

Разработка образовательной игры для изучения управления проектами на основе диаграмм Ганта

Анишкова Анастасия Сергеевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Цель работы изменить подход к изучению проектного управления, превратив сложные профессиональные дисциплины в увлекательный игровой процесс и разработав продукт, который позволит пользователям не только освоить теорию, но и сразу применять её на практике. Проект реализован с использованием фреймворка CakePHP 2.x, СУБД MySQL, HTML/CSS/JavaScript для интерфейса. Может использоваться как для обучения основам управления проектами, так и в развлекательных целях.

Ключевые слова: диаграмм Ганта, CakePHP, игра, управления проектами.

Developing an educational game for learning project management using Gantt charts

Anishkova Anastasia Sergeevna

Sholom-Aleichem Priamurskiy State University

Student

Abstract

The goal of the work is to change the approach to learning project management by turning complex professional disciplines into a fun gameplay and developing a product that will allow users not only to learn the theory, but also to apply it immediately in practice. The project is implemented using the CakePHP 2.x framework, the MySQL DBMS, and HTML/CSS/JavaScript for the interface. It can be used both for teaching the basics of project management and for entertainment purposes.

Keywords: Gantt charts, CakePHP, game, project management.

1 Введение

1.1 Актуальность

Актуальность проекта обусловлена возрастающей потребностью в эффективных методах обучения управлению проектами. Использование игровых технологий и диаграмм Ганта позволяет наглядно и доступно познакомить обучающихся с ключевыми аспектами планирования, распределения ресурсов и контроля сроков. Разработанная игра может применяться как в образовательных целях для студентов и начинающих

специалистов, так и в тренингах для сотрудников компаний, делая процесс обучения интерактивным и увлекательным.

1.2 Обзор исследований

А.А. Штанюк показал использование программы управления проектами taskjuggler в учебном процессе [1]. П.Свентицкий и Н.А. Иванова представили инструменты кроссплатформенной разработки мобильных приложений [2]. Управление проектами в сфере нанотехнологий: особенности и возможности их учета рассмотрели В.П.Мешалкин и др. [3]. Концепция управления проектами в САПР была показана Н.Е.Новаковой и др. [4]. С.А. Олейникова исследовала модификацию метода pert для решения задач сетевого планирования и управления [5]. Использование информационных технологий в управлении проектами представили Г.Н.Чусавитина и В.Н.Макашова [6]. С.В. Широкова описала управление проектами внедрения информационных систем для предприятия [7].

1.3 Цель исследования

Цель работы изменить подход к изучению проектного управления, превратив сложные профессиональные дисциплины в увлекательный игровой процесс и разработав продукт, который позволит пользователям не только освоить теорию, но и сразу применять её на практике.

2 Материалы и методы

Проект реализован с использованием фреймворка CakePHP 2.x, СУБД MySQL, HTML/CSS/JavaScript для интерфейса.

3 Результаты

Проект разработан на основе фреймворка CakePHP версии 2.x с использованием СУБД MySQL. Архитектура приложения построена по модели MVC: вся бизнес-логика сосредоточена в контроллерах, работа с данными осуществляется через модели, а внешнее представление формируется шаблонами. При этом реализация строго следует соглашениям фреймворка, включая автоматическую маршрутизацию и принятые стандарты именования таблиц базы данных.

Для обеспечения безопасности используется встроенный компонент SecurityComponent, защищающий от CSRF-атак. Аутентификация пользователей организована через AuthComponent с применением надежного хеширования паролей по алгоритму Blowfish. Все взаимодействие с базой данных проходит через ORM фреймворка, что предотвращает возможность SQL-инъекций.

Структура проекта включает традиционные для CakePHP модули: Controlllers, Models и Views. Также задействованы встроенные механизмы управления сессиями, cookies и отправкой электронной почты через компонент CakeEmail.

В структуре базы данных (рис1.) центральное место занимает таблица User, которая содержит основную информацию о пользователях — их учетные данные (email, имя, зашифрованный пароль), игровую статистику (количество матчей, побед, поражений, очков), а также данные для подтверждения аккаунта через верификационный код. Для отслеживания времени создания и последнего изменения записей используются автоматически обновляемые временные метки. Дополнительно реализованы таблицы Rewards и Users_Rewards, представляющие собой систему достижений: первая хранит описание всех возможных наград, а вторая — информацию о тех, которые были получены конкретными пользователями. Таблица Roles предназначена для определения ролей пользователей и ограничения доступа, обеспечивая разделение прав между обычными игроками и администраторами.

