

1. Повторение (3 часа)

Все проекты Модуля 3 рекомендуется выполнять в online-версии языка программирования Scratch. Поэтому на первом занятии необходимо повторить правила работы и поведения в компьютерном классе и напомнить основные правила безопасной работы в сети Интернет. Обучающиеся, не имеющие аккаунта на сайте <https://scratch.mit.edu>, создают его в начале занятия.

Первая тема модуля начинается с повторения ключевых понятий языка программирования Scratch и выполнения проектов по инструкции. Рекомендуем разработать совместно с обучающимися игру «Пинг-понг».

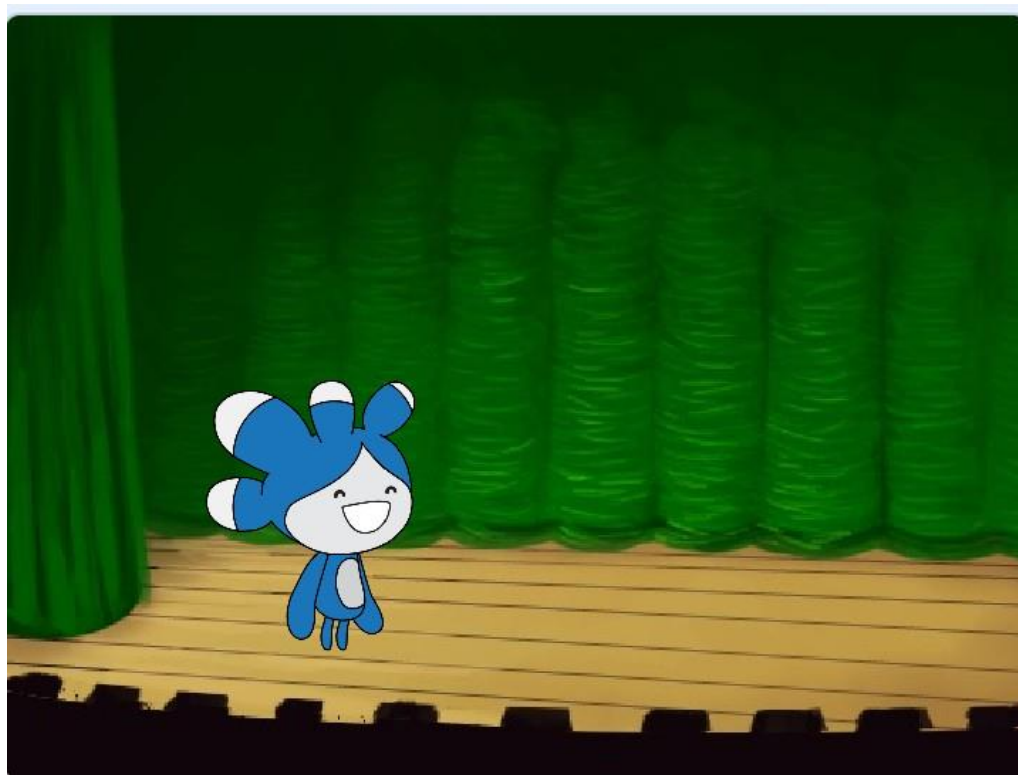
Pong — одна из ранних аркадных видеоигр. Это теннисная спортивная игра с использованием простой двумерной графики, разработанная и выпущенная фирмой Atari в 1972 году. Pong называют первой в истории коммерчески успешной видеоигрой; с её названием связывают появление индустрии интерактивных развлечений.

