

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 инициалы, фамилия
« _____ » _____ 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Android-приложение «Картинки из геометрических фигур»

Руководитель	_____	ст. преподаватель	В.С. Васильев
	подпись, дата		
Выпускник	_____		О.В. Волков
	подпись, дата		
Нормконтролер	_____	ст. преподаватель	В.С. Васильев
	подпись, дата		

Красноярск 2021

СОДЕРЖАНИЕ

1 Анализ задания.....	6
1.1 Анализ существующих аналогов игры.....	6
1.2 Разработка спецификации требований.....	8
1.2.1 Интерфейс программы.....	8
1.2.2 Функциональные требования.....	9
1.2.3 Формат файла уровня.....	11
1.3 Выводы по главе.....	12
2 Проектирование.....	14
2.1 Экран меню.....	15
2.1.1 Работа с базой данных.....	15
2.1.2 Динамическая модель.....	16
2.1.3 Статическая модель.....	17
2.2 Игровой экран.....	18
2.2.1 Алгоритм инициализации фигур.....	18
2.2.2 Алгоритм передвижения фигур.....	19
2.2.3 Алгоритм правильного положения фигур.....	19
2.2.4 Динамическая модель.....	20
2.2.5 Статическая модель.....	21
2.3 База данных.....	23
2.4 Выводы по главе.....	24
3 Реализация и тестирование.....	25
3.1 Реализация.....	25
3.1.1 Используемые инструменты.....	25
3.1.2 База данных.....	25
3.1.3 Работа с сервером.....	25
3.2 Инструкция разработчика.....	26
3.3 Инструкция пользователя.....	27
3.4 Тестирование.....	28
3.5 Выводы по главе.....	28

ВВЕДЕНИЕ

Классическая игра «Картинки из геометрических фигур» предназначена для детей дошкольного возраста (3-6 лет) и направлена на развитие умения различать геометрические фигуры, анализа положения предметов в пространстве, умения сравнивать геометрические фигуры по размеру, развитие внимания и мышления [1]. Игра заключается в том, что ребенку демонстрируется картинка, составленная из геометрических фигур, и выдается набор таких же фигур. Задача ребенка — составить аналогичную картинку, расположив фигуры в правильном порядке.

Существует множество вариантов игры — фигуры могут быть вырезаны из бумаги или ткани, могут отличаться по цвету, на холсте для их закрепления могут присутствовать метки (или пуговицы). Помимо этого, существуют компьютерные реализации этой же игры, каждая из которых имеет свои особенности.

Целью работы является создание базового варианты игры «Картинки из геометрических фигур» для платформы Android, предусматривающего возможность дальнейшей доработки (реализации новых режимов игры).

Задачами настоящей работы являются:

- анализ аналогов разрабатываемой игры;
- определение функциональных требований для приложения;
- спроектировать и разработать приложение, разместить приложение на сервере Google Play, выполнить тестирование;
- разработать инструкции пользователя и программиста.

1 Анализ задания

В соответствии с заданием на ВКР необходимо разработать мобильное приложение, которое реализует классическую игру «Картинки из геометрических фигур». Игра должна поддерживать несколько режимов. Для обеспечения конкурентоспособности разрабатываемого приложения, выполнен анализ сильных и слабых сторон аналогичных приложений. С их учетом составлено техническое задание на разработку.

1.1 Анализ существующих аналогов игры

В магазине Google Play по запросу «Картинки из геометрических фигур» найдено 247 приложений. Большая часть приложений не подходит под задачу ВКР. Из 247 приложений найденных на Google Play выбрано 3, которые подходят под задачи ВКР.

Критериями анализа является:

- оценка пользователей;
- число скачиваний;
- платное/бесплатное;
- наличие раздела статистики;
- наличие разных режимов игры.

Название	Оценка (0-5)	Число скачиваний (тысяч)	Платное/бесплатное	Раздел статистики	Разные режимы игры
Танграм для детей	3.3	100	бесплатное	отсутствует	отсутствует
Poly Shape	4.8	1	платное	отсутствует	отсутствует
Tangram king	4.5	100	бесплатное	присутствует	присутствует

Таблица 1 – Основные показатели приложений

- «Танграм для детей» [2]. Приложение имеет форму, в которую нужно подставлять фигуры. Данная особенность делает невозможным сбор картинки в произвольном месте экрана. Приложение содержит окно выбора уровня и окно игрового процесса. В приложении отсутствует раздел статистики. Основным недостатком данного приложения заключается в том, что приложение позволяет составлять фигуры только из простых многоугольников. Так же содержит сложное управление, на это указывает большое количество негативных отзывов об управлении.
- «Poly Shape» [3]. Приложение так же имеет особенность в виде заднего фона в виде фигуры, которую нужно собирать. Картинка собирается как из простых и сложных многоугольников. Фигуры имеют функцию поворота, а также присутствует погрешность установки, что добавляет удобство управлению. Приложение содержит удобное окно выбора уровня. В игре отсутствует раздел статистики. Так же присутствуют внутриигровые покупки, которые ограничивают количество уровней.
- «Tangram king» [4]. В приложение присутствует раздел статистики, который привязан к серверу Google Play, статистика позволяет соревноваться с другими пользователями. Так же в игре реализовано большое количество режимов. Но не все доступны в бесплатной версии. Приложение позволяет составлять фигуры только из простых многоугольников. Главным недостатком приложения является отсутствие погрешности установки фигуры.

По результатам просмотра аналогов, принято решение добавить раздел с выбором режима игры, и отображение прогресса пройденных уровней. Так же добавить возможность установки фигуры с погрешностью и возможность поворота для фигур.

1.2 Разработка спецификации требований

1.2.1 Интерфейс программы

В результате анализа существующих реализаций — решено разделить интерфейс приложения на набор диалоговых окон, карта переходов между окнами приведена на рисунке 1.

Окно «Главное меню» содержит кнопки для начала игры, выхода и отключения звука. «Выбор режима» содержит список режимов. «Выбор категории» содержит список всех категорий уровней для выбранного режима. «Выбор уровня» содержит список уровней и прогресс прохождения. После выбора уровня запустится «Игровое поле» с фигурами, соответствующими выбранному уровню.

