

## Компьютерная игра — своими руками (4 часа)

### 1. События.

Событие – сигнал, по которому запускаются определенные скрипты. Стандартные (системные) события: нажатие на зеленый флажок, клавишу.

### 2. Команды, рассматриваемые в ходе изучения темы:

Группа	Команда (оператор)
СОБЫТИЯ	«Когда клавиша нажата»

Управление спрайтом с помощью клавиш (ВВЕРХ, ВНИЗ, ВЛЕВО, ВПРАВО)

**Видеоурок:** Scratch для начинающих. Урок 3.  
(<https://www.youtube.com/watch?v=YjkKBuIU5Mo>)

### 3. Игра «Догонит ли кошка мышку?».

Сценарный план:

1. Кто? Что? В игре участвуют два персонажа: кошка (Cat 2) и мышка (Mouse1).
2. Где? Подходящий фон, например, лес (Forest).
3. Как? Мышка управляется игроком с помощью клавиш стрелок, кошка все время движется по направлению к мышке.



Рисунок 1. Пример сцены для игры "Догонит ли кошка мышку?"

Задание для учащихся: подумайте, как можно изменить третий пункт сценарного плана? Как это можно реализовать?

Примеры:

- поменять управление кошки и мышки (мышка убегает, кошка управляется игроком и пытается догнать мышку);
- управление мышкой с помощью других клавиш, например, WASD;
- управление мышкой с помощью компьютерной мыши;
- задать управление клавишами для обоих персонажей (на двоих игроков).

### 4. Дополнение игры «Догонит ли кошка мышку?».

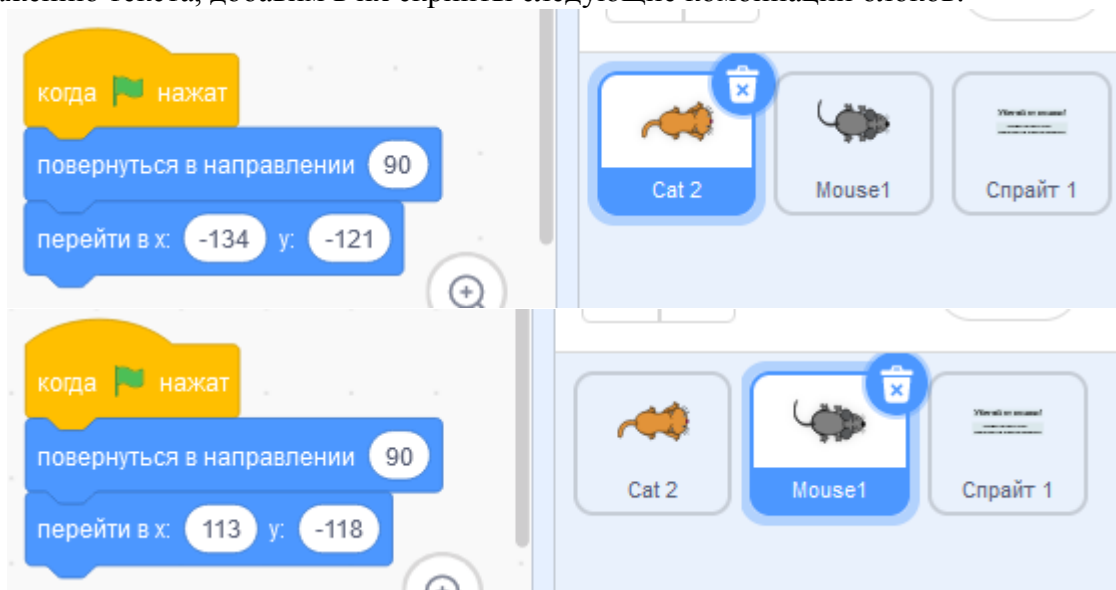
Предложите ученикам оформить заставку для игры. Пусть при запуске по зеленому флажку, прежде чем начать игру, игрок увидит надпись с целью игры и информацией о том, как начать играть. Сделаем так, чтобы для начала игры нужно было нажать на любую клавишу.

Это можно сделать с помощью добавления нового спрайта с начальным текстом к игре. Каждый может создать и расположить на сцене спрайт с текстом по своему усмотрению, стоит лишь обратить внимание учеников на читабельность и понятность текста на сцене, чтобы цветное, масштабное и прочее оформление соответствовало сцене игры.



