

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук  
Кафедра системного программирования**

РАБОТА ПРОВЕРЕНА

Рецензент  
старший преподаватель кафедры ЗИ

\_\_\_\_\_ В.М. Жернова

“ \_\_\_ ” \_\_\_\_\_ 2018 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,  
д.ф.-м.н., профессор

\_\_\_\_\_ Л.Б. Соколинский

“ \_\_\_ ” \_\_\_\_\_ 2018 г.

**РАЗРАБОТКА МОБИЛЬНОЙ ИГРЫ В ЖАНРЕ «ГОЛОВОЛОМКА»  
ДЛЯ ОС ANDROID**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 02.03.02.2018.115-023.ВКР

Научный руководитель  
старший преподаватель кафедры СП  
\_\_\_\_\_ К.В. Репина

Автор работы,  
студент группы КЭ-402  
\_\_\_\_\_ К.С. Аксенова

Ученый секретарь  
(нормоконтролер)

\_\_\_\_\_ О.Н. Иванова

“ \_\_\_ ” \_\_\_\_\_ 2018 г.

Челябинск-2018

## ОГЛАВЛЕНИЕ

|   |    |
|---|----|
| ВВЕДЕНИЕ .....  | 4  |
| 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....                            | 7  |
| 1.1. Обзор аналогов .....                                     | 7  |
| 1.2. Требования к разрабатываемому приложению.....            | 10 |
| 1.3. Обзор существующих платформ для создания приложений..... | 10 |
| 2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ .....                               | 14 |
| 2.1. Функциональные и нефункциональные требования .....       | 14 |
| 2.3. Диаграмма прецедентов .....                              | 15 |
| 2.4. Проектирование интерфейса.....                           | 17 |
| 3. РЕАЛИЗАЦИЯ КОМПОНЕНТОВ СИСТЕМЫ.....                        | 22 |
| 3.1. Инструменты, используемые для реализации.....            | 22 |
| 3.2. Файловая структура .....                                 | 22 |
| 3.3. Реализация компонентов системы.....                      | 23 |
| 4. ТЕСТИРОВАНИЕ .....   | 27 |
| ЗАКЛЮЧЕНИЕ .....  | 31 |
| СПИСОК ЛИТЕРАТУРЫ .....                                       | 32 |
| ПРИЛОЖЕНИЕ .....  | 34 |

## **ВВЕДЕНИЕ**

### **Актуальность**

В настоящее время мобильные телефоны играют огромную роль в жизни человека. Многим ежедневно требуется иметь под рукой большое количество информации, контактов, записей и т.д. И один из экономичных и быстрых способов получения этих данных – хранение их в электронном виде, в частности в телефоне, который всегда с собой.

Мобильные приложения – это программные продукты, разработанные специально для мобильных устройств, смартфонов, планшетных компьютеров или других мобильных устройств. Мобильные приложения распространяются через магазины приложений: Apple App Store, Google Play, Windows Phone Store, BlackBerry App World и др.

Мобильные приложения помогают решать различные прикладные задачи: от мобильной картографии и приема электронной почты до узкоспециализированных функций. Они призваны облегчить жизнь пользователей мобильных устройств, а также ее разнообразить [17].

Для этих целей было разработано большое количество приложений, ориентированных на разные области:

- 1) приложения для работы с файлами;
- 2) приложения для прослушивания аудио и просмотра видео;
- 3) приложения для дистанционного управления техникой;
- 4) приложения, ориентированные на быт и повседневную жизнь;
- 5) обучающие и развивающие приложения;
- 6) приложения для контроля управления финансами;
- 7) браузеры;
- 8) карты и навигация;
- 9) социальные сети.

Хотелось бы обратить внимание на то, что помимо стандартных необходимых для работы приложений люди также используют технику для обучения, развития и развлечения в свободное время.

В связи с этим, в последнее время возросло разнообразие игр на телефоны. В частности, большой популярностью пользуются игры-головоломки, с помощью которых можно не только отвлечься, но и с пользой провести время.

Головоломка – это жанр игр, в центре внимания которых оказывается необходимость решения различных логических задач. В основе игровой механики лежит изучение неких правил, установление закономерностей, поиск комбинацией с целью достижения определенного результата. Это требует концентрации внимания и задействования логики [10].

На сегодняшний день ассортимент головоломок широк и разнообразен, и найти подходящую можно практически для всех возрастных и социальных групп.

### **Цель и задачи работы**

Целью проекта является разработка приложения в жанре «головоломка» для ОС Android.

Для реализации данной цели были поставлены следующие задачи:

- 1) изучить существующие методы разработки мобильных игр для ОС Android;
- 2) выполнить анализ требований;
- 3) выполнить проектирование мобильной игры;
- 4) реализовать мобильную игру для ОС Android;
- 5) протестировать разработанную игру.

### **Структура и объем работы**

Работа состоит из введения, четырех глав, заключения, списка литературы и приложения. Объем работы составляет 34 страницы, объем списка литературы – 20 источников, объем приложения – 5 страниц.

### **Содержание работы**

Первая глава «Анализ предметной области» содержит обзор аналогичных проектов, на основе чего выявлены требования к разрабатываемому приложению. Также в данной главе описан обзор существующих платформ

для создания приложений и сделан выбор в пользу конкретной платформы (Unity).

Вторая глава «Проектирование системы» содержит описание и анализ функциональных и нефункциональных требований к разрабатываемому мобильному приложению, описание вариантов использования системы, подробности проектирования интерфейса, демонстрацию макетов и их описание.

Третья глава «Реализация компонентов системы» содержит описание и листинг некоторых основных функций программы.

В четвертой главе «Тестирование» в виде таблицы представлено тестирование системы.

В заключении описываются основные результаты, полученные при выполнении дипломной работы.

В приложении представлен графический интерфейс пользователя мобильного приложения.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Обзор аналогов

На сегодняшний день существует достаточно широкое разнообразие приложений жанра «головоломка». Но не все из них подходят для всех возрастов. Некоторые игры содержат уровни по нарастанию сложности. И дети младшего возраста не могут пройти все представленные уровни самостоятельно. Также зачастую во многих приложениях ограниченное количество времени либо уровней, которые пользователь может пройти бесплатно. Дальнейшее использование приложения – платное.

Для выявления преимуществ и недостатков для анализа были выбраны на рассмотрение 3 приложения: «Lolo», «100 дверей», «Слово за слово» и «Plumber 2».

### **LOLO: Puzzle Game**

«LOLO: Puzzle Game» – игра головоломка.

Перед установкой игры на телефон ее условия можно прочитать в описании. Основная задача игры – собирать квадраты одинакового цвета для достижения максимального счета [14].

Разработчик утверждает, что «очень легко играть, уместно для каждого возраста (4+)». Дизайн приложения достаточно минималистичный. Но в основном все меню на английском языке. Для людей, обладающих базовыми знаниями языка не составит большого труда разобраться, но все же могут возникнуть некоторые сложности.

Также со временем разработчики усовершенствовали игру, добавив большее количество уровней сложности и режимов игры. Большое количество режимов игры является «плюсом». Но с другой стороны, с появлением новых режимов стало гораздо сложнее ориентироваться в приложении. Далеко не с первого раза можно понять, в чем цель конкретных режимов игры. Приходилось несколько раз пробовать играть в данный режим, чтобы понять, в чем заключается его суть. По моему мнению, это большой минус приложения.

Еще одним достоинством данного приложения является наличие бонусов (3 вида). «Молния» удаляет 1 квадрат, выбранный пользователем. «Магнит» собирает все квадраты одного цвета, выбранного пользователем. «+3» - увеличение количества ходов в некоторых режимах игры.

### **100 дверей**

«100 дверей» – игра-головоломка.

В данном случае в описании приложения все было описано более структурированно и понятно, чем в приложении «Lolo».

В особенностях игры указано: увлекательные головоломки; крутые механики: перемещения по комнате, комбинирование предметов, поиск скрытых предметов и многое другое; великолепные анимации; современная 3D графика; играйте совершенно бесплатно. Главная задача в «100 дверей» - это «побег из комнаты. На каждом этаже вам предстоит решить сложную головоломку (в том числе используя возможности своего устройства, такие как: наклоны, тряска и другое), чтобы открыть дверь и перейти на следующий этаж» [13].

Игра действительно показалась увлекательной, удалось пройти ее до конца. Из недостатков – несколько уровней проходились «сами собой». Стоило лишь несколько раз нажать на экран, дверь сама открывалась и не было даже понимания, в чем именно заключалась головоломка. Из достоинств – игра действительно увлекательная, хоть и достаточно простая для моего возраста. Больше подойдет для детей и подростков. Также стоит отметить то, что каждый уровень в данном приложении уникальный. На каждом этаже игроку предстоит разгадать новую головоломку.

Дизайн приложения достаточно простой и понятный. Меню на русском языке. На первых уровнях присутствуют всплывающие окна, где описаны некоторые инструкции, в чем заключается суть игры и что предстоит сделать, чтобы ее пройти.

### **Слово за слово**

«Слово за слово» - развивающая игра-головоломка.

Цель игры – найти на поле как можно больше слов за 2 минуты [16]. Интересное в этом приложении то, что можно играть вместе со случайным соперником. С одним соперником можно сыграть от 1 до 3 раундов. Кто набирает большее количество баллов за найденные слова – тот и является победителем.

В данном приложении достаточно большое количество использованных цветов и иконок, которые отвлекают на себя внимание во время игры.