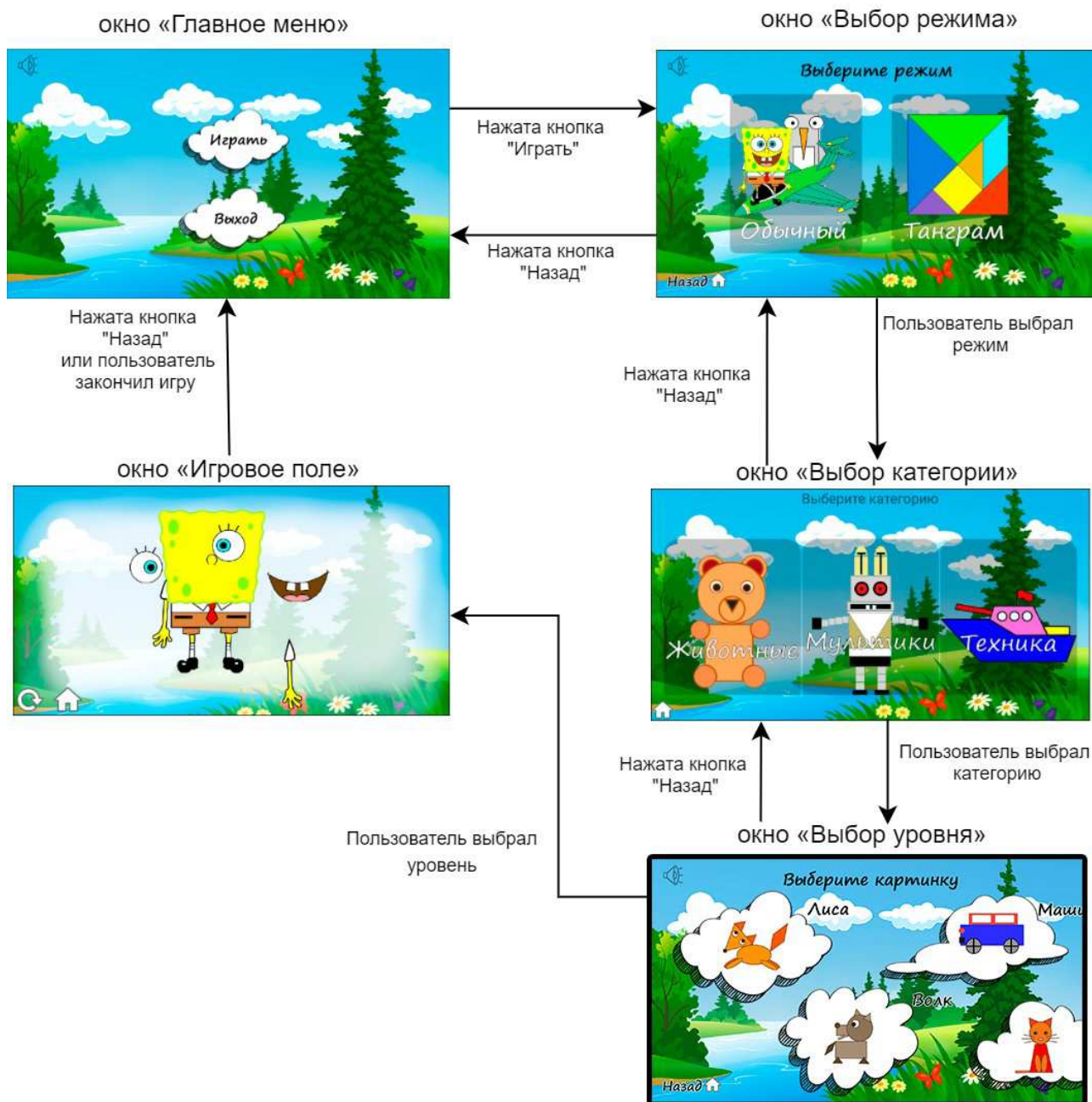


Рисунок 1 - Диаграмма потока экранов приложения

1.2.2 Функциональные требования

На основе анализа аналогов можно составить набор функций, который будет содержать разрабатываемое приложение.

Функциональные требования выражены в виде диаграммы прецедентов (рисунок 2).

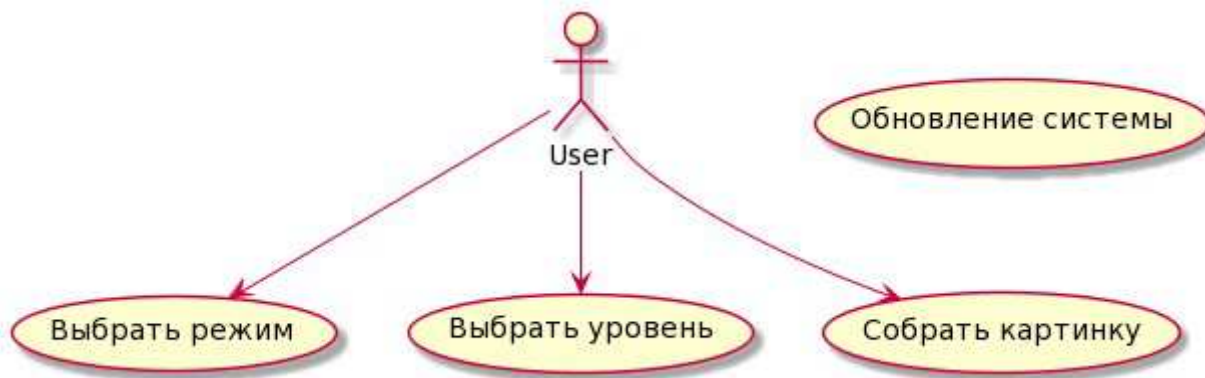


Рисунок 2 - Диаграмма прецедентов

Для каждого прецедента, приведенного на рисунке 1 проработан алгоритм действий пользователя.

Название прецедента: Выбор уровня

Предусловие: Пользователь находится в окне «Главное меню».

Цель: начать игру с выбранным уровнем.

Основная последовательность:

- 1) Пользователь выбирает пункт «Играть» в окне «Главное меню»;
- 2) Отображается окно «Выбор режима». Пользователь выбирает необходимый режим;
- 3) Отображается окно «Выбор категории», содержащий список категорий». Пользователь выбирает необходимый режим;
- 4) Отображается окно «Выбор уровня», содержащий список доступных уровней. Пользователь выбирает уровень;

Постусловие: выполняется прецедент «Собрать фигуру» с выбранными параметрами.

Название прецедента: Собрать фигуру

Предусловие: Пользователь выполнил прецедент «Выбор уровня».

Цель: собрать изображение, разбитое на фигуры.

Основная последовательность:

1) Программа инициализирует уровень (загрузка файла уровня, разбиение на фигуры, отрисовка).

2) Пользователь собирает изображение до тех пор, пока все фигуры не установятся в свою позицию.

Постусловие: Приложение генерирует диалоговое окно с информацией о прохождении уровня. Переход в окно «Главное меню», информация о прохождении записывается в базу данных.

Условия ввода и действие альтернативных сценариев:

Условие 1: Неполное завершение сбора изображения.

1) Переход в окно «Главное меню» без сохранения прогресса прохождения уровня.

Название прецедента: Выбор режима

Предусловие: Пользователь находится в окне «Главное меню».

Цель: Узнать информацию о количестве категорий и уровней.

Основная последовательность:

Пользователь выбирает пункт «Играть» в окне «Главное меню»;

Постусловие: Переход в окно «Выбор режима».

Название прецедента: Обновление системы

Предусловие: Приложение остановлено.

Цель: Обновить список категорий и уровней.

Основная последовательность:

Пользователь запускает приложение.

Постусловие: Переход в окно «Главное меню».

1.2.3 Формат файла уровня

Файл уровня должен описывать набор фигур, каждая из которых состоит из одного или более примитивов.

Примитивы описываются такими свойствами как: тип (прямоугольник, овал, линия и т.д.), цвет заливки, ширина и цвет границ. Положение примитива внутри фигуры описывается ограничивающим прямоугольником и не привязано к координатам рабочей области.

Фигура помимо набора примитивов, из которых она состоит, имеет уникальный идентификатор, необходимый для привязки фигур, поле, хранящее идентификатор фигуры, к которой привязана она и массив идентификаторов фигур, которые привязаны к ней и точку привязки.

Каждый уровень так же имеет название и тег, который позволит формировать тематические наборы уровней.

Структура файла в формате JSON приведена на рисунке 3.

