

СОДЕРЖАНИЕ

	Стр.
1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	2
1.1. Нормативно-правовая основа разработки дополнительной профессиональной программы	2
1.2. Цель реализации программы. Планируемые результаты обучения	4
1.3. Организационно-педагогические условия, формы аттестации	5
1.4. Учебный план	6
1.5. Календарный учебный график	7
2. РАБОЧИЕ ПРОГРАММЫ	9
3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ	24
4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ	33

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дополнительной профессиональной программе
по программе повышения квалификации
«Программирование мобильных приложений»

Программа повышения квалификации **«Программирование мобильных приложений»** представляет собой комплект документов, разработанных и утвержденных Государственным бюджетным профессиональным образовательным учреждением Иркутской области «Братский политехнический колледж» с учетом потребностей регионального рынка труда, требований федеральных органов исполнительной власти и соответствующих отраслевых требований, на основе профессионального стандарта «Разработчик Web и мультимедийных приложений» утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 18.01.2017 г., № 44н, зарегистрировано Министерством юстиции 31.01. 2017 г., № 45481. На основании Распоряжения Правительства Иркутской области № 595-рп от 08.02.2019г., определена компетенция опережающей профессиональной подготовки «Разработка мобильных приложений». В результате анализа функциональной карты вида профессиональной деятельности была определена трудовая функция, результат работы представлен в таблице 1.

Таблица 1.- Связь дополнительной профессиональной программы с профессиональным стандартам

Наименование программы	Наименование выбранного профессионального стандарта, трудовая функция.	Уровень квалификации трудовой функции
«Программирование мобильных приложений»	«Разработчик Web и мультимедийных приложений» Кодирование на языках web-программирования.	4

Программа регламентирует цели, ожидаемые результаты, содержание, условия и технологии реализации образовательного процесса, оценку качества подготовки слушателя по данной программе и включает в себя: учебный план, программу курса, календарный учебный график, а также оценочные и методические материалы, обеспечивающие качество подготовки обучающихся.

1.1. Нормативно-правовая основа разработки дополнительной профессиональной программы.

Нормативно-правовую базу программы составляют:

- Закон об образовании в Российской Федерации от 29.12.2012 г. № 273-ФЗ;

- Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденный Приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499, зарегистрированного Министерством юстиции Российской Федерации 20 августа 2013 г. Регистрационный № 29444 (В ред. Приказа Минобрнауки России от 15.11.2013 № 1244);

- Квалификационный справочник должностей руководителей, специалистов и других служащих, утвержденный Постановлением Минтруда РФ от 21 августа 1998 г. N 37 (в ред. Постановлений Минтруда России от 21.01.2000 N 7, от 04.08.2000 N 57, от 20.04.2001 N 35, от 31.05.2002 N 38, от 20.06.2002 N 44, от 28.07.2003 N 59, от 12.11.2003 N 75, Приказов Минздравсоцразвития России от 25.07.2005 N 461, от 07.11.2006 N 749, от 17.09.2007 N 605, от 29.04.2008 N 200, от 14.03.2011 N 194, Приказов Минтруда России от 15.05.2013 N 205, от 12.02.2014 N 96)

- Постановление Правительства Российской Федерации от 15 августа 2013 г. № 706 «Об утверждении правил оказания платных образовательных услуг

- Приказ Минтруда России от 12 апреля 2013 г. № 148н «Об утверждении уровней квалификаций в целях разработки проектов профессиональных стандартов»

- Приказ Минтруда России от 29 апреля 2013 г. № 170-н «Об утверждении методических рекомендаций по разработке профессионального стандарта»

- Методические рекомендации-разъяснения по разработке дополнительных профессиональных программ на основе профессиональных стандартов, письмо Министерства образования и науки Российской Федерации от 22 апреля 2015 г. № ВК-1032/06.

- Устав образовательного учреждения.

Учебный план составлен в соответствии с «Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам», утвержденный Приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499, и современными требованиями, обусловленными необходимостью перехода на качественно новый уровень подготовки специалистов, обеспечивающих их мобильность, социальную защищенность, конкурентоспособность на рынке труда города Братска и региона.

Программа предназначена для категории лиц, достигших возраста 16 лет.

К освоению дополнительных профессиональных программ допускаются:

- 1) лица, имеющие среднее профессиональное и (или) высшее образование;
- 2) лица, получающие среднее профессиональное и (или) высшее образование.

При освоении дополнительной профессиональной программы параллельно с получением среднего профессионального образования и (или) высшего образования удостоверение о повышении квалификации и выдаются одновременно с получением соответствующего документа об образовании и о квалификации.

Продолжительность обучения согласно календарного графика обучения. В ходе освоения программы слушатели приобретают знания и практические навыки создания мобильных приложений в среде визуального программирования MIT App Inventor.

1.2. Цель реализации программы. Планируемые результаты обучения

Цель и планируемые результаты обучения:

Цель программы направлена на освоение следующих профессиональных компетенций:

Профессиональные компетенции	Практический опыт	Умения	Знания
1	2	3	4
Создание программного кода в соответствии с техническим заданием (готовыми спецификациями) Оптимизация программного кода с использованием специализированных программных средств Написание программного кода с использованием языков программирования, определения и манипулирования данными	Создание программного кода в соответствии с техническим заданием (готовыми спецификациями) Оптимизация программного кода с использованием специализированных программных средств Написание программного кода с использованием языков программирования	Применять выбранные языки программирования для написания программного кода Использовать выбранную среду программирования и средства системы управления базами данных Использовать возможности имеющейся программной архитектуры ИР	Синтаксис выбранного языка программирования, особенности программирования на этом языке Особенности выбранной среды программирования и системы управления базами данных Стандартные библиотеки выбранного языка программирования

	рования, определения и манипулирования данными		Компоненты программно-технических архитектур ИР, существующие приложения и интерфейсы взаимодействия с ними
--	--	--	---

1.3. Организационно-педагогические условия, формы аттестации:

Программа обеспечена учебно-методической документацией и материалами по всем учебным курсам. Педагогические сотрудники формируют и хранят в кабинетах и лабораториях учебно-методические комплексы по каждому учебному курсу.

Колледж обеспечивает возможность доступа студентов к новой учебной и методической литературе по информационным курсам в читальном зале библиотеки.

В колледже обеспечена возможность выхода в информационные сети через Интернет. Для реализации программы учебного курса в колледже оборудованы 5 компьютерный класс.

Продолжительность занятий – 45 минут. Занятия проводятся парами. Между уроками пары перерыв 5 минут. Между парами перерыв 10 минут.

Контроль и оценка результатов освоения учебного курса, осуществляется преподавателями в процессе проведения устных опросов, практических занятий, лабораторных занятий. Текущий контроль по учебным курсам проводится в пределах учебного времени, как традиционными, так и инновационными методами, включая информационные технологии. Система оценок – пятибалльная. При освоении учебных курсов предусматривается проведение промежуточной аттестации в форме практической работы.

После завершения освоения программы итоговая аттестация проводится в форме дифференцированного зачета. Дифференцированный зачет проводится в виде выполнения практической работы в среде AppInventor по созданию готового мобильного приложения. В ходе выполнения зачетной работы должен получиться работоспособный программный продукт.

Добавлено примечание ([D1]): Посмотри, нормально так

1.4. Учебный план

по дополнительной профессиональной программе,
(программа повышения квалификации)
«Программирование мобильных приложений»

Индекс	Наименование модуля	Форма промежуточной аттестации ¹	Учебная нагрузка обучающихся (час.)			
			максимальная	самостоятельная учебная работа	Обязательная аудиторная	
					всего занятий	в т. ч. лаб. и практ. занятий
1	2		3	4	5	6
1	Создание приложений в среде MIT App inventor.	Практическая работа	34	18	16	12
2	Итоговая аттестация Дифференцированный зачет	Практическая работа	2		2	
Всего			36	18	18	12

¹ Промежуточная аттестация проводится за счет времени учебного курса

1.5. Календарный учебный график по дополнительной профессиональной программе

«Программирование мобильных приложений»

	Наименование модуля	Виды учебной нагрузки	Недели								Всего часов	
			1	2	3	4	5	6	7	8		
1	Создание приложений в среде MIT App inventor.	обяз.уч	2	2	2	2	2	2	2	2	2	16
		сам.р.с.	3	3	2	2	2	2	2	2	2	18
2	Дифференцированный зачет	Итоговая аттестация									2	2
Всего часов в неделю самостоятельной работы студентов			3	3	2	2	2	2	2	2	2	18
Всего часов в неделю			5	5	4	4	4	4	4	6	36	

РАБОЧАЯ ПРОГРАММА УЧЕБНОГО МОДУЛЯ
«Создание приложений в среде MIT App inventor»

Организация-разработчик: Государственное бюджетное образовательное учреждение среднего профессионального образования Братский политехнический колледж

Разработчики:

Васильева Н.С., преподаватель ГБПОУ ИО «Братский политехнический колледж».

Скоблова Н.И., преподаватель ГБПОУ ИО «Братский политехнический колледж».

Рассмотрена и одобрена на заседании предметно-цикловой комиссии протокол № 10 от 05. 06. 2020 г., председатель ПЦК Котова Е.Н.

СОДЕРЖАНИЕ

	Стр.
1.ПАСПОРТ ПРОГРАММЫ УЧЕБНОГО МОДУЛЯ	12
2.СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОГО МОДУЛЯ	13
3.УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОГО МОДУЛЯ	15
4.КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОГО МОДУЛЯ	16

1. ПАСПОРТ ПРОГРАММЫ УЧЕБНОГО МОДУЛЯ «Создание приложений в среде MIT App inventor»

1.1. Область применения программы

Программа модуля «Создание приложений в среде MIT App inventor» предназначена для лиц, желающих получить систему практических знаний при осуществлении функций разработчика мобильных приложений. Новый опыт получают, и начинающие работники в сфере информационных технологий и те, кто уже ведет работу в области мобильной разработки.

1.2. Цели и задачи модуля – требования к результатам освоения учебного модуля:

Модуль направлен на освоение следующих трудовых функций:

1. Разработка Web и мультимедийных приложений
2. Кодирование на языках web-программирования.

В результате изучения учебного модуля слушатель должен освоить следующие компетенции:

1. Создание программного кода в соответствии с техническим заданием (готовыми спецификациями).
2. Оптимизация программного кода с использованием специализированных программных средств.
3. Написание программного кода с использованием языков программирования, определения и манипулирования данными.

Модуль направлен на приобретение следующего практического опыта:

4. Создание программного кода в соответствии с техническим заданием (готовыми спецификациями).
5. Оптимизация программного кода с использованием специализированных программных средств.
6. Написание программного кода с использованием языков программирования, определения и манипулирования данными.