Рисунок 2. Пример оформления спрайта с текстом

Чтобы в начале игры при просмотре заставки спрайты кошки и мышки не мешали отображению текста, добавим в их скрипты следующие комбинации блоков:



Обратите внимание на то, что числа в команде «перейти в» должны соответствовать желаемому начальному расположению спрайтов на сцене.

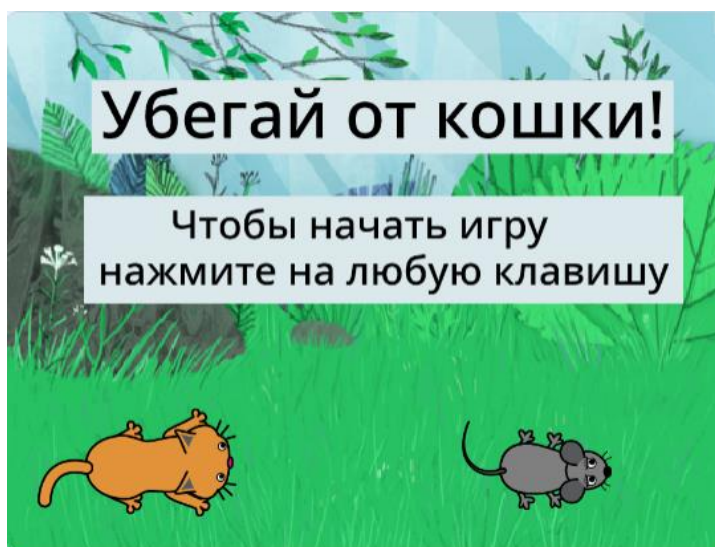
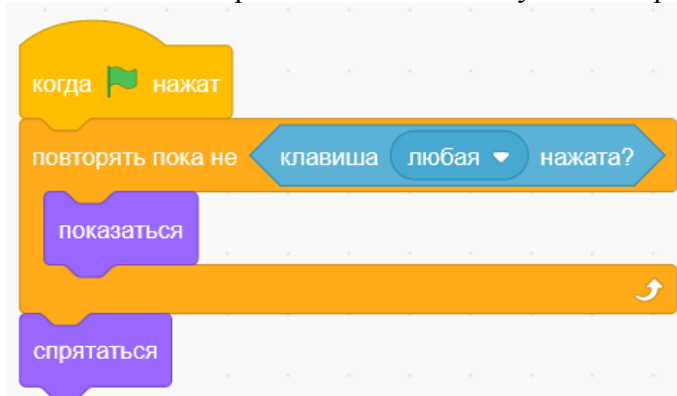
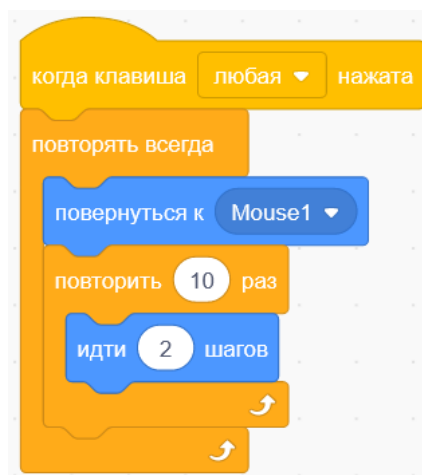


Рисунок 3. Пример заставки игры

Напишем для созданного нами спрайта с текстом следующий скрипт:



Теперь для того, чтобы кошка не начинала бежать за мышкой до тех пор, пока игрок не запустит игру нажатием любой клавиши, немного изменим скрипт кошки, заменив первый командный блок на команду «когда клавиша *любая* нажата» (обратите внимание учеников, что запускать скрипты можно нажатием любой клавиши из большого предложенного списка):