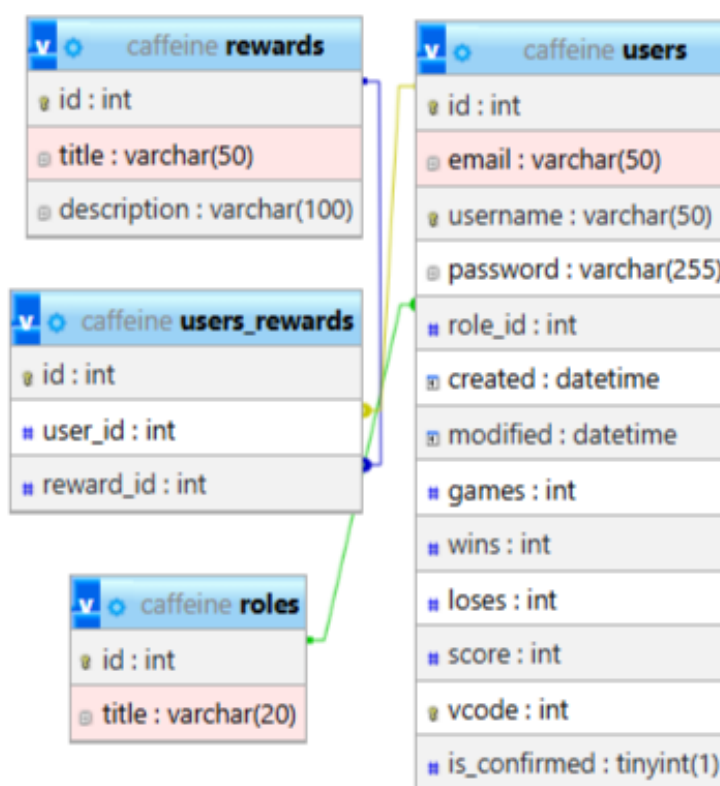


Рисунок 1. Схема базы данных

При запуске игры отображается главная страница (рис.2), содержащая удобное меню навигации, краткой информацией о проекте и двумя интересными цитатами.

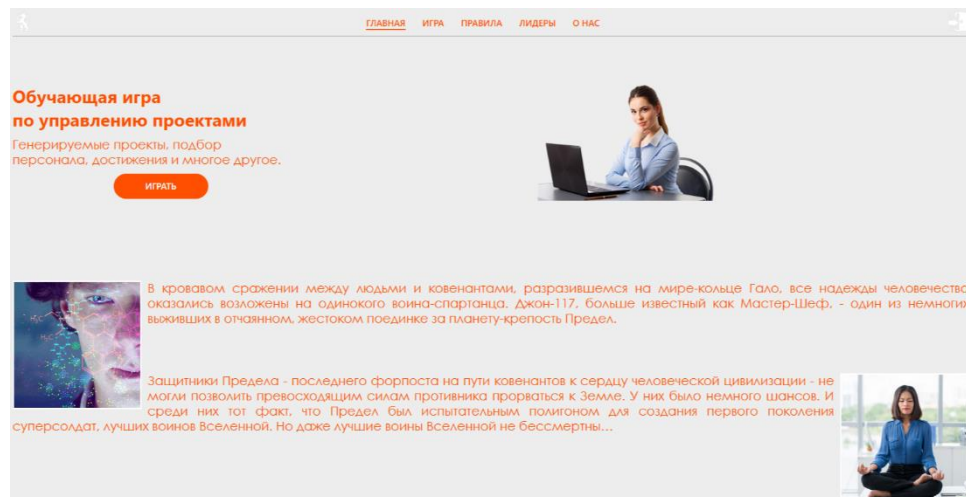


Рисунок 2. Главная страница

Код главной страницы выглядит следующим образом (рис.3):

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>
  Cherry Cake </title>
  <link href="/favicon.ico" type="image/x-icon" rel="icon"/><link href="/favicon.ico" type="image/x-icon" rel="shortcut icon"/><link rel="stylesheet"
</body>
  <div class="wrapper">
    <div class="header">
      <div class="navigation">
        <div class="navigation-col1">
          
        </div>
        <div class="navigation-col2">
          <a id="home" href="/home">Главная</a>
          <a id="game" href="/game">Игра</a>
          <a id="rules" href="/rules">Правила</a>
          <a id="leaders" href="/leaders">Лидеры</a>
          <a id="about" href="/about">О нас</a>
        </div>
        <div class="navigation-col3">
          <a href="/auth/login"></a>
        </div>
      </div>
      <div class="content">
        <div class="home-master">
          <div class="home-leftcol">
            <div class="home-heading">Обучающая игра<br>по управлению проектами</div>
            <div class="home-text">Генерируемые проекты, подбор<br>персонала, достижения и многое другое.</div>
            <div class="home-buttons">
              <a href="/game"><div class="home-button-play">Играть</div></a>
            </div>
          </div>
          <div class="home-rightcol">
            
          </div>
        </div>
        <div class="home-description">
          <p><a name="info"></a>
          В кровавом сражении между людьми и ковенантами, разразившемся на мире-кольце Гало, все надежды человечества оказались возложены на одинокого воина-1
          <p style="margin-top: 150px;">
          Защитники Предела - последнего форпоста на пути ковенантов к сердцу человеческой цивилизации - не могли позволить превосходящим силам противника пр
        </div>
        <div class="footer">
          <!--
          <div class="footer-row">
            <div class="footer-col1">© 2017 PM Game</div>
            <div class="footer-col2">Свяжитесь с нами: company@gmail.com</div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Рисунок 3. Код главной страницы

Для доступа к игровому процессу пользователю необходимо пройти процедуру авторизации в системе, либо, в случае отсутствия учетной записи, зарегистрироваться, создав новый аккаунт (рис.4,5).

