

КОНСТРУКТОРСКОЕ БЮРО
КУБИК
РОБОТОТЕХНИКА И ПРОГРАММИРОВАНИЕ

Создание игр
в SCRATCH

УРОК 25: Стрекоза

Цели

занятия:

- Настроить спрайтам поиск случайного положения, **БЕЗ касания другого спрайта**
- Использовать новый формат управления с **поворотами**
- Запрограммировать игру «**стрекоза**»



Привет!

Давайте вспомним,
что мы делали в
прошлый раз



000

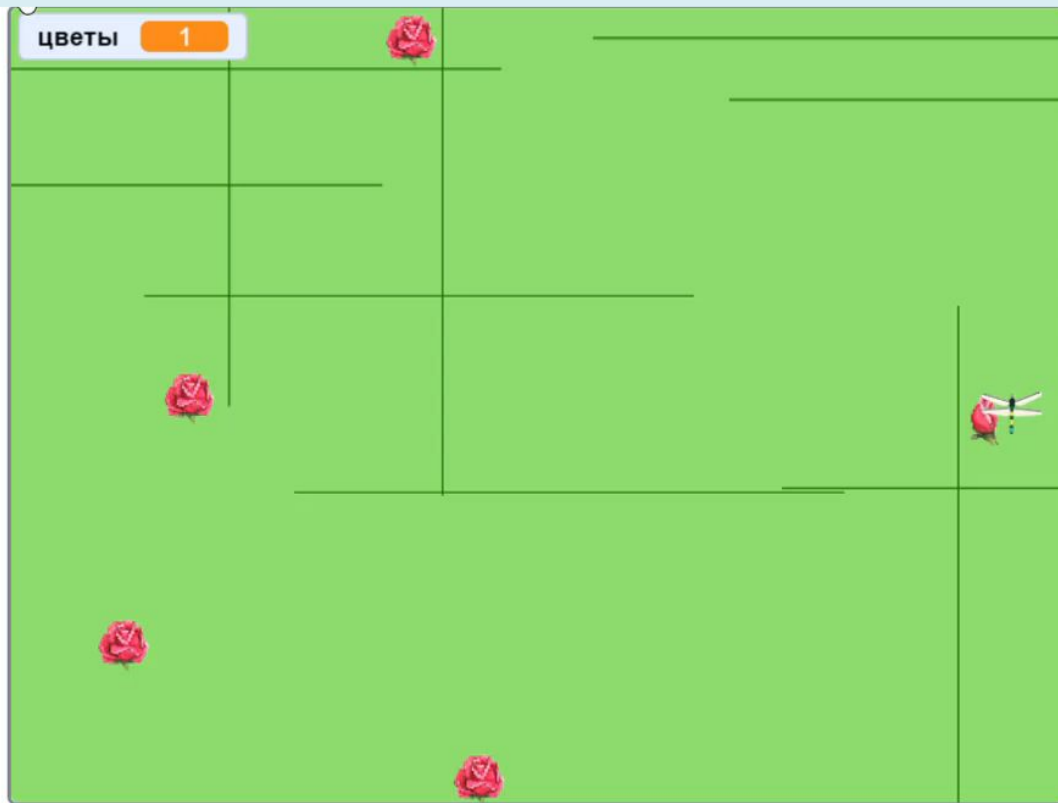




Расскажите мне:

- Что такое «СООБЩЕНИЯ»?
- Как мы их использовали?
Какие сигналы предавали?

Сегодня мы создадим игру, в которой стрекоза будет летать над препятствиями и собирать нектар с цветов.
Посмотрите и объясните правила игры?





Мы играем за **стрекозу**, которой управляем с помощью **стрелок**. Цель – собрать все цветы.

Заборы, цветы и сама стрекоза расставляются **случайно** в начале каждой игры (но никогда **не пересекаются**). Стрекоза может **перепрыгивать** заборы при нажатии **пробела**.

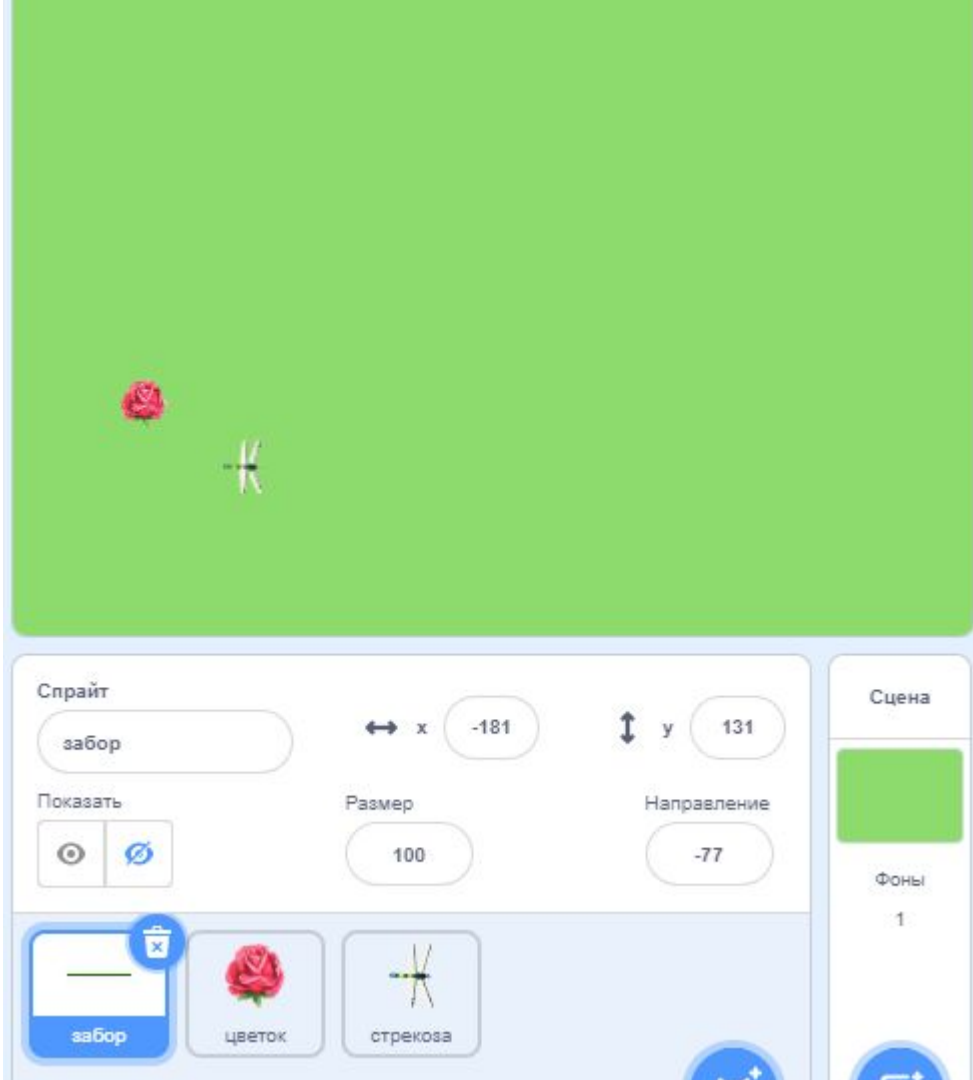
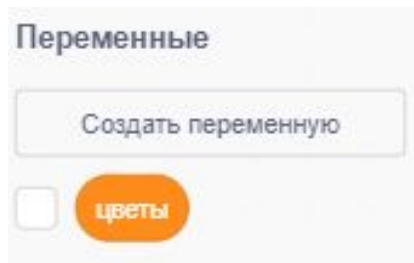
Игра заканчивается **выигрышем**, если стрекоза **коснулась всех цветов**. Игра заканчивается **проигрышем**, если стрекоза **коснулась забора**.

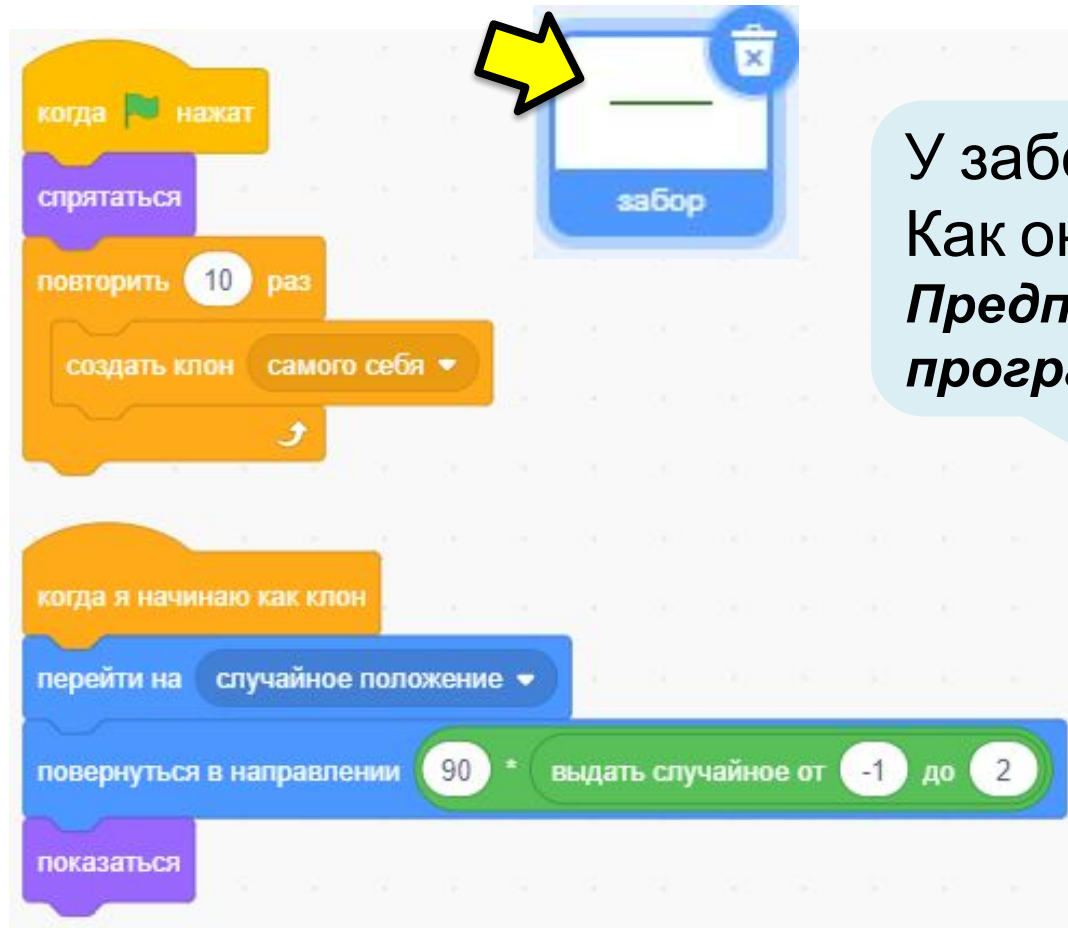
Этапы программирования игры:

1. Сделаем так, чтобы **цветы НЕ** росли на заборах
2. Запрограммируем стрекозе **точку старта** (*тоже НЕ на заборе*)
3. Добавим стрекозе **управление клавишами**
4. Запрограммируем **прыжок** стрекозы
5. Добавим **касание стрекозы и забора** – если оно случится игре конец!
6. **Дополнительно:** добавим подсчёт цветов и анимацию



В поле игры у вас уже
есть **цветок**, **стрекоза**,
забор и переменная
«**цветы**»





У забора есть программа.
Как она сработает?
Предположите, не запуская программу



У цветов тоже есть программа. Перед созданием клонов стоит **пауза** – цветы ждут, когда на поле появятся заборы

The image shows a Scratch script and a sprite. The script is divided into two sections. The first section, triggered by 'when green flag clicked', contains: a 'hide' block, a 'wait 1 second' block, a 'repeat 5 times' loop containing a 'create clone of self' block. The second section, triggered by 'when I start as a clone', contains: 'change costume to flower', 'show', and 'go to random position'. To the right is a 'flower' sprite, which is a red rose icon in a blue frame with a trash icon in the top right corner. A yellow arrow points from the 'create clone of self' block to the 'flower' sprite.

когда нажат

спрятаться

ждать 1 секунд

повторить 5 раз

создать клон самого себя

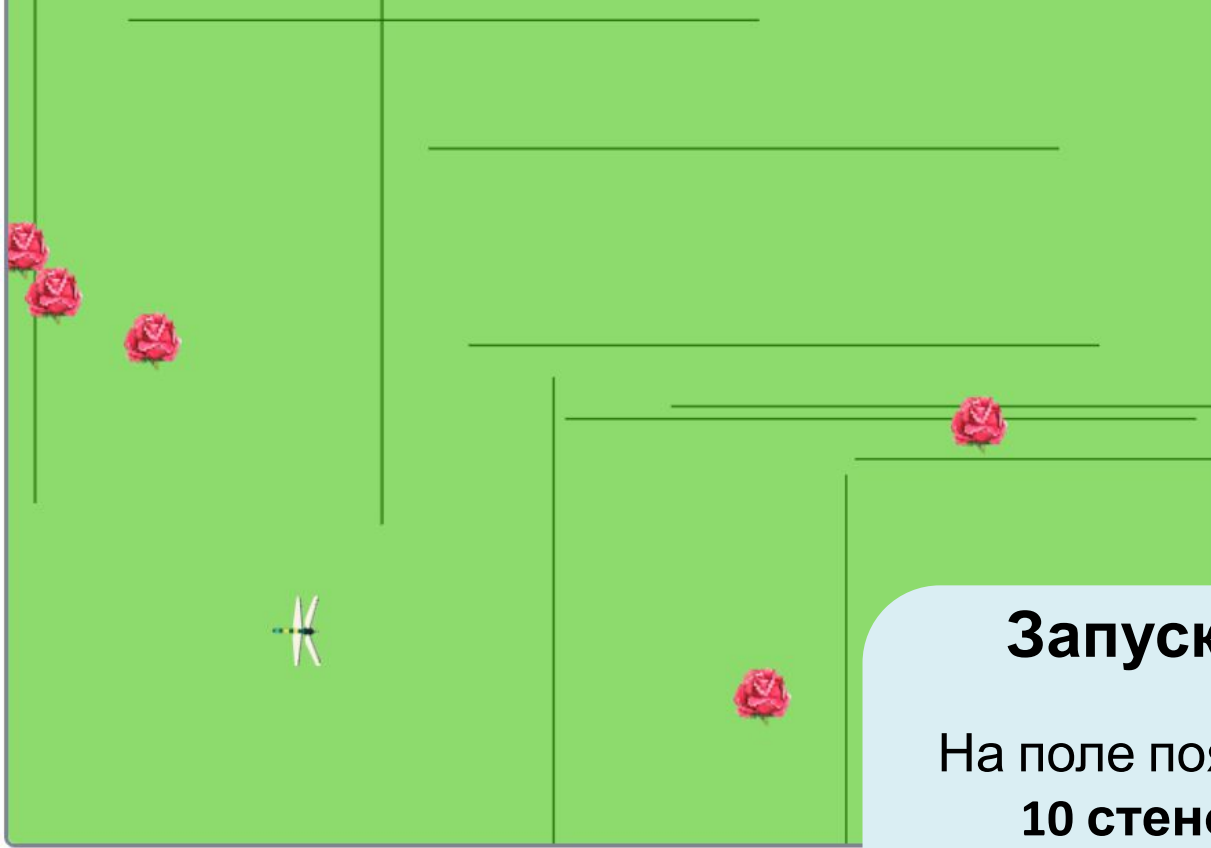
когда я начинаю как клон

изменить костюм на

показаться

перейти на

цветок



Запускаем!

На поле появляется
**10 стенок и 5
цветков**, они
расположены **в
случайных местах**



У стрекозы запрограммирована **победа в игре**. Эта программа пригодится и сработает в самом конце, когда мы начнём прибавлять очки за собранные цветы





Этап 1.

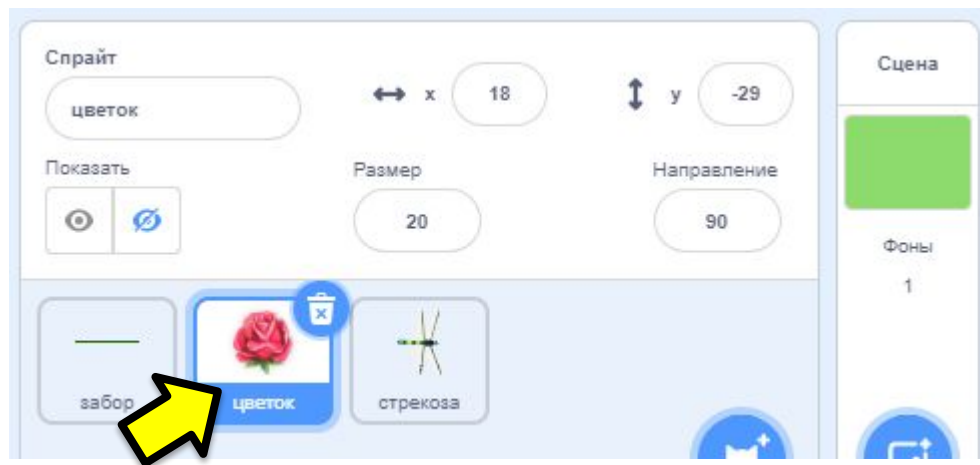
Запрограммируем цветы так,
чтобы они не появлялись на
заборах.

Для этого будем перемещать
их в случайное положение,
пока они будут касаться
заборов.

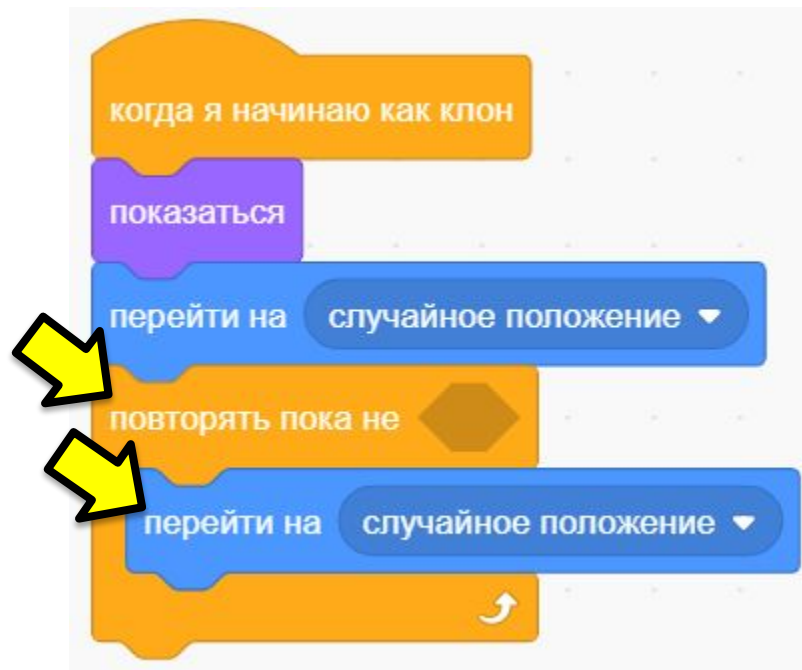
При написании программы всегда проверяйте, какому **СПРАЙТУ** вы добавляете команды.