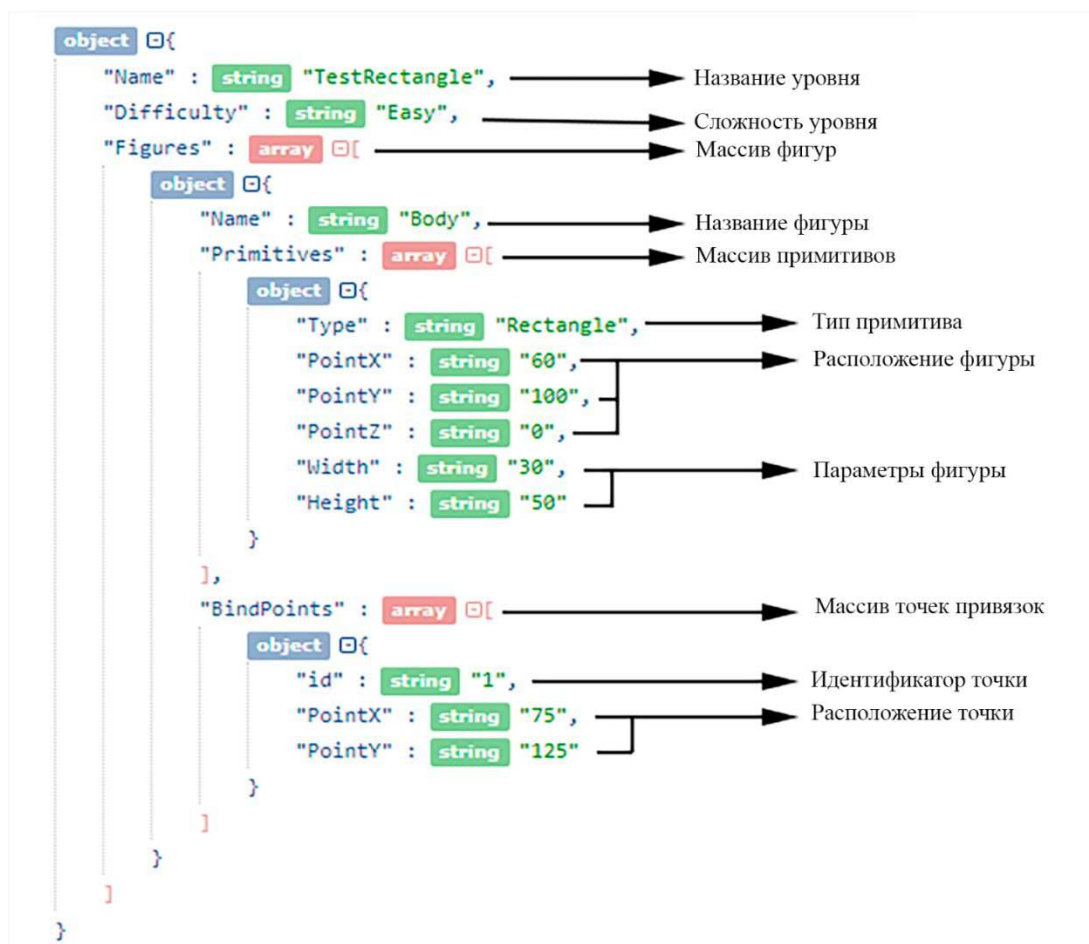


Рисунок 3 - Формат уровня в JSON - файле

1.3 Выводы по главе

- Проведен анализ аналогичных приложений.

- Сформулированы требования к разрабатываемому приложению в виде диаграмм прецедентов и их текстового описания.
- Разработаны макеты интерфейса.
- Сформулированы требования к формированию файла уровня, для дальнейшей проектировки разрабатываемого приложения.

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 уровней.



Рисунок 4 - Архитектура системы

Приложение состоит из двух экранов. Экран меню загружает список уровней, отображает их иконки. Игровой экран обрисовывает изображение, разбитое на фигуры, соответствующее выбранному в меню уровню. Модуль обновления также взаимодействует с базой данных и файловой системой телефона. Взаимодействие осуществляется через объект-фасад [5], инкапсулирующий работу с базой данных и файловой системой.

2.1 Экран меню

Данный экран является стартовым экраном приложения, и его основная задача предоставить пользователю выбор уровня и его категории, а также режима игры (рисунок 5). Так же экран меню предоставляет пользователю информацию о прогрессе с помощью отметки уже пройденных уровней.



Рисунок 5 – Экран меню

Информация об отображаемых уровнях загружается из локальной базы данных. Экран меню содержит в себе кнопки для навигации по разделам. При выборе необходимого уровня открывается экран игры с выбранным уровнем.

2.1.1 Работа с базой данных

При открытии экрана меню в соответствии с прецедентом «Обновление системы» происходит проверка обновления коллекций уровней на удаленной базе данных. При обнаружении новых уровней, приложение уведомляет пользователя о наличии обновления. Для вывода уровней и категорий на экран делается запрос в базу данных для получения информации о количестве уровней, их названий, прогресса прохождения и категории.

2.1.2 Динамическая модель

В разделе приведены диаграммы последовательностей, для наиболее значимых прецедентов, связанных со взаимодействием различных компонентов системы. На рисунке 6 изображена диаграмма последовательности для варианта использования «Обновление системы», наглядно представляющая процесс взаимодействия программы с сервером для получения новых уровней и их добавление в меню.

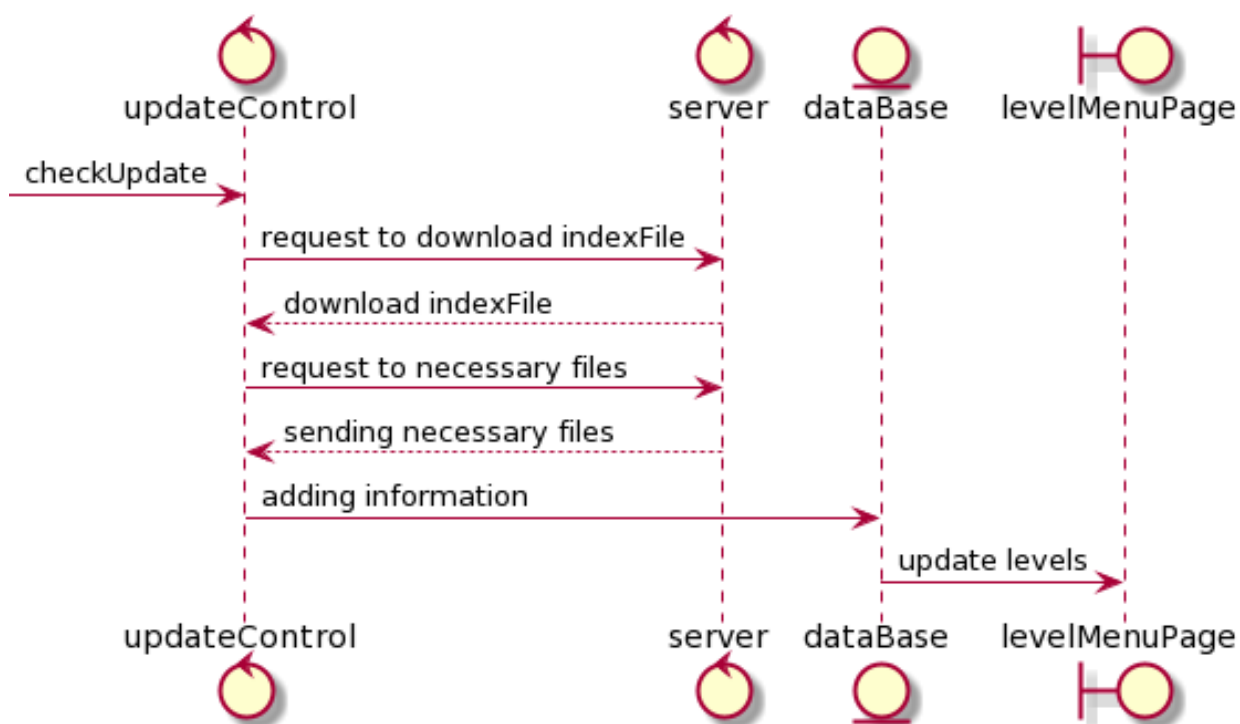


Рисунок 6 – Диаграмма последовательности «Обновление системы»

На рисунке 7 изображена диаграмма последовательности для варианта использования «Выбор уровня», наглядно представляющая процесс взаимодействия пользователя с приложением, в ходе которого пользователь сначала выбирает режим игры, после чего отображаются доступные для выбранного режима категории, после выбора категории отобразятся все соответствующие ей уровни. После выбора уровня отобразится экран игры.

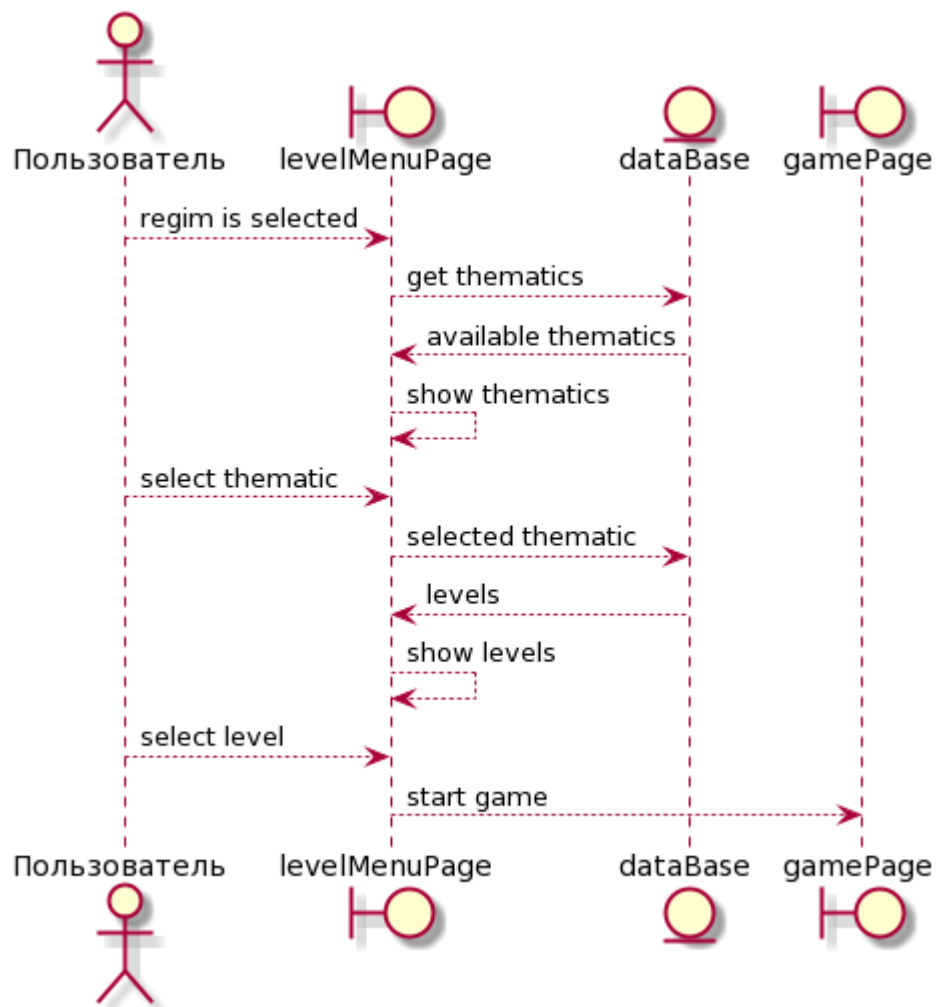


Рисунок 7 – Диаграмма последовательности «Выбор уровня»