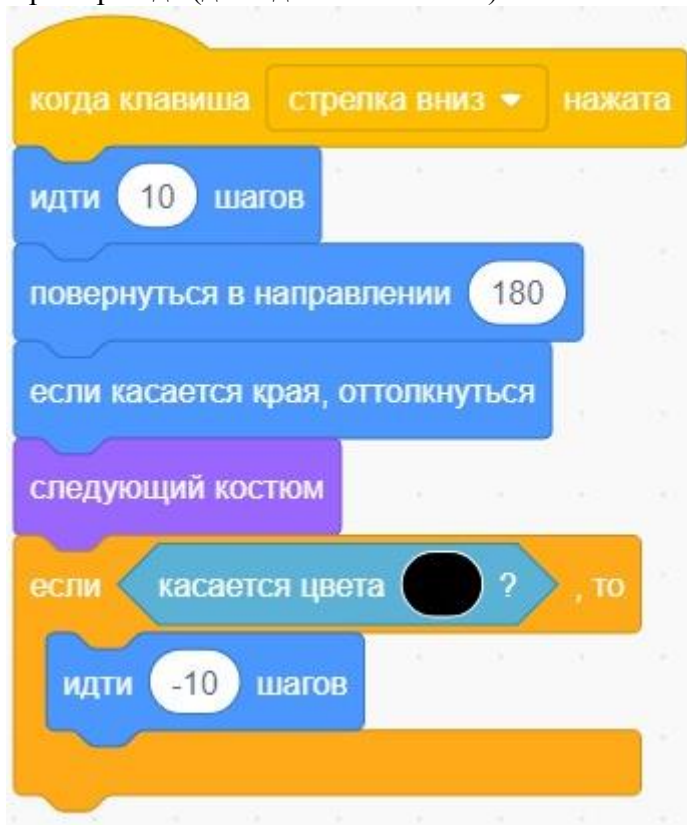


Готово!

### 6. Совершенствуем игру «Кошки-мышки»

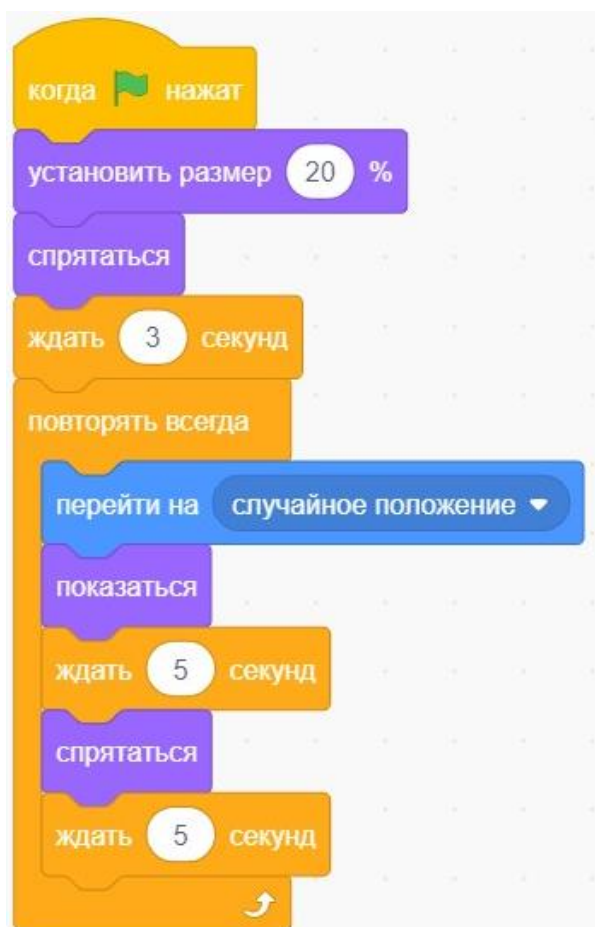
1. Кошка и мышка двигаются в лабиринте. Если герой касается цвета стенок, то происходит движение на такое же количество шагов, только со знаком минус.

Пример кода (для одной из клавиш)



2. На поле появляется сыр (или другая еда), которая исчезает, когда ее касается мышка.

Пример кода (для еды)

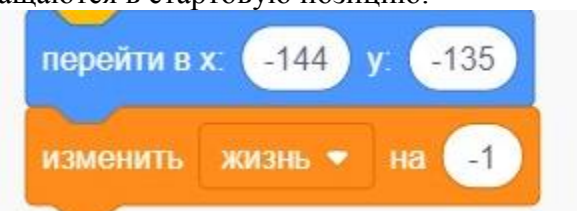


3. Можно добавить счет. Для этого в группе «Переменные» создаем переменную счет



и добавляем в программу для еды оператор:

4. Можно добавить мышке несколько жизней. Для этого создать переменную Жизни. При касании кошки и мышки жизнь уменьшается на одну единицу, а кошка и мышка возвращаются в стартовую позицию.



5. Можно добавить экран победы и поражения. Для поражения можно предложить: касание кошки и мыши или уменьшение жизни до нуля. Для победы: собранный сыр (или другая еда).