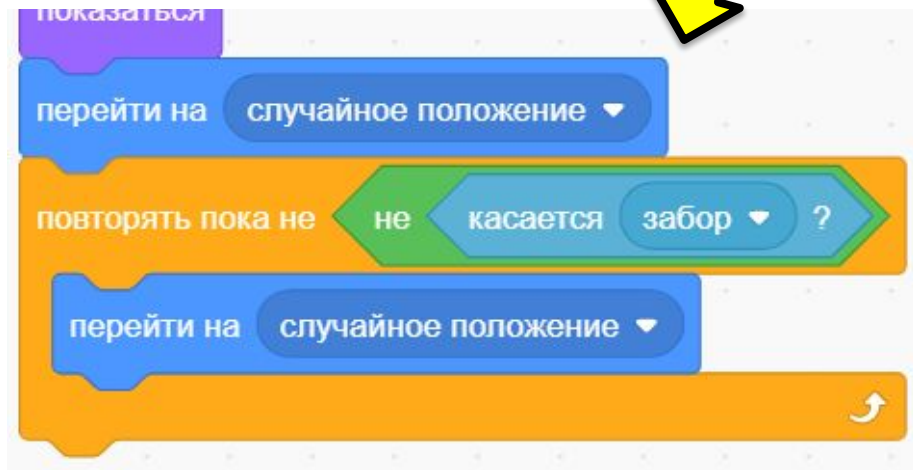
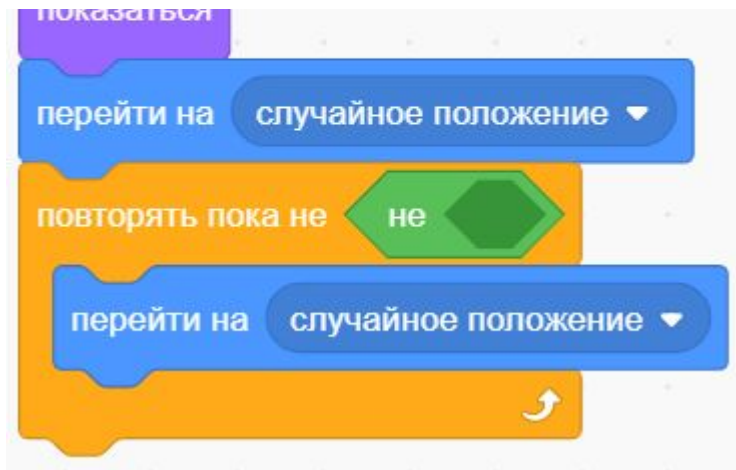
Проверьте, что здесь выделен **ЦВЕТОК**



В ЦИКЛЕ повторяем команду «ПЕРЕЙТИ НА СЛУЧАЙНОЕ ПОЛОЖЕНИЕ»



Пока **цветы касаются забора** – цикл размещает их в случайных положениях



Запускаем!
Цветы больше не
пересекаются с заборами!



Этап 2.

Запрограммируем стрекозе **точку старта** – она будет определяться случайно, но тоже НЕ на заборе.

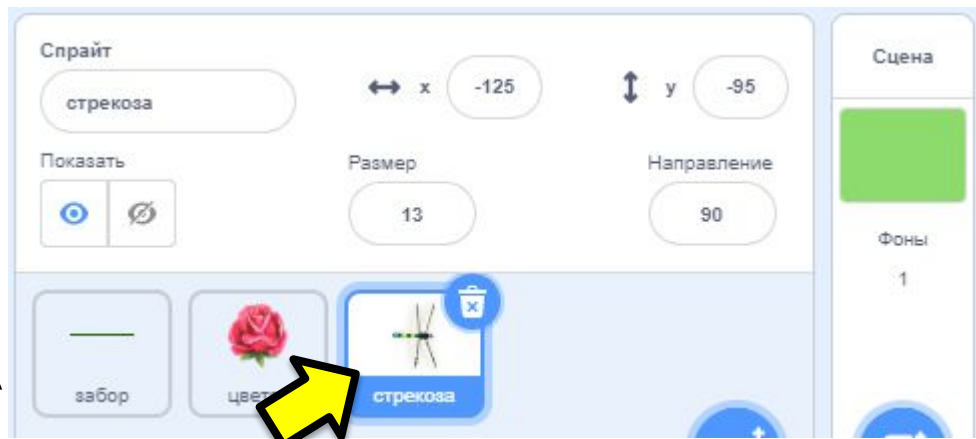
Для этого сначала нужно **подождать**, пока появятся заборы и цветы, а затем поставить **стрекозу в случайном месте**. Если она **касается забора** – нужно **переставить ее**



При написании программы всегда проверяйте, какому **СПРАЙТУ** вы добавляете команды.



Проверьте, что здесь выделена **СТРЕКОЗА**





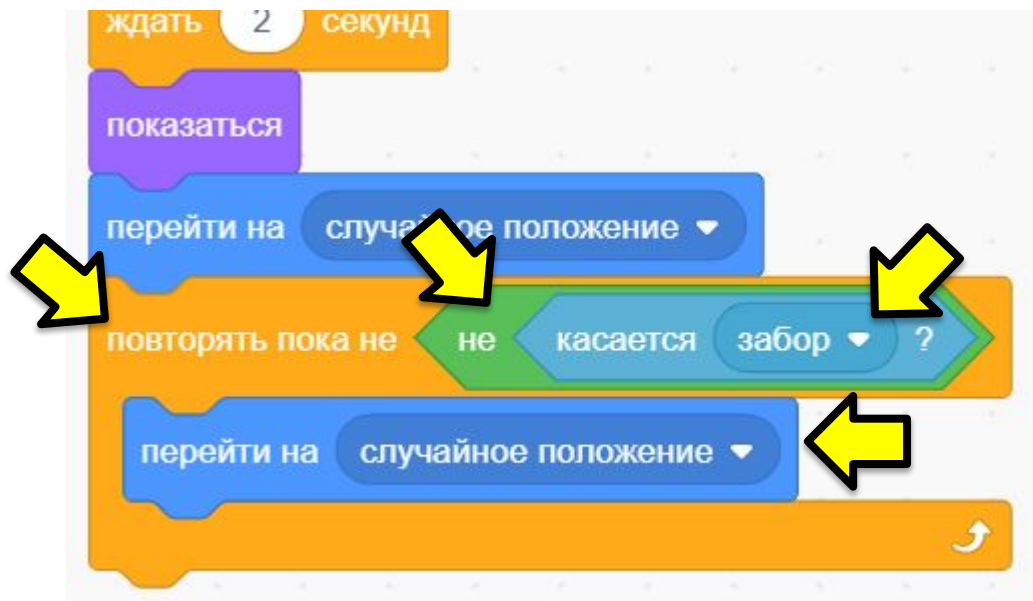
Самостоятельно добавьте в программу стрекозы:

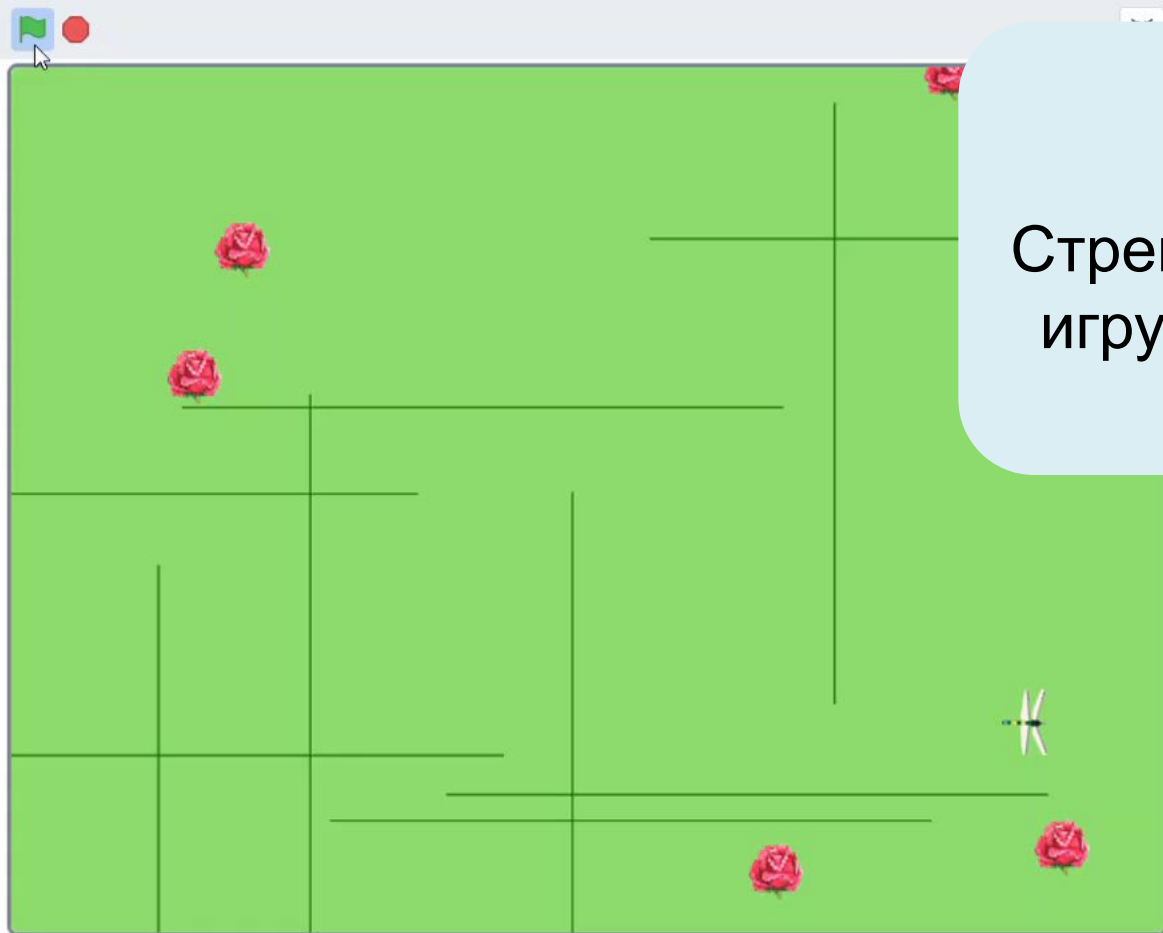
- **Когда флажок нажат**
- **Спрятать спрайт**
- **Ждать 2 секунды**
- **Показать спрайт**
- **Перейти им на случайное положение**

Сверим результат задания



Добавим такой же **ЦИКЛ**, что и цветам:
перемещать стрекозу в **случайное положение**,
пока она не касается забора





Запускаем!
Стрекоза теперь начинает
игру в случайном месте!





Этап 3.