Вход в систему

Имя пользователя

Пароль

Войти

[Забыли пароль?](#) [Зарегистрироваться](#)

Рисунок 4. Вход в систему

Регистрация

Имя пользователя

Электронная почта

Пароль

Готово

Рисунок 5. Регистрация

Код входа в систему выглядит таким образом (рис.6).

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>
    CakePHP: the rapid development php framework:
    Users
  </title>
  <link href="/favicon.ico" type="image/x-icon" rel="icon"/>
  <link href="/favicon.ico" type="image/x-icon" rel="shortcut icon"/>
  <link rel="stylesheet" type="text/css" href="/css/auth.css"/>
</head>
<body>
  <div id="container">
    <div id="content">
      <div class="auth-form">
        <div id="authMessage" class="message">You are not authorized to access that location.</div>
        <form action="/auth/login" id="UserLoginForm" method="post" accept-charset="utf-8">
          <div style="display:none;">
            <input type="hidden" name="_method" value="POST"/>
          </div>
          <fieldset>
            <legend>
              Вход в систему
            </legend>
            <div class="input text required">
              <label for="UserUsername">Имя пользователя</label>
              <input name="data[User][username]" maxlength="50" type="text" id="UserUsername" required="required"/>
            </div>
            <a href="/auth/restore">Забыли пароль?</a>
            <a href="/auth/register">Зарегистрироваться</a>
          </div>
        </form>
      </div>
    </div>
  </div>
</body>
</html>
```

Рисунок 6. Код окна авторизации

Код для регистрации (рис.7).

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>
    CakePHP: the rapid development php framework:
    Users
  </title>
  <link href="/favicon.ico" type="image/x-icon" rel="icon"/>
  <link href="/favicon.ico" type="image/x-icon" rel="shortcut icon"/>
  <link rel="stylesheet" type="text/css" href="/css/auth.css"/>
</head>
<body>
  <div id="container">
    <div id="content">
      <div class="auth-form">
        <form action="/auth/register" id="UserRegisterForm" method="post" accept-charset="utf-8">
          <div style="display:none;">
            <input type="hidden" name="_method" value="POST"/>
          </div>
          <fieldset>
            <legend>Регистрация</legend>
            <div class="input text required">
              <label for="UserUsername">Имя пользователя</label>
              <input name="data[User][username]" maxlength="50" type="text" id="UserUsername" required="required"/>
            </div>
            <div class="submit">
              <input type="submit" value="Готово"/>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</body>
</html>
```

Рисунок 7. Код окна регистрации

После успешного входа в систему игрок попадает на главный экран, где сразу видит свою игровую статистику (количество сыгранных матчей, побед и поражений) и характеристики доступных сотрудников, которых предстоит распределять по задачам в ходе игрового процесса (рис.8)

