

М. А. Мищенко^{1✉}, Н. С. Головачев¹

Разработка симулятора настройки сборочного конвейера для обучения программированию и автоматизации на виртуальной модели производственного участка

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: mmisha2702@gmail.com

Аннотация. В работе представлены результаты разработки учебного приложения «Автоматика», предназначенного для начального освоения программирования производственных манипуляторов на виртуальной модели производственного участка. Актуальность темы обусловлена высокой стоимостью реального робототехнического оборудования, требованиями к безопасности и ограниченной доступностью практических занятий для школьников. Цель работы состоит в создании доступного симулятора, позволяющего в безопасной среде изучать базовые операции автоматизации и логику управления манипулятором. В ходе исследования выполнен анализ существующих образовательных и инженерных симуляторов, сформулированы требования к пользовательскому сценарию, разработана структура уровней и реализована программная система на Unity с поддержкой пользовательских Lua-скриптов. В результате создан прототип, включающий пять учебных уровней, консоль программирования, встроенную справку, журнал сообщений и задания с конвейером. Разработанное решение может использоваться для профориентации и вводного обучения школьников.

Ключевые слова: учебный симулятор, промышленный манипулятор, программирование, автоматизация, Unity, Lua

M. A. Mishchenko^{1✉}, N. S. Golovachev¹

Development of an assembly line setup simulator for programming and automation training on a virtual model of a production site

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: mmisha2702@gmail.com

Abstract. The paper presents the results of developing the Automatica training application intended for introductory learning of industrial manipulator programming on a virtual model of a production area. The relevance of the study is determined by the high cost of real robotic equipment, safety requirements, and limited access to hands-on practice for school students. The aim was to create an accessible simulator that allows learners to study basic automation operations and manipulator control logic in a safe digital environment. The study included an analysis of existing educational and engineering simulators, formulation of user-scenario requirements, design of level structure, and implementation of a software system in Unity with support for user Lua scripts. The resulting prototype includes five training levels, a programming console, built-in reference materials, a message log, and conveyor-based tasks. The developed solution can be used for career guidance and introductory training in industrial automation.

Keywords: training simulator, industrial manipulator, programming, automation, Unity, Lua

Введение

Автоматизация производственных процессов и использование промышленных манипуляторов становятся базовой частью современной промышленности. Вместе с этим растет потребность в раннем знакомстве обучающихся с принципами программирования оборудования и логикой автоматизированных операций. Исследования по инженерному образованию показывают, что цифровые симуляторы позволяют организовать безопасную, повторяемую и сравнительно недорогую практику, особенно в тех случаях, когда работа с реальным оборудованием требует значительных затрат или связана с риском [1–3].

Для школьного обучения проблема особенно заметна. Реальные манипуляторы и учебные роботизированные ячейки дороги в закупке, требуют обслуживания и квалифицированного сопровождения, а ошибки начинающих пользователей способны привести к поломкам или опасным ситуациям. При этом современные виртуальные и смешанные среды обучения подтверждают, что даже сложные понятия кинематики, планирования траекторий и взаимодействия с оборудованием могут быть поданы в более наглядной форме [4, 5].

Целью работы является разработка симулятора настройки сборочного конвейера для обучения школьников базовым навыкам программирования производственного манипулятора. Для достижения цели были поставлены следующие задачи: проанализировать существующие аналоги; сформулировать требования к учебному сценарию; спроектировать структуру уровней и пользовательский интерфейс; реализовать программное управление манипулятором на основе текстовых скриптов; выполнить предварительное тестирование прототипа.

Исследование предметной области

Для определения функциональных требований к приложению были рассмотрены как образовательные, так и профессиональные решения в области робототехнического моделирования. К числу образовательных продуктов относятся Promobot M Edu, RoboSim и RoboMind [6–8]. Они ориентированы на обучение основам программирования и робототехники, однако используют различный подход к вовлечению пользователя: от практики на реальном оборудовании до работы в полностью виртуальной среде. Профессиональные платформы Gazebo и CoppeliaSim [9, 10] обеспечивают более широкий набор средств для моделирования, но рассчитаны прежде всего на инженерную и исследовательскую деятельность.

