

Методика создания HTML-игры для изучения делимости натуральных чисел с применением генеративной нейросети DeepSeek

Екимова Яна Сергеевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Статья посвящена разработке интерактивной образовательной веб-игры «Полёт знаний» для изучения темы «Делимость натуральных чисел» в курсе математики 5–6 классов с применением искусственного интеллекта DeepSeek для генерации программного кода. Подробно рассматривается поэтапное создание приложения от формулировки развернутого промта с описанием жанра, механик, визуального стиля и требований к двадцати пронумерованным воздушным шарам на фоне горного пейзажа до получения и адаптации готового HTML/CSS/JavaScript-кода, реализации игровой логики, модальных окон с математическими заданиями, системы визуальной обратной связи, отслеживания статистики ответов и финального экрана с дифференцированными итогами прохождения. В результате была разработана интерактивная образовательная игра с использованием искусственного интеллекта DeepSeek.

Ключевые слова: искусственный интеллект, математика, делимость натуральных чисел, HTML, CSS, javascript, deepseek.

A technique for creating an HTML game for studying the divisibility of natural numbers using the DeepSeek generative neural network

Ekimova Yana Sergeevna

Sholom Aleichem Priamurskiy State University

Student

Abstract

The article is devoted to the development of an interactive educational web game "Flight of Knowledge" for studying the topic "Divisibility of natural numbers" in a 5th-6th grade mathematics course using DeepSeek artificial intelligence to generate program code. The step-by-step creation of the application is considered in detail from the formulation of a detailed script describing the genre, mechanics, visual style and requirements for twenty numbered balloons against a mountain landscape to obtaining and adapting ready-made HTML/CSS/JavaScript code, implementing game logic, modal windows with mathematical tasks, visual feedback system, tracking response statistics, etc. the final screen with differentiated results of the passage. As a result, an interactive educational game using DeepSeek artificial intelligence was developed.

Keywords: artificial intelligence, mathematics, divisibility of natural numbers, HTML, CSS, javascript, deepseek.

1 Введение

1.1 Актуальность

Современный образовательный ландшафт характеризуется необходимостью трансформации традиционных методов обучения в сторону большей интерактивности и персонализации. В условиях высокой цифровой насыщенности среды обитания обучающихся стандартные формы контроля знаний часто не способны удерживать внимание и поддерживать внутреннюю мотивацию к учебе. Геймификация учебного процесса выступает эффективным ответом на этот вызов, однако создание качественных образовательных игр традиционно требовало от педагогов специфических навыков программирования и дизайна, что создавало высокий порог входа.

Появление генеративных искусственных нейросетей, таких как DeepSeek, кардинально меняет эту парадигму, демонстрируя процесс разработки цифрового контента. Возможность получения готового программного кода (HTML, CSS, JavaScript) по естественному текстовому запросу позволяет учителю выступать не только как методистом, но и как разработчиком собственных обучающих инструментов без глубоких технических компетенций. Это открывает перспективы для оперативного создания адаптивных тренажеров, таких как игра «Полёт знаний» по математике, где механика взаимодействия и визуальная обратная связь генерируются автоматически под конкретные учебные задачи.

1.2 Обзор исследований

В статье С.В. Рогатина, В.А. Антошкина рассматривается методика разработки компьютерных игр по технологии HTML5 с помощью CASE-средства GameMakerStudio компании YoYoGamesLtd. Проводится тестирование производительности работы компьютерных игр в браузере в зависимости от сложности использованных спрайтов игровых объектов [1]. Разработали игровое приложение «league of football heads» с использованием html5, css3, javascript И.А. Фёдоров, Е.Ю. Кузьменкова [2]. Я. А. Королук, Н. Ю. Куликова показали опыт разработки и использования интерактивных образовательных ресурсов на основе html5 для онлайн-обучения школьников информатике [3]. В работе рассмотрели возможности языка разметки гипертекста HTML5 для создания макета электронного учебника Д.М.Размологов, А.С. Панина, П.В. Захаров. Привели разработанный макет, а также часть таблицы стилей для дизайна макета [4]. И.С. Шегаев, А.Г. Полозов разработали новейший политологический глоссарий: основные категории, понятия, термины [5].

1.3 Цель исследования

Целью исследования является разработка и описание методического подхода к созданию интерактивной образовательной HTML-игры «Полёт

знаний» по математике для изучения темы «Делимость натуральных чисел» с использованием генеративного искусственного интеллекта DeepSeek.

2 Материалы и методы

Для разработки и тестирования интерактивной образовательной игры «Полёт знаний» в рамках данного исследования использовался генеративный искусственный интеллект DeepSeek (доступный по адресу <https://chat.deepseek.com>) — нейросетевая языковая модель, способная создавать программный код на HTML, CSS и JavaScript на основе текстовых описаний на естественном языке.

В качестве среды выполнения и тестирования использовался веб-браузер с поддержкой современных стандартов HTML5, CSS3 и ECMAScript 6, что обеспечивало корректное отображение векторной графики (воздушные шары, горный пейзаж, солнце), модальных диалоговых окон с вопросами на признаки делимости, интерактивных кнопок с вариантами ответов, системы визуальной обратной связи (подсветка правильных и неправильных ответов зелёным и красным цветами), а также анимированных элементов интерфейса.

3 Результаты

Для того чтобы создать игру, необходимо пройти процедуру регистрации в искусственном интеллекте DeepSeek <https://chat.deepseek.com>, после чего сформулировать и отправить промт — текстовый запрос на естественном языке, содержащий описание жанра, механик, целевой платформы и желаемого визуального стиля.

Создай полноценную HTML-игру в одном файле (HTML + CSS + JavaScript внутри) по следующей теме и параметрам:

ТЕМА УРОКА: [Делимость натуральных чисел]

ПРЕДМЕТ: [Математика]

КЛАСС: [5 класс]

НАЗВАНИЕ ИГРЫ: [Полёт знаний]

КОЛИЧЕСТВО ЗАДАНИЙ: [20]

ЗАДАНИЯ (каждое задание должно иметь: вопрос + 4 варианта ответа + номер правильного ответа (0-3)):

1. Вопрос: "..."
Варианты: ["...", "...", "...", "..."]
Правильный: X
2. Вопрос: "..."
- ...

[продолжай до нужного количества]

📌 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ:

- Всё в ОДНОМ HTML-файле (CSS и JavaScript встроены)
- Графика — ТОЛЬКО SVG (без внешних изображений, без библиотек)
- Адаптивность: viewBox с preserveAspectRatio="xMidYMid slice"
- Работает автономно, без интернета (кроме шрифтов)

🌀 ТРЕБОВАНИЯ К ГРАФИКЕ (SVG):

1. ФОН — детализированная сцена заката:
 - Градиентное небо: от тёмно-фиолетового (#1a0533) через розовый (#e8451e) к золотому (#ffd700)
 - Анимированные звёзды в верхней части (мигают с разной частотой)
 - Солнце с radialGradient, лучами и свечением (filter: strongGlow)
 - Горы в 3 слоя (дальние — тёмные, средние — серые, ближние — зелёные) с белыми снежными шапками и силуэтами ёлок

- Море внизу с анимированными волнами (path с animate)
- Отражение солнца в воде (ellipse с анимацией)
- Облака с плавным движением (animateTransform)
- Чайки (3 штуки) с анимацией парения
- Земля/трава с отдельными травинками
- Светящиеся частицы (маленькие кружки, плавающие вверх)
- 2. ВОЗДУШНЫЕ ШАРЫ (количество = количество заданий):
 - Каждый шар: эллипс с radialGradient (разные цвета для каждого), белый блик, горизонтальная полоска, номер задания в центре
 - Корзина: прямоугольник с текстурой дерева (rect с обводкой и линиями)
 - Верёвки: 3 линии от шара к корзине
 - Каждый шар анимирован (покачивание — animateTransform translate с рандомными значениями и разной длительностью 3-6 секунд)
 - Шары расположены слева в 3 ряда (по 5 штук)
- 3. ЭЛЕМЕНТЫ UI:
 - Название игры: крупный текст сверху по центру (font-size: 42px) с glow-фильтром
 - Счётчик: ✓ правильных | ✗ неправильных (прямоугольник с закруглёнными углами)
 - Прогресс-бар: показывает сколько заданий пройдено

🎮 МЕХАНИКА ИГРЫ:

