

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, информатики и информационных технологий
Кафедра информатики, информационных технологий
и методики обучения информатике

**Использование визуального языка
программирования Kodu для активизации
учебной деятельности школьников на уроках
информатики и ИКТ**

Выпускная квалификационная работа

Направление подготовки:
44.03.01 Педагогическое образование (профиль «Информатика»)
Направленность программы, уровень:
бакалавриат
Квалификация (степень):
бакалавр
Форма обучения:
заочная

Исполнитель: Выгузов А. А.
Руководитель: Рожина И. В.

Екатеринбург 2016 г.

Оглавление

Введение	3
Глава 1. Особенности учебной деятельности школьников в начальных классах	4
1.1. Психолого-педагогические особенности учащихся начальных классов ...	4
1.2. Использование различных программных комплексов для изучения информатики и ИКТ.....	10
1.3. Особенности визуального языка программирования Kodu.	13
Или же выберем пункт Игра -> Жизнь.	18
Глава 2. Методика использования визуального языка программирования Kodu на уроках информатики.....	19
2.1. Методические рекомендации по изучению языка программирования. ...	19
2.2. Разработка системы уроков	20
Список литературы	37

Введение

Данная тема очень актуальна в текущее время, поскольку при обучении информатике на начальных этапах из-за сильной фокусировки на теоретической части, у многих обучаемых пропадает интерес к предмету. Это приводит к малому количеству хороших специалистов в дальнейшем, т.к. все ИКТ – незаменимая часть нашего времени, и они постоянно развиваются, в связи с чем, необходимо максимально заинтересовать детей изучать данную сферу деятельности и сам предмет.

Объект исследования – процесс обучения учащихся начальных классов.

Предмет исследования – методика использования визуального языка программирования Kodu на уроках информатики и ИКТ в начальной школе.

Цель работы – разработка уроков для школьников начальных классов, направленных на активизацию учебной деятельности школьников на уроках информатики и ИКТ.

Задачи исследования:

1. Выполнить анализ:
 - психолого-педагогической литературы с целью изучения психолого-педагогических особенностей учащихся начальных классов.
 - специальной методической литературы по использованию программных комплексов и возможностей их применения в начальных классах.
2. Разработать серию уроков с использованием визуального языка программирования Kodu.
3. Разработать методические материалы к урокам.

Глава 1. Особенности учебной деятельности школьников в начальных классах

1.1. Психолого-педагогические особенности учащихся начальных классов

Начальными классами принято считать 1-4 класс. Данный этап развития ученика считается младшим школьным возрастом, который условно определен от 6-7 до 10-11 лет. В данном возрасте основным видом деятельности принято считать учебную деятельность. Сам младший школьный возраст – этап развития ребёнка, в котором развитие высших психических функций и личности в целом происходит в рамках учебной деятельности, которая заменяет ведущую на предыдущем этапе (дошкольный возраст) игровую деятельность. Соответственно, данный переход необходимо организовывать плавно и гармонично, чтобы не было разочарования в учебной деятельности, что вызывает отсутствие желания учиться.

Первоначально необходимо сформировать интерес обучаемого к самой учебной деятельности, чтобы в дальнейшем это переросло в желание учиться и получать новые знания. Данное формирование интереса в основном происходит за счет получения удовлетворения школьниками от своих достижений. А подкрепляется оно одобрением, похвалой со стороны учителя, который будет подчеркивать каждый успех, даже самое малое продвижение вперед, из-за чего школьники будут испытывать чувство гордости и своеобразный, особый подъем сил, помогающий им двигаться в обучении дальше.

Учебная деятельность в начальных классах стимулирует, прежде всего, развитие психических процессов непосредственного познания окружающего мира – ощущений и восприятий. Младшие школьники отличаются остротой и свежестью восприятия, своего рода созерцательной любознательностью.

Наиболее характерная черта восприятия в данном возрасте – малая дифференцированность, где совершают неточности и ошибки в дифференцировке при восприятии схожих объектов. Следующая особенность восприятия учащихся в начале младшего школьного возраста – тесная связь его с действиями школьника. Восприятие на этом уровне психического развития связано с практической деятельностью ребёнка. Воспринять предмет для ребёнка – значит что-то делать с ним, что-то изменить в нём, произвести какие-либо действия, взять, потрогать его. Характерная особенность учащихся – ярко выраженная эмоциональность восприятия.

В процессе обучения у детей происходит перестройка восприятия, оно становится более развитым и принимает характер целенаправленной и управляемой деятельности. В процессе обучения восприятие углубляется, становится более анализирующим, дифференцирующим, принимает характер организованного наблюдения.

Вниманию учащихся начальных классов присущи некоторые возрастные особенности. Основная особенность – слабое произвольное внимание, поскольку возможности волевого регулирования внимания и управления им в начале младшего школьного возраста сильно ограничены. Младший школьник может заставить себя сосредоточенно работать только при наличии близкой мотивации (заслужить похвалу учителя, возможность получить отличную отметку, справиться лучше всех и т.д.), которая в дальнейшем помогает улучшить произвольные внимания для более длительного его поддержания в старшем возрасте. Значительно лучше в младшем школьном возрасте развито непроизвольное внимание. Всё новое, неожиданное, яркое, интересное само собой привлекает внимание учеников, без всяких усилий с их стороны.

Особенности памяти в младшем школьном возрасте развиваются под влиянием обучения – запоминание становится регулируемым процессом, возможность сознательно управлять своей памятью развивается. Усиливается

роль и удельный вес словесно-логического, смыслового запоминания. Младшие школьники гораздо лучше запоминают и прочнее сохраняют в памяти конкретные сведения, события, лица, предметы или факты, чем какие-либо определения, описания или объяснения, т.к. в связи с возрастным относительным преобладанием деятельности первой сигнальной системы у младших школьников более развита наглядно-образная память, а не словесно-логическая. Из-за этих особенностей младшие школьники склонны к механическому запоминанию без осознания смысловых связей внутри запоминаемого материала.

Помимо внимания и памяти, у младших школьников начинает развиваться воображение. Основная тенденция развития воображения в данном возрасте – совершенствование воссоздающего воображения, которое связано с представлением ранее воспринятого или созданием образов в соответствии с предоставленным описанием, схемой, рисунком и т.д. Развитие воссоздающего воображения происходит за счет все более правильного и полного отражения действительности. Творческое воображение, в свою очередь, отвечает за создание новых образов, и полностью связано с преобразованием, переработкой впечатлений прошлого опыта, соединяет их в новые сочетания и так же развивается.

Под влиянием обучения происходит постепенный переход от познания внешней стороны явлений к познанию их сущности. Мышление начинает отражать существенные свойства и признаки предметов и явлений, что дает возможность делать первые обобщения, выводы, проводить первые аналогии и строить элементарные умозаключения, на основе чего у младшего школьника постепенно начинают формироваться элементарные научные понятия.

Младший школьный возраст – возраст достаточно заметного формирования личности. Для него характерны новые отношения с взрослыми и сверстниками, включение в целую систему коллективов,

включение в новый вид деятельности – учение, которое предъявляет ряд серьёзных требований к ученику. Всё это решающим образом сказывается на формировании и закреплении новой системы отношений к людям, коллективу, к учению и связанным с ними обязанностям, формирует характер, волю, расширяет круг интересов, развивает способности.

В младшем школьном возрасте закладывается фундамент нравственного поведения, происходит усвоение моральных норм и правил поведения, начинает формироваться общественная направленность личности. Характер младших школьников отличается некоторыми особенностями. Прежде всего, они импульсивны – склонны незамедлительно действовать под влиянием непосредственных импульсов, побуждений, не подумав и не взвесив всех обстоятельств, по случайным поводам. Причина – потребность в активной внешней разрядке при возрастной слабости волевой регуляции поведения.

Возрастной особенностью является и общая недостаточность воли: младший школьник ещё не обладает большим опытом длительной борьбы за намеченную цель, преодоления трудностей и препятствий. Он может опустить руки при неудаче, потерять веру в свои силы и невозможности. Нередко наблюдается капризность, упрямство. Обычная причина их – недостатки семейного воспитания. Ребёнок привык к тому, что все его желания и требования удовлетворялись, он ни в чём не видел отказа. Капризность и упрямство – своеобразная форма протеста ребёнка против тех твёрдых требований, которые ему предъявляет школа, против необходимости жертвовать тем, что хочется, во имя того, что надо.

Младшие школьники очень эмоциональны. Эмоциональность сказывается, во-первых, в том, что их психическая деятельность обычно окрашена эмоциями. Всё, что дети наблюдают, о чём думают, что делают, вызывает у них эмоционально окрашенное отношение. Во-вторых, младшие школьники не умеют сдерживать свои чувства, контролировать их внешнее

проявление, они очень непосредственны и откровенны в выражении радости. Горя, печали, страха, удовольствия или неудовольствия. В-третьих, эмоциональность выражается в их большой эмоциональной неустойчивости, частой смене настроений, склонности к аффектам, кратковременным и бурным проявлениям радости, горя, гнева, страха. С годами всё больше развивается способность регулировать свои чувства, сдерживать их нежелательные проявления.

