

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
федеральное государственное бюджетное образовательное учреждение высшего
образования
КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ
им.В.П.АСТАФЬЕВА
(КГПУ им.В.П.Астафьева)

Институт/факультет

Институт математики, физики и информатики
(полное наименование института/факультета/филиала)

Выпускающая кафедра

Базовая кафедра информатики и
информационных технологий в образовании
(полное наименование кафедры)

Колмогоров Андрей Эдуардович

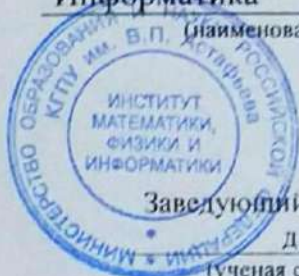
ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема **Новая технология создания презентаций научно-технического
содержания**

Направление подготовки 44.03.01 Педагогическое образование
(код и наименование направления)

Профиль

Информатика
(наименование профиля для бакалавриата)



ДОПУСКАЮ К ЗАЩИТЕ

Заведующий кафедрой

д.п.н., профессор Пак Н.И.

(ученая степень, ученое звание, фамилия, инициалы)

(дата, подпись)

Руководитель к.ф.-м.н., доцент базовой кафедры
Романов Д.В.

(ученая степень, ученое звание, фамилия, инициалы)

Дата защиты

Обучающийся Колмогоров А.Э.
(фамилия, инициалы)

(дата, подпись)

Оценка

удовлетворительно
(прописью)

Красноярск 2018

Оглавление	
Введение.	3
Глава 1. Основные требования к презентациям научно-технического характера.	5
1.1. Особенности восприятия при работе с новым материалом.	5
1.2. Онлайн-среды создания и демонстрации презентаций.	7
Prezi.	7
Learning Apps.	8
Latex.	9
1.3. Современные веб-технологии.	10
Глава 1. Презентация наукоемкого материала.	11
1.1. Latex.	12
1.2. Beamer.	12
1.3. Html,css,js	14
1.4. React JS.	15
Глава 2. Новая среда для презентаций.	16
2.1 Язык описания анимации.	16
2.2. Установка среды.	17
2.3. Анализ примеров.	20
Заключение.	25
Библиографический список.	27

Введение

В настоящее время требования к качеству и объему дидактического материала непрерывно растут, информационная нагрузка на обучающегося также непрерывно увеличивается.

Предметы становятся сложнее, глубина и сложность изучаемого на уроках материала, особенно в современных разделах информатики, увеличиваются. Актуализация межпредметных связей приводит к использованию на уроке концепций и понятий из смежных областей и дисциплин.

Облегчить процесс усвоения материала в таких условиях призваны различные дидактические материалы и методики ведения урока. Из них можно выделить:

- прямые демонстрации учителем (фронтальная диалоговая форма работы);
- обучающее видео с опорой на презентацию или демонстрацию;
- рабочие тетради, разнообразный раздаточный материал;
- книги, учебники, методические пособия.

Презентация среди дидактических материалов занимает особое место – она обладает всеми достоинствами текста: легко форматируется, быстро пишется (не так трудоемка в написании, как книга). Их используют в МООС и при фронтальной работе в классе. Презентации можно использовать для самостоятельной работы, как краткий конспект.

Грамотно сделанная презентация способна структурировать понятийное поле, помочь осознать связи между понятиями и разобраться в сложных многоуровневых понятиях.

С появлением ПК презентации приобрели возможность использовать мультимедиа-материалы; одновременная работа со звуком, видео, картинками, текстом, анимацией, интерактивностью открывает новые,

широкие возможности для захвата и удержания внимания, стимуляции зрителя.

Из всех мультимедийных возможностей наиболее полезной, на взгляд автора, является анимация. Этот инструмент позволяет при адекватном использовании фокусировать внимание ученика на ключевых элементах и положениях излагаемого тезиса, непрерывно поддерживая нужную степень мотивации и заинтересованности.