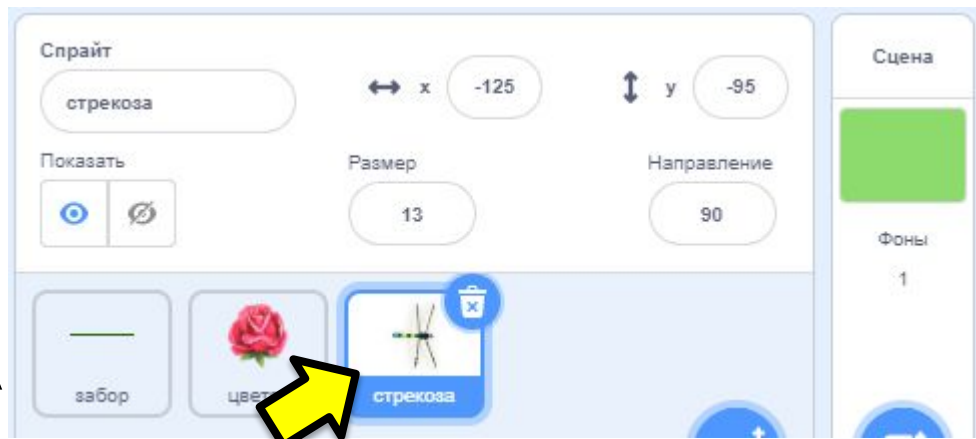
**Запрограммируем стрекозе
управление с помощью клавиш**

**На стрелки вперед и назад она
перемещается, на стрелки вправо
и влево – поворачивается вокруг
своей оси**

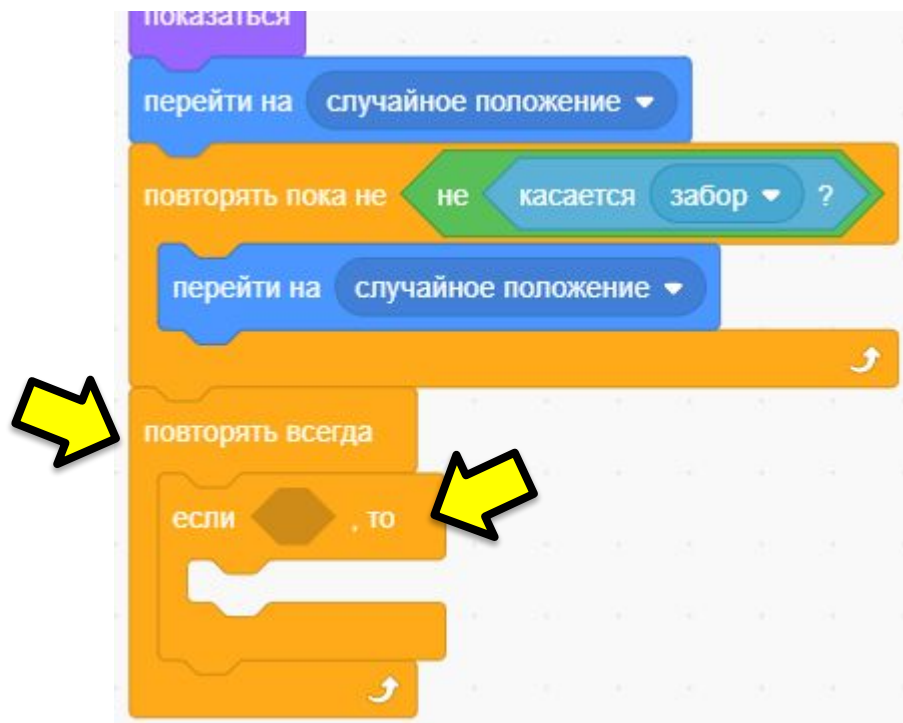
При написании программы всегда проверяйте, какому **СПРАЙТУ** вы добавляете команды.



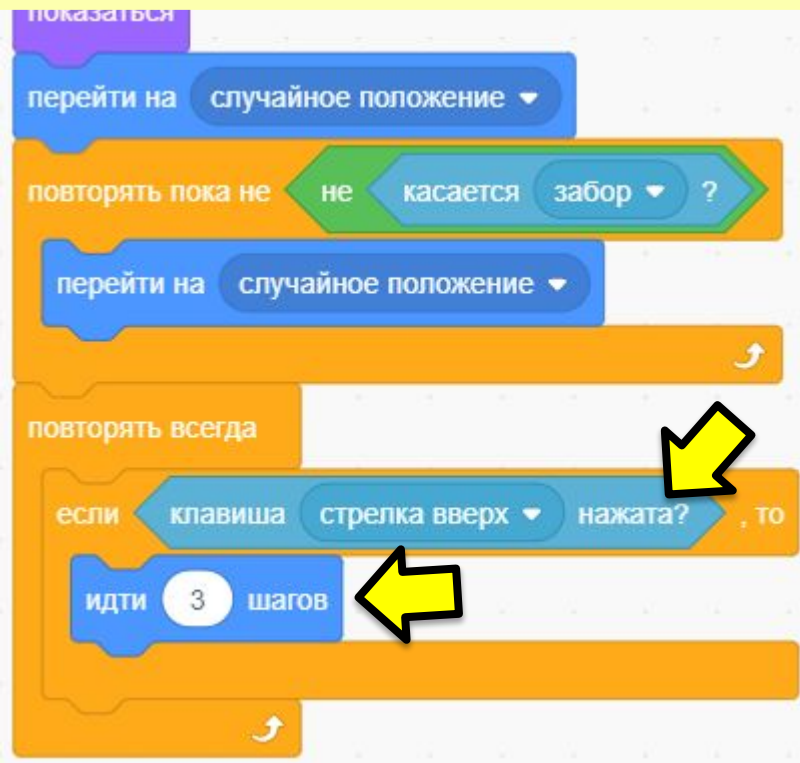
Проверьте, что здесь выделена **СТРЕКОЗА**



Добавим **БЕСКОНЕЧНЫЙ ЦИКЛ**, а в него –
команду **ЕСЛИ ТО**

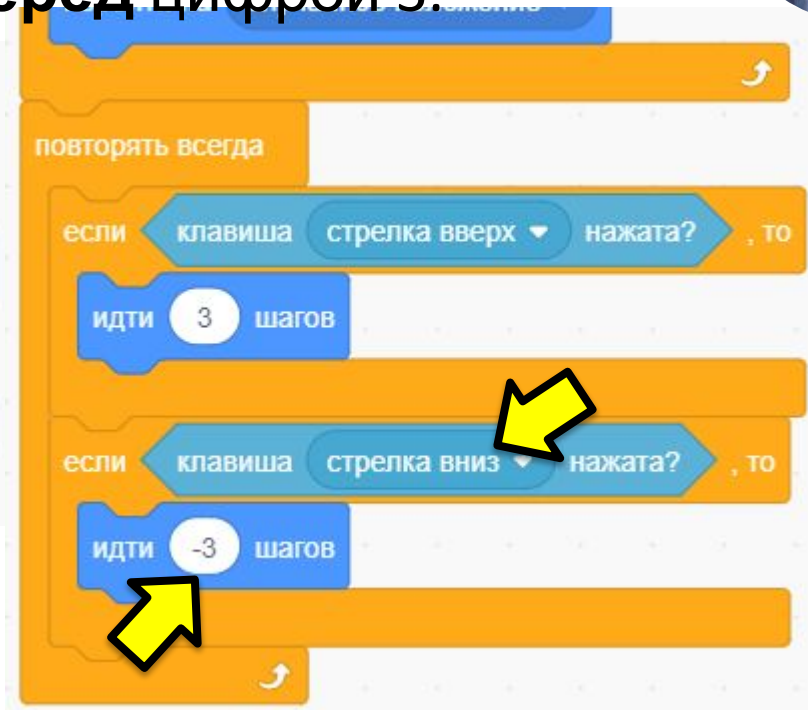
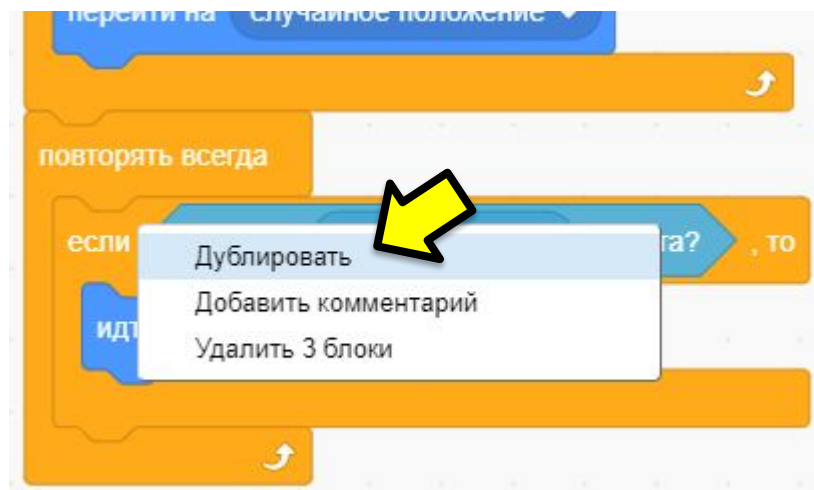


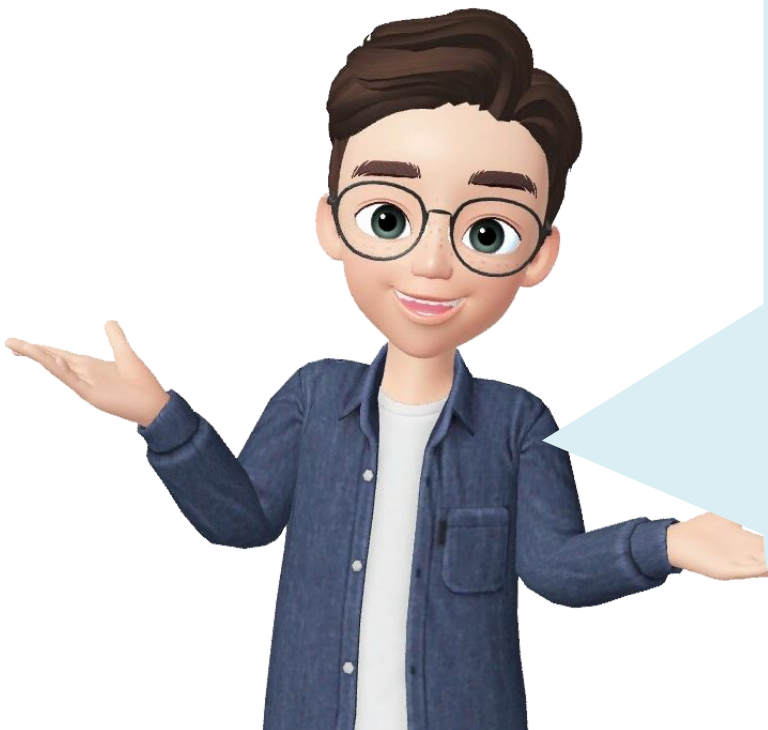
В условии проверяем, нажата ли клавиша вверх.
Если да – перемещаем стрекозу на 3 шага вперед.



