

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.42:621.395.721.5

ПЯТОСИН
Андрей Владимирович

**МЕТОДЫ И АЛГОРИТМЫ ДЛЯ ЛОГИЧЕСКОГО ИГРОВОГО
ПРИЛОЖЕНИЯ НА БАЗЕ ANDROID**

АВТОРЕФЕРАТ

на соискание степени магистра технических наук
по специальности 1-38 80 04 Технология приборостроения

Научный руководитель
Цырельчук Игорь Николаевич
кандидат технических наук,
доцент

Минск 2015

Работа выполнена на кафедре проектирования информационно-компьютерных систем учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»

Научный руководитель:

Цырельчук Игорь Николаевич,
заведующий кафедрой проектирования информационно-компьютерных систем учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент

Рецензент:

Дик Сергей Константинович,
проректор по учебной и воспитательной работе учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат физико-математических наук, доцент

Защита диссертации состоится «22» июня 2015 г. года в 9⁰⁰ часов на заседании Государственной комиссии по защите магистерских диссертаций в учреждении образования «Белорусский государственный университет информатики и радиоэлектроники» по адресу: 220013, г.Минск, ул. П.Бровки, 6, 1 уч. корп., ауд. 415, тел.: 293-89-92, e-mail: kafei@bsuir.by.

С диссертацией можно ознакомиться в библиотеке учреждения образования «Белорусский государственный университет информатики и радиоэлектроники».

ВВЕДЕНИЕ

Существует область человеческой деятельности, в которой лабиринты являются необходимым атрибутом – это компьютерные игры. Действительно, почти все игры приключенческих жанров и аркады используют лабиринты. Встречаются они и в других жанрах: логические игры, экшен, изредка в симуляторах. В связи с этим у пользователя компьютера появляются две задачи: прохождение лабиринтов в игровых программах и создание лабиринтов для «своих» программ.

Индустрия компьютерных игр является, пожалуй, наиболее быстро развивающейся и приносящей наибольшие прибыли разработчикам. Компьютерные игры предъявляют очень высокие требования к производительности компьютера, любые новшества в области аппаратного обеспечения тут же используются при разработке игр. Несмотря на кажущуюся несерьезность компьютерных игр, для их создания используется множество сложных и хитроумных алгоритмов и подходов. Это связано с тем, что компьютерные игры основаны на компьютерной графике, которая сложна сама по себе.

По мере роста продаж мобильных устройств во всем мире, растет и спрос на различные приложения для них. Каждая крупная компания стремится иметь хотя бы одно мобильное приложение, чтобы находиться всегда рядом со своим клиентом. В то же время, существование некоторых компаний и вовсе сложно представить без мобильных устройств и специализированных программ, при помощи которых можно, например, управлять базами данных или следить за состоянием своего продукта на рынке в любой момент времени.

Смартфоны становятся новой мобильной игровой платформой, соревнуясь с классическими карманными игровыми системами вроде *Nintendo DS* или *Playstation Portable*.

Платформа *Android* не имеет ограничений и полностью открыта для свободного пользования. Этот факт делает разработку приложений для *Android* абсолютно безграничной и максимально открытой для нововведений. Иными словами, характер разработки программ под *Android* может быть ограничен только фантазией. Более того, все приложения на базе *Android*, можно с легкостью настроить по усмотрению пользователя. Пластичность в настройке – огромный плюс в работе с приложениями для *Android*.

Идея разрабатываемого логического игрового приложения на базе *Android* является генерация и прохождение лабиринта. Эта задача является одной из задач оптимизации. Задача поиска пути, так или иначе, в той или иной мере встречается в большинстве современных игровых жанров. Так как игры становятся всё сложнее, то поиск пути также эволюционирует и развивается вместе с ними.

Часто оптимизационные задачи можно свести к формализованному виду и взаимосвязь составных частей математической модели представить в виде графа. Такой подход позволяет использовать алгоритмы и средства теории графов в процессе поиска оптимальных решений и минимизации аналитиче-

ских моделей критериев оптимальности. Решение задачи тогда сводится к поиску кратчайших путей между вершинами графа.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования

В наше время сложно представить себе человека без сотового телефона, планшетного компьютера, смартфона или любого другого портативного мультимедийного устройства. Мы привыкли к тому, что всегда под рукой не только средство связи, но и множество полезных функций, таких как: калькулятор, органайзер, конвертер, календарь, часы.

Мобильные устройства завоевывают всё большую популярность среди пользователей. Доминирующей на рынке мобильной платформой является операционная система *Android*. Её доля составляет более 59%. Согласно последнему отчету исследовательской компании *Gartner*, через несколько лет платформа *Android* может стать самой популярной операционной системой в мире, что является хорошим поводом для изучения как самой системы, так и сред разработки под неё.