2.1.3 Статическая модель

На основе анализа прецедентов, а также модели предметной области получена диаграмма классов.

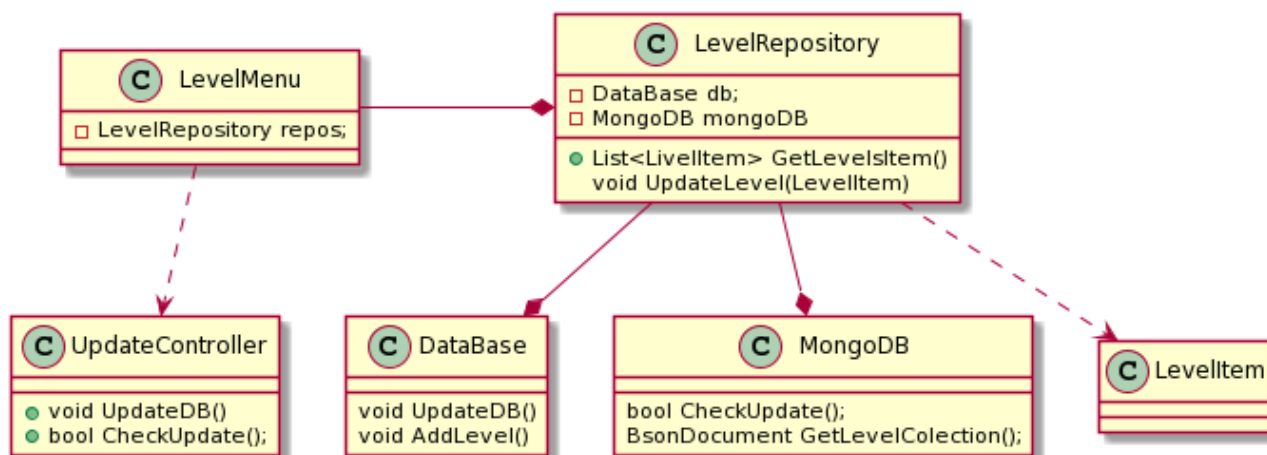


Рисунок 8 – Диаграмма классов экрана меню

2.2 Игровой экран

Данный экран содержит всю логику игры (рисунок 9), экран инициализирует и отрисовывает все фигуры выбранного уровня. Позволяет пользователю перемещать и вращать фигуры. При открытии экрана отображается конечная картинка, которую пользователь должен собрать из случайно разбросанных фигур. Экран имеет алгоритмы инициализации фигур, перемещения, проверки правильного положения фигур. Информация о фигурах, содержащихся в уровне, считывается из файлов, полный путь до которых передается из экрана меню при выборе уровня.

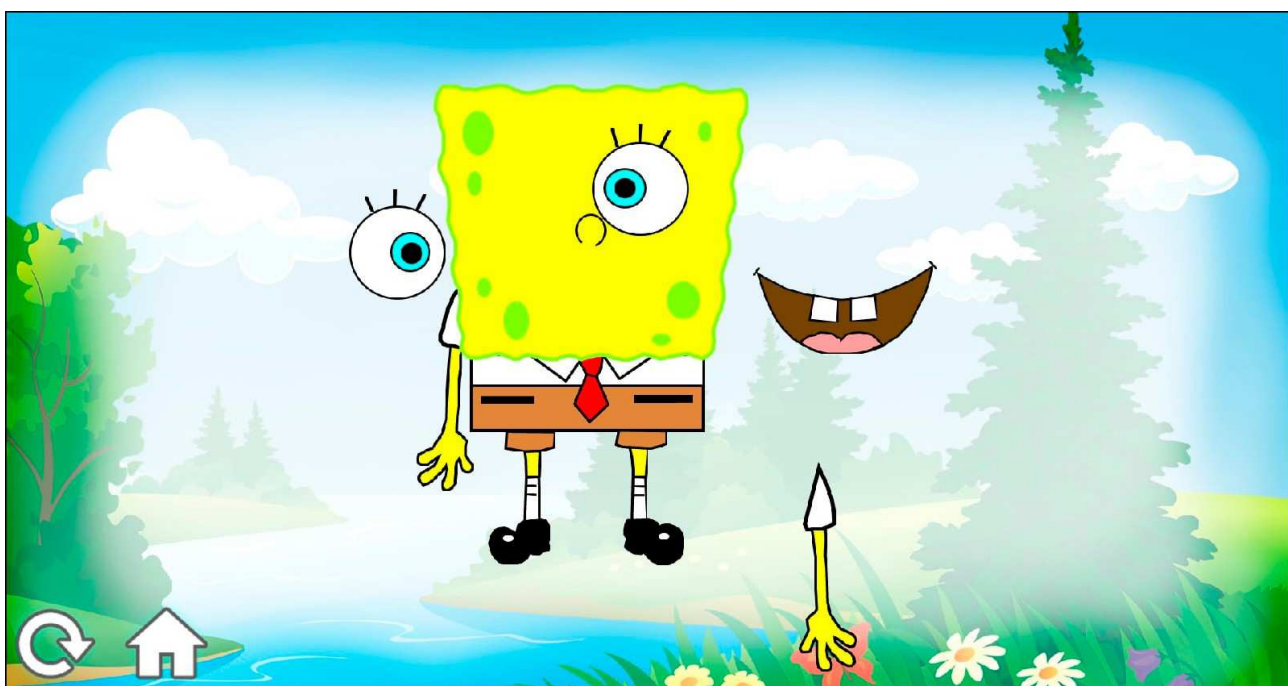


Рисунок 9 – Экран игры

2.2.1 Алгоритм инициализации фигур

Из файла уровня считываются данные, на основе которых создаются фигуры, содержащие примитивы. Каждый примитив представлен точечной картинкой, которая масштабируется в зависимости от размера экрана устройства. Конечный рисунок занимает $\frac{2}{3}$ видимой части экрана. Далее фигурам устанавливаются точки привязки, эффект масштабирования применяется для всех параметров фигуры. При слиянии примитивов генерируется точечная картинка, которая будет отправлена в массив содержащихся на игровом поле фигур. После инициализации всех фигур,

массив точечных картинок будет отправлен стандартному графическому отрисовщику.

2.2.2 Алгоритм передвижения фигур

Каждая фигура знает о своем местоположении на игровом поле и имеет функцию проверки принадлежности координаты, поэтому при обработке касания экрана, осуществляется вызов функции проверки принадлежности координаты у всех фигур по очереди возрастания `zIndex`'а. Для самого перемещения используется функция `TouchMove`, которая передвигает фигуру учитывая внутреннее местоположение касания для относительного перемещения. `TouchMove` вызывает функцию `Move` которая перемещает `DrawPoint` фигуры.

2.2.3 Алгоритм правильного положения фигур

Любые две фигуры образуют пару «главный – зависимый» в которой главная фигура хранит `Id` всех фигур, зависящих от нее, зависимая же фигура хранит `Id` главной фигуры (к которой она привязана) и точку привязки, образованную разницей от точки отрисовки данной фигуры до точки отрисовки главной фигуры по осям `X` и `Y` соответственно.

