

Кейс разработан: Прохоровой Екатериной Петровной-методист, педагог дополнительного образования ГБОУ

СОШ №14 СПДОД СЮТ г.о.Жигулевск

Предмет: Информационные технологии

Название кейса: Создание игры в среде визуально - объектного программирования KoduGameLab.

Формат обучения: очно-заочно; дистанционно.

Возраст обучающихся: 8-10 лет

Создание ландшафтов (миров), добавление объектов и персонажей с учетом игровой среды.

В нашем информационном обществе, IT-сфера становится одной из ведущих сторон жизни, поэтому необходимо с детства учиться сбору, обработке и хранению информации, а так же основам алгоритмизации и программирования.

Компьютерная игра — компьютерная программа, служащая для организации игрового процесса, связи с партнёрами по игре, или сама выступающая в качестве партнёра. Сейчас очень много компьютерных игр. Каждая по-своему интересна и увлекательна, но создание собственной игры или программы приносит большее удовольствие.

Microsoft Kodu Game Lab (визуальный конструктор), позволяет создавать трёхмерные игры без знания языка программирования. Разрабатывая игру изучаешь правила игры, разбираешься в меню программы и учишься создавать алгоритм действий для исполнителя. Когда ты видишь конечный продукт своих трудов, то приходит осознание своих возможностей, готовность к повышению своего образовательного уровня и продолжению обучения с использованием информационных технологий.

Тема:	Изучение среды визуально - объектного программирования KoduGameLab.
Количество часов/занятий:	6 часов
Проблемы, которые поставлены в кейсе:	Приобрести навыки алгоритмического мышления, т.е. научиться правильно мыслить и разумно планировать свои действия, создавая логические игры.
Цели и задачи:	<p><u>Цель:</u> научить работать с информацией, а также сформировать интерес к программированию через изучение среды визуально - объектного программирования KoduGameLab.</p> <p><u>Задачи:</u></p> <ul style="list-style-type: none"> • обучение поиску, отбору, организации и использования информации для решения стоящих задач и достижения поставленных целей; • изучение общих принципов построения алгоритмов, • ознакомление с основными алгоритмическими конструкциями; • формирование понятий об алгоритмических структурах — линейных, циклических, с ветвлением; • формирование навыков работы с объектами; • формирование первоначальных навыков программирования с применением учебных визуальных сред программирования.

	<ul style="list-style-type: none">• формирование навыков работы с различными исполнителями;• разработка алгоритмов для различных исполнителей;• пропедевтика базовых понятий программирования и получение первоначального практического опыта.
Предполагаемые результаты:	<p><u>Предметные:</u></p> <p><i>Учащийся должен понимать:</i></p> <ul style="list-style-type: none">• основные принципы• разработки и анализа алгоритмов• правила записи и выполнения алгоритмов,• понимать назначение среды KoduGameLab;• понимать назначение программы;• знать правила оформления программы;• создавать и редактировать программы; <p><i>Учащийся должен уметь:</i></p> <ul style="list-style-type: none">• разрабатывать структурные схемы различных алгоритмов,• организовывать в зависимости от требований задачи необходимые структуры• данных,• управлять движением объектов;

	<ul style="list-style-type: none"> • моделировать движение с разными скоростями; • моделировать движение с повторяющимися фрагментами; • разрабатывать программы; <p><u>Метапредметные:</u></p> <ul style="list-style-type: none"> • результатами изучения данного кейса является формирование универсальных учебных действий. • умение ставить и формулировать для себя новые задачи; развивать мотивы и интересы своей познавательной деятельности; • умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения задач. • уметь составлять алгоритмы для решения поставленных задач; • уметь разбивать задачу на подзадачи; • составлять алгоритмы, организующие данные в том виде, который необходим для решения задачи.
--	---

<p>Ресурсы и материалы:</p>	<p><u>Для наставника:</u> <i>Оборудование:</i> Ноутбук или стационарный компьютер, веб камера, микрофон. Постоянный доступ в интернет.</p>
-----------------------------	--

Материал и ресурсы:

- Визуальное программирование в KODU: первый шаг к ИТ-образованию –Самара, 2013
- Kodu Game Lab. *Электронный ресурс+ – Режим доступа: <http://gcup.ru/load/kodu/2-1-0-1504>
- Сайт «Кубок kodu» / <http://koducup2012.cloudapp.net/>
- Курс по Kodu Game Lab / <http://www.teachvideo.ru/course/427/>
- Ссылка на [сайт международного сообщества Kodu](#)
- Ссылка на бесплатную [программу для записи действий на экране монитора](#), с помощью которой учащиеся могут легко сделать ролик-презентацию своей игры и поделиться ей в группе проекта [ВКонтакте](#).

Для ребенка:

Оборудование:

Ноутбук или стационарный компьютер, веб камера, микрофон. Постоянный доступ в интернет.

Материал и ресурсы:

- ТБ при работе с компьютером <https://www.pc-school.ru/texnika-bezopasnosti-i-pravila-raboty-s-kompyuterom/>
- Kodu Game Lab. *Электронный ресурс+ – Режим доступа: <http://gcup.ru/load/kodu/2-1-0-1504>
- [Скачать среду Kodu](#)
- [Инструкция по скачиванию и установке среды Kodu на компьютер](#)

	<ul style="list-style-type: none">• Видео инструкция по скачиванию и установке среды Kodu на компьютер• Ссылка на подборку видео уроков курса (для самостоятельного знакомства учащихся со средой)• Ссылка на бесплатную программу для записи действий на экране монитора, с помощью которой учащиеся могут легко сделать ролик-презентацию своей игры и поделиться ей в группе проекта ВКонтакте.• Ссылка на сайт международного сообщества Kodu
Ход работы:	<ol style="list-style-type: none">1. Первоначальное знакомство обучающихся с определенными понятиями (характеристиками, свойствами объектов и т.п.) или видами деятельности через выполнение конкретных практических заданий;2. Знакомство с Kodu: установка, интерфейс, настройки, принципы и правила, создание ландшафта, объектов, первая игра.3. Новые возможности перемещения и связи объектов. Опции "путь", "родитель"4. Дополнительные опции в игре: подсчет баллов и здоровья, таймер.5. Сложное поведение объектов: опция «страницы».6. Разработка полноценной игры от «А» до «Я».

Карта кейса

№	Тема занятия	Кол-во часов	Сценарий проведения				Умения
			Лекция	Материалы	Практика	Материалы	
1.	Программирование в среде KoduGameLab. Знакомство с интерфейсом и основными принципами работы с программой KoduGameLab	1	Kodu Game Lab – визуальная среда разработки трехмерных игр. Знакомство с визуальной средой Kodu. Возможности Kodu. Главное меню Kodu. Знакомство с интерфейсом программы Kodu, возможностями программы Kodu, главным меню программы и режимом просмотра проекта.	Kodu Game Lab. Электронный ресурс Режим доступа: http://gcup.ru/load/kodu/2-1-0-1504 Ссылка на сайт международного сообщества Kodu	Скачивание и установка среды Kodu на компьютер.	Ссылки Скачать среду Kodu Инструкция по скачиванию и установке среды Kodu на компьютер Видео инструкция по скачиванию и установке среды Kodu на компьютер	Понимание назначения среды KoduGameLab; и назначение программы. Научиться самостоятельно скачивать и устанавливать ПО.