Наиболее близкими к тематике разрабатываемого приложения являются Promobot M Edu и RoboSim. Первый продукт ценен высокой связью с реальным оборудованием и позволяет формировать представление о физических ограничениях промышленного робота. Второй обеспечивает более доступный вход в обучение, так как не требует наличия роботизированной установки и дает возможность быстро переходить к выполнению учебных сценариев на обычном персональном компьютере. RoboMind, в свою очередь, полезен как пример удачной подачи алгоритмического мышления для начинающих пользователей, но слабее

связан именно с тематикой промышленной автоматизации и работы манипулятора на производственном участке.

С методической точки зрения образовательные аналоги демонстрируют несколько сильных сторон, которые представляют интерес для дальнейшего развития «Автоматики». У Promobot M Edu важным преимуществом является реалистичность учебного контекста, однако зависимость от физического оборудования ограничивает массовое применение такого решения в школьной среде. RoboSim выигрывает за счет доступности, понятного входа и ориентации на учебный процесс, но в большей степени охватывает робототехнику в целом, а не узкий сценарий программирования производственного манипулятора. RoboMind удачно использует постепенное усложнение и низкий порог входа, что особенно важно для школьников; именно эти качества целесообразно усиливать и в рассматриваемом проекте через короткие задания, шаблоны решений и быструю обратную связь по ошибкам.

Профессиональные симуляторы Gazebo и CoppeliaSim показывают другой полюс развития подобных систем. Их сильными сторонами являются поддержка сложной физики, гибкая работа с моделями роботов, возможность расширения под исследовательские задачи и сценарии цифровых двойников. Вместе с тем для целевой аудитории школьников такие платформы оказываются слишком сложными из-за насыщенного интерфейса, большого числа настроек и необходимости знания дополнительных инструментов. Тем не менее отдельные их достоинства могут быть интегрированы в «Автоматику» на следующем этапе: визуализация траектории манипулятора, более подробный анализ ошибок, редактор сценариев и автоматическая проверка решений.

На основе сравнительного анализа были сформулированы ключевые требования к проекту: низкий порог входа, ограниченный и понятный набор команд, ориентация на один тип манипулятора, постепенное усложнение задач, работа на персональном компьютере без специализированного оборудования и наличие встроенной справки. Таким образом, концепция приложения определяется не стремлением заменить профессиональный инженерный симулятор, а задачей создать доступную учебную среду для раннего погружения в промышленную автоматизацию.

Материалы и методы

Программная реализация выполнена в среде Unity 6 [11]. Для построения интерфейса использованы стандартные средства UGUI и пакет TextMeshPro [12]. В качестве языка пользовательских сценариев выбран Lua, хорошо подходящий для компактных учебных скриптов [13]. Выполнение скриптов внутри приложения обеспечивается интерпретатором MoonSharp [14]. Для анимации интерфейсных элементов применен DOTween [15], а для части асинхронных операций пользовательского интерфейса и сценической логики - UniTask [16].

Приложение построено по сценовой модели и включает главное меню и пять учебных уровней. Общая структура проекта продемонстрирована на рис.1. Каждая сцена содержит виртуальный производственный участок с манипулято-

ром, набором объектов, зонами контроля и условиями завершения задания. Для объектов предусмотрены свойства типа и качества, что позволяет переходить от простых задач переноса к заданиям по отбору деталей. Управление манипулятором выполняется через консоль, в которой пользователь вводит текст программы и запускает ее выполнение.