В настоящее время озвучен заказ государства на подготовку в старшей школе обучающихся, обладающих развитыми инженерными, ИТ, и другими компетенциями на очень высоком уровне, хорошо показанным требованиями олимпиады НТИ [15, 13, 1].

При подготовке таких кадров необходимо, помимо развития навыка анализа и решения задач, периодически быстро, в сжатые сроки, разбирать инженерные и наукоемкие концепции до уровня понимания и применения на практике. Всё это ужесточило требования к презентациям, используемым в классе, в MOOC и ЦОР.

В главе 1 рассмотрены основные техники построения современных презентаций. Проведен анализ требований, успешных решений таких задач, как создание текста, насыщенного математикой, и анимирования процесса работы сложного алгоритма или механизма.

В главе 2 описана платформа, построенная на базе веб-технологий и фреймворка ReactJS. Приведены примеры решения модельных задач, дан анализ, рекомендации по установке и использованию.

В заключении приведен обзор полученных результатов.

Глава 1. Основные требования к презентациям научно-технического характера.

1.1. Особенности восприятия при работе с новым материалом.

При демонстрации новых положений крайне желательно минимизировать количество шума при сохранении полноты высказываний. Также следует учитывать, что кратковременная память способна удерживать порядка 7 ± 2 элементов при обработке потока поступающей информации (“кошелёк Миллера”); большие объёмы информации можно группировать иерархически в блоки по ~ 7 элементов на каждом уровне, удерживая объём фразы до разумного минимума [7].

Процесс работы с иерархиями понятий, или процесс изучения нетривиального алгоритма или модели, описывающей изучаемую систему, при изучении в классе с использованием статической демонстрации и пояснений требует развитого навыка коммуникации, ораторского мастерства от педагога, рабочей атмосферы в классе. Любые возмущения этих условий приводят к потере концентрации и хода мысли учителя.

Аккуратное использование анимации для постепенного введения новых элементов при сохранении содержимого слайда (контекста) заметно облегчает следование за мыслью автора и выделение поступающих новых элементов. Особенно хорошо такая техника помогает осветить различные алгоритмы (например, обхода графов, выполнения однотипных действий при проведении математической или численной операции).

Единственный недостаток такого подхода заключается в том, что, традиционные инструменты создания презентаций такого уровня сложности требуют непропорционально много ручного труда, поскольку автор вынужден создавать руками и/или менять параметры каждого элемента в течение многих шагов алгоритма для того, чтобы ученик мог легко уловить закономерность, - фактически, он превращается в исполнителя, выполняющего много однотипных действий по изучаемому алгоритму. Как

следствие, это привело к тому, что множество людей стало исследовать и предлагать альтернативы программам, построенным по принципам PowerPoint. Последний также страдает из-за фрагментации рынка, вызванной большим количеством несовместимых версий разного возраста и наличием конкурентов.

Одно направление из альтернативных решений - онлайн среды: Prezi.com, LearninApps. Второе - логическое форматирование материала вместо визуального - издательская система LaTeX [5], troff [14], html и другие. Все они будут рассмотрены ниже.

1.2. Онлайн-среды создания и демонстрации презентаций.

Prezi.

Prezi позволяет создавать, редактировать и делиться своими презентациями, где и когда вы хотите. Можете потратить лишь пару минут, чтобы разработать презентацию для важной встречи. Сервис поможет сделать финальные штрихи [11].

Вместо слайдов пользователи Prezi пользуются специальными полями, на которых и располагается вся информация (данные, снимки, видео, диаграммы), чтобы её в дальнейшем распределить в нужном порядке для удобства аудитории. Все элементы доступны для приближения, поворотов и прочих эффектов.

Компания разработчиков регулярно улучшает собственный сервис. Теперь можно найти и вставить в презентацию фото из Google, не покидая Prezi, или таким же образом обработать снимки.