Дублируем условие и вставляем его в конец цикла.

В новом условии меняем клавишу на стрелку вниз и ставим знак минус перед цифрой 3.



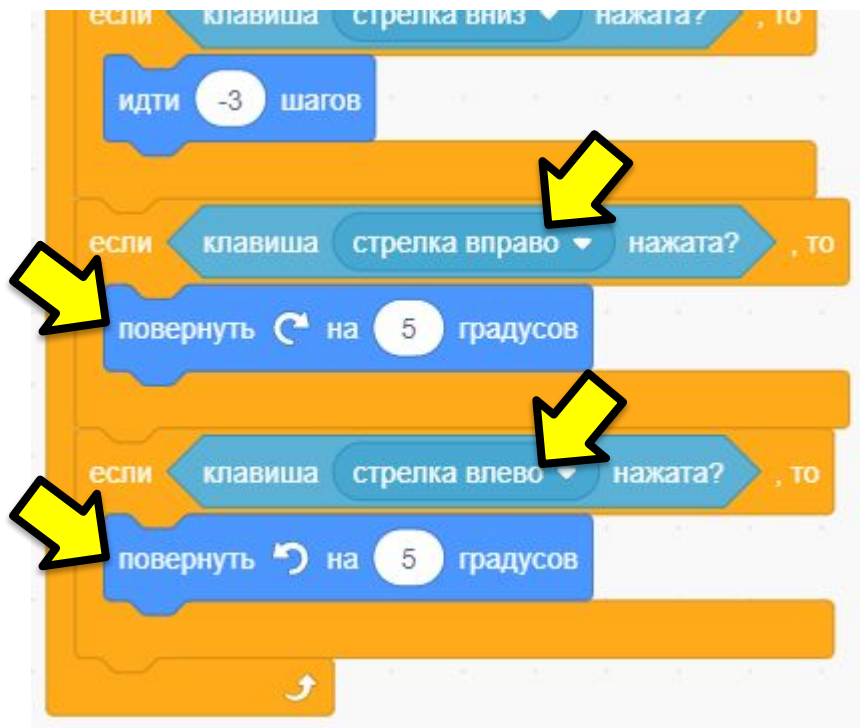


**Самостоятельно
запрограммируйте управление
на стрелки влево и вправо!**

ЕСЛИ нажата клавиша влево,
ТО нужно повернуть ПО
ЧАСОВОЙ СТРЕЛКЕ на 5
градусов.

ЕСЛИ нажата клавиша вправо –
повернуть на 5 градусов
ПРОТИВ ЧАСОВОЙ СТРЕЛКИ.

Сверим итог задания





Запускаем!
Теперь стрекоза
управляется на нажатие
стрелок!



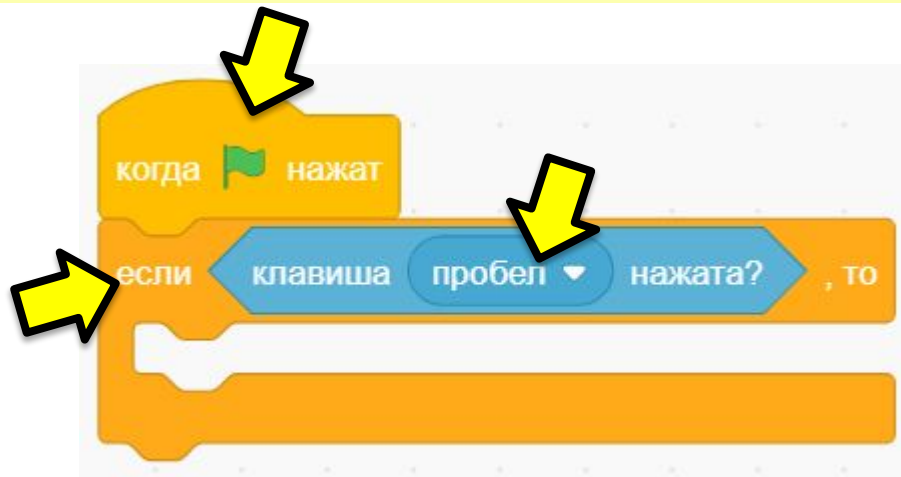


Этап 4.

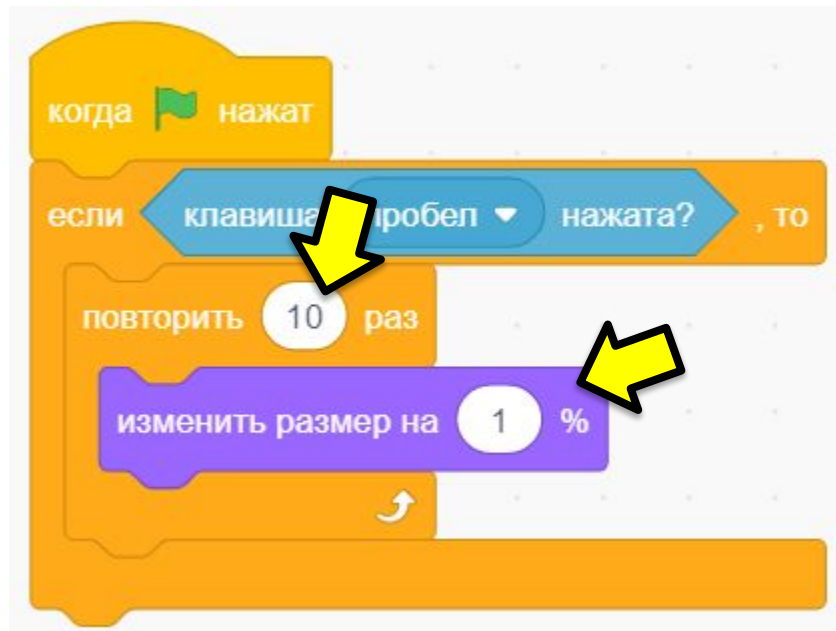
Запрограммируем **прыжок**
стрекозы

Прыжок будет происходить при нажатии на **пробел**, и при этом стрекоза **ненадолго увеличивается** в размере, чтобы было понятно, что она прыгнула – ведь мы видим прыжок сверху.

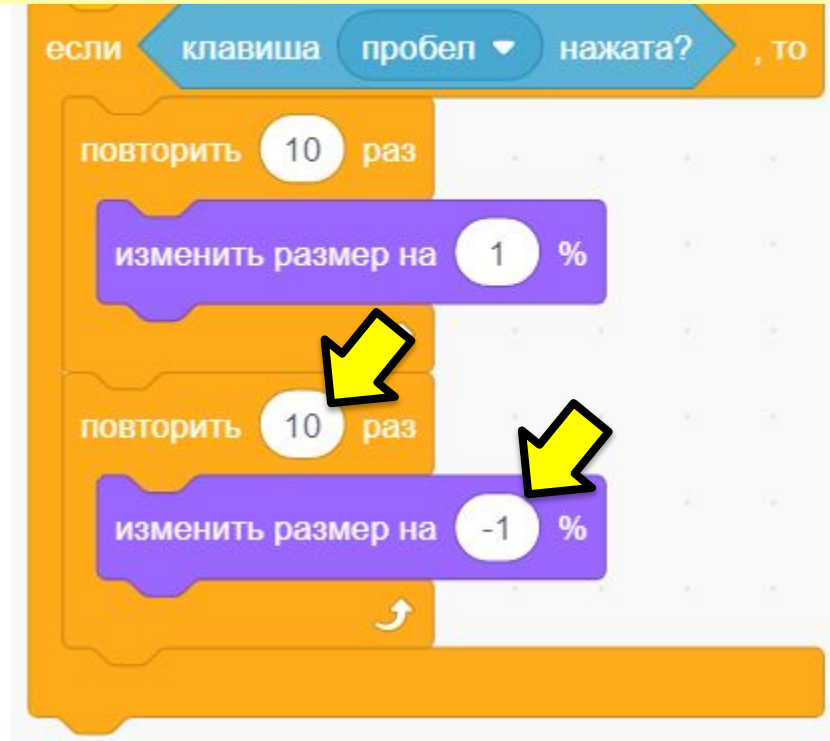
Добавим **НОВУЮ** программу для стрекозы. В ней напомним команду **ЕСЛИ ТО** и добавим условие, что **клавиша пробел нажата**



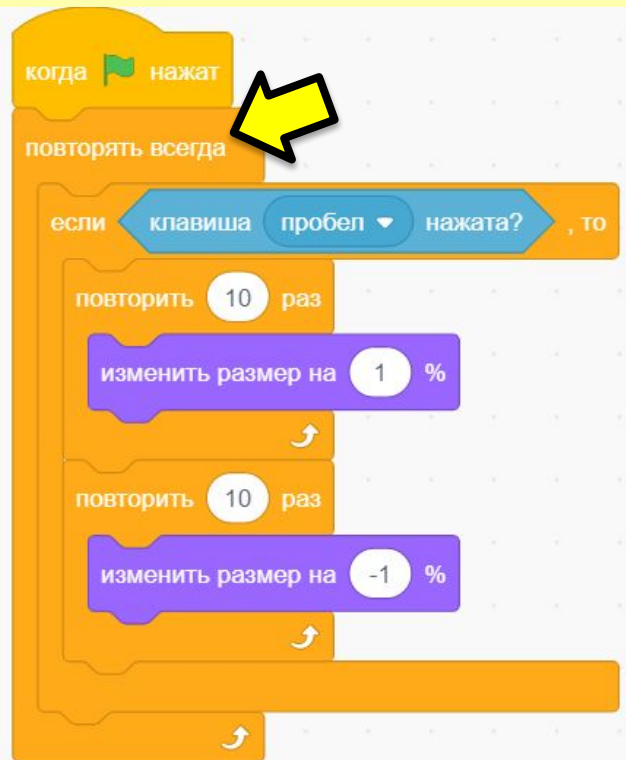
В условии добавим **ЦИКЛ** «Повторить 10 раз», где будем **изменять размер на 1 процент**



