

Паттерны проектирования усилили архитектуру. Классы GameController и HighscoreTable, реализованные как одиночки (Singleton), обеспечили глобальный доступ к управлению игрой и таблице рекордов [3]. Событийная модель (Observer) позволила компонентам, таким как UIController, реагировать на изменения счета или завершение игры без прямых зависимостей. Для оптимизации применялся пул объектов (Object Pooling), что сократило нагрузку при частом создании/уничтожении снарядов.

**Заключение.** Разработка игры «Астероиды» подтвердила эффективность применения объектно-ориентированного подхода в создании современных игровых проектов. Благодаря использованию принципов наследования, инкапсуляции, полиморфизма и абстракции удалось создать модульную и масштабируемую архитектуру, которая не только сохранила ретро-дух классической версии, но и обогатила её инновационными элементами.

Ключевым достижением стала реализация гибкой системы классов, где базовые сущности обеспечили единую логику для движения, обработки урона и взаимодействия объектов.

Практическая значимость работы заключается не только в создании игрового продукта, но и в демонстрации лучших практик ООП и геймдизайна. Разработанная архитектура может служить основой для более сложных проектов, включая сетевые режимы или VR-версии.

#### Список цитируемых источников

1. Хокинг, Д. Unity в действии. Мультиплатформенная разработка на C#. / Д. Хокинг. — 3-е межд. изд. — СПб.: Питер, 2023. — 528 с.
2. Ферроне, Х. Изучаем C# через разработку игр на Unity. / Х. Ферроне. — 5-е изд. — СПб.: Питер, 2022. — 464 с.
3. Фримен, Э. Паттерны проектирования. Введение в проектирование программного обеспечения / Э. Фримен, Э. Робсон, Б. Бейтс и др. — СПб.: Питер, 2022. — 672 с.

УДК 004.94

А. Е. Осипова, А. К. Крамаренко

Учреждение образования «Брестский государственный технический университет»,  
Брест, Республика Беларусь

## ТЕХНОЛОГИИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

**Введение.** Моделирование является одним из способов решения множества практических задач. В случаях, когда проблемы нельзя решить путём проведения экспериментов в реальности, например, вносить изменения или разрушать уже имеющуюся инфраструктуру, тогда можно построить модель реальной системы, учитывая все существенные свойства объекта (процесса) для того, чтобы максимально приблизить свойства модели к характеристикам реального объекта. Имитационное моделирование направлено на исследование сложных объектов и различных процессов при помощи их моделей, эквивалентов и симуляции.

В данной работе рассмотрены такие вопросы, как понятие имитационного моделирования, его основные виды и их применение, основные преимущества данного метода.

**Основная часть.** Имитационной моделью называется специальный программный комплекс, позволяющий имитировать деятельность какого-либо сложного объекта, он выполняет на компьютере параллельно взаимодействующие процессы, которые являются по своим временным параметрам (с точностью по масштабам времени и пространства) аналогами исследуемых процессов.

Имитационное моделирование — это частный случай математического моделирования, логико-математическое описание объекта. При имитационном моделировании логическая структура моделируемой системы адекватно отображается в модели, а процессы ее функционирования, динамика взаимодействия ее элементов воспроизводятся (имитируются) на модели. Поэтому построение имитационной модели включает структурный анализ моделируемой системы и разработку функциональной модели, отражающей динамические портреты моделируемой системы. Метод имитационного моделирования определяется, как экспериментальный подход к исследованию реальных систем по их имитационным моделям. Все компоненты, из которых она состоит, настраиваются автономно и могут представлять собой сложные подсистемы с собственными вложенными процессами, отражающими их изменчивое поведение [1].