Но как и у любого сервиса, есть свои недостатки:

- Использование Prezi требует определенной визуальной грамотности. В противном случае презентация может напоминать свалку разнородных по стилю элементов или, скажем более корректно, калейдоскоп. В Prezi это более заметно, чем в PowerPoint.
- В процессе подготовки презентации легко увлечься творческим представлением информации вместо смыслового. Это приведет, минимум, к потере времени. Максимум - к презентации с непонятным содержанием.
- Во время проведения презентации перед аудиторией, участники начинают переключать свое внимание от содержания на необычный формат ее представления. Соответственно, они отвлекаются, и результат проведения может значительно отличаться от запланированного.

- Некоторые участники презентации будут испытывать определенный эмоциональный дискомфорт из-за нарушения стандартного представления информации. Другие не смогут в привычном режиме делать пометки и т.д.

- Замечено также, что из-за динамики представления материалов презентации Prezi, изменения их ориентации в пространстве и масштабирования, могут возникнуть определенные проблемы со здоровьем у участников. Так возможны головокружения и даже приступы морской болезни. В связи с этим не стоит превращать презентацию Prezi в американские горки.

- Prezi - англоязычный сервис. Поэтому есть ограничения по использованию шрифтов. Microsoft PowerPoint в этом отношении ограничений не имеет, так как можно пользоваться всеми шрифтами, установленными в системе. А их, обычно, от нескольких сотен до нескольких тысяч.

Вывод.

Prezi.com ресурс хороший, но больше нацелен на красоту представления, а не на краткость и чёткость представляемых материалов.

Learning Apps.

Сервис предназначен для создания разного рода интерактивных заданий и составления ментальных карт. Очень хорошо подходит для индивидуальных заданий [6].

В нашем случае имеется ряд недостатков:

- Не подходит для фронтальной работы, так как обычно весь объем информации показан сразу на экране. Можно показывать частями, но это всё равно не очень удобно из-за не точного перемещения в нужное нам место на которое тратится время.

- Обязательное подключение к интернету.

· Установка дополнительного программного-обеспечения, которое далеко не везде присутствует.

Вывод.

На самом деле сервис хорош как для составления ментальных карт, которые очень хорошо подходят для видимости всей картины в целом. Но если нам нужно поделить на очень мелкие части выдаваемый материал, то увы, ничем не поможет.

Latex.

Другое, крайне интересное решение – язык вёрстки страницы не в визуальной среде, а в текстовой, как в языке программирования, после чего это представление конвертируется в pdf или PostScript. Это язык LaTeX.

Использование логического форматирования решает несколько важных вопросов:

- точное управление размерами структурных элементов текста;
- крайне удобный, быстрый и прозрачный ввод математических выражений;
- возможность переиспользовать любой фрагмент материала с помощью обычных копирования и вставки текста.

Для LaTeX-а создана очень остроумная система вёрстки слайдов - Beamer. При использовании этого пакета любой элемент страницы можно аннотировать интервалами кадров, когда он должен быть виден, скрыт или отсутствовать. Грамотно спроектированное API позволяет по одному размеченному слайду сгенерировать в выходном pdf-файле всю последовательность кадров анимации. Такие презентации хорошо конвертируются в pdf и при просмотре по слайдам можно получить эффект анимации, но всё же это просто набор слайдов.

При использовании latex+beamer, получается идеальная типографика, всё очень ровно, структурно и никуда не “едет”.

Но у LaTeX-а тоже есть свои недостатки:

- неудобно работать с видео- и мультимедиа-материалом;
- не очень удобно строить схемы, таблицы, рисунки;
- узкая популярность языка, отсутствие преемственности в большинстве образовательных учреждений (популярен в основном среди академических ученых).

- Для того чтобы нам нарисовать сложный качественный рисунок, нам надо освоить уже специализированный сложный язык, который если часто не использовать, быстро забывается, а человек, который редко делает презентации такой язык учить не будет.

Коллективный опыт показывает, что с превышением объема или уровня сложности статьи в научном журнале (20 перекрестных ссылок и формулы) альтернативы LaTeX нет.

Сам язык оказался настолько удачным, что был перенесен на множество других платформ и ниш: MathJax [8].

1.3. Современные веб-технологии.