В результате изучения учебного курса слушатель должен:

Уметь:

1. Применять выбранные языки программирования для написания программного кода.

2. Использовать выбранную среду программирования и средства системы управления базами данных.

3. Использовать возможности имеющейся программной архитектуры ИР.

Знать:

1. Синтаксис выбранного языка программирования, особенности программирования на этом языке.

2. Особенности выбранной среды программирования и системы управления базами данных.

3. Стандартные библиотеки выбранного языка программирования.

4. Компоненты программно-технических архитектур ИР, существующие приложения и интерфейсы взаимодействия с ними.

1.3. количество часов на освоение программы учебного модуля:

максимальной учебной нагрузки обучающегося 36 часов, в том числе: обязательной аудиторной учебной нагрузки обучающегося 16 часов; самостоятельной работы обучающегося 18 часов.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОГО КУРСА

2.1. Объем учебного курса и виды учебной работы

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	36
Обязательная аудиторная учебная нагрузка (всего)	16
в том числе:	
практические занятия	12
Самостоятельная работа обучающегося (всего)	18
Дифференцированный зачет	2

**2.2. Тематический план и содержание учебного модуля
«Создание приложений в среде MIT App inventor»**

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся	Объем часов
1	2	3
Тема 1	Содержание учебного материала	1
Интерфейс пользователя MIT App inventor.	1 Описание интерфейса пользователя Режим «Дизайнер» Экраны приложения Режим «Блоки» Функции режима «Блоки»	
	2 Загрузка и установка приложения на устройство Компоненты приложения	1
	3 Практические работы Создание первого мобильного приложения	2
	Самостоятельная работа Составление конспектов занятий. Чтение учебной и специальной литературы (по вопросам к параграфам, главам учебных пособий, составленным преподавателем).	6
Тема 2	Содержание учебного материала	
Создания приложений в среде MIT App inventor.	1 Практические приемы создания приложений	2
	2 Практические работы Создание приложения с несколькими экранами Обмен данными между экранами Создание приложения с использованием списков Создание приложения с использованием элементов группы «Рисование» Создание приложения с использованием анимации Создание приложения с использованием элементов группы «Медиа»	10
	Самостоятельная работа Составление конспектов занятий. Чтение учебной и специальной технической литературы (по вопросам к параграфам, главам учебных пособий, составленным преподавателем).	12
Всего		36

3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ УЧЕБНОГО МОДУЛЯ

3.1 Реализация учебного модуля требует наличия кабинета 402

Оборудование учебного кабинета:

Рабочие места по количеству обучающихся;

Рабочее место преподавателя;

Наглядные пособия (схемы, плакаты, и т.д.);

Технические средства обучения:

Компьютер с лицензионным программным обеспечением;

Телевизор;

Видеопроектор;

Экран;

Локальная сеть;

Выход в глобальную сеть;

Мультимедиапроектор.

3.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, интернет-ресурсов, дополнительной литературы

Основные источники:

1. Соколова В.В. Разработка мобильных приложений [Электронный 1. ресурс]: учебное пособие/ Соколова В.В.— Электрон. текстовые данные.— Томск: Томский политехнический университет, 2017.— 176 с.— Режим доступа: <http://www.iprbookshop.ru/34706>.— ЭБС «IPRbooks», по паролю Google Android : программирование для мобильных устройств / А. Л.

2. Голощапов. - 2-е изд. - СПб.: БХВ-Петербург, 2018. - 448 с. : ил. ; 23 см. - (Профессиональное программирование). - ISBN 978-5-9775-0729-5 Технологии обработки информации на Java: учеб. пособие для студ.

3. "Информационные системы и технологии", направлений "Информационные системы и технологии", "Информационные технологии", "Информатика и вычислительная техника", "Программная инженерия" / А. В. Ермаков; Саратовский гос. техн. ун-т. - Саратов :СГТУ, 2019. - 48 с. : табл. ; 21 см. - Библиогр.: с. 46 (8 назв.). ISBN 978-5-7433-2841-3

4. Windows 8 : разработка Metro-приложений для мобильных устройств / 4.

В. А. Дронов. - СПб. : БХВ-Петербург, 2017. - 528 с. : ил. ; 23 см. Профессиональное программирование). - ISBN 978-5-9775-0832-2; Дополнительные источники:

1. Разработка и продажа программ для iPhone и iPad // Дмитрий Елисеев // Издательство: БХВ-Петербург, 2018 г. // ISBN 978-5-9775-0687-8

2. Google Android. Системные компоненты и сетевые коммуникации // Алексей Голощапов // Издательство: БХВ-Петербург, 2018 г. // ISBN 978-5-9775-0666-3

Профильные периодические издания:

1. [Webdix.Journal](http://webdix.livejournal.com/) (он-лайн версия) (<http://webdix.livejournal.com/>)

Специализированные порталы:

1. <http://www.intuit.ru>

2. <http://habrahabr.ru/blogs/programming/>

3. <http://phpclub.ru/>

4. <http://www.webscript.ru/>

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОГО МОДУЛЯ

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
1	2
Умения	
<ul style="list-style-type: none"> -Применять выбранные языки программирования для написания программного кода -Использовать выбранную среду программирования и средства системы управления базами данных -Использовать возможности имеющейся программной архитектуры ИР 	Наблюдение за выполнением практических работ
Знания	
<ul style="list-style-type: none"> -Синтаксис выбранного языка программирования, особенности программирования на этом языке -Особенности выбранной среды программирования и системы управления базами данных -Стандартные библиотеки выбранного языка программирования -Компоненты программно-технических архитектур ИР, существующие приложения и интерфейсы взаимодействия с ними 	Текущий контроль в форме: <ul style="list-style-type: none"> - ответы на вопросы; Доклад, самостоятельная работа, практические работы

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ИРКУТСКОЙ ОБЛАСТИ
БРАТСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

Комплект оценочных материалов
для проведения промежуточной аттестации
по учебному модулю
«Создание приложений в среде MIT App inventor»

I. Паспорт комплекта оценочных материалов

1. Область применения комплекта оценочных материалов.

Комплект оценочных материалов предназначен для оценки результатов освоения учебного модуля «Создание приложений в среде MIT App inventor»

Таблица 1

Результаты освоения (объекты оценивания)	Основные показатели оценки результата и их критерии	Тип задания; № задания	Форма аттестации (в соответствии с учебным планом)
<p>ПК 1. Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)</p> <p>Уметь: Применять выбранные языки программирования для написания программного кода -Использовать выбранную среду программирования и средства системы управления базами данных -Использовать возможности имеющейся программной архитектуры ИР.</p> <p>Знать: -Синтаксис выбранного языка программирования, особенности программирования на этом языке -Особенности выбранной среды программирования и системы управления базами данных -Стандартные библиотеки выбранного языка программирования - Компоненты программно-технических архитектур ИР, существующие приложения и интерфейсы взаимодействия с ними</p>	<p>Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)</p> <ul style="list-style-type: none">-Оптимизация программного кода с использованием специализированных программных средств-Написание программного кода с использованием языков программирования, определения и манипулирования данными-Размещение программного кода в страницах, созданных при верстке ИР-Размещение программного кода в клиентской части ИР-Размещение программного кода в серверной части ИР-Оценка и согласование сроков выполнения поставленных задач	Практические задания	Практическая работа

2. Комплект оценочных материалов для промежуточной аттестации

2.1. Задания для проведения промежуточной аттестации

Задание 1

Создайте приложение «Записная книжка»

Описание. Приложение, в котором при вводе имени друга выводится его телефон (или любимый фильм, или день рождения)

Компоненты приложения:

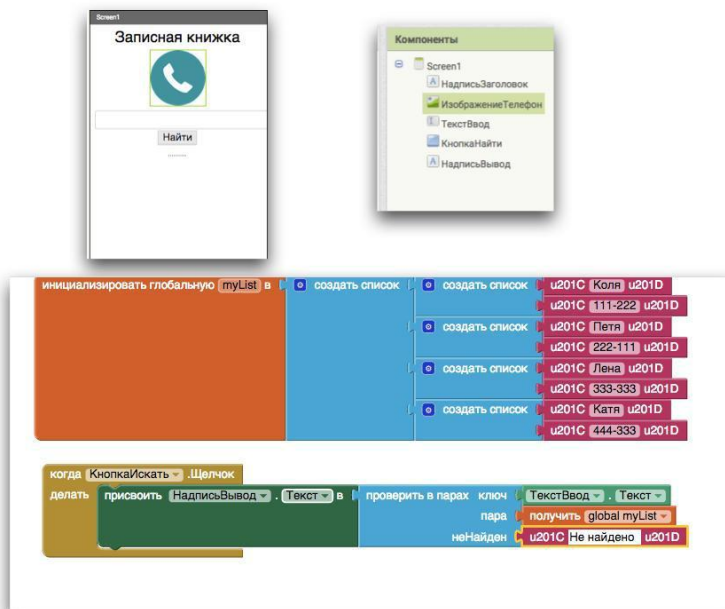
Надпись

Изображение

Текст

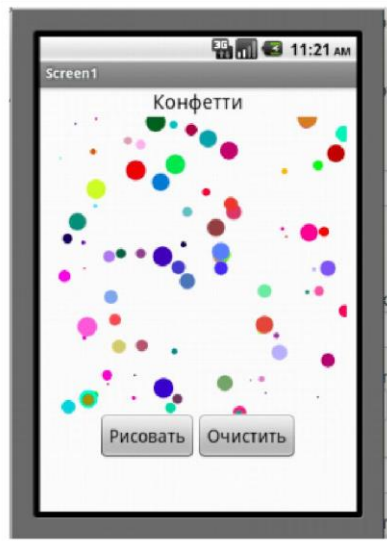
Кнопка

Приложение «Записная книжка»



ЗАДАНИЕ №2

Создайте приложение «Конфетти»



Компоненты

- Screen1
 - НадписьЗаголовок
 - ХолстРисование
 - ГоризонтальноеРасположение
 - КнопкаРисовать
 - КнопкаОчистить
 - Таймер

Logic (Scratch-style blocks):

- Инициализировать глобальную `id` в `истина`
- когда `КнопкаОчистить` нажата: `ХолстРисование` .Очистить, присвоить `global id` в `ложь`
- когда `КнопкаРисовать` нажата: присвоить `global id` в `истина`
- когда `Таймер` истек: `если` `получить global id` `==` `истина`
 - то `присвоить ХолстРисование` .ЦветКраски в `создать цвет`, `создать список`
 - `случайное целое в диапазоне от 1 до 255`
 - `случайное целое в диапазоне от 1 до 255`
 - `случайное целое в диапазоне от 1 до 255`
 - `вызов ХолстРисование` .РисоватьКруг
 - `centerX` `случайное целое в диапазоне от 1 до ХолстРисование` .Ширина
 - `centerY` `случайное целое в диапазоне от 1 до ХолстРисование` .Высота
 - `radius` `случайное целое в диапазоне от 1 до 10`
 - `fill` `истина`

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ИРКУТСКОЙ ОБЛАСТИ
БРАТСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

**Комплект оценочных материалов
для проведения итоговой аттестации
по программе повышения квалификации
«Программирование мобильных приложений»**

I. Паспорт комплекта оценочных материалов

1. Область применения комплекта оценочных материалов.