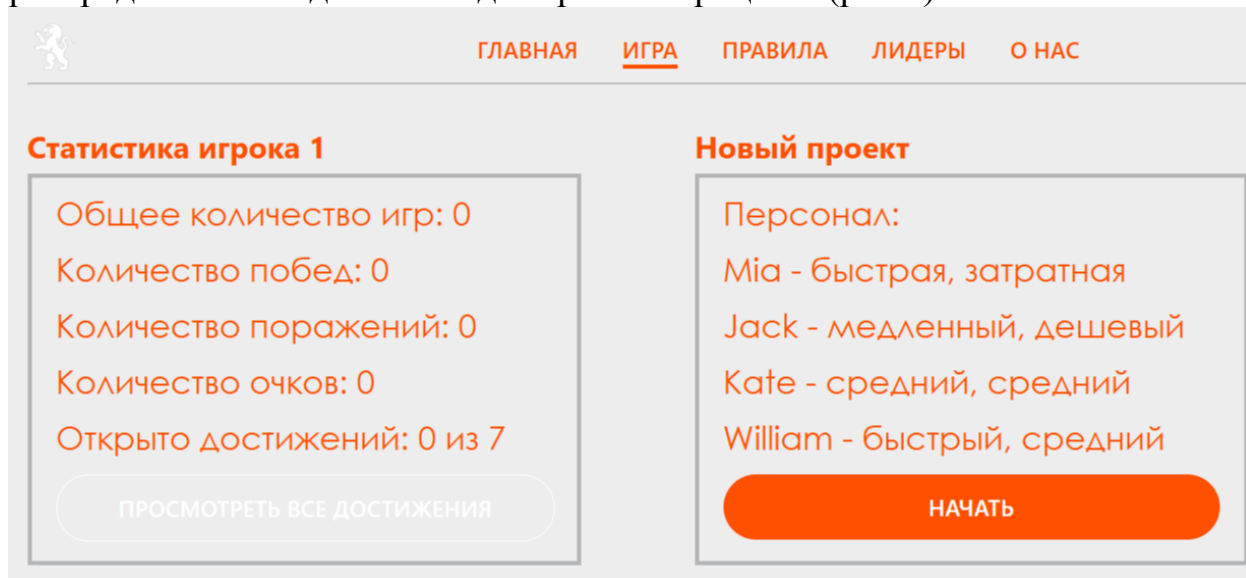


Рисунок 8. Игра

Код страницы выглядит таким образом (рис.9).

```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>
    Cherry Cake </title>
  <link href="/favicon.ico" type="image/x-icon" rel="icon"/><link href="/favicon.ico" type="image/x-icon" rel="shortcu
</body>
  <div class="wrapper">
    <div class="header">
      <div class="navigation">
        <div class="navigation-col1">
          
        </div>
        <div class="navigation-col2">
          <a id="home" href="/home">Главная</a>
          <a id="game" href="/game">Игра</a>
          <a id="rules" href="/rules">Правила</a>
          <a id="leaders" href="/leaders">Лидеры</a>
          <a id="about" href="/about">О нас</a>
        </div>
        <div class="navigation-col3">
          <a href="/auth/logout"></a>
        </div>
      </div>
      <hr>
    </div>

    <div class="content">
      <div class="game-left-col">
        <h2>Статистика игрока 1</h2>
        <div class="game-block-statistics">
          <p>Общее количество игр: 0</p>
          <p>Количество побед: 0</p>
          <p>Количество поражений: 0</p>
          <p>Количество очков: 0</p>
          <p>Открыто достижений: 0 из 7</p>
          <a href="/game/rewards"><div class="game-button-rewards">Просмотреть все достижения</div></a>
        </div>
      </div>
      <div class="game-mid-col"></div>
      <div class="game-right-col">
        <h2>Новый проект</h2>
        <div class="game-block-play">
          <p>Персонал:</p>
          <p>Mia - быстрая, затратная</p>
          <p>Jack - медленный, дешевый</p>
          <p>Kate - средний, средний</p>
          <p>William - быстрый, средний</p>
          <a href="/game/play"><div class="game-button-play">Начать</div></a>
        </div>
      </div>
    </div>

    <div class="footer">
      <!--
      <div class="footer-row">
        <div class="footer-col1">© 2017 PM Game</div>
        <div class="footer-col2">Свяжитесь с нами: company@gmail.com</div>
      </div>
    </div>
  </body>
</html>

```

Рисунок 9. Код страницы

После того как игрок готов приступить к игре ему необходимо нажать на кнопку «Начать» и он перейдет к самому игровому процессу (рис.10).

