



Python. Модуль 3. Урок 1.

Методические указания

Тема: Знакомство с pygame. Понятие "спрайт". Бесконечный цикл.

Цели урока:

- Обеспечить закрепление учащимися понятий: переменная, функция, цикл, создание объекта.
- Обеспечить усвоение учениками понятий: класс, бесконечный цикл, спрайт.
- Отработать навыки:
 - набора текста и его форматирования;
 - работы с консольным приложением в среде разработки;
 - решение задач с использованием функции `init()`;
 - решение задач с использованием метода `load()`;
 - решение задач с использованием метода `get()`;
 - решение задач с использованием метода `update()`;
 - решение задач с использованием бесконечного цикла.
- Создать условия для развития:
 - логического и алгоритмического мышления;
 - элементов творческой деятельности;
 - памяти;
 - критического мышления, культуры групповой самоорганизации;
 - коммуникативной культуры.
- Организовать самостоятельную работу учащихся, направленную на развитие алгоритмического мышления, познавательных интересов и навыков работы на компьютере.

Задачи урока:

- повторить и закрепить понятия, пройденные на предыдущих уроках;
- изучить правила создания класса;
- написать программу с использованием класса;
- изучить функцию `init()`;
- написать программу с использованием функции `init()`;
- изучить объект `self`;
- написать программу с использованием `self`;



- изучить методы `load()`, `get()`, `update()`;
- написать программу с использованием методов `load()`, `get()`, `update()`.

Перед началом урока:

1. Открыть страницу с логинами и паролями для детей.
2. Открыть и запустить на проекторе презентацию (на слайдах присутствует gif анимация. Для её отображения запустить презентацию).
3. Запустить тестовую программу `m3u1_test.py`.
4. Проверить наличие раздатки.

План урока:

1. Анонс занятия (5 мин.).
2. Повторение (10 мин.).
3. Знакомство с модулем `pygame` (5 мин.).
4. Создание игрового окна (20 мин.).
5. Создание класса для спрайта (15 мин.).
6. Добавление спрайта в игровое окно (20 мин.).
7. Рефлексия (5 мин.).
8. Выдача задания на дом (5 мин.)

1. Анонс занятия (5 мин.).

Слайд 2. Соберите детей в свободной от компьютеров зоне и обсудите, с какими понятиями им сегодня предстоит работать. С какими понятиями они уже знакомы, а что будет для них новым.

2. Повторение (10 мин.).

Слайд 3. Вспомните с детьми, что они изучали на прошлых занятиях, с какими понятиями познакомились. Получите от детей развернутые ответы, чтобы понять, на что обратить внимание в повторении, а какие понятия дети хорошо закрепили. Ответы на эти вопросы помогут им для усвоения материала по новому модулю.

Слайды 4-6. Вспомогательные слайды для повторения прошлого материала.



3. Знакомство с модулем pygame (5 мин.).

Слайд 7. Pygame – модуль для языка Python, созданный для разработки 2D игр. Обсудите с детьми, какие самые популярные 2D игры они уже встречали? Змейка, пинг-понг и т.п.

Слайды 8. Скрыт из основного показа презентации, содержит информацию по установке pygame. Информация и ссылка на инструкцию продублирована у детей на платформе. В конце занятия не забудьте озвучить эту информацию.

Слайд 9. Так как игры разрабатываются с использованием большого количества модулей, важно не запутаться, какой метод из какого модуля, поэтому мы будем использовать `import pygame` и для обращения к методу, сначала будет обращаться к модулю. Главным методом модуля pygame является `init()`, в отличие от других изученных модулей. Так в проект подключается весь функционал установленного модуля. То есть теперь подключение модуля будет состоять из 2 строк.

Слайд 10. Создайте новый проект и назовите его `first_game`. Обратите внимание детей, что НЕЛЬЗЯ называть файлы также, как и модуль, это приводит к ошибке запуска программы.

4. Создание игрового окна (20 мин.).

Слайды 11-14. Вспомните с детьми, как ранее мы создавали окно приложения. Какие методы использовали, какую информацию передавали в программу. Все это повториться и при создании игрового окна.

Слайды 15-17. Создание игрового окна происходит с помощью объекта `display` – экран, и его метода `set_mode`, в который необходимо передать ширину и высоту окна. Лучше всего сразу детей приучить к заданию размера окна через переменные. Обратите внимание, что размер окна задается в дополнительных скобках, значит возможны еще дополнительные настройки окна, о которых мы узнаем позже, но размер окна является единым элементом – кортеж.

