

к повышению энергопотребления и тепловыделения, что требует некоторых компромиссов, поиском которых постоянно занимаются учёные и инженеры. Тем не менее перспективы развития вычислительной техники в ближайшем будущем невероятно большие. Дальнейшее развитие рассмотренных тенденций, несомненно, приведет к созданию более совершенных и универсальных компьютерных систем, способных решать задачи более быстро и эффективно.

Список цитируемых источников

1. Красильщик, В. И. Вычислительные системы и сети : учебник для вузов / В. И. Красильщик. — Спб. : Питер, 2016. — 464 с.
2. Городецкий, Г. В. Компьютерные системы. Проектирование компьютеров, клоков. / Г. В. Городецкий. — М. : Физматгиз, 2020. — 384 с.

УДК 004.925

К. Н. Грушевский, Е. Г. Шапович

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

ВСТРОЕННЫЙ CHATGPT В КОМАНДНУЮ СТРОКУ WINDOWS

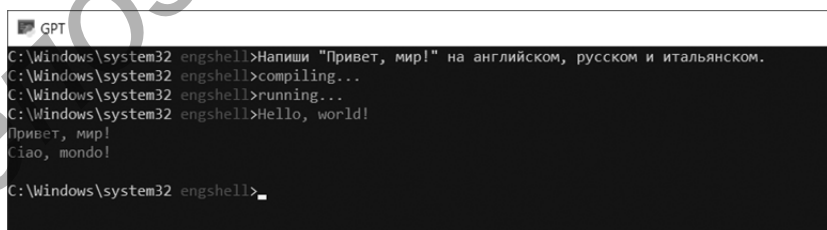
Введение. С развитием технологий и компьютеризации возникает все большая потребность в усовершенствовании средств взаимодействия с компьютером. Одним из последних достижений в этой области является создание ChatGPT — многофункциональной модели искусственного интеллекта, разработанной компанией OpenAI. Новшество заключается в том, что ChatGPT можно использовать встроенным в командную строку Windows. Это значительно упрощает процесс работы с искусственным интеллектом, поскольку нет необходимости устанавливать дополнительное программное обеспечение и настраивать его. Теперь пользователь может получить доступ к возможностям ChatGPT прямо из командной строки Windows и использовать их для решения различных задач и проблем.

Целью данного исследования является рассмотрение возможностей ChatGPT, встроенного в командную строку Windows, и выявление его необходимости в различных сферах деятельности.

Основная часть. ChatGPT 3.5-4 в командной строке может быть полезна:

1. Для пользователей, которые работают с командной строкой Windows и хотят использовать ChatGPT для выполнения различных задач.
2. Для разработчиков, которые хотят интегрировать ChatGPT в свои приложения и использовать его для различных задач.
3. Для людей, которые интересуются искусственным интеллектом и хотят узнать больше о том, как ChatGPT работает и как его можно использовать в различных сферах деятельности.

Основным способом использования ChatGPT — это сама командная строка, только с некоторыми ее изменениями. На рисунке 1 изображен интерфейс консоли Python для облегчения использования.



```
C:\Windows\system32 engshell>Напиши "Привет, мир!" на английском, русском и итальянском.
C:\Windows\system32 engshell>compiling...
C:\Windows\system32 engshell>running...
C:\Windows\system32 engshell>Hello, world!
Привет, мир!
Ciao, mondo!
C:\Windows\system32 engshell>
```

Рисунок 1 — Интерфейс ChatGPT в консоле

В модели искусственного интеллекта присутствуют особые команды [1]:

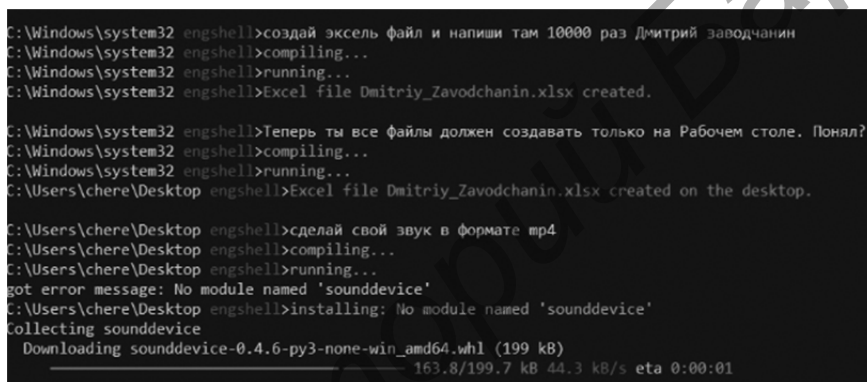
- llm — поощряет запросы LLM во время выполнения кода;
- debug — позволяет нейросети отлаживать свой собственный код в случае сбоя;
- showcode — показывает выполняемый код;
- clear — сбрасывает память вместе с консолью.

Для корректной работы необходимо иметь собственный ключ API (Application Programming Interface). Ключ API от OpenAI — это уникальный идентификатор, который выдается пользователям для доступа к API

платформы OpenAI. Ключ API позволяет пользователям отправлять запросы на серверы OpenAI и получать ответы в виде обработанных данных и результатов анализа. Это позволяет разработчикам использовать функциональность OpenAI в своих приложениях, интегрировать ее с другими системами и создавать новые продукты на основе мощных инструментов и возможностей, предоставляемых платформой. Основной код предоставлен в файле engshell.py.

В первой части кода устанавливаются переменные, которые определяют, что система находится в операционной системе Windows, устанавливается ключ API для доступа к OpenAI и устанавливается максимальный размер вводимых промтов. Максимальное значение промтов — это максимально допустимый размер текста, который можно передать в качестве входных данных для модели ChatGPT. В контексте кода, который был предоставлен, максимальное значение промтов равно 4096 символам. Это означает, что если передать в функцию LLM более 4096 символов в качестве входных данных, то будет выведено сообщение об ошибке “prompt too large”. Однако, это значение может варьироваться в зависимости от конкретной реализации ChatGPT и настроек использования модели. Так же создается отдельный .py скрипт с переопределенным ключом API от OpenAI. Затем определяются несколько функций для печати сообщений в консоль и очистки кода из ответа модели. В функции LLM определяется, как работать с ChatGPT для создания ответов на вводимые пользователем промпты. Функция LLM принимает текстовый промпт и режим (текст, код или установка) и возвращает ответ ChatGPT в соответствии с введенным промптом. Этот код может использоваться в командной строке Windows для взаимодействия с ChatGPT и получения ответов на заданные пользователем вопросы [2].

Через командную строку с ChatGPT можно, например создавать разные файлы и скачивать самостоятельно недостающие библиотеки. Пример предоставлен на рисунках 2, 3.