1. При клике на шар → открывается модальное окно с заданием:
 - Красивая карточка с градиентом (linear-gradient 135deg)
 - Номер задания бейджем
 - Текст вопроса крупным шрифтом
 - 4 кнопки-варианта в сетке 2×2
 - Анимация появления (scale + opacity)
 - Кнопки с hover-эффектами (подъём + тень)
2. При ПРАВИЛЬНОМ ответе:
 - Кнопка подсвечивается зелёным + анимация пульсации
 - Показывается надпись “✓ Верно!” с анимацией появления/исчезновения
 - Шар плавно летит к горам/закату (transform: translate + scale(0.6) + opacity: 0.7)
 - Правильные ответы +1
3. При НЕПРАВИЛЬНОМ ответе:
 - Выбранная кнопка красная + анимация тряски
 - Правильный ответ подсвечивается зелёным
 - Показывается надпись “✗ Неверно!” с анимацией
 - Шар падает на землю (translateY вниз + rotate(15deg) + scale(0.8) + цвет серый)
 - Неправильные ответы +1
4. После всех заданий:
 - Появляется финальное окно с результатом
 - Мотивирующее сообщение в зависимости от количества правильных ответов:
 - 15/15: “🏆 Идеально! Ты настоящий гений!”
 - 12-14: “🌟 Отличный результат!”
 - 8-11: “👍 Хороший результат! Но есть над чем поработать.”
 - <8: “📖 Стоит повторить материал и попробовать снова!”
 - Кнопка “🔄 Играть снова” (перезапуск без перезагрузки страницы)

🎨 СТИЛИ:

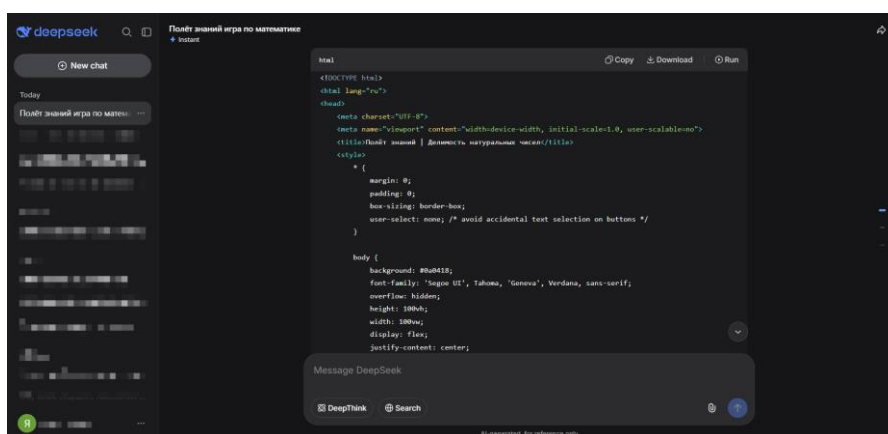
- Модальное окно: backdrop-filter: blur(5px), полупрозрачный фон
- Карточка задания: box-shadow, скруглённые углы (25px)
- Анимации: popIn, correctPulse, wrongShake, resultFade, fadeInUp
- Шрифт: ‘Segoe UI’, Tahoma, sans-serif
- Цветовая палитра заката: #1a0533, #2d1b69, #6b2fa0, #e8451e, #ff8c42, #ffd700
- Цвета шаров: розовый (#ff6b9d→#c0392b), бирюзовый (#4ecdc4→#1a936f), голубой (#a8e6cf→#3da5d9), жёлтый (#ffd93d→#ff8c00), фиолетовый (#c9b1ff→#8e44ad), синий (#87ceeb→#4682b4)

⚠ ВАЖНО:

- Все SVG-градиенты, фильтры и анимации должны быть прописаны внутри <defs>
- JavaScript-массив задач должен содержать ровно столько заданий, сколько указано
- Каждый элемент должен быть кликабельным только если ещё не отвечен
- Не использовать внешние библиотеки (jQuery, Bootstrap и т.д.)
- Код должен быть чистым, с комментариями

- The screenshot shows the DeepSeek web interface. On the left is a sidebar with a 'New chat' button and a list of recent chats. The main area displays a conversation. The user's message is a math problem in Russian about the divisibility of natural numbers. The AI's response provides a detailed solution, including the theorem, conditions, and a step-by-step calculation. The interface is clean and modern, with a dark theme.

Копируем программный код HTML/CSS структуры игрового интерфейса в текстовый редактор для создания базового каркаса веб-приложения с заданными стилями оформления, включая позиционирование элементов, цветовую схему и анимационные эффекты.



Переходим в блокнот и создаем файл с названием «index.html», затем вставляем туда код, который представляет собой полностью автономный веб-документ с интегрированной SVG-сценой заката, анимированными воздушными шарами, расположенными с чередованием направления, модальными окнами заданий по теме «Делимость натуральных чисел» для 5 класса, системой подсчёта правильных и неправильных ответов, прогресс-баром и финальным экраном с мотивирующим сообщением; после сохранения файла и открытия в любом современном браузере игра запускается без доступа к интернету, позволяя ученикам в интерактивной форме отвечать на 20 вопросов.

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
<title>Полёт знаний — Делимость чисел (5 класс)</title>
<style>
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    user-select: none;
  }

  body {
    min-height: 100vh;
    background: linear-gradient(145deg, #0a0515 0%, #1f0e32 100%);
    display: flex;
    justify-content: center;
    align-items: center;
    font-family: 'Segoe UI', 'Tahoma', 'Geneva', sans-serif;
    overflow: hidden;
  }

  .game-container {
    position: relative;
    width: 100vw;
    height: 100vh;
    overflow: hidden;
  }

  .game-svg {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    display: block;
    z-index: 1;
    pointer-events: auto; /* Шары должны получать события */
  }

  .ui-overlay {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    z-index: 20;
    pointer-events: none;
  }

  .game-header {
    position: absolute;
    top: 20px;
    left: 0;
    width: 100%;
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 0 30px;
    flex-wrap: wrap;
    gap: 15px;
    z-index: 25;
    pointer-events: none;
  }
```

```
.title-glow {
  font-size: 42px;
  font-weight: 800;
  background: linear-gradient(135deg, #FFE6B0, #FFB347, #FF7E05);
  -webkit-background-clip: text;
  background-clip: text;
  color: transparent;
  text-shadow: 0 0 15px rgba(255, 200, 0, 0.8);
  letter-spacing: 2px;
  filter: drop-shadow(0 0 6px #ffae42);
  white-space: nowrap;
}

.score-panel {
  pointer-events: auto;
  display: flex;
  gap: 20px;
  background: rgba(0, 0, 0, 0.55);
  backdrop-filter: blur(8px);
  border-radius: 60px;
  padding: 8px 24px;
  border: 1px solid rgba(255, 215, 0, 0.4);
}

.score-item {
  font-size: 24px;
  font-weight: bold;
  color: white;
  display: flex;
  align-items: center;
  gap: 8px;
}

.score-correct {
  color: #a3ffa3;
  text-shadow: 0 0 3px green;
}

.score-wrong {
  color: #ffaeae;
}

.progress-container {
  width: 280px;
  background: rgba(0, 0, 0, 0.6);
  border-radius: 30px;
  height: 12px;
  backdrop-filter: blur(4px);
  border: 1px solid #ffcd7e;
}

.progress-fill {
  width: 0%;
  height: 100%;
  background: linear-gradient(90deg, #FFD966, #FFA500);
  border-radius: 30px;
  transition: width 0.4s ease;
}

@keyframes floatBalloon {
  0% { transform: translate(0, 0) rotate(0deg); }
  25% { transform: translate(5px, -4px) rotate(2deg); }
  75% { transform: translate(-5px, 3px) rotate(-2deg); }
```

```
    100% { transform: translate(0, 0) rotate(0deg); }
  }

.balloon-item {
  cursor: pointer;
  transition: filter 0.2s;
  animation: floatBalloon 3.5s ease-in-out infinite;
}

.balloon-item:hover {
  filter: brightness(1.1);
  cursor: pointer;
}

.modal {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.8);
  backdrop-filter: blur(8px);
  display: flex;
  justify-content: center;
  align-items: center;
  z-index: 200;
  visibility: hidden;
  opacity: 0;
  transition: visibility 0.2s, opacity 0.2s;
  pointer-events: none;
}