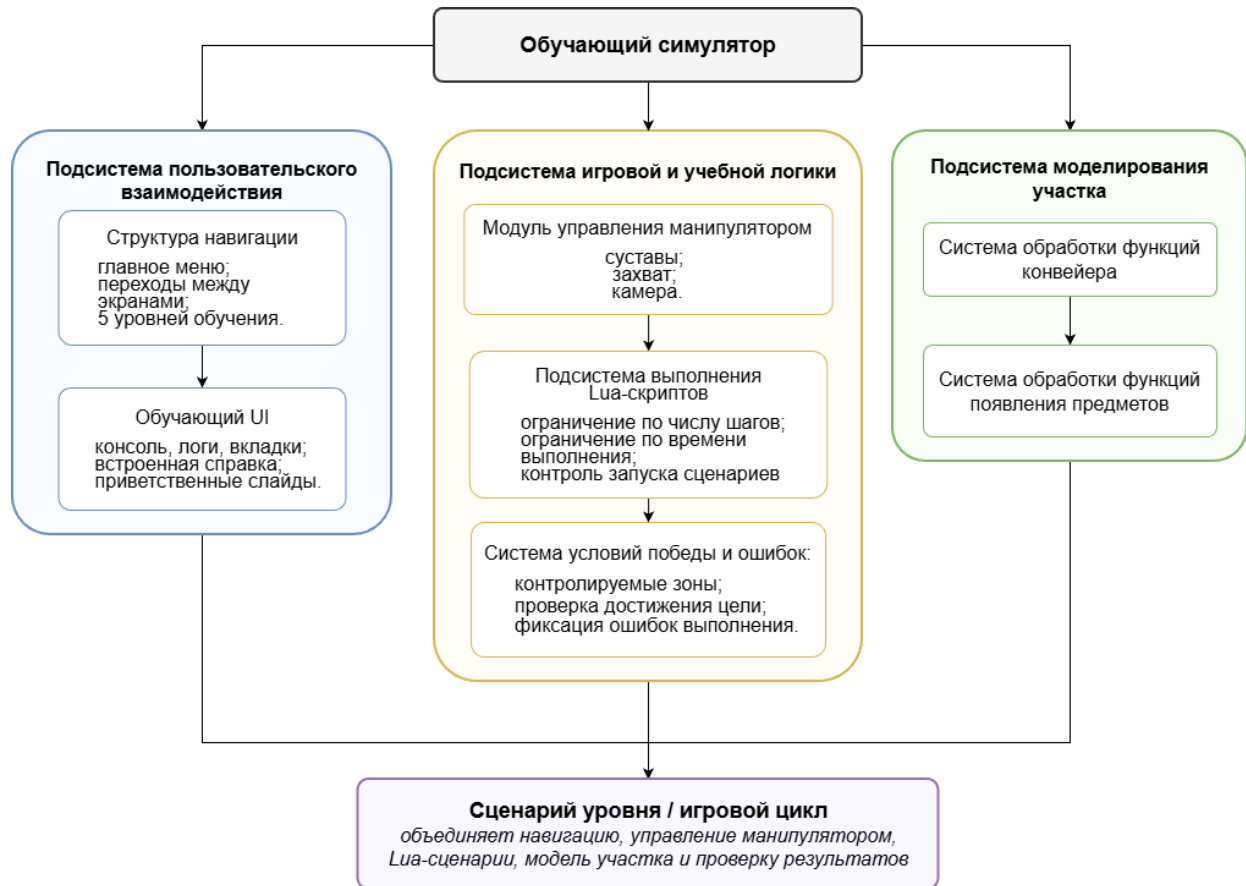


Рис. 1. Структура системы проекта

Учебный сценарий был спроектирован вокруг набора базовых команд: поворот суставов на заданный угол, захват и отпускание объекта, проверка состояния захвата, а также задание белых и черных списков по типам и качеству деталей. Такой подход позволяет сосредоточить внимание не на сложной математической модели, а на логике действий, последовательности операций и обработке ограничений. Отдельное внимание уделено встроенной справке, журналу сообщений и визуальному контролю выполнения задания, поскольку эти элементы напрямую влияют на понятность обучения для начинающего пользователя.

Проверка прототипа проводилась методом ручного тестирования на каждой сцене с контролем прохождения уровней, устойчивости выполнения скриптов и корректности интерфейсных элементов. Оценивались успешность выполнения учебных сценариев, корректность обработки ошибочных команд, удобство работы с консолью и справочными окнами, а также целостность визуального интерфейса на разных уровнях.

Результаты

В результате работы создан учебный прототип «Автоматика», ориентированный на школьников, делающих первые шаги в сфере промышленной автоматизации. Обучение построено по принципу поэтапного усложнения. На первом уровне пользователь переносит один куб с одного ящика на другой. На втором уровне количество объектов увеличивается до двух. На третьем уровне добавляются различие типов и требование соблюдать заданный порядок укладки. Четвертый уровень вводит в сценарий конвейер, а пятый - отбор объектов по качественным признакам.