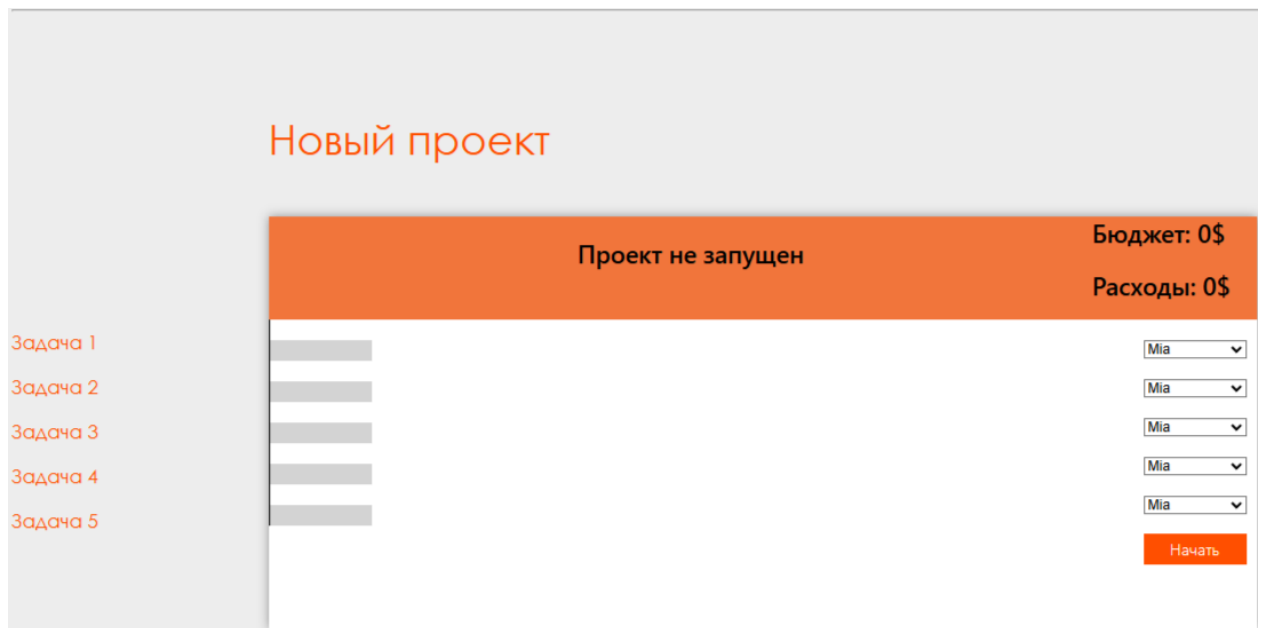


Рисунок 10. Игра

Код страницы:

```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv=«Content-Type» content=«text/html; charset=utf-8» /> <title>
    Cherry Cake  </title>
  <link href=«/favicon.ico» type=«image/x-icon» rel=«icon»/><link
href=«/favicon.ico» type=«image/x-icon» rel=«shortcut icon»/><link rel=«stylesheet»
type=«text/css» href=«/css/master.css»/><script type=«text/javascript»
src=«/js/interface.js»></script></head>
<body>
  <div class=«wrapper»>
    <div class=«header»>
      <div class=«navigation»>
        <div class=«navigation-col1»>
          <img src=«/img/logo.png»>
        </div>
        <div class=«navigation-col2»>
          <a id=«home» href=«/home»>Главная</a>
          <a id=«game» href=«/game»>Игра</a>
          <a id=«rules» href=«/rules»>Правила</a>
          <a id=«leaders» href=«/leaders»>Лидеры</a>
          <a id=«about» href=«/about»>О нас</a>
        </div>
        <div class=«navigation-col3»>
          <a href=«/auth/logout»><img src=«/img/avatar_casual.png»></a>
        </div>
      </div>
    </div>
    <hr>
  </div>

  <div class=«content»>
    <div style=«float: left;»>
      <p style=«font-size: 14pt; margin-top: 210px;» align=«right»>Задача 1</p>

```

```

<p style=«font-size: 14pt;»>Задача 2</p>
<p style=«font-size: 14pt;»>Задача 3</p>
<p style=«font-size: 14pt;»>Задача 4</p>
<p style=«font-size: 14pt;»>Задача 5</p>
</div>
<div class=«body» style=«margin: 100px 0px 100px 250px;»>
  <div class=«page»>
    <div class=«label»><p>Новый проект</p></div>
    <div class=«content»>
      <div class=«game-wrapper»>
        <div class=«top»>
          <div class=«top-left»>
            </div>
          <div class=«top-middle»>
            <p id=«state»>Проект не
запущен</p>
          </div>
          <div class=«top-right»>
            <p class=«counter»>Бюджет: 0$</p>
            <p class=«counter»>Расходы:
0$</p>
          </div>
        </div>
        <div class=«field»>
          <div class=«lead» style=«margin-
left:0px;»></div>
          <div class=«taskbar» style=«width:100px;
margin-left:0px;»>
            <div class=«progress»
style=«width:0px;»></div>
          </div>
          <div class=«taskbar» style=«width:100px;
margin-left:0px;»>
            <div class=«progress»
style=«width:0px;»></div>
          </div>
          <div class=«taskbar» style=«width:100px;
margin-left:0px;»>
            <div class=«progress»
style=«width:0px;»></div>
          </div>
          <div class=«taskbar» style=«width:100px;
margin-left:0px;»>
            <div class=«progress»
style=«width:0px;»></div>
          </div>
          <div class=«taskbar» style=«width:100px;
margin-left:0px;»>
            <div class=«progress»
style=«width:0px;»></div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```
<div class=«side»>
  <form>
    <select size=1 id=«sel1»>
      <option
value=«Mia»>Mia</option>
      <option
value=«Jack»>Jack</option>
      <option
value=«Kate»>Kate</option>
      <option
value=«William»>William</option>
    </select>
    <br>
    <select size=1 id=«sel2»>
      <option
value=«Mia»>Mia</option>
      <option
value=«Jack»>Jack</option>
      <option
value=«Kate»>Kate</option>
      <option
value=«William»>William</option>
    </select>
    <br>
    <select size=1 id=«sel3»>
      <option
value=«Mia»>Mia</option>
      <option
value=«Jack»>Jack</option>
      <option
value=«Kate»>Kate</option>
      <option
value=«William»>William</option>
    </select>
    <br>
    <select size=1 id=«sel4»>
      <option
value=«Mia»>Mia</option>
      <option
value=«Jack»>Jack</option>
      <option
value=«Kate»>Kate</option>
      <option
value=«William»>William</option>
    </select>
    <br>
    <select size=1 id=«sel5»>
      <option
value=«Mia»>Mia</option>
      <option
value=«Jack»>Jack</option>
      <option
```

```

value=«Kate»>Kate</option>
<option
value=«William»>William</option>
</select>
<br><br>
<div class=«button-game»
onclick=«Start()»> Начать</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
<script type=«text/javascript» src=«/js/gamerules.js»></script>
</div>
<div class=«footer»>
<!--<img src=«/files/woman_3.png» style=«position: absolute; bottom: -20px;
left: 1150px;»>-->
<div class=«footer-row»>
<div class=«footer-col1»>© 2017 PM Game</div>
<div class=«footer-col2»>Свяжитесь с нами: company@gmail.com</div>
</div>
</div>
</div>
</body>
</html>

```

Затем игрок назначает сотрудников на выполнение задач и запускает игровой процесс кнопкой «Начать». После запуска индикатор времени начнёт продвигаться вперёд, а ячейка задания постепенно окрасится зелёным цветом по мере завершения задачи. Если игроки успевают успешно выполнить все поставленные задачи, то получают выигрыш. Если же остаются невыполненные задачи — игра заканчивается поражением (рис.11).