2.	Создание ландшафтов (миров), добавление объектов. Создание персонажей с учетом игровой среды, действия персонажей Перемещение персонажей в макросреде Kodu с использованием клавиатуры и мыши.	2	Исследование миров. Просмотр миров. Просмотр кода. Параметры команд. Объекты Kodu. Импорт и редактирование объектов. «Миры в Kodu».	Ссылка на подборку видео уроков курса (для самостоятельного знакомства учащихся со средой) Сервис с играми: http://www.planetkodu.com	Создание нового мира. Создание ландшафта. Создание ландшафта с помощью «Кисти земли». «Работа с ландшафтом». Техника рисования. Цветокоррекция. Управление объектами. Взаимодействие объектов.	Использование среды Kodu Game Lab. Ссылка на подборку видео уроков курса (для самостоятельного знакомства учащихся со средой) Сервис с играми: http://www.planetkodu.com	Научиться создавать собственный мир и создавать различные персонажи, управлять движением объектов;
3.	Режим программирования, основные операторы Kodu	2			Разработка кода программы. Рассмотрение основных ошибок при написании кода программы. Написание кода программы. Создание клонов и порождаемых объектов. Программирование страниц. Программирование таймера игры. Подсчет очков. Понятие таймера. Запуск таймера.	Использование среды Kodu Game Lab. Ссылка на подборку видео уроков курса (для самостоятельного знакомства учащихся со средой) Сервис с играми: http://www.planetkodu.com	Научиться создавать и редактировать программы, моделировать движение с разными скоростями; моделировать движение с повторяющимися фрагментами. Освоить работу с таймером.

4.	Создание нового игрового мира	1					Получение личного творческого потенциала, осознание своих возможностей путем создания 3D игры. Умение логически анализировать и создавать различные пути решать проблем.
----	-------------------------------	---	--	--	--	--	--

Рабочий бланк учащегося

ФИО учащегося _____

Знакомство с визуальной средой программирования Kodu: делаем первые шаги

Для справки

Визуальное программирование — способ создания программы для компьютера путём манипулирования графическими объектами вместо написания её текста (кода).

Внимание!

Для начала работы скачайте ([Скачать среду Kodu](#)) и установите визуальную (от лат. visualis - зрительный) среду программирования Kodu:. Для установки воспользуйтесь инструкцией - [Видео инструкция по скачиванию и установке среды Kodu на компьютер](#).

Для начала посмотрите ролик “Первая игра (обучение)” (ссылка [на подборку видео](#))

Проверь себя!

После просмотра видео ответьте на вопросы:

Каким образом осуществляется переключение между режимом редактирования и режимом игры?

Как можно понять, что включен режим редактирования?

Каким образом отображаются в режиме редактирования пути, по которым перемещаются объекты?

Какие действия необходимо выполнить для создания программы, определяющей последовательность действий объекта?

Для справки:

Программа в **Kodu** - это набор правил, которые определяют действия объекта.

Для написания правил в Kodu используются два оператора:

When <условие> Do <действие>

When (англ. “когда”, “если”, “в то время как”) - оператор, определяющий условие;

Do (англ. “делать”) - оператор, определяющий непосредственное действие, которое должен выполнить объект при соответствующем условии.

Программа создается для каждого объекта индивидуально!

Упражнение 1. Создание первой программы.

Сюжет игры: Байкер объезжает деревья.

Для успешного выполнения упражнения и создания первой игры, четко следуйте предложенному алгоритму:

- Запустите программу Kodu.
- Выберите команду «Новый пустой мир» (Empty World). Появится зеленое поле - основа для размещения игровых объектов в мире. Внизу окна размещены иконки, отображающие основные команды программы (рис.1.1).