### **Plumber 2**

Plumber 2 – игра-головоломка.

Цель игры – соединить трубы, чтобы получился единый водопровод. В этой игре нет ограничения по времени, но есть ограниченное количество ходов в каждом раунде. Вам нужно спасти цветок, пока ваши ходы не закончились (одно прикосновение к трубе – один ход). Так что разработайте четкую водопроводную стратегию перед тем, как приступить к работе с трубами [15].

В данном приложении представлен один режим игры. Для пользователя доступны несколько уровней. Каждый следующий уровень открывается после успешного прохождения предыдущего.

Существенным недостатком данного приложения является то, что на игровом поле в состав водопровода входят не все трубы. Некоторые из них добавлены на поле с целью запутать пользователя.

Также в каждом уровне для игрока доступно ограниченное количество шагов. Следовательно, пользователю предлагается сначала построить водопровод мысленно и далее собрать его в приложении.

Какие-либо бонусы для упрощения прохождения уровней отсутствуют.

Исходя из вышесказанного можно сделать вывод о том, что данная игра подходит для детей более старшего возраста и взрослых.

Также стоит отметить, что во всех рассмотренных играх присутствует посторонняя реклама.



## **Требования к разрабатываемому приложению**

Анализируя аналоги, описанные выше, были определены требования, которыми разрабатываемое приложение в жанре «головоломка» должно обладать:

- 1) удобное, простое и понятное меню;
- 2) удобная навигация внутри приложения. В том числе возможность в любой момент вернуться на главный экран;
- 3) несколько режимов игры;
- 4) возможность выбора уровня сложности;
- 5) начисление баллов за прохождение уровней;
- 6) возможность покупать за баллы бонусы-подсказки для облегчения прохождения уровней.

## **Обзор существующих платформ для создания приложений**

Разработка приложений для мобильных устройств – это процесс, при котором приложения разрабатываются для небольших портативных устройств, таких, как КПК, смартфоны или сотовые телефоны. Эти приложения могут быть предустановлены на устройство в процессе производства, загружены пользователем с помощью различных платформ для распространения ПО или являться веб-приложениями, которые обрабатываются на стороне клиента (JavaScript) или сервера [20].

Каждая из платформ для мобильных приложений имеет интегрированную среду разработки, предоставляющую инструменты, позволяющие разработчику программировать, тестировать и внедрять приложения на целевую платформу [2].

Игровые платформы являются фундаментом для разработки игр. Создание даже самой простой игры с нуля требует огромных затрат времени и энергии на разработку функционала. Есть множество задач разработки и паттернов дизайна, которые могут быть применимы во многих играх. Именно поэтому сторонние разработчики усмотрели необходимость

разработать совместно используемые библиотеки и движки специально для Android-разработчиков [5].

Игровые платформы помогают облегчить разработку игр и позволить разработчику сфокусироваться на совершенствовании игровой механики. К тому же использование игровой платформы позволяет значительно сэкономить время, сократить объем тестирования и предоставить ряд преимуществ, например, возможность оптимизации под конкретные платформы [18].

Игровые платформы широко варьируются в зависимости от того, что они могут дать разработчику: некоторые предлагают полный игровой пакет, другие – поддержку низкоуровневых библиотек. Одни – 2D , другие – наоборот, 3D [7].

Некоторые игровые платформы являются характерными для платформы Android, в то время как другие могут быть кроссплатформенными и поддерживать разработку консольных игр в дополнение к мобильным. В настоящий момент существует множество платформ для Android. Наиболее популярных из них: «Unity3D», «Corona SDK», «Unreal Engine 4» и др.

Ниже представлено сравнение наиболее популярных платформ по некоторым критериям (табл. 1).

Табл. 1. Сравнение платформ для Android.

| <b>Критерий</b>                 | <b>Unity3D [3]</b>                                    | <b>Corona SDK [1]</b>    | <b>Unreal Engine 4 [4]</b>                      |
|---------------------------------|---|--------------------------|---|
| <b>Язык программирования</b>    | C#, JavaScript, Boo, другие языки, основанные на .NET | Lua                      | C++   |
| <b>Платформы для разработки</b> | Android, iOS (iPhone/iPad), PC, Mac, desk-            | iOS, Android, NOOK Color | ПК, Mac, Xbox, GameCube, Wii, PlayStation, iOS, |

| <b>Критерий</b>                          | <b>Unity3D [3]</b>   | <b>Corona SDK [1]</b>   | <b>Unreal Engine 4 [4]</b>        |
|--|--|---|-----------------------------------|
|  | top browser, XBOX360, PS3, Wii. BlackBerry Playbook, Nokia Symbian, Roku 2 и др. |   | <b>Android</b> , Nintendo Switch. |
| <b>Стоимость инструментов разработки</b> | бесплатная и коммерческая лицензии для разработки.                               | бесплатная неограниченная trial-версия; доступны коммерческие лицензии для внедрения. | Доступны коммерческие лицензии    |

### **Вывод**

Анализируя представленные аналоги, был выявлен ряд преимуществ и недостатков.

В качестве достоинств были выделены разные режимы в приложениях, простой и понятный интерфейс и наличие различных бонусов и подсказок для упрощения прохождения уровней.

Но с другой стороны, в качестве ярких недостатков в некоторых приложениях было выявлено слишком большое количество режимов игры и отсутствие их описания, большое количество навязчивой рекламы.

Из чего можно сделать вывод, что в разрабатываемом приложении хотелось бы видеть несколько режимов игры, простое к восприятию меню, а также возможность пользоваться подсказками. Отсутствие рекламы и других отвлекающих деталей приветствуется.

Исходя из этого были выявлены и описаны требования к разрабатываемому приложению.

Также исходя из анализа и сравнения вышеуказанных платформ для Android (табл. 1) было принято решение разрабатывать приложение при помощи Unity3D.

## 2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

### 2.1. Функциональные и нефункциональные требования

Функциональные требования определяют функциональность программного обеспечения, то есть описывают, какое поведение должна предоставлять разрабатываемая система.

1. Игра должна иметь 2 режима: режим «regular» и режим «training».
2. В режиме «regular» несколько возрастающих по сложности уровней («easy», «medium» и «hard»).
3. В режиме «regular» изначально к прохождению доступен только уровень 1. Следующие уровни становятся доступными после прохождения предыдущих.
4. В режиме «regular» все уровни фиксированные. При повторном прохождении одного и того же уровня пользователь знает, каким образом будут располагаться трубы на экране.
5. В режиме «training» доступно 3 уровня сложности: «easy», «medium» и «hard».
6. В режиме «training» изначально все уровни сложности доступны к прохождению.
7. В режиме «training» уровни генерируются автоматически. Пользователь не знает заранее, каким образом будут располагаться трубы на экране в следующий раз.
8. В случае успешного прохождения пользователем уровня начисляются баллы. За прохождение уровня сложности «easy» пользователь получает 50 баллов; за прохождение уровня сложности «medium» пользователь получает 100 баллов; за прохождение уровня сложности «hard» пользователь получает 150 баллов.
9. В обоих режимах игры для пользователя доступны подсказки «help». Во время игры пользователь может некоторое количество раз воспользоваться подсказкой «help» – любой случайно выбранный элемент на поле примет корректное положение.

10. В зависимости от уровня сложности подсказки «help» имеют определенную стоимость. В уровнях сложности «easy» 1 подсказка «help» стоит 10 баллов; в уровнях сложности «medium» 1 подсказка «help» стоит 20 баллов; в уровнях сложности «hard» 1 подсказка «help» стоит 30 баллов.

11. Пользователь в обоих режимах игры может 1 раз в течение прохождения одного уровня воспользоваться функцией-подсказкой «fix up», которая собирает все поле. Все элементы на поле принимают корректное положение.

12. В зависимости от уровня сложности подсказки «fix up» имеют определенную стоимость. В уровнях сложности «easy» подсказка «fix up» стоит 50 баллов; в уровнях сложности «medium» подсказка «fix up» стоит 100 баллов; в уровнях сложности «hard» подсказка «fix up» – 150 баллов.

13. В игре существует сохранение прерванной игры. Если пользователь прервал игру, то он сможет начать проходить желаемый уровень с начала или же продолжить незаконченный уровень с того места, на котором был прерван.

## **2.2. Нефункциональные требования**

Нефункциональные требования описывают свойства и ограничения, накладываемые на систему.

Для реализации приложения были выявлены следующие требования: приложение Water Pipe должно быть написано на языке C# под платформу Android с помощью платформы Unity3D.

## **2.3. Диаграмма прецедентов**

Диаграмма прецедентов отражает отношения между актерами и прецедентами системы, что позволяет описать систему на концептуальном уровне. Прецедент – это некая функциональная возможность системы, благодаря которой актер (т.е. пользователь), может получить конкретный, измеримый, нужный ему результат.

Прецедент соответствует отдельному сервису системы, определяет один из вариантов ее использования и описывает типичный способ взаимодействия пользователя с системой.

Варианты использования обычно применяются для спецификации внешних требований к системе [10].

На основании функциональных требований к системе с помощью языка графического описания для объектного моделирования UML [6] была создана диаграмма прецедентов, которая представлена на рис. 1.