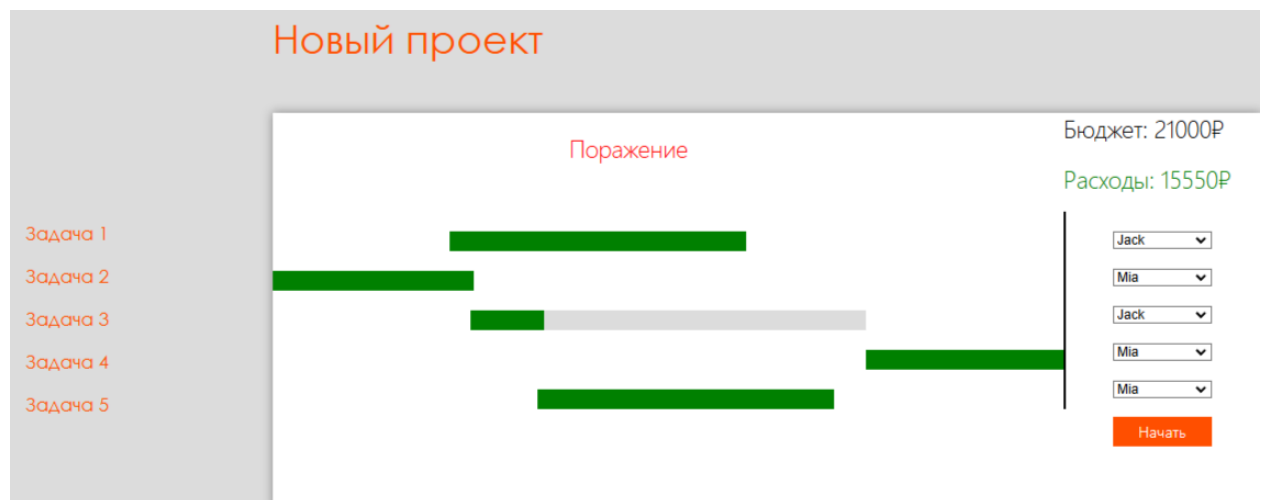


Рисунок 11. Поражение

Код механизма игры:

// Конфигурация игры

```
const GameConfig = {  
  initialBudget: 21000,  
  leadDuration: 800,  
  tickInterval: 100,  
  characterOptions: {  
    Mia: { speed: 6, cost: 100, luck: 1 },  
    Jack: { speed: 3, cost: 30, luck: 1 },  
    Kate: { speed: 4, cost: 70, luck: 1 },  
    William: { speed: 5, cost: 80, luck: 1 }  
  }  
};
```

// Состояние игры

```
const GameState = {  
  staff: [],  
  jobs: [],  
  lead: {  
    value: 0,  
    step: 5  
  },  
  budget: GameConfig.initialBudget,  
  cost: 0,  
  isRunning: false,  
  gameInterval: null,  
  isWon: false  
};
```

// DOM элементы

```
const GameElements = {  
  taskbars: document.getElementsByClassName('taskbar'),  
  progressBars: document.getElementsByClassName('progress'),  
  leadLine: document.getElementsByClassName('lead')[0],  
  costLabel: document.getElementsByClassName('counter')[1],  
  budgetLabel: document.getElementsByClassName('counter')[0],  
  gameStateLabel: document.getElementById('state'),  
  selectors: document.getElementsByTagName('select'),  
  startButton: document.querySelector('.button-game')  
};
```

// Инициализация игры при загрузке страницы

```
window.addEventListener('DOMContentLoaded', () => {  
  initializeGame();  
  GameElements.startButton.addEventListener('click', startGame);  
});
```

