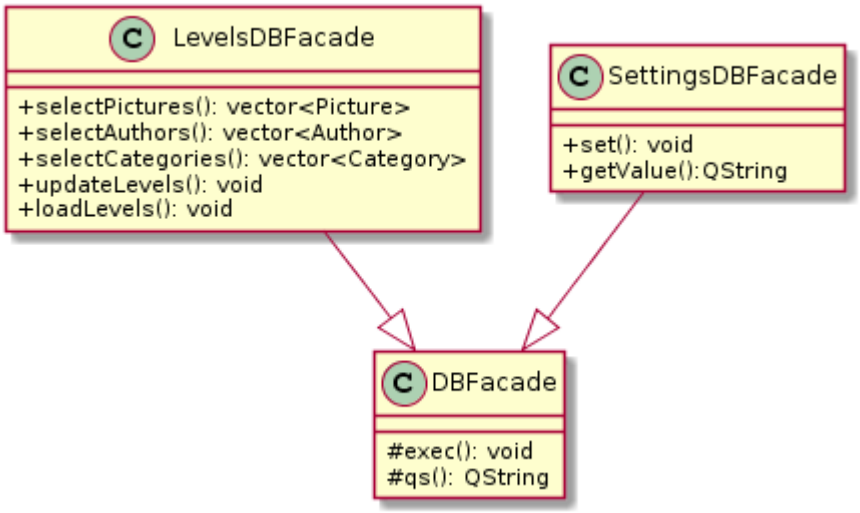


Рисунок 23 – Диаграмма классов фасадов базы данных

2.2 Выводы по главе

В соответствии с техническим заданием:

- **Предложена** архитектура системы и структура базы данных.
- С помощью нотации диаграмм последовательностей **провизуализированы** наиболее сложные отношения в системе, более детально проработано взаимодействие объектов.
- На основе анализа диаграмм последовательностей разработаны диаграммы классов, описывающие статическую модель системы.
- **Разработаны** структуры данных и алгоритм для формирования элементов пазла требуемой формы.

3 Реализация и документация

3.1 Реализация

3.1.1 Игровой процесс

В процессе игры пользователь перемещает фрагменты по сцене, при этом поддерживается возможность ее прокрутки сцены и масштабирования. Для реализации такого поведения целесообразно использовать графический фреймворк Qt, реализованный классами *QGraphicsScene, QGraphicsView, QGraphicsItem* и некоторыми другими. Класс сцены представляет собой модель, данными которой являются Фрагменты (наследуют *QGraphicsItem*), вид отвечает за отображение заданной части сцены.

При формировании стороны Фрагмента, необходимой формы, в соответствии с алгоритмом, приведенным в разделе 2.1.2, применяется интерполяция, реализуемая стандартной функцией Qt – *cubicTo* (кубическая интерполяция на основе кривых Безье).

Массивы точек, представляющих кривые до момента формирования контура, задаются как *std::vector<QPoint>* – это, в частности, обеспечивает возможность соединения между собой крючков за разные концы. Необходимость соединения сторон за разные концы проиллюстрирована на рисунке 24 – при добавлении в контур Фигуры 1, кривая, выделенная жирным, присоединяется справа-налево (точки 7-1), а при добавлении в контур Фигуры 2 – слева-направо (точки 1-7). Порядок присоединения влияет на заливку Фигуры.

