

Лабораторная работа № 3. Проект – разработка простого AR-Приложения для Android-устройства (смартфон, планшет и пр.). Создание в графическом редакторе Unity 3D сцены дополненной реальности: визуализация управляемой анимации 3D-Модели с использованием кнопки плоского меню. **Часть 2. Создание кнопки плоского меню для организации управляемой анимации 3D-Модели, подготовленной в ЛР № 3, Часть 1.**

Работа по созданию простейшей анимации простейшего виртуального объекта 3D Модели была выполнена в первой части ЛР №3.

Предметом исследования в первой части ЛР №3 стали элементы меню окна Animation, которые позволяют вам создавать трансформации любой сложности на любом временном интервале.

На выходе ЛР №3, Часть 1 вами могли быть разработаны 3 Приложения ДР для:

- **Вращения** выбранной 3D-модели вокруг выбранной оси (в зависимости от модели);
- **Перемещения** выбранной 3D-модели по диагонали, например – из нижнего левого угла в верхний правый;
- **«Наезда»/«Удаления»** вращающейся модели на зрителя.

Одну (а может быть и все три, попытки такой реализации приветствуются!) из разработанных в рамках этого задания анимаций в ЛР №3, Часть 2 можно использовать для связывания с кнопкой плоского меню для создания с ее помощью управляемой анимации.

В данном описании вашему вниманию предлагается подробное перечисление шагов, которые необходимо выполнить для добавления в Проект кнопки плоского меню, с помощью которой может быть осуществлено управление простейшей анимацией (вращение вокруг вертикальной оси) 3D-модели греческой амфоры – **контент разработанной сцены.**

Итак, по завершении создания простейшей анимации в ЛР №3, Часть 1, вам предлагалось закрыть окно Animation и убедиться в том, что в Assets окна Project появились два новых Assets – Vase_Animation и Vase_Hydria – связанные с анимацией → анимация и контроллер анимации. Эти имена будут использоваться в дальнейшем при разработке элемента плоского меню - кнопки управления анимацией.

Возвращаемся в Проект для следующих модификаций – дополнения:

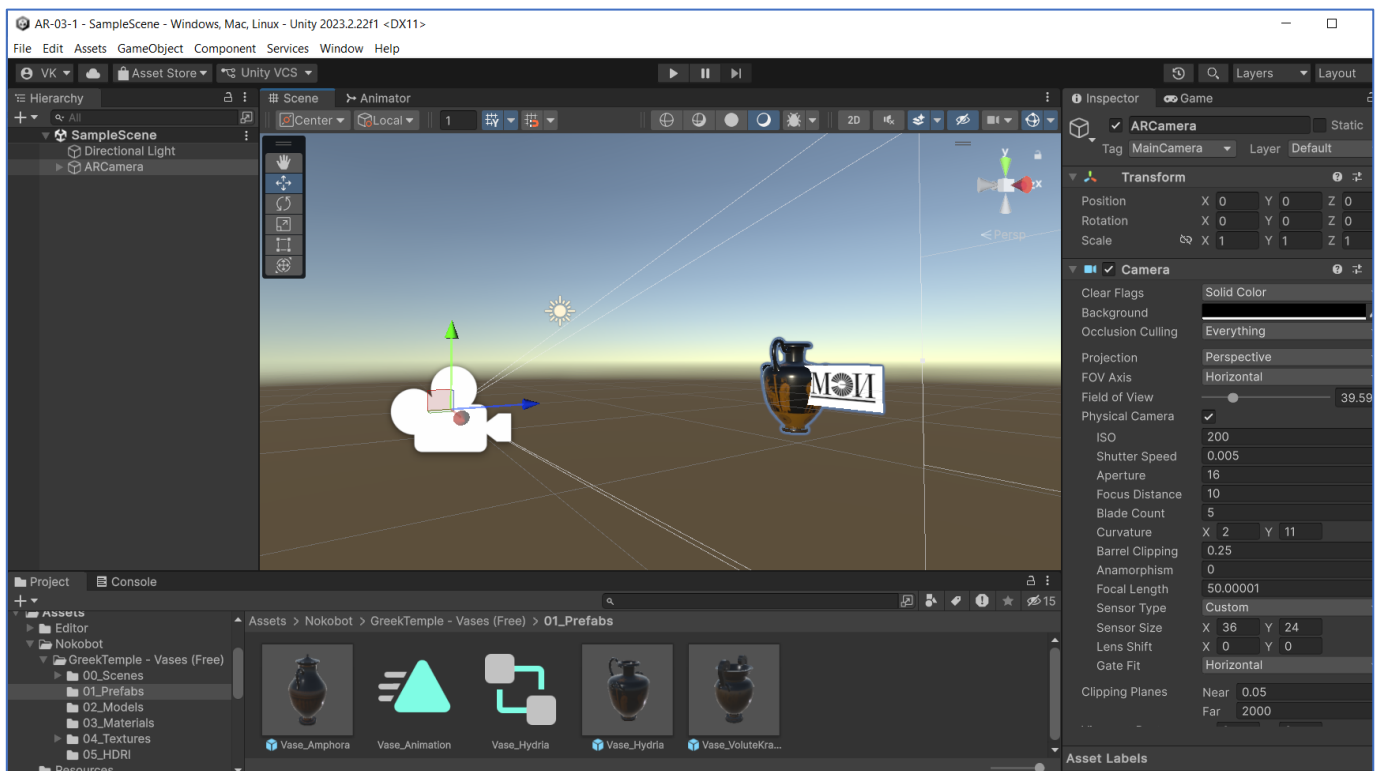
1. Добавление в плоское меню элемента управления – кнопка старта анимации

Вспомним: Задача ЛР №3 – разработка элементов виртуального интерфейса для работы в ДР, т.е. установление связей между поведением (**behaviour**) виртуальной **3D-модели** (у нас - «**анимация греческой вазы**») и состоянием элемента управления «**кнопка**».

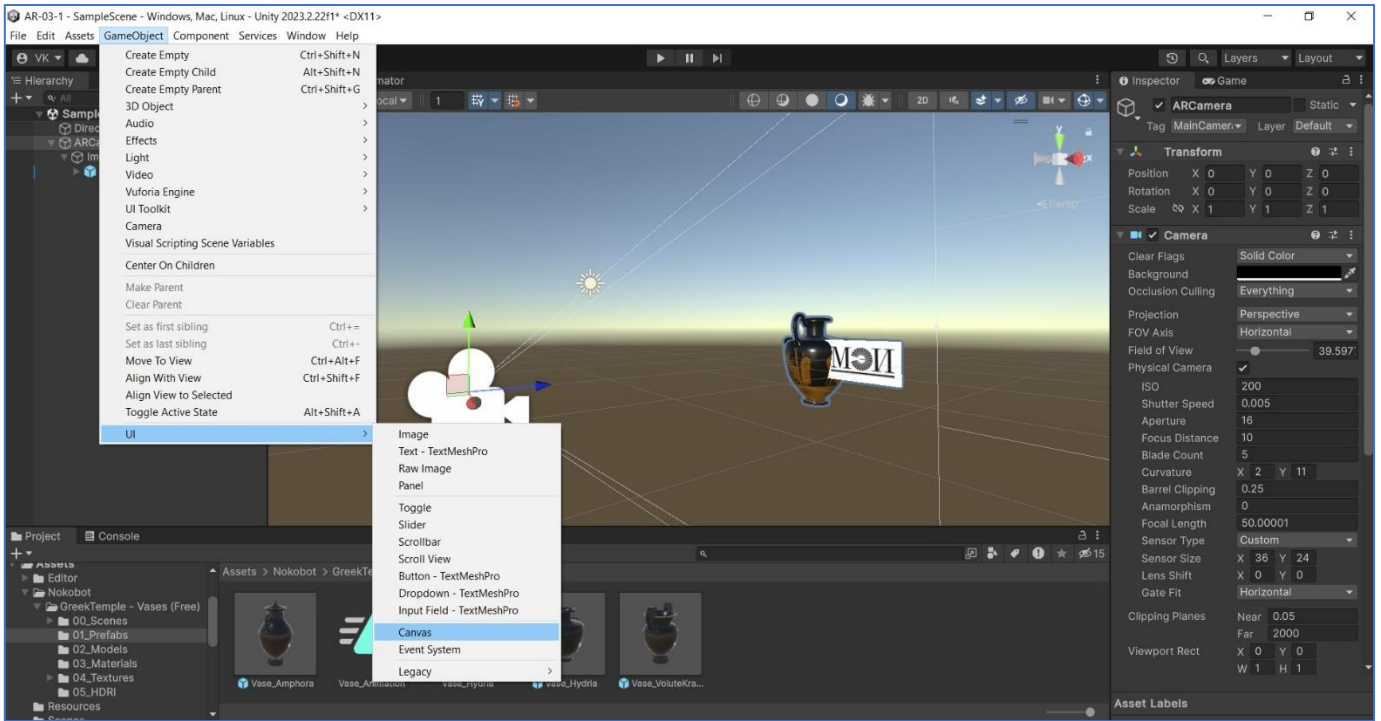
Для установления указанных связей в **Unity 3D** можно использовать скриптинг (программирование на языке **C#**) или **обойтись без программирования**, используя возможности контроллера анимаций.

В данном описании предлагается разработанную кнопку связывать с анимацией, используя возможности контроллера анимации, т.е. обойтись без программирования.

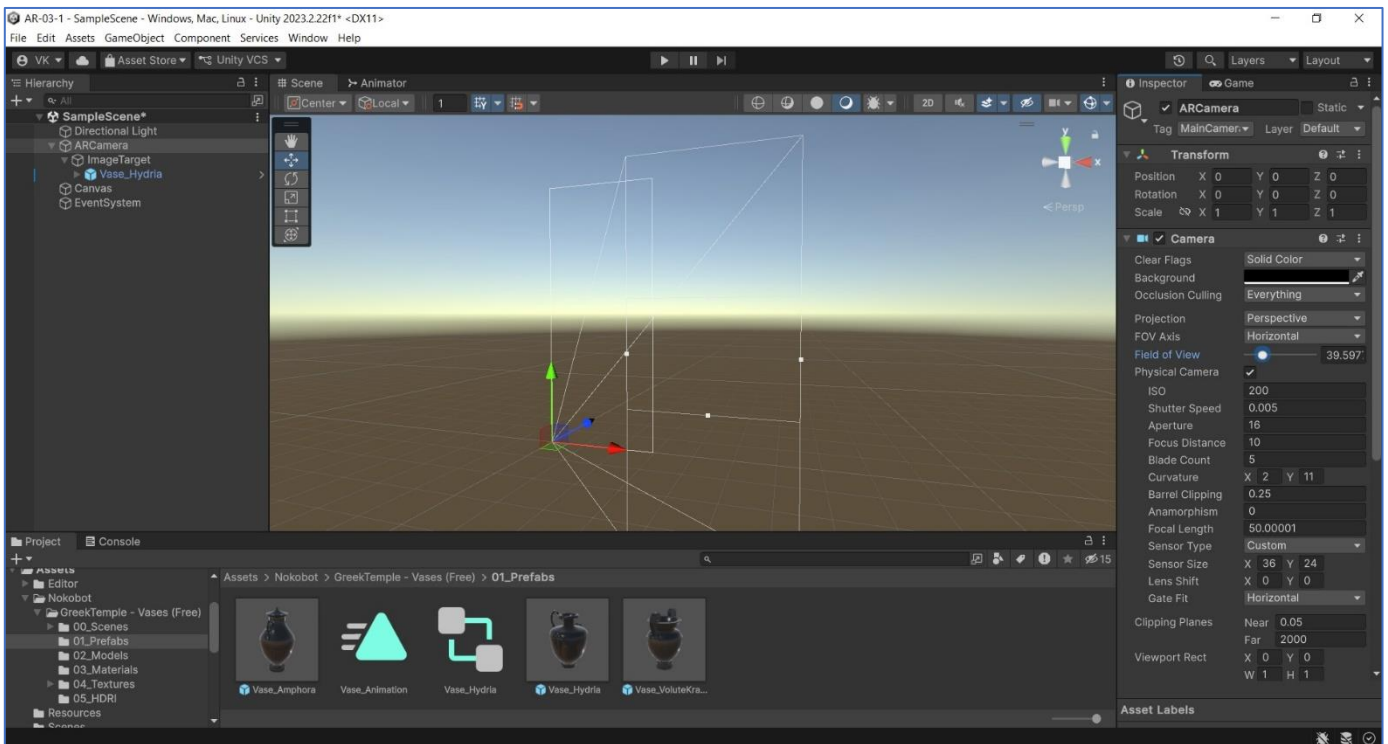
Входим в проект, в котором создана и сохранена одна анимация – вращение вазы (**модель Vase_Hydria в Иерархии**). Обратите внимание на корректность расположения объектов в **Иерархии**:



В GameObject → **UI** вызываем канву для формирования плоского меню:

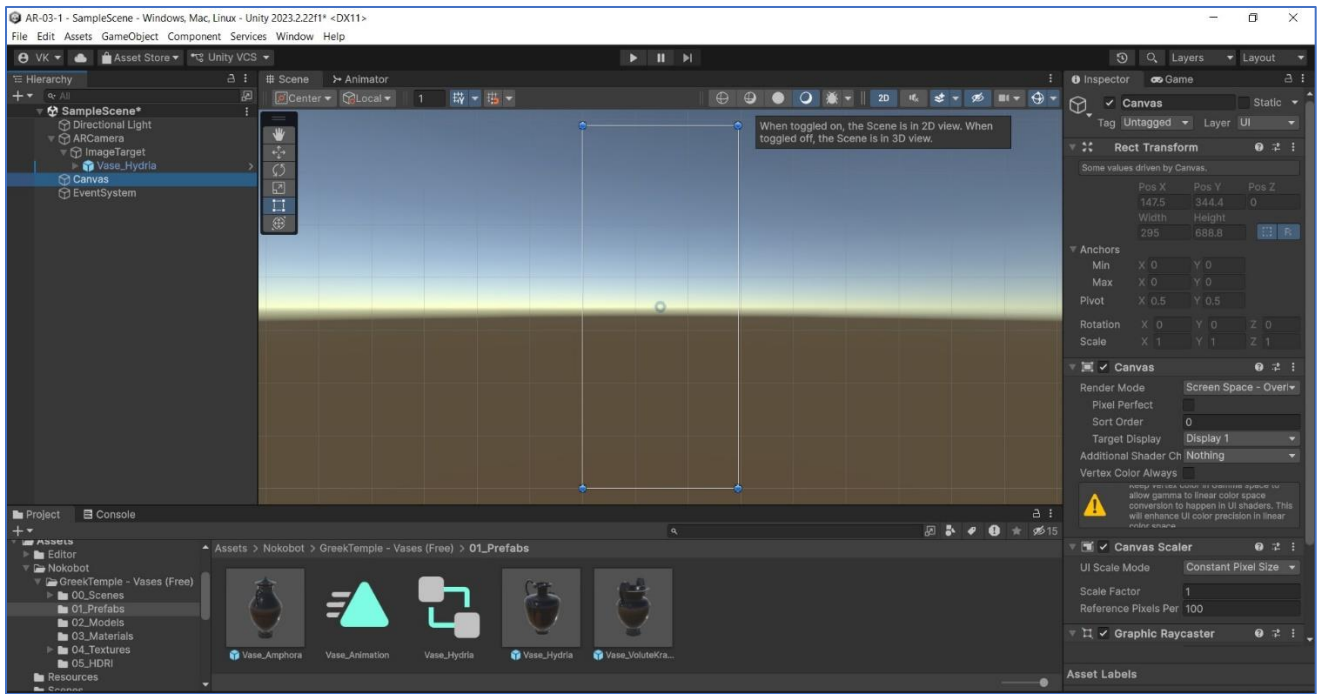


Переходим в режим работы в 2D-канве. Обратите внимание на ориентацию канвы.

