

ШУМ ПЕРЛИНА КАК ИНСТРУМЕНТ МОУШН-ДИЗАЙНЕРА

Современные анимационные проекты часто требуют включения элементов случайности для построения различных видов движения – колебания объектов, дрожания камеры, генерации и движения частиц, хаотичного изменения параметров эффектов и т.д. Покадровое ручное создание и редактирование подобных эффектов представляет собой весьма трудоемкий процесс. Автоматизировать эти процессы позволяет так называемая процедурная анимация, в основе которой для генерации эффектов движения в режиме реального времени лежат математические алгоритмы, законы движения и пр. Одним из часто используемых алгоритмов процедурной генерации псевдослучайных данных является шум Перлина.

Шум Перлина формирует сглаженные, взаимосвязанные колебания, обеспечивая непрерывность функции и плавные переходы между значениями, и широко применяется как в двухмерной, так и в трехмерной графике [1]. Преимущественно он используется при создании текстур и генерации поверхностей и рельефов, позволяя увеличивать графическую сложность поверхности геометрических объектов. Другой сферой применения функции шума Перлина является получение компьютерных спецэффектов природных явлений, таких как эффекты дыма, тумана, ветра, моделирование облаков, а также ряда эффектов движения непрерывного характера.

Обсудим некоторые аспекты использования функций, основанных на принципах шума Перлина, в одной из распространенных на текущий момент в медиаиндустрии программ для моушн-дизайна – Adobe After Effects.

В Adobe After Effects шум Перлина лежит в основе функций `noise` и `wiggle`. Функция `noise` принимает только один числовой аргумент – текущее время композиции в секундах (или любой другой параметр, изменяющейся во времени), и возвращает не превосходящее единицы по абсолютной величине псевдослучайное число. Ключевая особенность функции `noise` заключается в том, что она возвращает не независимые случайные числа, а взаимосвязанные значения — близкие входные аргументы дают близкие выходные результаты. Благодаря этому удастся сформировать непрерывную траекторию изменения параметра без резких скачков между кадрами. Таким образом, исполь-

зую функцию `noise`, можно создавать плавную анимацию, имитирующую естественные процессы, такие как мерцание света, покачивание объектов, легкое дрожание камеры и т.д.

Функция `wiggle`, также обеспечивающая непрерывные изменения значений без резких скачков, представляет собой более гибкий инструмент за счет наличия как обязательных, так и необязательных аргументов, позволяющих более точно управлять характером колебаний. В частности, с ее помощью можно дополнительно задать интенсивность движения.

Функции, основанные на шуме Перлина, применяются в широком спектре задач, где необходимо создание эффекта тряски или плавного движения объекта. Например, функция `noise` идеально подходит для имитации колебания стрелки компаса (рис. 1), при этом для определения диапазона отклонения на a градусов при вызове функции необходимо добавить коэффициент: $a * \text{noise}(\text{time})$.

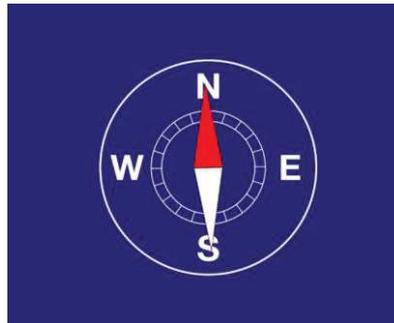


Рисунок 1 – Кадр анимации компаса

Другим примером может служить имитация съемки ручной камерой («дрожания» кадра). Этот эффект довольно часто применяется в киноиндустрии при решении разных задач – для создания ощущения документальной съемки или для имитации тряски, вызванной естественными причинами (взрыв или землетрясение) и т.д. В этом случае следует воспользоваться функцией `wiggle`. При этом, для достижения характерной для проекторов прошлого «тряски» изображения функцию `wiggle` необходимо применить к положению всех объектов, присутствующих в кадре.

Создание колебаний нередко применяется и в векторной анимации для «оживления» статичных изображений путем создания естественного движения. Одним из примеров подобного подхода является анимация полета космической ракеты (рис. 2), точнее имитации «тряски» ракеты при высокой скорости передвижения. При этом функция используется для двух параметров – Положение и Поворот.

Следует также заметить, что не существует универсального алгоритма для точного подбора значений аргументов функции `wiggle`,

их выбор обусловлен художественными и композиционными особенностями конкретной анимации.



Рисунок 2 – Кадр анимации ракеты

Та же анимация может быть реализована с помощью функции `poise`. Однако, для достижения реалистичного эффекта тряски данную функцию требуется доработать, создав для нее параметры амплитуды и частоты вручную. В противном случае анимация похожа не на тряску, а на покачивание, свойственное движению маятника. Увеличение скорости анимации также не способствует получению желаемого эффекта.

Еще одним преимуществом функции `wiggle` является возможность ее применения к анимации параметров, не связанных с перемещением объектов в пространстве. Так, например, можно анимировать непрозрачность объекта для создания эффекта мигания. Возможности использования данной функции не ограничены, и она находит свое применение в любом виде анимации, где необходимо создание естественного и плавного изменения состояния какого-либо объекта. Используя ее, пользователь может использовать для имитации естественных процессов – покачивания, дрожания, мерцания и других реалистичных эффектов.

Несмотря на то, что функция `wiggle` является более универсальной функцией по сравнению с `poise` для создания плавных колебаний за счет простоты использования и большого количества настраиваемых параметров, можно с уверенностью сказать, что знание принципов работы каждой функции позволяет осознанно выбирать инструменты и создавать более точные и гибко настраиваемые анимации. И, следовательно, позволяет разработчикам оптимизировать свои рабочие процессы и более эффективно справляться с поставленными задачами.

ЛИТЕРАТУРА

1. Слеповичев С.О. Шум Перлина как способ получения компьютерных спецэффектов природных явлений. // Инновационное развитие. 2017. №2 (7). С. 32-33.