Если при добавлении к текущим координатам зависимой фигуры разницы, полученное положение координаты совпадет с координатой точки отрисовки главной фигуры, то фигуры друг относительно друга расположены правильно. Для облегчения процесса сбора картинки существуют допуски. Допуск — это значение, являющееся величиной описывающее радиус окружности в центре которой должна находиться начальная точка фигуры при этом эта точка может находиться в любой координате, в пределах окружности и будет считаться что фигура стоит на своем месте. У фигуры может быть, множество зависимых фигур, но при этом главная фигура может быть только одна. Фигуры не могут являться главными друг для друга.

На рисунке 10 изображены начальная точка (1) зависимой фигуры, точка привязки (2) и область (3), в пределы которой должна попасть точка зависимой фигуры. При перемещении фигуры проверяются координаты точки «1» и если эти координаты совпадают с координатами области «3» (лежат внутри), то считается что фигура стоит на своём месте.

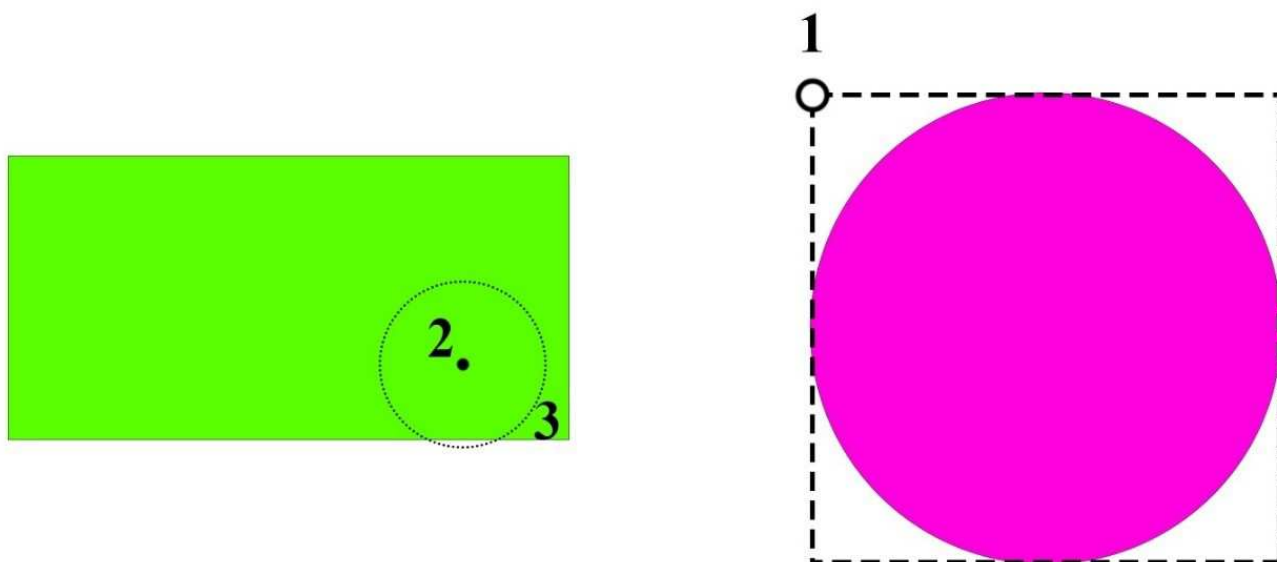


Рисунок 10 - Связь главный-зависимый

2.2.4 Динамическая модель

На рисунке 11 изображена диаграмма последовательности для варианта использования «Игра», представляющая процесс взаимодействия пользователя с игровым экраном при сборе картинки.

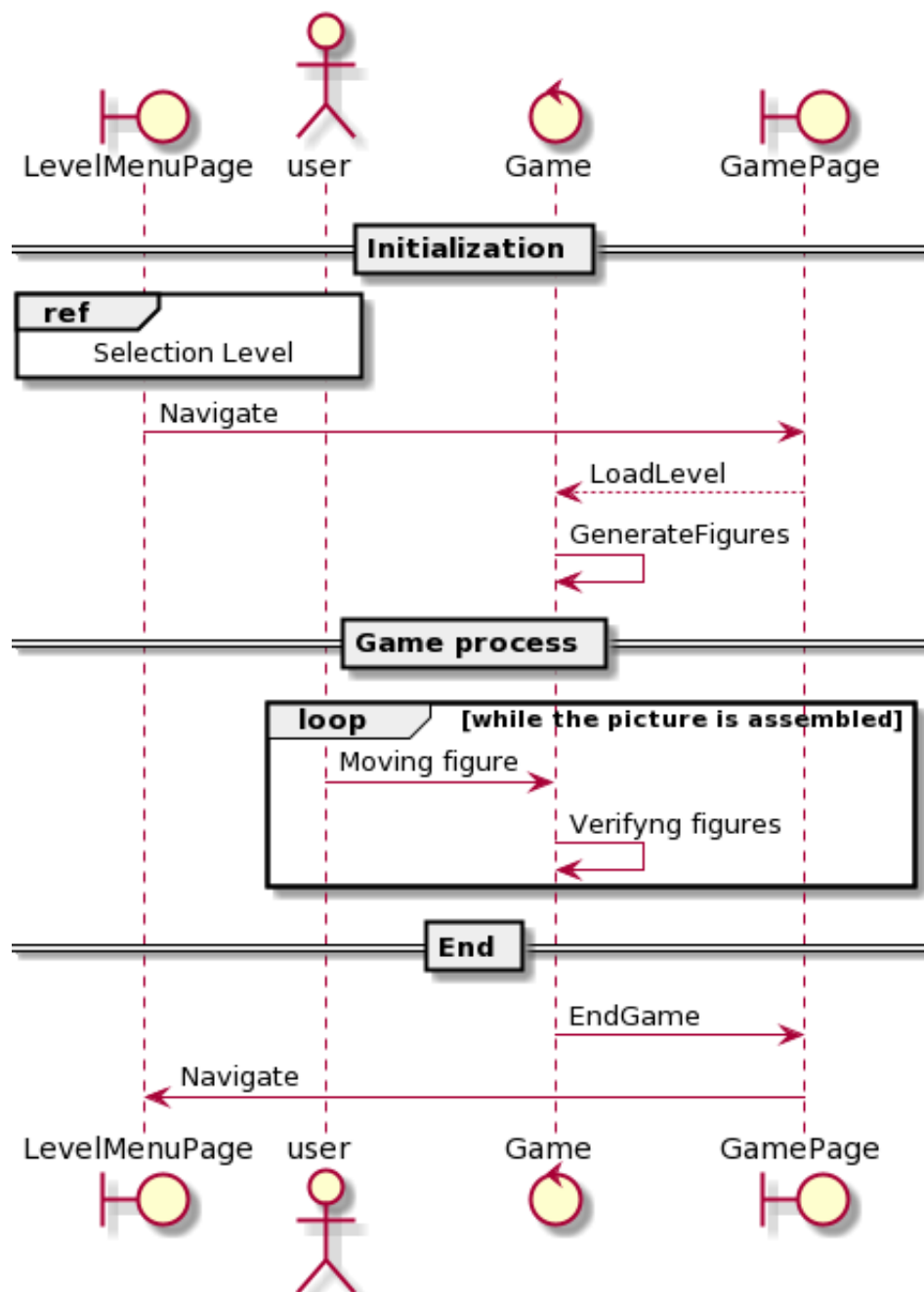


Рисунок 11 – Диаграмма последовательности «Игра»

2.2.5 Статическая модель

Фигуры генерируются из примитивов, в программе реализованы 4 вида примитивов, а также класс, у которого функция `CreateFromData` возвращает объект примитива сгенерированного из десериализованных данных.