Недостатки и малое распространение LaTeX привели к тому, что было создано несколько альтернатив. Ключевую роль в их появлении является бурное развитие web-технологий и возможностей браузеров [4]. Оформление языка ECMAScript в полноценный язык программирования и коммерческая привлекательность профессии веб-программиста привели к взрывному росту количества доступной информации, библиотек и ресурсов.

В том числе была написана библиотека MathJax, использующая все наработки LaTeX и AMS.

Веб-презентации не имели ряда недостатков латеха. В настоящее время:

- Можно работать с формулами как в latex + beamer, с помощью библиотеки MathJax;
- Легкая работа с таблицами;
- Быстрая работа с картинками, в частности с векторными(svg);
- Логическая разметка;
- Можно просматривать на любом компьютере, где есть браузер;
- Так же легко экспортируется в pdf;
- Можно использовать любые новые технологии web-разработки(webGL);
- Доступный исходный код;
- Html очень известен, его знает достаточно много людей;
- Очень много обучающих материалов;
- Простое и понятное, легко поправимое оформление слайдов;
- Можно использовать непосредственно интерактивность с помощью программирования. Например, как тут [3];
- Легкая работа со ссылками;
- Легко переиспользовать любой слайд и вносить поправки;
- Работа с поправками, с помощью некоторых инструментов (программа Meld [9]) намного удобнее чем с документами MS;

Библиотеки MathJax, Shower и аналоги позволили заменить язык TeX в основании, LaTeX-а на html5.

Целью настоящей работы было проанализировать инженерные решения, обеспечивающие успех Beamer, и перенести их на WEB-платформу.

Глава 1. Презентация наукоёмкого материала.

1.1. Latex.

LaTeX (произносится как латех) — наиболее популярный набор макрорасширений (или макропакет) системы компьютерной вёрстки TeX, который облегчает набор сложных документов.

Представляет собой язык разметки. Пишется в текстовом файле с расширением .tex с помощью простого текстового редактора, но есть и специализированные редакторы.

Выглядит язык как набор заготовленных заранее слов и знаков:

`\frac{\partial f}{\partial x} = 2\sqrt{a}x`

Из такого набора получается вот такая формула.

$$\frac{\partial f}{\partial x} = 2\sqrt{a}x$$

На первый взгляд может показаться, что не очень-то он и прост, но на самом деле довольно быстро учится и на много удобнее и быстрее набирать таким образом формулы чем на разных визуальных сервисах, которые переводят весь код в картинку. В LaTeX всё зависит от твоей скорости печати, а не от щелканья мышки.

1.2. Beamer.

LaTeX годится не только для составления всевозможной документации, но и для верстки качественных презентаций. При помощи пакета beamer можно создавать презентации, ничуть не уступающие презентациям PowerPoint или Impress [2].

Beamer – специальный класс для LaTeX, с помощью которого можно делать слайды, анимации, иллюстрации.

Имеет возможность определять слайд (frame) и синтаксис для управления цветом и порядком появления элементов многочастного слайда. Презентации очень чётко структурированы с помощью специальных «тегов»:

разделы (section), параграфы (subsection), списки, по которой удобно ориентироваться / перемещаться во время презентации.

Структура, а также фамилия докладчика/ов и название доклада могут быть представлены на полях слайдов. Там, где деление на фреймы не введено, Beamer „нарезает” текст на страницы сам (по правилам Tex'a). Есть возможность переходить по гиперссылкам (пакет подключен по умолчанию) и определять кнопки для переходов. Для определений, теорем и т.п. оформляются блоки (напоминающие наклейки-стикеры) определённого цвета. Существует набор готовых цветовых/структурных сочетаний/представлений, которые кодируются названиями животных/городов. Кроме того, поскольку LaTeX имеет огромные возможности для написания математических формул, beamer представляет пользователю всю мощь LaTeX'a для создания научных презентаций.