Комплект оценочных материалов предназначен для оценки результатов освоения дополнительной профессиональной программы по программе повышения квалификации

«Программирование мобильных приложений»

Таблица 1

Результаты освоения (объекты оценивания)	Основные показатели оценки результата и их критерии	Тип задания; № задания	Форма аттестации (в соответствии с учебным планом)
<p>ПК 1. Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)</p> <p>Уметь:</p> <ul style="list-style-type: none">-Применять выбранные языки программирования для написания программного кода-Использовать выбранную среду программирования и средства системы управления базами данных-Использовать возможности имеющейся программной архитектуры ИР. <p>Знать:</p> <ul style="list-style-type: none">-Синтаксис выбранного языка программирования, особенности программирования на этом языке-Особенности выбранной среды программирования и системы управления базами данных-Стандартные библиотеки выбранного языка программирования- Компоненты программно-технических архитектур ИР, существующие приложения и интерфейсы взаимодействия с ними	<p>Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)</p> <ul style="list-style-type: none">-Оптимизация программного кода с использованием специализированных программных средств-Написание программного кода с использованием языков программирования, определения и манипулирования данными-Размещение программного кода в страницах, созданных при верстке ИР-Размещение программного кода в клиентской части ИР-Размещение программного кода в серверной части ИР-Оценка и согласование сроков выполнения поставленных задач	Практические задания	Дифференцированный зачет

2. Комплект оценочных материалов для итоговой аттестации

2.1. Задания для проведения итоговой аттестации

ЗАДАНИЕ №1

1. Создайте приложение, включающее в себя элементы:

- Рисование
- Анимация
- Медиа

Условия выполнения задания

Место (время) выполнения задания (например, на учебной/ производственной практике, в цеху организации (предприятия), мастерской ОУ (ресурсного центра), на полигоне, в учебной фирме и т.п.): 405

Максимальное время выполнения задания: 90 мин.

Вы можете воспользоваться (указать используемое оборудование (инвентарь), расходные материалы, литературу и другие источники, информационно-коммуникационные технологии и проч.).

Указать другие характеристики, отражающие сущность задания: в реальных (модельных) условиях профессиональной деятельности.

Государственное бюджетное профессиональное
образовательное учреждение
Иркутской области
«Братский политехнический колледж»

Методическое пособие для обучающихся

**Основы работы с визуальной средой App Inventor
Практикум применения**

Братск, 2019

Васильева Н.С. , Скоблова Н.С. Основы работы с визуальной средой App Inventor, практикум применения– Братск, 2019. – 30 с.

Методическое пособие составлено для изучения основ мобильного программирования.

Цель данного пособия: оказание помощи обучающимся в изучении среды визуальной разработки AppInventor

Одобрено предметно-цикловой комиссией информатики и вычислительной техники

протокол № 10 от «10» июня 2019г.

Глава 1. Интерфейс пользователя MIT App inventor.

App Inventor (App – сокращение от application, переводится как приложение. Inventor — переводится как изобретатель) — это среда визуальной разработки приложений, не требующая больших знаний в программировании.

App Inventor — это облачная среда визуальной разработки Android-приложений. Построение программ осуществляется в визуальном режиме с использованием блоков программного кода. На компьютер устанавливать ничего не требуется, просто откройте среду разработки ai2.appinventor.mit.edu в браузере Google Chrome и начинайте творить.

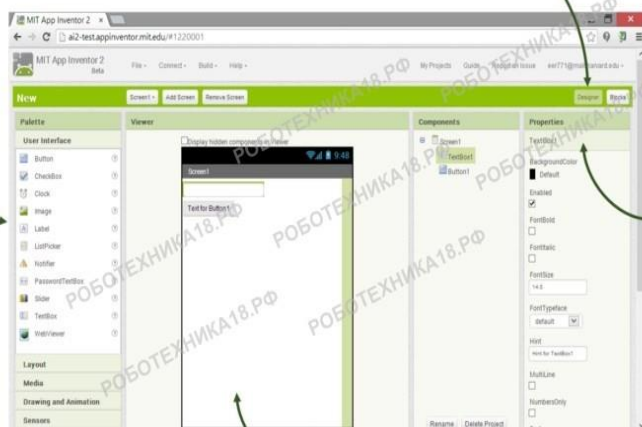
Обратите внимание, что для работы с App Inventor требуется аккаунт Google. Если вы не зарегистрированы в Google, то перед началом работы вам надо будет пройти регистрацию и авторизацию. После этого приложение App Inventor и все ваши сохраненные проекты будут доступны на любом компьютере после авторизации.

App Inventor состоит из конструктора и редактора блоков

Конструктор App Inventor

Palette (Палитра): Выберите нужный компонент и перетаскивайте их мышкой на вкладку Viewer (Просмотр).

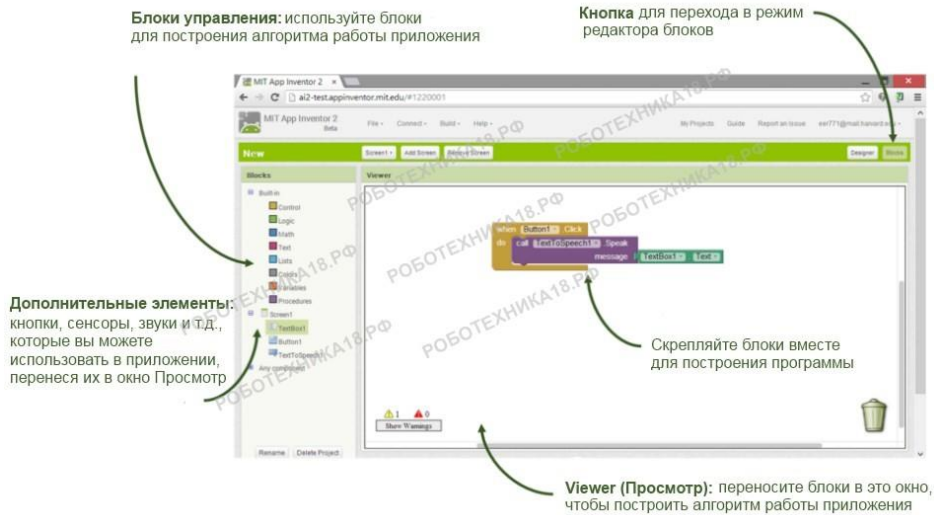
Кнопка для перехода в режим Дизайна (Конструктора)



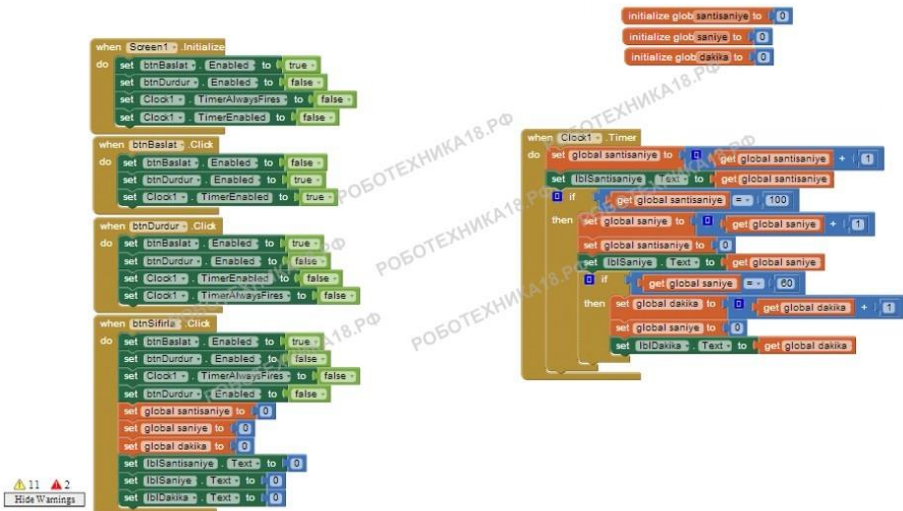
Properties (Свойства): Выберите компоненты и редактируйте их, меняйте цвет, шрифт и другие свойства.

Viewer (Просмотр): Визуальное отображение приложения с выбранными компонентами

Редактор блоков App Inventor



Программирование производится с помощью составления блоков



Блоки программы составляются подобно пазлам или конструктору лего

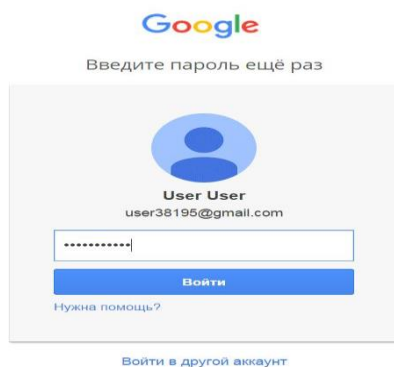
Глава 2. Практические приемы создания приложений

2.1 Основные приемы создания приложений

Практическая работа AppInventor № 1

Алгоритм работы:

1. Создать Google аккаунт, лучший вариант Google Apps!! или использовать созданный ранее.
2. Войти в среду визуального программирования MIT App Inventor по ссылке <http://ai2.appinventor.mit.edu/>



3. Выбрать “Разрешить”



Приложение MIT AppInventor Version 2 запрашивает разрешение на доступ к вашему аккаунту Google.

Выберите аккаунт, который необходимо использовать.

[user38195@gmail.com](#)

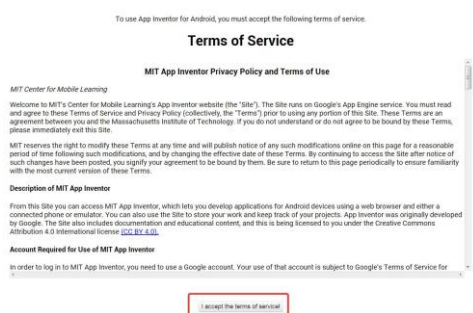
Компания Google не связана с содержанием гостевой книги MIT AppInventor Version 2 или ее владельцами. Если вы выполните вход, компания Google отправит в гостевую книгу MIT AppInventor Version 2 ваш адрес электронной почты, но не пароль и личные данные.