.modal.active {
  visibility: visible;
  opacity: 1;
  pointer-events: auto;
}

.quiz-card {
  background: linear-gradient(135deg, #2a1b44, #18122b);
  border-radius: 48px;
  max-width: 600px;
  width: 90%;
  padding: 30px 28px;
  box-shadow: 0 30px 45px rgba(0,0,0,0.6), 0 0 0 2px rgba(255,180,70,0.3);
  transform: scale(0.7);
  opacity: 0;
  transition: transform 0.3s cubic-bezier(0.34, 1.2, 0.64, 1), opacity 0.25s;
  text-align: center;
  color: #f8f3e0;
}

.modal.active .quiz-card {
  transform: scale(1);
  opacity: 1;
}

.question-badge {
  background: #ff8c42;
  display: inline-block;
  padding: 6px 20px;
  border-radius: 40px;
  font-weight: bold;
```

```
font-size: 1rem;
margin-bottom: 18px;
}

.question-text {
font-size: 1.7rem;
font-weight: 600;
margin: 20px 0 30px;
line-height: 1.3;
}

.answers-grid {
display: grid;
grid-template-columns: 1fr 1fr;
gap: 16px;
margin: 20px 0 25px;
}

.answer-btn {
background: rgba(30, 20, 50, 0.9);
border: 2px solid #ffb347;
border-radius: 60px;
padding: 14px 8px;
font-size: 1.1rem;
font-weight: bold;
color: #fff2db;
cursor: pointer;
transition: all 0.2s;
box-shadow: 0 4px 0 #7a3a0e;
}

.answer-btn:hover {
transform: translateY(-3px);
background: #ffb347;
color: #1f0e32;
box-shadow: 0 8px 14px rgba(0,0,0,0.3);
}

.feedback-msg {
font-size: 1.5rem;
font-weight: bold;
margin-top: 10px;
min-height: 60px;
}

.correct-highlight {
background: #2e8b57 !important;
border-color: #c7ffb1 !important;
animation: pulseGreen 0.5s ease;
}

.wrong-highlight {
background: #b22234 !important;
animation: shake 0.4s ease-in-out 0s 2;
}

@keyframes pulseGreen {
0% { transform: scale(1); background: #2e8b57; }
50% { transform: scale(1.03); box-shadow: 0 0 15px #6eff9e; }
100% { transform: scale(1); }
}

@keyframes shake {
```

```

    0% { transform: translateX(0); }
    25% { transform: translateX(-5px); }
    75% { transform: translateX(5px); }
    100% { transform: translateX(0); }
  }

.final-modal {
  background: linear-gradient(135deg, #332a55, #100b1f);
  text-align: center;
}

.restart-btn {
  background: #ffaa33;
  border: none;
  padding: 14px 30px;
  font-size: 1.3rem;
  border-radius: 60px;
  font-weight: bold;
  margin-top: 25px;
  cursor: pointer;
  transition: 0.2s;
  color: #2d1b2e;
}

.restart-btn:hover {
  transform: scale(1.03);
  background: #ffcc66;
}

.toast-feedback {
  position: fixed;
  bottom: 30%;
  left: 50%;
  transform: translateX(-50%);
  background: #000000cc;
  backdrop-filter: blur(16px);
  padding: 12px 28px;
  border-radius: 80px;
  font-size: 1.6rem;
  font-weight: bold;
  z-index: 300;
  pointer-events: none;
  animation: floatMessage 1s forwards;
  white-space: nowrap;
}

@keyframes floatMessage {
  0% { opacity: 1; transform: translateX(-50%) translateY(0); }
  100% { opacity: 0; transform: translateX(-50%) translateY(-50px); }
}
</style>
</head>
<body>
<div class="game-container">
  <svg class="game-svg" viewBox="0 0 1400 800" preserveAspectRatio="xMidYMid slice">
    <defs>
      <linearGradient id="skyGrad" x1="0" y1="0" x2="0" y2="1">
        <stop offset="0%" stop-color="#1a0533" />
        <stop offset="35%" stop-color="#4c1e5e" />
        <stop offset="65%" stop-color="#e8451e" />
        <stop offset="100%" stop-color="#ffd700" />
      </linearGradient>
      <radialGradient id="sunGrad" cx="50%" cy="50%" r="50%">

```

```

        <stop offset="0%" stop-color="#fff7b0" />
        <stop offset="40%" stop-color="#ffd966" />
        <stop offset="80%" stop-color="#ff8c42" />
        <stop offset="100%" stop-color="#e8451e" />
    </radialGradient>
    <filter id="glowSun" x="-50%" y="-50%" width="200%" height="200%">
        <feGaussianBlur stdDeviation="12" result="blur" />
        <feMerge>
            <feMergeNode in="blur" />
            <feMergeNode in="SourceGraphic" />
        </feMerge>
    </filter>
    <radialGradient id="ballPink" cx="35%" cy="30%">
        <stop offset="0%" stop-color="#ff9fcf" />
        <stop offset="70%" stop-color="#ff4d6d" />
        <stop offset="100%" stop-color="#c9184a" />
    </radialGradient>
    <radialGradient id="ballTeal" cx="35%" cy="30%">
        <stop offset="0%" stop-color="#8cf7f0" />
        <stop offset="70%" stop-color="#20bfb4" />
        <stop offset="100%" stop-color="#0b6e6a" />
    </radialGradient>
    <radialGradient id="ballGold" cx="35%" cy="30%">
        <stop offset="0%" stop-color="#fff2a8" />
        <stop offset="70%" stop-color="#f4b942" />
        <stop offset="100%" stop-color="#cc7b00" />
    </radialGradient>
    <radialGradient id="ballPurple" cx="35%" cy="30%">
        <stop offset="0%" stop-color="#e0b3ff" />
        <stop offset="70%" stop-color="#aa5eff" />
        <stop offset="100%" stop-color="#6b2ee5" />
    </radialGradient>
    <radialGradient id="ballBlue" cx="35%" cy="30%">
        <stop offset="0%" stop-color="#a2e3ff" />
        <stop offset="70%" stop-color="#3f8ecc" />
        <stop offset="100%" stop-color="#1e5791" />
    </radialGradient>
    <radialGradient id="ballOrange" cx="35%" cy="30%">
        <stop offset="0%" stop-color="#ffce9e" />
        <stop offset="70%" stop-color="#ff8c42" />
        <stop offset="100%" stop-color="#bf4d0a" />
    </radialGradient>
</defs>

<rect width="1400" height="800" fill="url(#skyGrad)" />

<!-- Звёзды -->
<g>
    <circle cx="120" cy="60" r="2" fill="white"><animate attributeName="opacity" values="1;0.2;1"
dur="2.1s" repeatCount="indefinite" /></circle>
    <circle cx="340" cy="95" r="2.5" fill="white"><animate attributeName="opacity"
values="0.5;1;0.5" dur="3.2s" repeatCount="indefinite" /></circle>
    <circle cx="680" cy="40" r="1.8" fill="#fff9c4"><animate attributeName="opacity"
values="0.3;1;0.3" dur="1.7s" repeatCount="indefinite" /></circle>
    <circle cx="900" cy="110" r="2.2" fill="white"><animate attributeName="opacity"
values="1;0.2;1" dur="2.9s" repeatCount="indefinite" /></circle>
    <circle cx="1150" cy="55" r="2" fill="#ffe6b3"><animate attributeName="opacity"
values="0.6;1;0.6" dur="2.4s" repeatCount="indefinite" /></circle>
    <circle cx="500" cy="150" r="1.5" fill="white"><animate attributeName="opacity"
values="0.8;0.2;0.8" dur="3.5s" repeatCount="indefinite" /></circle>
    <circle cx="1270" cy="180" r="2" fill="white"><animate attributeName="opacity" values="1;0.3;1"
dur="2s" repeatCount="indefinite" /></circle>

```

```

<circle cx="80" cy="200" r="1.5" fill="#fff0b3"><animate attributeName="opacity"
values="0.6;1;0.6" dur="2.8s" repeatCount="indefinite" /></circle>
<circle cx="750" cy="70" r="1.8" fill="white"><animate attributeName="opacity"
values="0.4;0.9;0.4" dur="3.6s" repeatCount="indefinite" /></circle>
</g>