Большие возможности предоставляет младший школьный возраст для воспитания коллективистских отношений. За несколько лет младший школьник накапливает при правильном воспитании важный для своего дальнейшего развития опыт коллективной деятельности – деятельности в коллективе и для коллектива. Воспитанию коллективизма помогает участие детей в общественных, коллективных делах. Именно здесь ребёнок приобретает основной опыт коллективной общественной деятельности.

Изучение информатики и информационных технологий в начальной школе направлено на достижение следующих целей:

- Развитие умений ориентироваться в информационных потоках окружающего мира;
- Овладение практическими способами работы с информацией: поиск, анализ, преобразование, передача, хранение информации, ее использование в учебной деятельности и повседневной жизни;
- Формирование начальной компьютерной грамотности и элементов информационной культуры;
- Развитие умений, позволяющих обмениваться информацией, осуществлять коммуникации с помощью имеющихся

технических средств (телефон, магнитофон, компьютер, телевизор и др.).

Целесообразно формировать у младших школьников не только элементы компьютерной грамотности, но и начальные знания основ информатики, осуществлять пропедевтику ее фундаментальных понятий и способов деятельности, т.е.:

- Формирование представлений о возможностях компьютера в области обработки информации (компьютер рисует, вычисляет, пишет...), хранения информации (компьютер запоминает рисунки, числа, тексты) и передачи информации (от человека к компьютеру, от компьютера к компьютеру, от компьютера к человеку).
- Знакомство с основными понятиями и их отношениями как общими: информация, язык, знания, модель, алгоритм, структура и др. так и частными: данные, таблицы данных, объекты, имена, значения, условия, структуры записей процедурных ответов.
- Формирование навыков работы с клавиатурой (набор текста, набор чисел, управление движением и фиксирование объектов на экране дисплея, выбор режимов работы по меню).
- Освоение способов деятельности, отражающих специфические методы информатики: формализацию (описание условий и решений задач с ориентацией на их компьютерное исполнение), алгоритмизацию (запись для некоторого типа процедурных ответов простейшей структуры типа "ветвление", "выбор", "цикл"), решение практических задач с применением компьютера (ведение компьютерных дневников при наблюдениях за погодой, жизнью растений и животных).

1.2. Использование различных программных комплексов для изучения информатики и ИКТ

В процессе обучения детей информатике используются различные программные комплексы. Большинство этих программных комплексов базируются на каком-то широко используемом языке программирования, реже – на своем собственном. Но у всех есть одно общее сходство – необходимо прописывать программу вручную, вводя слова согласно используемому в программе синтаксису языка. Самые распространенные из них представлены ниже:

1. ЛОГО

Ещё в конце 1960 годов американским педагогом и программистом С.Пейпертом для обучения детей алгоритмизации был разработан специальный учебный язык программирования ЛОГО, в состав которого входил исполнитель Черепашка, позволявший изображать на экране компьютера чертежи и рисунки, состоящие из отрезков прямых линий. Система команд Черепашки включала в себя команды: вперед, назад, налево, направо, поднять хвост, опустить хвост (Черепашка рисует хвостом, когда он опущен). Язык ЛОГО имел основные структурные команды и позволял обучать структурной методике программирования. Большим методическим достоинством исполнителя Черепашка являлась его наглядность в процессе выполнения команд. В языке Лого программа представляет собой список «слов». Данные слова разделяются между собой пробелами. Слово может быть либо командой для исполнения, либо литералом одного из 3-х видов – строкой, числом или списком. Команду можно сделать функцией, и в таком случае она будет возвращать значение. Список команд, переданных интерпретатору, выполняется строго по очереди, при этом каждое слово интерпретируется как команда из текущего словаря системы. Есть поддержка математических функций в привычном нам виде. Есть команда для расширения словаря пользовательскими командами, она принимает имя

новой команды, список параметров и список команд. Предусмотрены два основных механизма управления ходом выполнения программы: оператор цикла с фиксированным числом повторений и рекурсия.

2. Робик

Группой академика А.П. Ершова для обучения программированию был разработан язык Робик, в котором использовалось несколько различных исполнителей. Дальнейшее развитие идей академика А.П. Ершова по обучению алгоритмизации нашло в учебнике А.Г. Кушниренко, в котором основным методическим приёмом стало использование учебных исполнителей Робота и Чертежника. Робот предназначен для перемещения по полю из клеток с разными стенками и выполнению при этом различных заданий: закрашивать клетки, измерять температуру и радиацию. Причем Робот управляется компьютером, который подает ему управляющие команды, и получает от него ответы на запросы о текущей обстановке. Таким способом осуществляется идея обратной связи, что позволяет создавать для управления работой исполнителя алгоритмы сложной структуры, содержащие циклы и ветвления. Чертежник предназначен для выполнения в системе координат чертежей, графиков, рисунков, состоящих из прямолинейных отрезков. Его работа во многом подобна действиям Черепашки. Языком описания для этих исполнителей является учебный алгоритмический язык, основы которого разработал академик А.П. Ершов.

3. LightBot

Игрушечная среда для самых маленьких (можно использовать с 5-6 лет). Необходимо запрограммировать движение «виртуального» робота. Несмотря на простоту, с помощью LightBot можно не только формировать операционное мышление ребенка, но и изучать такие понятия программирования, как подпрограммы-процедуры.

4. RoboMind

Простая среда программирования, которая позволяет начинающим программировать поведение «машинки»-робота. Здесь в доступной форме изучаются популярные методы программирования и основы «искусственного интеллекта». Робот может программироваться на разных языках.

5. Karel, Karel ++, Karel J. Robot

Это языки для начинающих, они используются для составления программ управления «роботом». Karel использует собственный язык программирования, Karel ++ – язык программирования C++, Karel J. Robot – версия Karel на Java.

6. Guido van Robot

Здесь роботом управляют с помощью программы (как в Karel J. Robot), которая своим синтаксисом похожа на Python. Это свободно расширенная программа, работа с которой может быть началом к изучению любого языка программирования.

7. Greenfoot

Учебная, объектно-ориентированная Java-среда, разработанная, прежде всего, в учебных целях. Greenfoot позволяет разрабатывать программы для моделирования и диалоговые игры.

8. Little Wizzard

Это среда программирования для детей, предназначенное для изучения основных элементов программирования в начальной школе. Используя только мышку, дети получают возможность составлять программы и изучать такие понятия, как переменные, выражения, ветвления, условия и логические блоки. Каждый элемент языка программирования представляет собой интуитивно понятный символ.

9. Peter

Это средство визуального программирования, предназначенный для простого и быстрого создания приложений для Windows 95/98/NT/ME/2000/XP. Принцип работы похож на разработку графической презентации с созданием программной структуры для управления объектами. Составление программы несколько напоминает составление головоломки из отдельных частиц. Стрелка – программа – тренажер для развития алгоритмического мышления и формирования навыков составления управляющих алгоритмов.

10. Kodu Game Lab

Визуальный язык программирования Kodu предназначен для использования без знания синтаксиса языка. Все действия в программе заранее прописаны и не требуют ввода через клавиатуру, программирование идет в режиме реального времени. Все действия задаются через лист действий, в котором просто выбираются действия, которые без дополнительной реализации будет выполнять выбранный исполнитель. Исполнителей в данном языке программирования много, и для каждого список действий задан по-своему. Так же нет необходимости угадывания команды для исполнения, т.к. весь список выводится в виде удобного и наглядного меню, что облегчает процесс программирования.

1.3. Особенности визуального языка программирования Kodu.

Визуальный язык программирования Kodu – бесплатно распространяемая учебная среда, созданная при совместной деятельности компаний Fuse Lab и Microsoft. Данный язык программирования создан специально для наглядного изучения программирования без изучения сложного синтаксиса языка, как это обстоит в случае с тем же Pascal или Java, благодаря чему к изучению языка может приступить даже первоклассник.

Визуальный язык программирования Kodu рассчитан на создание 3D игр без знания самого языка программирования, для реализации чего в данном языке программирования имеется весь необходимый инструментарий, использование которого интуитивно понятно. В качестве устройства ввода можно использовать как стандартные (клавиатура и мышь), так и геймпад с другой платформы, поддержкой и развитием которой так же занимается компания Microsoft – контроллера для Xbox 360, в связи с чем, меню адаптировано под удобную работу на используемых устройствах ввода. Для начального этапа изучения в данном языке программирования имеются несколько обучающих «миров» (поскольку из-за особенностей работы языка программирования назвать их задачами очень сложно), где поэтапно рассмотрено, как можно реализовать ту или иную модель действий или создать модель поведения в соответствии с ситуацией. На этапе программирования для каждого объекта или предмета можно задать свои свойства, свои настройки поведения и реакции на действия игрока, что позволяет настроить игру полностью под свою идею.