[Войти в другой аккаунт](#)

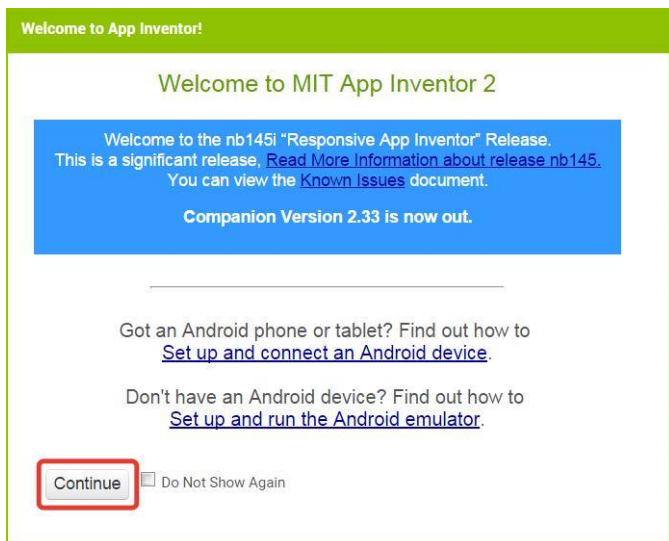
Помнить это подтверждение в течение 30 дней

©2011 Google - [главная страница Google](#) - [Условия использования](#) - [Политика конфиденциальности](#) - [Справка](#)

4.В окне “Terms of Service выбрать” I accept the terms of service”



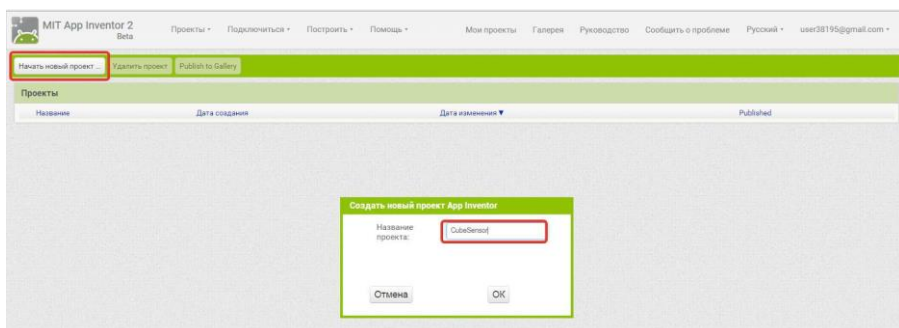
5. В окне “Welcome to App Inventor 2” выбрать “Continue”



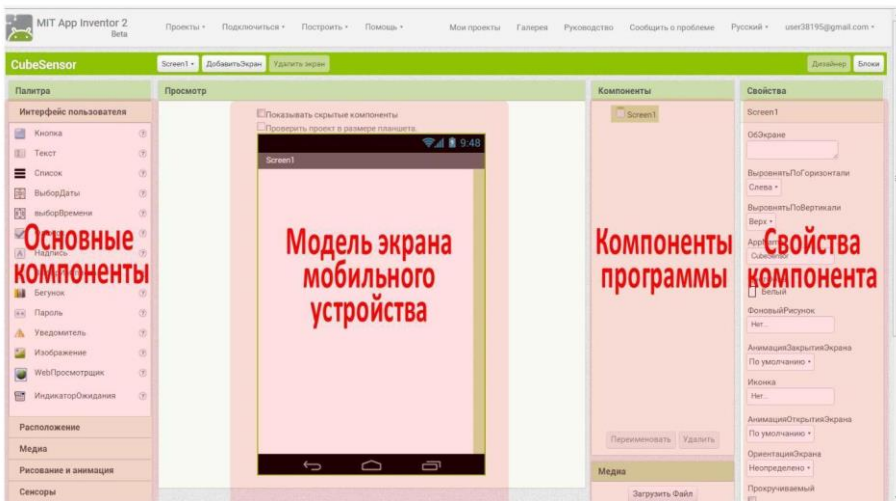
6. Выбрать язык “English-> Русский”



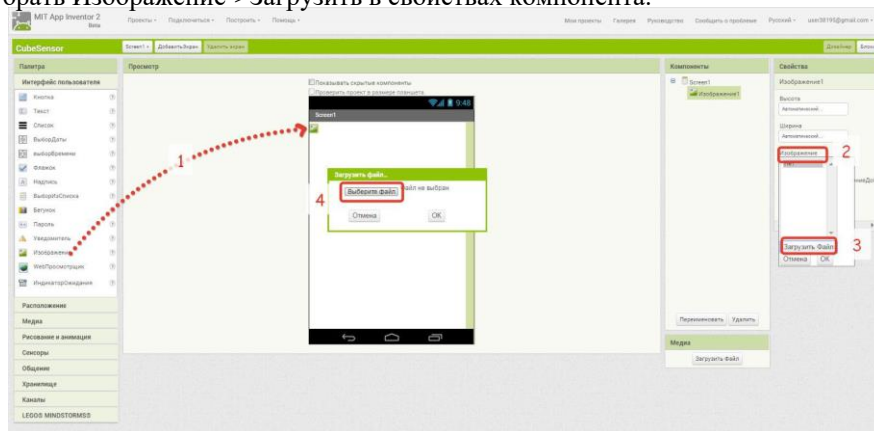
7. Создать новый проект “Начать новый проект->CubeSensor”



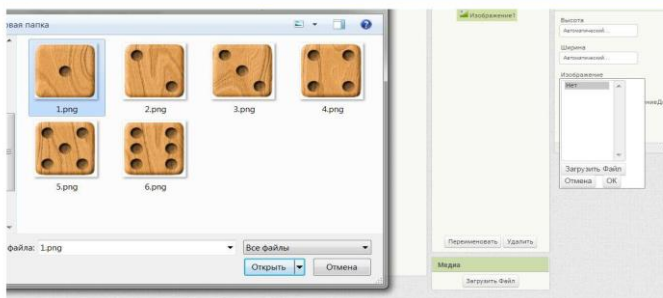
8. Рабочий экран среды визуального программирования будет выглядеть следующим образом:



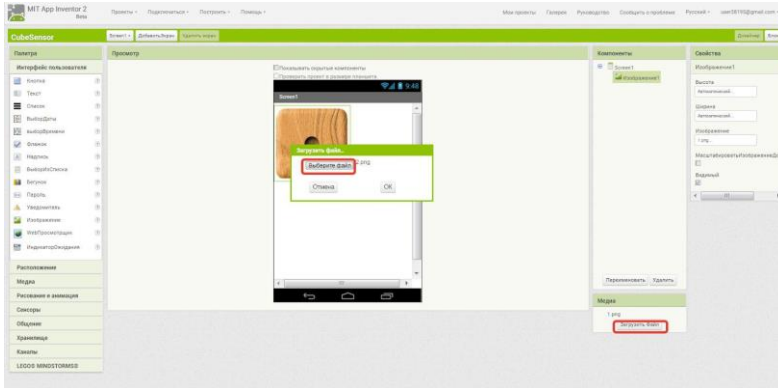
9. Перенести компонент Изображение в окно экрана мобильного устройства, выбрать Изображение->Загрузить в свойствах компонента.



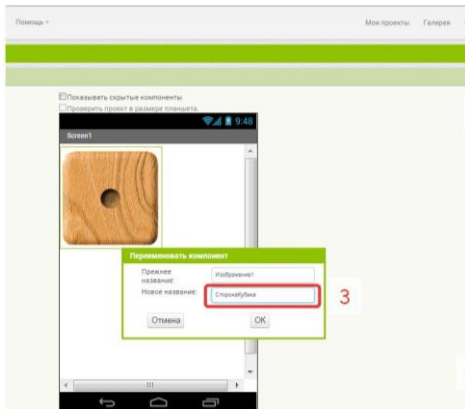
10. Загрузить графический файл для компонента Изображение.



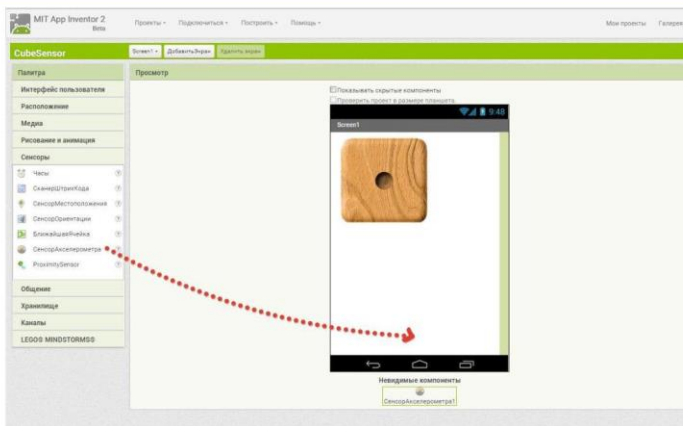
11. Загрузить последовательно 6 графических файлов (сторон кубика) с помощью функции "Загрузить файл".



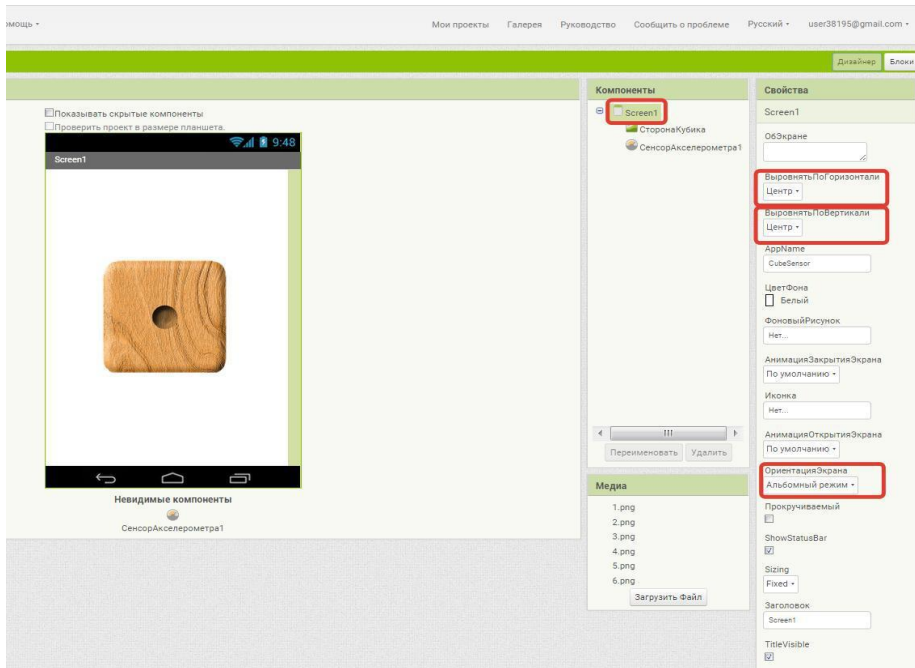
12. Переименовать компонент Изображение1 в СторонаКубика