Имитационное моделирование удобно для исследования практических задач: определение показателей эффективности, сравнение вариантов построения и алгоритмов функционирования систем, проверки устойчивости режимов системы при малых отклонениях входных переменных от расчётных значений. Полнота имитации может быть проверена путём построения серии последовательно уточняемых моделей. Если дальнейшая детализация свойств модели не влияет на конечные показатели, то усложнение модели можно прекратить. Как правило, моделируются те свойства процесса, которые могут влиять на выбранный показатель эффективности или критичны к наложенным ограничениям. Промежуточные результаты имитационного моделирования имеют четкий физический смысл и позволяют обнаружить ошибки программы [3].

Существуют ключевые принципы, которые необходимо учитывать при создании имитационной модели:

1. Время разделяется на некоторые интервалы, в каждом из которых происходит некоторые действия, это даёт возможность изучать изменения, происходящие в системе с течением времени и прогнозировать её будущее поведение в симуляции.

2. Учитываются факторы, которые могут оказать влияние на объект или процесс

3. В ходе работы модели происходит валидация — сравнение результатов моделирования с реальными данными.

4. При моделировании проводят различные эксперименты, проверяют гипотезы, исследуют стратегии и решения, анализируют полученные результаты, что позволяет оптимизировать параметры модели.

Имитационное моделирование представлено различными направлениями (рисунок 1), которые имеют свои методологии:



Рисунок 1 — Направления имитационного моделирования

Статистическое моделирование: модели здесь строятся для явлений и систем объектов, входы и (или) функциональные соотношения между различными компонентами которых содержат элементы случайности или полностью случайных процессов, подчиняющиеся вероятностным законам. Дискретно-событийный подход к имитационному моделированию ориентирован на бизнес-процессы и подразумевает их описание в виде последовательности отдельных событий или операций. Системная динамика применяется, когда нужно оценить явление или объект в целом, без детального анализа взаимодействия отдельных его элементов или единичных событий. В рамках подхода сложные системы или объекты моделируются на высоком уровне абстракции, индивидуальные свойства и другие детали конкретных объектов не принимаются во внимание. Зато можно смоделировать и тем самым систематизировать сложные, глобальные взаимосвязи. Агентное моделирование используется в ситуациях, когда легче и быстрее описать индивидуальное поведение каждого объекта, чем проектировать общий рабочий процесс. В отличие от вышеперечисленных методов, агентное моделирование помещает в центр внимания индивидуальных участников системы, учитывает их уникальные свойства и отношения [2].

Наиболее распространённые сферы применения имитационного моделирования: бизнес-процессы, системы поддержки принятия решений, ИТ-инфраструктура, дорожное движение, логистика, промышленность, управление проектами, прогнозирование и оценка финансовых рынков, экономика здравоохранения.

В обобщенном виде процесс разработки имитационной модели состоит из следующих шагов:

1. Изучение прототипа модели — реального объекта, его внутренней структуры, принципов работы, характеристик, взаимосвязей между элементами.

2. Разработка концептуальной модели, определяющей значимые для моделирования аспекты: состав и структуру объекта, свойства и связи элементов.

3. Формирование информационного пространства для моделирования, определение численных параметров функционирования объекта.

4. Выбор системы моделирования, функционал которой достаточен для реализации модели.

5. Разработка модели.

6. Тестирование и корректировка модели.

На рынке предоставлен широкий диапазон программного обеспечения для имитационного моделирования:

– для дискретного моделирования — системы, основанные на описании процессов: процессно-транзактно-ориентированные системы моделирования блочного типа — (Extend, Arena, ProModel, и др.);

– системы, основанные на сетевых концептах, применяются при структуризации причинных связей и моделировании систем с параллельными процессами;

– для систем, ориентированных на непрерывное моделирование — модели и методы системной динамики — Powersim, Vensim, Dynamo, Stella, Ithink и др.;

– динамические системы (MATLAB);

– агентное моделирование (AnyLogic).