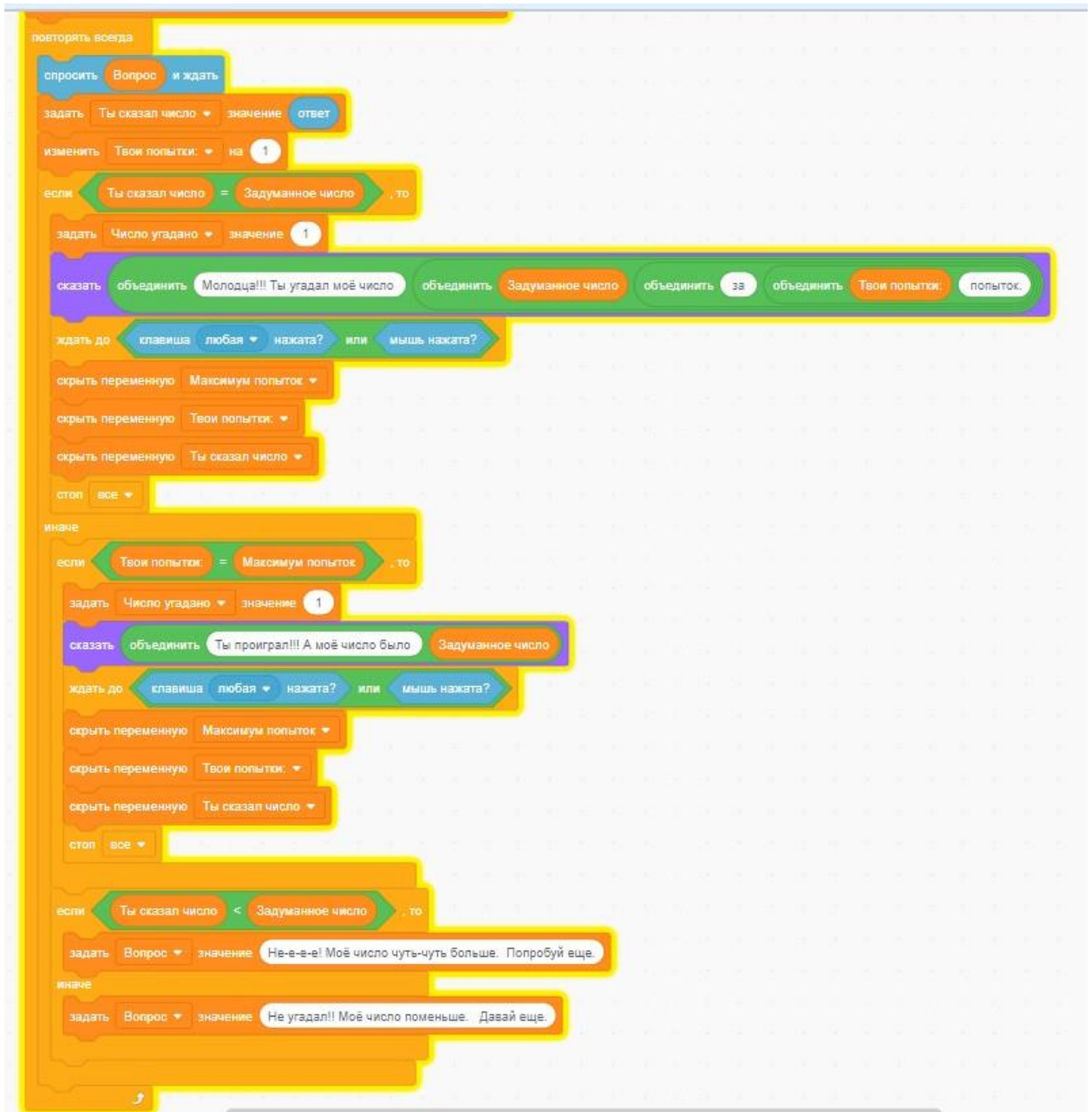
При создании игры возможно использование видеоинструкции, представленной на сайте <https://scratch.mit.edu> в разделе «Руководство» (<https://scratch.mit.edu/projects/editor/?tutorial=all>). Также возможно использование карточек с пошаговыми инструкциями по созданию игры. Они размещены по адресу: <https://resources.scratch.mit.edu/www/cards/ru/scratch-cards-all.pdf>

После создания игры по инструкции возможна доработка игры и развитие сюжета. Можно предложить создание версии игры для двух игроков. В качестве развития сюжета можно предложить игру «Атака вируса». В роли шарика выступает анимированная последовательность букв «V-I-R-U-S». А цель игрока – не допустить касания сервера в течении определенного интервала времени. Примеры игр размещены по адресу: <https://disk.yandex.ru/d/za5H5IUbXk5PtA>

Второе направление проектов – это игры с использованием диалогов и списков. Можно предложить разработать игры «Угадай-ка» и «Виселица».