Тот факт, что выделенные устройства, такие как мобильные телефоны, станут в один ряд с другими крупными платформами общего назначения очевиден. Эта новая тенденция открывает доступ к мобильным устройствам для традиционных языков программирования, благодаря чему диапазон применения мобильных приложений и их доля на рынке растут.

Таким образом, актуальность темы исследования обусловлена необходимостью решения ряда методических и практических задач, связанных с разработкой и сопровождением приложений на базе операционной системы *Android*, так как ближайшее время они будут наиболее востребованы на рынке программного обеспечения.

Степень разработанности проблемы

Основные теоретические и практические вопросы и особенности решения задач оптимизации раскрыты в трудах зарубежных ученых – Хемди Ф., Кини Р.Л., Райфа Х., Райта М., и др. Особенности реализации относительно теории графов нашли отражение в работах Дистеля Р., Басакера Р., Зыкова А.А., Оре О., Богомолова А.М., Салия В.Н., Уилсона Р., Харари Ф. и др.

Вместе с тем, материал, основывающийся на создании именно лабиринтов практически отсутствует, что вызывает необходимость проведения работы по разработке механизма их формирования. Это позволило определить цель исследования, задачи, предмет и объект.

Цель и задачи исследования. Целью магистерской диссертации является разработка логического игрового приложения на базе *Android*.

Для выполнения поставленной цели в работе были сформулированы следующие задачи:

- изучить процесс разработки приложений на базе *Android*;

- исследовать методы создания лабиринтов;
- проанализировать методы нахождения кратчайшего пути;
- разработать интуитивно-понятный интерфейс приложения.

Объектом исследования является логическое игровое приложение на базе *Android*.

Предметом исследования являются методы теории вероятностей, теории игр, математической логики и теории графов.

Область исследования. Содержание диссертационной работы соответствует образовательному стандарту высшего образования второй ступени (магистратуры) специальности 1-38 80 04 Технология приборостроения.

Теоретическая и методологическая основа исследования

В основу диссертации легли теоретические и практические положения отечественных и зарубежных исследований ученых по проблеме задач оптимизации: А.Ш. Абакарова, И.Л.Акулича, И.В. Гираснова, Ф. Гилла, У. Мюррея и др.

Теоретической основой исследований, проведенных в работе, являются общенаучные методы структурного, факторного, функционального и сравнительного анализов, также применялись методы нахождения оптимального пути, методы численных расчетов, методы теории игр и теории графов.

Методологической основой исследования являются разработки отечественных и зарубежных авторов, методические материалы, труды отечественных и зарубежных учёных и научные труды в области теории графов. Инновационные подходы исследуется в рамках структурного и компонентного подходов, при создании инструментального средства используется методология системного проектирования, графические нотации. В диссертации используется системный подход к разработке архитектуры инструментального средства из области научных информационных систем.

Ядро применяемого математического аппарата составляют методы нахождения оптимального пути и теории графов. Инструментальное средство является практической реализацией предложенных математических алгоритмов и моделей.

Информационная база исследования сформирована на базе сведений, представленных в научных изданиях, ресурсах Интернета, а также материалов конференций и семинаров.

Инструментальной базой исследования являются *LibGDX – Java Framework* для разработки игр, который позволяет создавать игры, работающие одновременно на многих платформах, *UML (Unified Modeling Language)* диаграммы. В качестве среды разработки используется *Android Studio*.

Научная новизна и значимость полученных результатов заключается в разработке игрового логического приложения на базе *Android*, демон-

стрирующего создание лабиринтов задаваемых размеров методами Прима, рекурсивного спуска и «охотья и убивай», а так же предоставляющего возможность их прохождения.

Основные положения, выносимые на защиту:

- классификация, основанная на анализе существующих алгоритмов создания и нахождения кратчайшего пути в лабиринтах;
- логическое игровое приложение, работающее на операционной системе *Android* отвечающее за построение лабиринтов и предоставляющее возможность его прохождения.

Теоретическая значимость диссертации заключается в том, что полученные результаты отображают основные преимущества и недостатки рассматриваемых алгоритмов создания лабиринтов и демонстрируют как эти алгоритмы реализуются при создании приложения на базе *Android*.

Практическая значимость диссертации заключается в том, что полученные результаты помогают разработчикам программ, в которых необходимо создание лабиринтов, в выборе оптимального для их цели метода создания, прохождения и реализации лабиринта.