Код для анимации на beamer выглядит примерно так:

```
\newcount\ooo
\newdimen\offset
\begin{frame}
  \animate<2-10>
  \animatevalue<1-10>\ooo{100}{0}
  \animatevalue<1-10>\offset{0cm}{5cm}
  \begin{colormixin}{\the\ooo!averagebackgroundcolor}
  \frametitle{Цветной слайд}
  выцветаем
  \hskip\offset\alert{и улетаем!}
  \end{colormixin}
  \transduration{0.1}
\end{frame}
```

Выглядит очень структурно и понятно.

Слайд с использованием beamer выглядит примерно так:

Название слайда

Нумерованный список:

- Раз
- Два
- Три

Пример

Текст примера

$$\mathbf{LoG}(\sigma; x, y) = \nabla^2 \mathbf{G} = \frac{1}{\sigma^3 \sqrt{2\pi}} \left(2 - \frac{x^2 + y^2}{\sigma^2} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}}. \quad (1)$$

Комбинация фильтров Лапласа и Гаусса и есть оператор лапласиана гауссианы (1).



1.3. React JS.

React — библиотека для браузерного языка программирования JavaScript, позволяющая программировать в удобном стиле для браузеров. Помогает создавать пользовательские интерфейсы [12].

Его главная задача — обеспечение вывода на экран того, что можно видеть на веб-страницах. React значительно облегчает создание интерфейсов благодаря разбиению каждой страницы на небольшие фрагменты. Мы называем эти фрагменты компонентами.

Вот пример разбивки страницы на компоненты:



Каждый выделенный фрагмент страницы, показанной на рисунке, считается компонентом.

Глава 2. Новая среда для презентаций.

2.1. Язык описания анимации.

Основная идея взята у пакета Beamer и реализована средствами ReactJS. Использована возможность любого парного JSX элемента получать содержимое html/jsx разметки в виде массива jsx элементов `this.props.children` и уже готового набора библиотечных функций для обхода и обработки этого массива.

Осталось только написать рекурсивную функцию, способную обходить html-дерево произвольной глубины с применением функции-фильтра, анализирующей атрибуты каждого узла. Эта функция приведена ниже:

```
// Deep copy of an React element.
```

```
export function recursiveMap(element, fn) {  
  let clone = (child) => (  
    React.cloneElement(  
      child,  
      [],  
      React.Children.map(  
        child.props.children,  
        (child) => mapper(child, fn)  
      )  
    )  
  )  
  ;  
  
  let mapper = (child) => {  
    if (!React.isValidElement(child)) {  
      return child;  
    }  
  
    let new_child = fn(child);
```



```
if (!new_child)
  return ;

if (new_child.props.children) {
  return clone(new_child);
}

return new_child;
};

if (element.props.children) {
  return clone(element);
} else {
  return element;
}
}
```

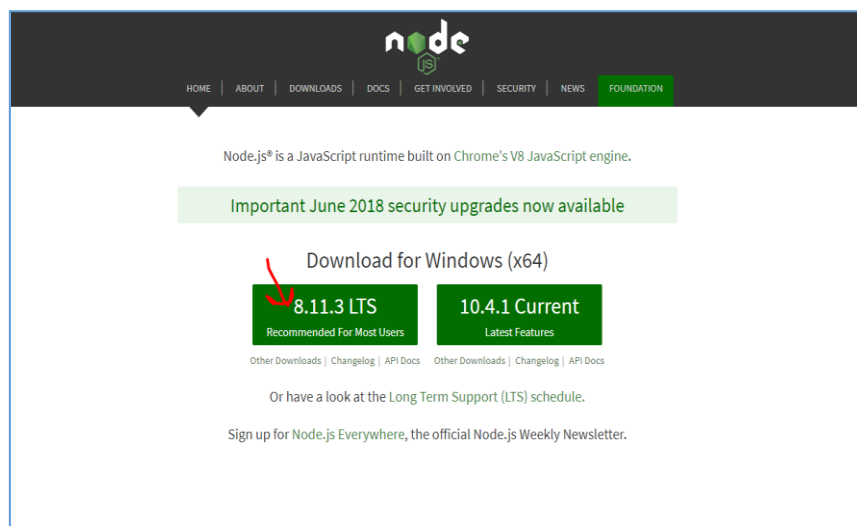
Теперь имеется возможность глубокого копирования (параметризованного номером кадра анимации) потомка в экспортный вариант страницы.