<!-- Солнце -->
<g transform="translate(1100, 280)">
<circle cx="0" cy="0" r="65" fill="url(#sunGrad)" filter="url(#glowSun)" />
<g opacity="0.35">
<animateTransform attributeName="transform" type="rotate" from="0" to="360" dur="25s"
repeatCount="indefinite" />
<line x1="0" y1="-90" x2="0" y2="-135" stroke="#ffd966" stroke-width="5" stroke-
linecap="round" />
<line x1="0" y1="90" x2="0" y2="135" stroke="#ffd966" stroke-width="5" stroke-
linecap="round" />
<line x1="-90" y1="0" x2="-135" y2="0" stroke="#ffd966" stroke-width="5" stroke-
linecap="round" />
<line x1="90" y1="0" x2="135" y2="0" stroke="#ffd966" stroke-width="5" stroke-
linecap="round" />
</g>
</g>

<!-- Горы -->
<path d="M0,620 L120,450 L260,540 L380,420 L520,560 L640,480 L790,520 L950,400 L1100,530
L1250,470 L1400,550 L1400,800 L0,800 Z" fill="#2a2a3e" />
<path d="M120,450 L155,480 L130,490 L108,480 Z" fill="white" opacity="0.5" />
<path d="M380,420 L415,460 L390,465 L365,455 Z" fill="white" opacity="0.5" />
<path d="M950,400 L985,445 L960,455 L930,440 Z" fill="white" opacity="0.6" />

<path d="M0,660 L180,530 L320,600 L470,490 L610,620 L780,550 L930,510 L1080,590 L1220,520
L1400,590 L1400,800 L0,800 Z" fill="#4a5a5e" />

<path d="M0,730 L250,640 L400,680 L590,590 L720,670 L880,620 L1010,680 L1150,610 L1300,660
L1400,640 L1400,800 L0,800 Z" fill="#4f7343" />

<!-- Ёлки -->
<g fill="#1a3a1a">
<polygon points="100,620 108,580 116,620" />
<polygon points="103,600 108,565 113,600" />
<polygon points="300,650 308,615 316,650" />
<polygon points="303,635 308,600 313,635" />
<polygon points="1300,620 1308,585 1316,620" />
<polygon points="1303,605 1308,570 1313,605" />
<polygon points="80,680 87,650 94,680" />
<polygon points="1320,660 1327,625 1334,660" />
</g>

<!-- Море -->
<path d="M0,750 Q200,730 400,750 Q600,770 800,745 Q1000,725 1200,750 Q1350,765 1400,750
L1400,800 L0,800 Z" fill="#1b4d6e" />
<path d="M0,765 Q180,745 370,765 Q550,785 740,760 Q930,740 1120,765 Q1280,780 1400,760
L1400,800 L0,800 Z" fill="#123b55">
<animate attributeName="d" values="M0,765 Q180,745 370,765 Q550,785 740,760 Q930,740
1120,765 Q1280,780 1400,760 L1400,800 L0,800 Z;M0,770 Q190,755 380,770 Q570,750 750,765 Q940,785
1130,770 Q1290,755 1400,770 L1400,800 L0,800 Z;M0,765 Q180,745 370,765 Q550,785 740,760 Q930,740
1120,765 Q1280,780 1400,760 L1400,800 L0,800 Z" dur="3s" repeatCount="indefinite" />
</path>

<!-- Отражение солнца -->
<ellipse cx="1100" cy="745" rx="90" ry="14" fill="#ffdd99" opacity="0.45">
<animate attributeName="rx" values="70;95;70" dur="2.5s" repeatCount="indefinite" />
<animate attributeName="opacity" values="0.4;0.7;0.4" dur="2.5s" repeatCount="indefinite" />

```

```

</ellipse>

<!-- Облака -->
<g>
  <animateTransform attributeName="transform" type="translate" values="0,0;60,0;0,0" dur="25s"
repeatCount="indefinite" />
  <path d="M280,170 Q300,145 330,155 Q360,140 380,160 Q400,170 380,185 L290,185 Q260,185
270,175 Q265,160 280,170Z" fill="#fffaec" opacity="0.45" />
  <path d="M920,200 Q940,180 960,188 Q980,175 995,190 Q1010,200 995,210 L930,210 Q910,210
918,202 Q915,192 920,200Z" fill="#fff5e0" opacity="0.4" />
</g>

<!-- Чайки -->
<g>
  <animateTransform attributeName="transform" type="translate" values="0,0;30,-18;0,0" dur="5.5s"
repeatCount="indefinite" />
  <path d="M480,200 L495,190 L510,200 L505,195 L495,198 L485,195 Z" fill="#2c2c2c" />
</g>
<g>
  <animateTransform attributeName="transform" type="translate" values="0,0;-25,-12;0,0"
dur="6.5s" repeatCount="indefinite" />
  <path d="M680,180 L695,170 L710,180 L705,175 L695,178 L685,175 Z" fill="#2c2c2c" />
</g>
<g>
  <animateTransform attributeName="transform" type="translate" values="0,0;20,-10;0,0" dur="4.8s"
repeatCount="indefinite" />
  <path d="M250,250 L260,243 L270,250 L267,247 L260,249 L253,247 Z" fill="#2c2c2c" />
</g>

<!-- Частицы -->
<circle cx="200" cy="700" r="3" fill="#ffd966">
  <animate attributeName="cy" values="700;640;700" dur="5s" repeatCount="indefinite" />
  <animate attributeName="opacity" values="0.8;0.2;0.8" dur="5s" repeatCount="indefinite" />
</circle>
<circle cx="450" cy="680" r="2.5" fill="#ffb347">
  <animate attributeName="cy" values="680;620;680" dur="4.3s" repeatCount="indefinite" />
  <animate attributeName="opacity" values="0.7;0.1;0.7" dur="4.3s" repeatCount="indefinite" />
</circle>
<circle cx="800" cy="710" r="3" fill="#ffaa66">
  <animate attributeName="cy" values="710;650;710" dur="6s" repeatCount="indefinite" />
  <animate attributeName="opacity" values="0.9;0.2;0.9" dur="6s" repeatCount="indefinite" />
</circle>
<circle cx="1200" cy="690" r="2.5" fill="#ffcc88">
  <animate attributeName="cy" values="690;630;690" dur="3.8s" repeatCount="indefinite" />
  <animate attributeName="opacity" values="0.6;0.1;0.6" dur="3.8s" repeatCount="indefinite" />
</circle>

<g id="balloonsGroup"></g>
</svg>

<div class="ui-overlay">
  <div class="game-header">
    <div class="title-glow">✧ ПОЛЁТ ЗНАНИЙ ✧</div>
    <div class="score-panel">
      <div class="score-item">✓ <span id="correctCount" class="score-correct">0</span></div>
      <div class="score-item">✗ <span id="wrongCount" class="score-wrong">0</span></div>
      <div class="progress-container"><div class="progress-fill" id="progressFill"></div></div>
    </div>
  </div>
</div>

<div id="quizModal" class="modal">

```

```

<div class="quiz-card">
  <div class="question-badge" id="questionNumber">Задание 1</div>
  <div class="question-text" id="questionText"></div>
  <div class="answers-grid" id="answersContainer"></div>
  <div class="feedback-msg" id="feedbackMsg"></div>
</div>
</div>

<div id="finalModal" class="modal">
  <div class="quiz-card final-modal">
    <h2 style="font-size: 2rem;">🎈 Итоги полёта 🎈</h2>
    <p style="font-size: 1.8rem; margin: 20px 0" id="finalResultText"></p>
    <p id="finalMessage" style="font-size: 1.3rem;"></p>
    <button class="restart-btn" id="restartGameBtn">🎮 Играть снова</button>
  </div>
</div>