Апробация и внедрение результатов исследования

Основные положения работы и результаты диссертации изложены в четырёх опубликованных работах общим объемом 4 страницы.

Структура и объем работы. Структура диссертационной работы обусловлена целью, задачами и логикой исследования. Работа состоит из введения, трёх глав и заключения, библиографического списка и приложений. Общий объем диссертации – 67 страниц. Работа содержит 40 иллюстраций. Библиографический список включает 85 наименований.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** рассмотрены основные направления исследований, а также дается обоснование актуальности темы диссертационной работы.

В **общей характеристике работы** сформулированы ее цель и задачи, показана связь с научными программами и проектами, даны сведения об объекте исследования и обоснован его выбор, представлены положения, выносимые на защиту, приведены сведения о личном вкладе соискателя, апробации результатов диссертации, наличие публикаций, а также, структура и объем диссертации.

В **первой главе** рассматривается состояние развития компьютерных игр на данный момент, определены перспективы на ближайшее будущее. Раскрыто понятие лабиринта, рассмотрены его основные классификации с множеством примеров.

Сформулирована постановка задачи, представлена диаграмма объектов предметной области.

Приведены основные алгоритмы создания лабиринтов: поиска в глубину, Эллера, Крускала, Прима, «охотья и убивай», рекурсивного деления. Изучены преимущества и недостатки рассмотренных алгоритмов. Предоставлен код реализации каждого алгоритма на языке программирования с комментариями относительно каждого действия.

Изучены волновой алгоритм и алгоритм одной руки, находящие путь между двумя любыми клетками лабиринта. Приведены их схемы реализации.

Произведён обзор существующих систем-аналогов.

Во второй главе рассматриваются вопросы проектирования системы. Представлена структурная схема системы, в состав которой входит подсистема управления, отвечающая за взаимодействие подсистем между собой и представлена в виде иерархического меню, подсистема составления лабиринта, подсистема нахождения пути, которая отвечает за поиск пути в лабиринте в соответствии с выбранным алгоритмом, файловая подсистема, которая отвечает за хранение лабиринта в файле заранее определенного типа, подсистема визуализации, которая отвечает за корректное отображение лабиринта у пользователя и дает возможность выбора оформления и справочная подсистема, которая содержит сведения о системе (руководство пользователю) и ее об ее разработчиках.

Рассмотрены основные принципы по созданию интерфейса пользователя.

Составлен информационно-логический проект системы, включающий в себя диаграмму вариантов использования, диаграмму состояний, диаграмму деятельности и диаграмму последовательности.

Для реализации проекта по созданию автоматизированной системы построения лабиринта и поиска выхода из него, языком программирования был выбран Java, средой разработки была выбрана новая интегрированная среда разработки приложений для платформы *Android* – *Android Studio*. В качестве фреймворка был выбран *LibGDX*.

Среда построена на базе исходных текстов продукта *IntelliJ IDEA Community Edition*, развиваемого отечественной компанией *JetBrains*. *Android Studio* развивается в рамках открытой модели разработки и распространяется под лицензией *Apache 2.0*.

Новая среда разработки адаптирована для выполнения типовых задач, решаемых в процессе разработки приложений для платформы *Android*. В том числе в среду включены средства для упрощения тестирования программ на совместимость с разными версиями платформы и инструменты для проектирования приложений, работающих на устройствах с различными разрешениями экрана (планшеты, смартфоны, ноутбуки, часы, очки и т.п.). Помимо возможностей, присутствующих в *IntelliJ IDEA*, в *Android Studio* реализовано несколько дополнительных функций, таких как новая унифицированная подсистема сборки, тестирования и развёртывания приложений, основанная на сборочном инструментарии *Gradle* и поддерживающая использование средств непрерывной интеграции.

Для ускорения разработки приложений представлена коллекция типовых элементов интерфейса и визуальный редактор для их компоновки, предоставляющий удобный предпросмотр различных состояний интерфейса приложения (например, можно посмотреть как интерфейс будет выглядеть для различных версий *Android* и для различных размеров экрана). Для создания нестандартных интерфейсов присутствует мастер создания собственных элементов оформления, поддерживающий использование шаблонов. В среду встроены функции загрузки типовых примеров кода из *GitHub*.

В состав также включены учитывающие особенности платформы *Android* расширенные инструменты рефакторинга, проверки совместимости с прошлыми выпусками, выявления проблем с производительностью, мониторинга потребления памяти и оценки удобства использования. В редактор добавлен режим быстрого внесения правок. Система подсветки, статического анализа и выявления ошибок расширена поддержкой *Android API*. Интегрирована поддержка оптимизатора кода *ProGuard*. Встроены средства генерации цифровых подписей. Предоставлен интерфейс для управления переводами на другие языки.