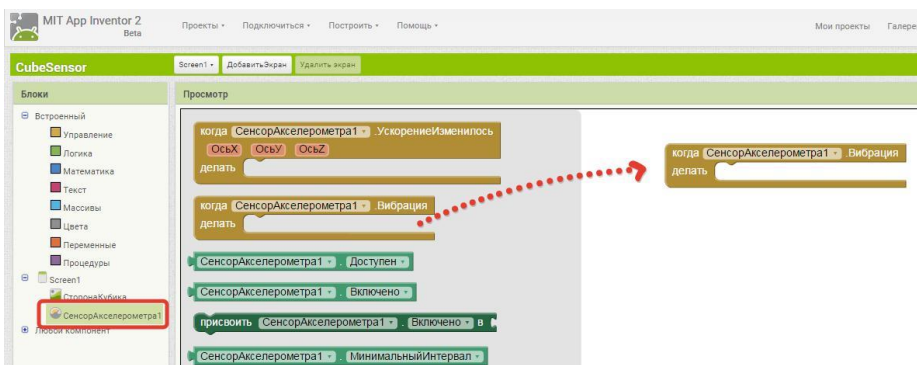
13. Выбрать в группе Сенсоры -> СенсорАкселерометра и перенести его в область экрана мобильного устройства



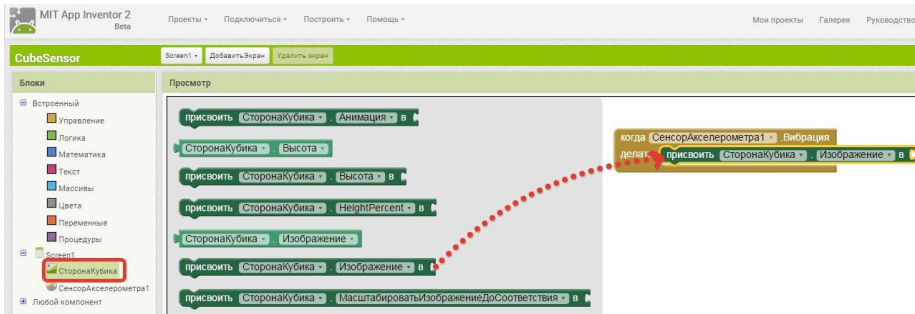
14. Выбрать компонент **Screen1** и установить его свойства **ВыровнятьПоГоризонтали**, **ВыровнятьПоВертикали**, **ОриентацияЭкрана**



15. Перейти в режим Блоки в меню справа, выбрать **СенсорАкселерометра1** и перетащите блок **когда.СенсорАкселерометра1.Вибрация** в поле блоков программы. Данный блок будет запускаться как только устройство будет подвержено вибрации.



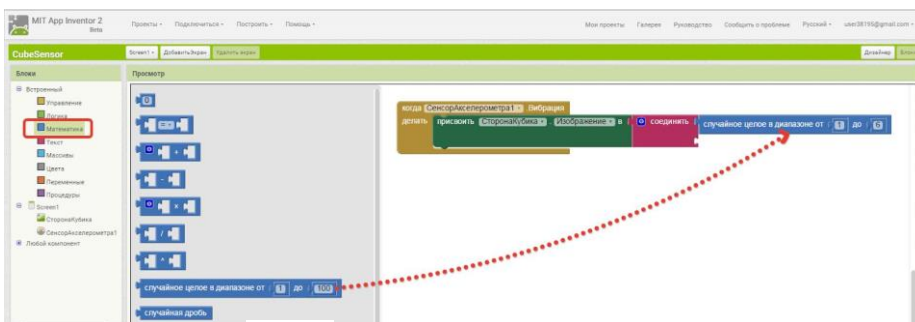
16. Выбрать компонент **СторонаКубика1** и перенести блок **присвоить.СторонаКубика1.изображение** в поле блоков программы. Данный блок выводит изображение графического файла на экран мобильного устройства.




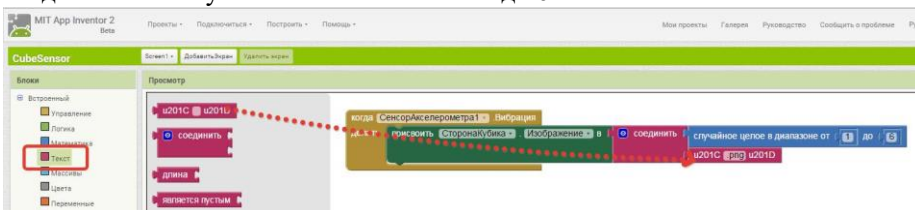
17. Для изображений сторон кубика (файлы 1.png-6.png) имя файла изображения формируется с помощью функции “соединить”: случайное число в диапазоне от 1 до 6 (у нас 6 сторон кубика), плюс расширение графического файла .png.



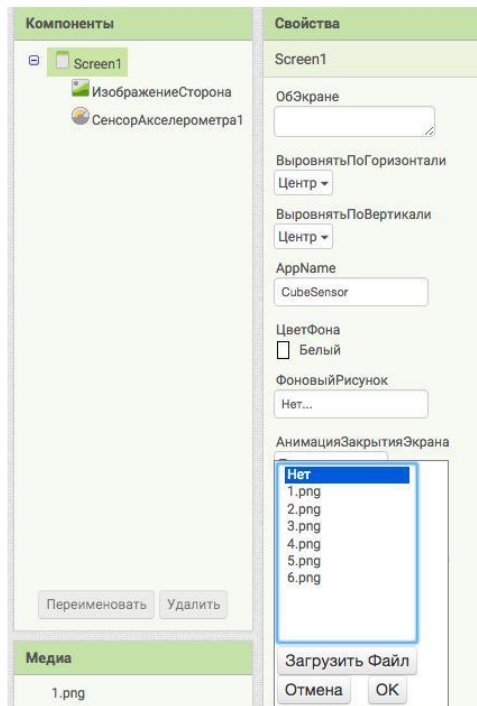
18. Выбрать Математика->Случайное целое от 1 до 100 и установить значения диапазона от 1 до 6.



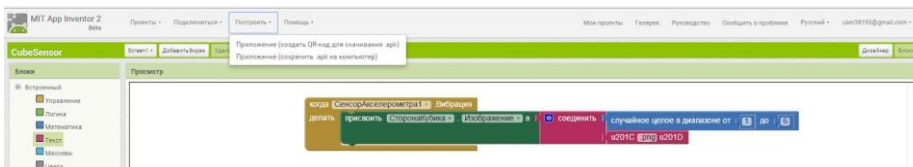
19. Добавить блок  , и вписать туда текст “.png” для последующего соединения со случайным значением от 1 до 6.



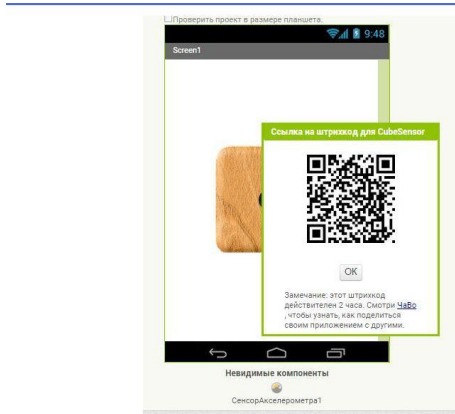
20. Оформить приложение и установить иконку в свойствах компонента Screen1.



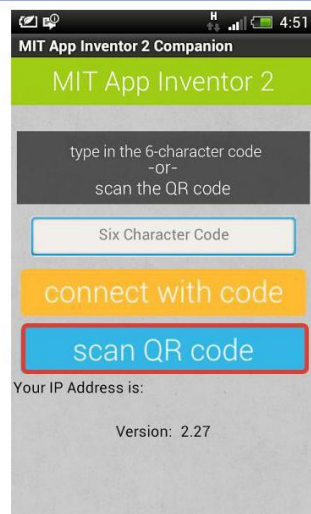
21. Программа готова, необходимо загрузить ее на мобильное устройство для этого нажать Построить->Приложение (Создать QR-код для скачивания .apk)



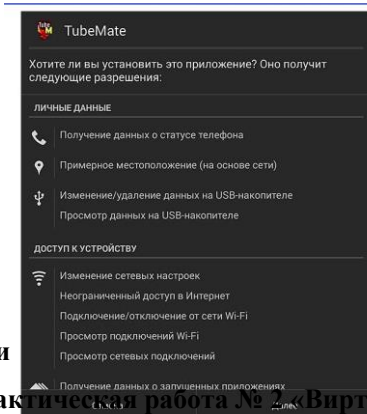
22. Получить QR-код программы



23. Запустить на мобильном устройстве MIT AI2 Companion App и просканировать QR-код приложения



24. Установить приложение на мобильное устройство. Открыть его.



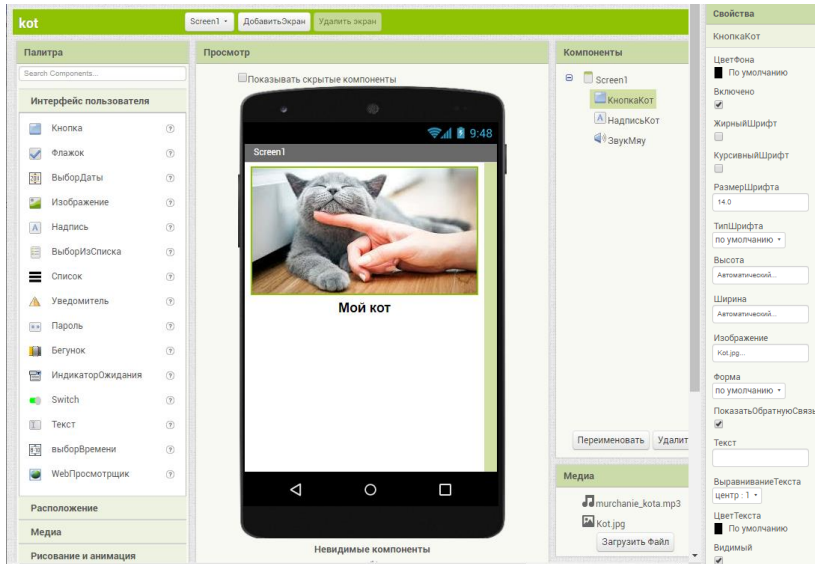
2.2 Кнопки

Практическая работа № 2 «Виртуальный кот»

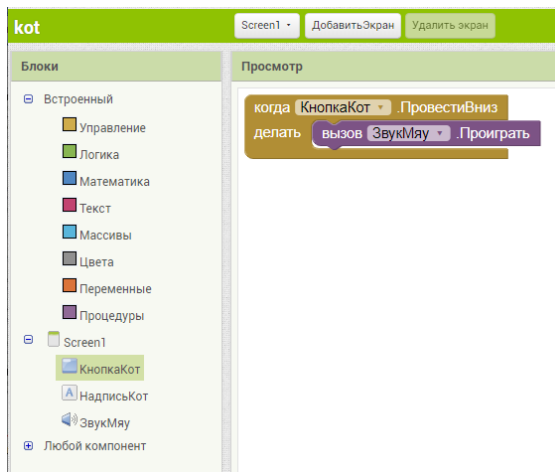
Приложение, в котором кот на экране издает звук, когда его погладят.