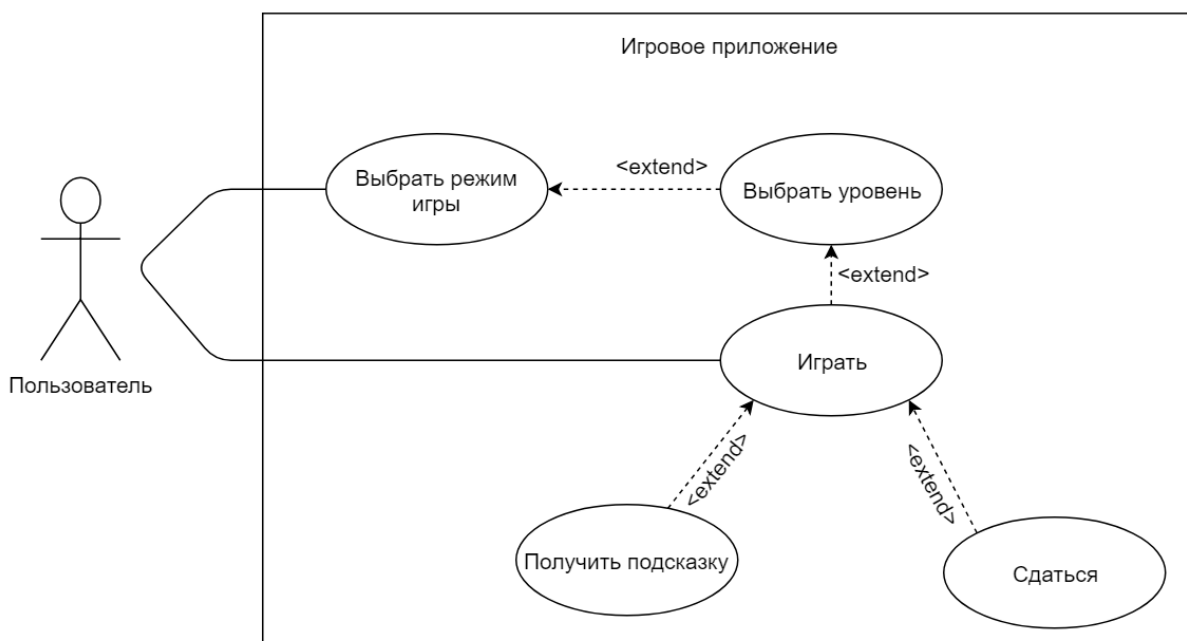


Рис. 1. Диаграмма прецедентов

### Основные актеры, взаимодействующие с системой

В рамках реализуемого приложения предусмотрен один актер – пользователь, который взаимодействует с проигрываемым сюжетом игры и компонентами меню.

### Краткое описание вариантов использования

Пользователь может *выбрать режим игры* – для пользователя предоставляется выбор их двух режимов игры: «regular» или «training».

Пользователь может *играть* – начать проходить уровень сначала или продолжить прерванный уровень.

Пользователь может *выбрать уровень* – выбрать желаемый уровень для дальнейшего прохождения.

Пользователь может *получить подсказку* – один случайный элемент на игровом поле примет корректное положение.

Пользователь может *сдаться* – игровое поле соберется автоматически и игра завершится.

## Проектирование интерфейса

Интерфейс пользователя (GUI — англ. user interface) — совокупность средств, при помощи которых пользователь общается с различными устройствами, чаще всего — с компьютером или бытовой техникой, либо иным сложным инструментарием (системой) [9]. Один из этапов разработки дизайна приложения – это создание макета экранов.

**Экран 1 «MenuScreen»** - стартовый экран при запуске приложения, на котором отображаются иконка «Продолжить», обозначающая продолжение прерванной игры; кнопка «Game mode».

Ниже представлен макет экрана 1 (рис. 2).

При нажатии иконки «Продолжить», обозначающей продолжение прерванной игры, пользователь переходит на игровой экран 5 «GameScreen» (рис. 7). При нажатии «Game mode» осуществляется переход на экран 2 «GameModeScreen» (рис. 3).

**Экран 2 «GameModeScreen»** - экран, появляющийся при нажатии «Game mode» на экране 1. На экране 2 отображаются 3 кнопки: «Regular»; «Training»; иконка «Дом». Ниже представлен макет экрана 2 (рис. 3).

При нажатии «Regular» осуществляется переход на экран 5 «GameScreen» (рис. 6).

При нажатии «Training» осуществляется переход на экран 5 «GameScreen» (рис. 6).

При нажатии иконки «Дом» осуществляется переход на экран 1 «MenuScreen» (рис. 2).





Рис. 2. Макет экрана 1

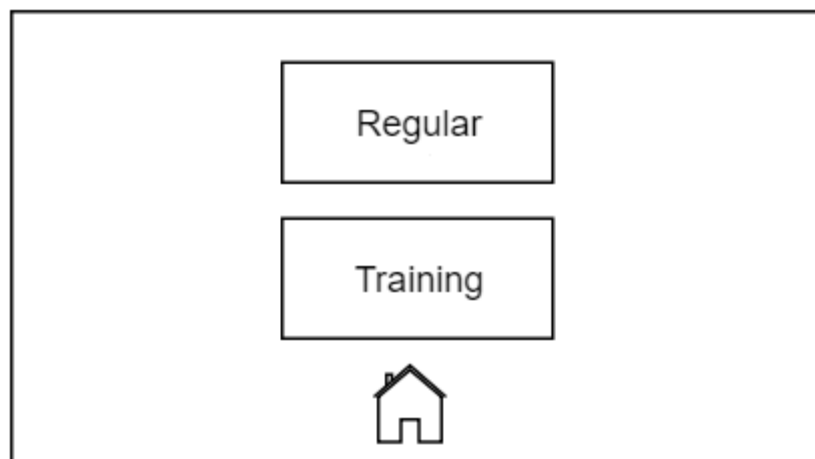


Рис. 3. Макет экрана 2

**Экран 3 «RegularModeScreen»** - экран, появляющийся при нажатии «Regular» на экране 2.

На экране присутствуют 6 иконок с цифрами (цифра обозначают номер уровня), иконка «Назад» и иконка «Дом».

Ниже представлен макет экрана 3 (рис. 4).

При нажатии на иконки с цифрами осуществляется переход на экран 5 «GameScreen» (рис. 6). При нажатии иконки «Назад» осуществляется переход на экран 2 «GameModeScreen» (рис. 3).

При нажатии иконки «Дом» осуществляется переход на экран 1 «MenuScreen» (рис. 2).

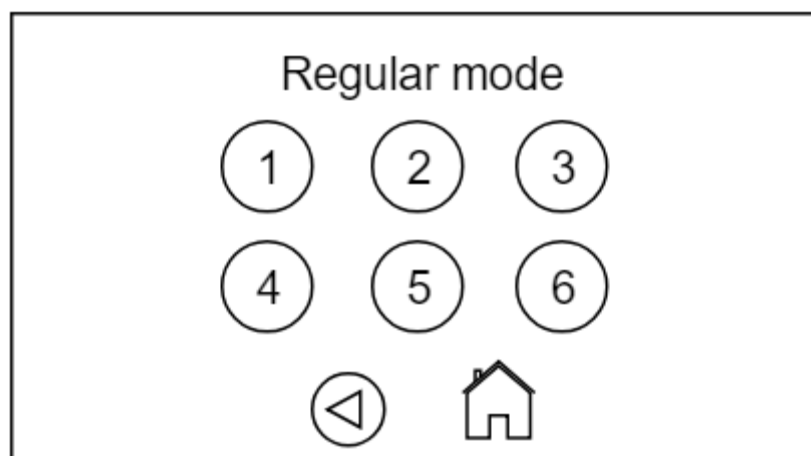


Рис. 4. Макет экрана 3

**Экран 4 «TrainingModeScreen»** - экран, появляющийся при нажатии «Training» на экране 2. На экране 3 кнопки: «Easy», «Medium», «Hard»; иконка «Назад»; иконка «Дом».

Ниже представлен макет экрана 4 (рис. 5).



Рис. 5. Макет экрана 4

При нажатии кнопок «Easy», «Medium», «Hard» осуществляется переход на экран 5 «GameScreen» (рис. 6).

При нажатии иконки «Назад» осуществляется переход на экран 2 «GameModeScreen» (рис. 3).

При нажатии иконки «Дом» осуществляется переход на экран 1 «MenuScreen» (рис. 2).

**Экран 5 «GameScreen»** - экран, появляющийся при нажатии иконок с цифрами в сцене 3 «RegularModeScreen» (рис. 4); при нажатии кнопок «Easy», «Medium», «Hard» на экране 4 «TrainingModeScreen» (рис. 5).

Ниже представлен макет экрана 5 (рис. 6).

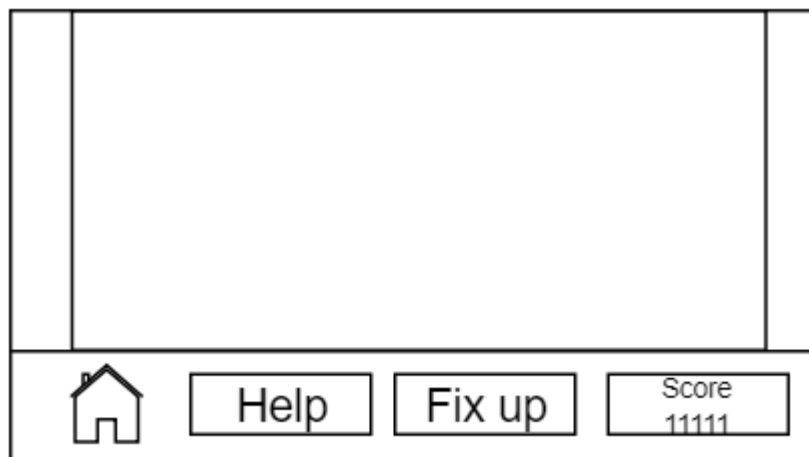


Рис. 6. Макет экрана 5

На экране 5 отображено игровое поле, иконка «Дом», иконка «Score» и 2 кнопки: «Help» и «Fix up». При нажатии иконки «Дом» осуществляется переход на экран 1 «MenuScreen» (рис. 2).