Gradle был разработан для расширяемых много-проектных сборок, и поддерживает инкрементальные сборки, определяя, какие компоненты дерева сборки не изменили и какие задачи, зависящие от этих частей, не требуют перезапуска. Кроме того *Gradle* существенно упрощает установку дополнительных библиотек в используемых приложениях, за счёт того, что добавление их осуществляется одной строкой кода, которая в дальнейшем при компиляции автоматически загружает из интернета необходимую библиотеку, устанавливает её и связывает её с текущими классами.

LibGDX – это *Java* фреймворк, предоставляющий кросс-платформенный *API* для разработки игр и приложений реального времени. Основные его возможности:

- 1) Кроссплатформенная разработка для *Android*, *iOS*, *Windows*, *Linux* и *HTML5* (32 бита и где возможно 64 бита).
- 2) Множество *back-end* основанные на *Jogl*, *LWJGL*, *Angle (NVIDIA 3D Vision)*, *Android APIs*, *iOS APIs* и *WebGL/HTML5 API*.
- 3) Написание кода и тестирование на компьютере и плавное разворачивание на *Android*, *iOS*, чистом *JavaScript*, *Applets*, *Webstart* или как обычное приложение для персонального компьютера.

За счёт использования фреймворка *LibGDX*, который позволяет разрабатывать кросс-платформенные приложения, разрабатываемая программная система будет поддерживаться на операционных системах *Android*, *iOS*, *Windows*.

В третьей главе рассматриваются особенности реализации системы. Отображены основные позиции по формированию интерфейса пользователя.

В целях предоставления пользователю удобного пользования системой был разработан графический интерфейс, максимально простой и интуитивно понятный.

При запуске приложения, в то время как в память загружаются текстуры, необходимые для отображения игры, пользователю показывается приветственный экран. Этот экран необходим, потому что процесс загрузки может быть достаточно долгим, особенно на малопроизводительных устройствах, и необходимо продемонстрировать пользователю, что приложение загружается. Без этого экрана пользователь будет видеть лишь чёрный экран, что обычно означает, что приложение зависло. По статистике 95% пользователей не дожидаются загрузки приложения, а выходят из него, нажимая клавишу *Back* или *Home*. Как только текстуры загрузятся, пользователь попадает в главное меню приложения.

В меню выбора настроек пользователю отображаются три группы настроек, область экрана, отвечающая за предпросмотр лабиринта и кнопка запуска *PLAY*. Первая группа настроек отвечает за выбор алгоритма создания лабиринта. На выбор предоставляются алгоритмы «рекурсивного спуска», Прима и «охотья и убивай». Вторая группа настроек отвечает за выбор размера создаваемого лабиринта. На выбор предоставляются маленький, средний и большой. Третья группа настроек отвечает за тему, в которой будет выполнен лабиринт. На выбор предоставляется классическая в чёрно-белом варианте, снежная и inferнальная.

Все группы настроек выполнены так, что возможно нажать только одну кнопку, по принципу *checkbox*. При нажатии каждой кнопки настроек, на области предпросмотра вызывается метод создания лабиринта, согласно текущим настройкам, что позволяет заранее знать как будет примерно выглядеть лабиринт, который пользователю предстоит пройти. При окончании формирования настроек создаваемого лабиринта, пользователю необходимо нажать кнопку *PLAY*, чтобы перейти к прохождению (рисунок 1).



Рисунок 1 – Экран логического игрового приложения во время прохождения лабиринта

При попадании на экран прохождения лабиринта, пользователю отображаются:

- лабиринт, созданный согласно выбранным на предыдущем экране настройкам;
- персонаж, находящийся в верхнем левом углу;
- место финиша в виде флага, находящегося в правом нижнем углу;
- стрелки управления игроком.

Целью прохождения является достижение персонажем места финиша. Управление осуществляется путём нажатия на стрелки управления, которые отображаются только в том случае, если персонаж стоит на месте и только в тех направлениях, в сторону которых игрок может передвигаться. Это сделано для того, чтобы не перегружать игровой экран и не ограничивать обзор пользователя ненужными элементами.

Для удобства прохождения лабиринта, было реализовано автоматическое перемещение игрока до следующего перекрёстка в направлении нажатой стрелки. Это существенно упрощает управление, ускоряет темп игры и избавляет пользователя от ненужных действий по управлению игроком на закрытых коридорах, оставляя лишь суть лабиринта – выбор направления движения на каждом перекрёстке.