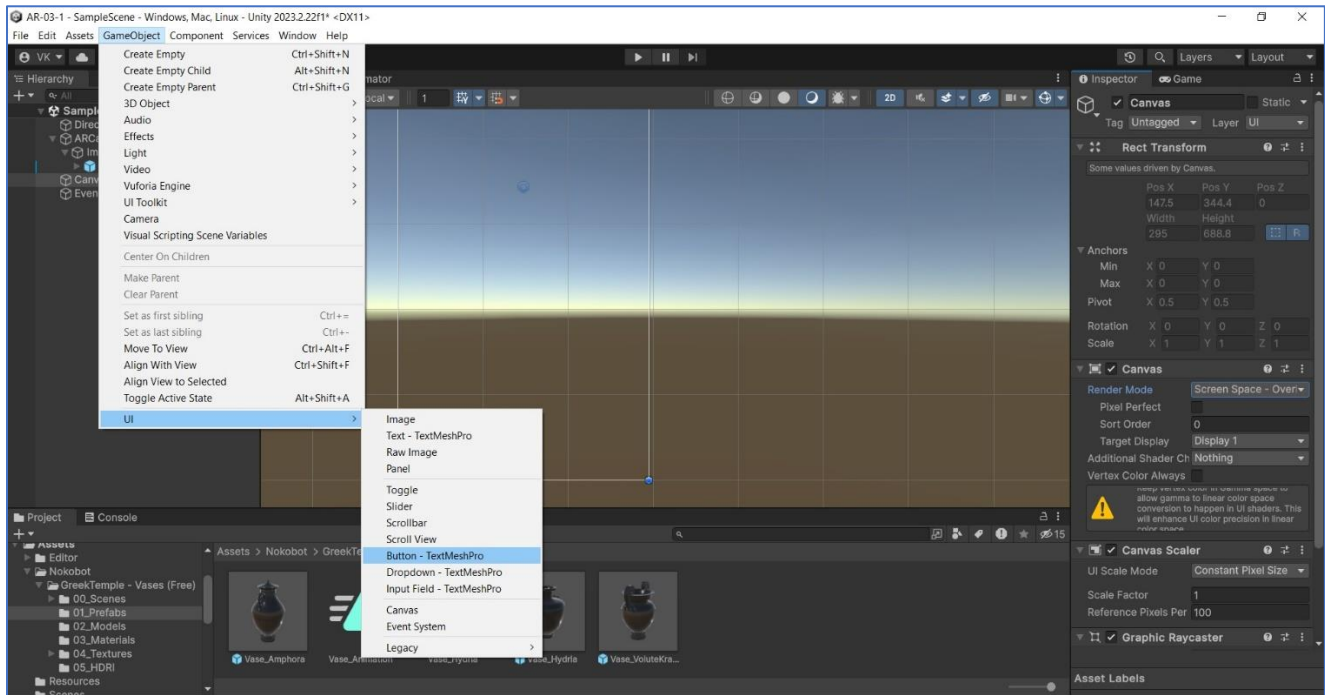


ВАЖНО!! Изменения в интерфейсе и в иерархии после перехода в 2D канву и манипуляций для совмещения канвы с экраном (юстировка – средняя кнопка - и перемещение – зажатая средняя кнопка).

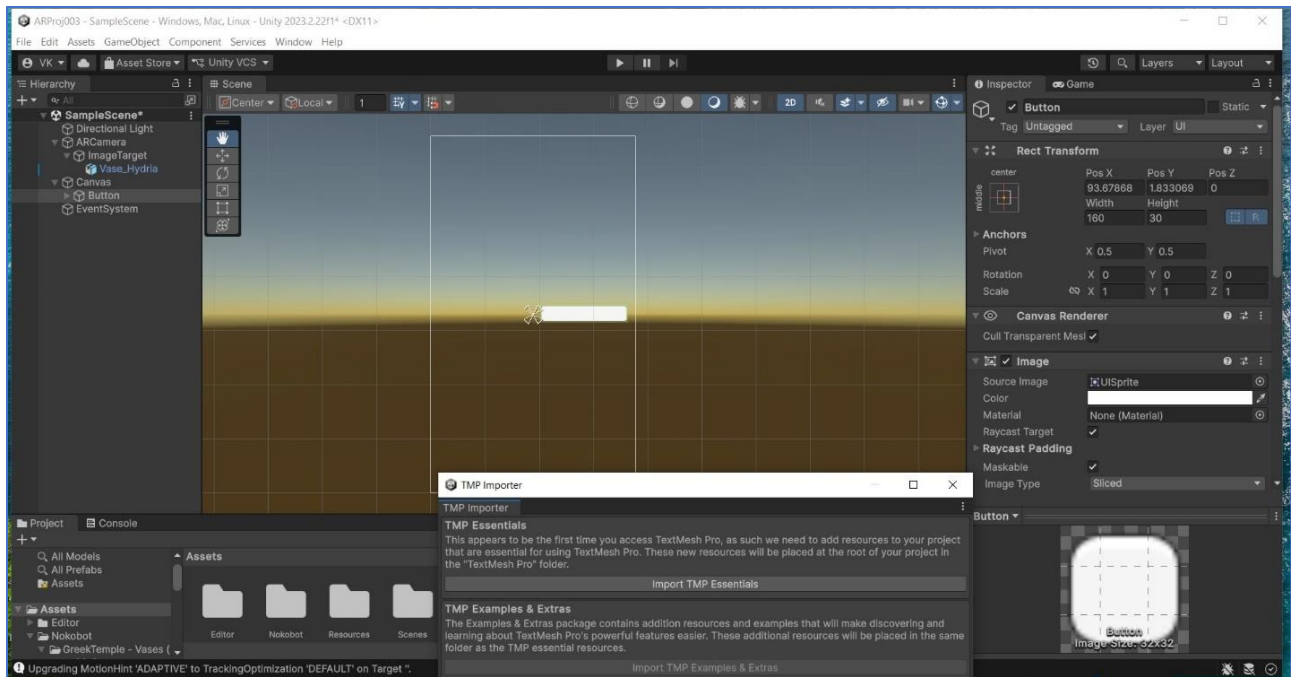
Через манипуляции в верхней линейке (**toolbar**) добиваемся ориентации канвы параллельно экрану (далее будем формировать т.н. **меню на стекле**):



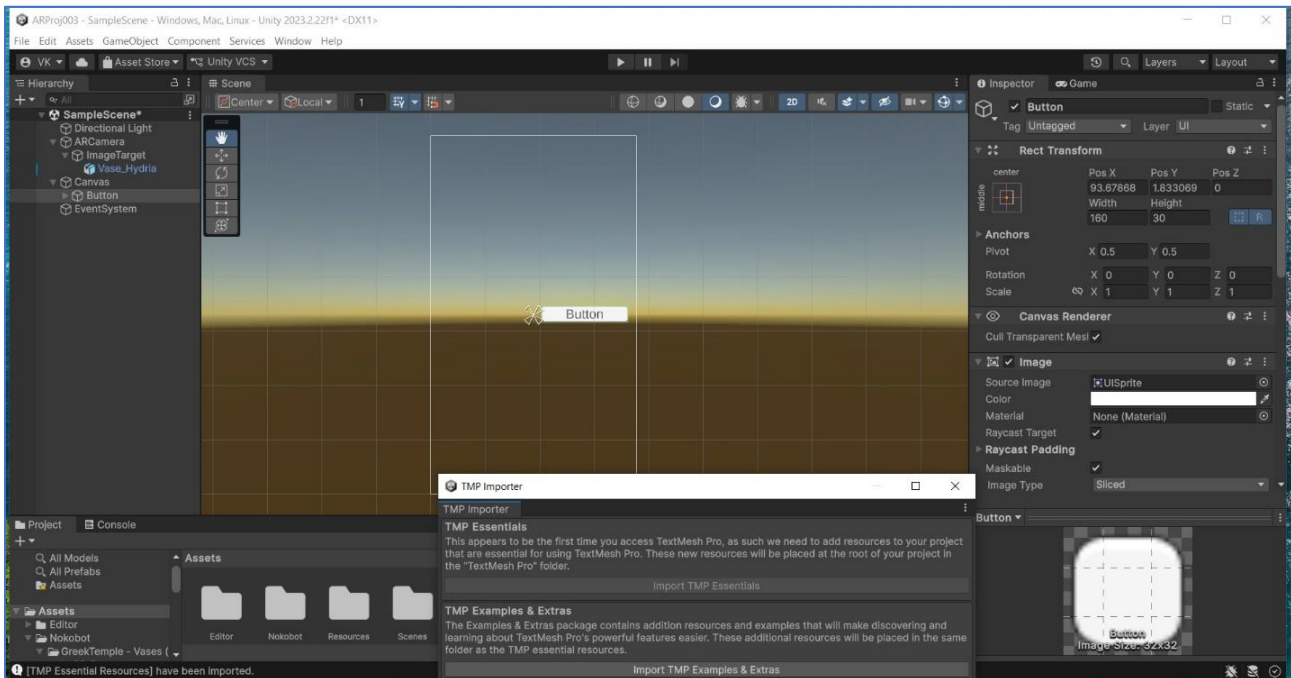
Добавляем кнопку (объект **Button - TextMeshPro**):



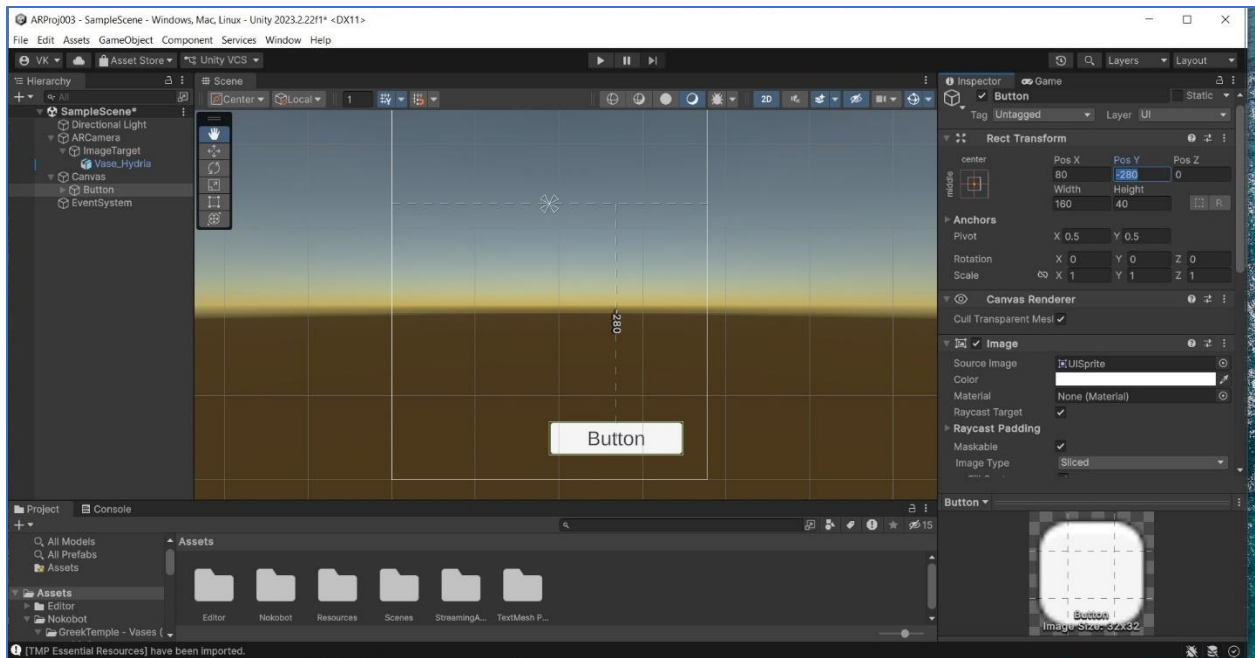
Результат:



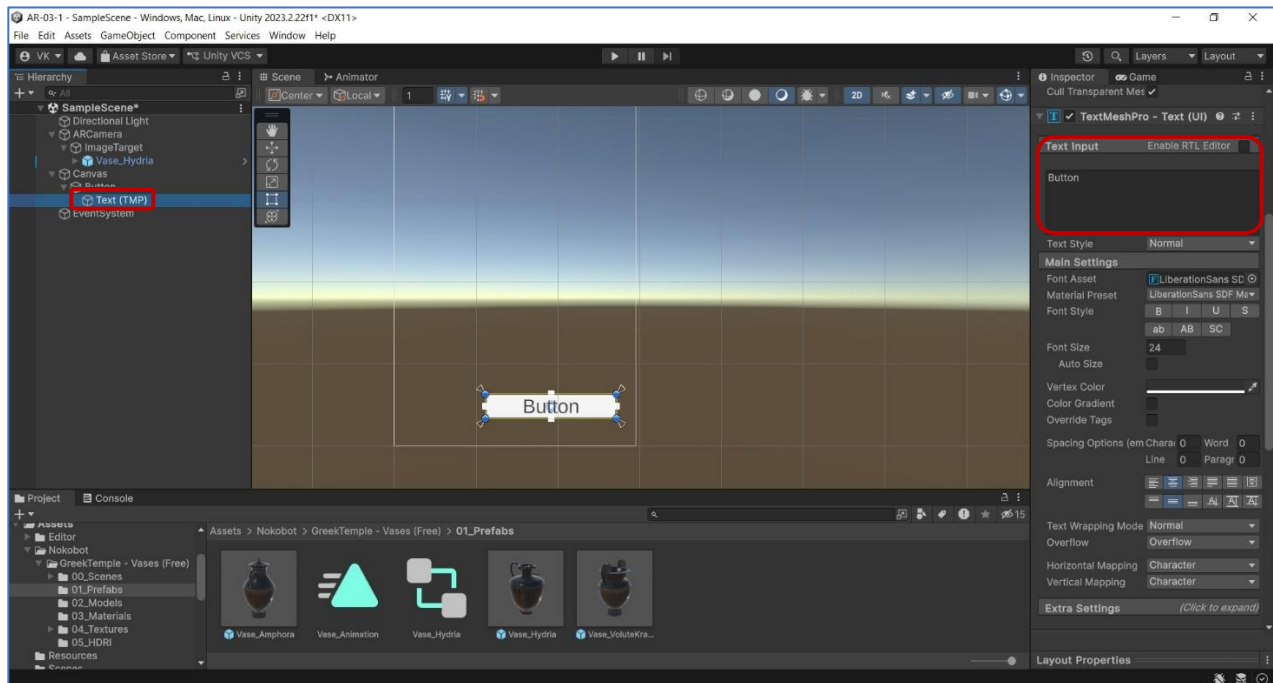
ВАЖНО!! Если выбрать **Import TMP Essentials**, то область кнопки изменится следующим образом и будет готова для введения текстового обозначения:



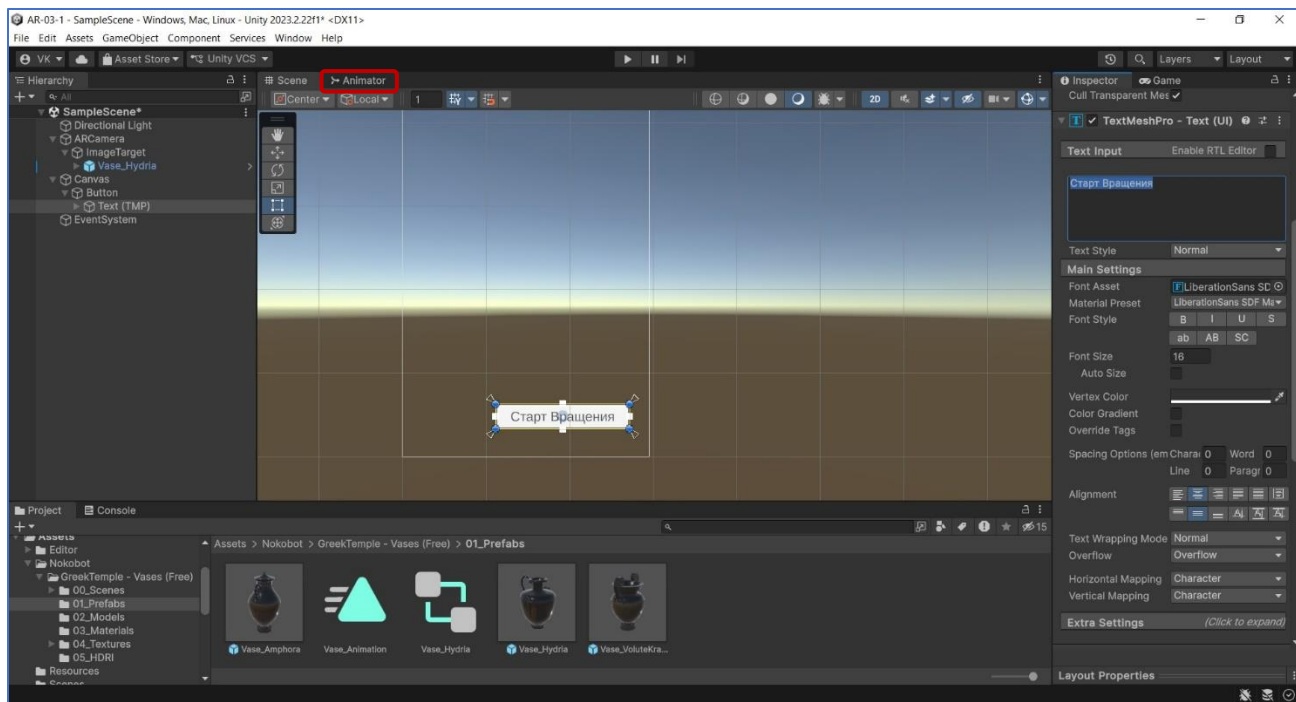
Перемещение кнопки в нужное место, в том числе с использованием якоря для интерактивного перемещения по канве, изменение ее размеров (высота/ширина), может быть осуществлено в Инспекторе – в поле **Rect Transform**. Организуйте удобное с вашей точки зрения положение кнопок меню.