В игре «Угадай-ка» игроку предлагается угадать случайное число в определенном интервале за ограниченное число попыток. Пример оформления экрана игры представлена на рисунке. Базовый программный код для ведущего игры представлен ниже.





В процессе тестирования игры рекомендуется познакомить обучающихся с оптимальной стратегией, позволяющей угадать число из любого интервала за наименьшее число ходов. Эта простейшая алгоритмическая задача, демонстрирующая мощь алгоритмического мышления. Смысл в том, чтобы каждый раз называть число, которое делит пополам диапазон возможных чисел. Этот приём каждый раз в 2 раза сокращает область поиска, и в конце нам становится легко угадать даже простым перебором.

Допустим, компьютер загадал число 63. Тогда делаем так:

Находим середину диапазона от 0 до 100 — это 50. Называем это число. Компьютер говорит «Больше». Значит, его число лежит в диапазоне от 50 до 100. Находим его середину — 75. Называем это число. Компьютер говорит «Меньше». Значит число находится в диапазоне от 50 до 75. Находим

середину — 62. Называем это число. Компьютер говорит «Больше». Поиск снова сузился: от 62 до 75. Середина — 67. Называем это число. Компьютер говорит «Меньше». У нас осталось 3 попытки и 4 числа. Найдём ещё одну середину — 64. Называем это число. Компьютер говорит «Меньше». Раз у нас число, которое больше, чем 62, и меньше, чем 64, то это число 63. Число угадано за шесть попыток.

Этим способом можно угадать любое число от 0 до 100 за 7 попыток или меньше. Главное — быстро и правильно считать в уме середину и помнить, как выглядит сейчас твой рабочий диапазон. А если число будет больше? На самом деле за 7 шагов можно угадать любое число от 0 до 127 или от 1 до 128. Всё потому, что два в седьмой степени — это как раз 128. Каждый раз, когда мы делим рабочий диапазон на 2, мы как будто убираем одну степень у двойки, постепенно уменьшая наш диапазон угадывания до двух чисел. Если бы у нас было 8 попыток, можно было бы угадывать числа до 256. 9 попыток — 512 и так далее.

После реализации базового проекта можно предложить следующие направления доработки программы:

1. Добавление звукового сопровождения (озвучивание реплик героя, добавление мелодии в случае победы или поражения).
2. Добавление мимики для ведущего (различные костюмы спрайта для различных ответов).
3. Добавление дополнительных сообщений в случае отсутствия ответа игрока. Пример приведен ниже.



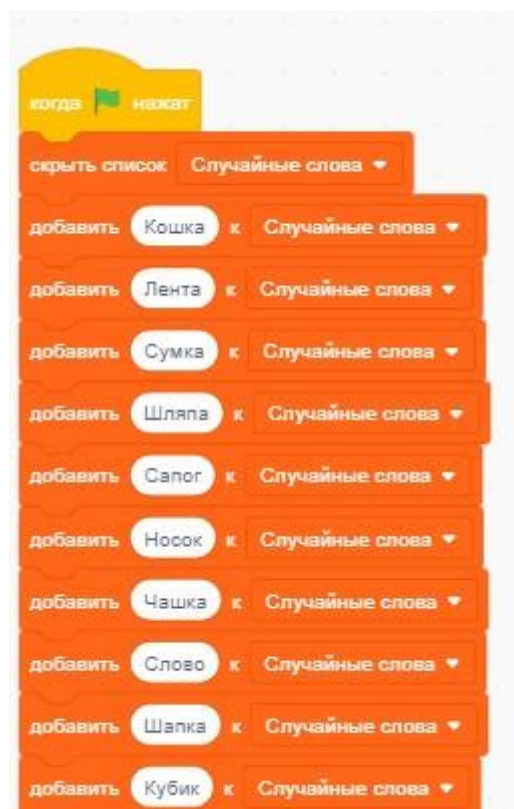
Более сложный в реализации проект – игра на угадывание загаданного слова, известная как «Виселица».