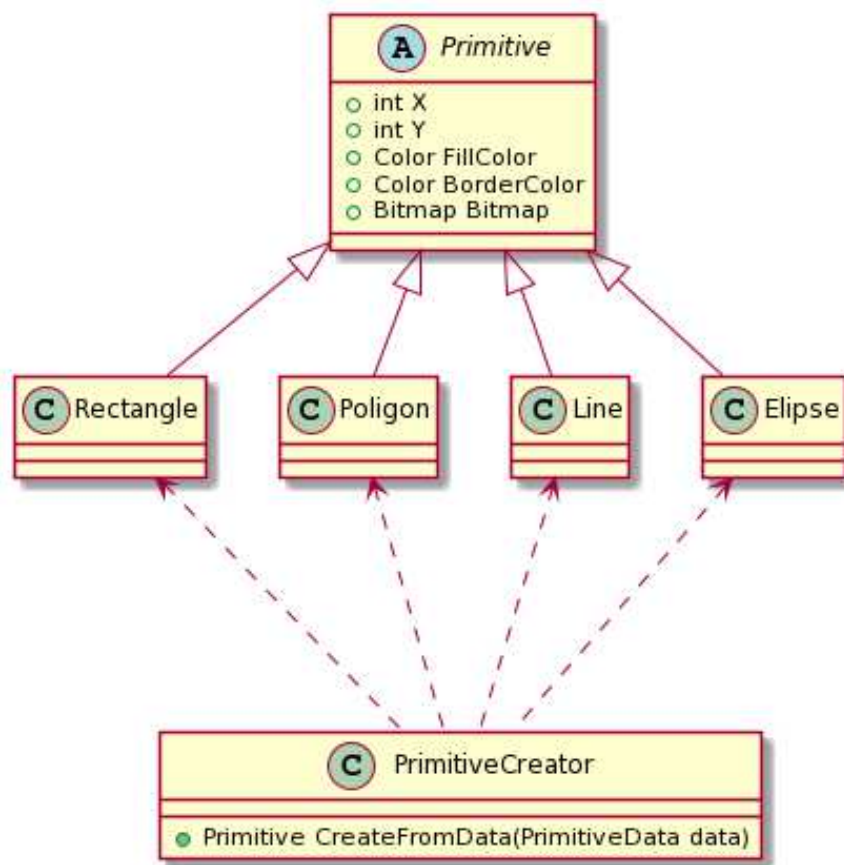


Рисунок 12 - Диаграмма классов примитивов

Считанные данные json десериализуются в класс **LevelData** который имеет идентичную структуру с файлом уровня.

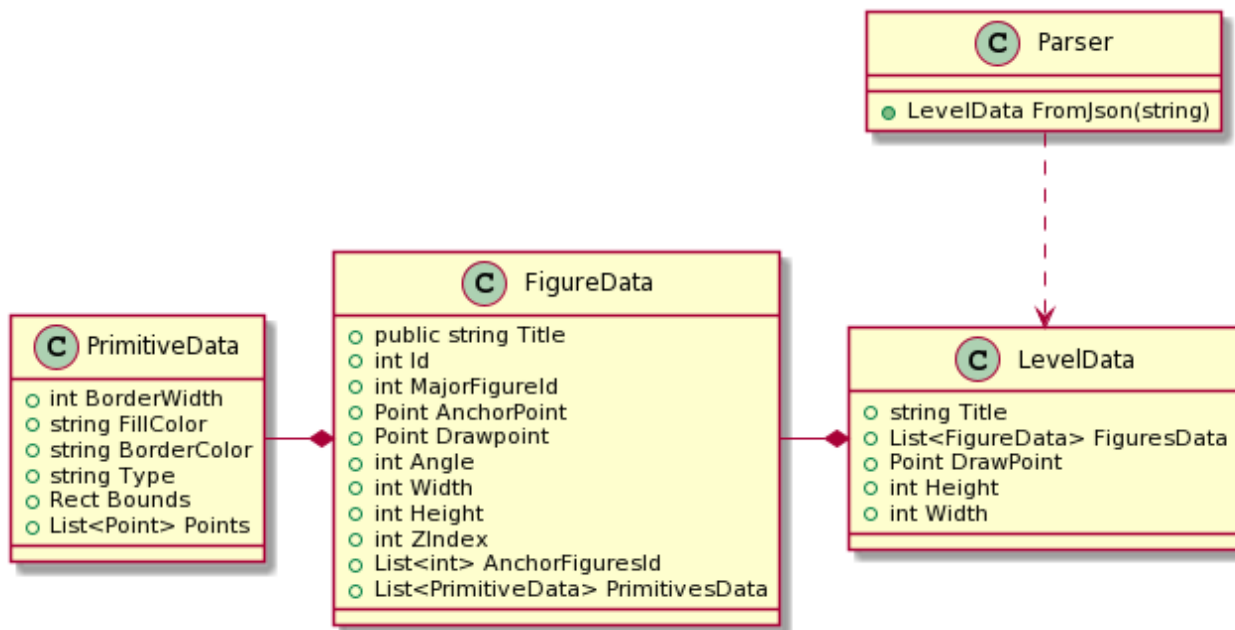


Рисунок 13 - Диаграмма классов парсера

Считанные и инициализированные фигуры хранятся в ассоциативном контейнере, это нужно для эффективного доступа к их элементам по Id фигуры.

GameMap содержит метод CheckStatus который проверяет правильность расположения фигур. CurrentFigure это ссылка на объект с которым взаимодействует пользователь.

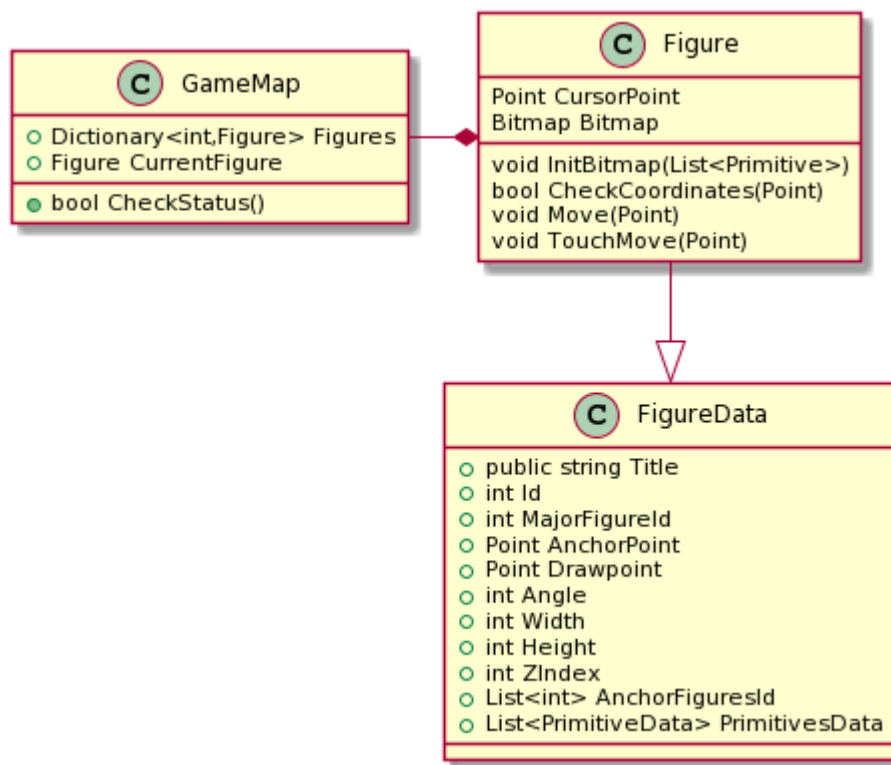


Рисунок 14 - Диаграмма классов игрового поля

2.3 База данных

База данных на сервере является документо-ориентированной системой управления данных, не требующей описания схемы таблиц. База данных хранит json документы, структура которого описана на рисунке 15.

При установке приложения, информация о уровнях считывается из файла json, а затем сохраняется в клиентскую базу данных. При этом размещаясь в 2х таблицах. Структура базы данных представлена на рисунке 16.