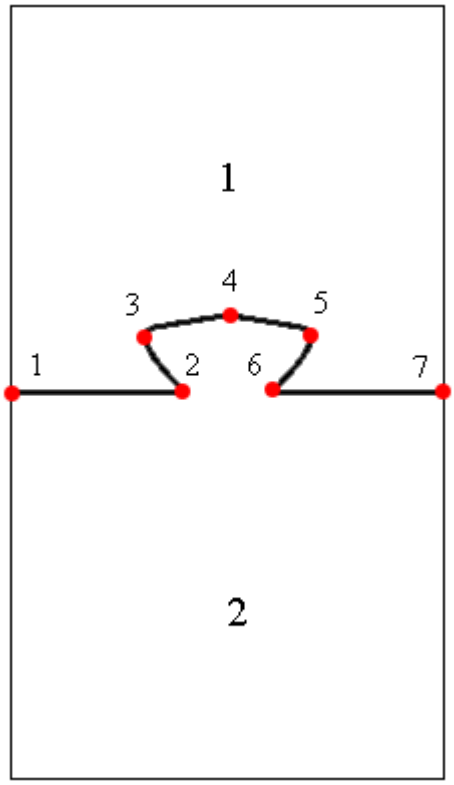


Рисунок 24 – Порядок точек для общей стороны

С использованием метода QPainterPath::connectPath стороны последовательно соединяются друг к другу. Таким образом, формируется контур Фрагмента. Контур описывается с помощью стандартного класса *QPainterPath.*

Для вырезания прямоугольной части изображения используется метод *QPixmap::copy*, для закрашивания контура – QImage::fill, для наложения маски – QPixmap::setMask.

Для задания точек привязки Фрагментов, то есть отображения их корректных позиций, на игровое поле добавляется сетка, состоящая из ячеек, реализованных с помощью класса QGraphicsRectItem. Для проверки правильности позиционирования Фрагмента на сетке, формируется двумерный массив корректных координат расположения Фрагментов.

Фрагменты при инициализации разбрасываются по полю в хаотичном порядке в определённой окрестности, которая рассчитывается в зависимости от размеров исходного изображения. Если пользователь поставит дополнительное усложнение в виде поворота, то Фрагменты будут иметь разные углы наклона (0,

90, 180 градусов). Поворот фрагментов осуществляется функциями QTransform::rotate (вращение относительно верхнего левого угла) и QTransform::translate (перемещение).

3.1.2 Меню

Приложение содержит несколько окон, поэтому необходимо осуществлять переход между ними. В соответствии с диаграммой 7, логика переключения экранов инкапсулирована в классе ScreenController, который хранит указатели на окна, подписывается на их события и, при необходимости, выполняет переключение.

ScreenController является наследником класса ScreensStack, взятого из открытого репозитория [14], реализующего возможность возврата на предыдущий экран. ScreensStack организует виджеты в стек так, что видимым является виджет, находящийся на вершине, имеется возможность добавления виджета на вершину или его удаления из нее.

3.1.3 Справка

В приложение встроена справка. Текст справки для ее структурированного отображения целесообразно использовать html теги. Для обеспечения ее адаптивности, то есть возможности удобного ее просмотра на устройствах с различным размером экрана, необходимо реализовать возможность прокрутки содержимого справки жестами пальцев. Такое поведение реализовано в классе TutorialViewier, взятом из открытого репозитория [15]. Этот класс представляет собой виджет, размещаемый внутри окна справки.

3.1.4 База данных

Данные приложения (настройки и информация об уровнях) хранятся в базе данных SQLite. В соответствии с разработанным проектом, используется фасад базы данных, который реализуется посредством стандартных классов библиотеки Qt, как показано на рисунке 25.