Вот правила классической игры. Один из игроков загадывает слово — пишет на бумаге любые две буквы слова и отмечает места для остальных букв, например чертами (существует также вариант, когда изначально все буквы слова неизвестны). Также рисуется виселица с петлёй. Согласно традиции

русских лингвистических игр, слово должно быть именем существительным, нарицательным в именительном падеже единственного числа, либо множественного числа при отсутствии у слова формы единственного числа. Второй игрок предлагает букву, которая может входить в это слово. Если такая буква есть в слове, то первый игрок пишет её над соответствующими этой букве чертами — столько раз, сколько она встречается в слове. Если такой буквы нет, то к виселице добавляется круг в петле, изображающий голову. Второй игрок продолжает отгадывать буквы до тех пор, пока не отгадает всё слово. За каждый неправильный ответ первый игрок добавляет одну часть туловища к виселице (обычно их 6: голова, туловище, 2 руки и 2 ноги, существует также вариант с 8 частями — добавляются ступни, а также самый длинный вариант, когда сначала за неотгаданную букву рисуются части самой виселицы). Если туловище в виселице нарисовано полностью, то отгадывающий игрок проигрывает, считается повешенным. Если игроку удаётся угадать слово, он выигрывает и может загадывать слово.

Немного адаптируем игру для ее реализации в Scratch.

Слова будем хранить в списке и добавлять в начале игры. Фрагмент кода приведен ниже.

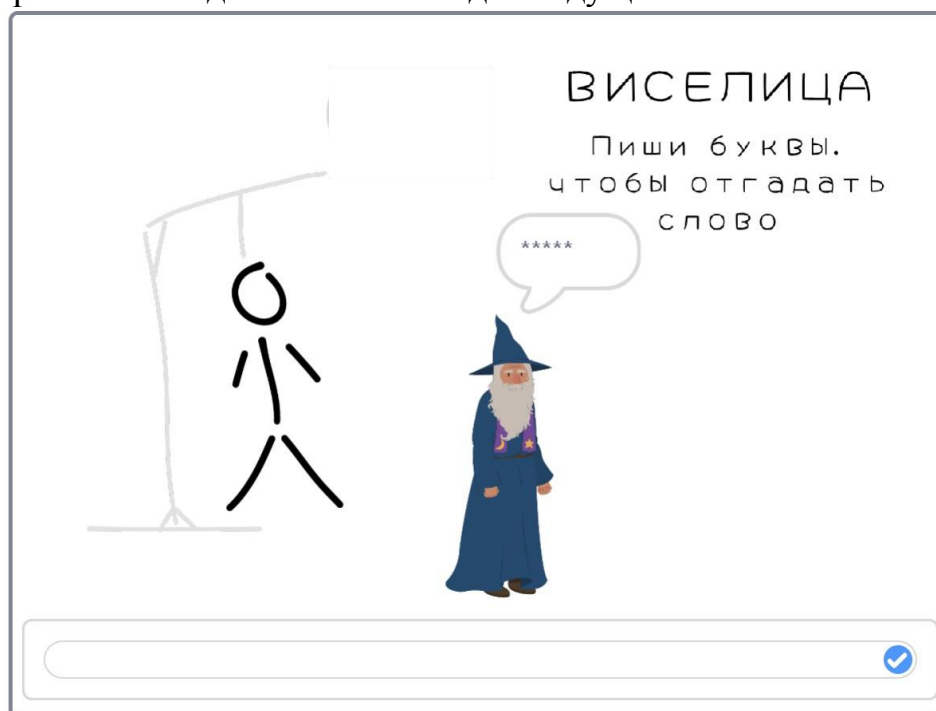


На фоне разместим правила игры и спрайт с изображением виселицы:

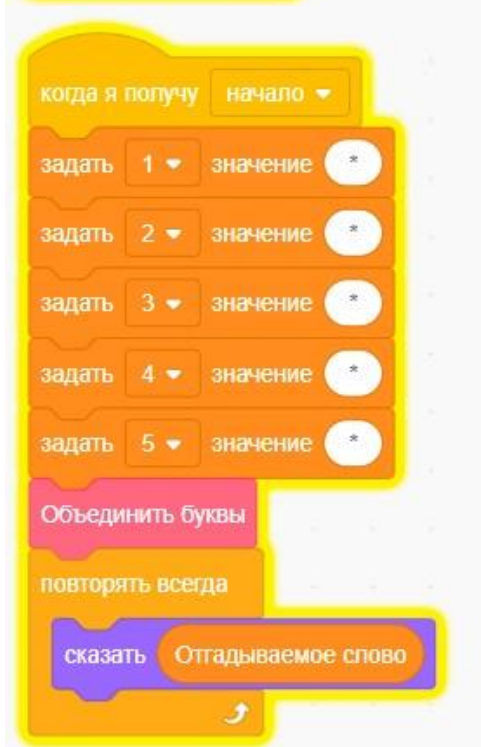
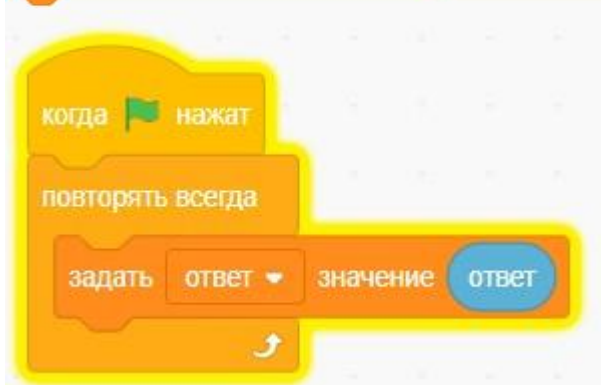
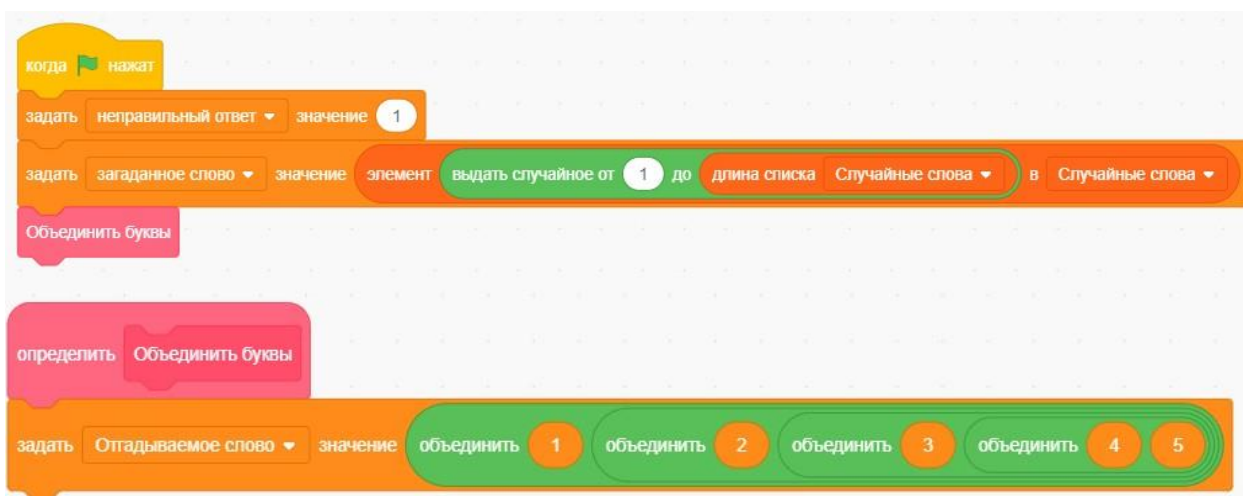