<script>
  const QUESTIONS = [
    { text: "Какое число делится на 2?", options: ["17", "24", "35", "41"], correct: 1 },
    { text: "Какое из чисел кратно 5?", options: ["42", "55", "63", "78"], correct: 1 },
    { text: "Выберите число, которое делится на 3:", options: ["20", "31", "45", "52"], correct: 2 },
    { text: "Какое число является делителем 18?", options: ["4", "6", "7", "8"], correct: 1 },
    { text: "Найдите НЕЧЁТНОЕ число, кратное 5:", options: ["15", "20", "30", "40"], correct: 0 },
    { text: "Из чисел 28, 33, 42, 55 выбери то, что делится на 6:", options: ["28", "33", "42", "55"],
correct: 2 },
    { text: "Число 72 кратно:", options: ["5", "6", "7", "11"], correct: 1 },
    { text: "Какой цифрой оканчивается число, делящееся на 10?", options: ["0", "2", "5", "8"],
correct: 0 },
    { text: "Какое число делится на 9 без остатка?", options: ["81", "73", "56", "44"], correct: 0 },
    { text: "Выберите число, кратное и 2, и 3:", options: ["14", "21", "30", "35"], correct: 2 },
    { text: "Наибольший общий делитель чисел 12 и 18:", options: ["3", "6", "9", "12"], correct: 1 },
    { text: "Наименьшее общее кратное чисел 4 и 5:", options: ["10", "15", "20", "25"], correct: 2 },
    { text: "Какое из чисел не делится на 4?", options: ["24", "36", "42", "48"], correct: 2 },
    { text: "Число 125 делится на:", options: ["2", "3", "5", "10"], correct: 2 },
    { text: "Выберите число, которое является делителем 56:", options: ["5", "7", "9", "11"], correct:
1 },
    { text: "Число 99 кратно:", options: ["5", "7", "9", "11"], correct: 2 },
    { text: "Какое из чисел кратно 8?", options: ["32", "44", "54", "62"], correct: 0 },
    { text: "Если число оканчивается на 0, оно делится на:", options: ["2, 5 и 10", "только на 2",
"только на 5", "только на 4"], correct: 0 },
    { text: "НОК(6,8) равно:", options: ["12", "16", "24", "48"], correct: 2 },
    { text: "Число 111 делится на 3? (признак делимости)", options: ["Да, 1+1+1=3", "Нет", "Только
на 9", "Только на 5"], correct: 0 }
  ];

  let gameState = {
    answered: new Array(20).fill(false),
    correctAnswers: 0,
    wrongAnswers: 0,
    currentModalActive: false
  };

  let balloonElements = [];

  function updateUI() {
    document.getElementById("correctCount").innerText = gameState.correctAnswers;
    document.getElementById("wrongCount").innerText = gameState.wrongAnswers;
    let answeredTotal = gameState.answered.filter(v => v === true).length;
    let percent = (answeredTotal / 20) * 100;
    document.getElementById("progressFill").style.width = percent + "%";
  }

```

```

        if (answeredTotal === 20) {
            !document.getElementById("finalModal").classList.contains("active") {
                showFinalModal();
            }
        }

        function showToast(msg) {
            let div = document.createElement("div");
            div.className = "toast-feedback";
            div.innerText = msg;
            document.body.appendChild(div);
            setTimeout(() => div.remove(), 1200);
        }

        function animateBalloon(index, isCorrect) {
            let balloon = balloonElements[index];
            if (!balloon) return;

            if (isCorrect) {
                balloon.style.transition = "transform 1.2s cubic-bezier(0.4,0,0.2,1), opacity 1s";
                balloon.style.transform = "translate(200px, -300px) scale(0.3)";
                balloon.style.opacity = "0";
            } else {
                balloon.style.transition = "transform 0.9s ease-in, opacity 0.7s";
                balloon.style.transform = "translateY(280px) rotate(15deg) scale(0.5)";
                balloon.style.opacity = "0.4";
                let body = balloon.querySelector(".balloon-body");
                if (body) body.setAttribute("fill", "#8b7355");
            }
            setTimeout(() => {
                if (balloon) balloon.style.visibility = "hidden";
            }, 1100);
        }

        function handleAnswer(selectedIdx, correctIdx, qIndex) {
            if (gameState.answered[qIndex]) return;

            let isCorrect = (selectedIdx === correctIdx);
            const modal = document.getElementById("quizModal");
            const feedback = document.getElementById("feedbackMsg");
            const btns = document.querySelectorAll(".answer-btn");

            if (isCorrect) {
                gameState.correctAnswers++;
                feedback.innerHTML = "✓ ВЕРНО! Молодец! ✓";
                feedback.style.color = "#a3ffa3";
                btns[selectedIdx].classList.add("correct-highlight");
                showToast("✓ Правильно! +1");
            } else {
                gameState.wrongAnswers++;
                let correctText = QUESTIONS[qIndex].options[correctIdx];
                feedback.innerHTML = `✗ НЕВЕРНО! Правильный ответ: ${correctText}`;
                feedback.style.color = "#ffb4a2";
                btns[selectedIdx].classList.add("wrong-highlight");
                btns[correctIdx].classList.add("correct-highlight");
                showToast(`✗ Неверно! Ответ: ${correctText}`);
            }

            gameState.answered[qIndex] = true;
            updateUI();
            animateBalloon(qIndex, isCorrect);
        }
    
```

```

    setTimeout(() => {
        modal.classList.remove("active");
        gameState.currentModalActive = false;
        feedback.innerHTML = "";
        btns.forEach(btn => btn.classList.remove("correct-highlight", "wrong-highlight"));
    }, 1500);
}

function openQuestion(qIndex) {
    if (gameState.answered[qIndex]) {
        showToast("🔴 Этот шар уже улетел!");
        return;
    }
    if (gameState.currentModalActive) return;

    const q = QUESTIONS[qIndex];
    document.getElementById("questionNumber").innerText = `Задание ${qIndex + 1} из 20`;
    document.getElementById("questionText").innerText = q.text;

    const container = document.getElementById("answersContainer");
    container.innerHTML = "";

    q.options.forEach((opt, idx) => {
        const btn = document.createElement("button");
        btn.className = "answer-btn";
        btn.innerText = opt;
        btn.addEventListener("click", () => handleAnswer(idx, q.correct, qIndex));
        container.appendChild(btn);
    });

    document.getElementById("feedbackMsg").innerHTML = "";
    document.getElementById("quizModal").classList.add("active");
    gameState.currentModalActive = true;
}

function showFinalModal() {
    let correct = gameState.correctAnswers;
    let msg = "";
    if (correct === 20) msg = "🏆 ИДЕАЛЬНО! Ты настоящий гений математики! 🏆";
    else if (correct >= 15) msg = "🌟 ОТЛИЧНО! Ты отлично знаешь делимость чисел! 🌟";
    else if (correct >= 10) msg = "👍 ХОРОШО! Повтори признаки делимости, будет ещё лучше!";
    else msg = "🔁 Стоит повторить тему и попробовать снова! У тебя получится! 🔁";

    document.getElementById("finalResultText").innerHTML = `✓ Правильных: ${correct} из 20`;
    document.getElementById("finalMessage").innerHTML = msg;
    document.getElementById("finalModal").classList.add("active");
}

function restartGame() {
    gameState.answered.fill(false);
    gameState.correctAnswers = 0;
    gameState.wrongAnswers = 0;
    gameState.currentModalActive = false;

    document.getElementById("quizModal").classList.remove("active");
    document.getElementById("finalModal").classList.remove("active");

    updateUI();

    document.getElementById("balloonsGroup").innerHTML = "";

```

```

        createBalloons();
    }

    function createBalloons() {
        const container = document.getElementById("balloonsGroup");
        container.innerHTML = "";
        balloonElements = [];

        const colors = ["url(#ballPink)", "url(#ballTeal)", "url(#ballGold)", "url(#ballPurple)",
            "url(#ballBlue)", "url(#ballOrange)"];
        const positions = [
            {x: 50, y: 370}, {x: 155, y: 370}, {x: 260, y: 370}, {x: 365, y: 370}, {x: 470, y: 370}, {x: 575, y:
370}, {x: 680, y: 370},
            {x: 80, y: 480}, {x: 185, y: 480}, {x: 290, y: 480}, {x: 395, y: 480}, {x: 500, y: 480}, {x: 605, y:
480}, {x: 710, y: 480},
            {x: 110, y: 590}, {x: 215, y: 590}, {x: 320, y: 590}, {x: 425, y: 590}, {x: 530, y: 590}, {x: 635, y:
590}
        ];

        for (let i = 0; i < 20; i++) {
            const pos = positions[i];
            const color = colors[i % colors.length];

            const group = document.createElementNS("http://www.w3.org/2000/svg", "g");
            group.setAttribute("class", "balloon-item");
            group.style.animation = `floatBalloon ${3 + Math.random() * 2.5}s ease-in-out infinite`;