2.2. Установка среды.

Методические рекомендации по установке и работе со средой.

Установка среды.

1. Качаем node js, с сайта [10].



2. Далее устанавливаем по умолчанию, как обычную программу.

3. Открываем командную строку. Пишем команду `npm install`.

4. Устанавливаем нужное нам приложение глобально **npm install -g create-react-app**

5. Далее в командной строке переходим в папку где будет храниться приложение и пишем команду **create-react-app slideshow**

где `slideshow`—это название проекта. Когда все установится, вы увидите такое сообщение:

```
Success! Created slideshow at C:\Users\Lenovo\Desktop\create-react-app\slideshow
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd slideshow
  npm start

Happy hacking!
```

6. Далее нам нужно перейти в папу с приложением и один файл заменить, один файл добавить.

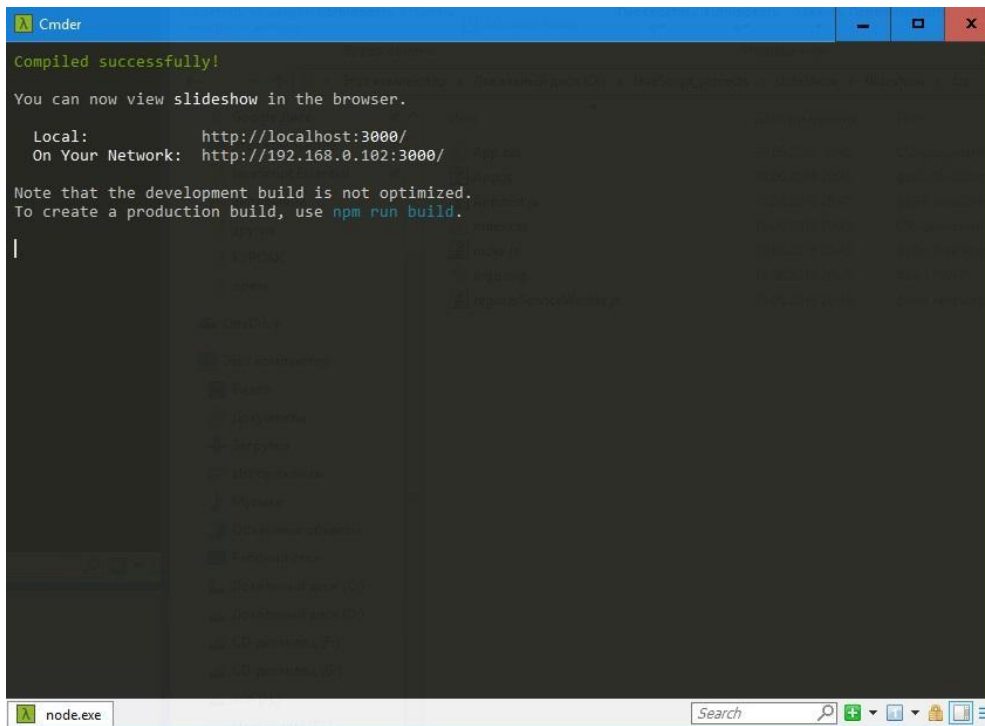
7. Файлы вы найдете по ссылке.

8. Заменить надо `App.js` и добавить `Slide.js`.

9. Среда готова, теперь переходим в командной строке в нашу папку и пишем команду **npm start**.

Должно появиться вот такое сообщение и автоматически браузер откроет наше

приложение.



10. Среда готова, теперь мы можем приступить к разработке.