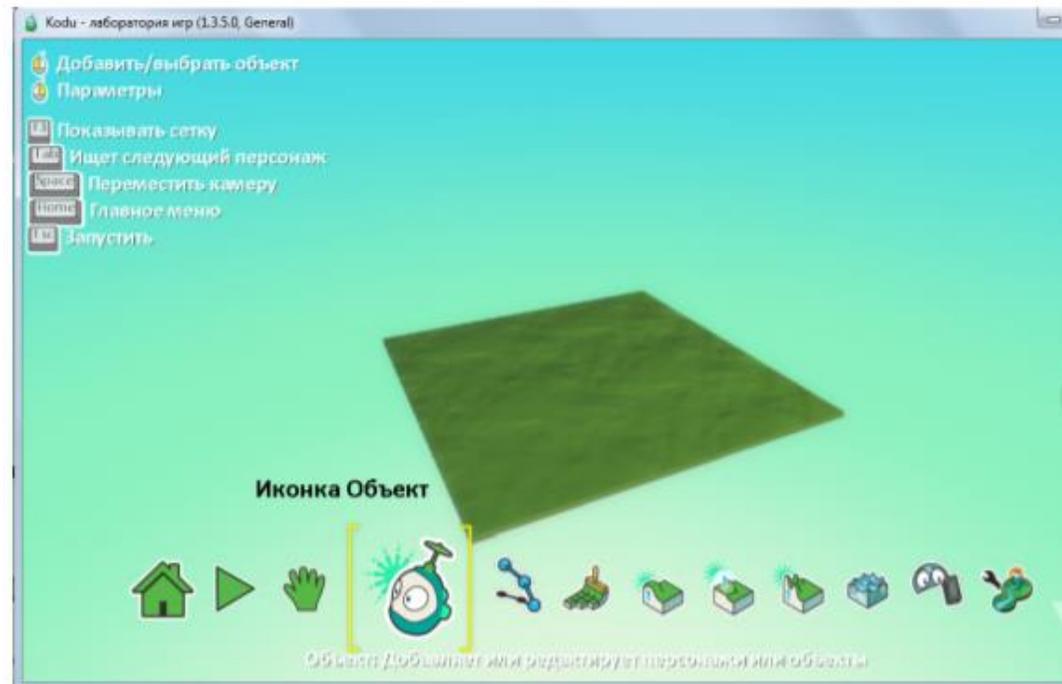


Рис. 1.1. Режим создания мира в Kodu

- Добавьте на поле 3 дерева. Для добавления объектов выберите иконку «Объект» и щелкните один раз по полю левой кнопкой мыши. Появится список доступных объектов. Выберите объект «Дерево». Повторите действия ещё два раза.



Рис. 1.2. Режим выбора объектов

- Добавьте объект «Байкер» и задайте для него программу действий - набор правил. Щелкнув правой кнопкой мыши на объекте «Байкер», вызовите контекстное меню и перейдите в режим создания Программы.
- В открывшемся окне кода составьте инструкцию для движения вокруг деревьев. Команды задаются выбором из списка необходимых инструкций. Пример программы приведен на рис. 1.3. При щелчке по карточке (иконка со знаком +) откроется перечень доступных действий, из которых следует выбрать необходимое (подтвердив действия нажатием левой клавиши мыши).



Рис. 1.3. Программа, позволяющая Байкеру объезжать деревья

- Запустите программу на выполнение клавишей Esc. Понаблюдайте за движением Байкера. Если траектория движения не соответствует поставленной задаче (объект не движется или не объезжает деревья), то проверьте корректность кода (рис. 1.3).
- Сохраните программу на жестком диске. Для этого перейдите в главное меню (клавиша Home) и выберите команду «Сохранить мой мир».

Задание для самостоятельной работы:

1. Попробуйте изменить цвет одного из деревьев, например, на желтый.
2. Напишите программу, в которой предусмотрите ситуацию столкновения Байкера с деревом.

Упражнение 2. Создание ландшафтов.

В этом упражнении вы научитесь проектировать вид местности (ее ландшафт), на которой происходят действия игры.

Для справки

Ландшафт (нем. Landschaft - вид местности, от Land - земля и schaft - суффикс, выражающий взаимосвязь, взаимозависимость).

Рельеф (фр. relief, от лат. relevo — поднимаю) — совокупность неровностей суши, дна океанов и морей, разнообразных по очертаниям, размерам, происхождению, возрасту и истории развития.

Для начала посмотрите ролик “Инструкции по созданию ландшафта” (ссылка [на подборку видео, http://www.planetkodu.com](http://www.planetkodu.com))

Проверьте себя! После просмотра видео, экспериментируя с местностью в среде Kodu, ответьте на вопросы:

1. Как выбрать тип ландшафта?

2. Как удалить часть ландшафта?

3. Как изменить тип кисти для создания территории? _____

4. А как изменяется тип материала, с помощью которого проектируется ландшафт?

5. Как добавить другой вид ландшафта к уже имеющейся территории?

6. А как изменить часть уже существующей местности?

7. Какой инструмент служит для создания рельефа?

8. Что умеет “волшебная” кисть?