1. Для начала в режиме **Дизайнер** перенесите компонент **Кнопка** в окно экрана мобильного устройства. Переименуйте компонент по примеру и

загрузите изображение «Kot.jpg» с помощью кнопки **Загрузить файл**.
Свойства кнопки задайте по примеру.



2. Выберите компонент **Надпись**, задайте новое имя.
3. Добавьте компонент **Звук**, переименуйте и загрузите файл «murchanie_kota.mp3».
4. Перейти в режим **Блоки**, в меню слева выбрать **КнопкаКот** и перетащите соответствующие блоки.



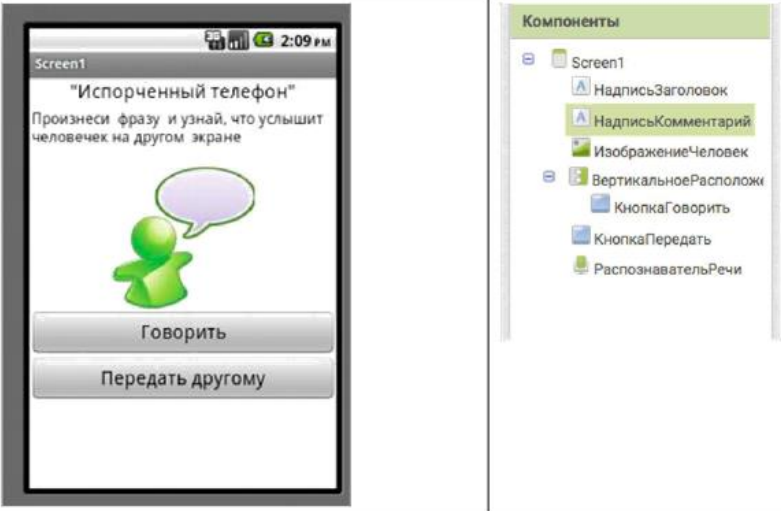
5. Программа готова. Необходимо загрузить ее на мобильное устройство, для этого нажать **Построить/Приложение (Создать QR-код для скачивания .apk)**.
6. Получить QR-код программы и скачать на мобильное устройство.

2.3 Создание приложений с несколькими экранами

Практическая работа № 3 Приложение «Испорченный телефон»

Описание: Приложение, которое будет передавать услышанный текст от одного собеседника (первый экран), второму собеседнику (второй экран) с использованием функции передачи значений между экранами.

Screen1

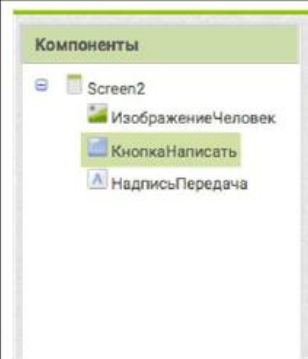
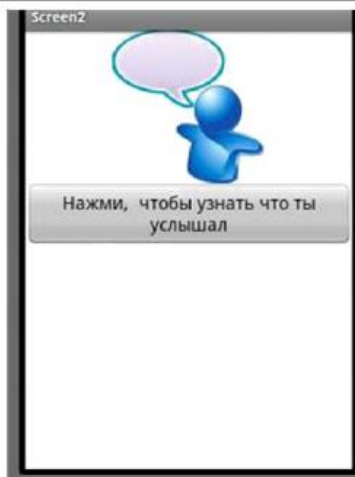


The screenshot shows the application interface on the left and a component list on the right. The interface has a title "Испорченный телефон" and instructions "Произнеси фразу и узнай, что услышит человек на другом экране". It features a green person icon with a speech bubble, and two buttons: "Говорить" and "Передать другому". The component list includes: Screen1, НадписьЗаголовок, НадписьКомментарий, ИзображениеЧеловек, ВертикальноеРасположи, КнопкаГоворить, КнопкаПередать, and РаспознавательРечи.

когда КнопкаГоворить.Щелчок
делать вызов РаспознавательРечи.ПолучитьТекст

когда КнопкаПередать.Щелчок
делать открыть ещё один экран с начальным значением названиеЭкрана и201C Screen2 и201D начальноеЗначение РаспознавательРечи.Результат

Screen 2



```
когда Screen2 .Инициализировать
  делать присвоить global текст в получить начальное значение

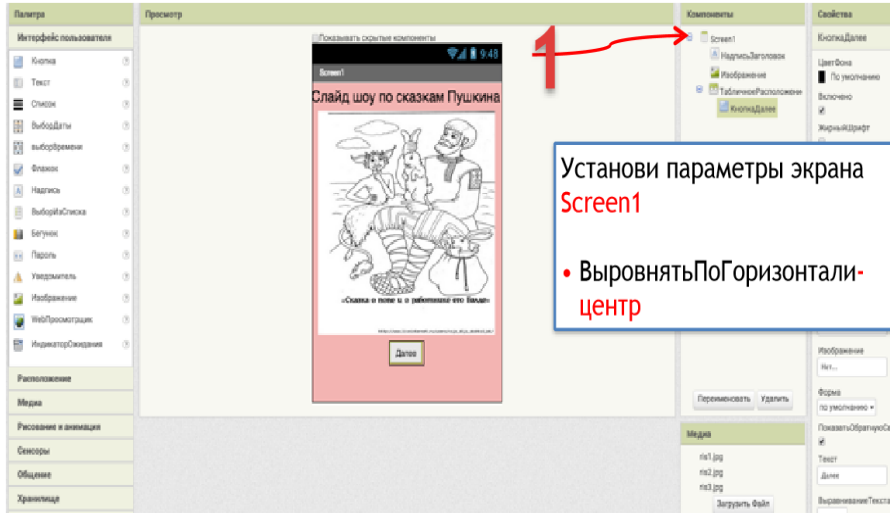
инициализировать глобальную текст в u201C u201D

когда КнопкаНаписать .Щелчок
  делать присвоить НадписьПередача .Текст в получить global текст
```

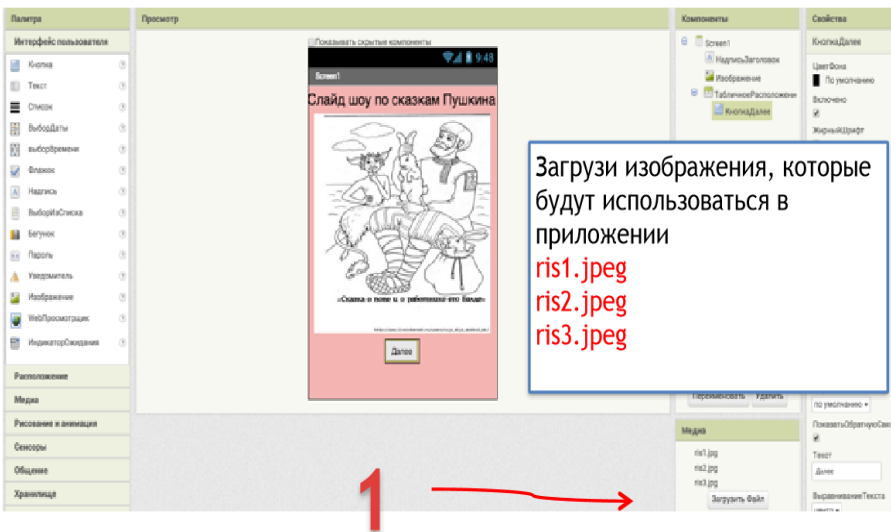
2.4 Создание приложения с использованием списков

Практическая работа №4 «Слайд-шоу»

1)



2)

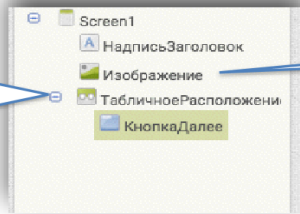
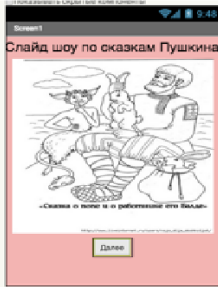


3

3)

Проектируем дизайн приложения

Перенеси на экран и настрой следующие компоненты



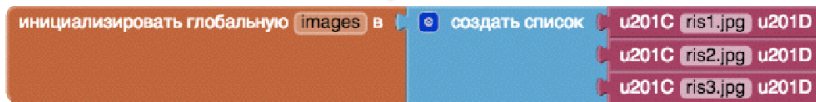
Ширина 300 px, высота 200 px
источник изображения
ris1.jpeg

4

4)

Собираем конструкцию. Шаг 1. Определяем переменные

Массив изображений

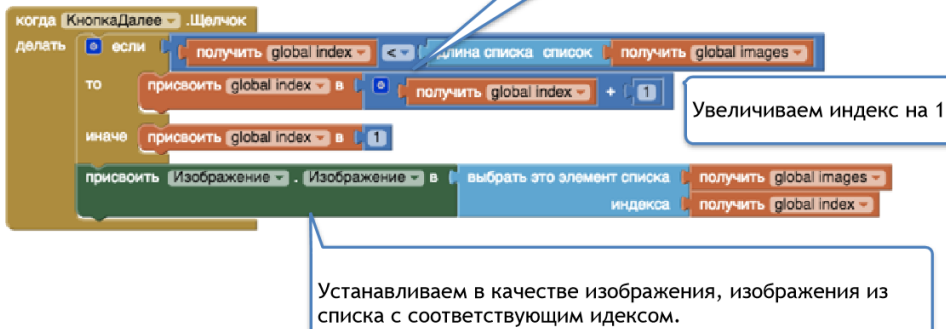


инициализировать глобальную `index` в `1`

Индекс-номер изображения для вывода на экран, изображение Ris1 выводится на первом экране

5)