Важным результатом стала реализация текстовой консоли программирования, объединенной со встроенной справкой и журналом сообщений. Пользователь может выбирать манипулятор, вводить скрипт, запускать его и получать диагностические сообщения при ошибках. В справочном окне представлены доступные команды, ограничения и примеры применения. Такой формат сближает приложение с реальной практикой программирования оборудования, но при этом сохраняет учебную понятность интерфейса.

Кроме консоли, в приложении реализованы вспомогательные механики, повышающие удобство обучения: переключение между несколькими камерами, панель задания, перезагрузка уровня и визуальный контроль состояния сцены. На поздних уровнях пользователь может применять фильтрацию деталей с помощью белых и черных списков по типам и качеству. Это расширяет учебный сценарий от простого перемещения объектов до принятия логических решений внутри программы.

Интерфейс симулятора показан на рис. 2. На изображении видны виртуальный производственный участок, окно написания программы и встроенная справка по доступным командам. Такое совмещение пространства сцены и инструментов программирования позволяет учащемуся сразу соотносить написанный код с поведением манипулятора и результатом на конвейере.

По итогам предварительного тестирования были выявлены и устранены отдельные недочеты интерфейса и сценарной логики. В частности, был увеличен допустимый интервал работы скрипта, устранены ошибки прокрутки текстовых областей и переработано оформление элементов управления. Исправления позволили повысить устойчивость прототипа и целостность восприятия интерфейса при прохождении уровней.

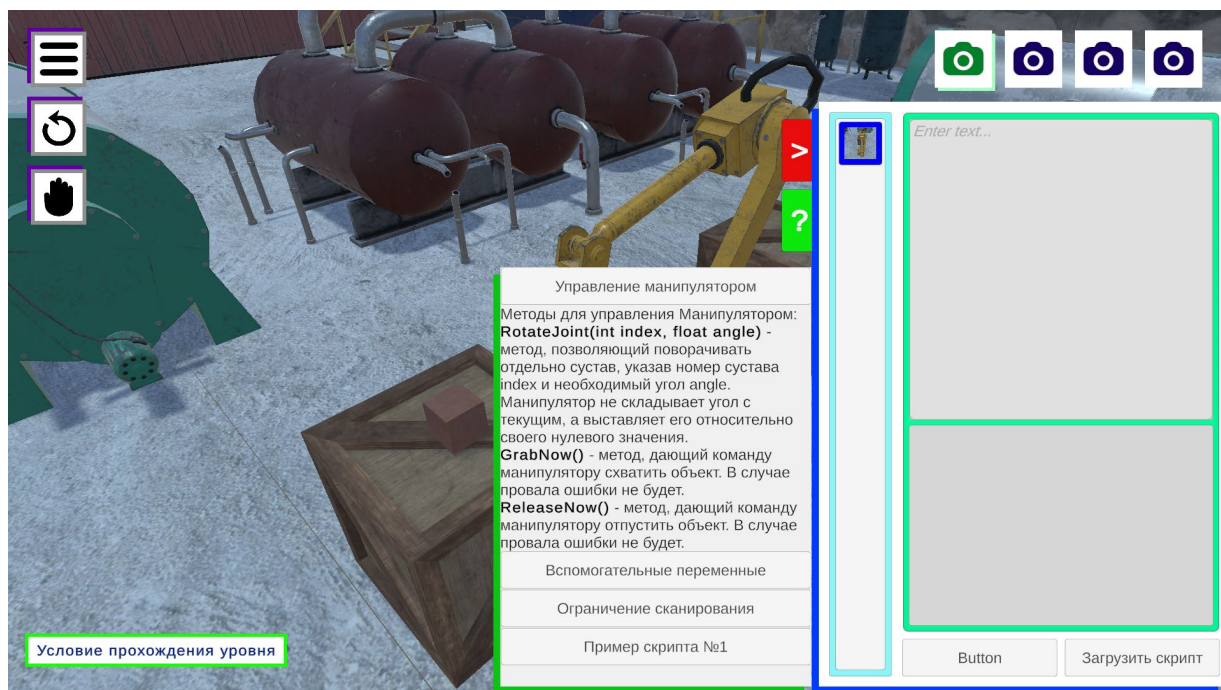


Рис. 2. Интерфейс симулятора «Автоматика»

Обсуждение

Полученный результат занимает промежуточное положение между простыми средами для начального знакомства с программированием и сложными профессиональными симуляторами. В отличие от решений, ориентированных на работу с реальным учебным оборудованием, разработанный прототип не требует закупки роботизированной установки и снижает риск повреждения техники. По сравнению с профессиональными платформами Gazebo и CoppeliaSim проект сознательно ограничивает функциональность и уровень физической детализации, поскольку его основная задача заключается не в точном моделировании производственной ячейки, а в раннем погружении в логику программирования манипулятора.