При нажатии кнопки «Help» должен поворачиваться в правильное положение один случайно выбранный элемент на игровом поле.

При нажатии кнопки «Fix up» все элементы на игровом поле должны принимать корректное положение. Иконка «Score» носит только информативный характер и показывает текущее количество бонусов.

Для создания дизайна приложения оптимальнее всего использовать векторный графический редактор.

Векторные изображения (иногда называемые векторными фигурами или векторными объектами) состоят из линий и кривых, определяемых векторами – математическими объектами, которые описывают изображение в соответствии с его геометрическими характеристиками. Векторные изображения можно свободно перемещать и изменять без потери детализации и четкости, поскольку такие изображения не зависят от разрешения.

Края изображений остаются четкими при изменении размера, печати на принтере PostScript, сохранении в файле PDF и импорте в векторные графические приложения.

Таким образом, векторные изображения – это наилучший выбор для иллюстраций, которые выводятся на различные носители и размер которых приходится часто изменять [12].

### **Вывод**

В ходе анализа были установлены основные функциональные и нефункциональные требования к системе, определены пользователь и варианты использования системы.

Для создания дизайна приложения было принято решение использовать программу Adobe Illustrator – векторный графический редактор, разработанный и распространяемый фирмой Adobe Systems.

С итоговым дизайном приложения можно ознакомиться в приложении 1.

### **3. РЕАЛИЗАЦИЯ КОМПОНЕНТОВ СИСТЕМЫ**

#### **3.1. Инструменты, используемые для реализации**

Для реализации приложения для ОС Android был выбран язык программирования C# и среда разработки Unity3D. Среда разработки Unity3D – межплатформенная среда разработки компьютерных игр.

Язык программирования C# – объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework [8].

#### **Файловая структура**

В разработанной приложении содержится ряд каталогов, в которых хранятся:

- 1) шрифты;
- 2) префабы;
- 3) ресурсы;
- 4) сцены;
- 5) скрипты;
- 6) спрайты;
- 7) дополнительные библиотеки.

Файловая структура представлена на рис. 7.

В каталоге Fonts хранятся шрифты, применяемые в игре. В каталоге Prefabs хранятся компоненты, которые используются в игре более 1 раза.

В каталоге Resources хранятся документы формате json с информацией о расположении элементов на поле для уровней в режиме игры «Regular». В каталоге Scenes содержатся 2 сцены игры: «MenuScene» и «GameScene».

В каталоге Scripts содержатся все скрипты с описанием всех классов и взаимодействий. В каталоге Sprites находятся 2 папки: «Old» и «Previews»,

в которых содержатся картинки и иконки - элементы пользовательского интерфейса. В каталоге UnityPackages содержатся дополнительные библиотеки.

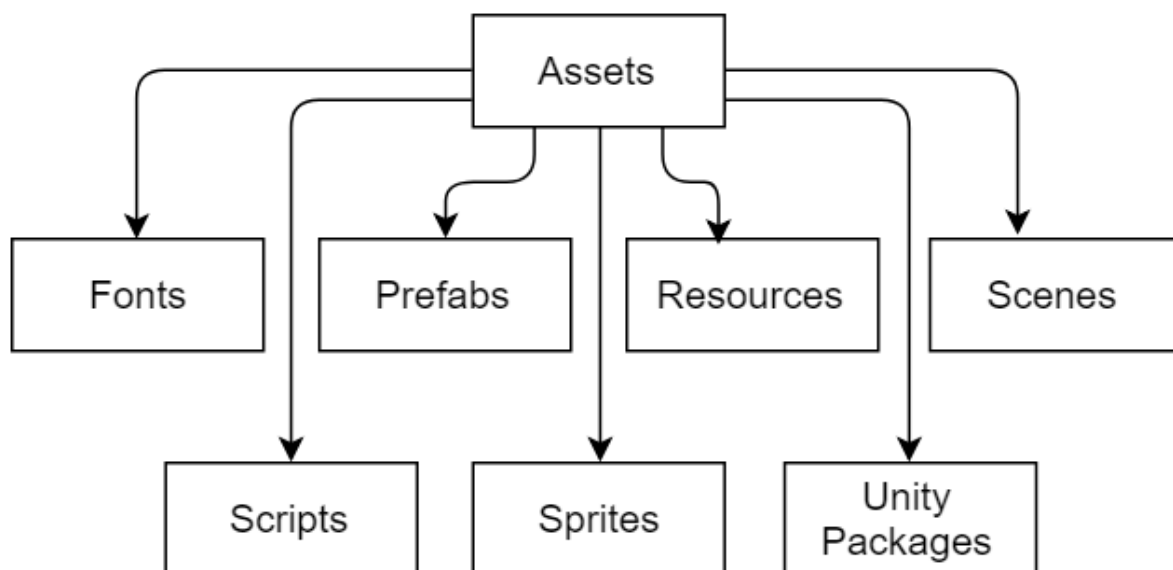


Рис. 7. Файловая структура

### 3.3. Реализация компонентов системы

Реализация некоторых основных функций программы представлена в листинге 1.

В листинге 1 представлен код описания запуска игры. В зависимости от выбранного уровня сложности задаются определенные параметры высоты и ширины игрового поля и загружается игровая сцена.

#### Листинг 1. «Описание запуска игры»

```
// Функция запуска легкого уровня игры
public void startEasyGame() {
    FieldController.width = 12;
    FieldController.height = 6;
    startGame ();
}
// Функция запуска среднего уровня игры
public void startMediumGame() {
    FieldController.width = 20;
    FieldController.height = 10;
    startGame ();
}
// Функция запуска сложного уровня игры
public void startHardGame() {
    FieldController.width = 35;
    FieldController.height = 16;
    startGame ();
}
```

```

}
// Функция запуска игры
private void startGame() {
    SceneManager.LoadScene ("GameScene", LoadSceneMode.Single);
}

```

В листинге 2 представлена реализация прогресса прохождения уровней в режиме игры «Regular». После прохождения одного уровня игры в режиме «Regular» становится доступным следующий уровень.

### Листинг 2. «Прогресс прохождения уровней в режиме игры “Regular”»

```

// сохранение прогресса прохождения уровней
private void SaveLevelsProgress()
{
    if (LevelController.CurrentLevel != -1)
        LevelController.MaxLevel = Mathf.Max(LevelController.MaxLevel, LevelController.CurrentLevel + 1);
}

```

В листинге 3 представлена реализация сохранения труб, номеров и сложности игры. Создается файл с информацией о текущем состоянии игры (положение труб на экране, номер и сложность уровня).

### Листинг 3. «Сохранение труб, номеров и сложности игры»

```

// сохранение труб, номера и сложности уровня
public static void Save(Pipe[,] pipes)
{
    PipeData[,] data = new PipeData[pipes.GetLength(0), pipes.GetLength(1)];

    for (int i = 0; i < pipes.GetLength(0); ++i)
        for (int j = 0; j < pipes.GetLength(1); ++j)
            data[i, j] = pipes[i, j] == null ? null : pipes[i, j].GetSerializableInfo();
    string json = JsonConvert.SerializeObject(data);
    PlayerPrefs.SetString("Pipes", json);
    PlayerPrefs.SetInt("CurrentLevel", LevelController.CurrentLevel);
    PlayerPrefs.SetInt("CurrentDifficulty", LevelController.CurrentDifficulty);
    PlayerPrefs.Save();
}

```

В листинге 4 представлена реализация восстановления труб и настроек игры из сохранения.

При использовании возможности продолжения прерванной игры поле генерируется исходя из заранее сформированного файла с информацией о состоянии прерванной игры (положение труб на экране, номер и сложность уровня).

#### Листинг 4. «Восстановление труб и настроек игры из сохранения»

```
// функция восстановления всех труб и настроек из сохранения
void LoadGame()
{
    cleanContent();

    FieldGenerator generator = GetComponent<FieldGenerator>();
    PipeData[,] data = SaveLoad.Load();
    width = data.GetLength(0);
    height = data.GetLength(1);
    contentWidth = width;
    contentHeight = height;

    content = generator.generateContent(data);

    updateCamera();
}
```

В листинге 5 представлена реализация проверки состояния игры (собрано игровое поле полностью или нет) и контроль начисления и списания баллов. В зависимости от текущего состояния и уровня сложности на экране у игрока отображается различное количество баллов за подсказки и текущий счет игры.