Настраиваем объект **Button** для нанесения на него текстовой метки. Для этого в иерархии раскрываем **Button**, переходим в позицию **Text (TMP)** и переходим в инспектор – в поле **Text Input**:

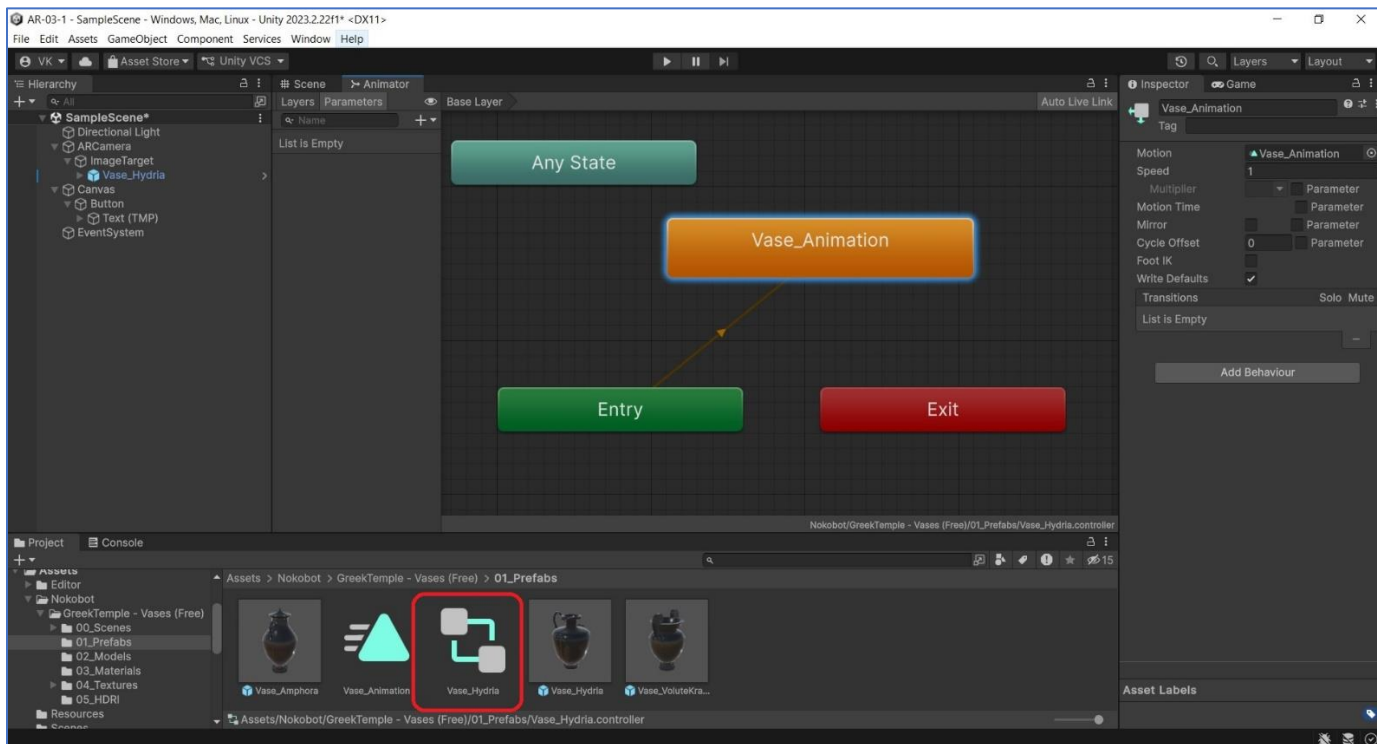


В инспекторе, используя свойства текста, указанные в **Main Settings**, вводим текст **TMP (TextMeshPro)** → текстовая метка на кнопке:



Далее необходимо связать кнопку с ее функционалом – **Старт Вращения**.

Для этого начинаем работать с контроллером анимации 3D Модели **Vase_Hydria** → В рабочей области **Editor**'а переходим с вкладки **Scene** на вкладку **Animator** → сформированный контроллер анимации автоматически загружается в рабочую область **Editor**'а:

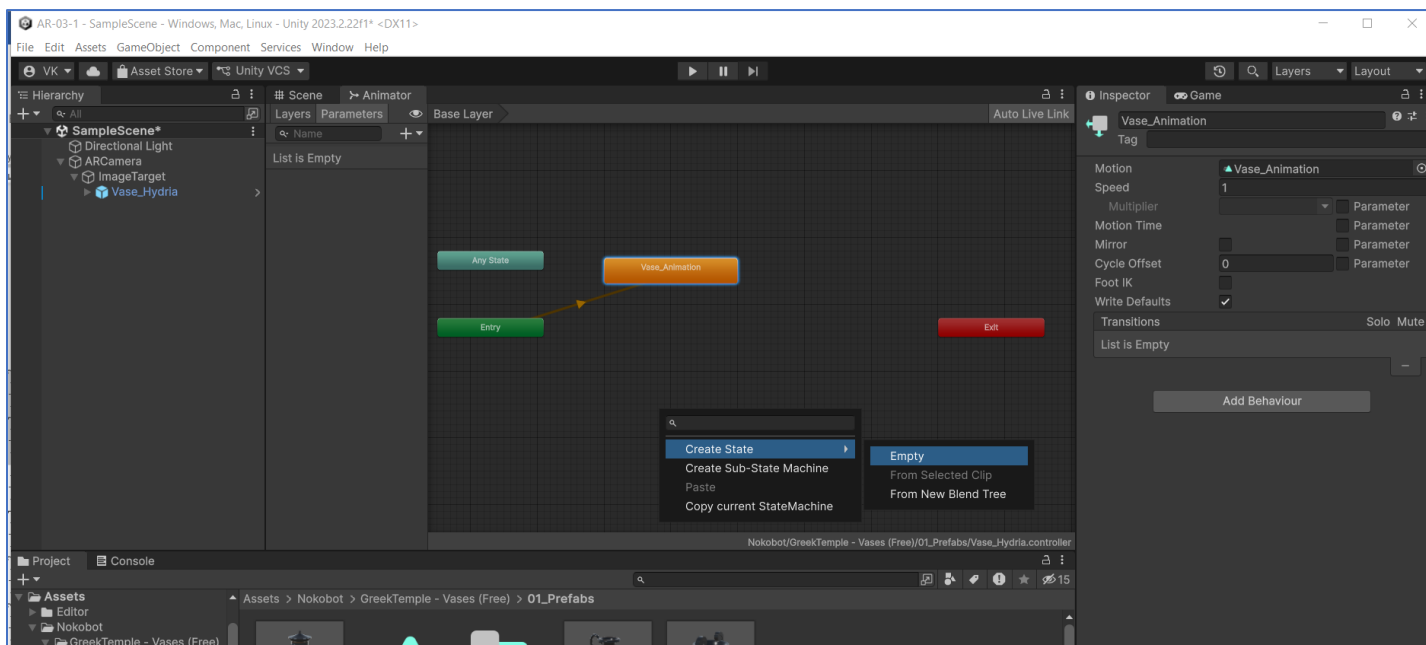


Очевидно, что контроллер управляет анимацией сразу после старта (Состояние **Entry**).

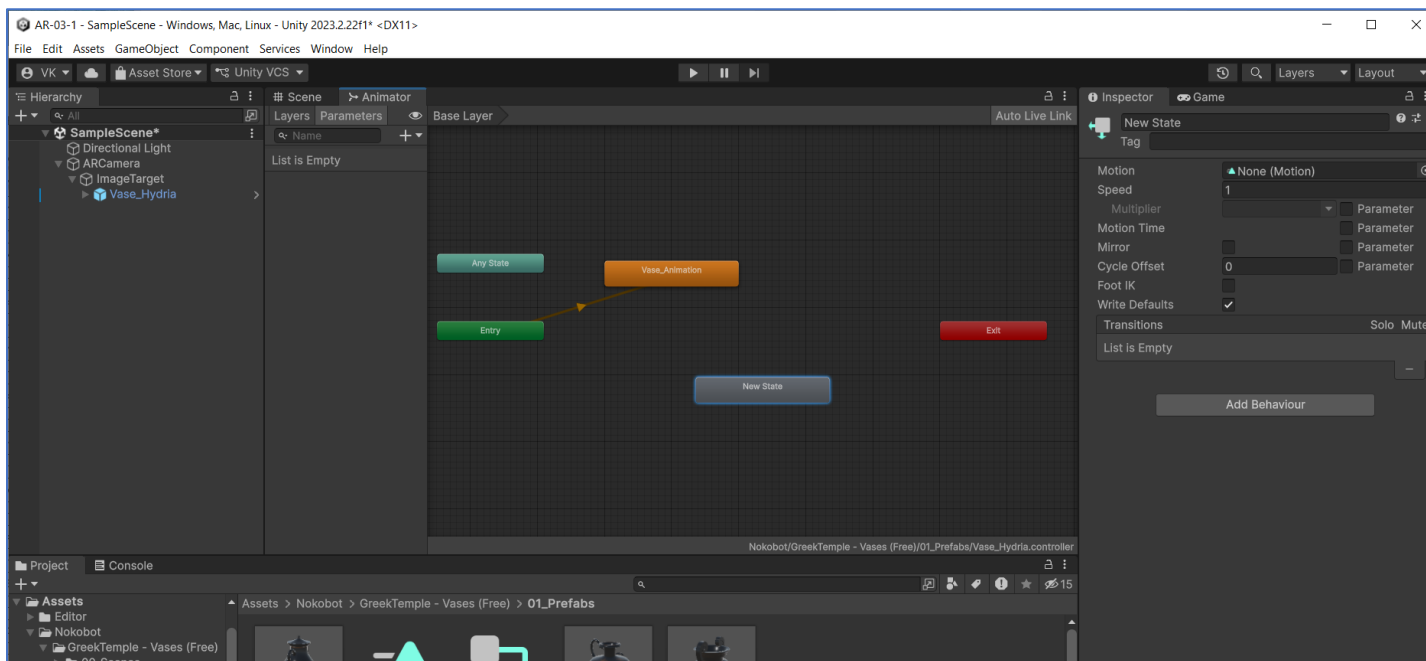
Из состояния **Entry** управление передается непосредственно в состояние **Vase_Animation** (т.е. всегда будет стартовать собственно анимация). В нашем случае необходимо изменить этот базовый сценарий, т.е. включать анимацию с помощью элементов управления (кнопка **Button**).

Чтобы «разорвать» передачу от **Entry** к **Vase_Animation**, ввести в контроллер какое-то дополнительное состояние, предназначением которого будет «замыкание» на себя передачи управления из состояния **Entry** и установление передачи управления от этого нового состояния на анимацию **Vase_Animation**. И, кроме того, необходимо будет воспользоваться возможностями (свойствами) контроллера по **управлению включения/отключения** передачи управления, т.е. найти способ контроля над переходами между состояниями.

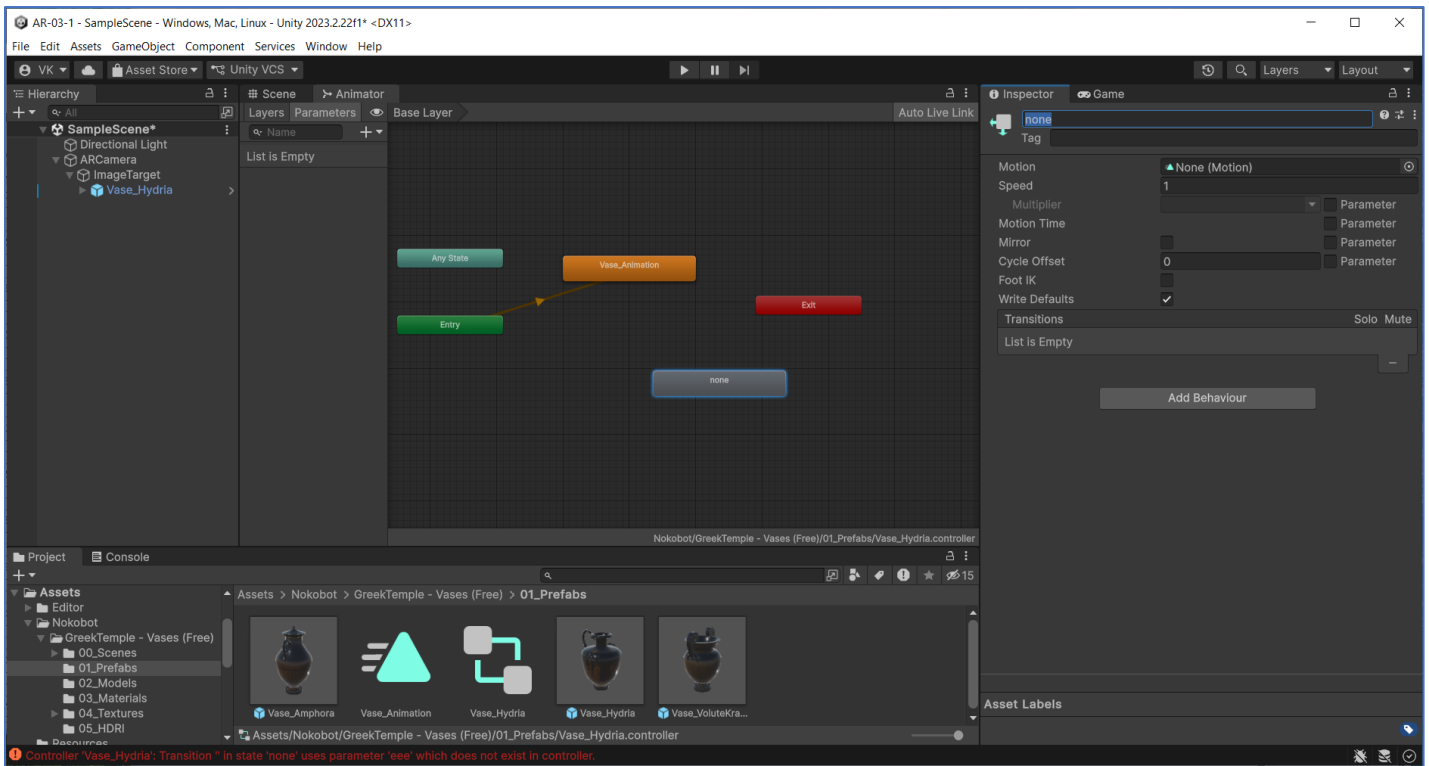
Изменим структуру контроллера – введём дополнительное новое состояние **Empty**. Для этого в пустом поле Аниматора по правой клавише мыши (**RMB**) вызываем меню **Create** и в нем в поле **Create State** выбираем **Empty**:



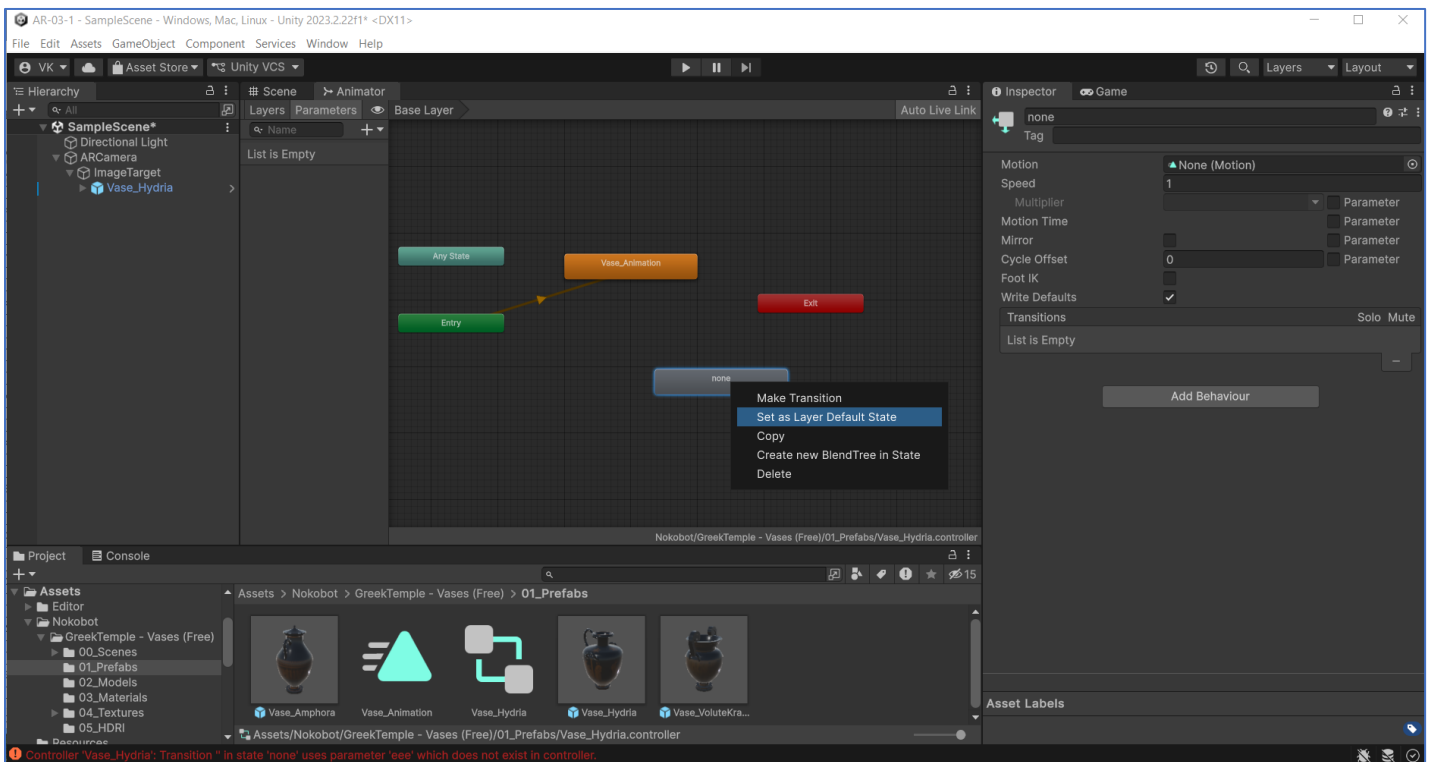
В результате появляется новое пустое, т.е. ничего не содержащее, состояние **New State**:



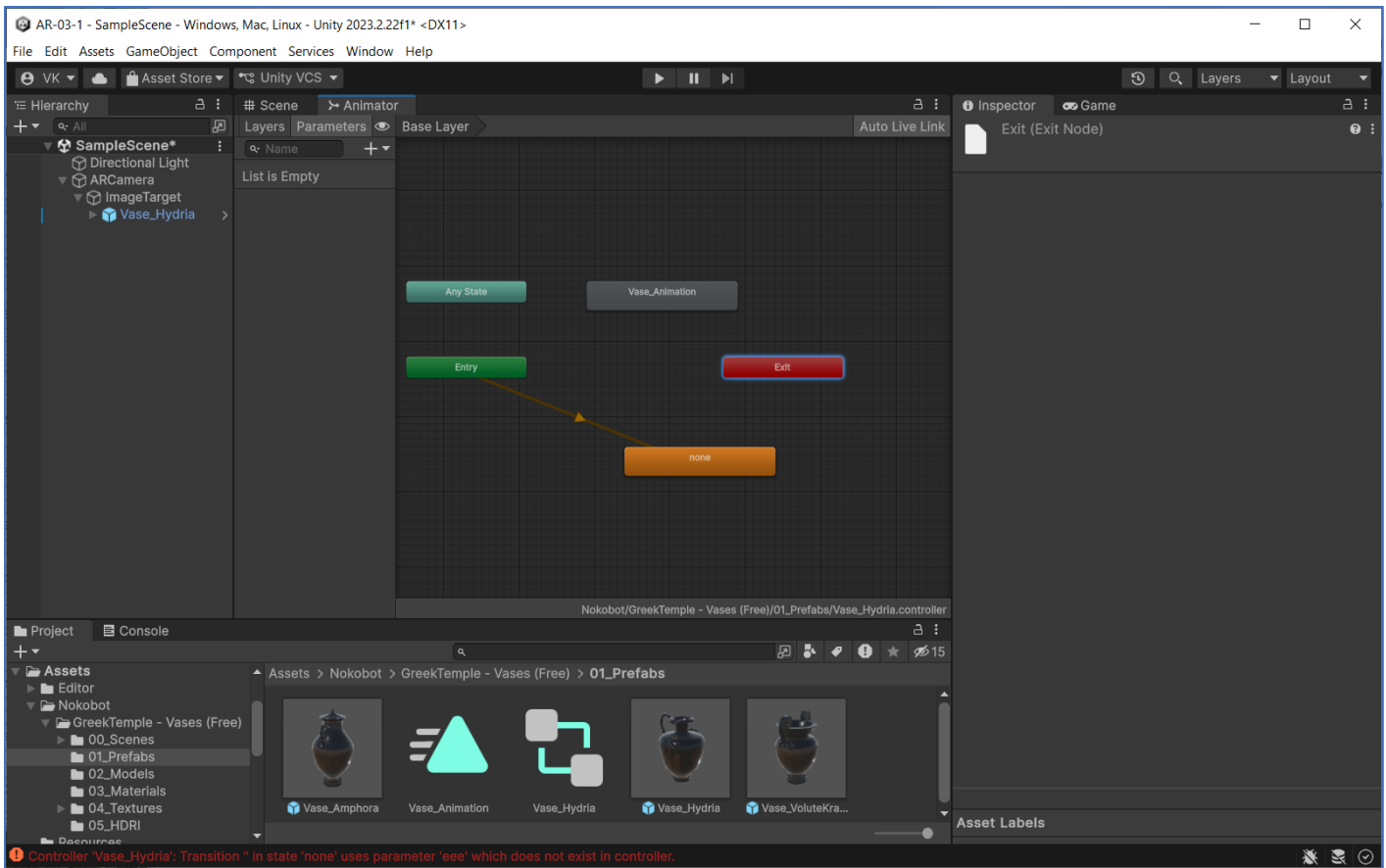
Переименуем это состояние, например, в **none**:



Для того, чтобы переключить **Entry** на **none**, устанавливаем его состояние по **(RMB)**, как **Set as Layer Default State**:



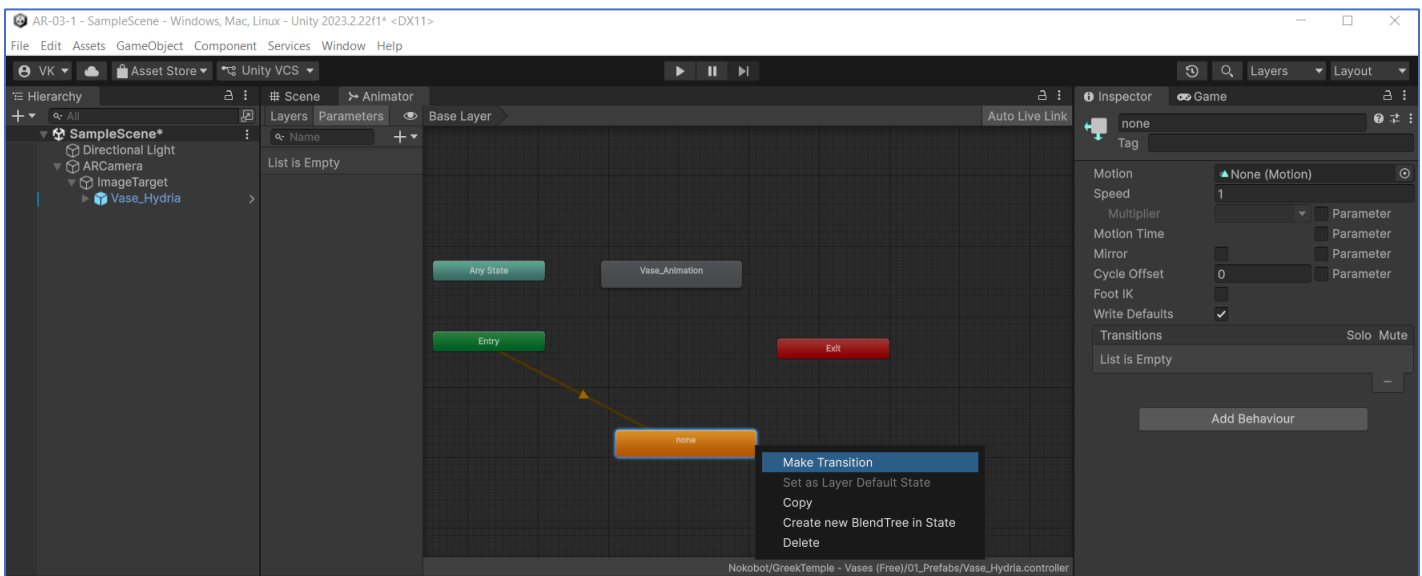
Автоматически передача от **Entry** переключится на **none** вместо **Vase_Animation**:



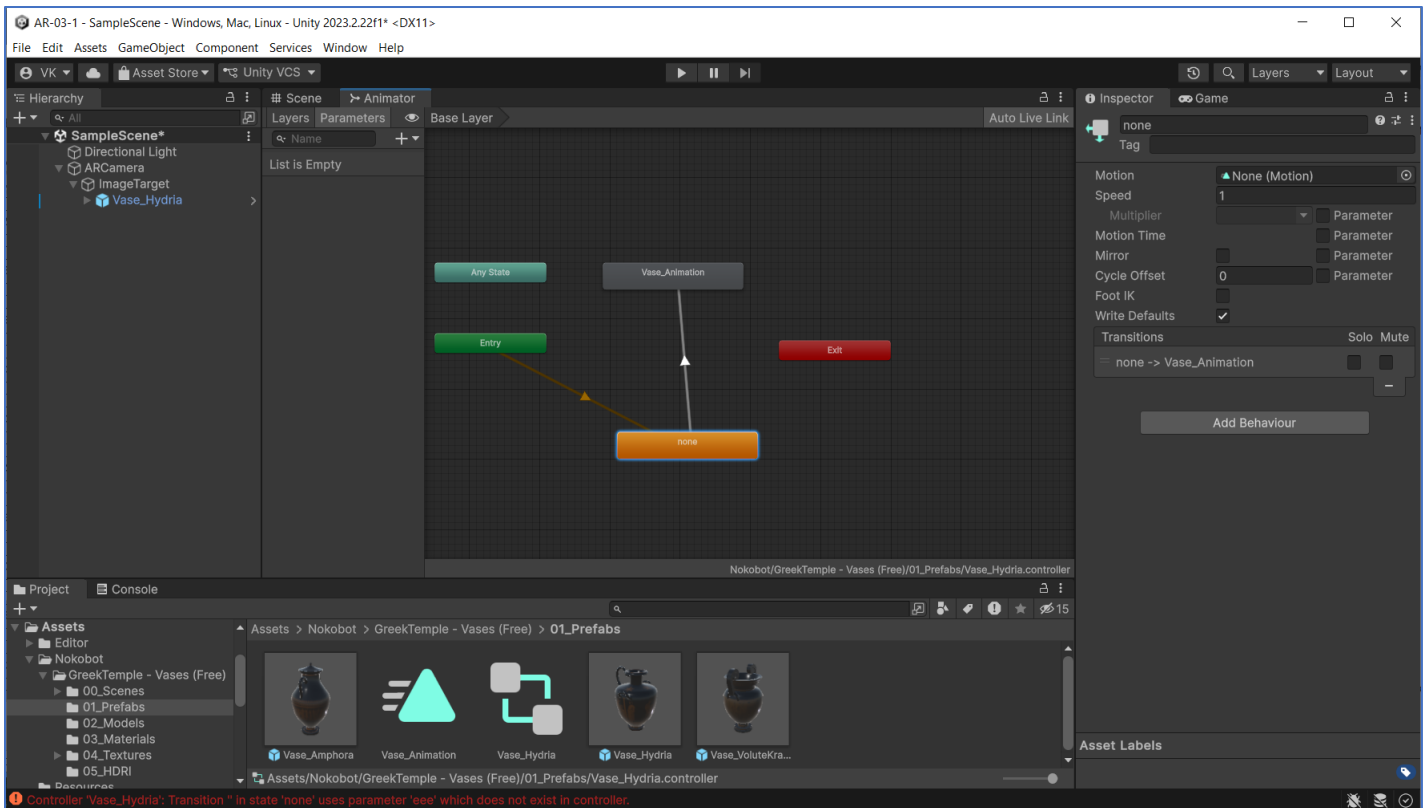
Теперь контроллер анимации сразу переключает управление на состояние **none**, к которому мы не привязали никакой анимации и, следовательно, и анимации не происходит.

Чтобы тем не менее предусмотреть в контроллере возможность анимации, то есть перехода в состояние **Vase_Animation**, установим такой переход: теперь от состояния **none** в состояние **Vase_Animation**.

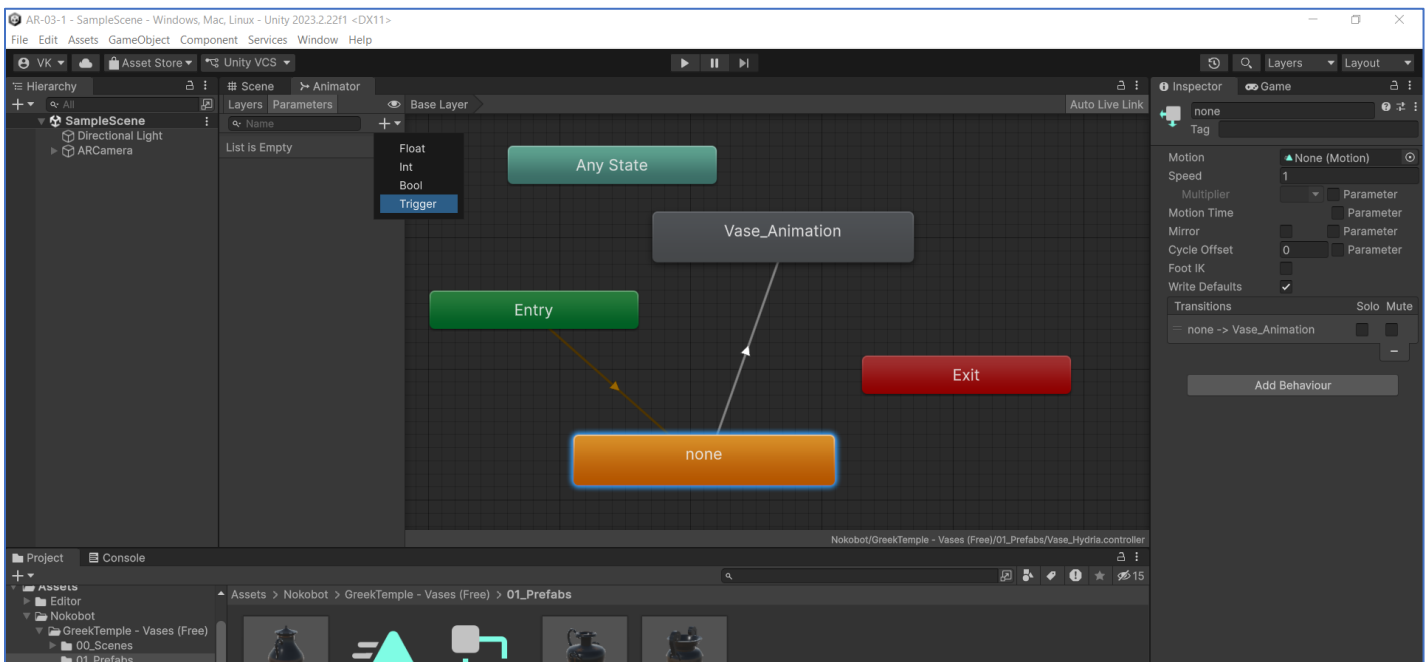
Для этого выбираем по **(RMB)** для **none** режим **Make Transition Animation**:



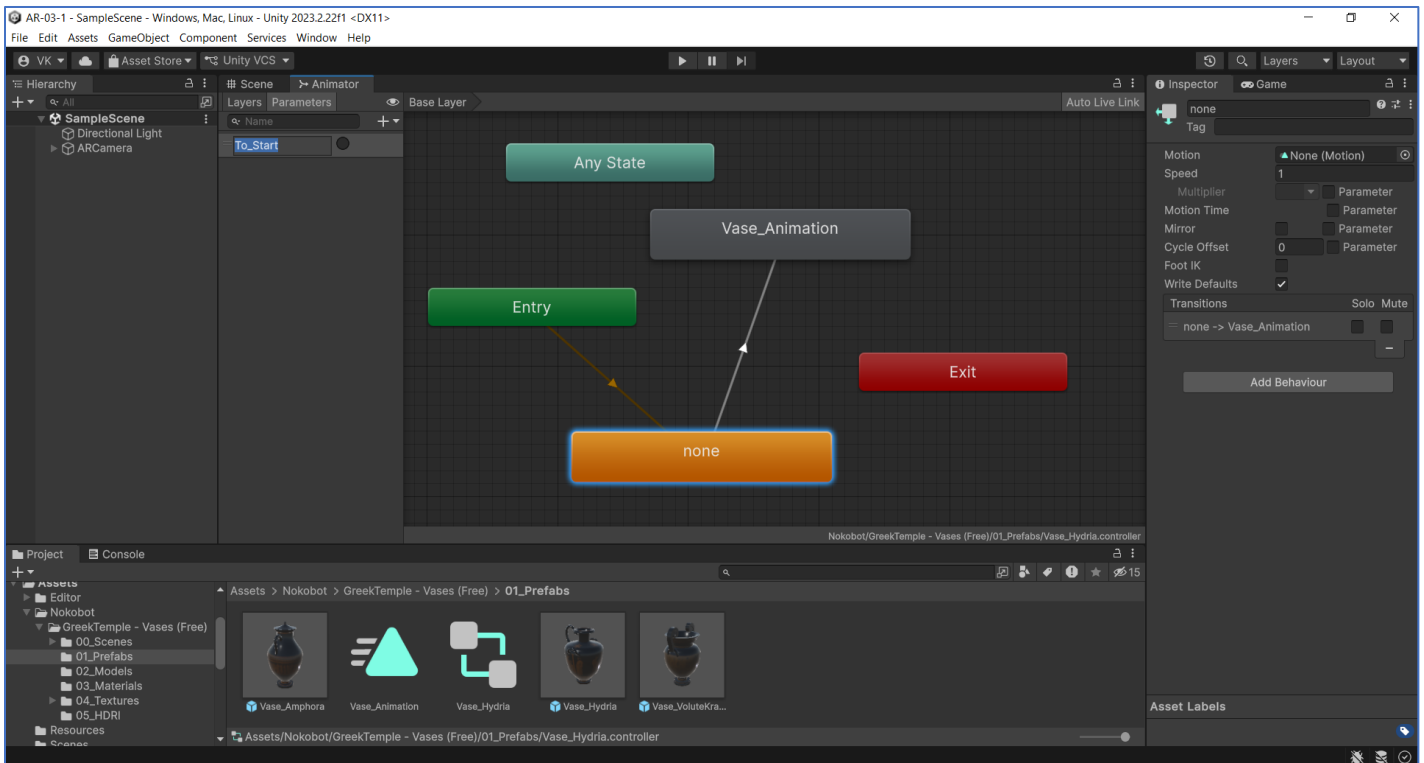
И в интерактивном режиме проведём «стрелку» от **none** к **Vase_Animation**:



И еще – для возможности как-то управлять (подключать по требованию) тот или другой переход – а в нашем случае только переход от **none** к **Vase_Animation** - давайте «промаркируем» переходы между состояниями для того, чтобы можно было управлять ими (активировать) с помощью этих маркеров. Маркеры – т.н. **триггеры** в терминах Аниматора - создаются в закладке **Parameters**. Выберем закладку **Parameters**, нажимаем «+» и в выпадающем меню выбираем **Trigger**:

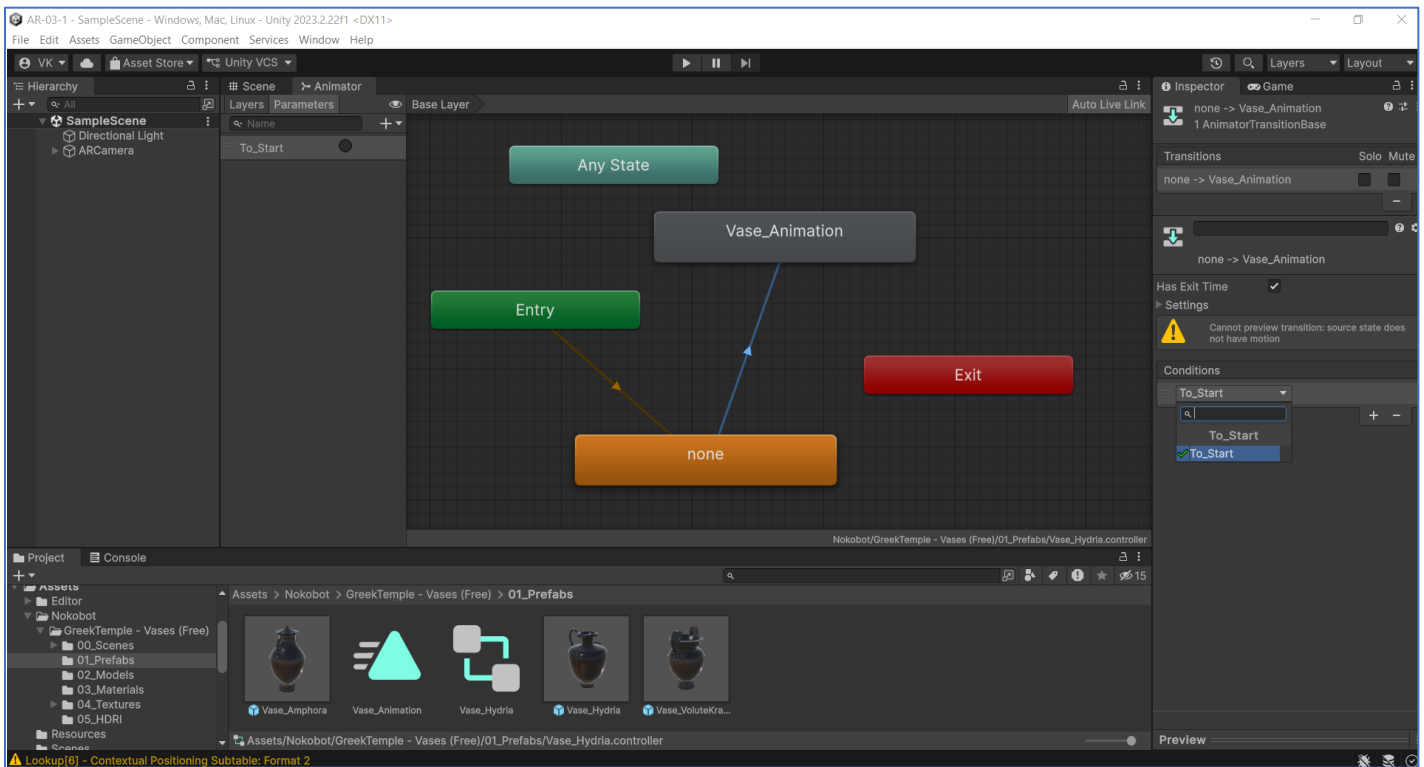


Создадим триггер, который мы планируем использовать в дальнейшем для управления включением анимации (т.е. активацией перехода от состоянию **none** к состоянию **Vase_Animation**) и назовем его, например **To_Start**:



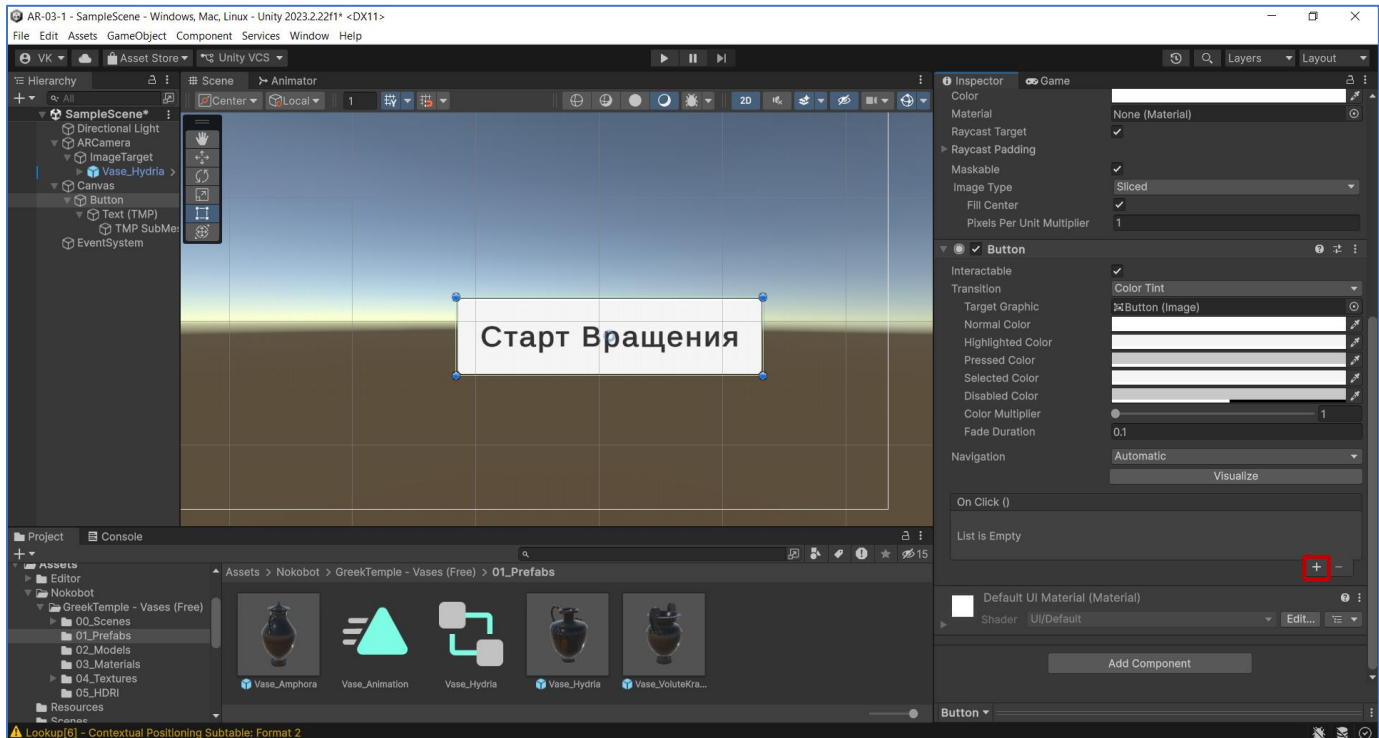
Промаркируем переход **none - Vase_Animation** триггером **To_Start**. Для этого выберем (выделим) переход **none → Vase_Animation**, кликнув на стрелку этого перехода.

При этом в **Инспекторе** в поле **Conditions** из выпадающего меню условий необходимо выбрать триггер **To_Start**:

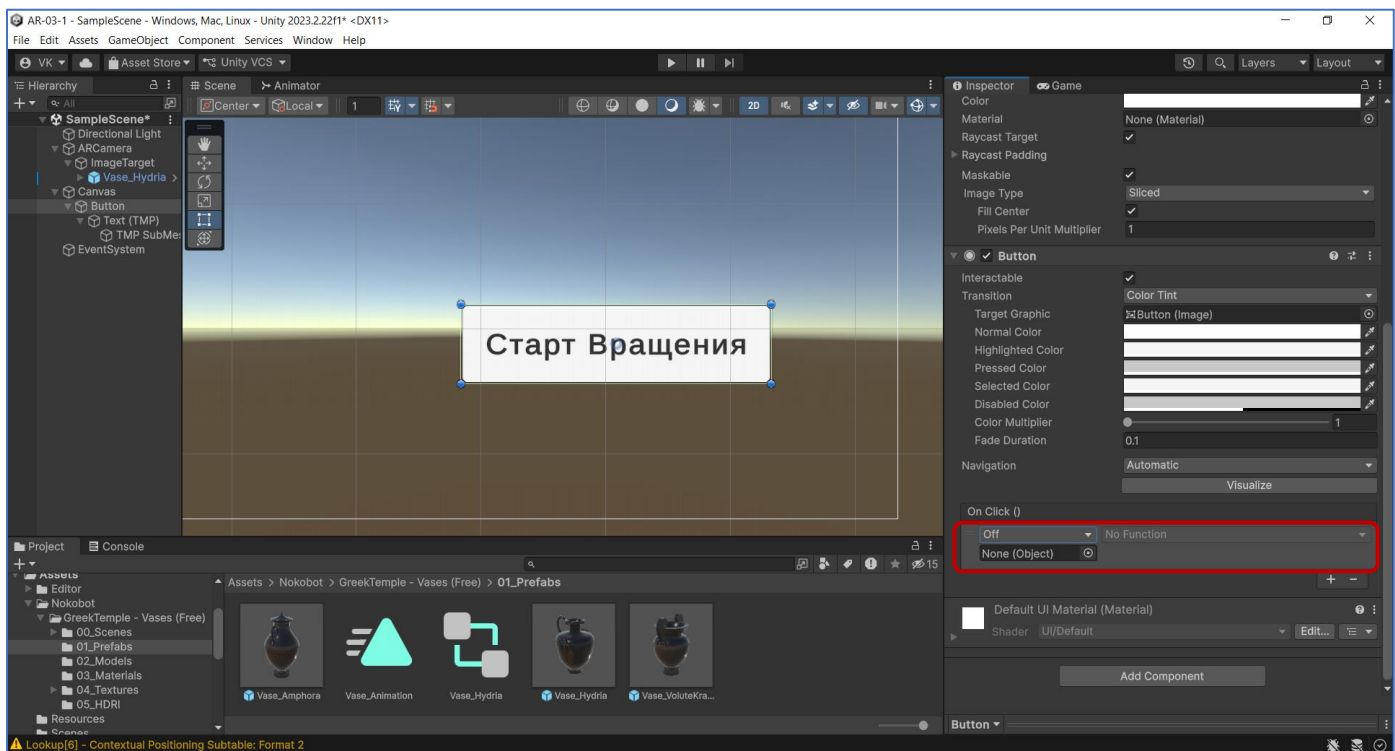


Таким образом работа по модификации/настройке контроллера в нашем случае завершена.

Возвращаемся в Scene в 2D Canvas для работы с кнопкой (**Button** в Иерархии) **Старт Вращения**. В Инспекторе кнопки **Button** выбираем настройку **On Click ()**.

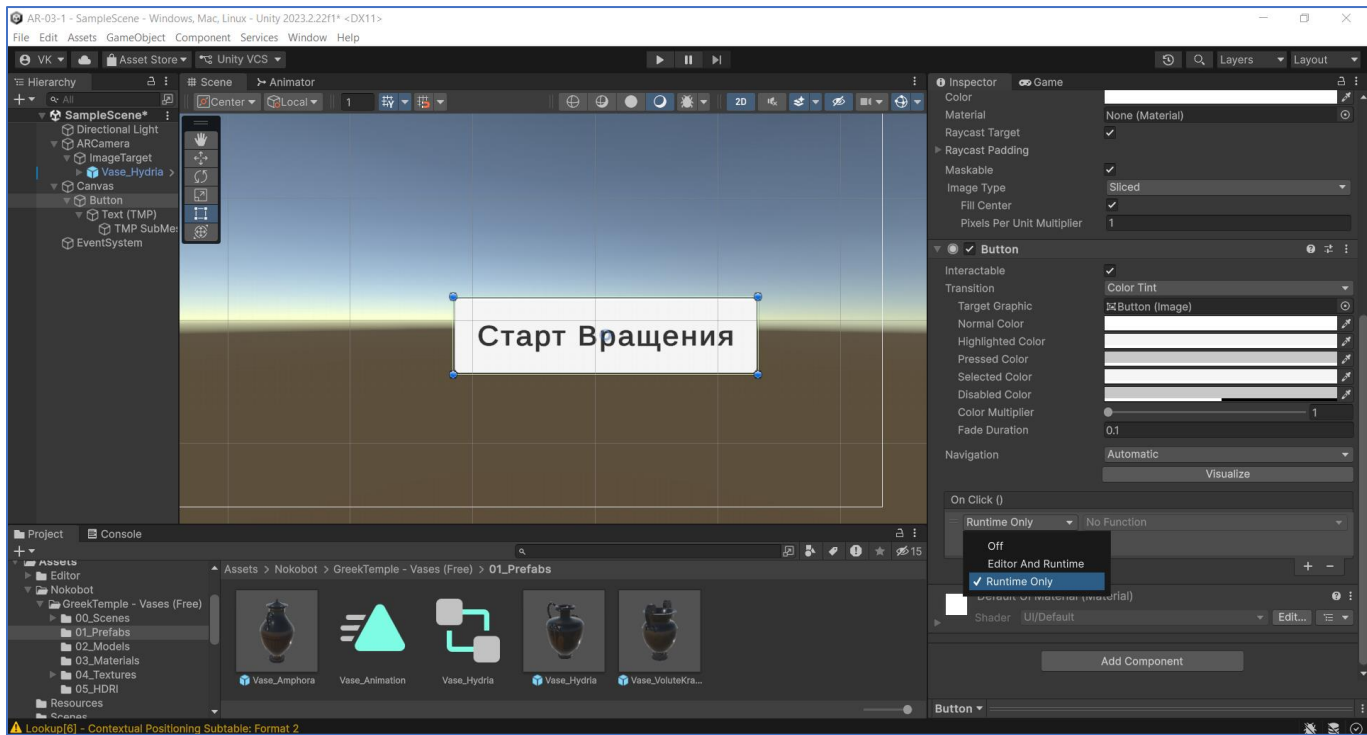


Список настроек реакций пока пуст (**List is Empty**). Настраиваем реакцию на нажатие кнопки **Button** через «+» - пополнение списка реакций:

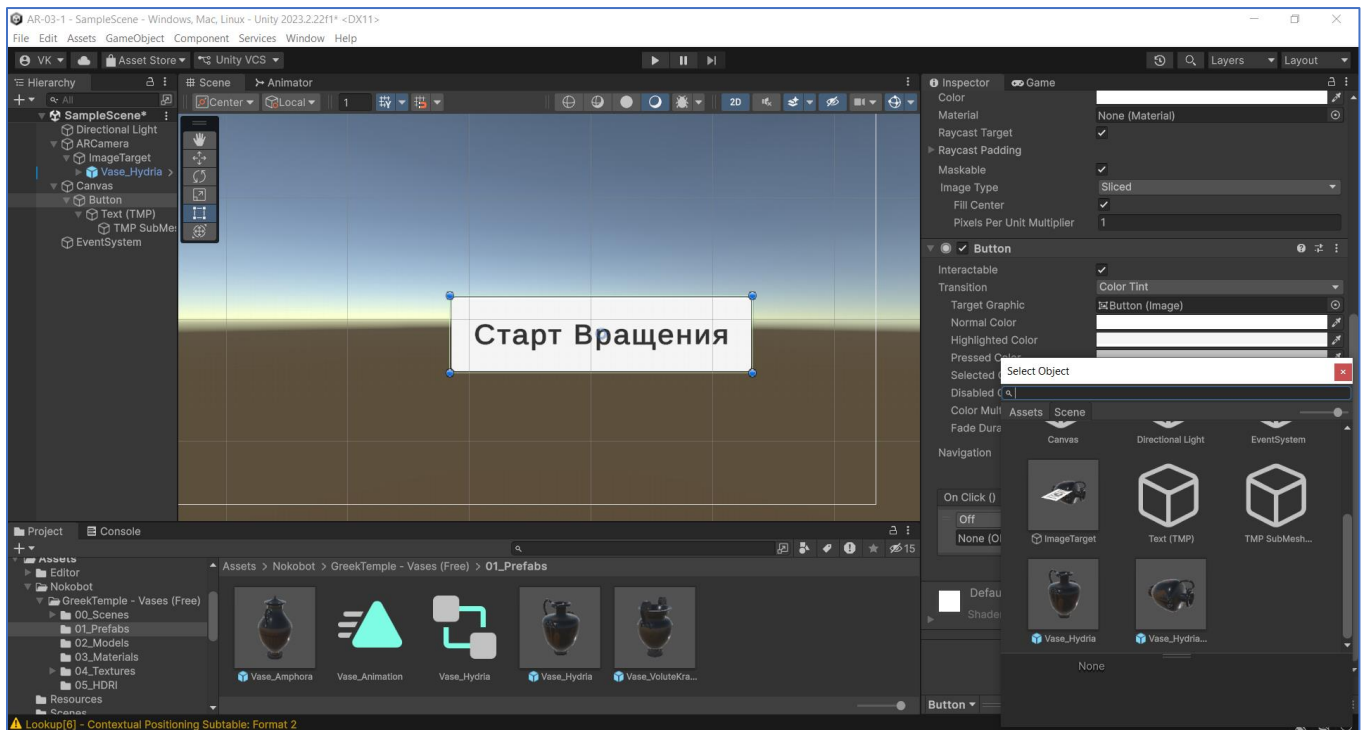


Для **On Click ()** нет режима (в левом падающем меню – **Off**), не выбран **объект**, на который направлены реакции от нажатия кнопки (**None (Object)**), и не установлен **функционал** (**No Function**). Последовательно задаем реакции:

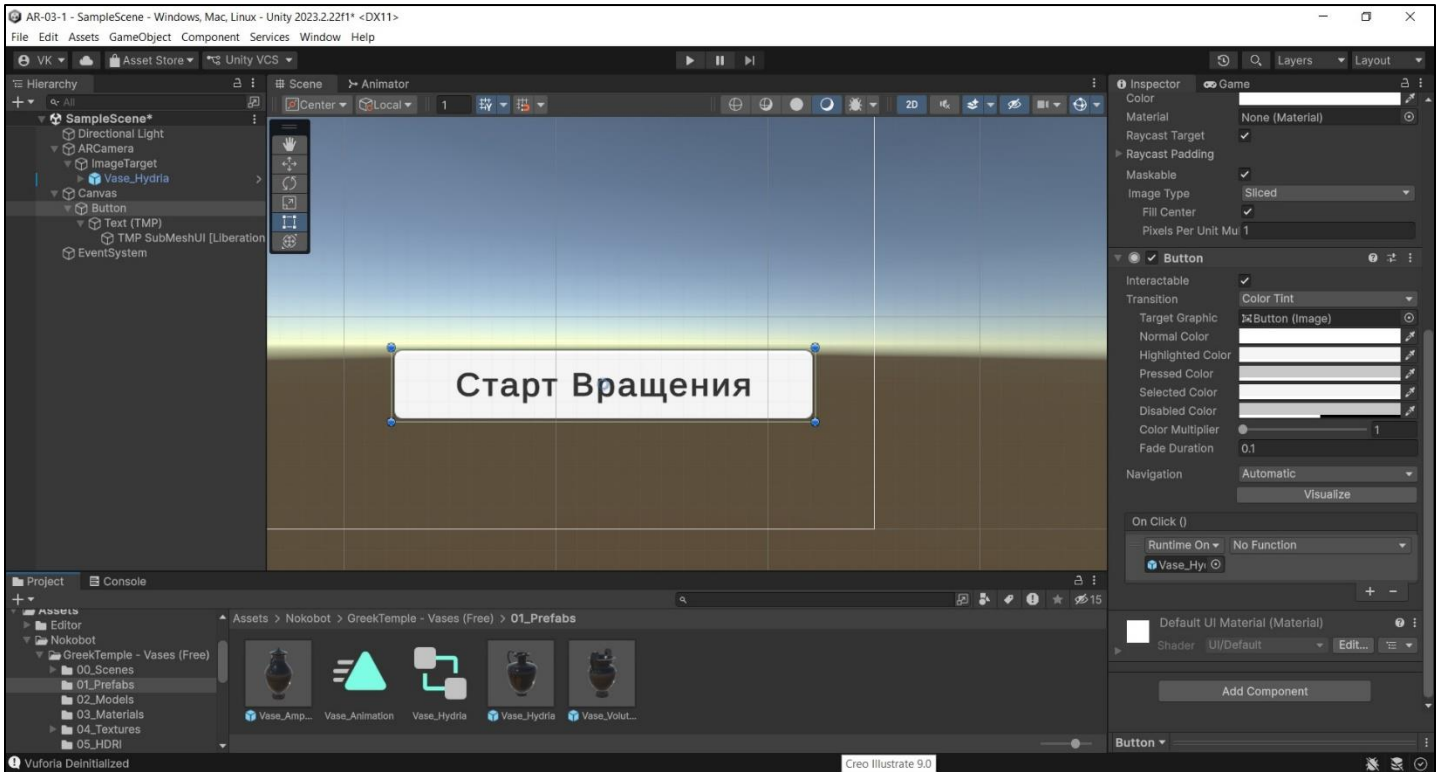
Выбираем режим Runtime Only:



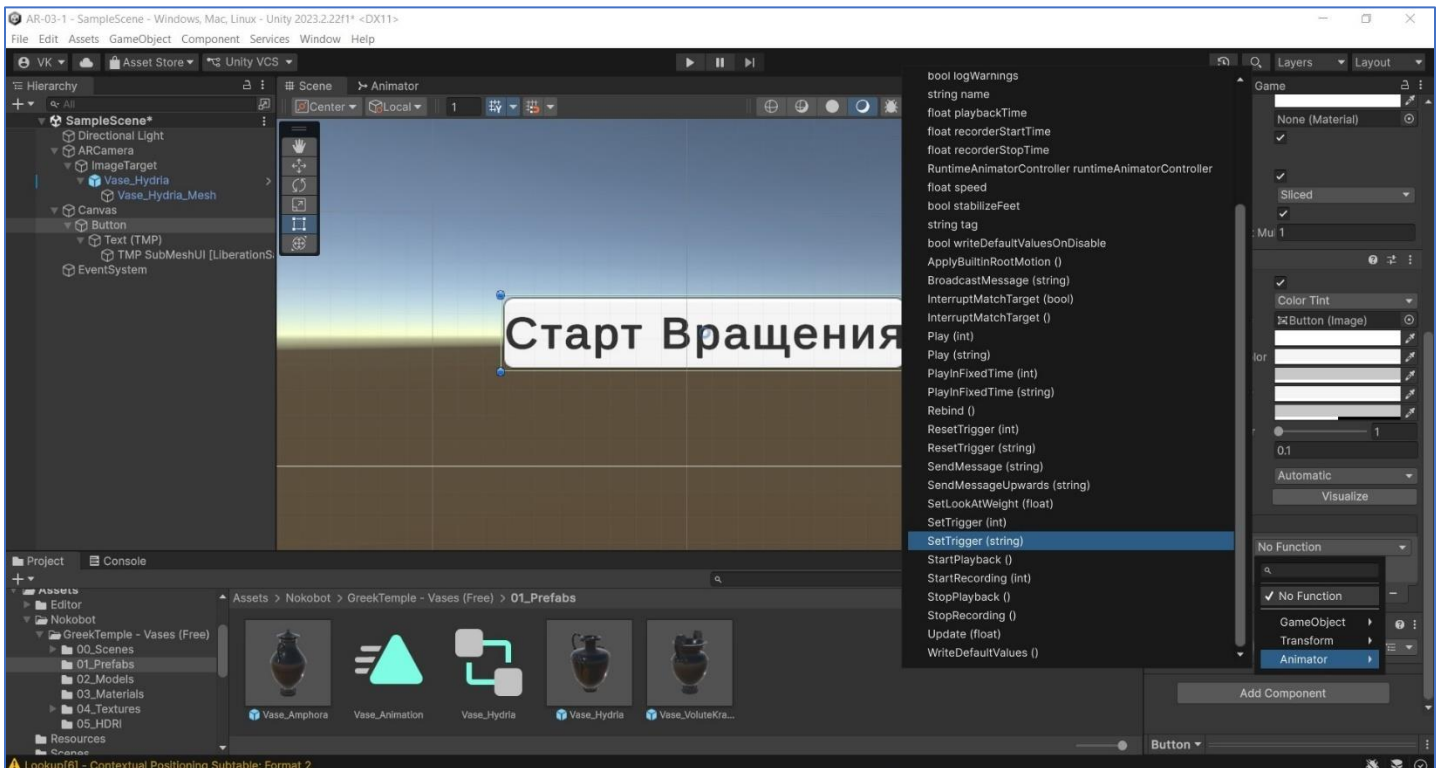
Выбираем объект (в нашем случае это 3D Модель Vase_Hydria): в поле **Select Object** → закладка **Scene** → ищем объект - **Vase_Hydria**, выбираем его:



Результат:

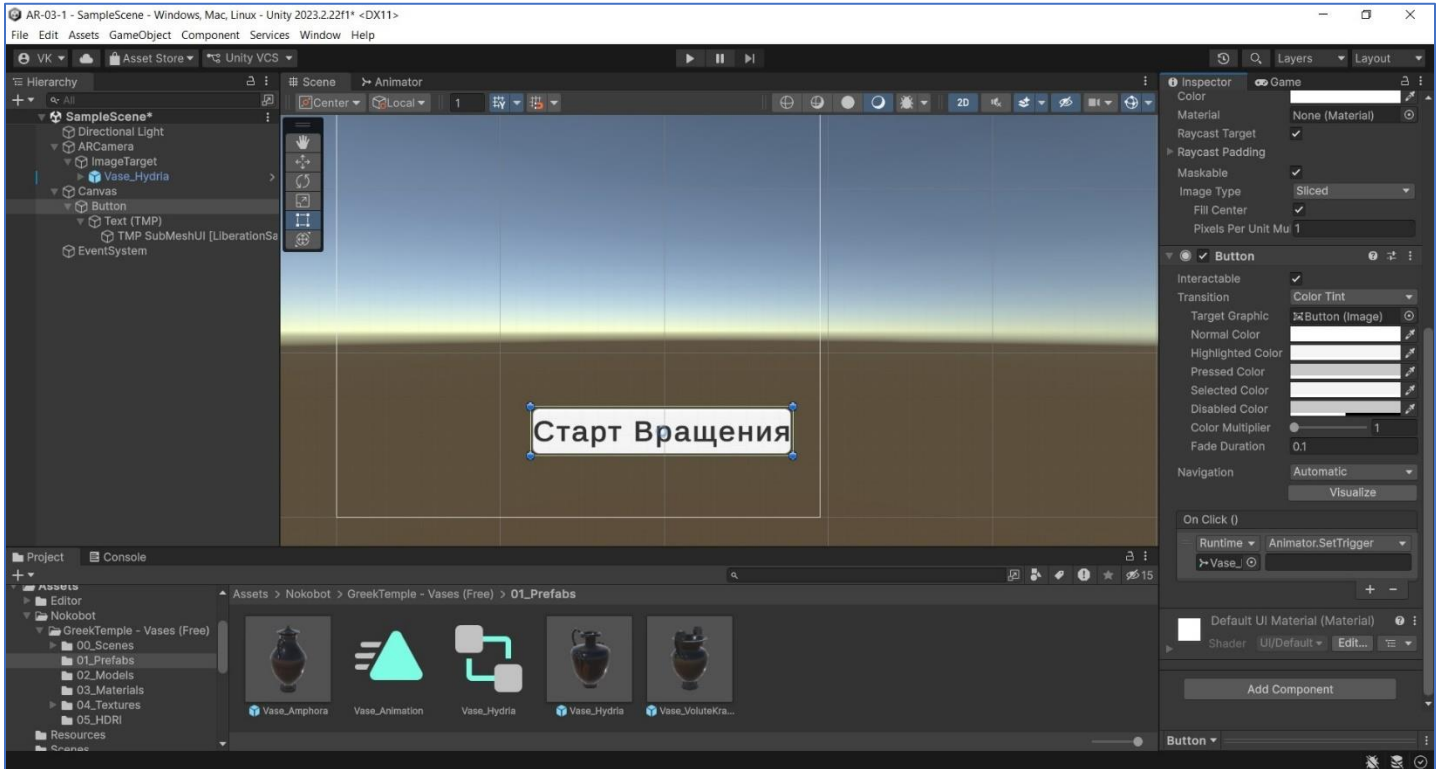


После настроек выбора режима и объекта переходим к **установке функционала**. В соответствующем поле **No Function** в появившемся новом меню выбираем **Animator**, и среди всех открывшихся функций **Animator'a** нашего объекта ищем подходящую, т.е. – функцию работы с триггерами, созданными ранее для контроллера вращения вазы **Vase_Hydria**.



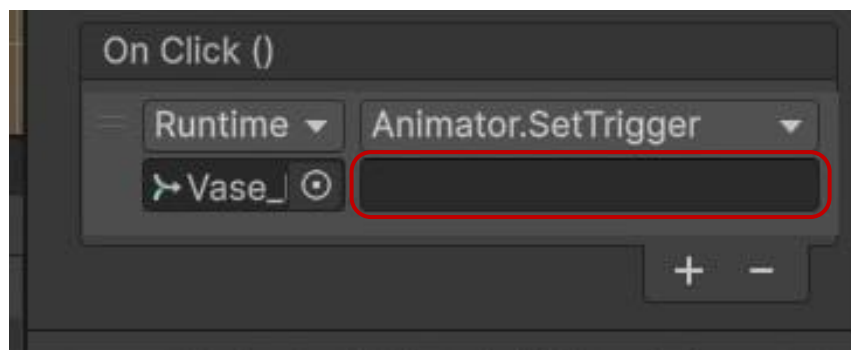
Обратите внимание, что список может потребовать скроллинга, мы можем обнаружить несколько функций работы с триггерами, надо выбрать наиболее подходящую. Ранее мы

создали триггер **To_Start**. Обращение к нему идет по имени, имя – это тип «строка» (**string**) → в рассматриваемом списке есть наиболее вероятный подходящий вариант – **SetTrigger(string)**.

