ТЕМА РАЗДЕЛА 7.3

**Разработка прототипа интерактивного учебного курса с ИИ и геймификацией по математике для 7 класса**

**Вступление**

В ходе стажировки выполнялась учебная задач по разработки прототипа учебного курса с ИИ и геймификацией по математике. За время занятий были реализованы следующие задачи:

1. Краткий обзор предметной области (техн. программирования, LLM, способы решения подобных задач)

2. Определились с выбором учебного контента согласно ФГОС по математике. Протестировали запросы к LLM GigaChat.

3. Разработали и реализовали программные алгоритмы для решения поставленной задачи

4. Протестировали полученную программу

5. Собрали полученный код в отдельный модуль (ПРИЛОЖЕНИЕ 1)

**Ход выполнения**

1. **Выбор темы и платформы:** Учитывая мой профессиональный профиль (учитель математики) и растущий запрос учеников на цифровые образовательные ресурсы с элементами игры, была выбрана тема разработки интерактивного учебного курса. Оптимальной платформой для реализации концепции геймифицированного обучения с ИИ был определен Telegram-бот, созданный в рамках модулей 7.1 и 7.2.
2. **Разработка учебного контента:** Разработка учебного контента была выполнена, основываясь на ФГОС по предмету математика за 7 класс. В качестве дополнительных материалов были созданы различные медиа-материалы: уроки по темам, презентации, таблицы и тд. Функционал отправки перечисленных медиа-материалов также был добавлен в чат бот.
3. **Внедрение геймификации:** С целью повышения мотивации и вовлеченности учащихся, что особенно актуально для данной возрастной группы, в курс были интегрированы игровые механики. В частности, в Telegram-боте реализована интерактивная математическая викторина (квиз). Важной особенностью является алгоритм адаптивной сложности вопросов квиза, автоматически подстраивающийся под уровень знаний и прогресс ученика.
4. **Реализация адаптивного обучения:** Адаптивность системы обеспечивается двумя ключевыми функциями: динамической сложностью квиза (п.3) и модульной подачей учебного контента. Новые материалы становятся доступны учащемуся только после успешного прохождения тестирования по предыдущему разделу, обеспечивая поэтапное освоение тем.
5. **Технологическая платформа:** В качестве базовой технологической платформы для развертывания всего функционала интерактивного учебного курса использовался Telegram бот. (Приложение 1)
6. **Тестирование прототипа:** Проведение масштабного тестирования курса на целевой аудитории (учениках 7 класса) в период летних каникул оказалось затруднительным. Апробация функционала была осуществлена на группе из 10 слушателей в рамках специализированного образовательного мероприятия.

**Ключевые преимущества разработки:**

1. **Персонализация обучения:** Адаптивная сложность заданий и поэтапная выдача материалов учитывают индивидуальный темп и уровень подготовки каждого ученика.
2. **Повышение мотивации:** Элементы геймификации (квиз с прогрессом) делают процесс изучения математики более увлекательным и соревновательным для подростков.
3. **Доступность и удобство:** Использование популярного мессенджера Telegram в качестве платформы обеспечивает легкий доступ к учебным материалам в любое время, прямо с мобильных устройств учащихся.
4. **Соответствие стандартам:** Курс гарантированно покрывает необходимые темы школьной программы 7 класса благодаря разработке на основе ФГОС.
5. **Интерактивность и мультимедийность:** Разнообразные медиаресурсы (презентации, таблицы) и интерактивные элементы (квиз) способствуют лучшему усвоению материала.

Приложение 1

Telegram bot фрагмент кода, включающий геймификацию, адаптивное обучение

def math\_game(message):

    if message.text == 'Меню': # Возврат в меню по желанию пользователя

       show\_menu(message)

       return

    user\_id = str(message.from\_user.id)

    global user\_progres

    user\_progres[user\_id] = [0,0,'']

    markup\_line = types.InlineKeyboardMarkup()

    level\_math = load\_user\_data()[user\_id]['level\_math'] # берем данные уровня математики

    problem, answer = mathgen.genById(math\_levels[level\_math][random.randint(0,len(math\_levels[level\_math])-1)])

    problem = problem.replace(r'\cdot','\*').replace('$','')

    answer = answer.replace('$','')

    corr = random.randint(0,3)

    for i in range(4):