Примеры доработанных проектов размещены здесь: [https://disk.yandex.ru/d/68Pa-bW\\_GECYjg](https://disk.yandex.ru/d/68Pa-bW_GECYjg)

### 7. Игра по мотивам игры «Кошки-мышки». Игра «Любитель яблок»

Для создания игры нам необходимо проделать несколько этапов:

- Создание фона – игрового поля,

- Расстановка различных предметов (объектов) на игровом поле;
- Программирование главного героя;
- Программирование собираемых объектов;
- Программирование препятствий;
- Программирование финиша (финишной кнопки).

Обратите внимание: инструкция неполная, Вам необходимо доработать программу, чтобы игра получилась. Возможны и другие вариации сюжета, костюмы героя и объектов для сбора и препятствий.

### Этап 1. Создание игрового поля

Для создания игрового поля мы должны войти в меню «Сцена», вкладка «Фоны» и с помощью инструментов для рисования «*прямоугольник*» и «*линия*» создать поле 8x8 клеток. (рис. 4). Такого размера вполне достаточно для создания первой игры. При желании можно добавить какой-либо текст (кнопка Т).

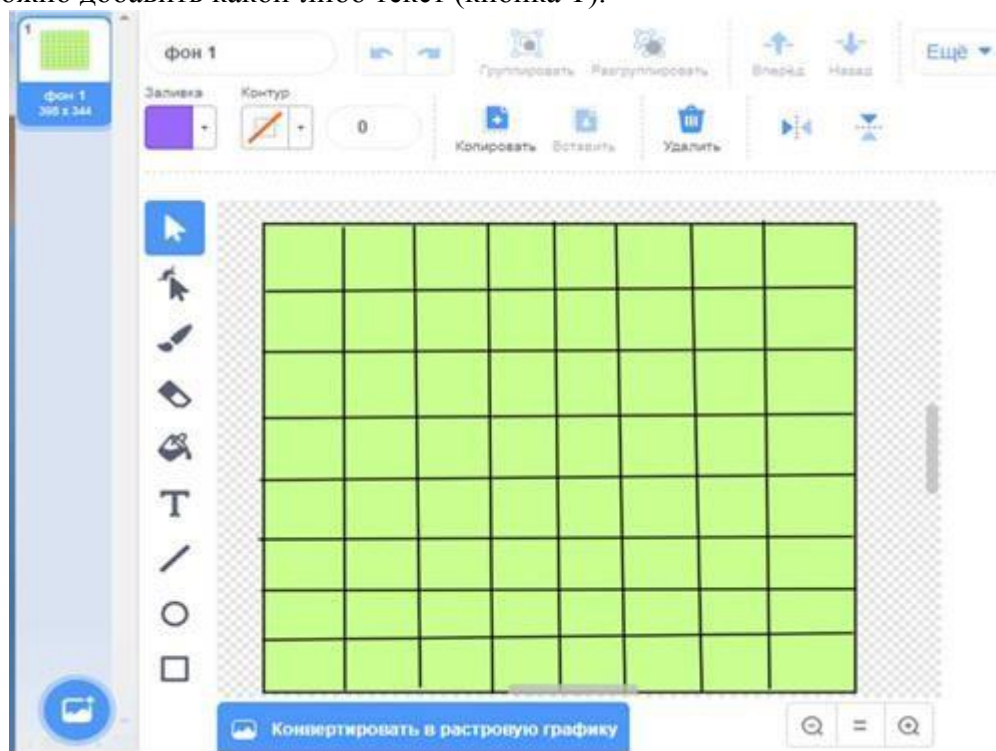


Рис. 4. Создание игрового поля.

### Этап 2. Расстановка игровых объектов

На втором этапе мы расставляем объекты разных видов (рис. 5 и 6). Для этого используем добавление спрайтов из библиотеки. Объекты в нашей игре будут двух видов – те, которые по условию надо собрать (например, яблоки), и те, которые будут являться препятствиями. Количество объектов можно задавать самостоятельно. Рекомендую размер объектов уменьшить до 50.