#### Листинг 5. «Проверка состояния игры и контроль начисления и списания баллов»

```
void Update()
{
    // Проверка состояния игры

    if (gameIsRunning && field.isComplete())
    {
        field.blockPipes();
        showVictoryMenu();
    }

    ScoreLabel.text = string.Format("score\n{0}", ScoreController.Score);
    HelpPriceLabel.text = string.Format("-{0}", HelpPrice * LevelController.CurrentDifficulty);
    FixPriceLabel.text = string.Format("-{0}", FixPrice * LevelController.CurrentDifficulty);
}
```

В листинге 6 представлена реализация подсказки «Help». При возникновении затруднения поиска решения пользователь может воспользоваться подсказкой «Help». Происходит проверка поля. Алгоритм пропускает незаполненные ячейки (ячейки без труб) и корректно повернутые трубы. Далее с использованием рандома находится некорректно повернутая труба, которая возвращается в корректное положение.



## Листинг 6. «Реализация подсказки “Help”»

```
// Функция подсказки - установка случайной трубы в правильное положения
public void fixPipe()
{
    // Шанс выбора рассматриваемой трубы в качестве исправляемой
    int chance = 1;
    Pipe pipe = null;
    // Перебор игрового поля
    for (int x = 0; x < width; x += 1)
    {
        for (int y = 0; y < height; y += 1)
        {
            if (content [x, y] == null) {
                continue;
            }

            if (content [x, y].isCorrect ()) {
                continue;
            }

            if (Random.Range (0, chance) == 0) {
                pipe = content [x, y];
            }
            // Уменьшение шансов для каждой последующий рассматриваемой трубы
            chance += 1;
        }
    }
    // Исправление трубы
    if (pipe != null) {
        pipe.fix ();
    }
}
```

В листинге 7 представлена реализация функции подсказки «fix up» – исправление всех труб. Происходит проверка всего игрового поля. Все неправильно повернутые трубы принимают корректное состояние.

## Листинг 7. «Реализация функции исправления всех труб»

```
// Функция исправления всех труб (решение игрового поля)
public void fixAllPipes()
{
    for (int x = 0; x < width; x += 1)
    {
        for (int y = 0; y < height; y += 1)
        {
            if (content [x, y] != null) {
                content [x, y].fix ();
            }
        }
    }
}
```

#### 4. ТЕСТИРОВАНИЕ

Для тестирования системы применялось функциональное тестирование.

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определенных условиях решать задачи, нужные пользователям.

Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает [19].

Ниже в табл. 2 представлено тестирование системы: название теста, шаги, необходимые для проверки, ожидаемый результат и его подтверждение или опровержение.

В результате проведения тестирования программа показала корректную работу. Все задачи решены верно.

Табл. 2. Тестирование системы

| № | Название теста                | Шаги  | Ожидаемый результат   | Тест пройден? |
|---|-------------------------------|---|---|---------------|
| 1 | Отображение «MenuScene»       | Запустить приложение                        | Отображение главного меню игры  | Да            |
| 2 | Продолжить незаконченную игру | Нажать на иконку «Продолжить» в «MenuScene» | Осуществляется переход в сцену «GameScene», отображение элементов на поле игры в том состоянии, в котором эти элементы остались при выходе из игры. | Да            |

| № | Название теста                       | Шаги   | Ожидаемый результат   | Тест пройден? |
|---|--------------------------------------|--|---|---------------|
| 3 | Выбор режима игры                    | Нажать на кнопку «Game mode» в сцене MenuScene         | Переход на экран GameModeScreen                                       | Да            |
| 4 | Выбор режима игры «Regular»          | Нажать на кнопку «Regular» на экране GameModeScreen    | Переход на экран RegularModeScreen                                    | Да            |
| 5 | Выбор режима игры «Training»         | Нажать на кнопку «Training» на экране GameModeScreen   | Переход на экран TrainingModeScreen                                   | Да            |
| 6 | Выбор уровня в режиме игры «Regular» | Нажать на иконку с цифрой на экране RegularModeScreen  | Переход на сцену GameScene  | Да            |
| 7 | Выбор уровня «Easy»                  | Нажать на кнопку «Easy» на экране TrainingModeScreen   | Переход на сцену GameScene. Игровое поле генерируется со сложностью 1 | Да            |
| 8 | Выбор уровня «Medium»                | Нажать на кнопку «Medium» на экране TrainingModeScreen | Переход на сцену GameScene. Игровое поле генерируется со сложностью 2 | Да            |
| 9 | Выбор уровня «Hard»                  | Нажать на кнопку «Hard» на экране TrainingModeScreen   | Переход на сцену GameScene. Игровое поле генери-                      | Да            |

| №  | Название теста                                       | Шаги  | Ожидаемый результат   | Тест пройден? |
|----|--|---|---|---------------|
|    |  |   | руется со сложностью 3  |               |
| 10 | Корректная работа иконки «Дом»                       | Нажать на иконку «Дом» на экране GameModeScreen       | Переход на сцену MenuScene  | Да            |
| 11 | Корректная работа иконки «Дом»                       | Нажать на иконку «Дом» на экране RegularModeScreen    | Переход на сцену MenuScene  | Да            |
| 12 | Корректная работа иконки «Дом»                       | Нажать на иконку «Дом» на экране TrainingModeScreen   | Переход на сцену MenuScene  | Да            |
| 13 | Корректная работа иконки «Дом»                       | Нажать на иконку «Дом» в сцене GameScene              | Переход на сцену MenuScene  | Да            |
| 14 | Корректная работа иконки «Назад»                     | Нажать на иконку «Назад» на экране RegularModeScreen  | Переход на экран GameModeScreen                                       | Да            |
| 15 | Корректная работа иконки «Назад»                     | Нажать на иконку «Назад» на экране TrainingModeScreen | Переход на экран GameModeScreen                                       | Да            |
| 16 | Проверка корректной работы кнопки «Help» в GameScene | Нажать на кнопку «Help» в GameScene                   | Случайно выбранная неправильно повернутая труба поворачивается в пра- | Да            |

| №  | Название теста  | Шаги   | Ожидаемый результат  | Тест пройден? |
|----|---|--|--|---------------|
|    |   |  | <p>вильное состояние. Списывается корректное количество баллов из текущего количества баллов. При нехватке баллов использование подсказки «Help» недоступно.</p>                                 |               |
| 17 | <p>Проверка корректной работы кнопки «Fix up» в сцене GameScene</p> | <p>Нажать на кнопку «Fix up» в сцене GameScene</p> | <p>Все трубы поворачиваются в правильное состояние. Списывается корректное количество баллов из текущего количества баллов. При нехватке баллов использование подсказки «Fix up» недоступно.</p> | Да            |
| 18 | <p>Поворот трубы при нажатии на нее</p>                             | <p>Нажать на трубу</p>                             | <p>Труба поворачивается</p>  | Да            |

## **ЗАКЛЮЧЕНИЕ**

Целью работы являлась разработка приложения в жанре «головоломка» для ОС Android.

Для достижения данной цели мною были решены следующие задачи:

- 1) изучены существующие методы разработки мобильных игр для ОС Android;
- 2) выполнен анализ требований;
- 3) выполнено проектирование интерфейса компьютерной игры;
- 4) реализована компьютерная игра для ОС Android;
- 5) проведено тестирование разработанной игры.

## СПИСОК ЛИТЕРАТУРЫ

1. Corona. [Электронный ресурс] URL: <https://ru.coronalabs.com/learn/> (дата обращения: 11.02.2018).

2. Komatineni S., MacLean D., Hashimi S. Pro Android 3. – USA: Apress, 2011. – 1200 с.

3. Unity User Manual [Electronic resource] URL: <https://docs.unity3d.com/ru/current/Manual/UnityManual.html> (date of access: 11.02.2018).

4. Unreal Engine Features. [Electronic resource] URL: <https://www.unrealengine.com/en-US/features> (date of access: 11.02.2018).

5. Варакин М.В. Разработка мобильных приложений под Android. – М.: УЦ «Специалист», 2012. – 128 с.

6. Гома Х. UML-проектирование систем реального времени, распределенных и параллельных приложений. – М.: ДМК Пресс, 2011. – 704 с.

7. Идеальная платформа для разработчиков игр – Android [Электронный ресурс] URL: <http://apps4all.ru/post/06-25-14-idealnaya-platforma-dlya-razrabotchikov-igr-android> (дата обращения: 01.03.2018).

8. Либерти Д. Язык программирования С# // Программирование на С#. – С

а 9. Новожилова Н. Особенности проектирования дружественных интерфейсов. [Электронный ресурс] URL:

<https://cyberleninka.ru/article/n/osobennosti-proektirovaniya-druzhestvennyh-interfeysov-dlya-polzovateley-ekonomistov> (дата обращения: 10.04.2018).

- 10. О

П 11. Основы UML – диаграммы использования. [Электронный ресурс] URL: <https://pro-prof.com/archives/2594> (дата обращения: 07.04.2018).

т 12. О

е 13. П

Р

Б

Д

Р

Ж

14. Приложение в Google Play «LOLO: Puzzle Game». [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=com.gamelolo.game&hl=ru> (дата обращения: 03.03.2018).

15. Приложение в Google Play «Plumber 2 ». [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=com.tedrasoft.plumber.expert> (дата обращения: 03.03.2018).

16. Приложение в Google Play «Слово за слово». [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=me.incrdbl.android.wordbyword> (дата обращения: 03.03.2018).

17. Рынок мобильных приложений в России и мире. [Электронный ресурс] URL: [http://www.json.ru/files/reports/2013-03-12\\_Mobile\\_Apps\\_MW.pdf](http://www.json.ru/files/reports/2013-03-12_Mobile_Apps_MW.pdf) (дата обращения: 10.02.2018).

18. Сайт Android Studio. [Электронный ресурс] URL: <https://developer.android.com/studio/features.html> (дата обращения: 10.02.2018).