// Инициализация игровых заданий

```
function initializeGame() {  
  // Создаем массив заданий  
  GameState.jobs = Array(5).fill().map(() => ({  
    duration: 0,
```

```
        relation: -1,
        position: 0,
        value: 0,
        step: 0,
        isTriggered: false
    ));

    // Генерируем основные связанные задания
    const mainJobs = generateMainJobs();

    // Генерируем дополнительные задания
    generateAdditionalJobs(mainJobs);

    // Обновляем интерфейс
    updateGameUI();
}

function generateMainJobs() {
    const indexes = [
        getRandomInt(0, GameState.jobs.length - 3),
        getRandomInt(indexes[0] + 1, GameState.jobs.length - 2),
        getRandomInt(indexes[1] + 1, GameState.jobs.length - 1)
    ];

    const durations = [
        getRandomInt(1, 4) * 100,
        getRandomInt(1, (GameConfig.leadDuration / 100) - durations[0]/100 - 1) * 100,
        GameConfig.leadDuration - (durations[0] + durations[1])
    ];

    // Настраиваем основные задания
    GameState.jobs[indexes[0]] = createJob(durations[0]);
    GameState.jobs[indexes[1]] = createJob(durations[1], indexes[0]);
    GameState.jobs[indexes[2]] = createJob(durations[2], indexes[1]);

    return indexes;
}

function generateAdditionalJobs(mainIndexes) {
    GameState.jobs.forEach((job, i) => {
        if (!mainIndexes.includes(i)) {
            const duration = getRandomInt(1, 3) * 100;
            const position = getRandomInt(GameConfig.leadDuration * 0.2,
GameConfig.leadDuration * 0.5);
            GameState.jobs[i] = createJob(duration, -1, position);
        }
    });
}

function createJob(duration, relation = -1, position = 0) {
    const job = {
        duration,
```

```
    relation,
    position,
    value: 0,
    step: 0,
    isTriggered: false
  };

  if (relation !== -1) {
    job.position = GameState.jobs[relation].position + GameState.jobs[relation].duration;
  }

  return job;
}

// Запуск игры
function startGame() {
  if (GameState.isRunning) return;

  // Получаем выбранных персонажей
  GameState.staff = Array.from(GameElements.selectors).map(select => ({
    ...GameConfig.characterOptions[select.value],
    name: select.value
  }));

  // Настраиваем задания
  setupJobs();

  // Проверяем дублирование персонажей
  checkStaffDuplicates();

  // Запускаем игровой цикл
  GameState.isRunning = true;
  GameState.gameInterval = setInterval(gameTick, GameConfig.tickInterval);
  GameElements.gameStateLabel.textContent = «Проект в разработке...»;
  GameElements.startButton.disabled = true;
}

function setupJobs() {
  GameState.jobs.forEach((job, i) => {
    job.step = GameState.staff[i].speed;
    job.value = 0;
    job.isTriggered = false;
  });

  GameState.lead.value = 0;
  GameState.cost = 0;
  GameState.isWon = false;
}

function checkStaffDuplicates() {
  GameState.staff.forEach((char1, i) => {
    GameState.staff.forEach((char2, j) => {
```

```
        if (i !== j && char1.name === char2.name) {
            const job1 = GameState.jobs[i];
            const job2 = GameState.jobs[j];

            if (job1.position <= job2.position) {
                job2.relation = i;
            } else {
                job1.relation = j;
            }
        }
    });
});
}

// Игровой цикл
function gameTick() {
    updateLeadProgress();
    checkJobTriggers();
    updateJobProgress();
    updateGameCost();
    checkRandomEvent();
    checkGameEnd();
}

function updateLeadProgress() {
    if (GameState.lead.value < GameConfig.leadDuration) {
        GameState.lead.value += GameState.lead.step;
        GameElements.leadLine.style.marginLeft = `${GameState.lead.value}px`;
    }
}

function checkJobTriggers() {
    GameState.jobs.forEach((job, i) => {
        if (job.relation !== -1) {
            job.isTriggered = GameState.jobs[job.relation].value >=
GameElements.leadLine.style.marginLeft;
        } else {
            job.isTriggered = GameState.lead.value >= job.position;
        }
    });
}

function updateJobProgress() {
    GameState.jobs.forEach((job, i) => {
        if (job.isTriggered && job.value < job.duration) {
            job.value += job.step;
            GameElements.progressBars[i].style.width = `${job.value}px`;
        }
    });
}

function updateGameCost() {
```

```
    let newCost = 0;

    GameState.jobs.forEach((job, i) => {
      if (job.isTriggered && job.value < job.duration) {
        newCost += GameState.staff[i].cost;
      }
    });

    GameState.cost += newCost;
    GameElements.costLabel.textContent = `Расходы: ${GameState.cost}$`;
    GameElements.costLabel.style.color = GameState.cost < GameState.budget ? «green» :
«red»;
  }

  function checkRandomEvent() {
    // 20% шанс случайного события
    if (Math.random() < 0.2) {
      const charIndex = getRandomInt(0, GameState.staff.length - 1);
      if (Math.random() < 1/GameState.staff[charIndex].luck) {
        GameState.jobs[charIndex].step = GameState.staff[charIndex].speed / 2;
        showMessage(`Персонаж ${GameState.staff[charIndex].name} заболел. Скорость
работы снижена.`);
      }
    }
  }

  function checkGameEnd() {
    const jobsDone = GameState.jobs.filter(job => job.value >= job.duration).length;
    const allJobsDone = jobsDone === GameState.jobs.length;
    const leadFinished = GameState.lead.value >= GameConfig.leadDuration;
    const budgetExceeded = GameState.cost > GameState.budget;

    if (allJobsDone || leadFinished) {
      endGame(allJobsDone && !budgetExceeded);
    }
  }

  function endGame(isWon) {
    clearInterval(GameState.gameInterval);
    GameState.isRunning = false;
    GameState.isWon = isWon;

    GameElements.gameStateLabel.textContent = isWon ? «Победа!» : «Поражение»;
    GameElements.gameStateLabel.style.color = isWon ? «green» : «red»;

    // Отправляем результаты на сервер
    saveGameResult(isWon);
  }

  function saveGameResult(isWon) {
    const result = {
      won: isWon,
```

```

login: 'vecryd', // Здесь должно быть реальное имя пользователя
cost: GameState.cost,
timestamp: new Date().toISOString()
};

// Здесь должна быть реализация отправки данных на сервер
console.log('Game result:', result);
}

// Вспомогательные функции
function getRandomInt(min, max) {
  return Math.floor(Math.random() * (max - min + 1)) + min;
}

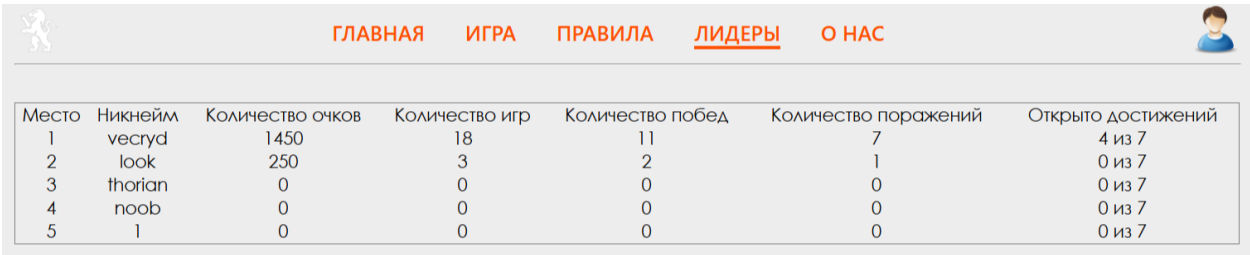
function updateGameUI() {
  GameState.jobs.forEach((job, i) => {
    GameElements.taskbars[i].style.width = `${job.duration}px`;
    GameElements.taskbars[i].style.marginLeft = `${job.position}px`;
    GameElements.progressBars[i].style.width = '0px';
  });

  GameElements.budgetLabel.textContent = `Бюджет: ${GameState.budget}$`;
  GameElements.costLabel.textContent = `Расходы: 0$`;
  GameElements.costLabel.style.color = «green»;
  GameElements.leadLine.style.marginLeft = «0px»;
  GameElements.gameStateLabel.textContent = «Проект не запущен»;
  GameElements.gameStateLabel.style.color = «black»;
  GameElements.startButton.disabled = false;
}

function showMessage(text) {
  alert(text); // Можно заменить на более красивый вывод сообщений
}

