

Разработка мобильных игр в жанре «Поиск предметов» с помощью MIT App Inventor

А.Н. Вильданов

*Нефтекамский филиал ФГБОУ ВО «Башкирский государственный университет»,
г. Нефтекамск*

Аннотация: В данной работе рассматривается процесс разработки игры, наподобие поиска предметов, для мобильных устройств, работающих на базе ОС Android, с помощью MIT App Inventor. Разработан алгоритм игры, приведены блоки кода. Статья предназначена для школьников и студентов, увлекающихся программированием, и будет полезна для преподавателей информатики, при использовании на дополнительных занятиях или организации проектной и исследовательской деятельности школьников и студентов по информатике.

Ключевые слова: MIT App Inventor, мобильные приложения, обучение созданию мобильных приложений, дополнительное образование детей, поиск предметов.

1. Постановка задачи

Роль мобильных приложений в сегодняшней повседневной жизни трудно переоценить. Мобильные технологии используются для решения самых разнообразных задач: навигация внутри зданий [1], поиск нужных рейсов [2], строительство [3], вызов экстренных служб [4], и т.д. И, конечно же, немалую долю приложений составляют игры.

Возможно, многие из вас играли в компьютерные игры на поиск предметов. Это игра в первую очередь на внимательность, с элементами небольших головоломок. Перед вами появляется список предметов, которые нужно собрать в комнате. Они могут находиться в любом углу помещения, быть малозаметными или сливаться с фоном. Их нахождение дает игроку определенное количество игровых очков.

В последние годы главным направлением развития IT отрасли являются облачные вычисления, в том числе и для мобильных ПО [5]. В данной работе рассматривается алгоритм разработки подобной игры с помощью среды MIT App Inventor. MIT App Inventor

(<http://ai2.appinventor.mit.edu/>) – это облачная среда визуальной разработки приложений для платформы OS Android.

Работа в ней не требует знания языка программирования Java и Android SDK, достаточно знания элементарных основ алгоритмизации [6]. Для работы в MIT App Inventor необходимо наличие Google или Google Apps аккаунта, а построение программ осуществляется в визуальном режиме с использованием блоков программного кода [6].

MIT App Inventor имеет несложный, интуитивно понятный интерфейс, который позволяет разрабатывать простейшие приложения даже школьникам, и не требует специальных знаний и опыта работы в данной области (достаточно лишь знаний основ алгоритмизации и программирования) [7], что поможет избежать или упростить проблемы, с которыми сталкиваются учителя и ученики при изучении темы «Алгоритмизация и программирование» [8].

Язык MIT App Inventor очень похож на Scratch, в его основе лежит тот же принцип перетаскивания визуальных кирпичиков и сборки программы из блоков. Отличие App Inventor от Scratch состоит в том, что App Inventor ориентирован не на десктопное использование, а предназначен для создания приложений под мобильное устройство – смартфон или планшет с ОС Android. Приложения на App Inventor умеют «понимать» данные акселерометра мобильного гаджета, управлять встроенной камерой, видят, как ориентирован телефон в пространстве и многое другое [9]. Например, в работе [10] на App Inventor разработан проект удаленной автоматизированной системы голосового управления контроллером arduino с помощью телефона или планшета.

Итак, создадим новый проект. Интерфейс языка программирования MIT App Inventor состоит из двух основных частей – дизайнера и редактора блоков [9]. Сначала мы находимся в режиме «Дизайнер». В этом режиме

создается интерфейс («внешний вид») приложения [6]. Данный режим используют для выбора и размещения различных компонент приложения: кнопок, текстовых полей, изображений и др., которые отображаются на экране вашего устройства, при запуске приложения [6].

Подготовим картинки и загрузим их в игровую среду проекта. Одна картинка будет «пустой». На нее мы будем заменять изображение уже найденного предмета. Предметы на экране будем создавать в виде обычных кнопок с иконками. Игровое поле сделаем прямоугольным, поэтому выберем табличное расположение, к примеру, из четырех строк и трех столбцов (рис. 1).

