УДК 004.422.833

Студ. М.Г. Матарас
Науч. рук. ст. преп. О.А. Нистюк
(кафедра информационных систем и технологий, БГТУ)

ВИДЕОИГРОВОЙ СИМУЛЯТОР МАГАЗИНА: СИМУЛЯЦИЯ ЖИЗНИ

Создание игр имеет высокую актуальность как из коммерческой, так и из культурной и технологической перспективы. Оно предоставляет возможность сочетать развлечение, образование и инновации, и играет важную роль в современном обществе.

Цель работы: создание интерактивной среды, в которой игроки могут управлять собственным супермаркетом, принимая различные стратегические решения (управление персоналом, пополнение ассортимента и максимизация прибыли).

Рассмотрим основные классы и их зависимости: класс BotBehaviour отвечает за общие функции и поведение для всех ботов в игре. Он управляет их движением, взаимодействием с окружающим миром и воспроизводит анимации; WorkerBehaviour является абстрактным классом, который расширяет BotBehaviour. Он содержит методы для доступа к данным о работнике магазина. Эти два класса являются фундаментом для всех персонажей игры. Далее от них уже наследуются конкретные реализации: от WorkerBehaviour – JanitorBehaviour (уборщик), MerchandiserBehaviour (мерчендайзер), CashierBehaviour (кассир), от BotBehaviour – CustomerBehaviour (покупатель).

Каждый бот в проекте использует машину состояний (конечный автомат) для управления своими состояниями. Познакомимся с ним поближе. Игры заставляют нас отслеживать множество систем, которые изменяются во время выполнения. Для облегчения этого «отслеживания» и применяется данный паттерн.

Если верить оригинальной книге «Банды четырёх», машина состояний решает две проблемы:

– Объект изменяет свое поведение при изменении внутреннего состояния;

– Поведение, зависящее от состояния, определяется независимо. Добавление новых состояний не влияет на поведение существующих состояний.

Для реализации поведения ботов была реализована своя машина состояний. Основные составные части класса StateMachine: поле CurrentState (текущее состояние объекта), поле stateOwner (владелец состояния), метод UpdateState (вызывается каждый кадр для обновления текущего состояния), метод ChangeState (изменяет текущее состояние на новое состояние), метод NullifyState (сбрасывает текущее со-

стояние). В проекте инициализация состояний происходит при создании объекта бота.

ЛИТЕРАТУРА

1. Руководство по программированию на C# [Электронный ресурс] – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/csharp/programming-guide – Дата доступа: 29.03.2024.

2. Официальная документация Unity [Электронный ресурс] – Режим доступа: https://docs.unity.com – Дата доступа: 08.04.2023.

3. Официальная документация GitHub [Электронный ресурс]. Режим доступа: https://docs.github.com. Дата доступа: 11.04.2024.

УДК 004.62

Студ. А.М. Жук
Науч. рук. ассист. О.А. Нистюк
(Кафедра информационных систем и технологий, БГТУ)

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ХРАНЕНИЯ ИСПОЛЬЗУЕМЫХ В МАШИННОМ ОБУЧЕНИИ ДАННЫХ

Обеспечение безопасности данных, используемых в процессе машинного обучения, является критически важной задачей. Данные, используемые для обучения моделей, могут содержать конфиденциальную информацию, которую необходимо защищать.

На практике данные хранятся в базах данных, которые могут быть размещены на физических серверах или в облаке. Физические серверы обеспечивают полный контроль над данными и их безопасностью. Но они также требуют значительных затрат. Облачные серверы, с другой стороны, предлагают гибкость и масштабируемость, с другой – могут представлять риски для безопасности.

Особое внимание следует уделить анонимизации данных. Этот процесс позволяет скрыть идентифицирующие атрибуты. Очень важно проводить анонимизацию данных уже на этапе сбора и подготовки, сделать это можно с помощью маскирования [1], которое представляет собой процесс скрытия чувствительных атрибутов данных путем замены их на заглушки или маски.

Если не анонимные данные уже попали в руки инженера машинного обучения, важно применить процедуры псевдонимизации [2] и шифрования данных. Первая позволяет заменить идентифицирующую информацию на искусственные идентификаторы или псевдонимы. Шифрование преобразует данные в форму, которую можно прочитать только с использованием специального ключа, что обеспечивает дополнительный уровень защиты, предотвращая несанкциониро-