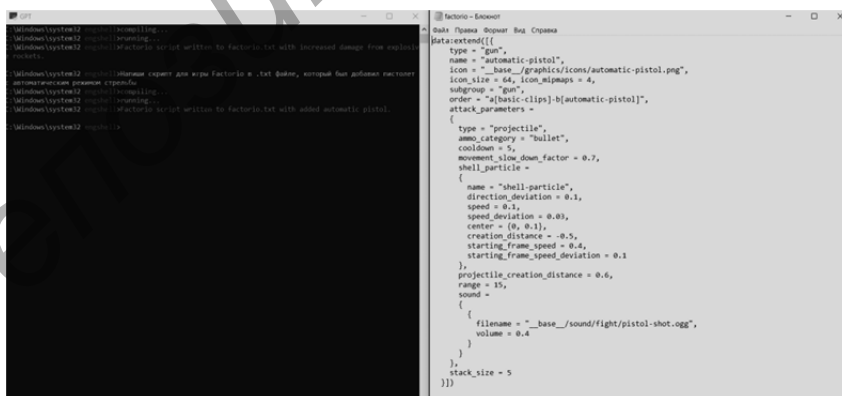


```
C:\Windows\system32 engshell>создай эксель файл и напиши там 10000 раз Дмитрий заводчанин
C:\Windows\system32 engshell>compiling...
C:\Windows\system32 engshell>running...
C:\Windows\system32 engshell>Excel file Dmitry_Zavodchanin.xlsx created.

C:\Windows\system32 engshell>Теперь ты все файлы должен создавать только на Рабочем столе. Понял?
C:\Windows\system32 engshell>compiling...
C:\Windows\system32 engshell>running...
C:\Users\chere\Desktop engshell>Excel file Dmitry_Zavodchanin.xlsx created on the desktop.

C:\Users\chere\Desktop engshell>сделай свой звук в формате mp4
C:\Users\chere\Desktop engshell>compiling...
C:\Users\chere\Desktop engshell>running...
got error message: No module named 'sounddevice'
C:\Users\chere\Desktop engshell>installing: No module named 'sounddevice'
collecting sounddevice
Downloading sounddevice-0.4.6-py3-none-win_amd64.whl (199 kB)
163.8/199.7 kB 44.3 kB/s eta 0:00:01
```

Рисунок 2 — Пример использования



```
class Gun:
    def __init__(self, name, icon, icon_size, subgroup, order):
        self.name = name
        self.icon = icon
        self.icon_size = icon_size
        self.subgroup = subgroup
        self.order = order
        self.attack_parameters = {}

class AutomaticPistol(Gun):
    def __init__(self):
        super().__init__(name="Automatic Pistol", icon="automatic-pistol.png", icon_size=64, subgroup="gun", order=1)
        self.attack_parameters = {
            "type": "projectile",
            "ammo_category": "bullet",
            "cooldown": 5,
            "movement_slow_down_factor": 0.7,
            "shell_particle": {
                "name": "shell-particle",
                "direction_deviation": 0.1,
                "speed": 0.1,
                "speed_deviation": 0.05,
                "center": (0, 0.1),
                "creation_distance": -0.5,
                "starting_frame_speed": 0.4,
                "starting_frame_speed_deviation": 0.1
            },
            "projectile_creation_distance": 0.6,
            "range": 15,
            "sound": {
                "filename": "_base_/sound/flight/pistol-shot.ogg",
                "volume": 0.4
            }
        }

def shoot(x, y):
    gun = AutomaticPistol()
    gun.shoot(x, y)

def main():
    shoot(100, 100)
```

Рисунок 3 — Пример использования

Заключение. В заключении можно отметить, что ChatGPT 3.5-4, встроенный в командную строку Windows, представляет собой удобный инструмент для обработки естественного языка, создания текстовых ответов на запросы и генерации текстовых данных. Он может быть использован в различных областях, таких как машинное обучение, генерация текстов, обработка естественного языка и многих других. Благодаря своей простоте и доступности, ChatGPT 3.5-4 может быть полезен как начинающим разработчикам, так и опытным профессионалам. К минусам стоит отнести не всегда точное понимание контекста запроса и соответственно получение некорректного ответа, который не всегда есть возможность исправить.

Список цитируемых источников

1. *Ричардсон, К.* Микросервисы. Паттерны разработки и рефакторинга / К. Ричардсон. — СПб. : Питер, 2019. — 544 с.
2. *Симан, М.* Внедрение зависимостей на платформе .NET / М. Симан, Стивен Ван Дерсен. — СПб. : Питер, 2021. — 608 с.