Слайды 18-33. Игровое окно создалось, но как и раньше, быстро закрылось. Причина этого нам уже хорошо известно, но решение этой проблемы в pygame совсем иная. Так как в приложениях есть статичные элементы: виджеты, пользователь может их только включать/выключать и т.п., но при это не надо отслеживать постоянно состояние всего окна. В играх все иначе, здесь пользователь взаимодействуя со всем игровым окном, управление персонажем или героем тоже зависит от пользователя и программа заранее не знает, какой решение примет пользователь, поэтому необходимо отслеживать все окно. К тому же неизвестно, как



долго это необходимо делать. Здесь мы знакомимся с понятием бесконечного цикла. Важно вспомнить, при каком условии завершался условный цикл – при ложном, как сделать так, чтобы условие было всегда истинным – True? Существует несколько вариантов, но сейчас самый доступный и понятный для детей – это в условие указать True. Передавать мы его будет через переменную, чтобы иметь возможность циклом управлять, вдруг, нам все таки захочется его завершить, не может же игра длиться вечно.

Слайд 34. Разберите с детьми конструкцию бесконечного цикла.

Слайды 35-38. Ранее мы отслеживали место, где пользователь совершал действие – нажимал левую клавишу мыши, делали мы это с помощью event – событие. Здесь происходит аналогичная ситуация, нам необходимо получать и обрабатывать каждого действие пользователя, что он сделал, как он это сделать и что делать дальше программе. Для того, чтобы получить значение этого действие используем уже известный нам метод get().

Слайд 40. Разберите с детьми конструкцию получение событий пользователя. Почему там используется арифметический цикл? Число действий пользователя конечно, его можно посчитать, поэтому нет необходимости использовать цикл условный.

Слайды 41-47. Какое же действие пользователя нам необходимо отследить? Как он же завершит игру? Он просто закроет окно! Как мы говорили ранее, у каждой кнопки есть свой идентификатор, что выбрали именно её. Так кнопке закрытия соответствует метод QUIT, обратите внимание детей, что пишется все заглавными буквами, иначе ничего не будет работать. В случае, если действие пользователя совпадет с QUIT, окно и должно закрыться. Так как мы не знаем точно, в какой момент времени это произойдет и произойдет ли вообще, нам необходимо сравнивать каждое действие с QUIT. Каждого событие в программе имеет свой тип, который можно получить с помощью метода type, его то мы и будем сравнивать с QUIT. Для этого нам понадобится условный оператор.

Слайд 48. Выполните задание, добавьте в программу обработку действий пользователя.

Слайды 49-54. После того, как пользователь нажал кнопку закрыть, цикл должен прекратить свою работу, то есть условие должно стать ложным. Это возможно сделать, только изменив значение, на которое указывает переменная game.

Слайды 55-56. Выдайте задание детям на добавление изменения переменной game.



5. Создание класса для спрайта (15 мин.).

Слайды 57-61. Обсудите с детьми из чего состоит любая игра, какие основные части должны присутствовать. Наша первая игра будет – гонка. Что в ней должно быть? Кто будет героем игры?

Слайды 62-63. Так как персонажей может быть большое количество, язык дает возможность описывать его своими инструментами, но дальше уже работа зависит от того, кто пишет программу. Героем гонки будет машина, поэтому нам необходимо будет создать специальный класс, который будет создавать нам такие машины, которые нужны именно в нашей игре.

Слайды 64-70. Каждый класс состоит из функций, которые выполняют определенные действия. Главной функцией является `__init__`, мы уже познакомились с ней в начале занятия. Но теперь мы сами должны написать, что же мы будем использовать и как создавать наш объект. Основным объектом игры является Спрайт.

Слайды 71-72. Добавляем в программу создание класса для спрайта. Обратите внимание, что файл с картинкой должен лежать в той же папке, что и код программы.

6. Добавление спрайта в игровое окно (20 мин.).

Слайд 73-76. Класс – это только описание объекта, но не сам объект. Следующим шагом необходимо создать объект, который станет экземпляром этого класса. Обратите внимание детей на схему, что помещается в `self`.

Слайды 77-83. Машина снова не появилась, потому что за обработку окна у нас отвечает бесконечный цикл. Причем, прежде чем разместить там объект необходимо задать цвет фона. Цвет задается 3 числами: красный, зелёный и синий. Белому цвету соответствуют максимальные значения – 255.

Слайды 84-87. Изменяем программу. Добавляем обработку спрайта на сцене.

Слайды 88-99. Обсудите с детьми, как должна вести себя машина. Нам не хватает движения. Машина должна начать двигаться сверху вниз. Какая координата будет изменяться? До какого момента должно длиться изменение? Что должно произойти, когда машина достигнет края?

Слайды 100-101. На основе обсуждений, добавляем в программу условный оператор, который будет отвечать за движение машины.



7. Рефлексия (5 мин.).

Слайд 102. Повторите материал занятия. Не забудьте загрузить проекты детей на склад.

«Отлично! Вы все большие молодцы! Но не забывайте, что только при регулярной практике и работе дома вы сможете добиться поставленных целей. Всем спасибо, жду вас на следующем занятии!»

8. Выдача задания на дом (5 мин.)

Слайды 103-104. Задание с двумя уровнями сложности:

1. Базовое, определяет степень усвоения нового материала.
2. Расширенное, связанное с материалами прошлых занятий.