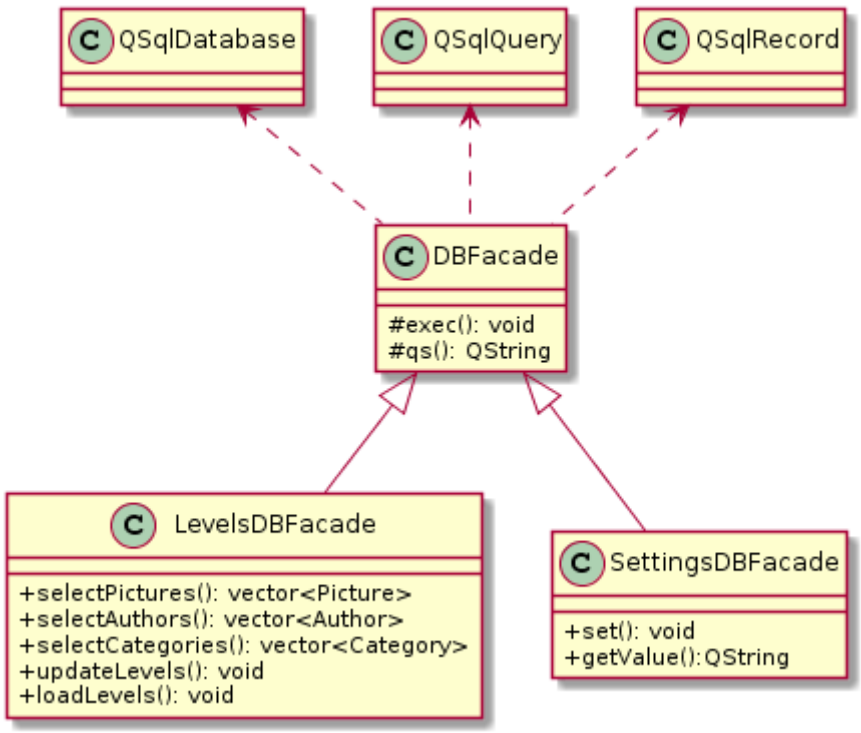


Рисунок 25 – Фрагмент диаграммы классов эскизирующая работу с БД

LevelDBFacade, помимо функций работы с данными в базе, предоставляет функцию loadLevels, выполняющую начальное заполнение базы на основе JSON файла, находящегося в ресурсах приложения [16].

SettingsDBFacade оперирует единственной таблицей для хранения и получения настроек приложения.

Фасад использует несколько таблиц, для установки текущей таблицы (с которой будет работать модель), применяется метод QSqlTableModel::settable.

3.1.5 Интерфейс

Скругление углов, изменение цвета и других визуальных свойств приложения, осуществляется за счет использования таблиц стилей Qt Style Sheets (QSS) [17].

Для выравнивая компонентов и масштабирования использован класс QGridLayout, позволяющий размещать вложенные виджеты по сетке [18].

Для отображения красивых шрифтов используется класс QFontDatabase, который позволяет загрузить шрифт из файла и сделать его доступным приложению [19]. Для того, чтобы файл шрифтов был всегда доступен, он добавляется в качестве ресурса [16].

3.2 Инструкция разработчика

Настройка окружения программиста для разработки приложений с использованием Qt под платформу Android осуществляется нетривиально. Однако, доставку окружения можно обеспечить средствами виртуализации.

В рамках работы предпринималась попытка использования трех готовых докер контейнеров [20, 21, 22]. Ни один из них корректно развернуть не удалось.

Поэтому, была настроена виртуальна машина на базе VMware, содержащая необходимый комплект разработки на базе операционной системы Ubuntu-18.04.2 x64.

Полученный образ [23, 24] позволяет выполнить сборку проектов, а также их отладку с использованием эмулятора и реального устройства. Для получения доступа к ссылкам необходимо иметь ключи дешифровки: f jz7ZQLN2xeO7gKw24hLjWQ, cw_Ma32lOnLoEw-5FHOdcA.

В приложении А описан порядок установки Qt для платформы Android, на случай, если аналогичный процесс будет необходимо повторить для другой операционной системы.

3.3 Тестирование

Тестирование приложения осуществлялось с помощью стандартных возможностей Google Play – была выпущена бета версия приложения, представленная на конференции (приложение В) с целью получения информации о возникающих ошибках и проверки работоспособности на разных версиях Android - устройств.

Для тестирования обновления базы данных приложения с сервера, был развернут локальный http-сервер Apache [25]. После чего по пути «~\Apache24\htdocs» размещены файл с «индексом» и изображения, доступные для загрузки. При этом, внутри файла «индекса» пути расположения картинок содержат адрес локального сервера «http://localhost/».

3.4 Выводы по главе

1. Реализовано приложение «Пазлы-Живопись».
2. Составлена инструкция по настройке окружения разработчика под платформу Android с использованием библиотеки Qt.
3. Создан образ виртуальной машины для ОС Ubuntu-18.04.2 с установленным окружением Qt-Android.
4. Проведено тестирование «Пазлы-Живопись» с использованием возможностей Google play, а также модуля обновления с применением локального http-сервера.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы:

1. Спроектировано, реализовано и протестировано приложение «Пазлы-Живопись» [26].
2. Создан образ с установленным и настроенным комплектом разработчика под платформу Android с использованием Qt, который будет использоваться в курсе «Организации процесса проектирования программного обеспечения».