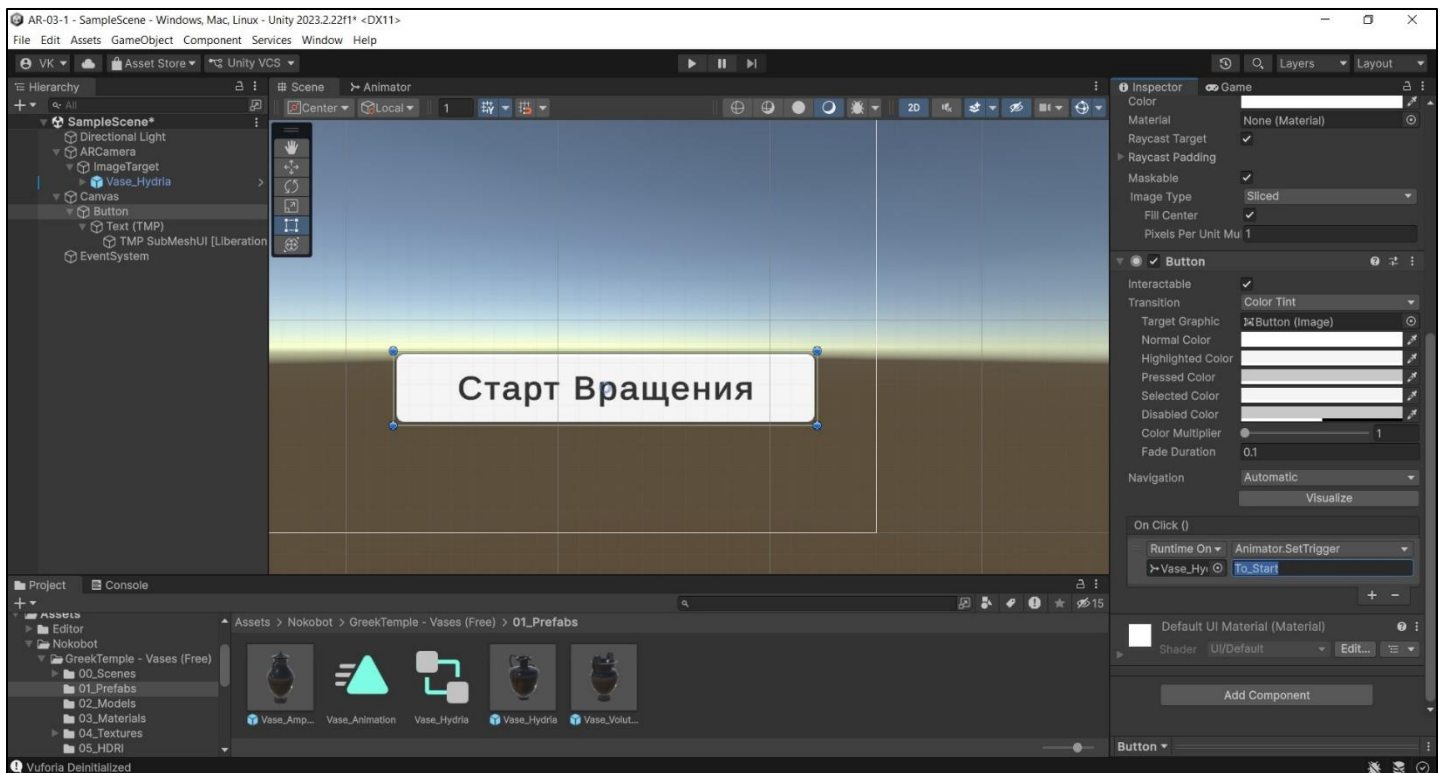


В результате – в списке **On Click ()** появится ссылка на контроллер вместо ссылки на объект, вместо **No Function** появляется указание на функцию **Set Trigger** контроллера (**Animator.Set Trigger**), и новое поле для установки ссылки на выбранный параметр - текстовое обозначение триггера.

Выбираем «наш» триггер – **To_Start** (Обратите внимание – наименование необходимо вводить вручную!!! В будущих версиях **Unity Editor** должно появиться меню выбора их списка доступных имен триггеров проекта):



В результате:

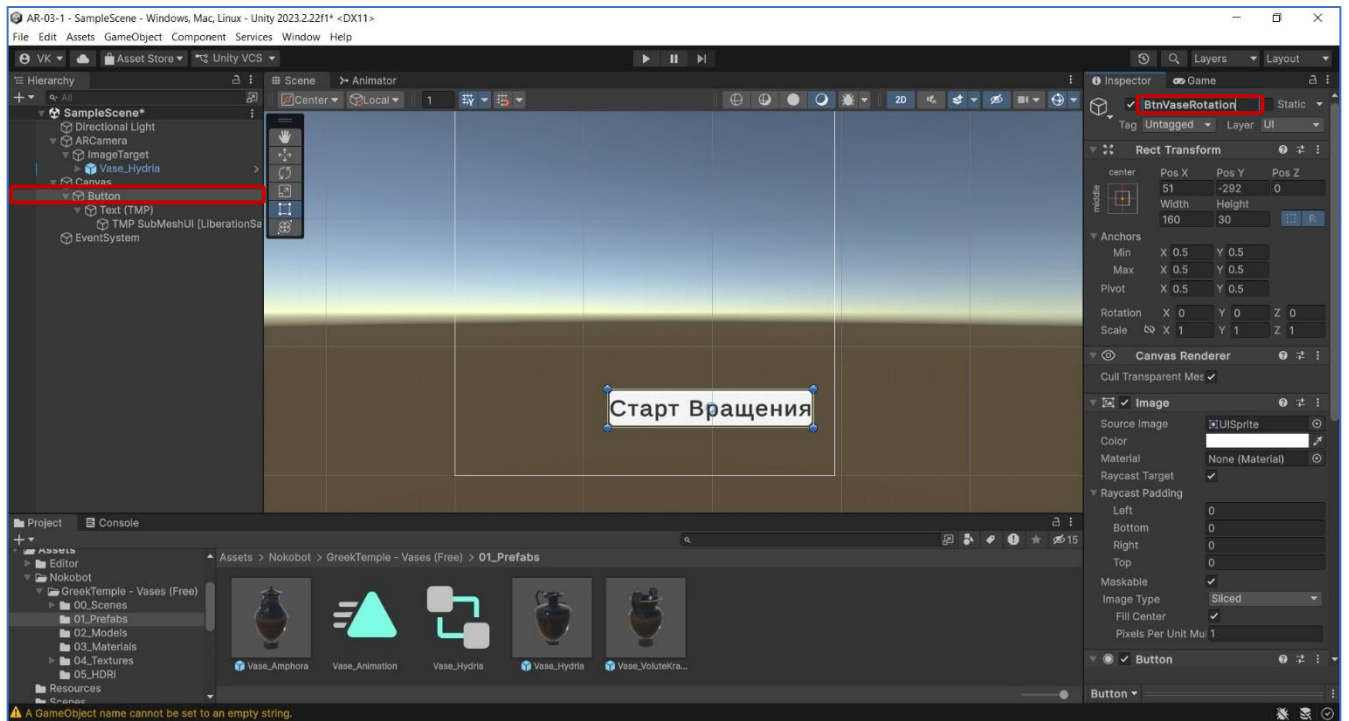


Настройки завершены.

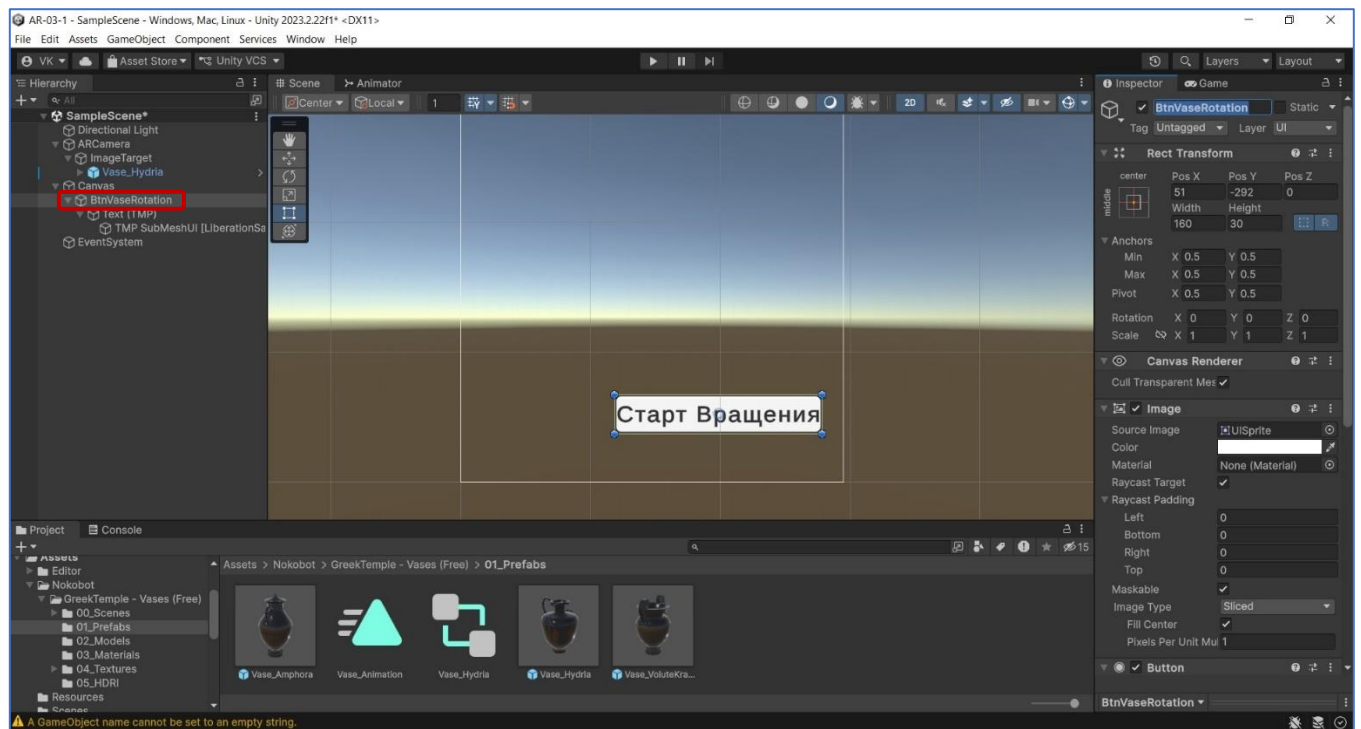
Поскольку в данной ЛР предусмотрено несколько управляющих элементов плоского меню (кнопок **Button**) рекомендуется каждой из них присвоить индивидуальные имена, чтобы различать их в **иерархии** для последующего администрирования.

Первую оформленную Button переименуем в **BtnVaseRotation** →

- в **иерархии** выбираем кнопку для переименования **Button** →
- в **инспекторе** в верхнем поле с содержанием **Button** вводим новое имя **BtnVaseRotation**:

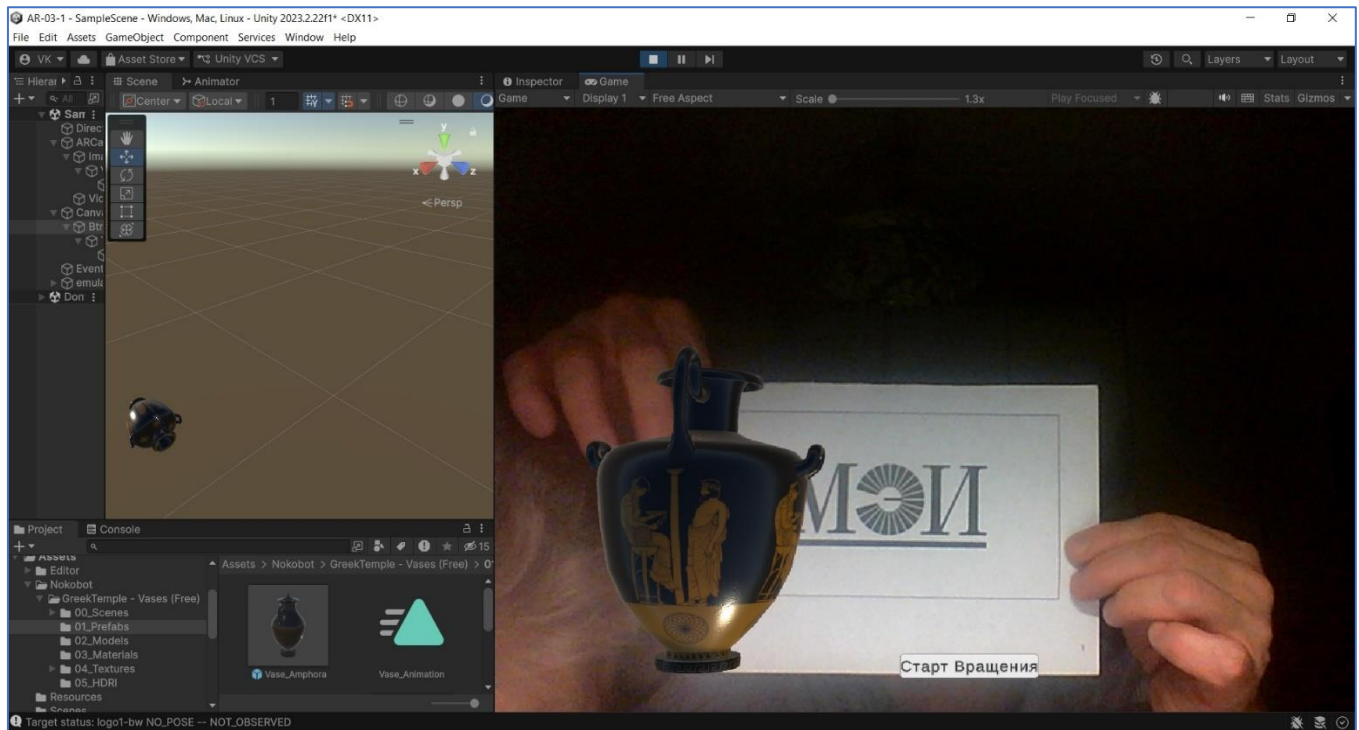
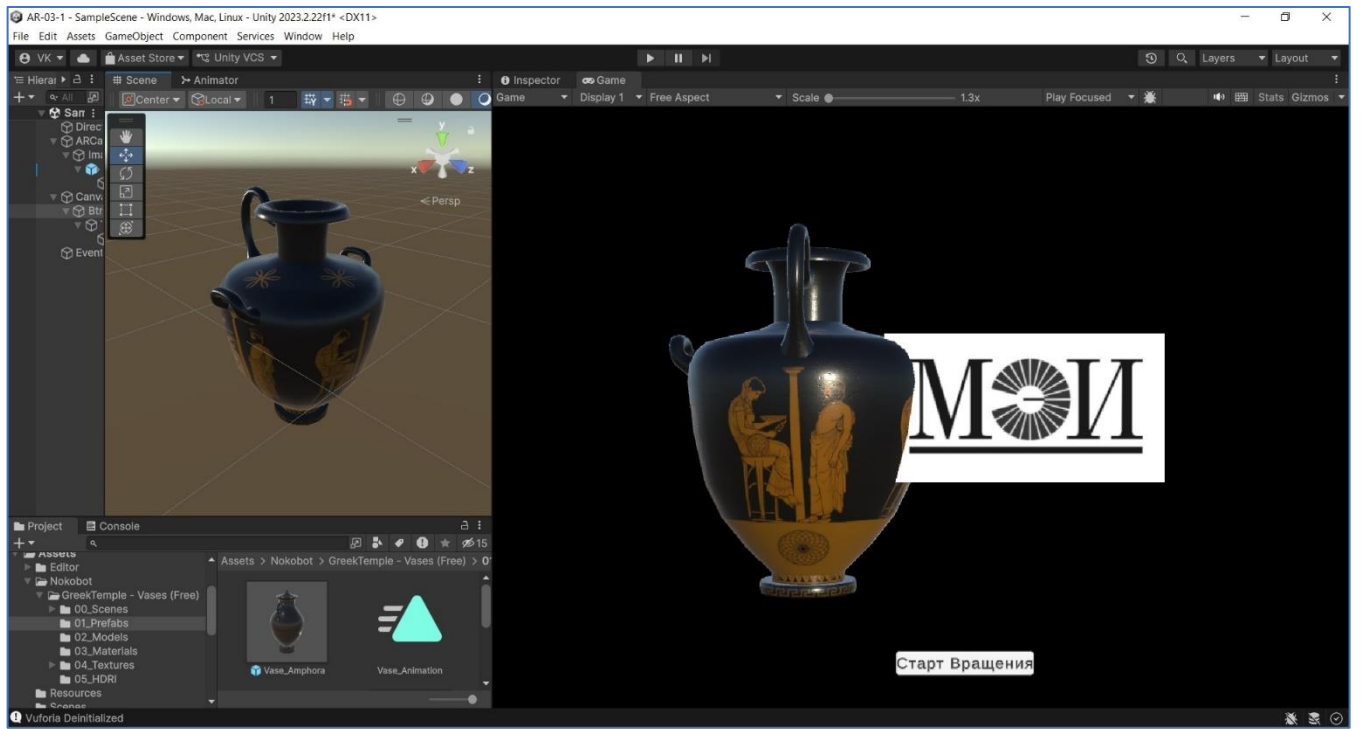


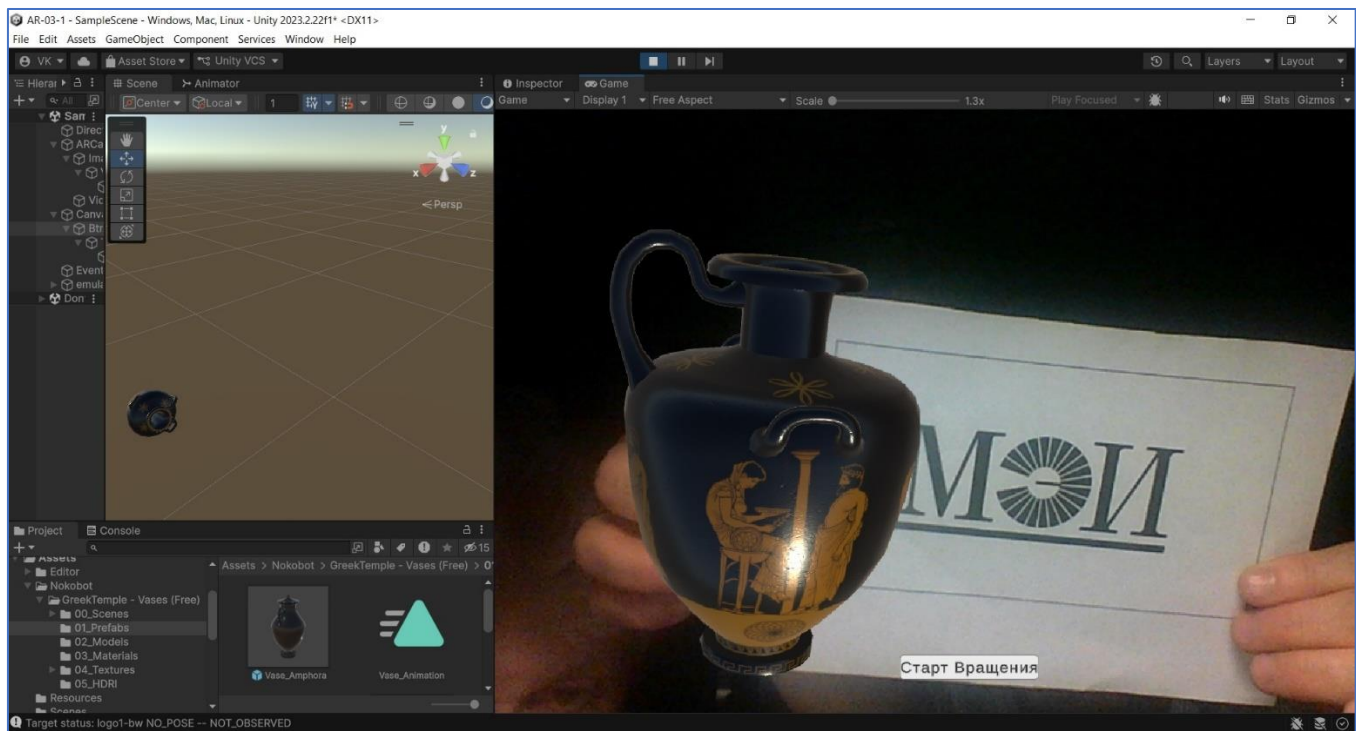
В результате:



Работа по созданию элемента управления кнопка плоского меню **Button** для старта ранее разработанной анимации завершена.