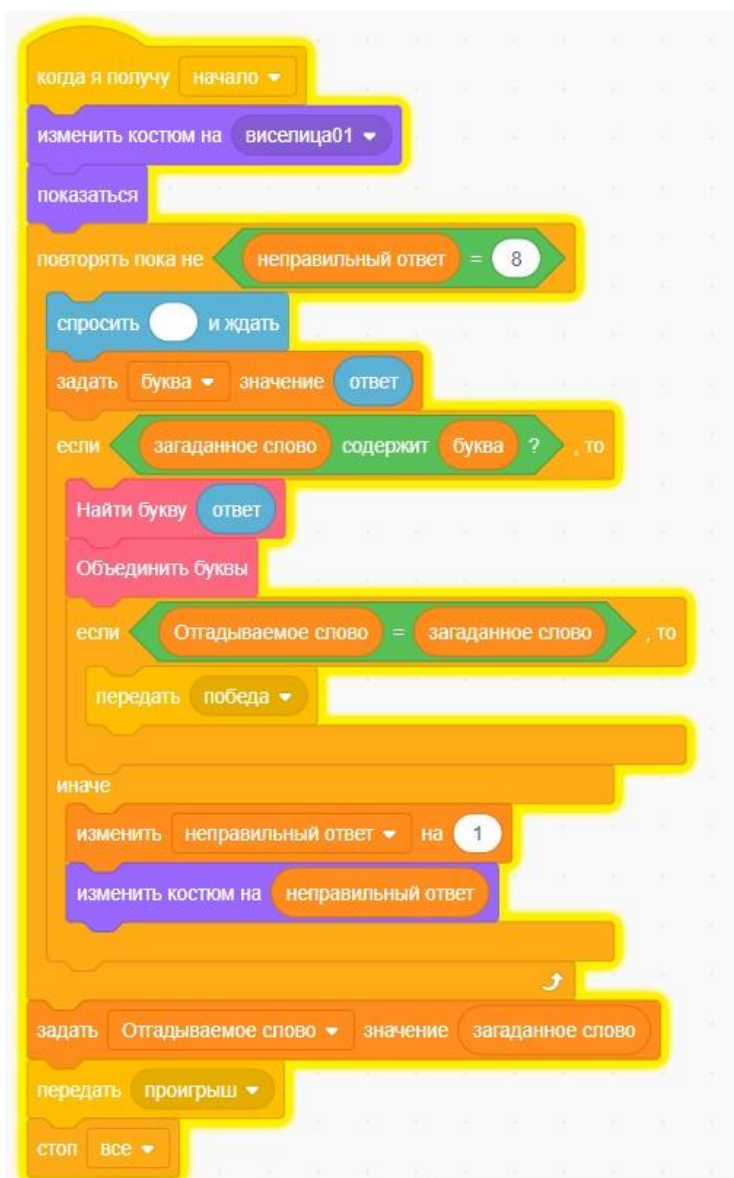
Спрайт с изображением виселицы должен иметь столько костюмов, сколько попыток мы даем. При каждой неверной попытке происходит смена костюма и одна из цветных линий становится черной. Игра завершается в тот момент, когда игрок назовет восемь неправильных букв.

Программный код можем писать для ведущего.



Программный код игры приведен далее.







Возможные направления доработки программы:

1. Расширение словарного запаса.
2. Добавление экрана-заставки, победы и поражения.
3. Изменение демонстрации угадываемого слова.
4. Ведение протокола игры (в виде списка).

Примеры реализации игр «Угадай-ка» и «Виселица» размещены по адресу: https://disk.yandex.ru/d/F_ggYCR182HfCA