Основные отличительные признаки визуального языка программирования Kodu:

- Низкий порог вхождения – за счет интуитивно понятного интерфейса и хорошего справочного материала, предоставляемого напрямую в программе, освоение языка дается очень легко.
- Многообразие задач – так как этот язык программирования представлен в виде игры, на основе которой можно создавать другие игры, в игре можно смоделировать огромное множество задач, как похожих друг на друга, так и полностью различных между собой.
- Уникальные способы освоения – программировать в данном языке можно не только при помощи стандартных устройств

ввода (такие как клавиатура и мышь), но и при помощи контроллера, чего нет в других языках программирования.

- Количество действий ограничивается только фантазией программиста. В каждой придуманной игре можно задать до 12 листов с действиями, переход между которыми возможен при определенных условиях, или использовать действия с другого листа, оставаясь в основном. Действия могут быть заданы и без условия, при активации программы действия применяются сразу по умолчанию;
- Список задач в данной среде изучения программирования не ограничен обучающими примерами, задачи можно придумывать и реализовывать самостоятельно, что позволяет при помощи игры более глубоко разобрать способы организации действий и основы алгоритмов.

Программирование в данной среде происходит следующим образом (поэтапно):

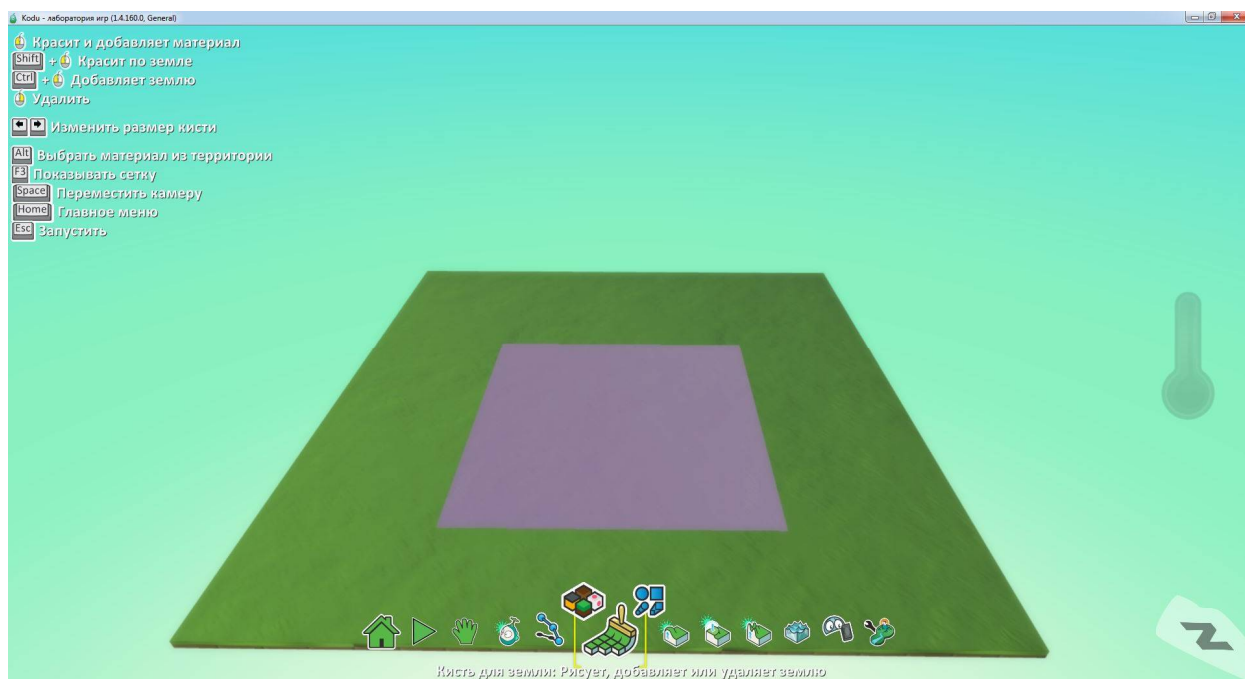
1. Выбирается объект или персонаж для редактирования;
2. Открывается меню действий (программирование объекта);
3. Согласно условию (графа When) задаются действия (графа Do);
4. Выход в основное меню мира;
5. Запуск мира.

Поскольку язык программирования представляет собой игру, то, как и в любой игре, ученик может создать своего персонажа и задать ему определенные характеристики (свойства и модель). Что примечательно, при длительном бездействии персонажи игры пытаются привлечь к себе

внимание – Пушка может перевернуться и улыбнуться, Коду либо оборачивается по сторонам, либо начинает качаться вперед-назад.

В языке программирования используется визуальный подход к написанию программы, т.е. все действия задаются набором картинок, а не написанием большого количества текста со сложным синтаксисом. И все, что остается – задать этот набор картинок для всех используемых в создаваемой игре персонажей и объектов.

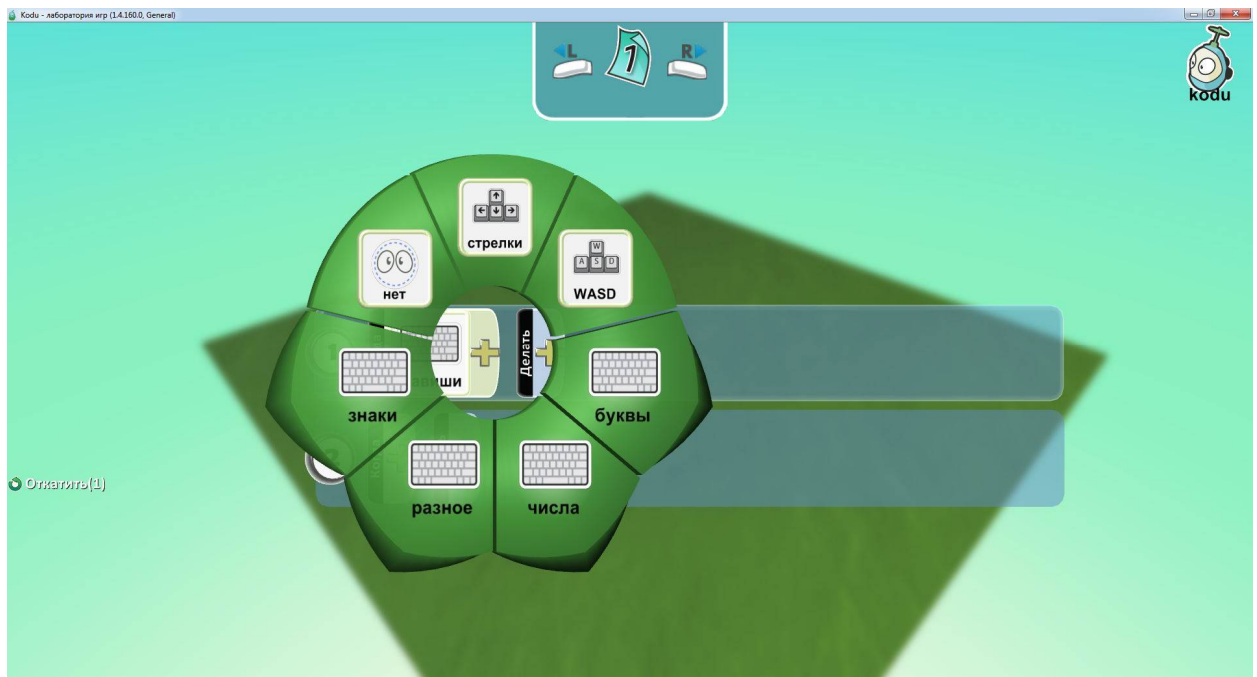
Создать собственный ландшафт в игре так же не сложное дело – все делается через отдельный пункт меню, в котором расположены только инструменты для изменения высоты, глубины, цвета ландшафта и его наполнения (поверхности или жидкости). В связи с этим персонажу можно задать определенные действия в зависимости от типа поверхности, по которой он идет или в которой находится.



Меню, через которое задается программа для действий, раскрывается лепестками, по выбору которого открывается дополнительно меню, в котором имеются подпункты с действиями или объектами и персонажами. При всем этом, для каждого объекта можно настроить фильтр, согласно которому некоторые объекты будут игнорироваться, а другие – наоборот,

восприниматься как целевые. К примеру, если задать фильтр, что персонаж будет собирать только красные яблоки (а для этого в действиях лишь добавляется цвет искомого объекта), то он будет игнорировать все яблоки, кроме лежащих на поле красных.