            const ellipse = document.createElementNS("http://www.w3.org/2000/svg", "ellipse");
            ellipse.setAttribute("cx", pos.x + 18);
            ellipse.setAttribute("cy", pos.y + 20);
            ellipse.setAttribute("rx", "28");
            ellipse.setAttribute("ry", "34");
            ellipse.setAttribute("fill", color);
            ellipse.classList.add("balloon-body");

            const highlight = document.createElementNS("http://www.w3.org/2000/svg", "ellipse");
            highlight.setAttribute("cx", pos.x + 10);
            highlight.setAttribute("cy", pos.y + 10);
            highlight.setAttribute("rx", "7");
            highlight.setAttribute("ry", "9");
            highlight.setAttribute("fill", "white");
            highlight.setAttribute("opacity", "0.45");

            const band = document.createElementNS("http://www.w3.org/2000/svg", "path");
            band.setAttribute("d", `M${pos.x + 2},${pos.y + 32} Q${pos.x + 18},${pos.y + 39} ${pos.x +
34},${pos.y + 32}`);
            band.setAttribute("stroke", "#8B5E3C");
            band.setAttribute("stroke-width", "2.5");
            band.setAttribute("fill", "none");

            const text = document.createElementNS("http://www.w3.org/2000/svg", "text");
            text.setAttribute("x", pos.x + 18);
            text.setAttribute("y", pos.y + 26);
            text.setAttribute("text-anchor", "middle");
            text.setAttribute("fill", "white");
            text.setAttribute("font-weight", "bold");
            text.setAttribute("font-size", "15");
            text.setAttribute("font-family", "sans-serif");
            text.textContent = i + 1;

            const basket = document.createElementNS("http://www.w3.org/2000/svg", "rect");
            basket.setAttribute("x", pos.x + 8);
            basket.setAttribute("y", pos.y + 54);

```

```
        basket.setAttribute("width", "20");
        basket.setAttribute("height", "16");
        basket.setAttribute("fill", "#a56b2f");
        basket.setAttribute("stroke", "#5c3a1a");
        basket.setAttribute("stroke-width", "1.2");
        basket.setAttribute("rx", "2");

        const line1 = document.createElementNS("http://www.w3.org/2000/svg", "line");
        line1.setAttribute("x1", pos.x + 8);
        line1.setAttribute("y1", pos.y + 54);
        line1.setAttribute("x2", pos.x + 12);
        line1.setAttribute("y2", pos.y + 42);
        line1.setAttribute("stroke", "#8B5E3C");
        line1.setAttribute("stroke-width", "1.8");

        const line2 = document.createElementNS("http://www.w3.org/2000/svg", "line");
        line2.setAttribute("x1", pos.x + 28);
        line2.setAttribute("y1", pos.y + 54);
        line2.setAttribute("x2", pos.x + 24);
        line2.setAttribute("y2", pos.y + 42);
        line2.setAttribute("stroke", "#8B5E3C");
        line2.setAttribute("stroke-width", "1.8");

        const line3 = document.createElementNS("http://www.w3.org/2000/svg", "line");
        line3.setAttribute("x1", pos.x + 18);
        line3.setAttribute("y1", pos.y + 54);
        line3.setAttribute("x2", pos.x + 18);
        line3.setAttribute("y2", pos.y + 43);
        line3.setAttribute("stroke", "#8B5E3C");
        line3.setAttribute("stroke-width", "1.5");

        group.appendChild(ellipse);
        group.appendChild(highlight);
        group.appendChild(band);
        group.appendChild(text);
        group.appendChild(basket);
        group.appendChild(line1);
        group.appendChild(line2);
        group.appendChild(line3);

        group.addEventListener("click", (e) => {
            e.stopPropagation();
            if (!gameState.answered[i]) openQuestion(i);
            else showToast("💣 Этот шар уже улетел!");
        });

        container.appendChild(group);
        balloonElements.push(group);
    }
}

createBalloons();
updateUI();

document.getElementById("restartGameBtn").addEventListener("click", () => {
    document.getElementById("finalModal").classList.remove("active");
    restartGame();
});

document.getElementById("quizModal").addEventListener("click", (e) => {
    if (e.target === document.getElementById("quizModal")) {
        document.getElementById("quizModal").classList.remove("active");
    }
});
```