19. Тестирование программного обеспечения – основные понятия и определения. [Электронный ресурс] URL: <http://www.protesting.ru/testing/> (дата обращения: 10.05.2018).

20. Цехнер М. Программирование игр под Android. – Пер с англ. – СПб.: ООО Издательство «Питер», 2013. – 688 с.



## ПРИЛОЖЕНИЕ

На рисунках 1-10 представлены скриншоты разработанной мобильной игры.

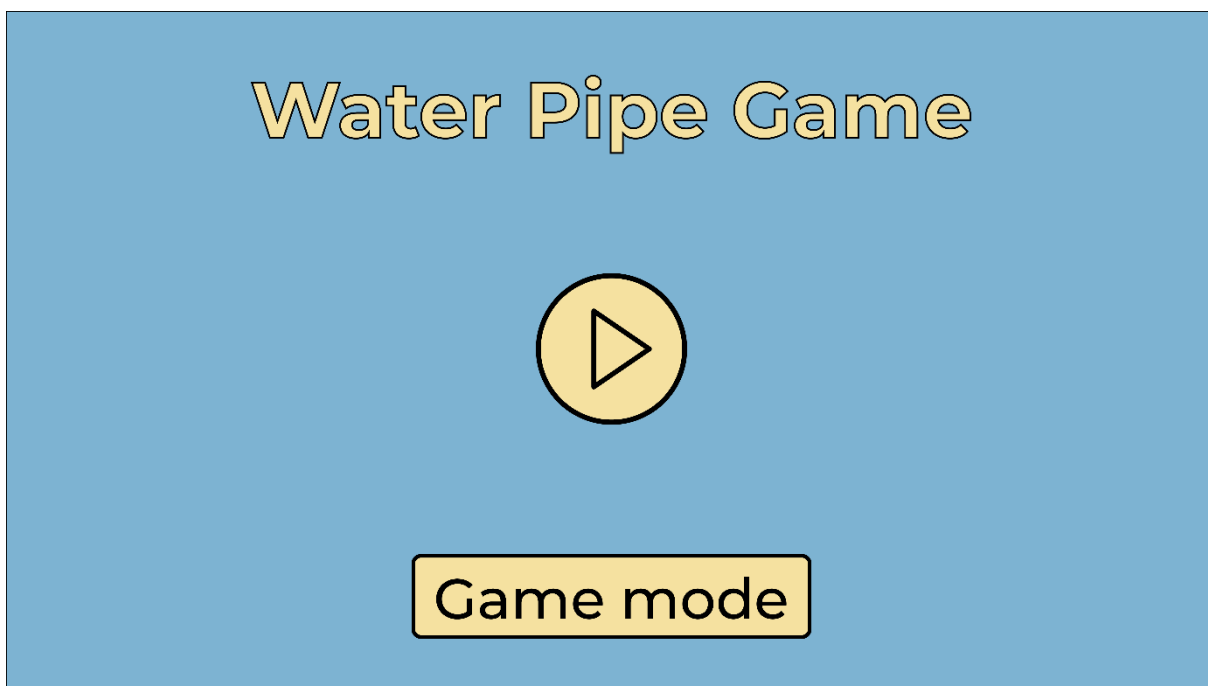


Рис. 1. MenuScene