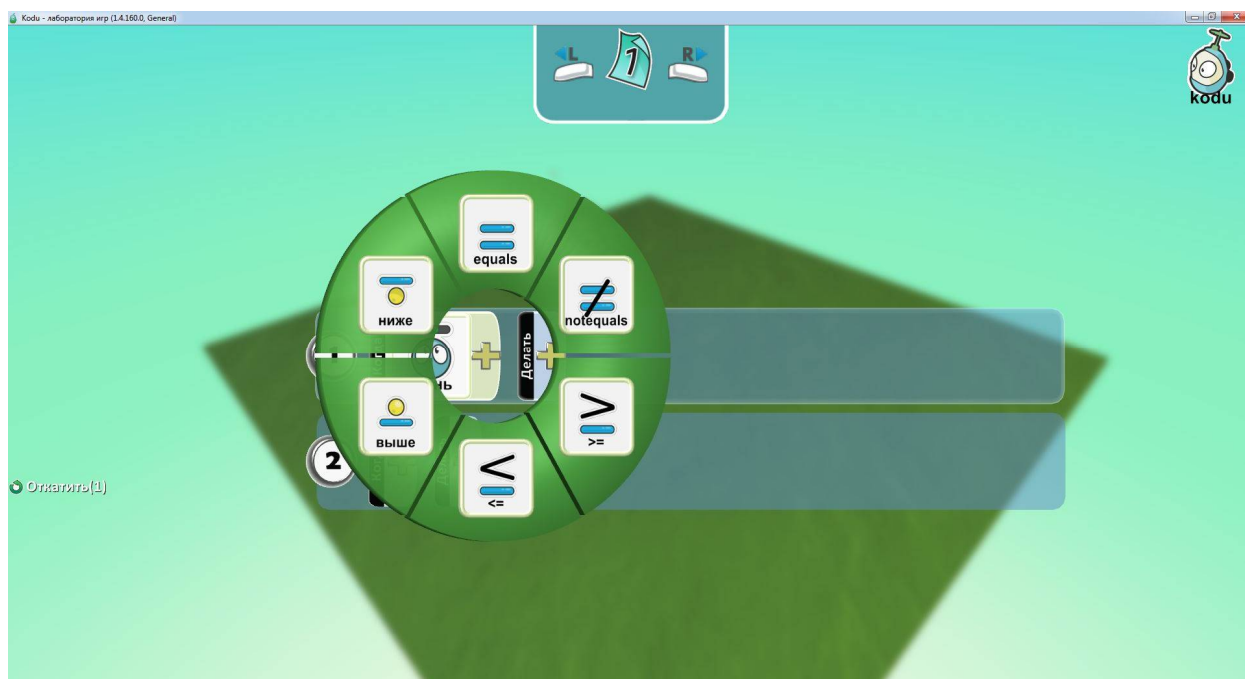
Таким образом, в визуальном языке программирования предоставлены гибкие настройки игрового мира без необходимости изучения синтаксиса языка и изучения методических материалов. Так же, результат можно увидеть в режиме реального времени, сразу произведя корректировку, и нет проблем с ошибкой при компиляции проекта, т.к. все действия заранее подбираются согласно введенным условиям и предыдущим действиям. Например, если выбрать ввод с клавиш, нам предоставляется меню, изображенное на картинке ниже.



Или же выберем пункт Игра -> Жизнь.



А вот и все имеющиеся для него параметры, после которых можно выбрать значение в подобном меню.



Глава 2. Методика использования визуального языка программирования Kodu на уроках информатики

2.1. Методические рекомендации по изучению языка программирования.

Согласно ФГОС, основная образовательная программа начального общего образования реализуется образовательной организацией через урочную и внеурочную деятельность. Внеурочная деятельность направлена на развитие личности и не учитывается при определении максимально допустимой недельной нагрузки обучающихся. Выбор формы организации внеурочной деятельности определяется образовательной организацией. Важная цель начального образования – создание прочного фундамента для развития, основанного на базовых знаниях и умениях, а так же развитии способности к совместной деятельности и рефлексии.

Изучение информатики начинается во 2 классе, т.к. школьники учатся видеть окружающую действительность с точки зрения информационного подхода. Начинается изучение устройства компьютера, освоение средств обработки и создания электронных документов. В 3 классе идет освоение представления и кодирования информации, а так же изучают возможность хранения информации на информационных носителях.

На данной ступени обучения информатики крайне важно дать младшим школьникам представление о логике и алгоритмах, что не всегда возможно в рамках конкретной образовательной программы, поэтому на основе факультативного курса, который будет являться внеурочной деятельностью, основан курс программирования в Визуальном языке программирования Kodu.

Для изучения данного языка программирования рекомендуется игровая форма проведения уроков, в связи с чем под данный язык программирования рекомендуется выделить факультативный курс по 2 академических часа в

неделю для учащихся от 1 класса (рекомендуется для начальных классов). Факультативный курс рассчитан на создание собственно мини-игры, которую, после завершения ее создания, необходимо предоставить к ознакомлению остальным учащимся. Длительность факультативного курса составит 2 месяца. Итого будет проведено 16 занятий, из которых 10 практических занятий (изучение программы, ее свойств и форм) и 5 – контрольных (отведены под создание собственной игры). Последнее, 16 занятие, отведено под проверку созданных игр самими учащимися. Согласно форме проведения курса созданы следующие методические рекомендации:

Форма проведения уроков – Игровая, чтобы привить учащимся интерес к программированию, привлечь их внимание к более углубленному изучению программирования и дать основные понятия о программировании, алгоритмах и событиях.

На момент практической работы (10 занятий) учащиеся совместно с учителем проводят рассмотрение многих предоставленных в языке программирования обучающих примеров, о которых будет сказано позднее.

На момент проведения контрольных занятий возможно как совместное написание игры (игровой проект на языке программирования Kodu), так и самостоятельный игровой проект.

Изучение языка программирования можно разделить по урокам, что поможет ученикам изучить язык программирования и более глубоко разобраться в синтаксисе.

2.2. Разработка системы уроков

Урок № 1. Основы задания ландшафта, создание и перекраска поверхности, удаление ненужной земли и использование кисти для корректной организации пространства (высота, глубина и заполняющие объекты).

План урока:

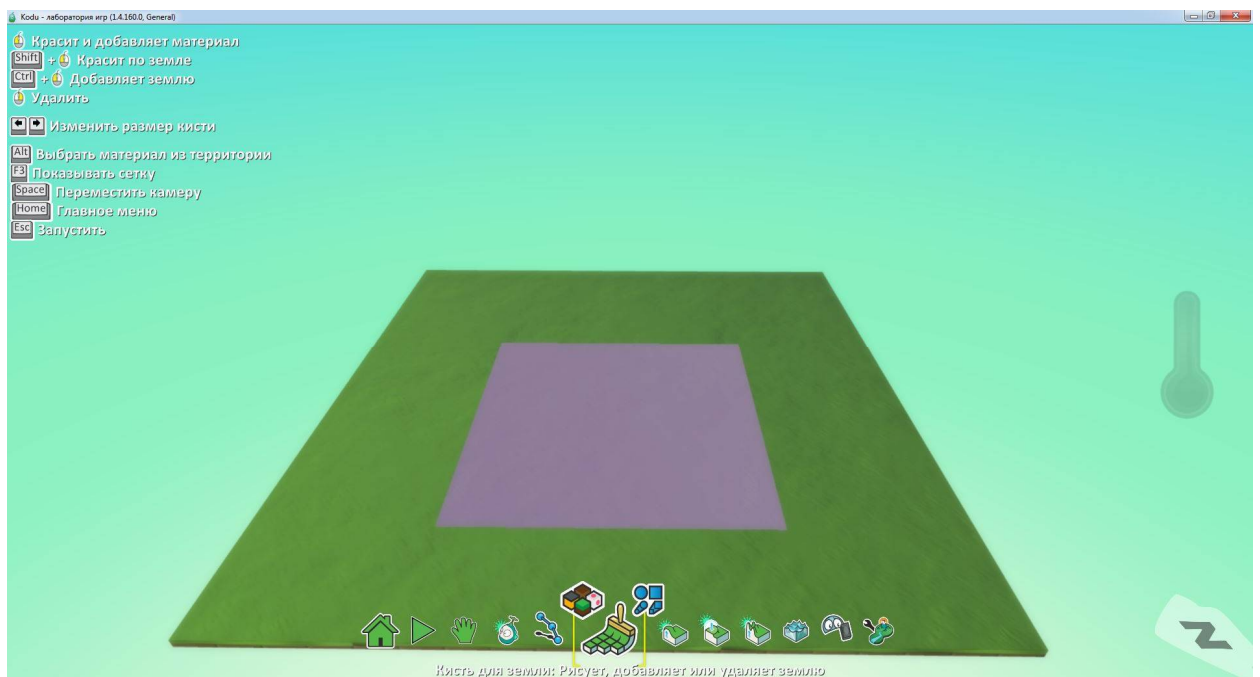
1. Теория занятия.

Для создания ландшафта в визуальном языке программирования Kodu используется отдельный инструмент, расположенный в панели программирования в основном меню мира. Первоначально, при создании собственного мира, по умолчанию создается небольшой участок поверхности, на которой можно разместить персонажей, предметы и объекты. Данный участок поверхности можно, и, даже, нужно изменить в зависимости от предполагаемого места действия в создаваемой игре. Это может быть лабиринт, это может быть обширное поле или скалистая местность, а может быть глубокое море или различные мосты через многие реки. Все это можно задать при помощи специализированного инструмента.