        \_, fake = mathgen.genById(math\_levels[level\_math][random.randint(0,len(math\_levels[level\_math])-1)])

        fake = fake.replace('$','')

        btn = types.InlineKeyboardButton(

            text = f'{answer if i==corr else fake}', # если i и corr равны, пишется answer, иначе fake

            callback\_data=f'math\_{i}\_{corr}\_{problem}'

        )

        markup\_line.add(btn)

    msg = bot.send\_message(user\_id, f'Решите пример {problem}', reply\_markup=markup\_line)

    user\_progres[user\_id][2] = msg.message\_id

def lesson\_selection(message):

    if message.text == 'Меню': #если пользователь нажал меню - возвращаемся

        show\_menu(message)

        return

    lesson = message.text #считываем урок, который ползователь выбрал

    lesson\_folder = lessons[lesson] #считываем путь к конкретной папке

    try:

        send\_materials(message,lesson\_folder)

        bot.send\_message(message.chat.id, f'Файлы урока {lesson} успешно отправлены')

    except BaseException:

        bot.send\_message(message.chat.id, 'Ошибка при отправке файлов')

        show\_menu(message)

def send\_materials(message, folder\_path):

    for filename in os.listdir(folder\_path):

        file\_path = os.path.join(folder\_path, filename)

        if filename.endswith('.txt'):

            with open(file\_path, 'r', encoding='utf-8') as file:

                bot.send\_message(message.from\_user.id, f"Текст урока {file.read()}")

        elif filename.endswith(('.jpg','.jpeg','.png')):

            with open(file\_path, 'rb') as photo:

                bot.send\_photo(message.from\_user.id,photo)

        elif filename.endswith(('.mp4','.mov')):

            with open(file\_path, 'rb') as video:

                bot.send\_video(message.from\_user.id,video)

        elif filename.endswith('.pdf'):

            with open(file\_path, 'rb') as pdf:

                bot.send\_document(message.from\_user.id,pdf, caption='Учебный файл')

        elif filename.endswith('.mp3'):

            with open(file\_path, 'rb') as audio:

                bot.send\_document(message.from\_user.id,audio, caption='Аудио файл')

def test\_mode(message):

    if message.text == 'Меню': #если пользователь нажал меню - возвращаемся

        show\_menu(message)

        return

    global questions, user\_progres

    user\_id = str(message.from\_user.id)

    user\_progres[user\_id] = [0,0,'']

    markup = types.ReplyKeyboardMarkup()

    markup.add('Меню')

    lesson\_name = list(lessons.keys())[int(message.text)-1]

    bot.send\_message(user\_id, f"Начинает тест по уроку {lesson\_name}",reply\_markup=markup)

    try:

        questions = open('test\_'+str(message.text)+'.txt','r',encoding='utf-8').readlines()

    except BaseException:

        bot.send\_message(user\_id,'Ошибка чтения файла')

        show\_menu(message)

        return

    send\_question(message, questions[user\_progres[user\_id][0]])

def send\_question(message, question):

    markup = types.InlineKeyboardMarkup()

    parts = question.split('\_')

    user\_id = str(message.from\_user.id)

    #Сколько будет 2+2?\_1\_2\_3\_4\_3

    #В переменной parts получим список ['Сколько..','1','2','3','4','3']

    for i, answer in enumerate(parts[1:-1]):

        btn = types.InlineKeyboardButton(

            text = answer,

            callback\_data=f'answer\_{user\_progres[user\_id][0]}\_{i}\_{parts[-1]}'

        )

        markup.add(btn)

    msg = bot.send\_message(user\_id, f'{user\_progres[user\_id][0]+1}. Вопрос {parts[0]}',reply\_markup=markup)

    user\_progres[user\_id][2] = msg.message\_id

@bot.callback\_query\_handler(func=lambda call: call.data.startswith("answer\_"))

def handle\_answer(message):