```

        gameState.currentModalActive = false;
    }
});
</script>
</div>
</body>
</html>

```

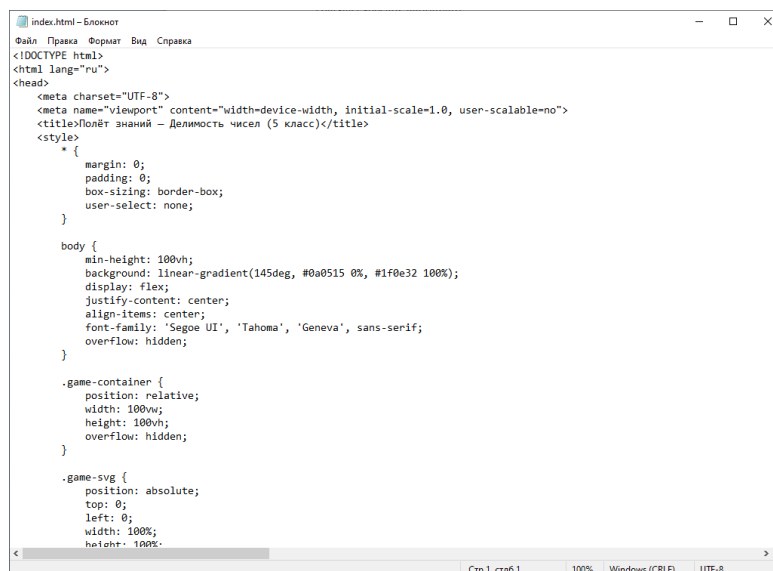


Рисунок 3. Сохранение файла

Открываем созданный файл index.html в веб-браузере для запуска и тестирования игрового приложения «Полёт знаний». Визуальный интерфейс игры включает двадцать пронумерованных воздушных шаров различных цветов, расположенных в три ряда на фоне горного пейзажа и закатного неба. В верхней части экрана отображается название игры и панель статистики, фиксирующая количество правильных и неправильных ответов пользователя.

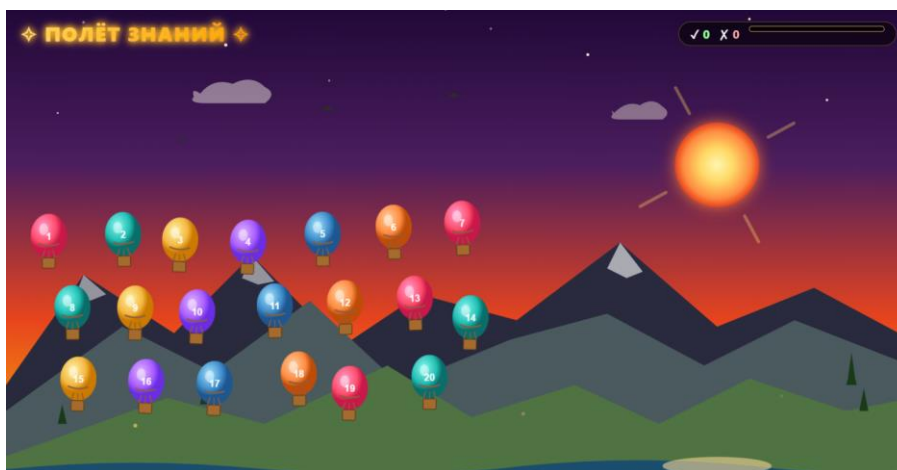


Рисунок 4. Главный экран

Далее нажимаем на любой воздушный шар, после чего на экране появляется модальное окно с математическим заданием, например, «Какое число делится на 2?», и четырьмя вариантами ответа, расположенными в виде кнопок. В верхней части окна отображается индикатор прогресса «Задание 1

из 20», фиксирующий текущий номер вопроса. Фон при этом затемняется для акцентирования внимания пользователя на активном диалоговом окне. Подобная механика обеспечивает интерактивное взаимодействие с игроком и поэтапную проверку знаний по математике в игровой форме.

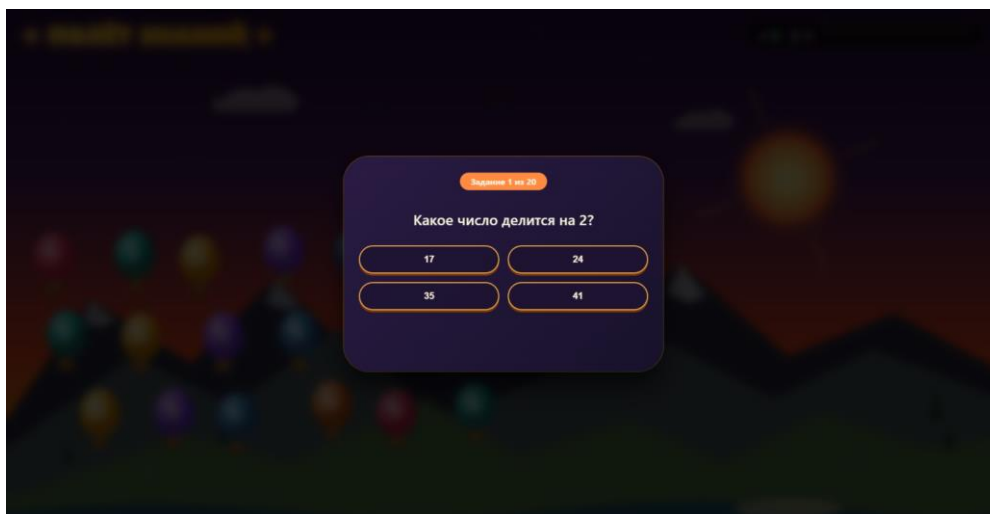


Рисунок 5. Задание 1

При правильном варианте ответа выбранная кнопка подсвечивается зелёным цветом, визуализируя корректность выбора пользователя. Под вариантами ответов появляется сообщение «✓ ВЕРНО! Молодец! ✓», сопровождаемое графическими элементами в виде галочек, что обеспечивает положительное подкрепление и мотивирует обучающегося к дальнейшему прохождению игры. После подтверждения правильного ответа система автоматически переходит к следующему заданию, обновляя индикатор прогресса и увеличивая счётчик верных ответов в панели статистики. Такая система обратной связи соответствует принципам геймификации образовательного процесса и способствует повышению вовлечённости учащихся.

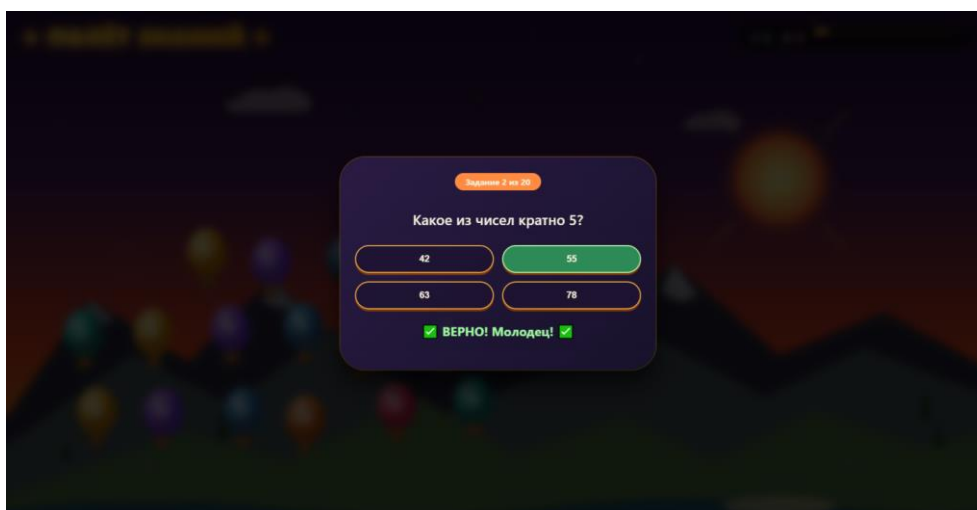


Рисунок 6. Правильный вариант ответа

Если выбран неверный вариант ответа, система подсвечивает ошибочный выбор красным цветом, одновременно отображая правильный ответ зелёным цветом. В нижней части модального окна появляется сообщение «X НЕВЕРНО! Правильный ответ: 30», информирующее пользователя о допущенной ошибке и демонстрирующее корректное решение. Такая система визуальной обратной связи способствует осознанному усвоению учебного материала, поскольку обучающийся не только фиксирует факт ошибки, но и получает возможность сопоставить свой выбор с правильным ответом. Данный механизм соответствует принципам формирующего оценивания в образовательных играх и поддерживает развитие метапредметных компетенций учащихся.

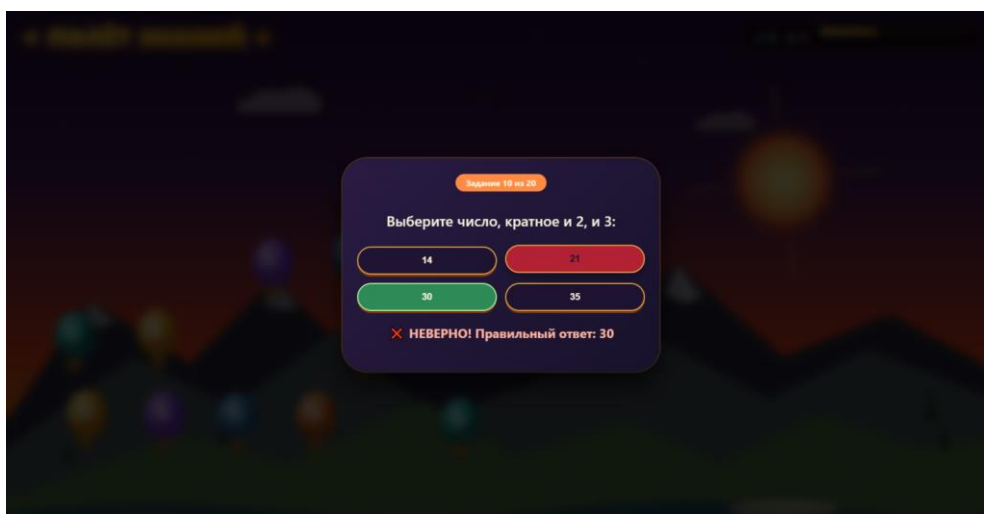


Рисунок 7. Ошибка в варианте ответа

Когда пользователь отвечает на задание, соответствующий воздушный шар исчезает с игрового поля, что демонстрирует визуальный прогресс прохождения игры. В панели статистики в правом верхнем углу одновременно обновляются счётчики правильных и неправильных ответов, отображая текущие результаты обучающегося в формате «✓ 4 X 8». Подобная динамическая система обратной связи обеспечивает наглядное представление успеваемости учащегося и формирует мотивацию к завершению всех заданий. Исчезновение отработанных элементов игрового поля снижает визуальную нагрузку и помогает пользователю сосредоточиться на оставшихся вопросах, что соответствует принципам юзабилити образовательных приложений.

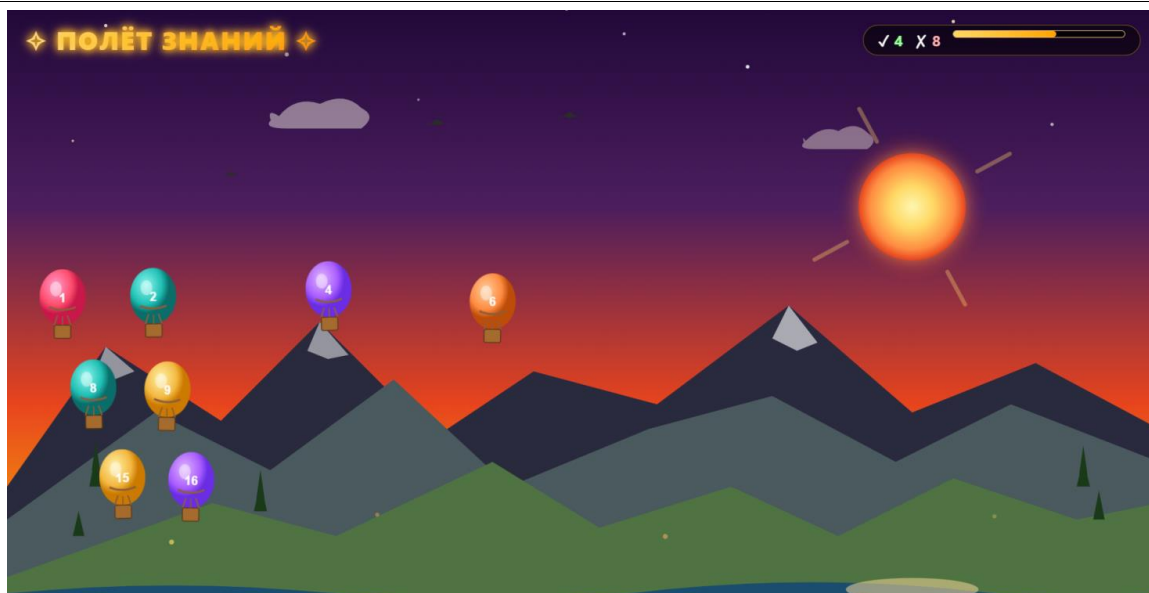


Рисунок 8. Игровое поле

