Для наглядности анимаций нам необходим следующий модификатор, который мы вешаем на элемент:

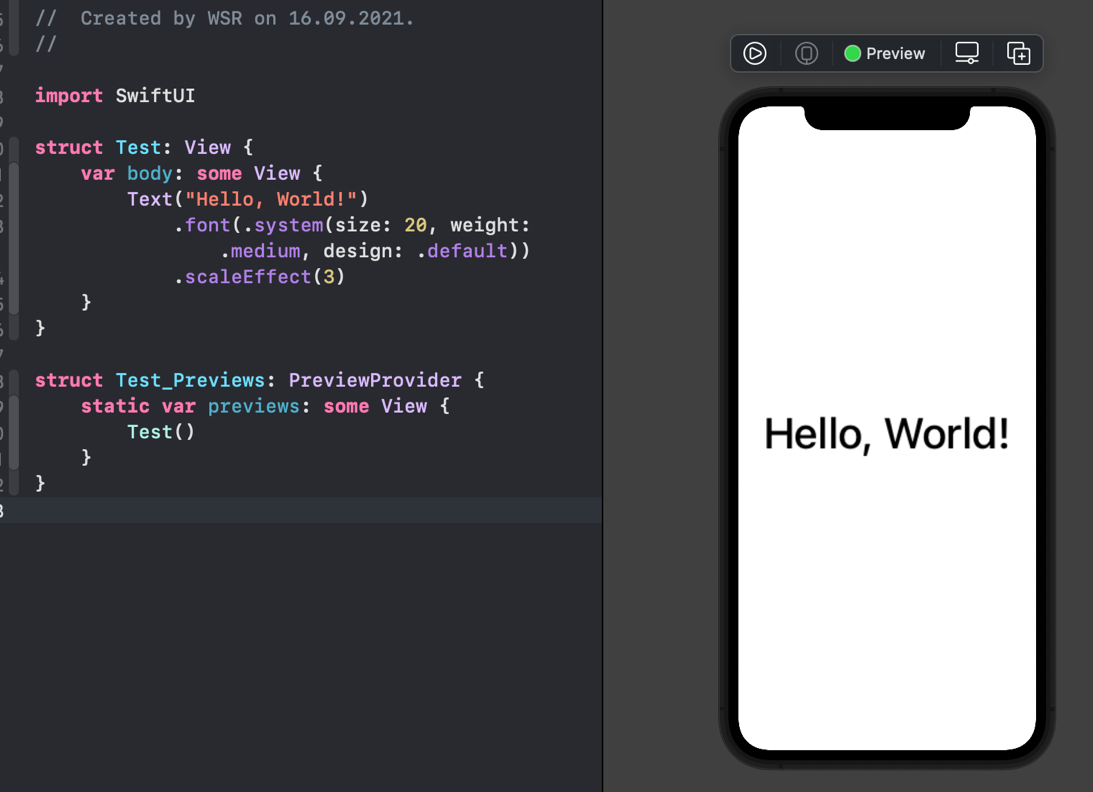
.scaleEffect-масштабирует визуальный вывод на заданную величину относительно точки привязки( (achor:) - устанавливает эту точку, изначально точка привязки-центр)

, немного отклоняются, а затем возвращаются в норму.(“пружинная анимация”)

Изображение выглядит как текст

Автоматически созданное описание







Как использовать модификатор . animation?

Чтобы использовать . animation, поместите его после любых других модификаторов для ваших представлений, сообщите ему, какую анимацию вы хотите, а также убедитесь, что вы прикрепили его к определенному значению, чтобы анимация срабатывала только при изменении этого конкретного значения(value:).

Пример: .animation(.linear**, value: animation**)

Какие анимации мы сегодня разберем?

.linear – линейная

Пример: .animation(.**linear**)

Вы можете указать точное значение за которое должна выполнится анимация:

(duration:1)

Пример: .animation(.linear(**duration**:1) , value: animation)

.spring - представляют собой анимации, которые перемещаются в свою целевую точку

Пример: .animation(.**spring(),** value: animation)

У .spring есть некоторые параметры, которые можно указать:

1. (**response**:) - Жесткость пружины, определяемая как приблизительная продолжительность в секундах. Нулевое значение требует бесконечно жесткой пружины, подходящей для управления интерактивными анимациями.
2. (**dampingFraction**:) - Величина сопротивления, применяемая к анимируемому значению, как часть оценки величины, необходимой для создания критического демпфирования.

.**repeatForever**() – анимацию нужно выполнять вечно

Пример: .animation(.linear.repeatForever(), value: animation)

.repeatForever(**autoreverses**: false) – анимацию нужно выполнять вечно без автореверса

Пример: .animation(.linear.repeatForever(**autoreverses**: false), value: animation)

.**delay**(2) – задержка перед выполнением(повторным выполнением, если присутствует repeatForever) анимации

Пример: .animation(.linear.**delay**(1), value: animation)