```

Так же есть возможность наблюдать за таблицей результатов игроков (рис.12).



Место	Никнейм	Количество очков	Количество игр	Количество побед	Количество поражений	Открыто достижений
1	vecryd	1450	18	11	7	4 из 7
2	look	250	3	2	1	0 из 7
3	thorian	0	0	0	0	0 из 7
4	noob	0	0	0	0	0 из 7
5	l	0	0	0	0	0 из 7

Рисунок 12. Рейтинг игроков

4. Выводы

Была успешно разработана и реализована образовательная игра, направленная на обучение основам управления проектами с использованием диаграмм Ганта. Результаты можно использовать как для обучения основам управления проектами, так и в развлекательных целях.

Библиографический список

1. Штанюк А.А. Использование программы управления проектами taskjuggler в учебном процессе // В сборнике: Наука и образование в современном обществе: вектор развития Сборник научных трудов по материалам Международной научно-практической конференции: В 7 частях. ООО "Ар-Консалт". 2014. С. 144-146.
2. Свентицкий П., Иванова Н.А. Инструменты кроссплатформенной разработки мобильных приложений // Инновации в науке. 2014. №40. С. 58-62
3. Мешалкин В.П., Стоянова О.В., Дли М.И. Управление проектами в сфере нанотехнологий: особенности и возможности их учета // Теоретические основы химической технологии. 2012. Т. 46. № 1. С. 56.
4. Новакова Н.Е., Горячев А.В., Горячев А.А. Концепция управления проектами в САПР // Программные системы и вычислительные методы. 2013. № 3. С.6-6.
5. Олейникова С.А. Модификация метода PERT решения задач сетевого планирования и управления // Системы управления и информационные технологии. 2008. Т. 34. № 4. С. 42-45.
6. Чусавитина Г.Н., Макашова В.Н. Использование информационных технологий в управлении проектами: учеб. пособие. Магнитогорск, 2011. 216 с.
7. Широкова С.В. Управление проектами. Управление проектами внедрения информационных систем для предприятия. Учебное пособие. СПб.: Санкт-Петербургский гос. политехнический ун-т, 2012.