9. Какие функции выполняют клавиши Ctrl, Shift и пробел при редактировании территории?

10. Каким способом можно дублировать тип местности при создании ландшафта?

Для успешного выполнения упражнения и создания игры, четко следуйте предложенному алгоритму:

- Создайте новый пустой мир (Empty World).
- Создайте ландшафт игрового мира в виде зеленой травы, небольших гор и холмов. Для создания зеленого поля выберите инструмент «Кисть» для земли и произвольно нарисуйте на игровом поле землю. Цвет земли выберите под номерами 15 или 54.

- Поднимите отдельные участки земли, сделав из них холмы и горы. Для создания неровностей используйте инструмент «Неровности».
- Добавьте воду в созданный ландшафт при помощи инструмента «Вода», цвет воды под номером 8.
- Измените цвет неба при помощи инструмента «Параметры мира». Включите волны для воды.
- Нарисуйте облака на небе.
- Сохраните мир под именем «Ландшафт» и запустите игру на выполнение. Обратите внимание на движение воды и набегающие волны.

Задание для самостоятельной работы:

- Создайте в Kodu модель местности, используя кисть и ее различные типы, а также увеличение и уменьшение размера кисти.
- Добавьте на созданном рельефе холмы и впадины, используя соответствующий инструмент.
- Постройте стены на моделируемой местности при помощи волшебной кисти.
- Добавьте несколько дорожек на создаваемой территории.
- Измените модель местности, добавив водоемы, парки, здания.

Упражнение 3. Создание клонов объектов.

Сюжет игры: три Аэростата стреляют по Самолету. (ссылка [на подборку видео](http://www.planetkodu.com), <http://www.planetkodu.com>)

Для справки

Аэростат (воздушный шар) — летательный аппарат легче воздуха, использующий для полета подъемную силу заключенного в оболочке газа (или нагретого воздуха) с плотностью меньшей, чем плотность окружающего воздуха (согласно закону Архимеда).

Для успешного выполнения упражнения и создания игры, четко следуйте предложенному алгоритму:

- Создайте новый мир (Empty World).
- Разместите в нем объекты Самолет и Аэростат.
- Напишите для Аэростата код, выполняющий следующие действия: когда Аэростат замечает вдали Самолет, звучит громкая музыка. Приближаясь к Самолёту, Аэростат начинает светиться.
- Составьте программу, позволяющую управлять Аэростатом с помощью клавиатуры. Пример программы приведен на рис.2.6, строки 1 - 3.
- Сохраните мир под именем «Самолет».
- Запустите программу на выполнение клавишей Esc. Подведите Аэростат к Самолету, послушайте музыку, которая была выбрана для игры.

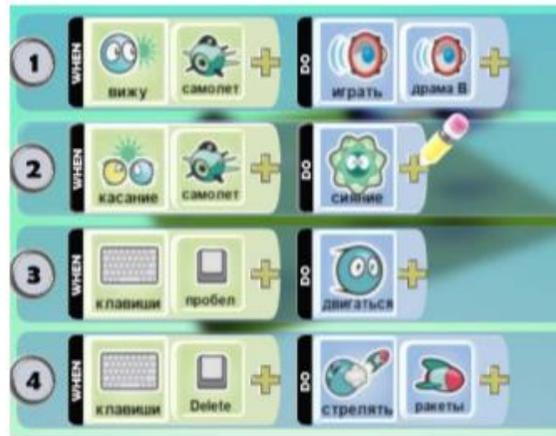


Рис. 2.6. Код, управляющий работой Аэростата

- Создайте эскадрилью Аэростатов из трех объектов. Для этого щелкните на уже имеющемся Аэростате правой кнопкой мыши и выберите пункт «Копировать», затем поместите указатель мыши туда, где надо создать новый Аэростат, щелкните правой кнопкой мыши и выберите пункт «Вставить (Аэростат)». Повторите вставку так, чтобы на поле оказалось три Аэростата.
- Запустите игру и убедитесь, что все Аэростаты управляются и двигаются одновременно.
- Добавьте код, позволяющий каждому Аэростату выстреливать Импульсы, рис.2.6, строка 4.