Можно проверить работоспособность проекта в режиме Game и выполнить сборку Приложения ДР для загрузки на МУ (получение файла .apk)



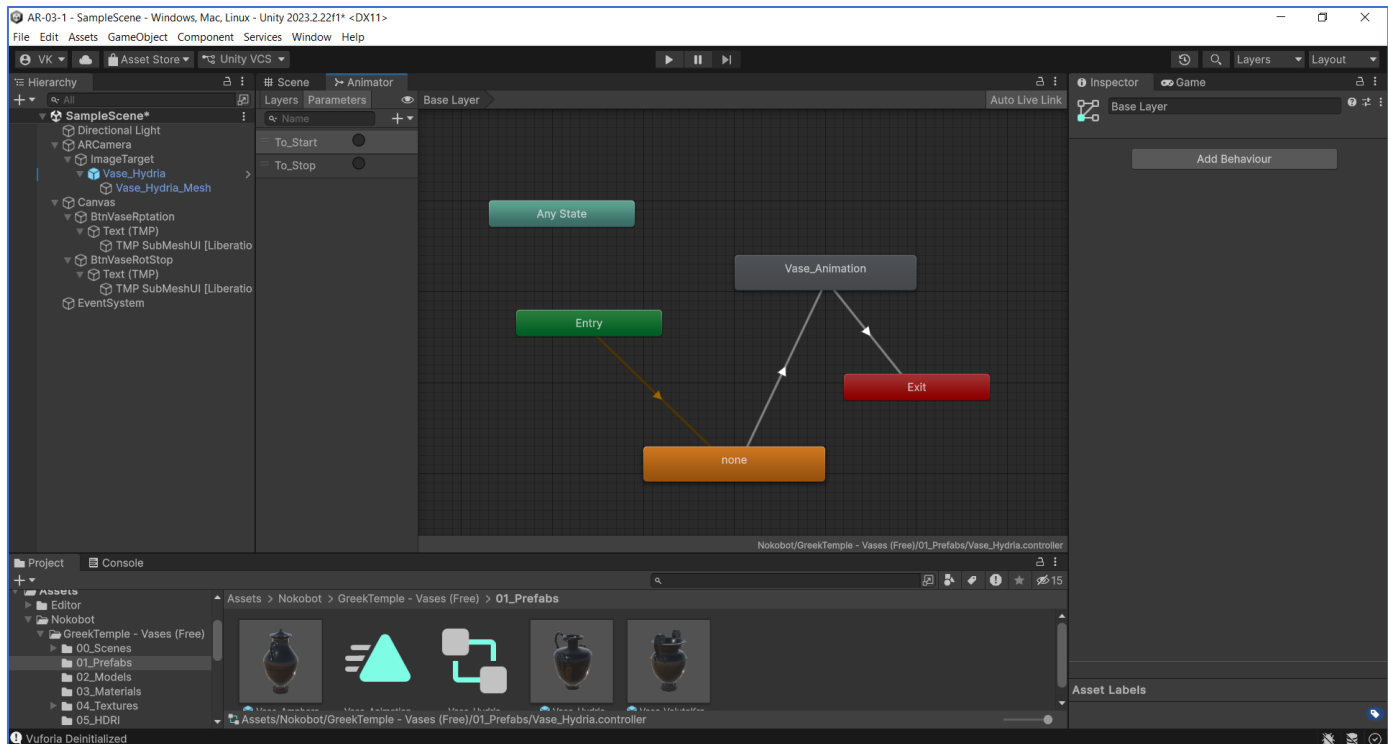


2. Добавление в плоское меню элемента управления – кнопка остановки анимации

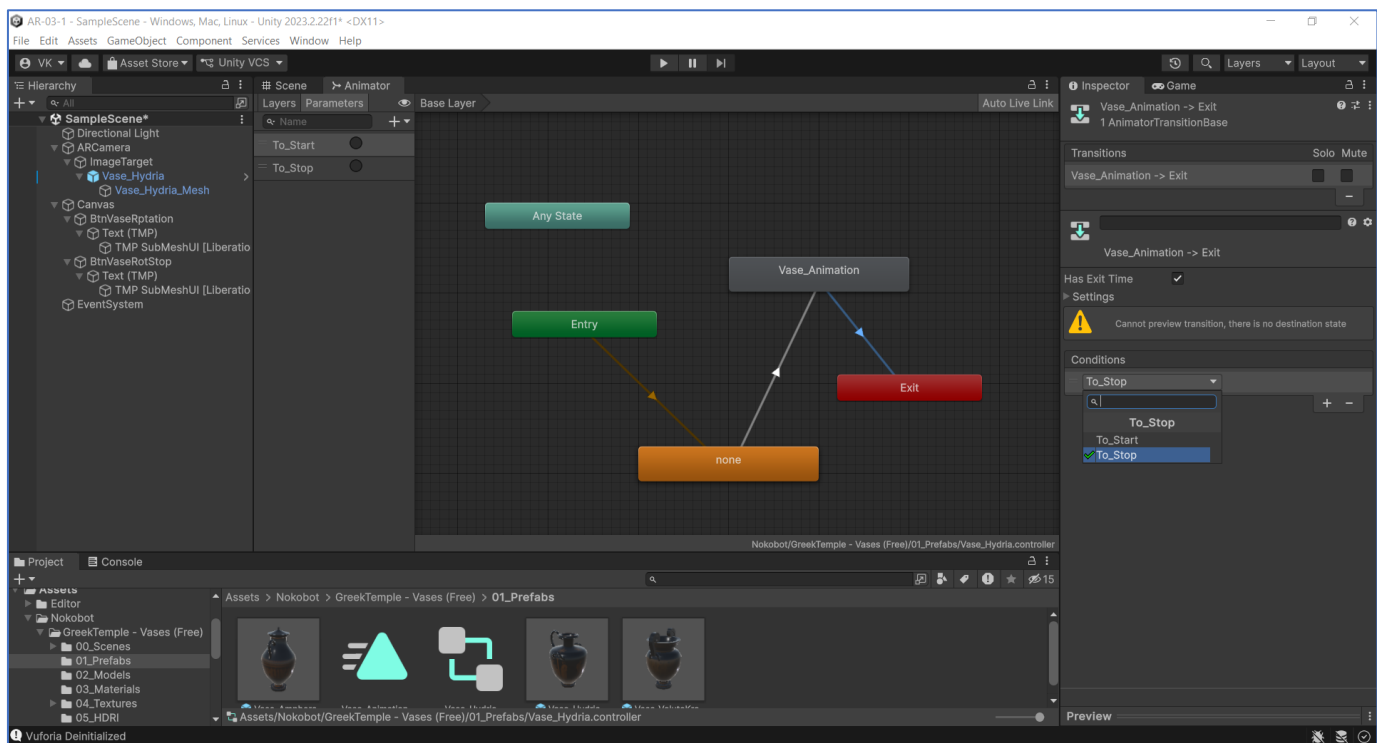
Добавим в плоское меню управления поведением объектов в Приложении ДР (анимация греческой вазы) еще один элемент управления – кнопка остановки вращения.

По аналогии с только что рассмотренными выполним необходимые шаги по модификации контроллера анимации (**Animator**) и добавления нового **триггера** срабатывания остановки.

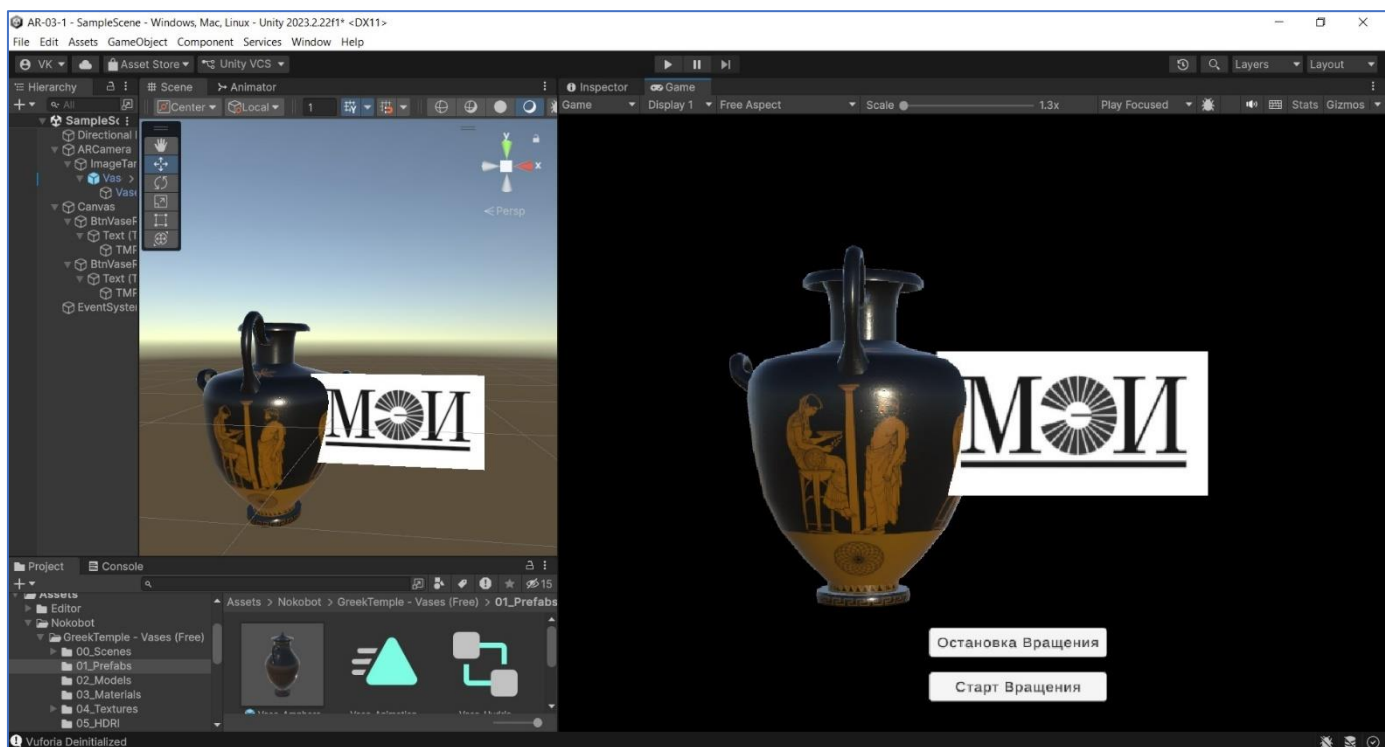
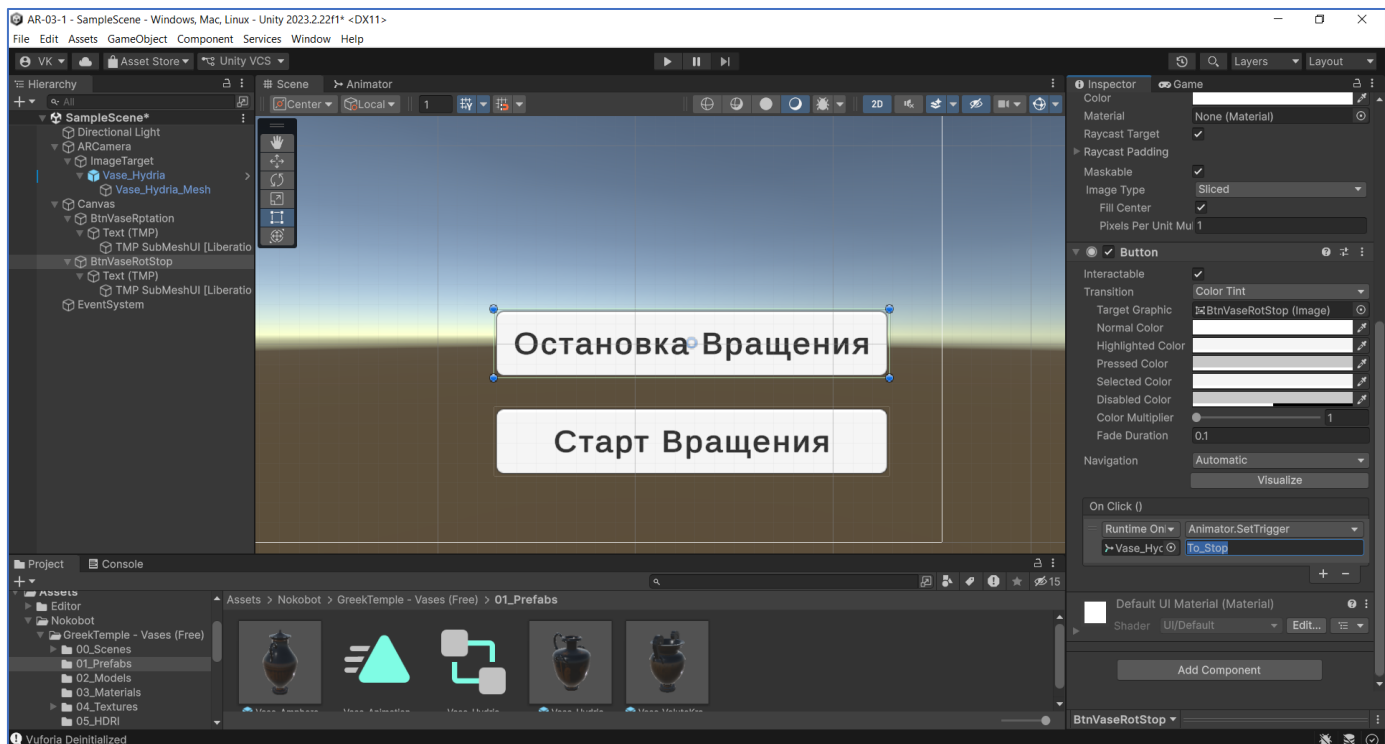
В контроллере (**Animator**) добавляем ещё один переход между состояниями (**Vase_Animation** → **Exit**) и вводим еще один триггер **To_Stop**:



На переход **Vase_Animation** → **Exit** подключаем триггер **To_Stop**:



В сцене создаем на канве ещё одну кнопку с окончательным именем в **Иерархии** **BtnVaseRotStop** и по аналогии с предыдущей настраиваем для неё **On Click ()** с тем только различием, что **SetTrigger** для неё устанавливаем в **To_Stop** для управления остановкой анимации:



Можно проверить работоспособность проекта в режиме Game и выполнить сборку Приложения ДР для загрузки на МУ (получение файла .apk).

Задание к ЛР№3, Часть2:

На данном этапе разработки плоского меню для управления объектами в сцене ДР добавьте кнопки старта и остановки выбранной анимации. Подготовьте файл .apk для загрузки на МУ – ТУ ДР.

Полученный результат продемонстрировать преподавателю. Составить короткий отчет об основных этапах работы.