Для того, чтобы начать редактирование ландшафта, необходимо:

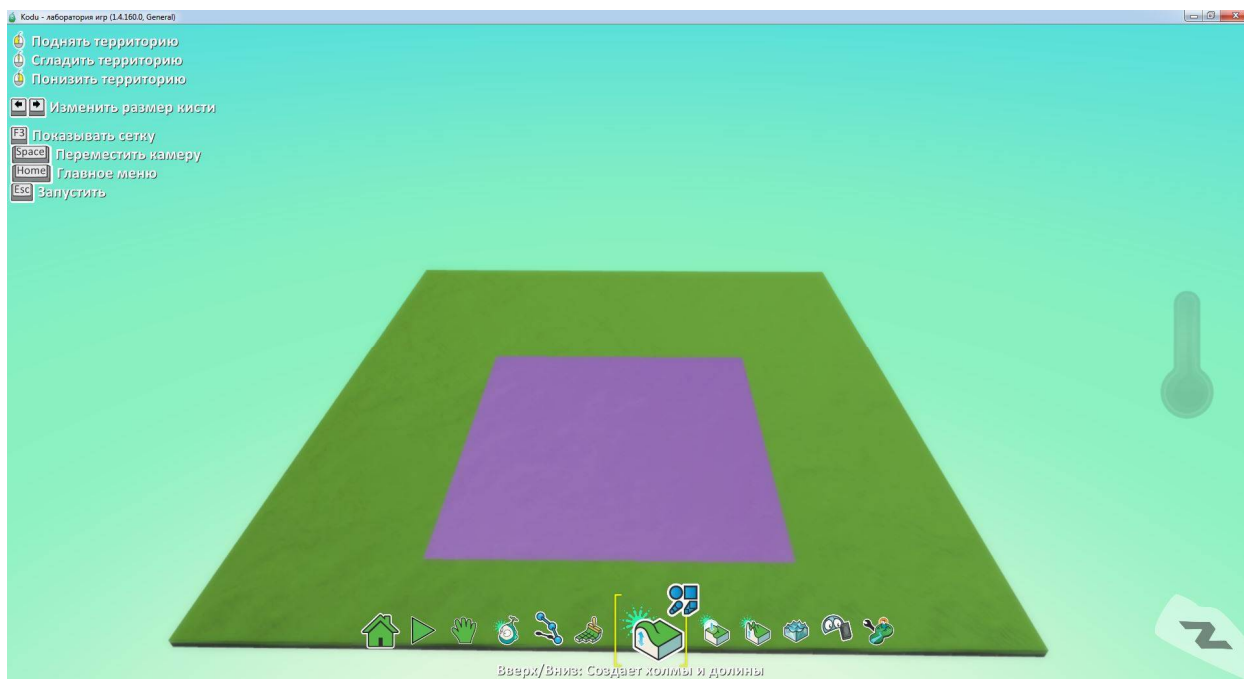
1. В игре нажать клавишу Esc, чтобы выйти в меню редактирования.

2. В меню выбрать инструмент Кисть для земли (рисунок ниже).



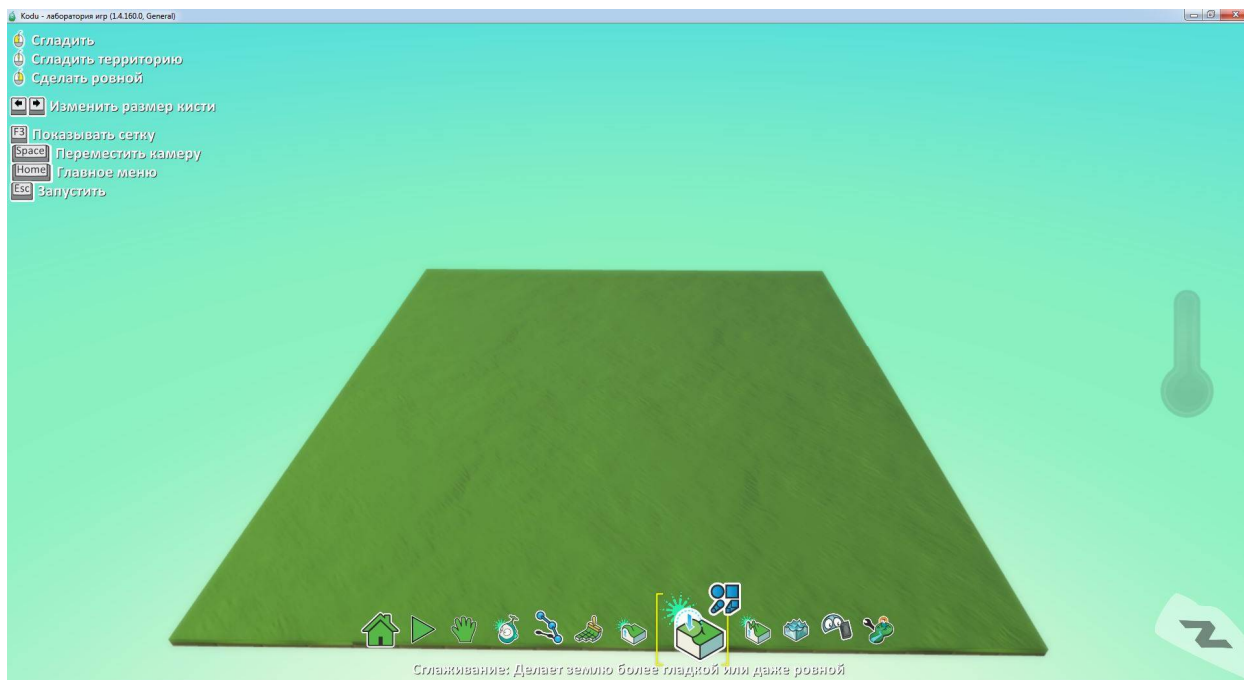
3. При помощи левой кнопки мыши добавить землю и покрасить ее в заданный цвет. Если зажать клавишу Shift, то будет происходить только перекрашивание уже существующей поверхности. Если зажать кнопку Ctrl, то будет добавляться новая земля с новой расцветкой, а старая поверхность меняться не будет.

4. После того, как мы сделали необходимый по площади участок земли, необходимо его отрегулировать по высоте и глубине. Для этого будем использовать инструмент Вверх/Вниз, который создает холмы и долины. По левой кнопке мыши мы будем добавлять высоту нашей поверхности, тем самым создавая возвышенности и горы, а по правой кнопке мыши мы можем делать углубления, в которых позднее сможем добавить воду и создать реку или озеро.

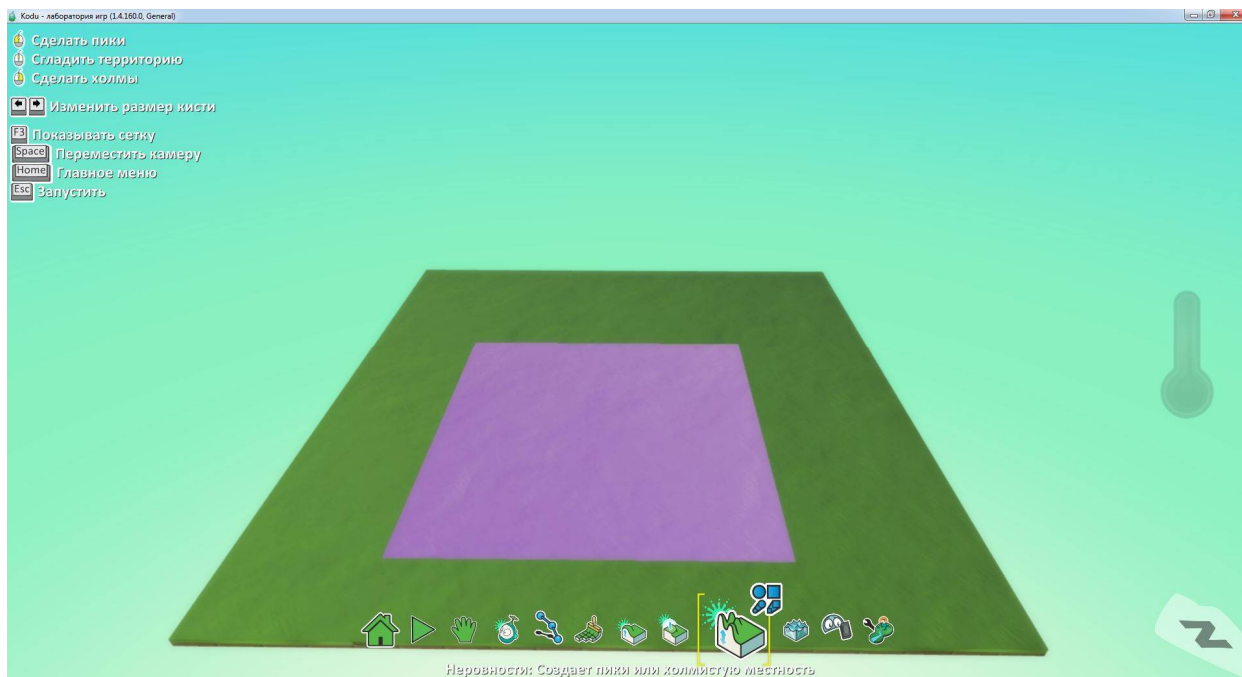


5. Далее мы будем использовать инструмент Сглаживание. Он позволяет выровнять наш ландшафт по отношению к близлежащим участкам и убрать резкие переходы между имеющимися возвышенностями и равнинами. По левой кнопке мыши происходит данное сглаживание. Правая же кнопка мыши выравнивает территорию