По достижению персонажем места финиша, пользователь возвращается в меню настроек создания лабиринта и может заново его пройти, но уже с другими настройками.

Навигация в приложении реализована путём нажатия на телефоне кнопки *Back*, что перемещает пользователя на предыдущий экран, а если он находится в главном меню, то приложение закрывается и освобождается память устройства от занятых этим приложением ресурсов.

В приложении реализуются три алгоритма генерации лабиринтов:

- алгоритм «рекурсивного спуска»;
- алгоритм Прима;
- алгоритм «охотья и убивай».

Каждый из них обладает своими преимуществами и недостатками. Для наглядности будем использовать средний размер лабиринта.

Алгоритм «рекурсивного спуска» является быстрым, лёгким в понимании и простой в реализации. Именно по-этому многие разработчики предпочитают именно его, разрабатывая различные игры. Однако необходимость хранить весь лабиринт в памяти компьютера, делает его неэффективным при создании очень больших лабиринтов. Кроме того, за счёт малого количества тупиков, которые не обладают большой продолжительностью, достаточно легко без ошибок пройти лабиринт с первого раза.

Алгоритм Прима – алгоритм работающий по принципу разрастающегося из одной точки дерева. Особенностью данного алгоритма является создание большого количества тупиков. Не всем пользователям нравится данное обстоятельство, однако однозначно можно сказать, что при создании больших лабиринтов, большое количество тупиков сильно осложняет прохождение.

Алгоритм «охотиться и убивай» так же как и алгоритм «рекурсивного спуска» создаёт длинные коридоры с наименьшим количеством тупиков по сравнению с другими алгоритмами, однако обладает более нетривиальным решением за счёт использования другого критерия поиска непосещённых клеток, что заставляет иной раз задуматься, в какую из сторон направиться на данном перекрёстке.

Предоставлено описание контрольного примера, демонстрирующего работоспособность приложения.

Составлены диаграмма реализации системы, включающая в себя диаграмму развёртывания и диаграмму компонентов.

Составлен комплекс технических средств, необходимых для запуска приложения на различных платформах.

ЗАКЛЮЧЕНИЕ

Выполнен анализ теоретических основ создания лабиринтов мобильного приложения на базе *Android* с использованием фреймворка *LibGDX*.

В соответствии с техническим заданием был произведен анализ предметной области, реализовано программное и информационное обеспечение, подготовлены контрольные примеры и произведены тестирование и отладка автоматизированной системы генерации лабиринта и поиска выхода из него.

Рассмотрены основные алгоритмы создания и прохождения лабиринтов. Предоставлена реализация данных алгоритмов в программировании. Изучены их преимущества и недостатки. Сделаны выводы, относительно применимости данных алгоритмов.

Анализируя результаты работы по разработке программного продукта, можно сделать вывод, что весь запланированный функционал, определенный на этапе спецификации системы, был успешно реализован. В результате разработки проекта была создана система, работающая на многих платформах (*Windows, Android, IOS*) которая дает возможность создавать лабиринты разных размеров и различными алгоритмами. Созданный лабиринт в дальнейшем возможно пройти, управляя персонажем на экране приложения. Спроектированная система имеет интуитивно понятный интерфейс для пользователя, проста в использовании, не требует серьезных аппаратных затрат.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

1. Пятосин, А.В. Поиск в глубину в контексте мобильных систем / А.В. Пятосин // 51-я научная конференция аспирантов, магистрантов и студентов БГУИР, 2015 г., Минск, Беларусь, 22 – 26 апр. 2015 г., Беларусь : БГУИР, 2015. – 460 с. – С. 71.

2. Пятосин, А.В. Алгоритм Крускала в контексте мобильных систем / А.В. Пятосин // 51-я научная конференция аспирантов, магистрантов и студентов БГУИР, 2015 г., Минск, Беларусь, 22 – 26 апр. 2015 г., Беларусь : БГУИР, 2015. – 460 с. – С. 72.

3. Пятосин, А.В. Алгоритм Прима в контексте мобильных систем / А.В. Пятосин // 51-я научная конференция аспирантов, магистрантов и студентов БГУИР, 2015 г., Минск, Беларусь, 22 – 26 апр. 2015 г., Беларусь : БГУИР, 2015. – 460 с. – С. 73.

4. Пятосин, А.В. Поиск кратчайшего пути в контексте мобильных систем / А.В. Пятосин // 51-я научная конференция аспирантов, магистрантов и студентов БГУИР, 2015 г., Минск, Беларусь, 22 – 26 апр. 2015 г., Беларусь : БГУИР, 2015. – 460 с. – С. 74.