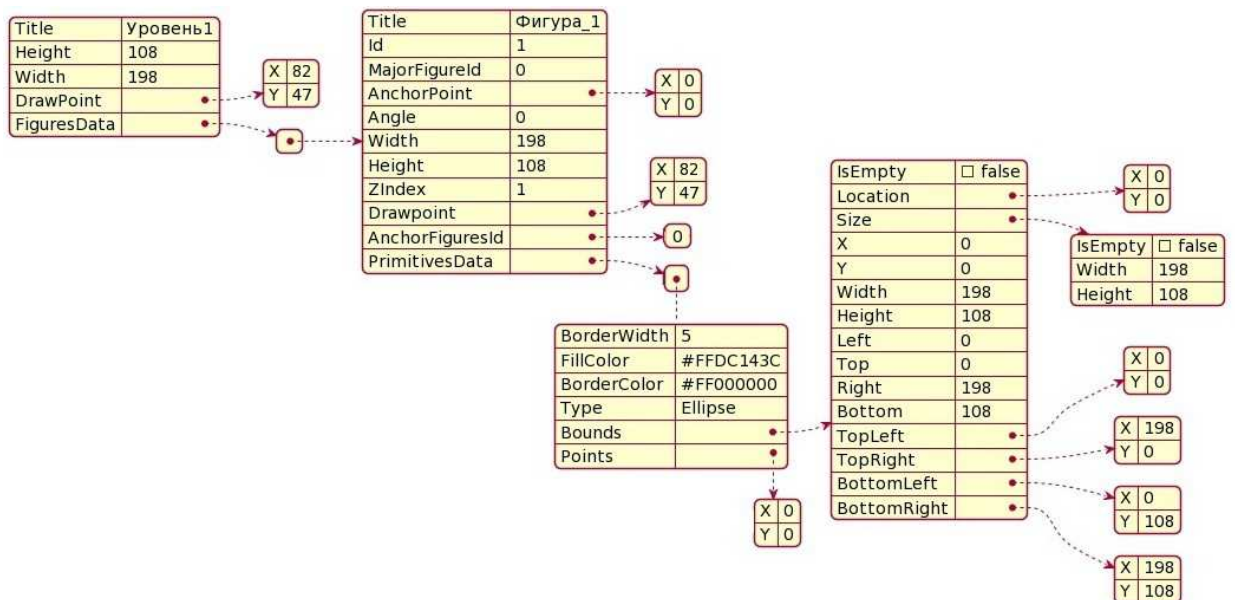


Рисунок 15 – Структура JSON документа

Кроме того, для хранения и обновления настроек в игре используется таблица settings. Поле «property» задает тип свойства. Всего используется 2 разновидности свойств: «date» - дата последнего успешного обновления; «path» - путь к каталогам для хранения изображений. При изменении значения одного из свойств, оно перезаписывается в БД.

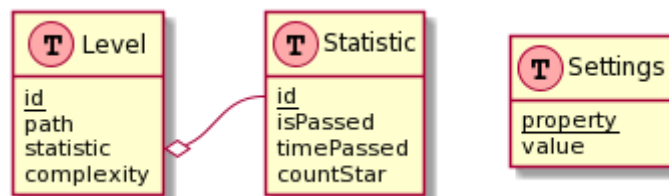


Рисунок 16 – Структура базы данных

2.4 Выводы по главе

В соответствии с техническим заданием:

- Предложена архитектура системы;
- На основе анализа диаграмм последовательностей разработаны диаграммы классов, описывающие статическую модель системы;
- Разработаны структуры данных и алгоритмы объединения и привязки фигур.

3 Реализация и тестирование

3.1 Реализация

3.1.1 Используемые инструменты

Реализация приложения производилась с использованием языка C#. Для мобильной разработки использовался фреймворк Xamarin [6]. В нем реализованы все необходимые библиотеки для создания игры. Так же фреймворк позволил сделать кроссплатформенное приложение. В роли графического движка используется библиотека Skia [7]. Skia предоставляет графический элемент Xamarin, интерфейс которого позволяет отрисовывать векторные изображения.

3.1.2 База данных

Данные приложения (настройки и информация об уровнях) хранятся в базе данных SQLite [8]. В соответствии с разработанным проектом, используется фасад базы данных, который реализуется посредством стандартных классов библиотек C#.

3.1.3 Работа с сервером

При запуске приложения создается сервис, который делает запрос на удаленный сервер, и, если сервер возвращает информацию о наличии новых или обновлении уже существующих уровней, в приложении вызывается диалоговое окно, в котором пользователь может разрешить приложению обновить список уровней.

В качестве удаленной базы данных используется MongoDB [9] развернутая на облачном сервере с Ubuntu 20.04[10].

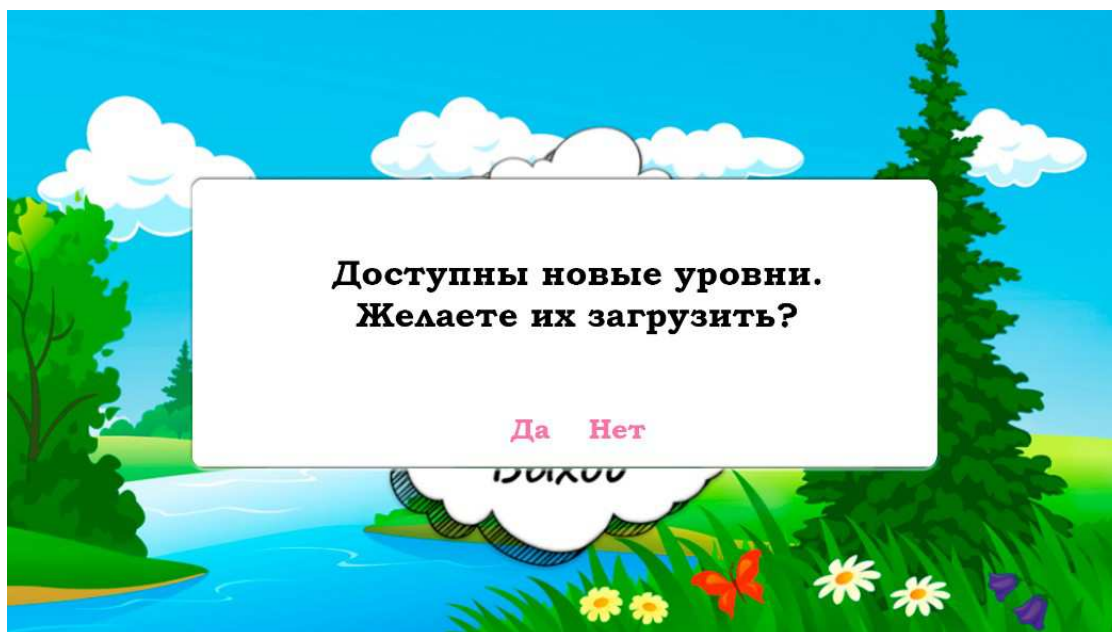


Рисунок 17 – Диалоговое окно с запросом на скачивание

3.2 Инструкция разработчика

Для полноценной работы с Xamarin может быть использована Visual Studio [11]. Для того чтобы собрать и скомпилировать проект необходимо выполнить следующие действия:

- скачать Visual Studio 2019 [11] версии не ниже 16.9;
- в окне выбора компонентов, представленном на рисунке 18, выбрать пункт «Разработка мобильных приложений .NET»;
- запустить Visual Studio и в открывшемся окне, представленном на рисунке 19, выбрать «Клонирование репозитория»;
- установить пакет Android SDK [12];
- ввести url-адрес git-репозитория [13] в соответствующее поле и нажать кнопку «Клонировать».

После этого проект клонируется на компьютер и будет доступен для редактирования и запуска в Visual Studio.

