

УДК 004

## Использование межплатформенной среды разработки UNITY для создания имитационных моделей

**Е.М. Гриценко, А.А. Абунагимов\***

Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева, просп. Мира, 82, г. Красноярск, 660049, Российская Федерация

E-mail: aa.abunagimov@yandex.ru

**Аннотация.** В настоящее время растет необходимость в быстром и качественном обучении специалистов разных областей. Компьютерные технологии помогают решать данную проблему при помощи визуализации, интерактивности. На основе игровых движков можно создавать имитационные модели, моделировать природные системы и системы человека, что дает возможность анализа процессов и их прогнозирования.

**Ключевые слова:** Симуляция, имитационная модель, Unity, Игровой движок, C#

## Using the UNITY cross-platform development environment to create simulation models

**E.M. Gritsenko, A.A. Abunagimov**

Reshetnev Siberian State University of Science and Technology, 82, prospekt Mira, Krasnoyarsk, 660049, Russian Federation

E-mail: aa.abunagimov@yandex.ru

**Abstract.** Currently, there is a growing need for fast and high-quality training of specialists in various fields. Computer technologies help to solve this problem with the help of visualization, interactivity. Based on game engines, it is possible to create simulation models, simulate natural systems and human systems, which makes it possible to analyze processes and predict them.

**Keywords:** simulation, simulation model, Unity, Game engine, C#

## 1. Введение

Компьютерные технологии плотно укрепились в жизни человека, изучение новых технологий, дистанционное образование, средство коммуникации, а также игровые приложения все это есть в обычном персональном компьютере.

В настоящее время игры создаются на основе игровых движков - программные обеспечения имеющие наборы инструментов, которые упрощают и ускоряют процесс создания приложения. Такие среды разработки существуют не только для реализации программного обеспечения с целью развлечь пользователя, но и для разработки образовательных программ и симуляторов.

В большинстве игровых движков уже имеется физический движок, что позволяет имитировать физические процессы с определенной точностью. Благодаря этому разработчик экономит большое количество времени, имея возможность сконцентрироваться на других важных аспектах разработки.

## 2. Имитационная модель (симуляция)

Имитационное моделирование - отображает логику и закономерности поведения моделируемой системы. В отличие от физического моделирования, такого как создание макетов и лабораторных стендов, имитационное моделирование основано на компьютерных технологиях, использующих разнообразные алгоритмы. Имитационную модель можно анализировать в динамике, а также просматривать визуализацию процессов в 2D и 3D.

Компьютерное моделирование используется в науке, когда проведение экспериментов на реальной системе невозможно или непрактично, чаще всего из-за их стоимости, опасности или длительности.

Отличительная особенность имитационной модели в том, что она предоставляет возможность пользователю изучать и вносить изменения в процессе работы, что позволяет улучшить анализ системы.

Возможность анализировать модель в действии отличает имитационное моделирование от других методов, например, от использования Excel или линейного программирования. Пользователь изучает процессы и вносит изменения в имитационную модель в ходе работы, что позволяет лучше проанализировать работу системы и быстро решить поставленную задачу.

## 3. Unity

В качестве средства для создания имитационных моделей, нами предлагается использовать межплатформенную среду разработки Unity.

Unity разработанный компанией Unity Technologies, является очень популярным игровым движком, так как прост в освоении и не требует большой команды, чтобы реализовать проект. В данной среде разработки можно создавать 2D и 3D приложения, поддерживает множество платформ: Windows, Linux, MacOS, Playstation, Xbox, Nintendo, Android, IOS. Unity имел свой язык программирования UnityScript, но в 2017 году он перестал поддерживаться, на данный момент скрипты в Unity пишутся на C#.

Unity имеет гибкий интерфейс, элементы которого можно перемещать, растягивать, убирать. Расположение элементов можно сохранять или использовать готовые варианты. Ниже приведены основные используемые вкладки (инструменты):

- Hierarchy (Иерархия), окно, где отображаются игровые объекты;
- Project (Проект), окно, предоставляющее доступ к файлам проекта;
- Console (Консоль), окно, где выводятся ошибки, предупреждения. Пользователь может через скрипт вывести сообщение в консоль при помощи метода Debug.Log();
- Inspector (Инспектор) отображает информацию о компонентах и их свойствах находящихся на игровом объекте;
- Scene (Сцена), визуальное пространство проекта;
- Game (Игра), при запуске проекта именно в этом окне будет отображаться игровой процесс;
- Animation (Анимация), состоит из трех инструментов: Animation (покадровая анимация), Animator (переходы между анимациями), Animator parameter (параметры анимации);
- Package Manager (Менеджер пакетов),

На рисунке 1 представлена рабочая область Unity версии 2021.1.25f1.

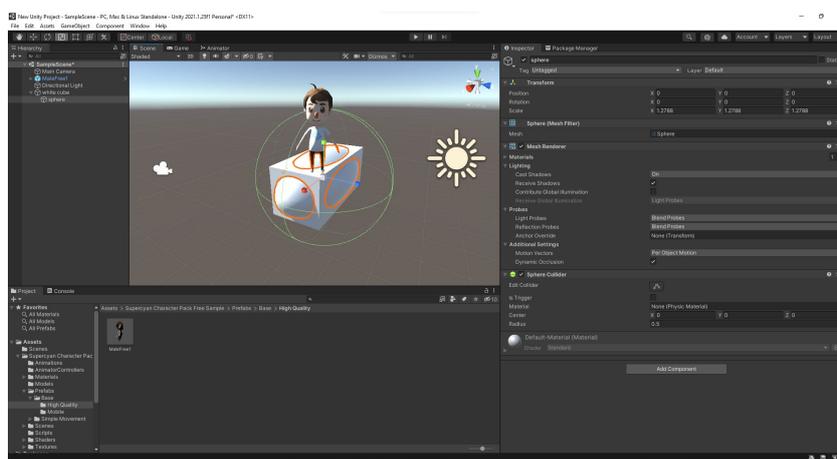


Рисунок 1. Рабочая область Unity версии 2021.1.25f1.