    \_,ques,answ,corr = message.data.split('\_')

    user\_id = str(message.from\_user.id)

    bot.edit\_message\_text(

        chat\_id=user\_id,

        message\_id=user\_progres[user\_id][2],

        text=f'{ques} \nНомер ответа - {int(answ)+1}',

        reply\_markup=None

    )

    user\_progres[user\_id][0]+=1

    if int(answ) == int(corr):

        user\_progres[user\_id][1]+=1

    if user\_progres[user\_id][0] != len(questions):

        bot.send\_message(user\_id, 'Следующий вопрос')

        send\_question(message, questions[user\_progres[user\_id][0]])

    else:

        bot.send\_message(user\_id, 'Тест завершён!')

        bot.send\_message(user\_id, f'Ты правильно ответил на {user\_progres[user\_id][1]}'

                         f' из {len(questions)} вопросов')

        user\_data = load\_user\_data()

        score = round(user\_progres[user\_id][1]\*100/len(questions),2)

        if score >= 60:

            user\_data[user\_id]['level']+=1

            bot.send\_message(user\_id, 'Вы прошли тест по этому модулю')

        else:

            bot.send\_message(user\_id, 'Вы не прошли тест, попробуйте снова')

        save\_user\_data(user\_data)

        show\_menu(message)

@bot.callback\_query\_handler(func=lambda call: call.data.startswith('math\_'))

# срабатывает когда пользователь отправляет ответ на мат. пример

def math\_answer(message):

    \_,answ,corr,problem = message.data.split('\_')

    user\_id = str(message.from\_user.id)

    bot.edit\_message\_text(

        chat\_id=user\_id,

        message\_id=user\_progres[user\_id][2],

        text=f'Пример {problem}, ваш ответ №{answ}',

        reply\_markup=None

    )

    user\_data = load\_user\_data()

    user\_id = str(message.from\_user.id)

    if corr==answ:

        user\_data[user\_id]['score\_math']+=1

        if user\_data[user\_id]['score\_math']>=5:

            if user\_data[user\_id]['level\_math'] <2:

                user\_data[user\_id]['level\_math']+=1

            user\_data[user\_id]['score\_math']=0

        save\_user\_data(user\_data)

        bot.send\_message(user\_id,'Правильно!')

    else:

        bot.send\_message(user\_id,'Не правильно!')

    msg = bot.send\_message(user\_id,f'Продолжить? Ваши очки - {user\_data[user\_id]['score\_math']}'

                           f', ваш уровень - {user\_data[user\_id]['level\_math']}')

    bot.register\_next\_step\_handler(msg, math\_game)

Файл config.py

main\_menu = {

    'btn1':'Расписание',

    'btn2':'ДЗ',

    'btn3':'Фото',

    'btn4':'Вопрос GigaChat',

    'btn5':'F.A.Q.',

    'btn6':'Игра в математику',

    'btn7':'Начать обучение',

    'btn8':'Начать тестирование'

}

question\_menu = {

    'btn1':'Сколько будет 2+2?',

    'btn2':'Адрес школы',

    'btn3':'Работа баскетбольной площадки',

    'btn4':'-',

    'btn5':'-'

}

lessons = {

    'Урок 1':'Уроки\Урок1',

    'Урок 2':'Уроки\Урок2',

    'Урок 3':'Уроки\Урок3',

    'Урок 4':'Уроки\Урок4',

    'Урок 5':'Уроки\Урок5'

}

Файл test\_1.txt - пример теста, при прохождении адаптивного обучения:

Какой из одночленов записан в стандартном виде?\_5a²b\_5a²b³\_5\*2a²b\_5a² + b\_0

Чему равен коэффициент одночлена -7x³y?\_7\_-7\_3\_-3\_1

Какова степень одночлена 4a²b⁵?\_2\_5\_7\_10\_2

Какие слагаемые являются подобными в выражении 3x - 5y + 2x² + 7x?\_3x и 2x²\_3x и 7x\_-5y и 7x\_2x² и 7x\_1

Упростите выражение: 8k + 5m - 3k - 2m?\_5k + 3m\_11k + 3m\_5k - 3m\_5k + 7m\_0

Упростите выражение: -2a + 4b - a - 5b?\_-3a - b\_-a - b\_-3a + 9b\_3a - b\_0

Найдите значение выражения 3c - 4d + c + 5d при c=2, d=1, предварительно упростив его?\_4c + d = 9\_4c + d = 10\_4c + d = 8\_4c + d = 7\_0