11. Откроем файл App.js. В месте указанном на рисунке мы пишем свой html код.

```
    } else if (e.key === 'ArrowRight' && N < 100) {
      this.setState({
        N: N + 1
      })
    }
  }
}

componentWillMount() {
  document.addEventListener("keydown", this.handleKeyDown, false);
}

componentWillUnmount() {
  document.removeEventListener("keydown", this.handleKeyDown, false);
}

render() {
  const slide_left = <Slide frame={0}>
    HTML код
  </Slide>

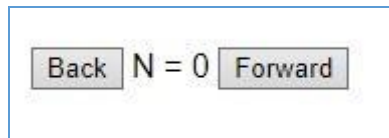
  const slide_right = React.cloneElement(
    slide_left,
    {frame: this.state.N}
  );
```

12. В файле App.css мы пишем CSS стили.

13. В React.js мы вместо атрибута `class=""`, мы пишем `className=""`.

14. Чтобы сделать анимацию, надо в теге добавить атрибуты с номерами в нужной нам последовательности. Атрибут выглядит вот так `num={1}`, в фигурных скобках пишем номер шага при анимации.

15. Управлять анимацией можно с помощью стрелок «влево», «вправо» на клавиатуре или кнопок в самом браузере.

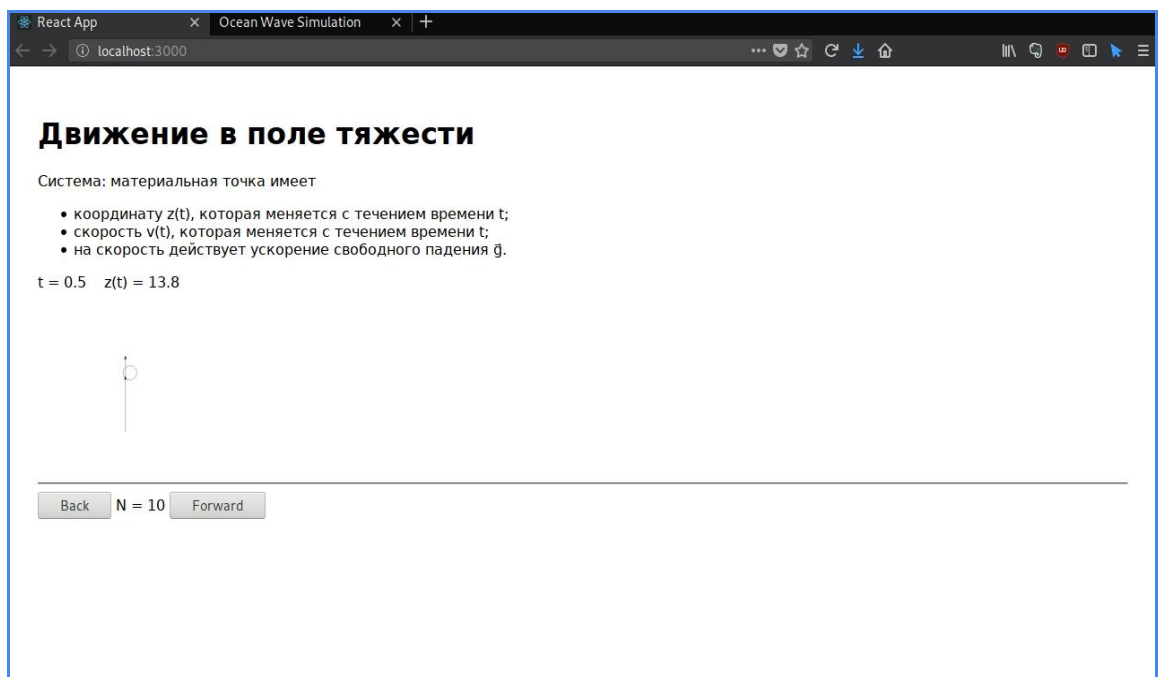


2.3. Анализ примеров.

Демонстрация среды.

Пример 1. Движение тела.

В данном примере показывается анимация движения тела в поле тяжести.



В работе пример можно посмотреть, запустив среду и добавив нужные файлы.

Пример 3. Сложение в столбик.

$$\begin{array}{r} 343 \\ + 578 \\ \hline 21 \end{array}$$

The image shows a vertical addition problem. The numbers 343 and 578 are written in green. Above the 343, there are two red '1's, one above the 4 and one above the 3. The result 21 is written in blue below a horizontal line.