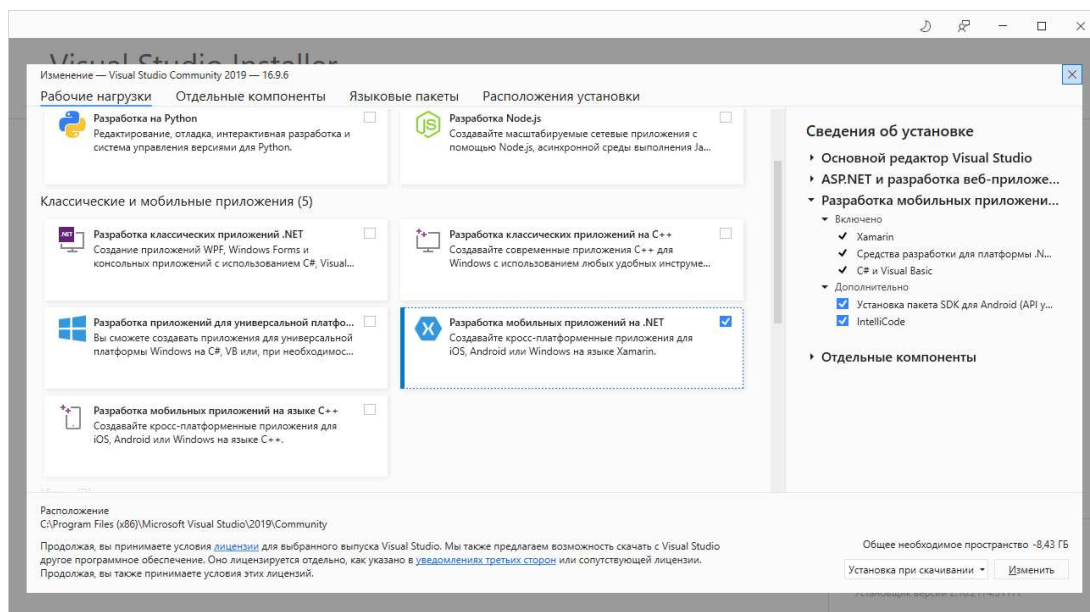


Рисунок 18 – Окно выбора компонентов

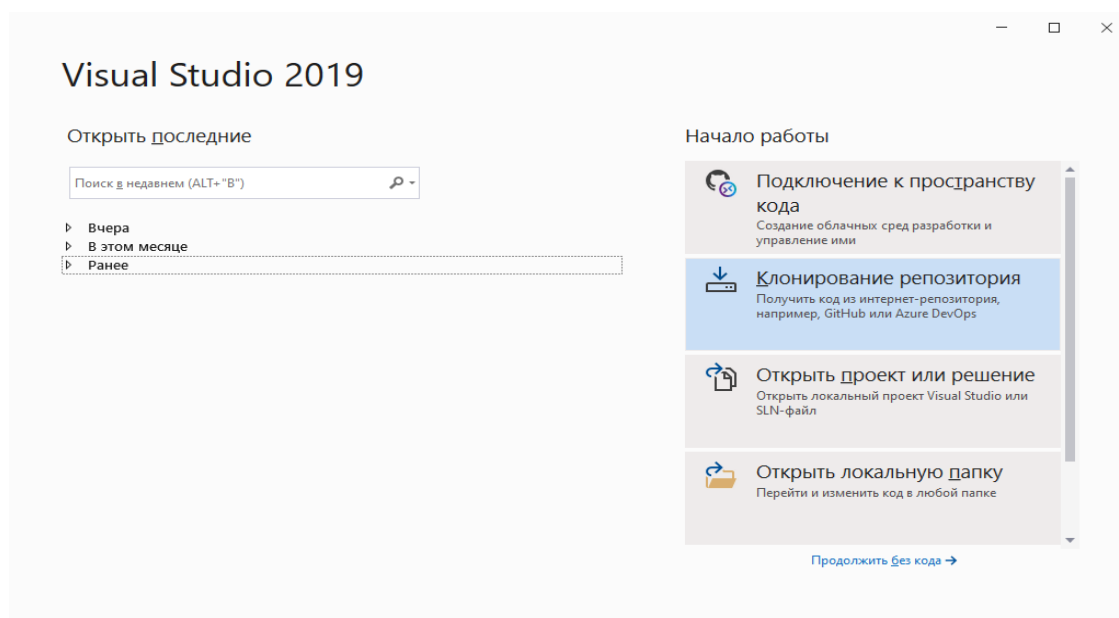


Рисунок 19 – Окно клонирования репозитория

3.3 Инструкция пользователя

Для запуска приложения потребуется эмулятор или смартфон, функционирующий на ОС Android версии не ниже 8.

Установочный файл можно скачать с облачного хранилища Google Disk [14].

В качестве инструкций по работе с приложением могут использоваться разработанные в первом разделе прецеденты.

3.4 Тестирование

Тестирование игровой логики и корректности работы парсера проводилось с использованием модульных тестов. Проект с модульными тестами расположен рядом с проектом приложения на GitHub [13].

Тестирование графического интерфейса проводилось вручную. Например, для проверки корректного завершения игры были выполнены действий:

- в главном окне нажата кнопка «Играть»;
- в окне с выбора был нажат уровень;
- все фигуры были перемещены в правильные позиции;
- проверено наличие всплывающего окна сообщающего о удачном завершении прохождения уровня;
- проверено состояние прохождения в меню уровней.

Подобное тестирование было проведено для всех прецедентов в главе 1.

Тестирование работы с удаленной базой данных проводилось с помощью программы MongoDB Compass [15].

3.5 Выводы по главе

- реализована удалённая база данных для хранения уровней, использующихся в игре;
- проведено тестирование приложения;
- разработана инструкция пользователя, содержащая информацию для установки и запуска приложения;
- разработана инструкция разработчика, содержащая информацию для установки среды разработки и открытия проекта для редактирования и запуска.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы реализовано мобильное приложение «Картинки из геометрических фигур». Развернут сервер для взаимодействия с удаленной базой данных, а также для взаимодействия с редактором уровней.

Для разработчика написана инструкция по установке рабочей среды и получения проекта для редактирования и запуска. Корректная работа приложения проверена путём ручного тестирования и с помощью модульных тестов. Архитектура была построена таким образом, что она позволяет легко дорабатывать приложение, расширяя его функционал.

Приложение корректно функционирует на операционной системе Android версии 8 и выше.

Проект доступен для скачивания из git-репозитория по ссылке [13].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дидактическая игра «Собери картинку из геометрических фигур» // Маам URL: <https://www.maam.ru/detskijasad/didakticheskaja-igra-soberikartinku-iz-geometricheskih-figur-991796.html> (дата обращения: 08.02.2021).
2. Танграм для детей // Google Play URL: <https://play.google.com/store/apps/details?id=com.robotifun.tangram&hl=ru&gl=US> (дата обращения: 10.03.2021).
Poly Shape - игра-головоломка
3. Poly Shape - игра-головоломка Tangram // Google Play URL: https://play.google.com/store/apps/details?id=com.easygame.simplepuzzle&hl=en_US&gl=US (дата обращения: 10.03.2021).
4. Танграм King // Google Play URL: <https://play.google.com/store/apps/details?id=com.mobirix.tangram&hl=ru&gl=US> (дата обращения: 10.03.2021).
5. Создание уровня доступа к данным (C#) // Microsoft Docs URL: <https://docs.microsoft.com/ru-ru/aspnet/web-forms/overview/data-access/introduction/creating-a-data-access-layer-cs> (дата обращения: 27.03.2021).
6. Xamarin.Android // Microsoft Docs URL: <https://docs.microsoft.com/ru-ru/xamarin/android/> (дата обращения: 04.02.2021).
7. Skia // Skia URL: <https://skia.org/> (дата обращения: 04.02.2021).
8. Общие сведения о Microsoft.Data.Sqlite // Microsoft Docs URL: <https://docs.microsoft.com/ru-ru/dotnet/standard/data/sqlite/?tabs=netcore-cli> (дата обращения: 26.03.2021).
9. MongoDB // MongoDB URL: <https://mongodb.com> (дата обращения: 30.04.2021).
10. Облачные серверы // Reg URL: <https://www.reg.ru/vps/cloud/> (дата обращения: 30.04.2021).

11. Visual Studio 2019 для Windows и Mac // visualstudio.microsoft.com
URL: <https://visualstudio.microsoft.com/ru/downloads/> (дата обращения: 30.04.2021).
12. Android Studio // [developers.android](https://developers.android.com) URL:
<https://developer.android.com/studio> (дата обращения: 30.04.2021).
13. Tangram // GitHub URL: <https://github.com/9PK4U/Tangram> (дата обращения: 21.04.2021).
14. Google Disk // [drive.google](https://drive.google.com) URL:
https://drive.google.com/drive/folders/10CKD0HRL7cN24h7xY-ttrxb4q_yDc0Fg?usp=sharing (дата обращения: 30.04.2021).
15. Compass // [mongodb.com](https://www.mongodb.com) URL:
<https://www.mongodb.com/products/compass> (дата обращения: 30.04.2021).

СПИСОК СОКРАЩЕНИЙ

БД – база данных

ВКР – выпускная квалификационная работа

ОС – операционная система

ТЗ – техническое задание

UML – унифицированный язык моделирования (Unified Modeling Language)

Федеральное государственное автономное
Образовательное учреждение
Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Вычислительная техника

УТВЕРЖДАЮ

Заведующий кафедрой ВТ


подпись

О.В. Непомнящий

инициалы, фамилия

« 08 »


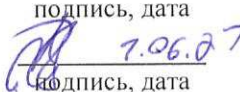
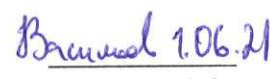
06

2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Android-приложение «Картинки из геометрических фигур»

Руководитель	 подпись, дата	ст. преподаватель	В.С. Васильев
Выпускник	 подпись, дата		О.В. Волков
Нормконтролер	 подпись, дата	ст. преподаватель	В.С. Васильев

Красноярск 2021