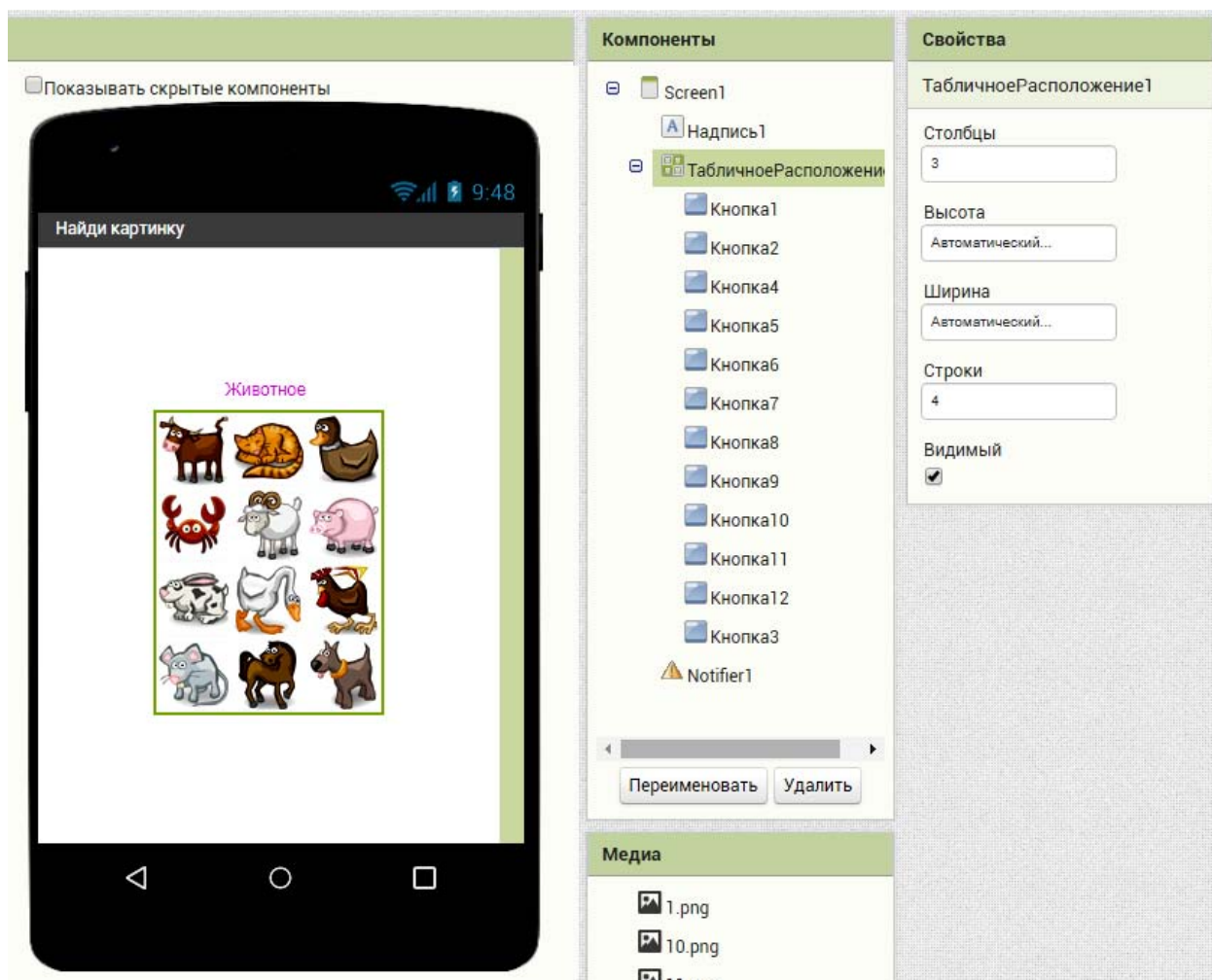


Рис. 1. – Режим «Дизайнер» для игры

Элемент «надпись» будет содержать название животного, изображение которое игрок должен найти и кликнуть мышкой (рис. 1). Уведомитель Notifier будет нас уведомлять об окончании игры (когда найдены все предметы).