**Заключение.** Имитационное моделирование активно развивается и открывает новые возможности для автоматизации и оптимизации работы различных объектов и процессов. Мы наблюдаем, что компании начинают использовать эти технологии не только для улучшения текущих процессов, но и для создания новых бизнес-моделей, которые основаны на данных и виртуальных симуляциях.

Это открывает путь к созданию «умных» производств, где цифровая модель постоянно адаптируется к изменениям в реальном времени.

Технологии имитационного моделирования помогают снижать затраты, минимизировать отходы и оптимизировать использование ресурсов, что является важным аспектом в условиях растущих экологических требований.

Имитационное моделирование становится неотъемлемой частью современного производства. Оно позволяет значительно улучшить эффективность и гибкость процессов, создавая основу для будущих инноваций и устойчивого развития. Важно продолжать исследовать эти направления и внедрять их в практику.

#### Список цитируемых источников:

1. Эльберг, М. С. Имитационное моделирование : учеб. пособие / М. С. Эльберг, Н. С. Цыганков. — Красноярск : Сиб. федер. ун-т, 2017 — 128 с.
2. Лёвина А. П. Обзор метода имитационного моделирования // Современная техника и технологии. — 2017. — № 5. — URL: <https://technology.snauka.ru/2017/05/13530> (дата обращения: 03.06.2025).
3. Имитационное моделирование: конспект лекций / Н. Л. Леонова. — СПбГТУРП. — СПб., 2015 — 94 с.

УДК 004.35

А. А. Пачук

*Учреждение образования «Барановичский государственный университет»,  
Барановичи, Республика Беларусь*

## ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ИНТЕРАКТИВНОЙ ИГРЫ ДЛЯ ТРЕНИРОВКИ ПАМЯТИ НА ПЛАТФОРМЕ ARDUINO

**Введение.** Игры на повторение последовательности, такие как знаменитая электронная игра «Саймон», давно используются для тренировки памяти, внимания и скорости реакции [1]. Суть подобных игр заключается в том, что устройство генерирует определённую последовательность сигналов (например, мигание светодиодов или звуковые сигналы), а игрок должен точно воспроизвести эту последовательность с помощью кнопок или других элементов управления [2]. Такие игры находят широкое применение не только в сфере развлечений, но и в образовательных целях, поскольку способствуют развитию когнитивных способностей и моторики [3].

В данном исследовании рассматривается создание электронной версии игры «Повтори последовательность», в которой пользователь должен запомнить и повторить заданную устройством комбинацию сигналов. Проект направлен на демонстрацию принципов построения интерактивных обучающих устройств и алгоритмов генерации и проверки последовательностей.

Цель исследования: создание интерактивной игры «Повтори последовательность» на базе одноплатного компьютера Arduino Uno с использованием среды программирования Arduino IDE [4]. В рамках исследования предполагается освоить возможности Arduino IDE, изучить основные библиотеки для работы с кнопками и светодиодами, а также реализовать устройство, способное генерировать и проверять последовательности нажатий для пользователя.

Объектом исследования выступает процесс проектирования и программирования электронных устройств на платформе Arduino, а также методы создания интерактивных обучающих и развивающих игр с использованием светодиодов и кнопок.

**Основная часть.** Устройство для игры «Повтори последовательность» представляет собой электронную систему, специально разработанную для формирования и проверки последовательностей световых сигналов. Принцип работы основан на использовании микроконтроллера Arduino и набора светодиодов и кнопок, что позволяет точно контролировать процесс генерации и повторения последовательностей пользователем.

Основные принципы работы устройства:

1. Микроконтроллер Arduino генерирует случайную последовательность мигания светодиодов, которую необходимо запомнить.
2. Пользователь воспроизводит увиденную последовательность, нажимая соответствующие кнопки, каждая из которых связана с определённым светодиодом.
3. После ввода последовательности устройство сравнивает введённый пользователем порядок с исходным и выводит результат (например, индикация правильности с помощью светодиода или звукового сигнала).

На рисунке 1 изображена модель устройства.