Сколько раз нужно изменить код? _____

- Сохраните игру, запустите ее и убедитесь в работоспособности.

Упражнение 4. Разработка своей оригинальной игры от “А” до “Я”

(ссылка [на подборку видео, http://www.planetkodu.com](http://www.planetkodu.com))

Для справки:

Проект (от лат. projectus — брошенный вперед, выступающий, выдающийся вперед) — это работа, планы, мероприятия и другие задачи, направленные на создание нового продукта.

В нашем случае проектом будет созданная игра. Какая игра? Это решать вам! Но постарайтесь, чтобы она была интересна не только вам, но и вашим друзьям, родителям, близким, которые будут в нее играть.

Итак, изучив все возможности Kodu, приступаем к проектированию своей игры от “А” до “Я”!

Шаг 1. Для начала выберите жанр игры. Что это будет за игра?

Результат этого шага - разработка ключевой идеи игры. Вы должны в общих чертах представлять, где будут происходить игровые действия, какими объектами будет управлять играющий и с какой целью.

Шаг 2. Проектируем сюжет игры.

Проектирование сюжета, естественно, потребует:

- придумать и создать эскиз на бумаге как будет выглядеть виртуальный игровой мир (ландшафта и его объектов: вода, дороги, деревья, горы и т.д.), в котором будут развиваться события;
- придумать героев (персонажей, объекты), управление которыми происходит в процессе игры (развития сюжета) или которые будут действовать самостоятельно. Заметим, что сюжет игры, во многом будет определяться возможностями тех объектов, которыми будет управлять играющий. В любой игре используют такие элементы как:
- Основной персонаж/персонажи (главные герои).

- Основной противник или препятствия, которые противодействуют достижению цели игры (победе). Например, препятствием может стать условие или ограничение, например, по времени.
- Второстепенные персонажи.

Сюжет игры предполагает описание последовательности действий, которые происходят в процессе игры с главным героем/персонажем. Обязательно сначала сформулируйте и запишите идею сюжета вашей игры, подберите соответствующее название. Для того, чтобы сюжет игры был реализуемым, надо хорошо изучить возможности управляемых объектов и персонажей. Постарайтесь ответить на вопросы:

- Какую миссию будут выполнять главные герои (персонажи, объекты)?
-

- Будут ли персонажи и объекты, препятствующие достижению цели?
-

- Что каждый из героев умеет/будет делать (роль каждого из них в сюжете)?
-

- Каковы свойства и характеристики выбираемых объектов и как они будут использоваться в процессе игры?
-

- При каких условиях и в каких обстоятельствах будут использоваться те или иные свойства (возможности) персонажей (объектов)?
-

- Как будет осуществляться управление персонажами (объектами)?
-

Итак, на этом шаге вы должны:

- придумать ландшафт и выбрать управляемые объекты;
- изучить/освежить в памяти возможности управляемых персонажей (объектов);
- выбрать варианты поведения персонажей согласно сюжету.

Шаг 3. Детализируем цель игры.

Результат этого шага - определить цель проектируемой вами игры и миссию игровых персонажей (героев, объектов) в процессе достижения этой цели.

Шаг 4. Создаём раскадровку и описываем правила игры.

Вам потребуется описать правила для каждого объекта и персонажа, который встретится нам по ходу сюжета. Здесь также можно предусмотреть реплики персонажей и другие свойства и варианты поведения: сияние, исчезание, движение, звуковой сигнал, выражение эмоций, захват предметов и т.д.

Шаг 5. Создаем игровой мир, ландшафт, атмосферу согласно нашей раскадровке и сюжету.

Такие вещи как цвет неба, текстура ландшафта, наличие или отсутствие воды влияют на восприятие игроком сюжета и его отношение к игре. Вид игрового мира должен соответствовать сюжету и жанру игры.

Шаг 6. Добавляем персонажей. Программируем действия героев (персонажей, объектов) согласно разработанным правилам и сюжету игры.

Шаг 7. Проводим тестирование игры.

Шаг 8. Презентация игры.

Свое мнение о пройденном