В разработанном приложении присутствуют недостатки, исправить их можно путем пополнения функциональных возможностей:

- Возможность выбора определенных изображений с помощью checkbox, а не загрузка всего каталога.
- Отображение процесса загрузки каждого изображения с помощью progress bar.
- Возможность загрузки собственных изображений и последующий сбор пазлов на их основе.
- Отображение статистики собранных изображений по уровням сложности.
- Возможность проходить тестирование по выбранным темам.

Исходный код приложения доступен для скачивания с git-репозитория [27].

СПИСОК СОКРАЩЕНИЙ

БД – база данных

ВКР – выпускная квалификационная работа

ОС – операционная система

СГИИ - Сибирский государственный институт искусств

ТЗ – техническое задание

UML (Unified Modeling Language) - унифицированный язык моделирования

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Google Play [Электронный ресурс]: – Режим доступа: https://play.google.com/store

2. Процесс разработки программного обеспечения ICONIX [Электронный ресурс]: – Режим доступа: https://pro-prof.com/archives/4126

3. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч, Р. А. Максимчук, М. У. Энгл, Б. Д. Янг, Д. Коналлен., К. А. Хьюстон – Калифорния, 2010. – 720 с.

4. Plant UML [Электронный ресурс]: – Режим доступа: http://plantuml.com

5. Фаулер, М. UML. Основы / М. Фаулер, К. Скотт; пер. с англ. – СПб.: Символ – Плюс, 2002. – 192 с

6. JigSawPuzzle2-Android [Электронный ресурс]: – Режим доступа: https://github.com/yuvaraj119/JigSawPuzzle2-Android

7. Canvas Jigsaw [Электронный ресурс]: – Режим доступа: https://github.com/andysellick/jigsaw

8. JigsawPuzzle [Электронный ресурс]: – Режим доступа: https://github.com/artursDerkintis/JigsawPuzzle

9. Tetzle [Электронный ресурс]: – Режим доступа: https://github.com/gottcode/tetzle

10. Tetzle [Электронный ресурс]: – Режим доступа: https://github.com/cryer/puzzle

11. 2ГИС: справочник и навигатор 4.3.4.2360 Для Андроид [Электронный ресурс]: – Режим доступа: https://one-android.ru/2gis-spravochnik-i-navigator.html

12. android-SDKs-for-tiles [Электронный ресурс]: – Режим доступа: https://github.com/2gis/android-SDKs-for-tiles

13. Telegram Messenger [Электронный ресурс]: – Режим доступа: https://download-telegram.ru/telegram-na-telefon

14. Использование БД SQL. Шаблон проектирования «Фасад» (Facade) [Электронный ресурс]: – Режим доступа: https://pro-prof.com/archives/882

15. Вывод справки в html формате [Электронный ресурс]: – Режим доступа: https://pro-prof.com/forums/topic/qt-html-tutorial

16. The Qt Resource System [Электронный ресурс]: – Режим доступа: https://doc.qt.io/Qt-5/resources.html

17. Разработка игры на C++, Qt [Электронный ресурс]: – Режим доступа: https://pro-prof.com/archives/1520

18. Создание собственных виджетов Qt. Сигналы, слоты и события. [Электронный ресурс]: – Режим доступа: https://pro-prof.com/archives/619

19. QFontDatabase Class [Электронный ресурс]: – Режим доступа: https://doc.qt.io/archives/qt-4.8/QFontDatabase.html#details

20. Compiling Qt Android app with Docker [Электронный ресурс]: – Режим доступа: https://blog.lasconic.com/compiling-qt-android-app-with-docker/

21. qt-android-docker [Электронный ресурс]: – Режим доступа: https://github.com/winie/qt-android-docker

22. frankwolf/qt-android [Электронный ресурс]: – Режим доступа: https://hub.docker.com/r/frankwolf/qt-android