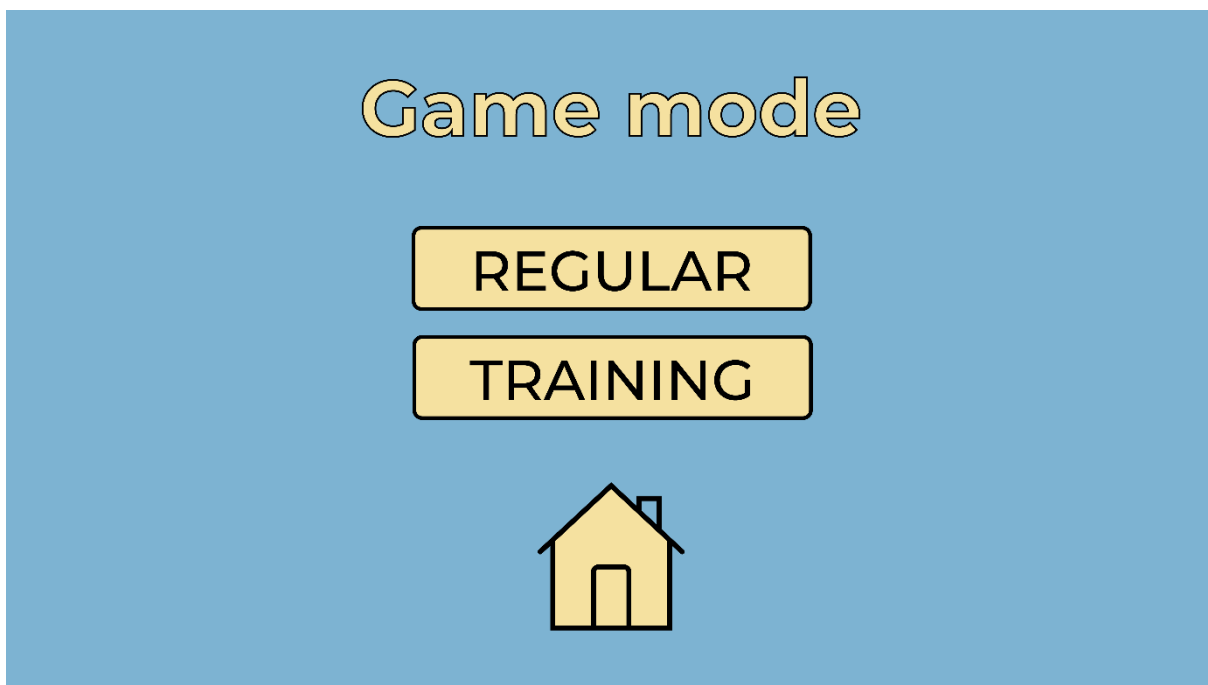


Рис. 2. GameMode

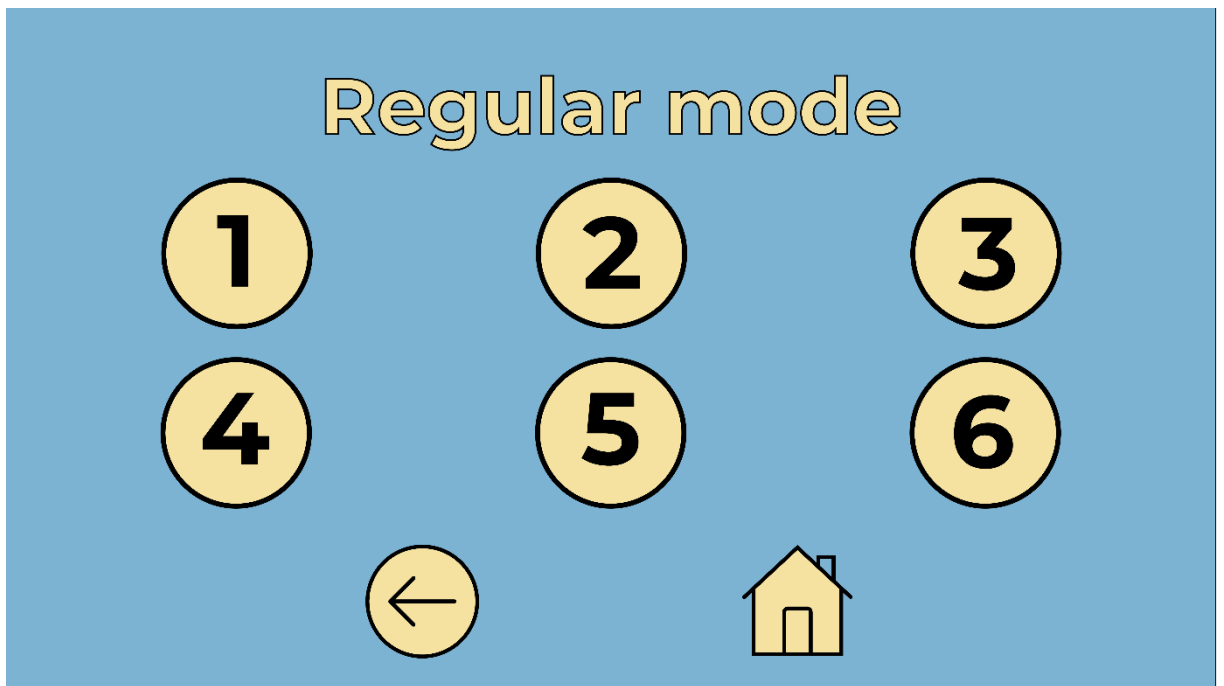


Рис. 3. RegularMode

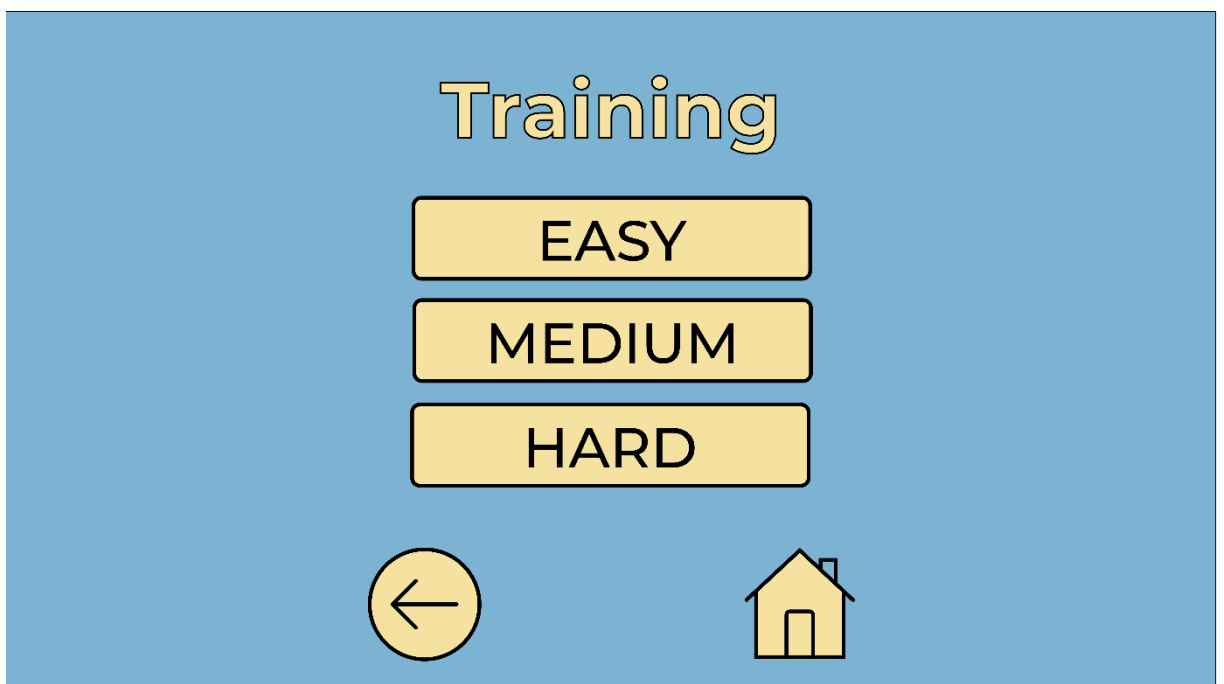


Рис. 4. TrainingMode

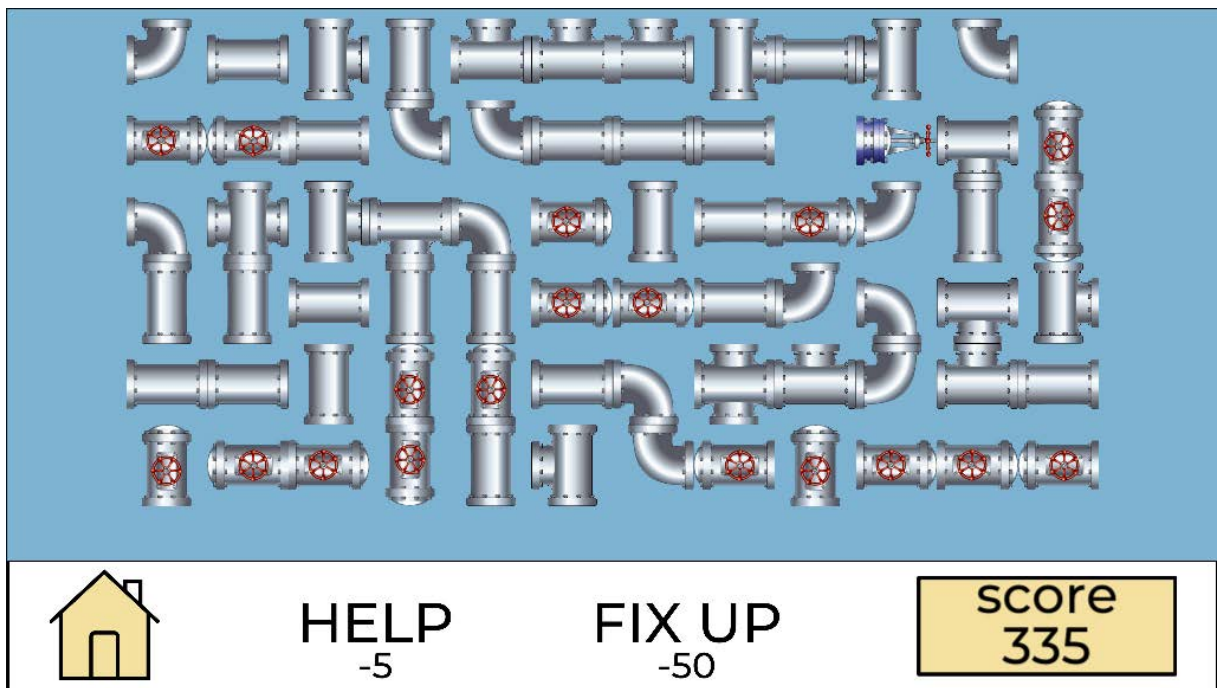


Рис. 5. Несобранный уровень сложности «Easy»

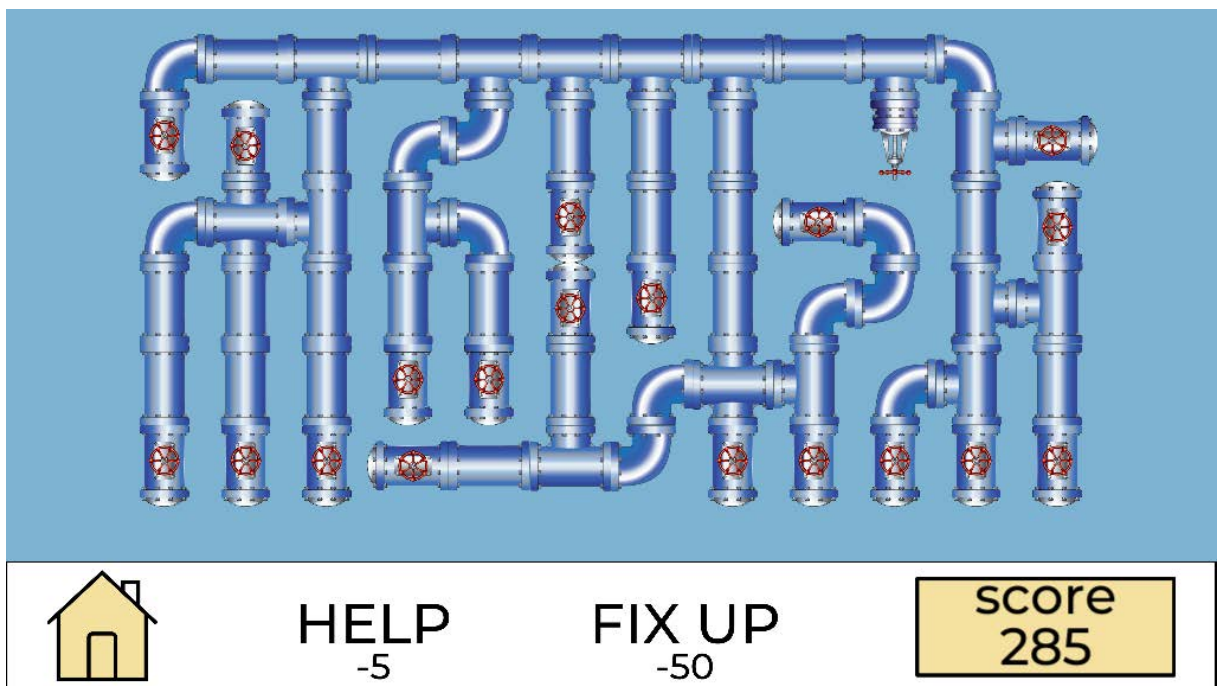


Рис. 6. Собранный уровень сложности «Easy»

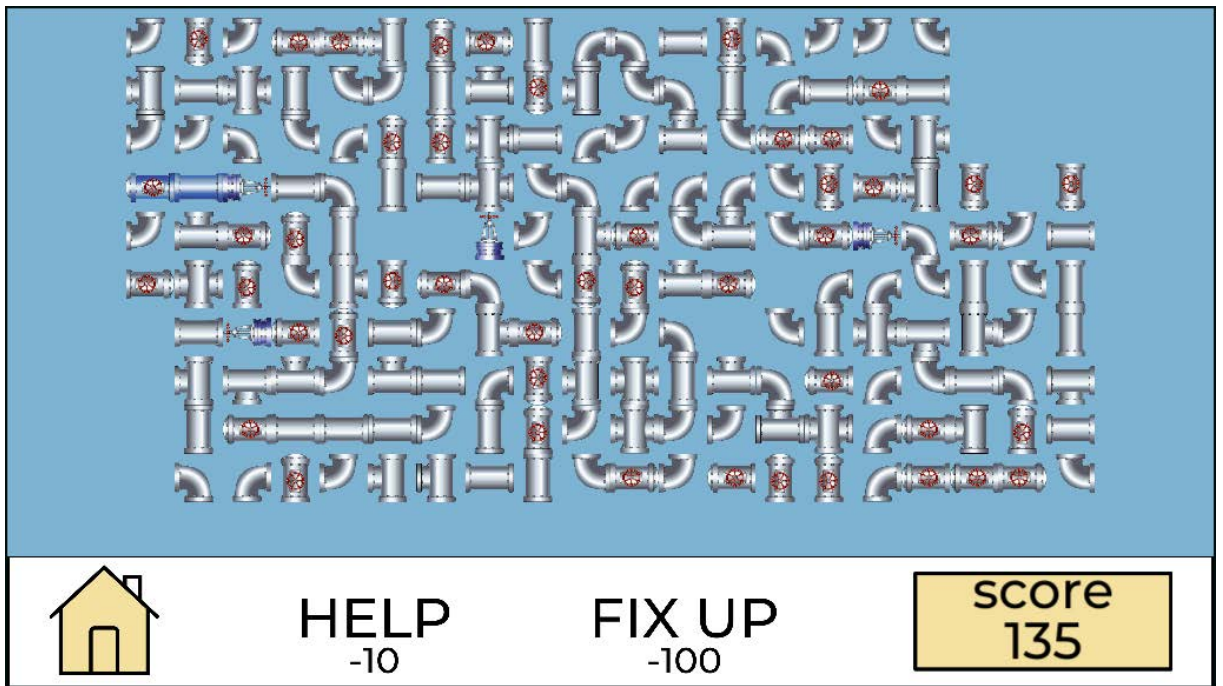


Рис. 7. Несобранный уровень сложности «Medium»