Одновременно проект имеет и ограничения. На текущем этапе не реализованы расширенные средства анализа результатов, редактор пользовательских уровней, автоматическая проверка решений и высокоточная кинематическая или динамическая модель оборудования. Перспективными направлениями развития являются расширение библиотеки заданий, добавление сценариев с несколькими манипуляторами, внедрение системы оценки действий учащегося и интеграция с более подробной моделью производственного процесса.

Заключение

Разработанный учебный симулятор позволяет в безопасной и доступной форме познакомить школьников с базовыми принципами программирования производственных манипуляторов и автоматизации операций на конвейере. Реализованный прототип содержит последовательный учебный сценарий, средства написания и отладки скриптов, а также набор заданий, постепенно усложняю-

ших логику взаимодействия с оборудованием. Практическая значимость работы заключается в возможности использования приложения для профориентации, демонстрационных занятий и вводного обучения без необходимости применения дорогостоящего реального оборудования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Negahban, A. Simulation in engineering education: The transition from physical experimentation to digital immersive simulated environments // *Simulation*. 2024. Vol. 100, No. 7. DOI: 10.1177/00375497241229757.
2. Strojny, P., Duzmanska-Misiarczyk, N. Measuring the effectiveness of virtual training: A systematic review // *Computers & Education: X Reality*. 2023. Vol. 2. Art. 100006. DOI: 10.1016/j.cexr.2022.100006.
3. Shamsuzzoha, A., Toshev, R., Vu Tuan, V., Kankaanpaa, T., Helo, P. Digital factory-virtual reality environments for industrial training and maintenance // *Interactive Learning Environments*. 2021. Vol. 29, No. 8. P. 1339-1362. DOI: 10.1080/10494820.2019.1628072.
4. Nathanael, D. et al. Development and evaluation of a virtual training environment for on-line robot programming // *International Journal of Industrial Ergonomics*. 2016. Vol. 53. P. 274-283. DOI: 10.1016/j.ergon.2016.02.004.
5. Tarng, W., Wu, Y.-J., Ye, L.-Y. et al. Application of Virtual Reality in Developing the Digital Twin for an Integrated Robot Learning System // *Electronics*. 2024. Vol. 13, No. 14. Art. 2848. DOI: 10.3390/electronics13142848.
6. Promobot M Edu. URL: <https://promo-bot.ai/promobot-m-edu/> (дата обращения: 13.05.2026).
7. Робосим. URL: <https://robosim.io/> (дата обращения: 13.05.2026).
8. RoboMind. URL: <https://www.robomind.net/> (дата обращения: 13.05.2026).
9. Gazebo. Getting Started. URL: <https://gazebo.org/docs/latest/getstarted/> (дата обращения: 13.05.2026).
10. CoppeliaSim. URL: <https://www.coppeliarobotics.com/> (дата обращения: 13.05.2026).
11. Unity User Manual 6000.0. URL: <https://docs.unity3d.com/6000.0/Documentation/Manual/> (дата обращения: 13.05.2026).
12. TextMeshPro package documentation. URL: <https://docs.unity3d.com/Packages/com.unity.textmeshpro@3.2/manual/index.html> (дата обращения: 13.05.2026).
13. Lua 5.4 Reference Manual. URL: <https://www.lua.org/manual/5.4/> (дата обращения: 13.05.2026).
14. MoonSharp - A Lua interpreter written entirely in C#. URL: <https://www.moonsharp.org/> (дата обращения: 13.05.2026).
15. Demigiant. DOTween. URL: <https://github.com/Demigiant/dotween> (дата обращения: 13.05.2026).
16. Cysharp. UniTask. URL: <https://github.com/Cysharp/UniTask> (дата обращения: 13.05.2026).

© М. А. Мищенко, Н. С. Головачев, 2026