Теперь можно приступить к разработке программного кода игры. Для этого переходим в режим «Блоки». Режим «Блоки» используется для программирования поведения мобильного приложения и его компонент [6].

Сначала создаем список животных (рис. 2):

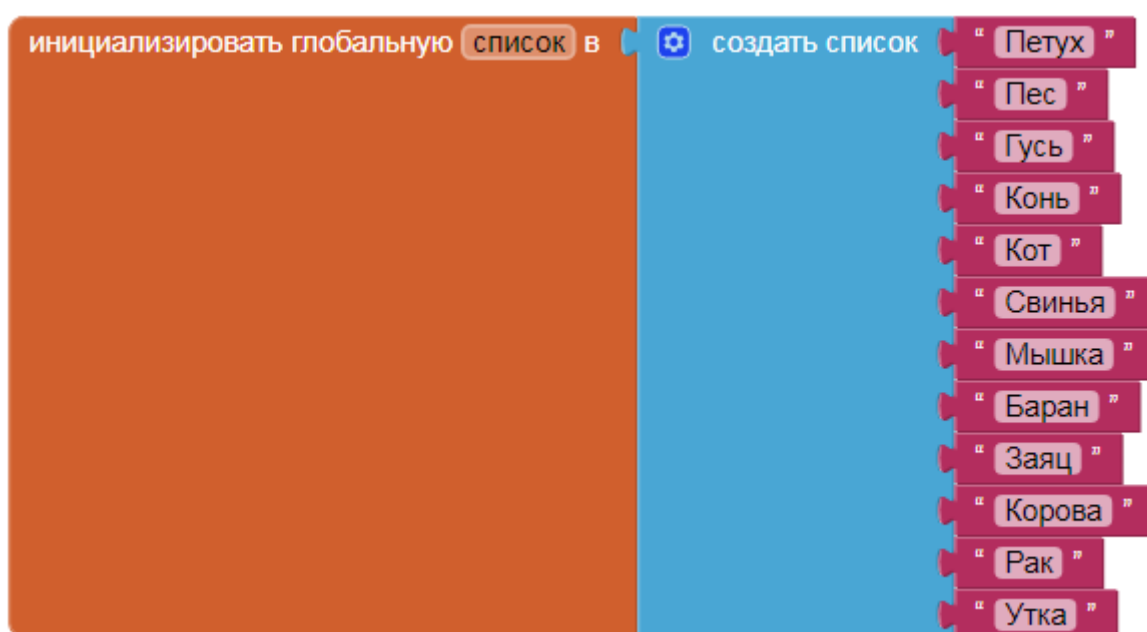


Рис. 2. – Создание списка

За номер картинки, которую в данный момент игрок должен найти в игре, будет отвечать переменная «индекс». В начале игры на экране, соответственно, нужно отобразить первый элемент списка (рис. 3):

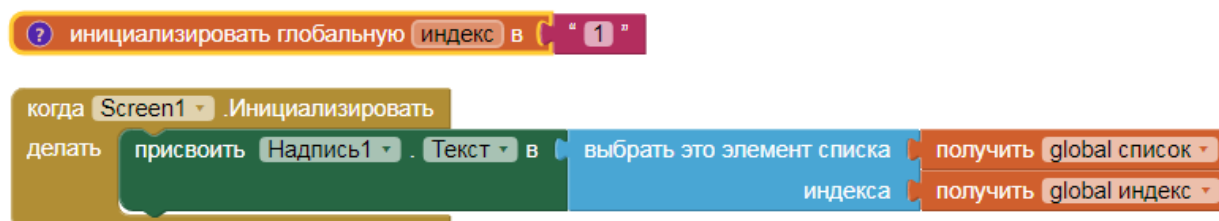


Рис. 3. – Создание индекса текущей картинки

Поскольку действия кнопок будут однообразными, удобно создать процедуру. Назовем ее, соответственно, «действие_кнопки» (рис. 4). Алгоритм примерно следующий:

- 1) Сравниваем загаданную надпись и надпись на нажатой кнопке;
- 2) Если они совпадают, то предмет найден, заменяем изображение картинки на пустое изображение (предмет как бы исчезает).
- 3) Переменную индекс увеличиваем на единицу и загадываем название следующего животного из списка;
- 4) Если вдруг значение индекса достигло значения 13, то это означает, что все картинки найдены, и игру нужно заканчивать.

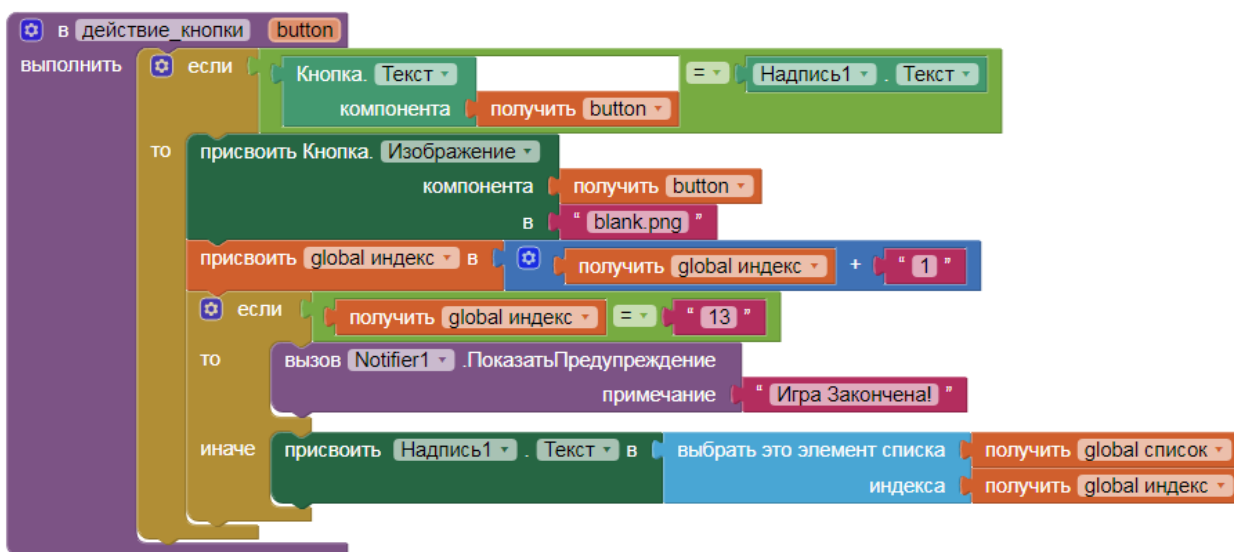


Рис. 4. – Действие кнопки

Теперь наше действие «навешиваем» на все кнопки. В цикле, увы, это сделать не получится, поэтому делаем это вручную для всех кнопок (рис. 5):

