

Результаты модели, построенной на агентном подходе, представлены на рисунке 2.

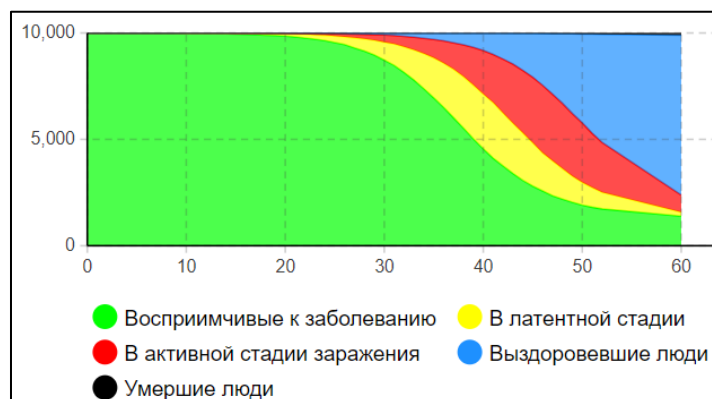


Рисунок 2 – Результат проведения имитационного эксперимента для агентного подхода

Модель показала, что за 60 дней переболело и получило иммунитет – 75 % населения, не болело – 14 %, болеет – 8 % в латентной стадии и 2 % в активной стадии.

Построенные модели позволяют предсказать динамику распространения и развития вирусного заболевания. Можно наглядно увидеть количество здоровых, зараженных, выздоровевших и умерших людей, продолжительность эпидемии, предсказывать, когда будут новые волны заболеваний.

Литература

1. Шеннон, Р. Имитационное моделирование систем – искусство и наука / Р. Шеннон. – М. : Мир, 1978. – 420 с.
2. Карпов, Ю. Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 / Ю. Г. Карпов. – Санкт-Петербург : БХВ-Петербург, 2006. – 400 с.

А. И. Лукьянов

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Е. А. Ружицкая**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА WEB-САЙТА КАФЕ “TOAST GUARD”

Архитектура сайта “Toast Guard” включает в себя две основные части: API и Django Templates. API разработано с использованием Django REST Framework (DRF), в то время как клиентская часть сайта разработана с помощью шаблонов Django. Такая архитектура позволяет обеспечить масштабируемость и гибкость приложения, позволяя легко вносить изменения и расширять функционал сайта в будущем.

Для хранения данных в приложении используется Object-Relational Mapping (ORM) и база данных SQLite. Однако, если потребуется, базу данных можно легко заменить на любую другую, такую как MySQL, PostgreSQL или Oracle, благодаря использованию ORM. Такой подход позволяет обеспечить гибкость и масштабируемость приложения в долгосрочной перспективе.

API приложения позволяет взаимодействовать с любой клиентской частью, что делает возможным расширение функционала сайта и интеграцию с другими сервисами. Такая гибкость позволяет легко внедрять новые функции и улучшения в приложение, а также интегрировать его с другими системами и сервисами, такими как мобильные приложения, платежные системы и службы доставки.

Для разработки клиентской части сайта был выбран Django framework. Это позволило создать удобный и интуитивно понятный интерфейс для пользователей, а также обеспечить надёжную и безопасную работу приложения. Кроме того, Django framework предоставляет множество встроенных функций и инструментов, которые упрощают разработку и поддержку приложения.

Благодаря выбранной архитектуре и технологиям, использованным в разработке приложения “Toast Guard”, его функционал может быть легко расширен в будущем. Новые функции и улучшения могут быть внедрены в приложение без необходимости изменять его основную структуру, а интеграция с другими сервисами и системами позволяет расширить возможности приложения и удовлетворить потребности пользователей.

А. Н. Луферов

(ГГТУ имени П. О. Сухого, Гомель)

Науч. рук. **И. Л. Стефановский**, ст. преподаватель

ИГРОВОЕ ПРИЛОЖЕНИЕ “МЕCHANIC МАУНЕМ” В ЖАНРЕ АРКАДНОГО СИМУЛЯТОРА АВТОМЕХАНИКА НА ПЛАТФОРМЕ UNITY

Введение. Unity – это кроссплатформенный движок для разработки игр, который позволяет создателям создавать 2D- и 3D-игры, а также другой интерактивный контент. Unity поддерживает разработку для самых разных систем. Создатели могут использовать Unity для создания приложений и игр для разных устройств. Это позволяет выпускать проекты на нескольких операционных системах и платформах, таких как игровые консоли, настольные компьютеры, телефоны и многое другое [1].

В игре жанра аркадного симулятора автомеханика игроки пытаются успевать обслуживать клиентов, до того, как те станут жаловаться. В ходе игры игроки могут использовать окружающие приспособления для обслуживания автомобилей. К примеру, сдувшееся колесо игрок сначала снимает, а затем относит к накачивающему устройству. Однако если он не успеет вовремя забрать колесо, то оно может лопнуть и придётся заменять. Игрокам предстоит заниматься менеджментом времени, чтобы успевать обслуживать всех клиентов.

Игровое приложение “Mechanic Mayhem”. В этой игре игрокам предстоит обслуживать одновременно сразу несколько стоек с автомобилями. Требования у них могут совпадать и различаться в мелочах, потому игрокам придётся выстраивать свой маршрут и распоряжаться временем так, чтобы успевать вовремя обслуживать клиентов.

Цель игроков – успешно закрыть смену в мастерской. Смена считается успешной, в зависимости от отзывов. Изначально шкала отзывов находится в нейтральном положении и может сдвигаться в разные стороны на основании того, как игрок обслужил клиента. Провал какого-либо требования – отрицательный отзыв, успешное выполнение – положительный. Отзывы от всех действий игрока складываются и тем самым постоянно изменяется шкала отзывов.

Заключение. В заключение отметим, что игра “Mechanic Mayhem” в жанре аркадного симулятора автомеханика, предоставляет уникальную возможность создать увлекательный опыт с необыкновенным восприятием игровых условностей. Игра этого жанра помогает развивать ценные практические навыки, такие как распоряжение временем, стратегия, реакция и решение проблем в экстренных условиях.

Литература

1. Бонд, Дж. Unity и C#. Геймдев: от идеи до реализации. / Дж. Бонд, Р. Лемарчанд. – С.-Петербург : Питер, 2022. – 928 с.