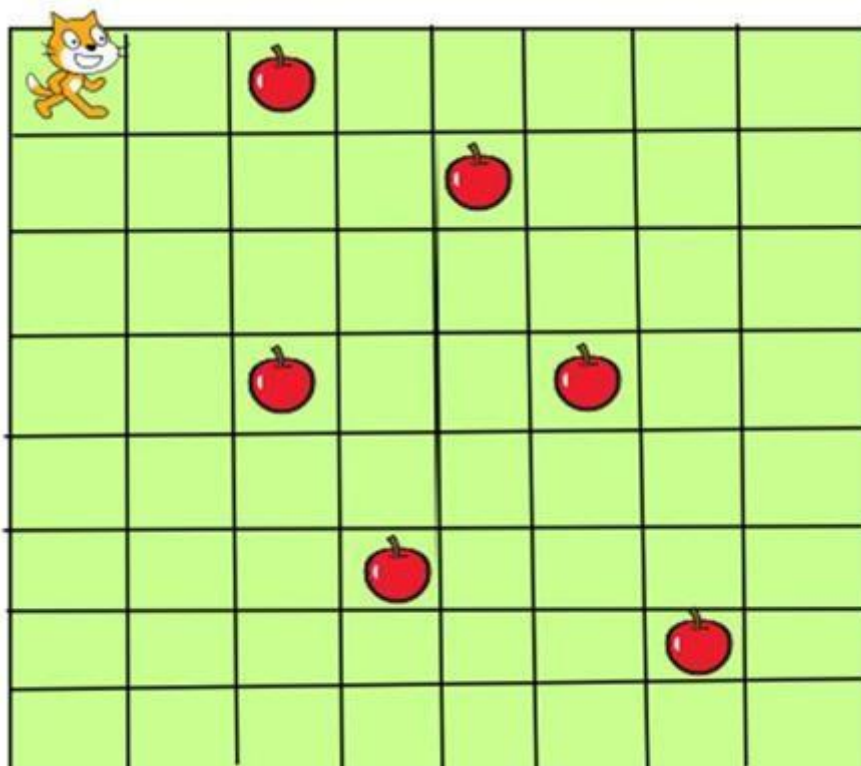


Рис. 5. Добавление объектов на игровое поле.

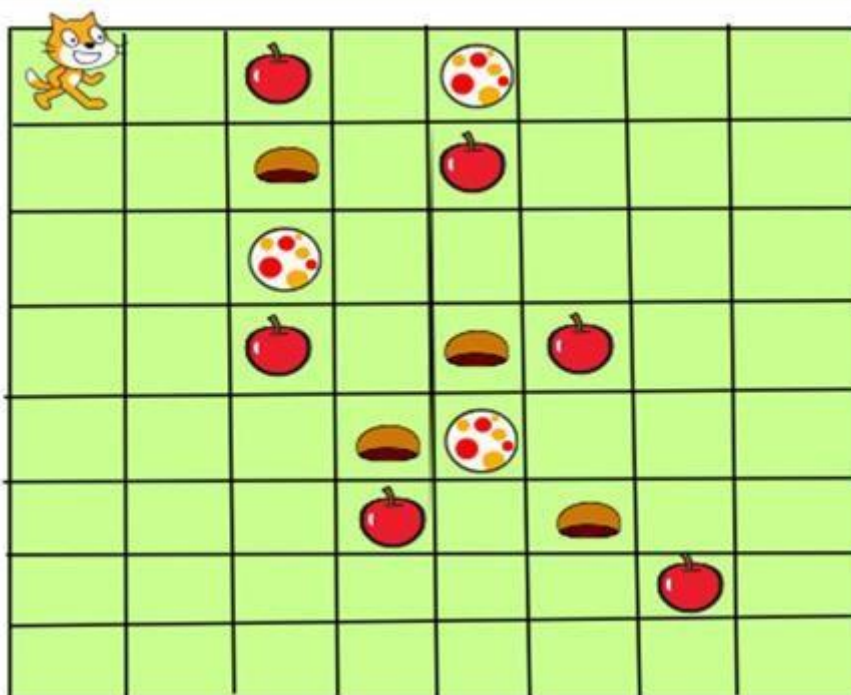


Рис. 6. Добавление объектов-препятствий на игровое поле.

### Этап 3. Программирование «героя»

В качестве героя можно оставить исходного спрайта-котенка под именем «Спрайт1». При желании можно выбрать (назначить) «героем» любого другого. Имя спрайта также можно изменить.