23. vmUbnt [Электронный ресурс]: – Режим доступа: https://mega.nz/#F!mMFA0KIC

24. vmUbnt [Электронный ресурс]: – Режим доступа: https://mega.nz/#F!u2R3CCoL

25. Apache [Электронный ресурс]: – Режим доступа: https://apache.org

26. Пазлы [Электронный ресурс]: – Режим доступа: https://play.google.com/store/apps/details?id=org.qtproject.puzzle

27. Bitbucket - репозиторий проекта [Электронный ресурс]: Андроид - приложение «Пазл» – Режим доступа: https://chubenry@bitbucket.org/chubenry/puzzle.git

28. Java SE Downloads [Электронный ресурс]: – Режим доступа: https://www.oracle.com/technetwork/java/javase/downloads/index.html

29. Android Studio [Электронный ресурс]: – Режим доступа: https://developer.android.com/studio

30. Download Qt [Электронный ресурс]: – Режим доступа: http://www.qt.io/download-open-source

31. Java SE Development Kit 8 Downloads [Электронный ресурс]: -Режим доступа: https://www.oracle.com/technetwork/java/javase/downloads/2133151

ПРИЛОЖЕНИЕ А

Установка QT для платформы Android

1. Скачать компонент JDK по ссылке [28].
2. Скачать Android Studio по ссылке [29].
3. В Android Studio с помощью SDK Manager загрузить NDK (Configure -> SDK Manager -> SDK Tools -> NDK -> Apply).
4. Скачать установочный файл Qt актуальной версии с официального сайта [30].
5. Запустить файл (в случае Ubuntu) с помощью команд:
   sudo chmod +x «имя».run
   ./«имя».run
6. Следовать инструкциям установки, кроме пункта возможности выбора загружаемых компонентов. В Windows кроме галочек по умолчанию выделить «Android» и «MinGW» (рисунок 1, рисунок 2). А в Ubuntu согласно рисунку 3.
7. Зайти в окно настроек Qt, во вкладку Devices -> Android (рисунок 4). В разделе Android JDK Location выбрать папку, где установлены JDK библиотеки. В разделе Android SDK Location выбрать папку, где установлены SDK библиотеки. В разделе Android NDK Location выбрать папку, где установлены NDK библиотеки.
8. Если использовать компоненты от Android Studio, то их расположение можно посмотреть в SDK Manager.
9. Если после указания всех путей в окне настройках «Devises» напротив поля «Платформа SDK установлена» будет крестик, необходимо выбрать версию JDK 8 (Java SE Development Kit 8u201) [31].
10. Посмотреть наличие репозитория в MaintenanceTool. Для этого перейти в папку, где установлен Qt, запустить MaintenanceTool -> Настройки -> Репозитории. Если репозиториев нет, то добавить во вкладке

«Использовать назначенные репозитории». Для Ubuntu: https://download.qt.io/online/qt5/linux/x64/online_repository/.

11. Если возникают какие - либо ошибки необходимо использовать следующие команды:

    sudo apt-get install gcc

    sudo apt-get install g++

    sudo apt-get install clang

    sudo apt-get install libgl1-mesa-dev

    sudo apt-get install build-essential

12. Перед компиляцией проекта перейти в «Project» убедиться, что выбран «Выпуск» Android. Нажать на «Create Templates», указать минимальную версию api для сборки.
13. Запустить проект. Если устройство подключено, но заблокирована возможность компиляции, необходимо доустановить в «SDK Manager» соответствующую версию Platforms.
14. При создании виртуального устройства установить все Packages версии (рисунок 5).