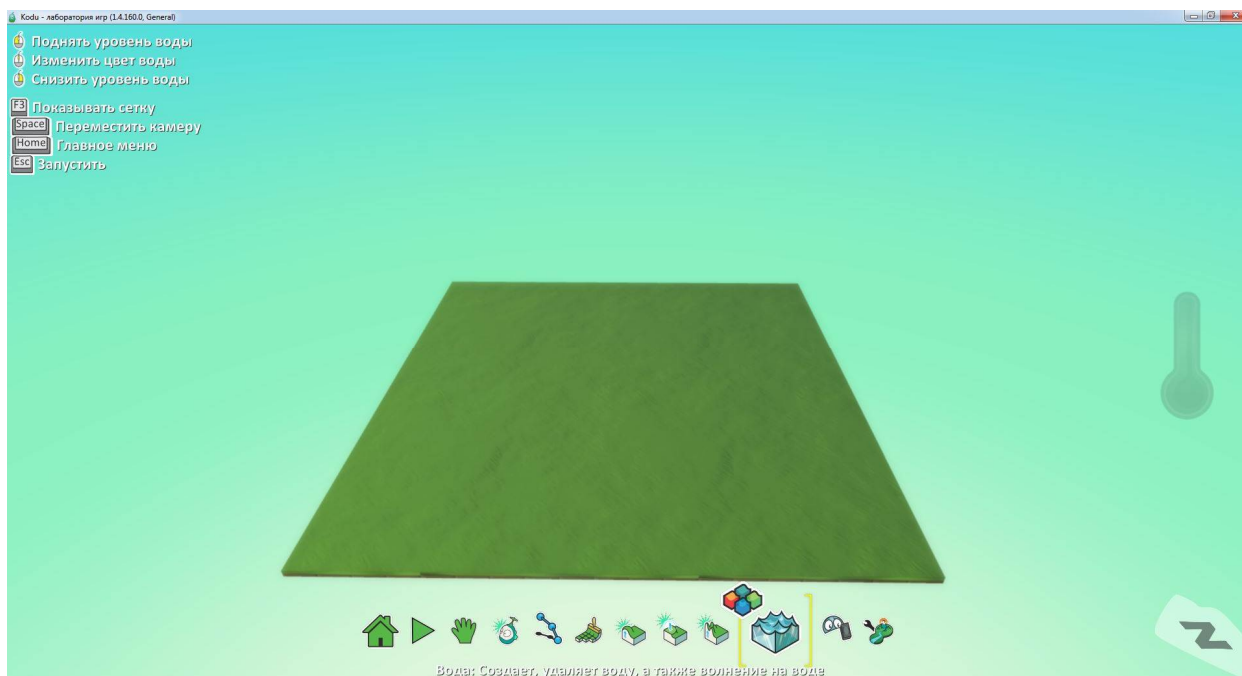
до среднего уровня близлежащих территорий, делая идеально ровную площадку.



6. Если мы будем делать в игре горы, то нам обязательно потребуется еще один инструмент – Неровности. В данном инструменте имеется возможность сделать неприступные горные пики (при использовании левой кнопки мыши), либо создавать возвышения, являющиеся основой горы либо какой-либо местности (которые создаются по нажатию на правую кнопку мыши).



7. Немало важно правильно добавить воду в наш мир, если она необходима. Для этого будем использовать инструмент Вода, который по нажатию на левую кнопку мыши поднимает уровень воды в мире, а по нажатию на правую кнопку мыши – понижает уровень воды.



Ну вот наш ландшафт и готов, можно переходить к дальнейшему построению мира.

2. Выполнение задания на практике.
3. Проверка выполнения задания.
4. Завершение работы в программе и небольшой опрос по проведенной работе:
 - Какой из примененных инструментов вам больше всего понравился в использовании?
 - Какой тип местности и ее заполнения вы бы предпочли, создавая свой собственный мир в игре?
 - Как вы думаете, есть ли возможность задать определенное поведение персонажа в зависимости от выбранного типа земли или воды?
5. Подведение итогов урока.

Урок № 2. Рассмотрение основных объектов и предметов, их характеристик и уникальных свойств.

На прошлом уроке мы рассмотрели способ создания земли, на которой будет происходить все, что мы решим использовать в нашей игре. Теперь нам нужно рассмотреть тех персонажей, которых мы сможем добавить в нашу игру.

Для данного урока нам потребуется инструмент Объект, в котором представлены всевозможные объекты и персонажи. Познакомимся со всеми поближе:



1. Kodu. Небольшой и неторопливый летающий кораблик с глазами. В основном, главным персонажем выбирают именно его.

2. Rover. Марсоход, имеющий несколько особых действий – сканирование, выстрел лучом, фотографирование и взятие пробы для анализа. В основном используется в футуристических мирах для симуляции изучения другой планеты

3. Яблоко. Самое обычное яблоко, которое можно перекрасить в любой из доступных цветов. Что примечательно, яблоко не может двигаться, но можно сделать яблоки-ловушки, которые по касанию будут съедать коснувшегося их персонажа.

4. Боты 1. В данном блоке представлено несколько классов персонажей, отличающихся своим внешним видом и описанием:



- Шар. Просто воздушный шар, к которому можно прикрепить камеру и заставить его двигаться за персонажем, или сделать невидимым.
- Байкер. Одноколесный мотоцикл с мордочкой вместо фар. Один из достаточно маневренных персонажей.
- Летающая рыба. Персонаж, находящийся в воде, один из немногих плавающих персонажей. Очень быстро плавает и поворачивает – идеальный герой для внезапных атак на воде.
- Самолет. Летающий персонаж, обычнодвигающийся параллельно заданной поверхности. Один из немногих персонажей, который может менять высоту своего полета.
- Осьминог. Водный персонаж, который так же может передвигаться и по суше. Медленно плавает, но может создавать чернильные пятна.
- Летающая тарелка. Самый быстрый летающий персонаж, который может мгновенно поменять направление

движения. Так же, как и самолет, имеет возможность менять высоту полета.

- Подводная лодка. Водный персонаж, который умеет всплывать и погружаться. Не такой шустрый, как летающая рыба, но со своими приятными особенностями.

- Пушка. Наземный персонаж, который в основном выбирают для стрельбы ракетами. И, очень даже, не зря его для этого выбирают – пушка очень подходит на такую роль, и не важно, что периодически он откидывается назад и улыбается.

- Аэростат. Медленно летающий персонаж, который, как и все летающие персонажи, может менять свою высоту полета.

- Плазма. Последний персонаж из данного блока объектов. Персонаж летающий, достаточно быстрый, и, как и остальные, может менять высоту полета. Единственная отличительная характеристика данного персонажа – за ним остается след полета, который постепенно исчезает.

5. Боты 2. Второй блок персонажей, представленных в Kodu Game Lab.



- Облако. Очень символический персонаж, который добавлен в игру в честь погоды в городе Редмонд, где располагается корпорация Microsoft (там всегда дождливо).
- Батут. Персонаж, который подбрасывает других персонажей при их приземлении на него.
- Корабль. Плавающий персонаж, который не может быть использован на суше.
- Свет. Летающий объект, по совместительству являющийся источником света, цвет которого можно изменять.
- Мина. Персонаж, имеющий шипы. В действиях данного персонажа присутствуют 2 дополнительных пункта – открыть и закрыть, по которым шипы выпускаются и убираются обратно соответственно.
- Шайба. Персонаж, летающий по округе без какого-либо трения и хорошо отскакивающий от препятствий. Идеален для быстрых игр.

- Тумба. Персонаж, на которого можно поставить другого персонажа, как на подставку. В основном используется для указания доступных для установки других персонажей мест в играх жанра Защита башни.