В начале игры наш герой должен встать в исходную точку, которую мы определяем с помощью координат. Поэтому команда должна выглядеть так:

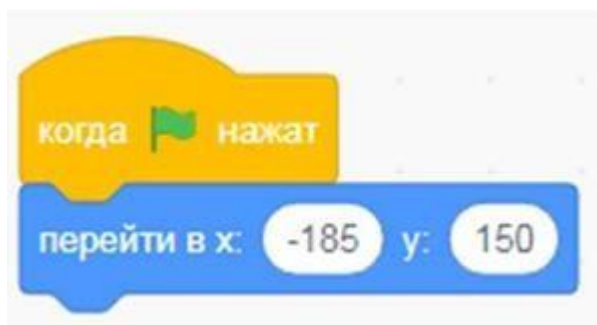


Рис. 7. Задание начальной координаты для спрайта.

При необходимости можно задать и начальное направление – например, «вернуться в направлении 90».

Далее для того, чтобы наш спрайт ходил сразу по клеткам, мы должны запрограммировать кнопки управления его передвижением (стрелки вверх, влево, вправо, вниз), которые удобно расположены на клавиатуре.

Команды для кнопок должны выглядеть следующим образом:

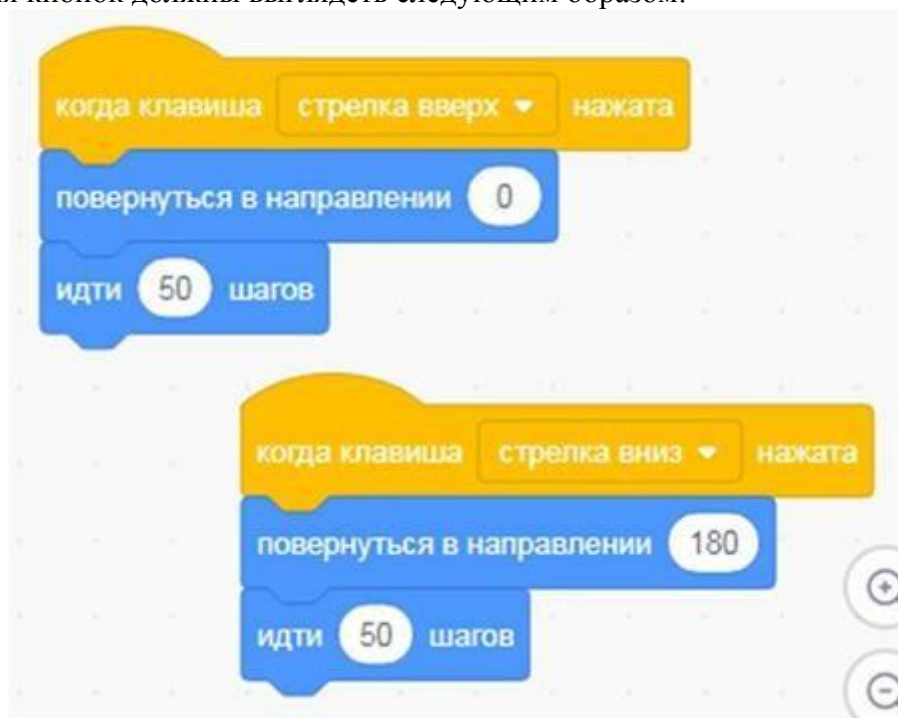


Рис. 8. Программирование управляющих кнопок (стрёлок).