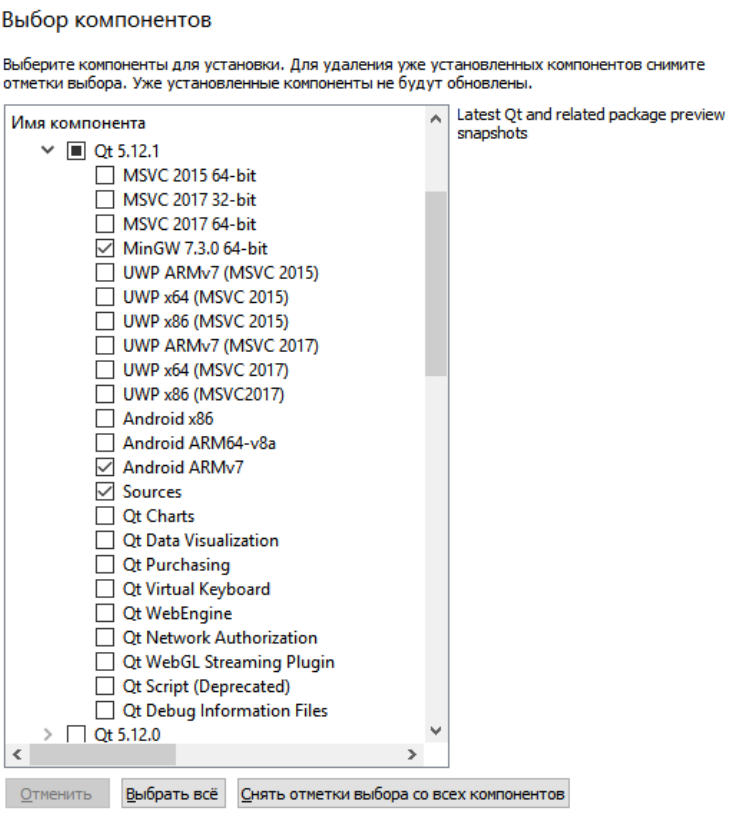


Рисунок А.1 – Выбор компонентов для Windows

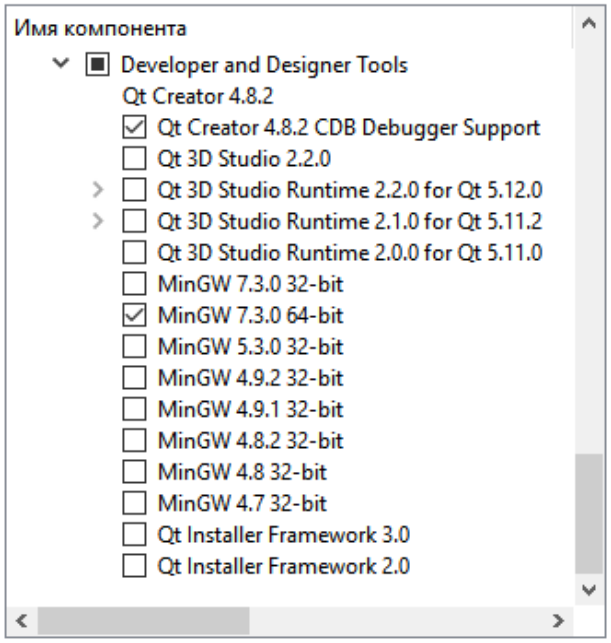


Рисунок А.2 – Выбор компонентов для Windows

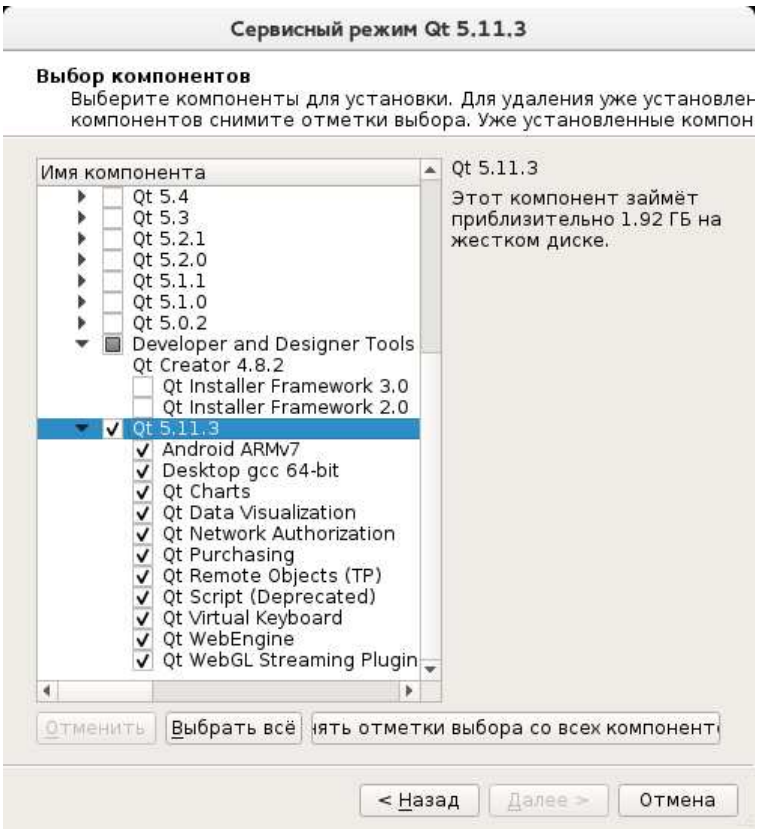


Рисунок А.3 – Выбор загружаемых компонентов в Ubuntu

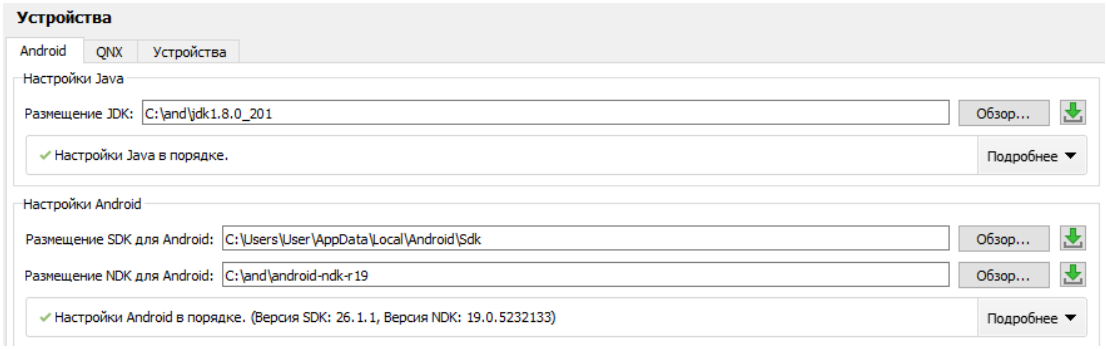


Рисунок А.4 – Настройка путей к компонентам для android на ОС Windows