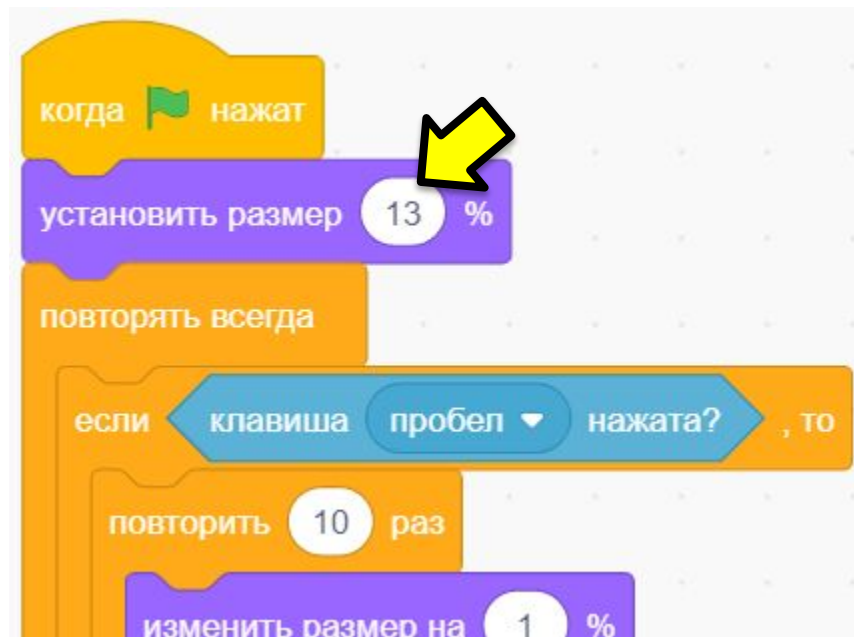
Добавим в условие еще один ЦИКЛ «Повторить 10 раз», где, наоборот, будем **уменьшать** размер на -1 процент

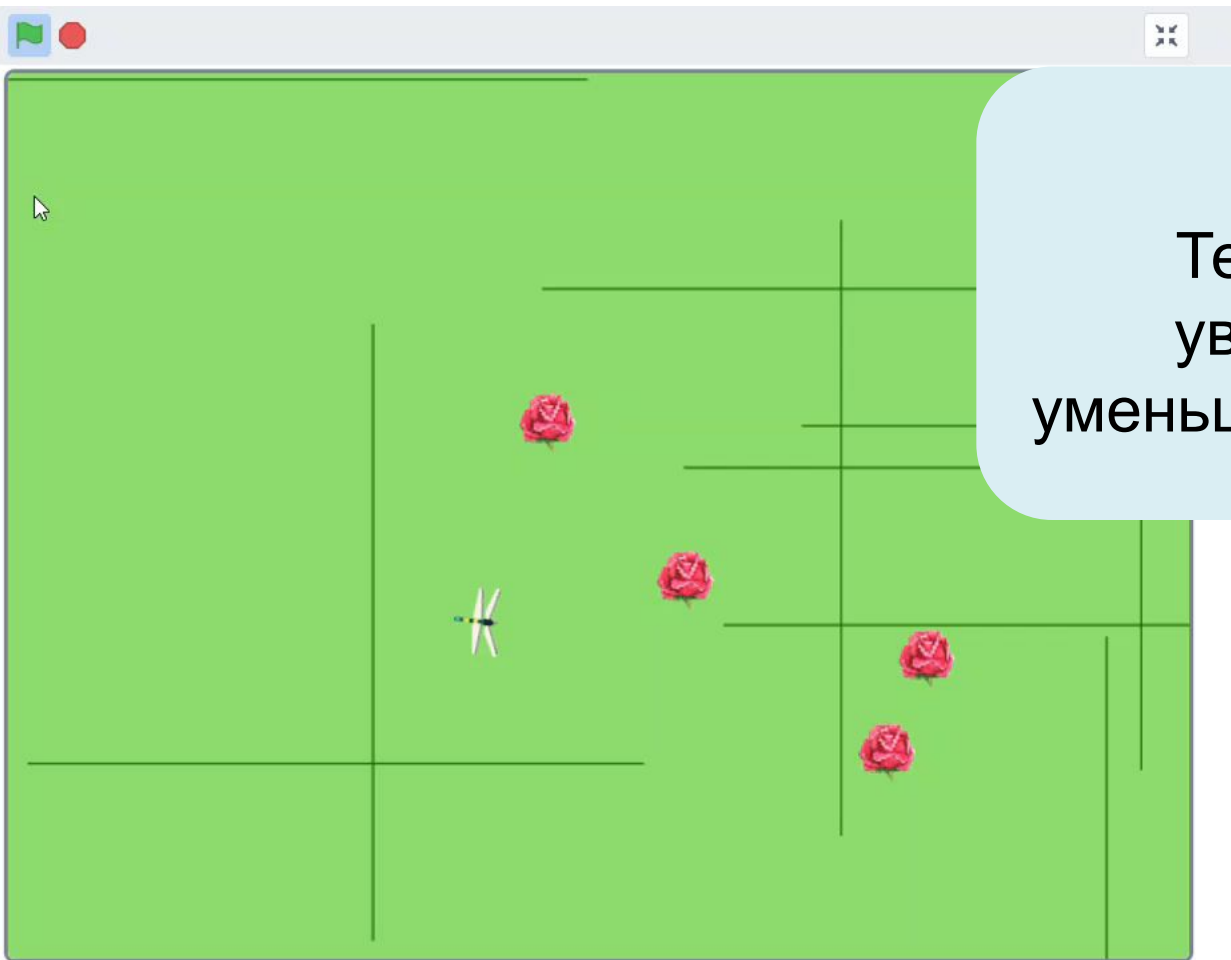


ВСЮ получившуюся программу поместим в **БЕСКОНЕЧНЫЙ ЦИКЛ**



И сразу после нажатия на флажок **установим размер в 13 процентов**





Запускаем!
Теперь стрекоза
увеличивается и
уменьшается при прыжке!





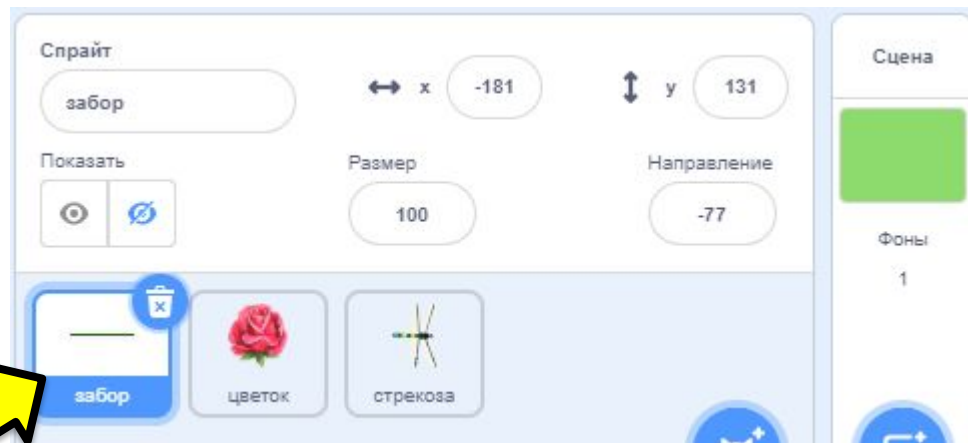
Этап 5.

Добавим **конец игры** – в случае, если стрекоза коснется забора!

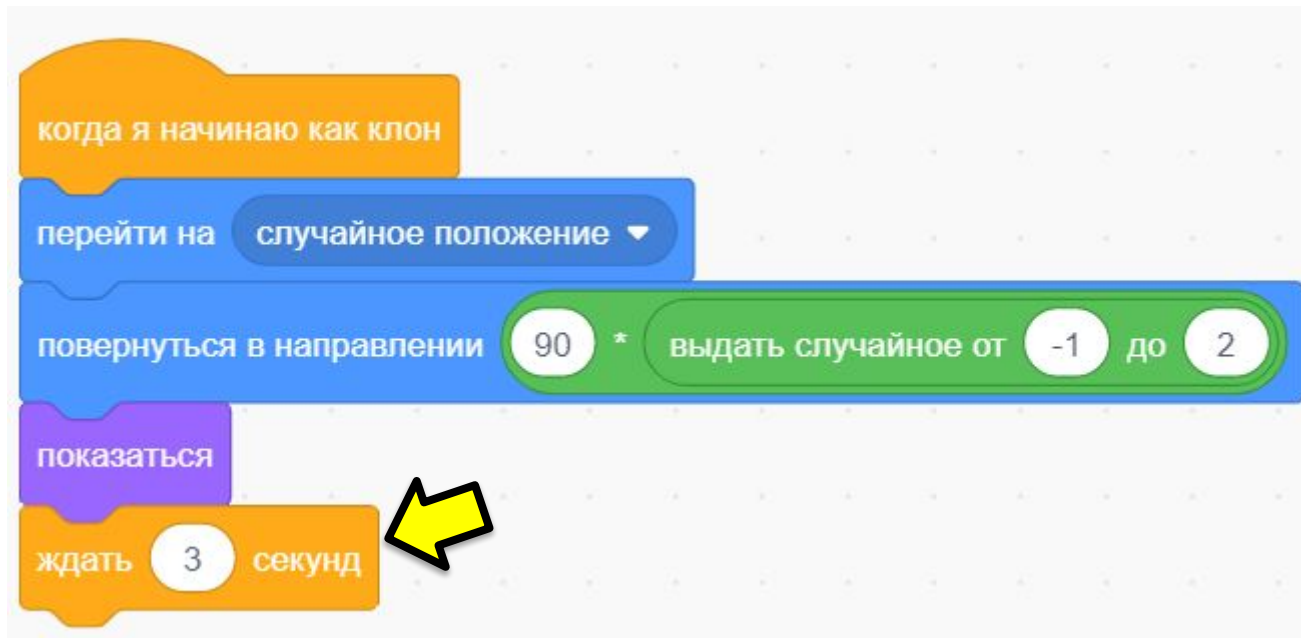
При написании программы всегда проверяйте, какому **СПРАЙТУ** вы добавляете команды.