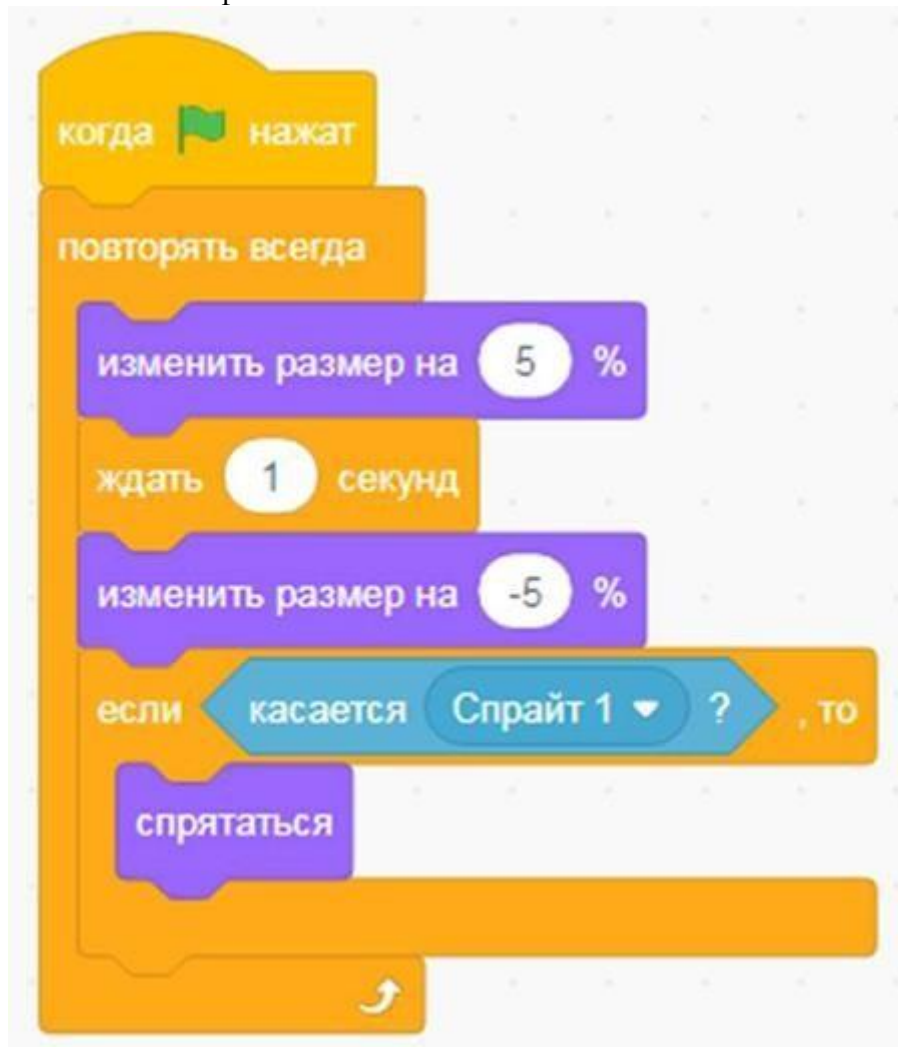
Таких команд должно быть четыре. Направление задается следующим образом: вверх – **0**; вправо – **90**; вниз – **180**; влево – **-90**. Длина шага определяется размером клеток на игровом поле. При размере поля 8x8 клеток шаг получается – примерно 50. Далее можно проверить, как ходит герой и подредктировать длину шага.

#### Этап 4. Программирование собираемых объектов

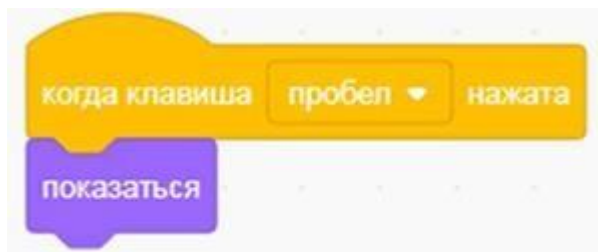
Для того, чтобы любой объект реагировал на прохождение «героя» и мог менять свою форму или исчезать, необходимо, чтобы он совершал небольшое движение. Это можно задать двумя способами:

1) небольшое движение «фишки» (например, «идти 1 шаг», «ждать 1 секунду», «идти -1 шаг» либо 2) «изменить размер на 5%», «ждать 1 секунду», «изменить размер на -5%».

Реакция на прохождение героя задается через ветвление «Если – то» с условием «касается Спрайт1» - команда «Спрятаться». Все команды запускаются по команде «Когда флажок нажат» и через цикл «Повторять всегда», так как мы не знаем, в какой момент игры наш герой коснется именно этой фишки.



Чтобы объекты можно было «восстановить» в начале игры задаем дополнительную команду:

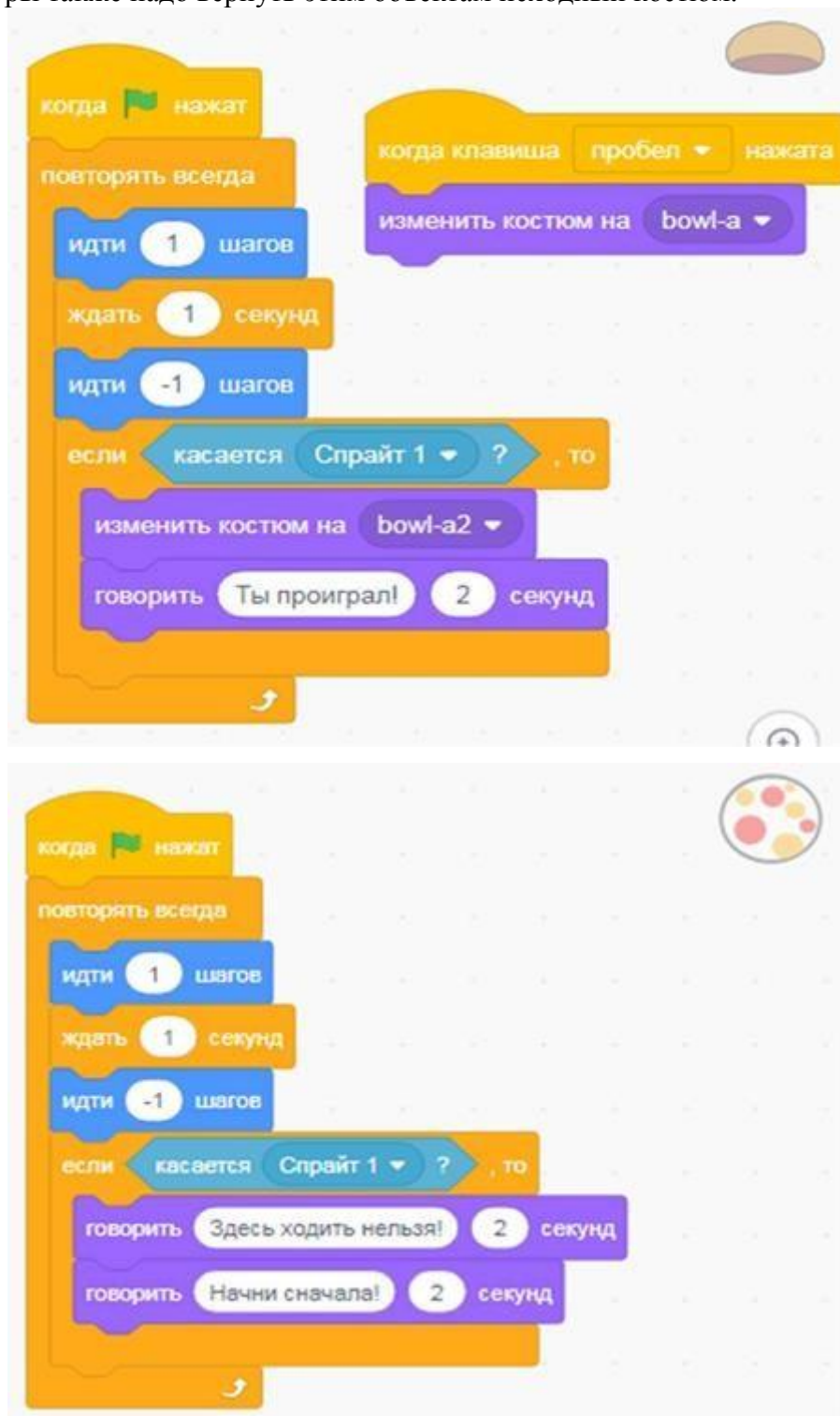


Эти команды надо задать для каждого собираемого объекта (яблока, монеты или звезды).

#### Этап 5. Программирование препятствий

На игровом поле должны также присутствовать объекты, которые будут усложнять игру, т.е. являться препятствиями. Чтобы они могли взаимодействовать с героем, когда он попадает на соответствующую клетку, необходимо им также задать небольшое движение (аналогично собираемым объектам).

Взаимодействие с героем можно задавать по-разному: препятствие может «взрываться», «переворачиваться» (т.е. изменить свой костюм), выдавать сообщение «Ты проиграл» или «Здесь ходить нельзя» и т.д. Можно задать команду, которая будет перемещать героя в начало игры, т.е. в исходную точку. Варианты могут придумать сами дети. В начале игры также надо вернуть этим объектам исходный костюм.



#### Этап 6. Программирование кнопки финиш

По желанию можно создать кнопку «Финиш» и поместить ее в конце игрового поля. Кнопка «Финиш» программируется аналогично предыдущим объектам. Отличие будет в

тех командах, которые будут выполняться при реакции на прохождение Спрайта. Например, можно задать «Говорить «Ты победил!» или «Переходи на второй уровень» и при этом кнопка может менять цвет или мигать и т.д. Здесь дети могут использовать свою фантазию и желание.