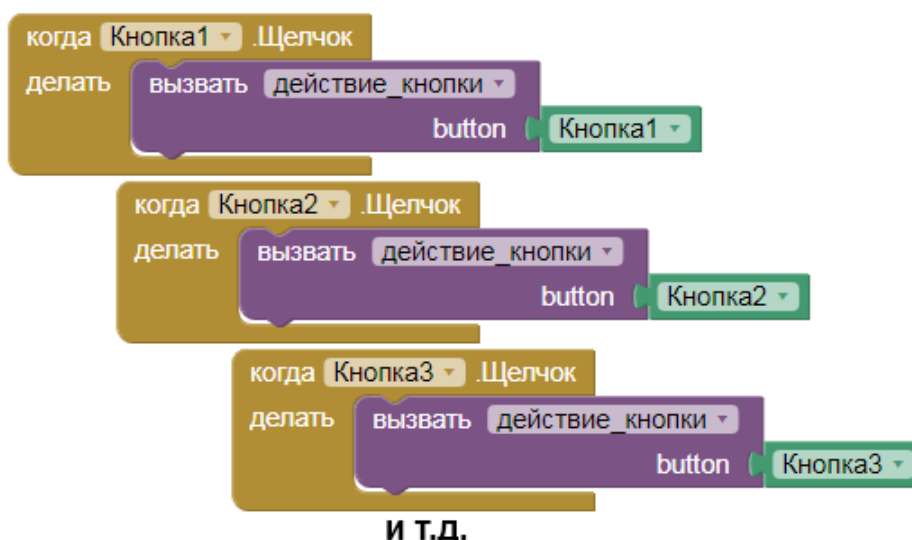


Рис. 5.

Приложение готово, осталось скомпилировать программу в исполняемое приложение арк. Компиляция происходит в облачной среде MIT App Inventor. Скачать и установить приложение можно с помощью другого приложения – MIT App Inventor Companion, которое позволяет считывать QR код созданного вами мобильного приложения для установки его на ваше устройство [6]. MIT App Inventor Companion, в свою очередь, можно загрузить и установить из магазина Google Play.

В меню проекта выбираем «Построить» – «Приложение (создать QR-код для скачивания. арк)» (рис. 6):

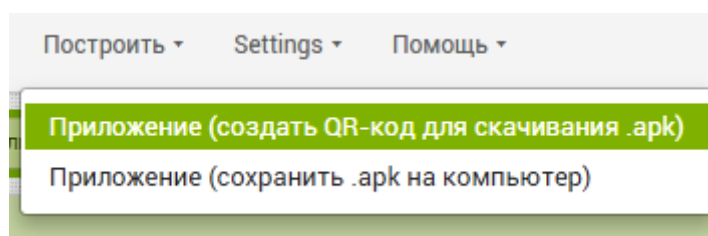


Рис. 6.

Компиляция приложения достаточно быстрая и составляет порядка нескольких секунд. В случае успешной компиляции появляется QR код мобильного приложения (рис. 7).

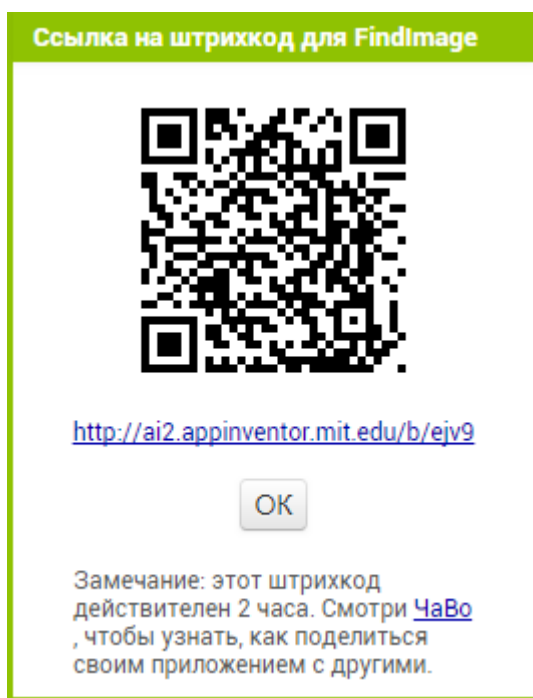


Рис. 7.



Рис. 8.

Теперь фотографируем QR код с помощью MIT App Inventor Companion, и происходит скачивание приложения на телефон. Устанавливаем приложение (при этом необходимо разрешить в мобильном устройстве установку приложений из неизвестных источников).

Приложение успешно установлено и работает (рис. 8).

Заключение

Возможные пути совершенствования игры – добавление игрового фона, таймера, звуков. Например, при правильном нажатии проигрывается один мелодичный звук, при неправильном – другой. Можно также заменить названия животных на аналогичные названия в другом языке, и тогда это уже будет приложение для проверки знания иностранного языка.