При разных результатах прохождения игры система генерирует итоговое окно «Итоги полёта», содержащее количественные показатели успеваемости (количество правильных и неправильных ответов) и качественную оценку достижений обучающегося. В зависимости от процента верных ответов система демонстрирует дифференцированные сообщения: при низких результатах — рекомендацию повторить учебный материал («ХОРОШО! Повтори признаки делимости, будет ещё лучше!»), при высоких — похвалу и подтверждение успешного усвоения темы («ОТЛИЧНО! Большинство шаров в полёте к солнцу! Ты хорошо знаешь тему!»). Такая система адаптивной обратной связи соответствует принципам личностно-ориентированного обучения и обеспечивает индивидуализацию образовательной траектории учащегося на основе его актуального уровня знаний.

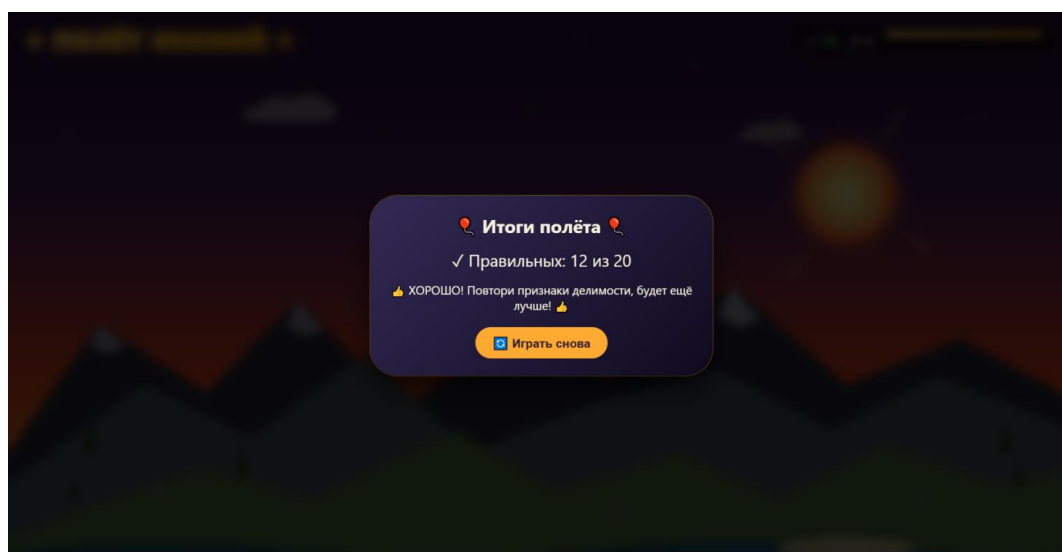


Рисунок 9. Итог 1

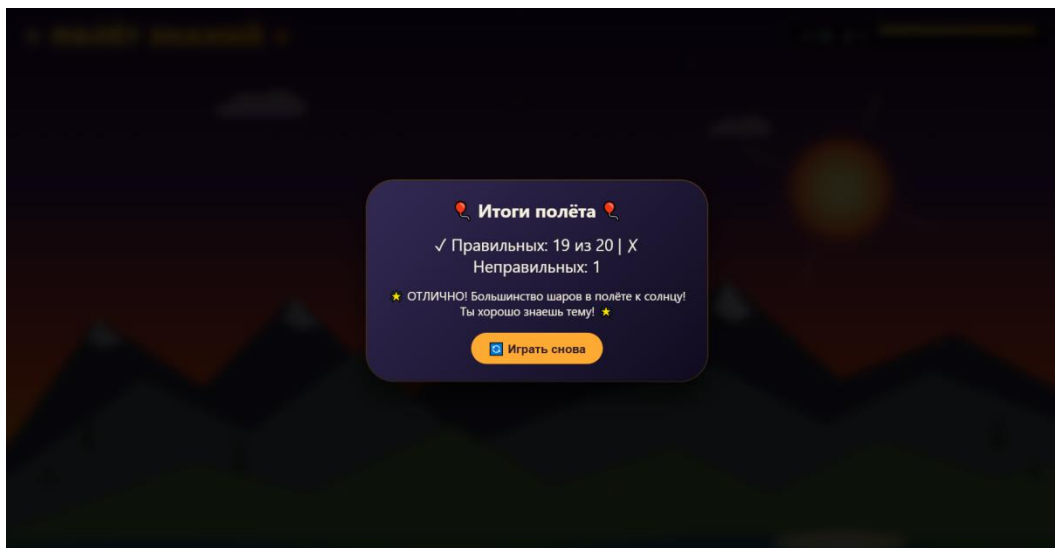


Рисунок 10. Итог 2

4 Выводы

В данной работе детально рассмотрена методика создания интерактивной образовательной игры с использованием генеративного искусственного интеллекта DeepSeek. Методика охватывает полный цикл разработки, от регистрации в нейросети и составления структурированного промта до получения, полностью функционирующего HTML/CSS/JavaScript-кода, адаптации игровой логики и публикации готового продукта.

Использование искусственного интеллекта DeepSeek делает процесс создания профессионального игрового образовательного контента доступным для пользователей без глубоких навыков программирования, в первую очередь для преподавателей, методистов и организаторов учебных мероприятий. Генеративные возможности нейросети (создание семантической HTML-разметки, адаптивного CSS-дизайна с градиентным фоном и горным пейзажем, интерактивной JavaScript-логики, 20 уникальных пронумерованных воздушных шаров, системы модальных окон с вопросами и механизмов визуальной обратной связи) позволяют не только значительно сократить время на подготовку интерактивных материалов, но и обеспечивают вовлекающий игровой формат, напрямую влияющий на мотивацию и вовлеченность учащихся.

Практическая реализация методики на примере игры «Полёт знаний» по теме «Делимость натуральных чисел» для 5–6 классов подтвердила её эффективность и тиражируемость для различных учебных дисциплин и возрастных категорий. Игровая механика, включающая последовательное решение 20 математических заданий, систему подсчёта правильных и неправильных ответов, а также дифференцированную систему поощрения в зависимости от результатов, демонстрирует высокий потенциал для формирования устойчивых вычислительных навыков. Таким образом, генеративные нейросетевые модели становятся важным инструментом образовательной геймификации, способствующим развитию цифровых педагогических компетенций и отвечающим на вызовы современного

образования, связанные с цифровизацией, необходимостью удержания внимания и адаптацией контента под интерактивные форматы. Освоение и внедрение подобных решений представляет собой закономерный шаг в эволюции методов обучения в условиях повсеместного использования цифровых технологий.

Библиографический список

1. Рогатин С. В., Антошкин В. А. Разработка компьютерных игр по технологии html5 //Информатика и прикладная математика. 2015. №. 21. С. 12-16.
2. Фёдоров И. А., Кузьменкова Е. Ю. Разработка игрового приложения «league of football heads» с использованием html5, css3, javascript //Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях. 2018. С. 311-311.
3. Королук Я. А., Куликова Н. Ю. Опыт разработки и использования интерактивных образовательных ресурсов на основе html5 для онлайнобучения школьников информатике //XXV Всероссийская студенческая научно-практическая конференция Нижневартковского государственного университета. 2023. С. 48-55.
4. Размологов Д. М., Панина А. С., Захаров П. В. Использование html5 для разработки электронного учебника //Современные тенденции развития науки и технологий. 2016. №. 4-4. С. 92-94.
5. Шегаев И. С., Полозов А. Г. Новейший политологический глоссарий: основные категории, понятия, термины. Общество с ограниченной ответственностью Издательство Перо, 2015. 242 с.