Unity поддерживает физику твердых тел, ткани и Ragdoll (твердое тело находящиеся в системе скелетной анимации графического движка) - для реалистичного перемещения твёрдых тел, на последние воздействуют сила вращения и другие силы. Любой объект проекта должен содержать в себе твёрдое тело, чтобы быть подверженным гравитации, действовать согласно назначенным путём скриптинга (scripting; от англ. script - сценарий) силам, или взаимодействовать с другими объектами через физический движок NVIDIA PhysX.

Среда разработки имеет внутренний магазин - Asset Store куда можно выкладывать готовые скрипты, объекты, сцены, звуки, шейдеры/частицы, инструменты и так далее. Так же есть возможность скачать бесплатно или купить любой понравившийся ассет.

Unity ускоряет время разработки и ее итерацию, позволяя видеть результат изменений в реальном времени. Игровой движок позволяет создавать 2D- и 3D-сцены, анимации и катсцены.

Существует возможность визуального программирования при помощи Bolt. Используя его для разработки и работы над вашим приложением исчезает надобность писать код.

Визуальный скриптинг — это способ манипуляции объектами и разработки алгоритмов в графическом интерфейсе Unity без написания кода. Логика реализуется узлами (нодами) и связями между ними, что позволяет художникам, дизайнерам и программистам с легкостью прорабатывать игровой процесс и интерактивные системы.

Систему Bolt Visual Scripting можно скачать совершенно бесплатно с Unity Asset Store. На рисунке 2 представлена рабочая область инструмента.

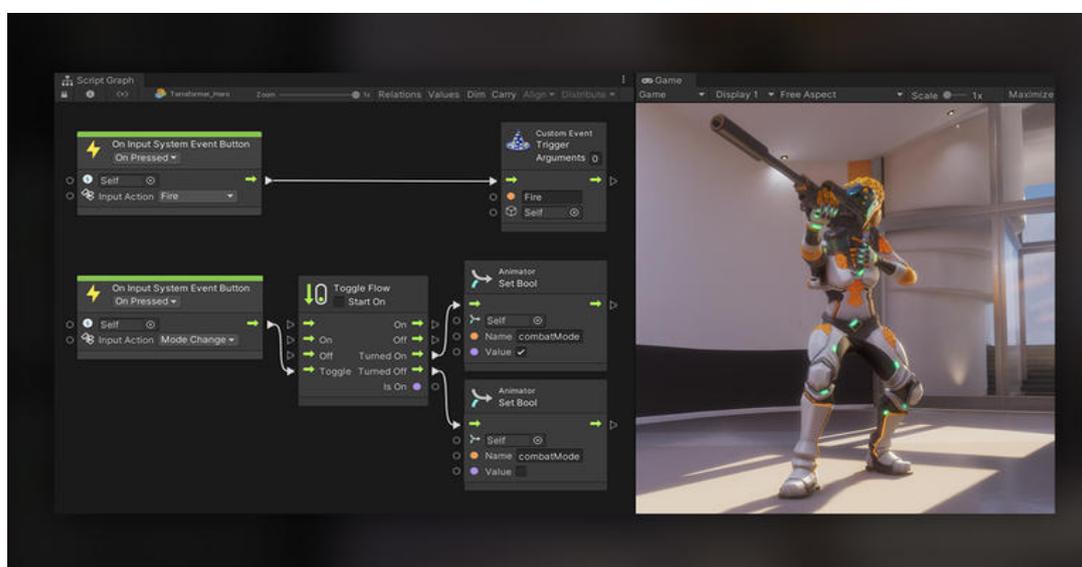


Рисунок 2. Рабочая область Bolt.

#### 4. Заключение

Благодаря компьютерным симуляциям люди могут удобно и безопасно наблюдать за имитационной системой, а также влиять на нее в процессе работы и прогнозировать поведение системы в реальной жизни.

Unity обладает огромным инструментарием для создания имитационных моделей, он имеет удобный и простой интерфейс, что ускоряет процесс разработки. Нет необходимости в создании компьютерной симуляции с нуля, такой процесс требует много времени и человеческих сил.

#### Список литературы

1. Unity - Manual: Unity User Manual (2020.3) // Unity Technologies. <https://docs.unity3d.com/Manual/>.
2. Unity Game Development Essentials // Will Goldstone. – 2009. – 316 с. <https://studfile.net/preview/2989929/>.
3. Unity visual scripting // Unity Technologies. <https://unity.com/ru/products/unity-visual-scripting>
4. Моделирование и симуляция - Краткое руководство // CoderLessons. – 2018. <https://coderlessons.com/tutorials/akademicheskii/modelirovanie-i-simulatsiia/modelirovanie-i-simulatsiia-kratkoe-rukovodstvo>.