- 1) $3 + 8 = 11$
- 2) $4 + 1 + 7 = 12$

[Back](#) N = 4 [Forward](#)

Здесь демонстрируется пошаговое сложение в столбик. Посмотрим на исходный код.

Заключение.

Созданная система написания веб-презентаций объединяет достоинства web-платформы и пакета Beamer, расширенные использованием хорошо апробированного фреймворка ReactJS, как раз и предназначенного для удобного создания веб-интерфейсов любой сложности. Использование html5 как основы позволит получить доступ к быстро развивающемуся сообществу и написанными им библиотеками для самых разных нужд - от 2D/3D интерактивной графики, математики типографского качества, множества редакторов, 2D и 3D движков, самым современным возможностям визуализации практически любого материала, вплоть до VR и AR технологий на всех операционных системах, включая мобильные.

Все эти, как и новые разработки, теперь автоматически станут доступными для использования при решении учебных и методических задач.

Перенос работы на уровень исходного html/jsx кода стал возможен только благодаря огромному прогрессу в этой стремительно развивающейся области, нацеленной сейчас на упрощение процесса работы, создания более ёмких, компактных, простых и мощных версий языка, созданию мощных и удобных средств разработки.

Дополнительно, язык ECMAScript впитал в себя лучшие наработки множества проработанных промышленных языков (Си++, Lisp, Perl, Python, Lua, Smalltalk, ...) и несравненно удобнее языка TeX как язык общего назначения, благодаря чему, в отличие от Beamer, мы можем генерировать сам визуализационный материал непосредственно с помощью изучаемого алгоритма или модели, поднимаясь выше на один уровень абстракции, используя гибкий выразительный синтаксис ES6, циклы, условия, библиотеки и т.п. Теперь на слайде вместо иллюстрации физической модели, работы алгоритма или инженерного устройства можно выполнять моделирование самой физической системы с 3D визуализацией с аппаратным

ускорением графики, выполнять работу по алгоритму с визуализацией состояния управляемой системы, давать интерактивную модель самого устройства.

Всё это позволит как создавать дидактические материалы принципиально нового уровня, просто переиспользуя готовые наработки сообщества, так и переносить само обсуждение на модель, описываемую языком ECMAScript с мгновенным изучением результата изменений, обеспечивающим быструю обратную связь.

Библиографический список.

- [1] Aero [Электронный источник] URL: <http://nti-contest.ru/profiles/aero/>
- [2] Beamer, Wikipedia. [https://ru.wikipedia.org/wiki/Beamer_\(LaTeX\)](https://ru.wikipedia.org/wiki/Beamer_(LaTeX))
- [3] Cloth [Электронный источник] URL:
<https://codepen.io/dissimulate/pen/KrAwx>
- [4] Html 5 [Электронный источник] URL:
<https://www.youtube.com/watch?v=DZnaVp3CkMg&t=>
- [5] LaTeX, Wikipedia <https://ru.wikipedia.org/wiki/LaTeX>
- [6] LearningApps <https://learningapps.org>
- [7] Magic of numbers [Электронный источник] URL:
https://ru.wikipedia.org/wiki/Магическое_число_семь_плюс-минус_два
- [8] Mathjax <https://en.wikipedia.org/wiki/MathJax>
- [9] Meld <https://ru.wikipedia.org/wiki/Meld>
- [10] Node [Электронный источник] URL: <https://nodejs.org>
- [11] Prezi [Электронный источник] URL: <https://prezi.com>
- [12] React, Wikipedia [Электронный источник] URL:
<https://ru.wikipedia.org/wiki/React>
- [13] Smarthouse [Электронный источник] URL: <http://nti-contest.ru/profiles/smarthouse/>
- [14] Troff [Электронный источник] URL:
<https://en.wikipedia.org/wiki/Troff>
- [15] Water robosystem [Электронный источник] URL: <http://nti-contest.ru/profiles/wrs/>