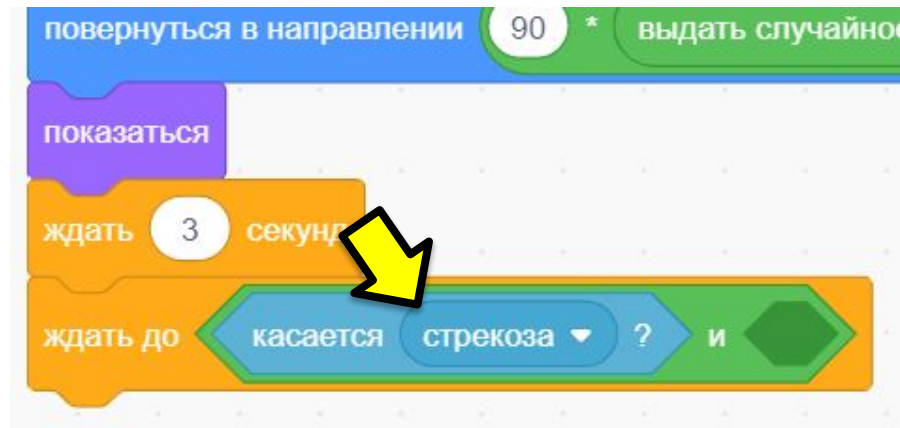
Проверьте, что здесь выделен **ЗАБОР**



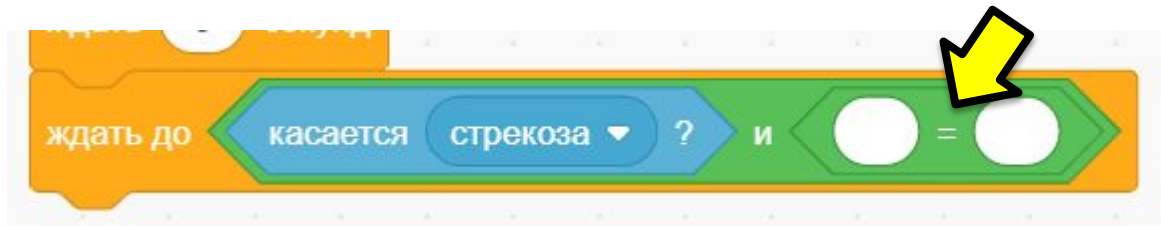
Допишем программу клона: зададим ему ждать 3 секунды



Конец игры наступает, когда **выполняются два условия. Первое** – что забора касается стрекоза.



А **второе** условие – что стрекоза в этот момент **не прыгает**. Это можно проверить через **ее размер**. Добавляем **знак равно** и **слева** от него – команду «фон# от Сцена» из категории Сенсоры. Затем **вместо** Сцены выбираем Стрекоза.

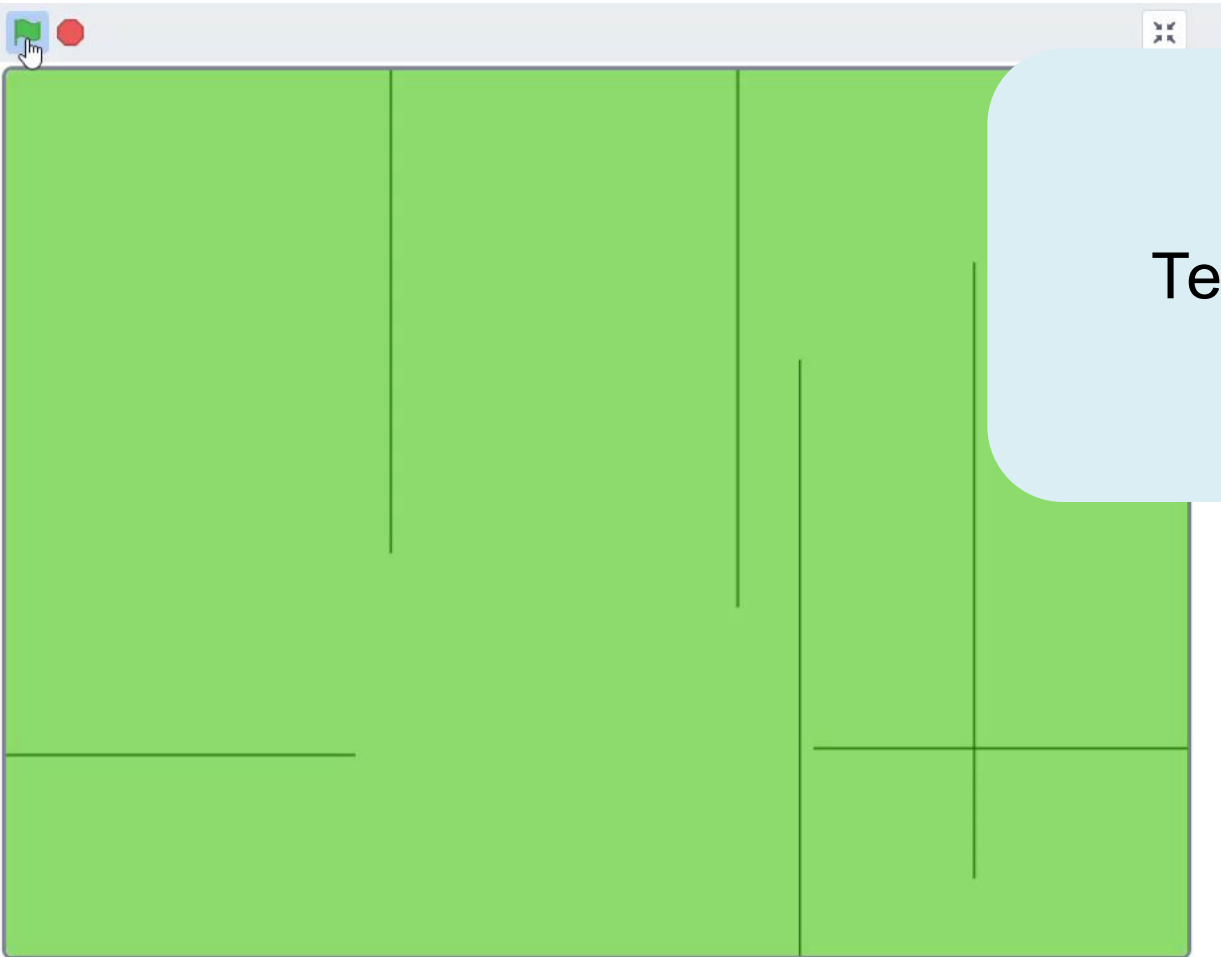


Затем вместо «положение x» выбираем «размер».



В программе стрекозы мы установили ее **размер на 13 процентов, если она не прыгает**. Это и будет вторым условием окончания игры.





Запускаем!
Теперь у игры есть
окончание!



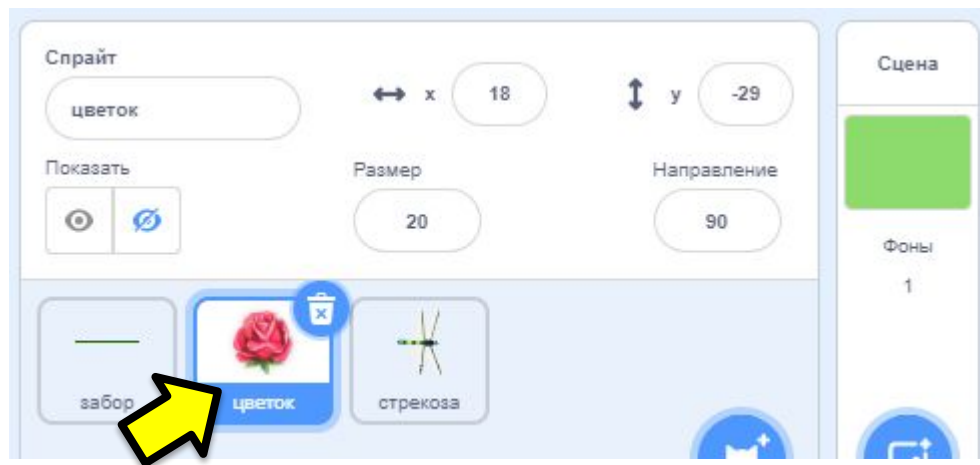


Этап 6. (дополнительно)
Добавим переменную для подсчёта
цветов и их анимацию!

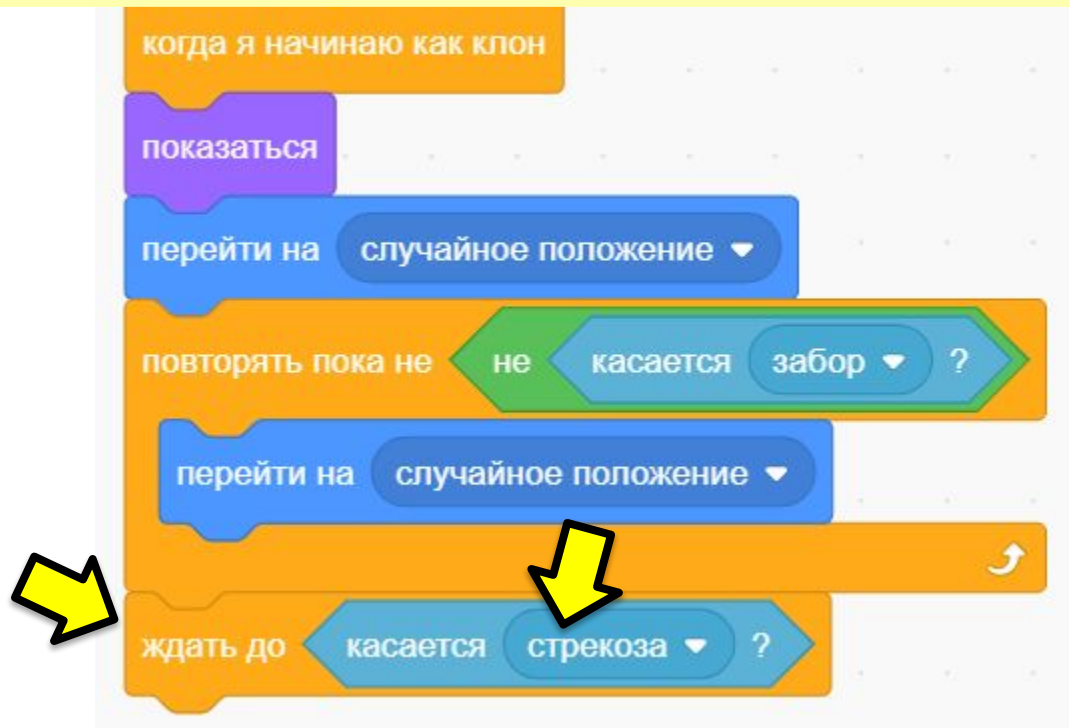
При написании программы всегда проверяйте, какому **СПРАЙТУ** вы добавляете команды.