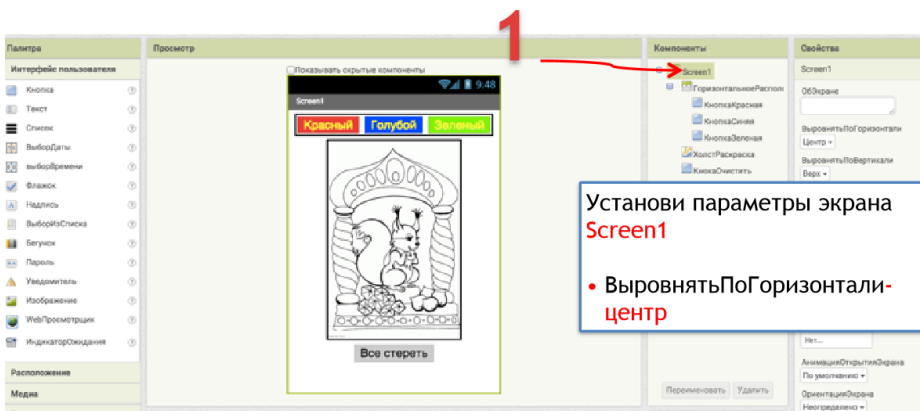
Собираем конструкцию. Шаг 2. Программируем кнопку



2.5 Создание приложения с использованием элементов группы «Рисование»

Практическая работа №5 «Рисование»

1)



2)

Проектируем дизайн приложения

1

Перенеси на **Screen1**, переименуй и настрой следующие компоненты

- НадписьЗаголовок
- КнопкаКрасный
- КнопкаСиний
- КнопкаЗеленый
- ХолстРаскраска
- КнопкаОчистить

3

3)

1

В качестве изображения для Холста установи файл-раскраску **raskraska.jpeg**

4

4)

Собираем конструкцию. Шаг 1. Программируем Кнопки



Scratch code blocks for button actions:

- КнопкаКрасная: когда Щелчок, делать присвоить ХолстРаскраска ЦветКраски
- КнопкаСиняя: когда Щелчок, делать присвоить ХолстРаскраска ЦветКраски
- КнопкаЗеленая: когда Щелчок, делать присвоить ХолстРаскраска ЦветКраски
- КнопкаОчистить: когда Щелчок, делать вызов ХолстРаскраска Очистить

Callouts:

- Задаем цвет для рисования
- Вызываем действие для Очистки холста

5)

Собираем конструкцию. Шаг 2. Программируем Рисование круга



Scratch code block for drawing a circle:

- когда ХолстРаскраска .Касание: x, y, touchedAnySprite
- делать вызов ХолстРаскраска .РисоватьКруг: centerX, centerY, radius, fill

Callouts:

- Определение координат точки касания, для задания центра круга
- Радиус круга, которым рисуем
- Заливка круга цветом

Собираем конструкцию. Шаг 3. Программируем Рисование линий



Scratch code block for drawing a line:

- когда ХолстРаскраска .Перетащенный: начальнаяX, начальнаяY, предыдущX, предыдущY, текущийX, текущийY
- делать вызов ХолстРаскраска .РисоватьЛинию: x1, y1, x2, y2
- присвоить ХолстРаскраска .ШиринаЛинии: 10

Callouts:

- Определение координат начала и конца линии.
- Толщина линии

2.6 Создание приложения с использованием анимации

Практическая работа № 6 «Поймай крота»

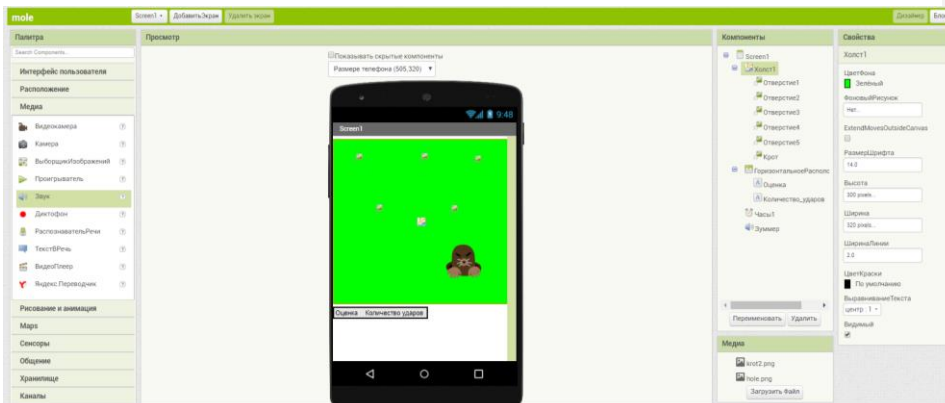
Цель состоит в том, чтобы нажать на крота, который случайно выпадает из одного из пяти фиксированных отверстий. Каждый раз, когда вы добиваетесь успеха, ваш счет увеличивается на одно очко.

Начало

Откройте ai2.appinventor.mit.edu и начните новый проект. Установите для свойства *заголовка* экрана подходящее имя, например, «Моле Меш». Загрузите приведенные ниже файлы изображений на свой компьютер, щелкнув их правой кнопкой мыши, а затем добавьте их в свой проект, нажав кнопку «Загрузить файл ...» на панели «Медиа».

Установите компоненты

Пользовательский интерфейс будет содержать в общей сложности 6 ImageSprites: 5 неподвижных отверстий и 1 моль. Крот будет двигаться по-верх дыр. Используйте конструктор для создания пользовательского интерфейса. Когда вы закончите, это должно выглядеть примерно так, как на картинке ниже. Не беспокойтесь о выравнивании отверстий равномерно. Вы будете определять их местоположение через их свойства *X* и *Y*. Дополнительные инструкции находятся под изображением.



Внесите следующие изменения в свойства компонентов:

Компонент	действие
Холст1	Установите <i>Цвет фона</i> в зеленый. Установите <i>ширину</i> до 320 пикселей. Установите <i>Высота</i> до 320 пикселей.
Отверстие1	Установите <i>X</i> на 20 и <i>Y</i> на 60 (вверху слева).
Отверстие2	Установите <i>X</i> на 130 и <i>Y</i> на 60 (верхний центр).
Отверстие 3	Установите <i>X</i> на 240 и <i>Y</i> на 60 (вверху справа)
Отверстие 4	Установите <i>X</i> на 75 и <i>Y</i> на 140 (внизу слева).
Отверстие 5	Установите <i>X</i> на 185 и <i>Y</i> на 140 (внизу справа).
Крот	Установите <i>изображение</i> на "krot2.png". Установите <i>Z</i> на 2, чтобы крот появился перед другими <i>ИзображениямиСпрайта</i> , которые имеют значение <i>Z</i> по умолчанию 1.

Надпись11 Установите *текст* в «Оценка:».

Надпись2 Установите *текст* в «0».

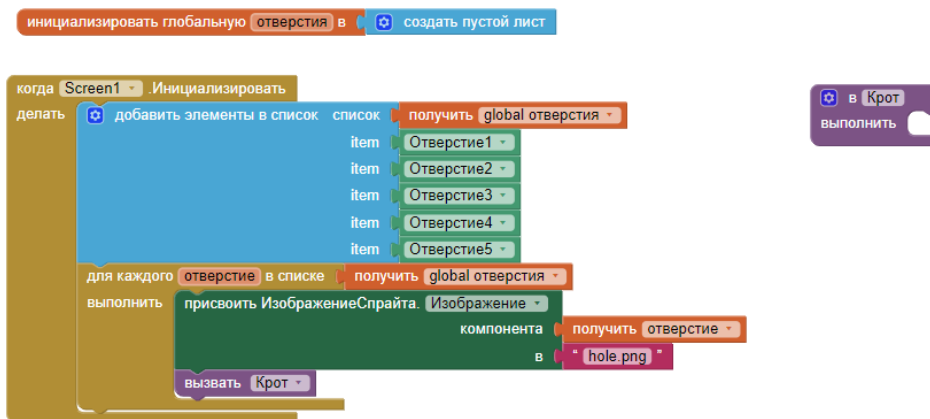
Не беспокойтесь о настройке свойства *Изображение* для отверстий; мы установим свойство в редакторе блоков.

Создание переменных

Создайте переменную и назовите ее `отверстиями`.

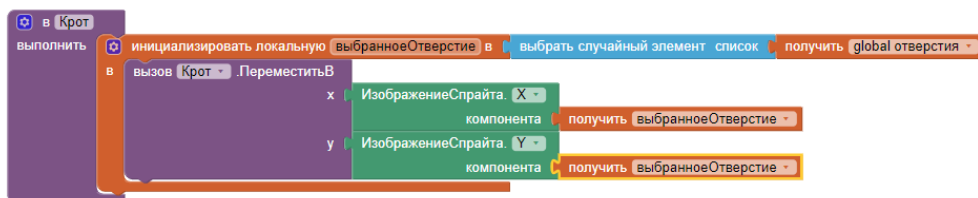
инициализировать глобальную `отверстия` в `создать пустой лист`

Обратите внимание, что ящик «присвоить ИзображениеСпрайта» находится на вкладке «Любой компонент» в нижней части списка блоков в редакторе блоков.

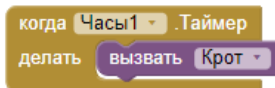


Перемещение Крота

Теперь давайте заполним тело процедуры `Крот`, которую мы будем вызывать при запуске программы, при касании крота и при отключении нашего таймера каждую секунду. Процедура должна выбрать случайное отверстие и переместить родинку поверх него. Вот скомпилированные блоки и таблица используемых блоков:



Теперь нам нужно указать, что **Крот** должен вызываться всякий раз, когда срабатывает таймер. Нам просто нужно два блока для этого:

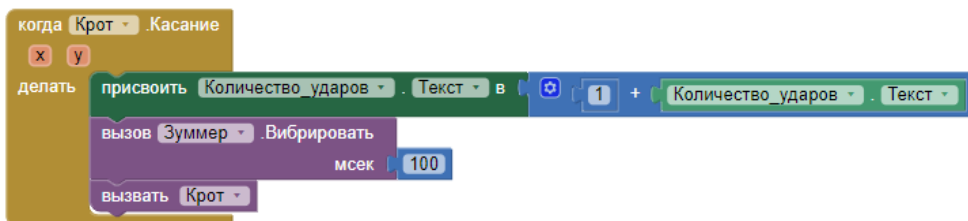


Регистрация штрихов

Наконец, нам нужно указать, что происходит, когда крот прикасается. В частности, мы хотим:

1. Увеличить счет.
2. Заставить телефон ненадолго вибрировать.
3. Переместить крота.

Мы можем легко перевести их в блоки:



Вариации

Вот некоторые варианты, которые вы можете реализовать:

- Добавление кнопки сброса, чтобы установить счет обратно на 0.
- Количество очков зависит не только от количества попаданий, но и от числа пропущенных и избежавших отверстий.
- Увеличение скорости игры с кротом, если у игрока все в порядке, и ее снижение, если у игрока плохо.