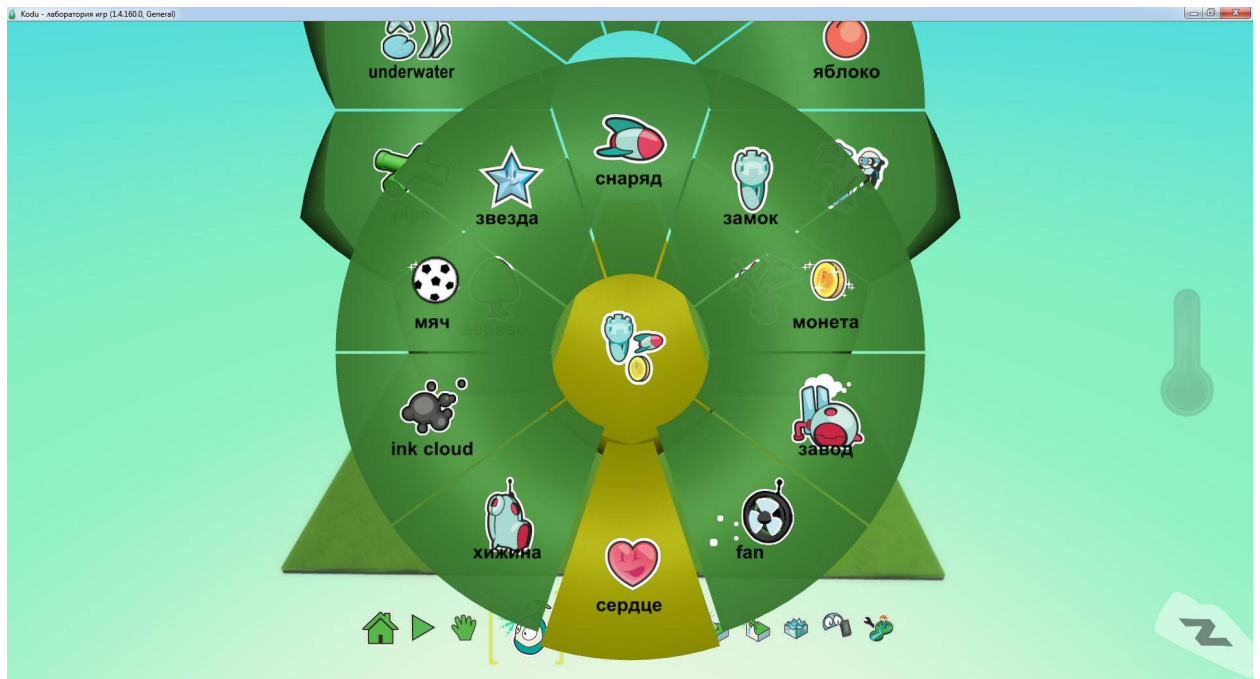
- Спутник. Очень схожий по характеристикам с летающей тарелкой персонаж, не уступает ей ни в чем. Так же может менять высоту полета.

- Задвижка. Разновидность пушки, которая по команде может закрываться и открываться, т.е. может прятаться в землю и высовываться из нее. Из минусов – заглушка не может передвигаться.

- Рыба. Схожа с летающей рыбой, но не такая быстрая, как она. На суше не пригодна к использованию, поскольку является плавающим персонажем.

- Черепаха. Как мина и заглушка, имеет команды «Заккрыть» и «Открыть», по которым может прятаться и выходить из панциря соответственно. Данная особенность может быть применена в некоторых играх, т.к. для закрытого состояния можно снизить урон, получаемый персонажем.

6. Объекты. Объекты могут быть использованы как декорации в имеющемся мире, а могут быть и условиями правильного прохождения игры.



- **Снаряд.** Объект, необходимый в том случае, если в игре задан счетчик патронов. При подборе данного предмета может восполнять количество боеприпасов.
- **Замок.** Замок может служить основой для стен здания, в котором будет происходить действие игры, а может быть красивым дополнением к красочному пейзажу, ведь выглядит замок как очень большая башня. Передвигаться замок, естественно, не может.
- **Монета.** Монета просто необходима в любой аркадной игре, т.к. может использоваться как внутриигровая валюта или начислять больше очков, чем сбор других предметов.
- **Завод.** Как и Замок, является зданием. Завод по своей сути является декоративным решением для игры, хотя можно использовать его как условие выполнения квеста.
- **Вентилятор.** Может отталкивать или притягивать других персонажей. Разные действия можно задать на специальные кнопки.

- Сердце. Отлично подходит, если в игре есть шкала здоровья, и необходимо указать, чем ее можно восполнить.

- Хижина. Еще одно здание, которое является частой декорацией игры. Но ее так же можно использовать как условие выполнения квеста либо как инициатора определенного события (условия для действия).

- Чернильное облако. Помимо создаваемых Осьминогом, чернильные облака можно задавать просто так, что будет своеобразным украшением игры.

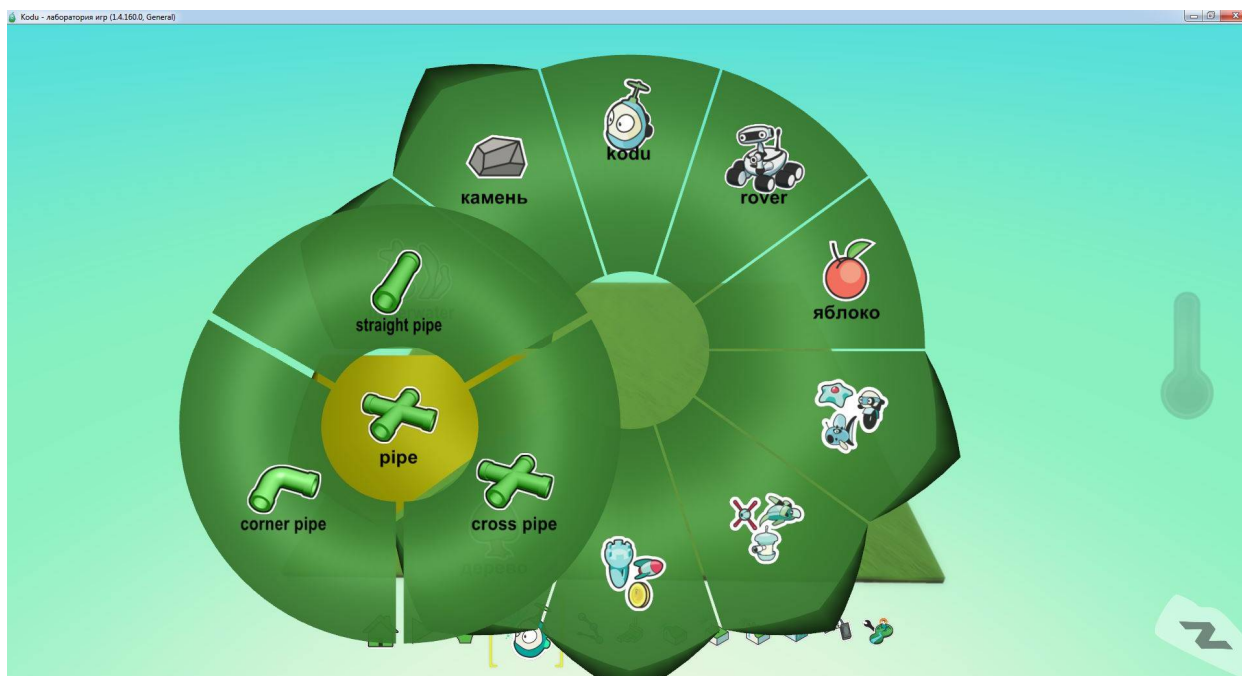
- Мяч. Как и Шайба, может использоваться для быстрых игр. Все таки, это же мяч, поэтому его можно пинать просто так, или запрограммировать и смотреть, что будет происходить при каких-либо действиях.

- Звезда. Как и все объекты, звезду можно запрограммировать. Но звезда может служить ключевым объектом для получения другого объекта, если это задать в игре.

7. Деревья. Здесь рассматриваются 4 типа деревьев, которые можно добавить в игру. Это как узкое и высокое, так и большое ветвистое дерево. Каждому найдется то дерево, которое будет ему по вкусу.