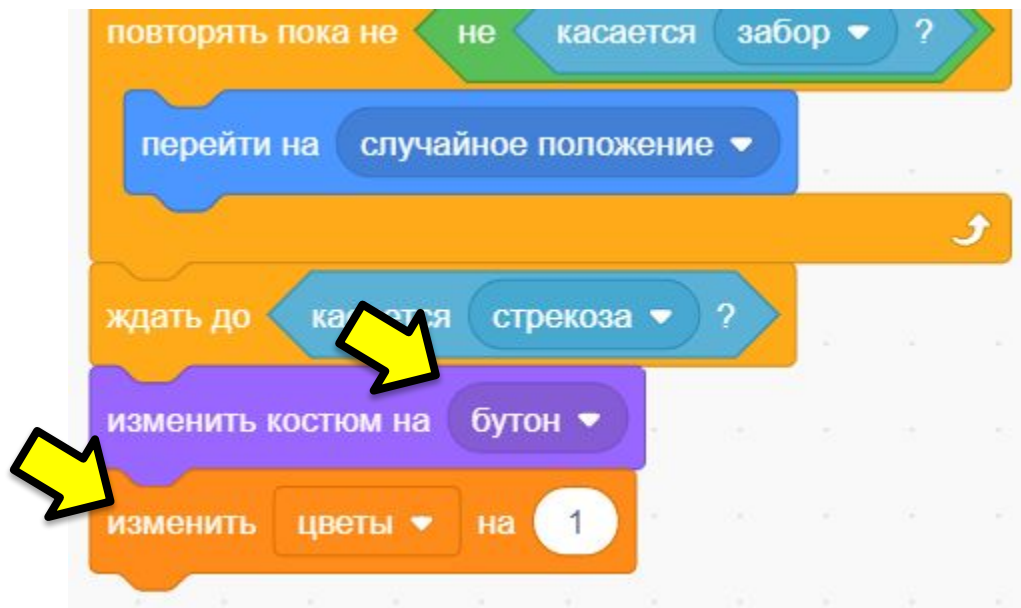
Проверьте, что здесь выделен **ЦВЕТОК**

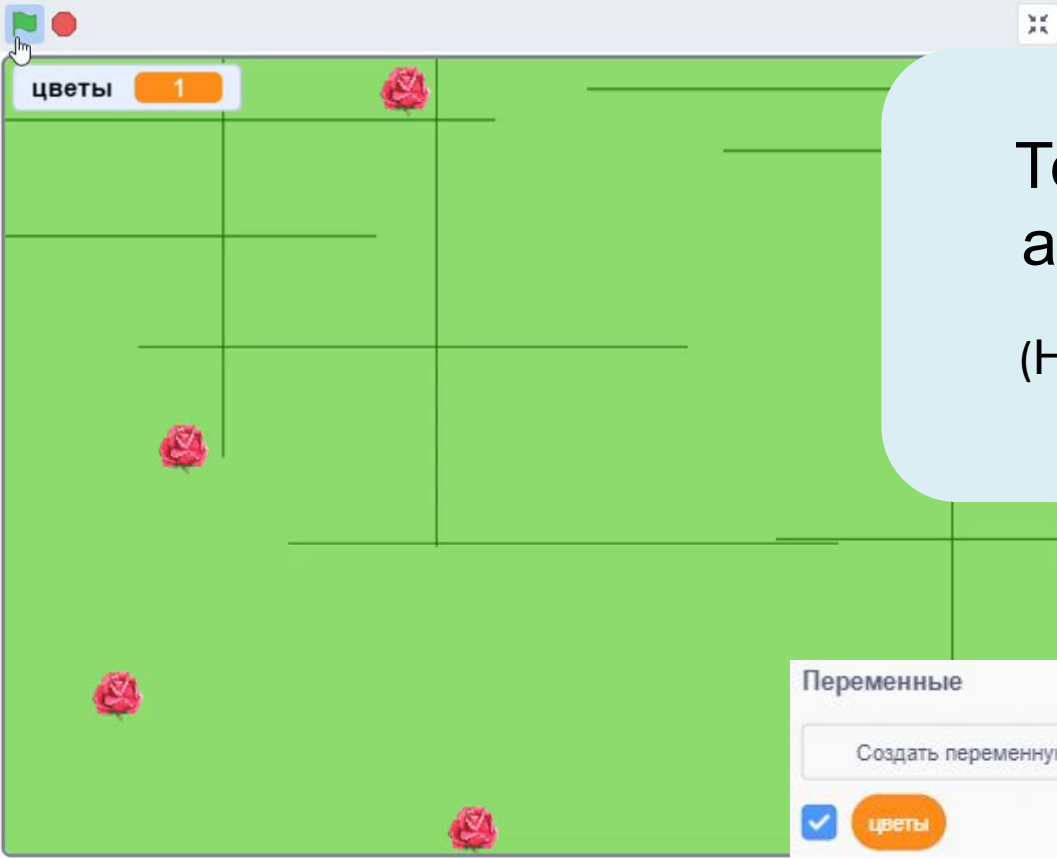


И подсчет, и анимацию мы будем добавлять после того, как цветок коснется стрекозы – поэтому ждем этого момента



После касания стрекозы меняем костюм на **букон** и добавляем одно очко к подсчету цветов





Запускаем!
Теперь у цветов есть
анимация и счетчик!

(Не забываем включить его
видимость в категории
Переменные!)



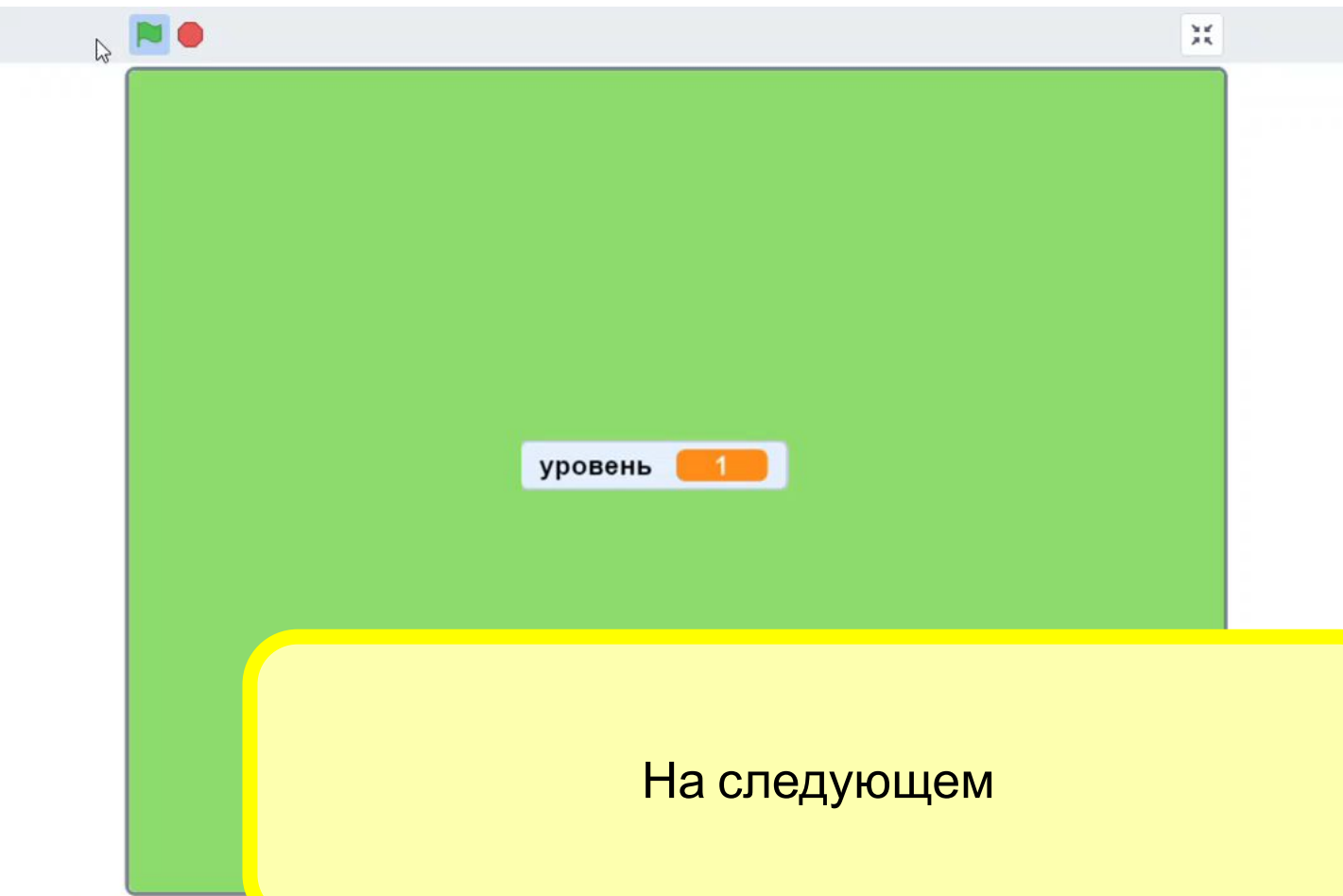
Вы молодцы!!! Расскажите:

- Какую игру мы создали сегодня?
- Какое управление было у стрекозы?
- Как мы проверяли, что спрайты не пересекались друг с другом?



Покажите родителям и
расскажите, как
работают ваши
программы!





На следующем



До встречи
на следующем
занятии!