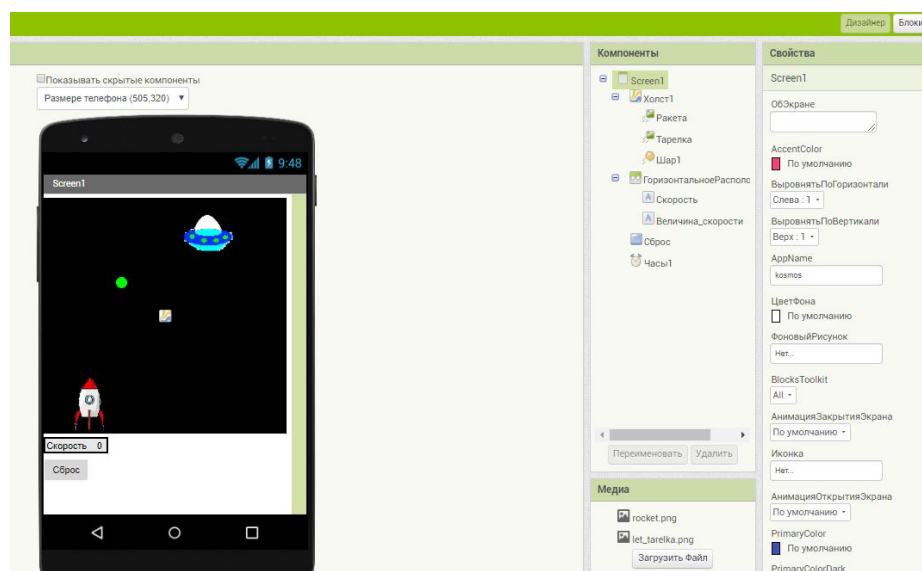
2.7 Создание приложения с использованием элементов группы «Медиа»

Практическая работа № 7 Игра «Космические захватчики»

Построив эту игру, вы получите практику с использованием компонентов Часов и таймеров, используя компоненты анимации, такие как изображения спрайтов и холсты, настройки видимости и обнаружения столкновений в App Inventor. Вы запрограммируете приложение, в котором есть корабль-шутер, цель которого - выстрелить во все летающие тарелки на экране.

Установите компоненты

Используйте конструктор компонентов, чтобы создать интерфейс для игры. Когда вы закончите, он должен выглядеть примерно, как снимок экрана ниже (более подробные инструкции ниже снимка).



Чтобы создать этот интерфейс, поместите следующие компоненты в Designer, перетащив их из палитры компонентов в Viewer и установите свойства компонентов, как описано ниже:

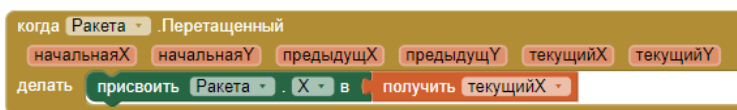
Тип компонента	Назначение компонента	действие
Холст	Фон, на котором мы будем помещать наши спрайты	Измените свойство Высота на «300» и свойство Ширина на 320. Установите для свойства Цвет значение «Черный».
Изображение спрайта	Ракетный корабль в нашей игре	Загрузите изображение ракеты и установите для свойства Изображение значение

		«rocket.png». Установите для свойства Y значение 230. Это поместит ракету внизу холста.
Изображение спрайта	Летающая тарелка в нашей игре	Загрузите изображение блюда и установите для свойства Изображение значение «let_tarelka.png».
Шар	Пуля с ракетного корабля.	Измените Цвет на Зеленый и установите для свойства Радиус значение 8.
Часы	Мы используем Часы для его метода Таймер, чтобы переместить блюдо	Измените свойство ИнтервалТаймера на 3000.
Горизонтальное расположение	Содержать надписи	
Надпись	Содержать слово «Оценка»	Измените свойство текст на «Оценка:».
Надпись	Содержать текущий числовой счет	Измените свойство текст на «0».
Кнопка	Сбросить игру, чтобы игрок мог снова играть	Измените свойство текст на «Сброс».

Теперь, когда вы настроили все необходимые свойства, не стесняйтесь менять цвета любых компонентов, которые вы хотите.

Перемещение ракеты

В этой игре пользователь будет перемещать ракету из стороны в сторону. Это означает, что мы будем изменять только направление X спрайта ракеты. Для этого мы будем использовать обработчик событий Ракета.Перетащенный. Когда ракета перетаскивается, мы настроим ее свойство X, чтобы оно было текущим X, к которому мы перетаскивали спрайт.

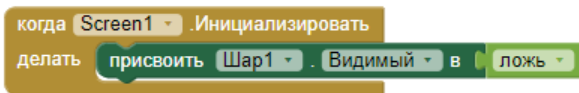


Как только вы соедините эти блоки, подключите свой телефон и протестируйте эту функцию!

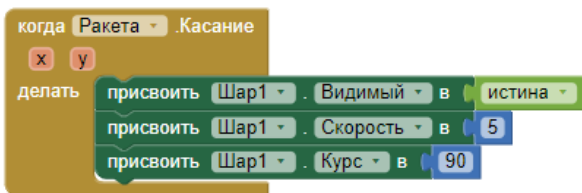
Программирование поведения пули

Есть несколько функций, которые мы хотим, чтобы наша пуля имела в этой игре. Мы хотим, чтобы она стреляла из ракеты, сталкивалась с блюдцем и была невидима после столкновения и до того, как его застрелили.

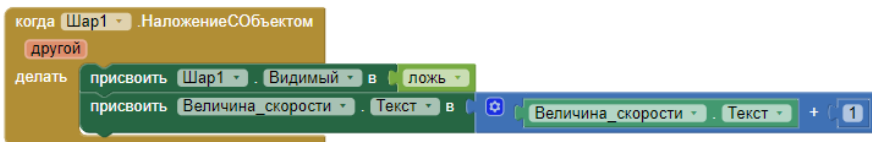
Начнем с использования блока Screen1.Инициализировать. Когда экран инициализируется, мы запрограммируем пулю как невидимую. Мы делаем это, устанавливая свойство видимости пули в Ложь.



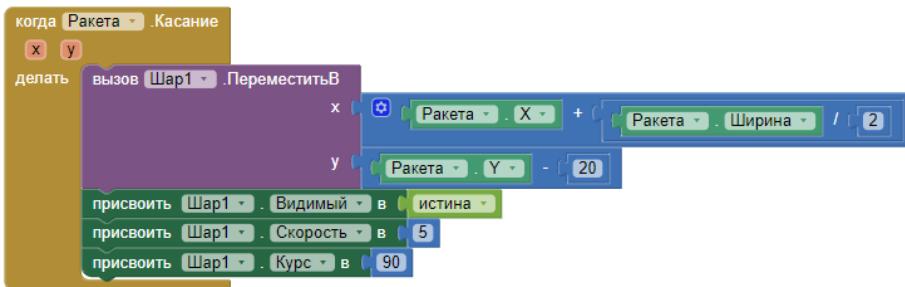
Далее мы хотим убедиться, что пуля появляется снова, когда мы стреляем из ракеты. Когда мы дотрагиваемся до ракеты, мы хотим, чтобы пуля направилась к тарелке. Мы сделаем это с помощью обработчика событий Ракета.Касание. При касании ракеты мы не только хотим, чтобы ракета была видимой, но мы также хотим установить скорость и курс ракеты. Направление - это значение от 0 до 360, которое указывает, в каком направлении должен двигаться спрайт. 0/360 слева, 90 вверх, 180 справа и 270 вниз. Скорость измеряется в пикселях / сек.



Последнее, что нам нужно запрограммировать, это то, что происходит, когда пуля попадает в блюдо. Мы будем использовать обработчик событий Bullet.CollidedWith. Это событие вызывается всякий раз, когда пуля сталкивается с другим спрайтом. Так как наш спрайт ракеты заблокирован в Y внизу экрана, пуля никогда не столкнется с ракетой и только с блюдцем. При столкновении мы хотим, чтобы произошли две вещи. 1. Счет должен увеличиться на 1. 2. Пуля должна стать невидимой.

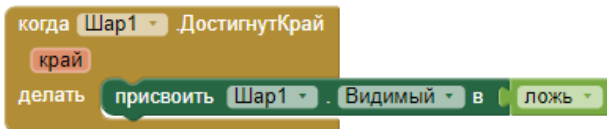


Если вы начали тестировать эту игру, вы, возможно, заметили, что, как только вы стреляете в пулю, она, похоже, не позволяет вам снова стрелять в нее. Нам нужно запрограммировать, чтобы пуля возвращалась к месту перед ракетой, когда мы стреляем. Мы можем сделать это с помощью блока Шар.ПереместитьВ.



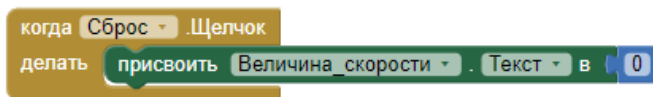
Теперь проверь это!

Возможно, вы заметили, что, если вы пропустите блюдо, пуля перемещается в верхнюю часть экрана и застревает там, пока вы не попытаете стрелять снова. Чтобы пуля исчезла при попадании в верхний край нашего холста, нам нужно использовать обработчик событий Шар.ДостигнутКрай.



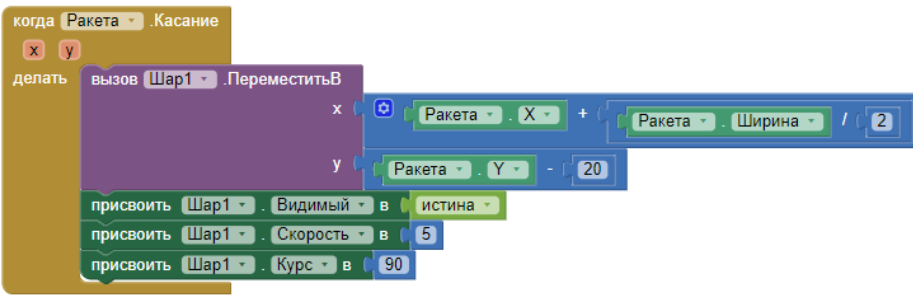
Программирование кнопки сброса

Иногда пользователи могут захотеть перезапустить игру и сбросить счет. Когда это происходит, нам нужно установить счет обратно на 0.



Увеличение сложности - изменение положения блюда

Давайте сделаем игру немного сложнее! Теперь, когда пуля сталкивается с тарелкой, давайте изменим местоположение тарелки. Тарелка будет сохранять то же значение Y, поэтому нам нужно будет только изменить X. Мы можем сделать это с помощью случайного блока.



Чтобы сделать это еще сложнее, мы также изменим положение тарелки, когда таймер выключится.

```
когда Часы1 . Таймер  
  делать присвоить Тарелка . X в случайное целое в диапазоне от 0 до Холст1 . Ширина - Тарелка . Ширина
```