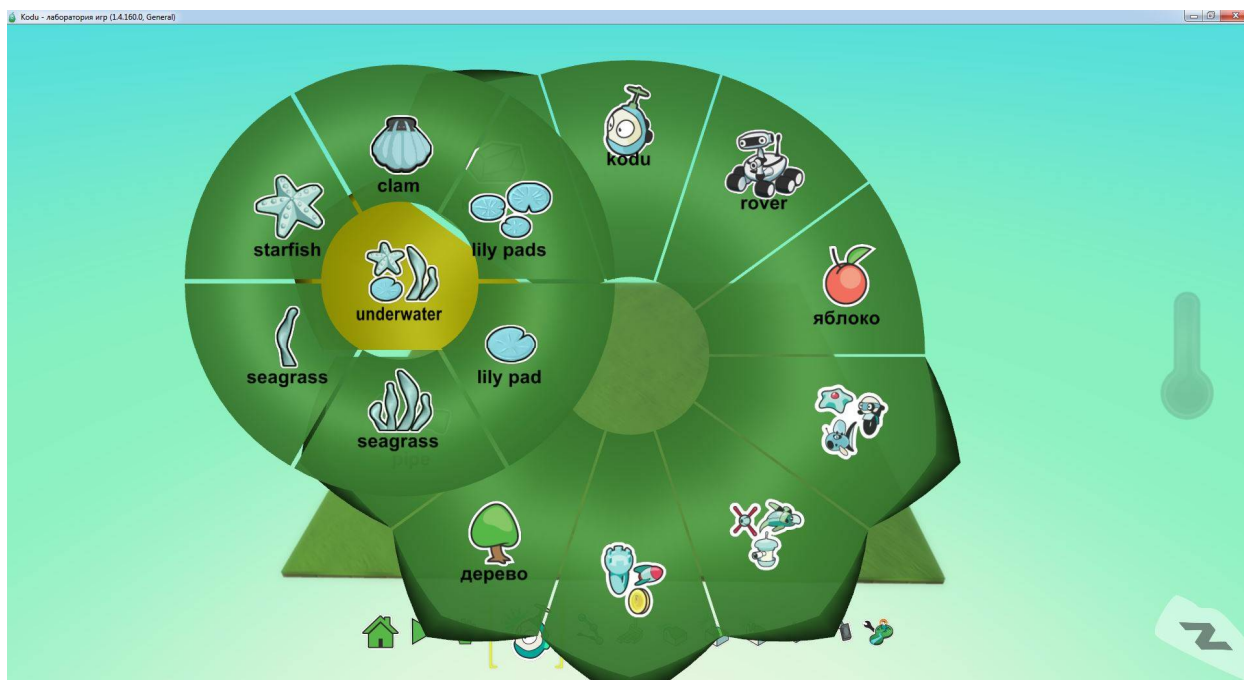


8. Трубы. Труб всего 3 – прямая, загнутая и перекрестная. Из труб можно составить конструкции, по которым могут проходить какие-либо объекты либо жидкости. Или просто огородить территорию.



9. Декорирование моря. Это различные объекты, которые можно использовать при создании красочного подводного мира. К примеру, здесь имеются Ракушка, Кувшинки (одна и несколько),

Морская звезда и Водоросли (в малом количестве и в большом количестве). Все объекты этой категории можно так же запрограммировать и перекрасить.



10. Камень. В данном пункте меню объектов представлены 6 разных камней, которые можно использовать при создании игры. Что примечательно, один из камней используется для создания ледников.



Теперь давайте добавим что-нибудь из этого в наш мир, который мы создали на прошлом уроке, и посмотрим, что с этим можно сделать.

Урок № 3. Основы движения. В данном уроке будет проводиться изучение 3-х способов управления персонажем – 2 автоматических (вижу искомый предмет – действую и использование маршрутов) и один ручной (ввод с клавиатуры/геймпада). На обучающих примерах будет рассмотрен оптимальный выбор для задания движения персонажа и всех неигровых персонажей (они же - NPC). Рассмотрение получат способы «Двигаться к нему», «Двигаться от него» и «Избежать».

Урок № 4. Корректировка свойств объектов и предметов. В данном уроке будет рассмотрена полная корректировка свойств персонажей.

Урок № 5. Создание основных алгоритмов игры. В данном уроке будут рассмотрены основные способы задания условий завершения игры, использование счетчика и взаимодействие с предметами.

Урок № 6. Основы боя. В данном уроке будут рассмотрены способы взаимодействия с другими объектами, рассмотрение характеристик урона и снарядов.

Урок № 7. Основные алгоритмы игры. В данном уроке будут рассмотрены основные алгоритмы для взаимодействия с предметами – поднять и перенести, подобрать и сложить в рюкзак, съесть по касанию или съесть по понижению здоровья. Так же будут рассмотрены способы расчета здоровья и задание определенных характеристик здоровья для разных объектов.

Урок № 8. Продвинутый алгоритм игры. В данном уроке будет рассмотрен способ и алгоритм задания персонажа противника, с определенным поведением и действиями. Будут рассмотрены 2 способа – «поиск и уничтожение» и «движение по заданной траектории с реакцией на игрока».

Урок № 9. Создание стен и отдельных возвышений. В данном уроке будет рассмотрено использование отдельных инструментов создания ландшафта для корректировки мира и задания лабиринтов или помещений с крытой частью, в которой можно сделать определенные секретные помещения.

Урок № 10. Обобщение полученных знаний. В данном уроке будет совместно с учителем создан небольшой игровой мир. В данном мире будут заданы 2 персонажа (игрок и NPC), на которых будут рассматриваться различные варианты взаимодействия персонажей. Конкретно, вариантов будет 2 - содействие и противоборство. При варианте «Содействие» будет рассмотрен алгоритм помощи игроку и движение за игроком. В варианте «Противоборство» будет рассмотрен вариант настройки реакции на игрока в зависимости от положения игрока.

Урок № 11-15. Создание собственной игры (одна на всех или самостоятельный проект).

Урок № 16. Проверка созданных(-ой) игр(-ы).

Заключение (по задачам)

После прохождения всех уроков младшие школьники смогут понять, как задавать условия для действия, и смогут в визуализированной форме понять базовые основы алгоритмизации. Помимо этого, визуализация действий с такими персонажами, которые представлены в данном языке программирования, привлекают внимание детей, что позволяет стимулировать произвольное внимание, и, тем самым, активизировать их деятельность на изучение языка программирования и информатики в целом.

Список литературы

<https://www.kodugamelab.com/> - основная информация о визуальном языке программирования Kodu, сайт разработчиков данного приложения.

http://www.pedsovet.pro/index.php?option=com_content&view=article&id=261:sobennosti-prepodavaniya-informatiki-i-ikt-v-nachalnoy-shkole&catid=62:informatics&Itemid=68

Информатика. 2 класс: учебник в 2 ч. Ч.1 / Н.В. Матвеева, Е.Н. Челак, Н.К. Конопатова и др.

Авторы: Матвеева Н. В., Челак Е. Н., Конопатова Н. К., Панкратова Л. П., Нурова Н. А.

[https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%BE_\(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F\)#.D0.9E.D0.B1.D1.89.D0.B8.D0.B5_.D1.81.D0.B2.D0.B5.D0.B4.D0.B5.D0.BD.D0.B8.D1.8F](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%BE_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)#.D0.9E.D0.B1.D1.89.D0.B8.D0.B5_.D1.81.D0.B2.D0.B5.D0.B4.D0.B5.D0.BD.D0.B8.D1.8F) – язык программирования Лого.

Аляев Ю.А., Гладков В.П., Козлов О.А. Практикум по алгоритмизации и программированию на языке Паскаль: учебное пособие. –+ Финансы и статистика, 2007. – 528 с.

Аляев Ю.А., Козлов О.А. Алгоритмизация и языки программирования: учебно-справочное пособие. – М.: Финансы и статистика, 2007. – 320 с.

Батрашина Г.С. Игра как метод изучения моделей в начальной школе // Информатика и образование.- 2008, №8. – С.5-8.

Батрашина Г.С. Формирование и развитие логико-алгоритмического мышления учащихся начальной школы // Информатика и образование.-2007. №9. –с.7-23.

Белова Г.В. Программирование в среде ЛОГО. Первые шаги. – М.: Солон, 2007

Белова Г.В. Программирование в среде ЛОГО. Первые шаги. – М.: Солон, 2007

Бин Нгуен. Объектно-ориентированное программирование на IBM Smalltalk. – М.: Диалог-МГУ, 1996.

Бокучава Т.П., Тур С.Н. Методическое пособие по информатике для учителей 2–4 классов. М.: ВНУ, 2007

Буйлова, Л. Н., Кленова, Н. В. Содержание дополнительного образования в школе. – Режим доступа:

http://www.ucheба.com/met_rus/k_dopobraz/dop_obrazovanie.htm

Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений – М.: Вильямс, 2008.

Великович Л., Цветкова М. Программирование для начинающих. – М.: Бином, 2007

Гейн А.Г. Информатика. 10 класс. Учебник. М.: Просвещение, 2008

Гейн А.Г., Сенокосов А.И., Юнерман Н.А. Информатика: Учебник для 10–11 классов общеобразовательных учреждений. – М.: Просвещение, 2005

Горячев А.В., Волкова Т.О., Горина т.и. Информатика в играх и задачах: Учебник-тетрадь для 2 класса четырехлетней начальной школы: В 2 томах. М.: Баласс, 2006

Еремин Е.А. Газета «Информатика». Среда Scratch – первое знакомство. – М.: Первое сентября, 2008 – №20 (573) – С. 17–24.

Еремин Е.А. Газета «Информатика». Среда Scratch – первое знакомство. – М.: Первое сентября, 2008 – №20 (573) – С. 16–28.

Ершов А.П. и др. Основы информатики и вычислительной техники. Учебник для 10–11-ых классов средних школ. – М.: Просвещение, 1985

Кнут Е. Дональд. Искусство программирования. Том 1. Основные алгоритмы. – М.: Вильямс, 2007

Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. 2-е издание, 2007

Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. 2-е издание, 2007

Программированное обучение и контроль в начальной школе [Электронный ресурс]. Режим доступа: <http://my-kuban.ru/?p=354>