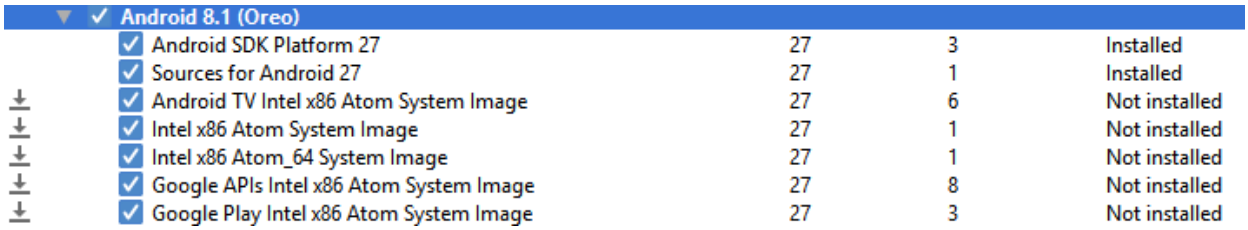
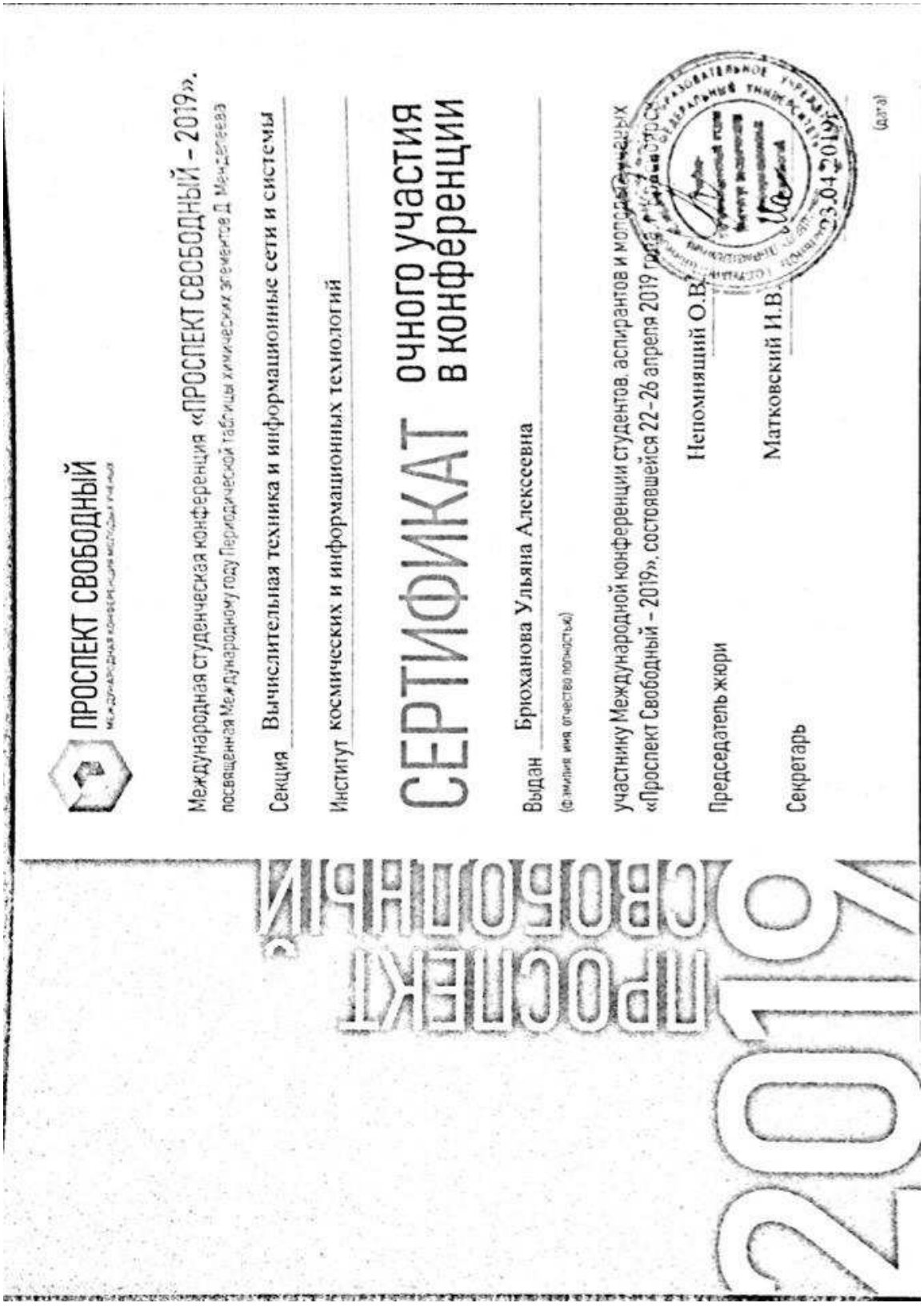


Рисунок А.5 – Packages для Android 8.1 (Oreo)

ПРИЛОЖЕНИЕ Б

Сертификат участника в конференции



Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Вычислительная техника

УТВЕРЖДАЮ
Заведующий кафедрой ВТ
_____ О.В. Непомнящий
подпись
«26» июня 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Android-приложение для запоминания картин в игровой форме

Руководитель	Васильев 15.6.19 подпись, дата	ст. преподаватель	В.С. Васильев
Выпускник	Брюханова 15.06.2019 подпись, дата		У.А. Брюханова
Консультант	15.06.2019 подпись, дата	доцент, канд. техн. наук	Л. И. Покидышева
Нормоконтролер	15.06.19 подпись, дата	доцент, канд. техн. наук	В. И. Иванов

Красноярск 2019