Литература

1. Сафронов Р.Ю. Разработка мобильного приложения для iOS с использованием технологии «iBeacon» для решения задач навигации внутри



зданий на основе беспроводных сетей // Инженерный вестник Дона, 2014, №1. URL: ivdon.ru/ru/magazine/archive/N4y2014/2726/.

2. Вильданов А.Н. Проектирование мобильного приложения для поиска междугородних рейсов автобусного транспорта с помощью API Яндекс.расписания // Инженерный вестник Дона, 2018, №4. URL: ivdon.ru/magazine/archive/n4y2018/5416/.

3. Александровский В.Г. Мобильные технологии в строительстве. Программное обеспечение на платформе Android. Часть 1. Инженерный вестник Дона, 2019, №4 URL: ivdon.ru/ru/magazine/archive/n4y2019/5874.

4. Марченко И. О., Слугина А. Е. Система оповещения в случае дорожно-транспортного происшествия // Инженерный вестник Дона, 2016, №2. URL: ivdon.ru/ru/magazine/archive/n2y2016/3601/.

5. Старицкий И.Н. Разработка сервиса виртуализации мобильных приложений в браузере. Инженерный вестник Дона, 2018, №1. URL: ivdon.ru/ru/magazine/archive/n1y2018/4780.

6. Ливенец М.А., Ярмахов Б.Б. Программирование мобильных приложений в MIT App Inventor: Практикум. – 2016. – 100 с.

7. Шалина О.Н. Некоторые аспекты обучения школьников созданию Android-приложений // Достижения и приложения современной информатики, математики и физики. Материалы VI Всерос. науч.-практ. заоч. конф. - Уфа: Издательство «Башкирский государственный университет», 2017 – С. 134-139.

8. Пьянзина И.Н. Программируем с Mit App Inventor 2. Информатика в школе. 2018. №3. – С. 19-22.

9. Ярмахов Б.Б. MIT App Inventor – каждый может создать мобильное приложение. URL: newtonew.com/app/mit-app-iventor-intro.



10. Арменков А.Г. Обучение программированию школьников. Создание приложения в среде Mit App Inventor // Вестник науки и образования, 18 (72). – 2019, С. 73-76.

References

1. Safronov R.YU. Inzenernyj vestnik Dona, 2014. №1. URL: ivdon.ru/ru/magazine/archive/N4y2014/2726.

2. Vildanov A.N. Inzenernyj vestnik Dona, 2018. №4. URL: ivdon.ru/magazine/archive/n4y2018/5416.

3. Aleksandrovsky V.G. Inzenernyj vestnik Dona, 2019. №4. ivdon.ru/ru/magazine/archive/n4y2019/5874.

4. Marchenko I. O., Slugina A. E. Inzenernyj vestnik Dona, 2016. № 2. URL: ivdon.ru/ru/magazine/archive/n2y2016/3601.

5. Staritsky I.N. Inzenernyj vestnik Dona, 2018. №1. URL: ivdon.ru/ru/magazine/archive/n1y2018/4780.

6. Livenec M.A., Yarmahov B.B. Programmirovanie mobil'nyh prilozhenij v MIT App Inventor: Praktikum. [Programming Mobile Applications in the MIT App Inventor: Workshop]. 2016. 100 p.

7. Shalina O.N. Dostizheniya i prilozheniya sovremennoj informatiki, matematiki i fiziki: sb. st. Ufa: Izdatel'stvo «Bashkirskij gosudarstvennyj universitet», 2017. pp. 134-139.

8. Pyanzina I.N. Informatika v shkole, 2018, №3, pp.19-22.

9. Yarmahov B.B. MIT App Inventor – kazhdyj mozhet sozdat' mobil'noe prilozhenie. [MIT App Inventor - anyone can create a mobile application]. URL:newtonew.com/app/mit-app-iventor-intro, svobodnyj.

10. Armenkov A.G. Vestnik nauki i obrazovaniya. 2019. №18. pp. 73-76.