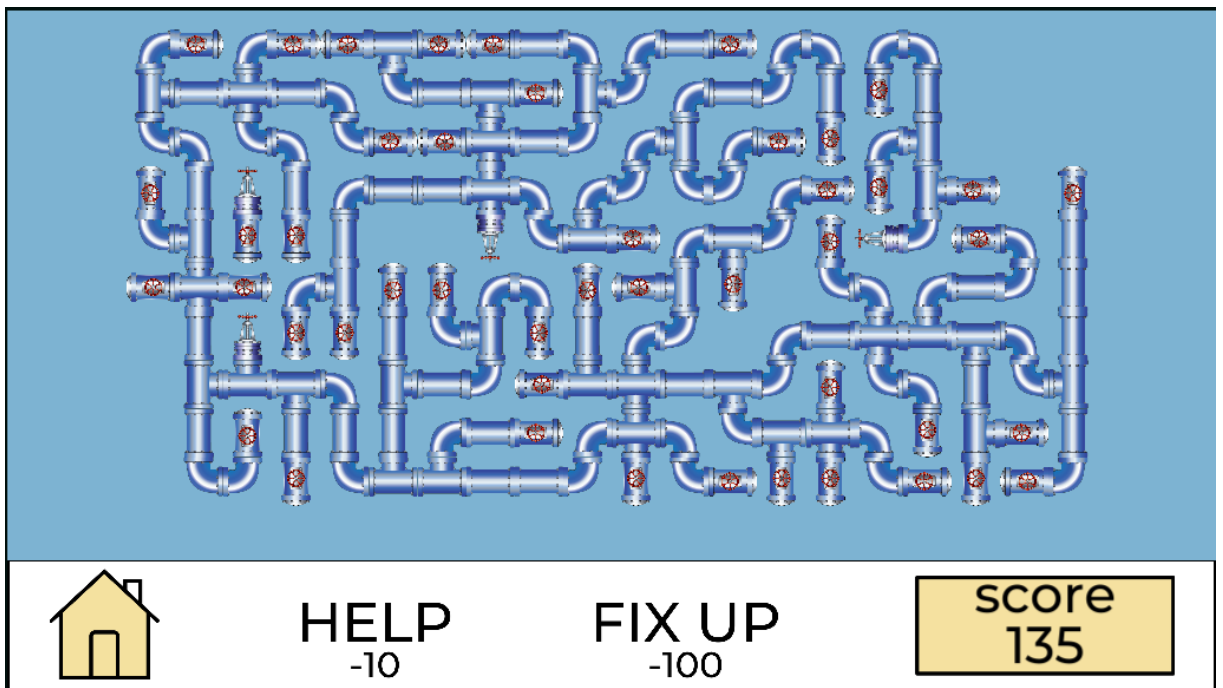


Рис. 8. Собранный уровень сложности «Medium»

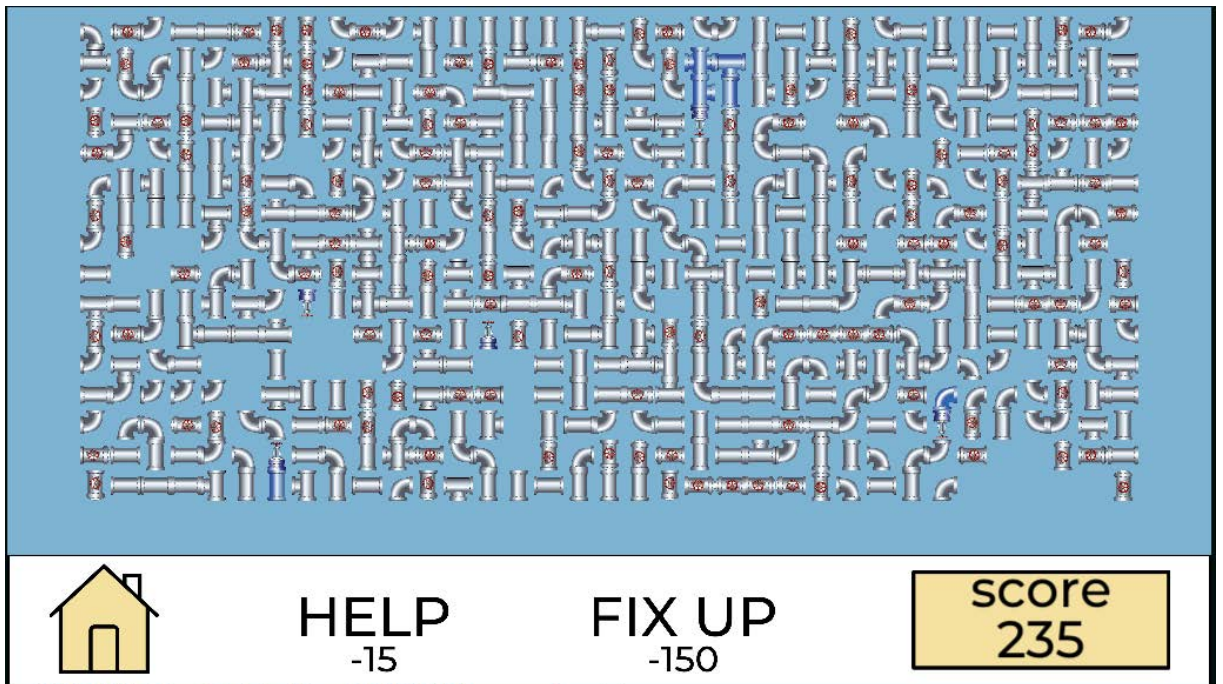


Рис. 9. Несобранный уровень сложности «Hard»

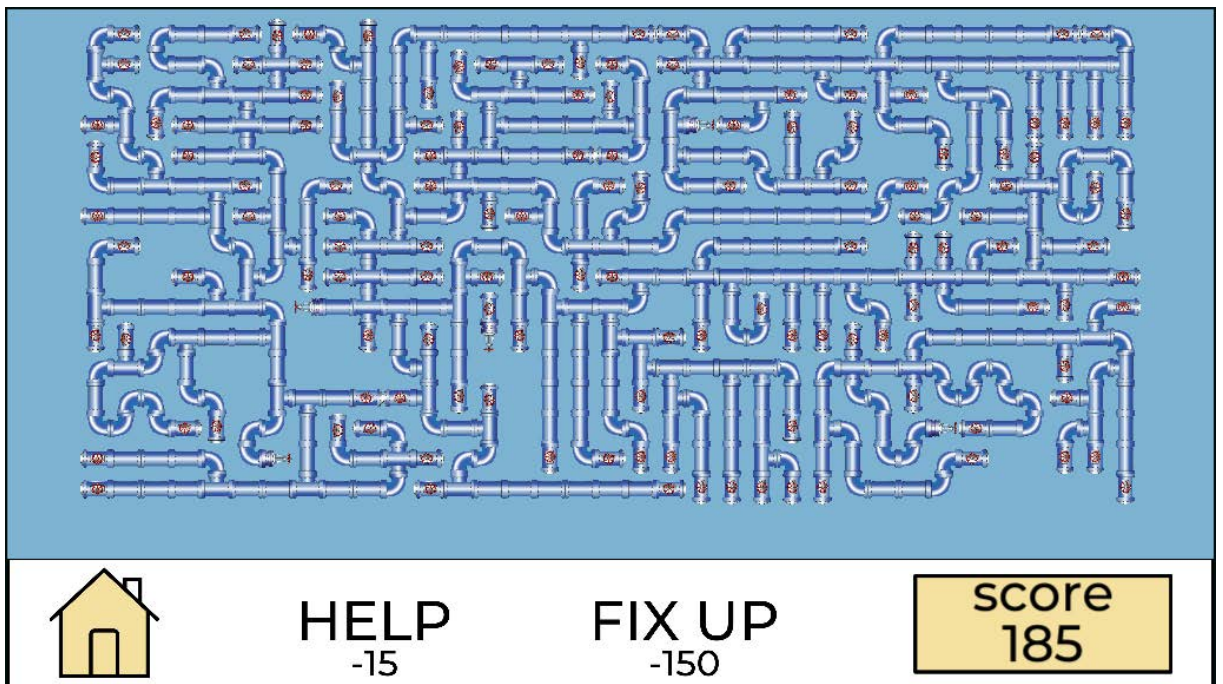


Рис. 10. Собранный уровень сложности «Hard»