УДК 004.9

Р. Д. Дедулько, О. Д. Кравчук

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

РАЗРАБОТКА ИГРЫ «СОКОБАН» С ПРИМЕНЕНИЕМ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА

Введение. Игровая индустрия играет важную роль в различных аспектах жизни и общества, таких как:

1. Развлечение и отдых. Самая очевидная роль игровой индустрии — предоставление различных видов игр для развлечения и отдыха. Игры могут быть использованы для расслабления, увлечения, улучшения моторики и координации движений, получения культурных знаний и т. д.

2. Развитие технологий. Игровые разработки стоят у истоков развития многих новых технологий. Например, в процессе создания игр были разработаны уникальные алгоритмы, графические движки, системы управления и распределения ресурсов.

3. Обучение. Различные жанры игр также могут быть использованы как эффективное средство обучения. Обучающие игры предоставляют пользователям целый ряд преимуществ, таких как облегчение запоминания информации, улучшение реакции, увеличение мотивации и интереса к учебному процессу.

4. Создание новых профессий. Игровая индустрия создала множество новых профессий, таких как разработчик игр, дизайнер персонажей, гейм-мейкер, игровой архитектор, тестировщик игр и т. д.

5. Экономическое развитие. Игровая индустрия является одной из самых быстрорастущих в мире. Она создает рабочие места, привлекает инвестиции, вносит вклад в общее экономическое развитие и содействует улучшению условий жизни многих людей.

Таким образом, игровая индустрия играет важную роль в различных сферах жизни и общества и продолжает расти и развиваться, предоставляя новые возможности и выгоды, как для потребителей, так и для разработчиков и инвесторов.

Основная часть. Цель исследования: разработка игры «Сокобан» с использованием современных технологий программирования. Объект исследования: игровой процесс игры «Сокобан».

Сокобан — это головоломка, в которой надо переставлять ящики на определенные места на игровом поле. Эта игра очень популярна во всем мире, и ее играют люди всех возрастов.

Процесс игры начинается с того, что игрок появляется на игровом поле вместе с тремя или четырьмя ящиками. Игрок может передвигаться только вверх, вниз, вправо и влево, и каждый раз, когда игрок двигается, все ящики на игровом поле тоже двигаются вместе с ним.

Цель игры состоит в том, чтобы передвинуть все ящики на определенные места на игровом поле. Эти места обычно помечены различными символами или цветами, чтобы обозначить, куда нужно поместить каждый ящик. Игрок должен продвигать ящики по-игровому полю так, чтобы они не заблокировались в углах или у краев поля [1].

Каждый уровень игры имеет свой уникальный дизайн, что делает игру более интересной и вызывающей. Уровни становятся все сложнее и сложнее по мере продвижения игрока в игре, и требуют более тщательного планирования и размышления, чтобы достичь цели.

В целом, процесс игры «Сокобан» представляет собой игру рефлексов и логического мышления, требующую от игрока многократных попыток и ошибок, чтобы достичь цели и пройти все уровни игры.

Диаграмма классов является одним из главных инструментов для моделирования различных аспектов объектно-ориентированной системы. Ее назначение заключается в создании графического представления классов и их связей в системе, что позволяет лучше понимать структуру и архитектуру системы. На рисунке 1 представлена диаграмма классов разрабатываемого приложения.

В `ClassLevelMatrix` имеются значения ячеек матриц уровня игрок, ящик, стена, целевая точка, ящик на целевой точке, пустая клетка, игрок на цели. Так же имеются методы `loadfromfile` и `savetofile` наследуемые из интерфейса `InterfaceWorkingWithFile` для работы с файлами.

В интерфейсе `InterfaceMoving` описаны прототипы функций для перемещения по лабиринту `moveUp()`, `moveDown()`, `moveLeft()`, `moveRight()`.

В интерфейсе `InterfaceGameVictory` описаны прототипы функций `VictoryValidate()` для проверки условий победы и `Victory()` для обработки победы.

`ClassScoreRecord` содержит три поля результат, имя игрока при входе, пройденный уровень.

`ClassHichStoreTable` содержит поле «таблица с результатами» и методы `LoadData()` для отображения