**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**ОТЧЁТ**

Предметная область: «Разработка интерактивных заданий для когнитивных тестов проекта 4К»

По дисциплине: «УП.05 Учебная практика»

Специальность 09.02.07 Информационные системы и программирование курс 3

Выполнил:

Ольшевский Даниил Андреевич

Проверил преподаватель:

Гжегожевский С.В.

**2023 год**

**Оглавление**

[Общие сведения 3](#_Toc154167380)

[1. Предметная область 3](#_Toc154167381)

[2. Постановка задачи 3](#_Toc154167382)

[Требования к проекту 3](#_Toc154167383)

[Структура заданий 4](#_Toc154167384)

[Макеты заданий 4](#_Toc154167385)

[1. Скриншоты макета “Спектакль” 4](#_Toc154167386)

[2. Скриншоты макета “Дино”: 4](#_Toc154167387)

[3. Скриншоты макета “Торт”: 5](#_Toc154167388)

[Реализация сборки заданий и развертывание проекта 5](#_Toc154167389)

[1. Сборка заданий. 5](#_Toc154167390)

[2. Развертывание проекта. 6](#_Toc154167391)

[Ссылки на задания 6](#_Toc154167392)

[Верстка заданий 6](#_Toc154167393)

# Общие сведения

## Предметная область

4К-тест — это инструмент диагностики универсальных (мягких) навыков, который позволяет определить уровень креативности, критического мышления, коммуникации и кооперации. Эти навыки необходимы, чтобы ученики могли успешно адаптироваться в современном мире: учиться новому, критически оценивать информацию, работать в команде и нестандартно решать задачи. Вы можете выбрать, какие навыки хотите измерить: какой-то один, несколько или все четыре.

4К-тест определяет исходный уровень навыков и позволяет измерить прогресс в их развитии. Тест разработан для учеников 4-8 классов. Время тестирования - минимум один школьный урок.

## **Постановка задачи**

Программирование тестовых заданий по оценке правовой грамотности для учащихся 7-х и 9-х классов. Сценарии предоставляются заказчиком в текстовом формате doc. Программирование осуществляется с использованием фреймворка Vue JS.

# Требования к проекту

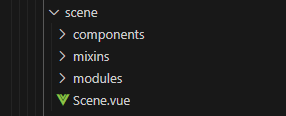
Тех. требования: поддержка Google Chrome последних версий (>= 120.0.6099.62); Разрешение экрана 1024x768; поддержка ввода с клавиатуры/мыши.

Задания представляют собой симуляторы с логикой распространенных приложений: браузер, мессенджер, графический редактор, блокнот и тп. В заданиях должен быть предусмотрен сбор индикаторов для их последующей отправки в систему скоринга на базе байесовской сети. Индикатор – элемент ассоциативного массива (ключ/значение), где ключ – уникальный идентификатор узла байесовского графа строчного типа, и значение – дихотомическое (0 или 1) или политомическое (0, 1, 2, …) представление числового типа. Задания должны состоять из нескольких модулей (отдельных изолированных частей/экранов), в соответствии со сценарием.

Необходимо разработать след. задания:

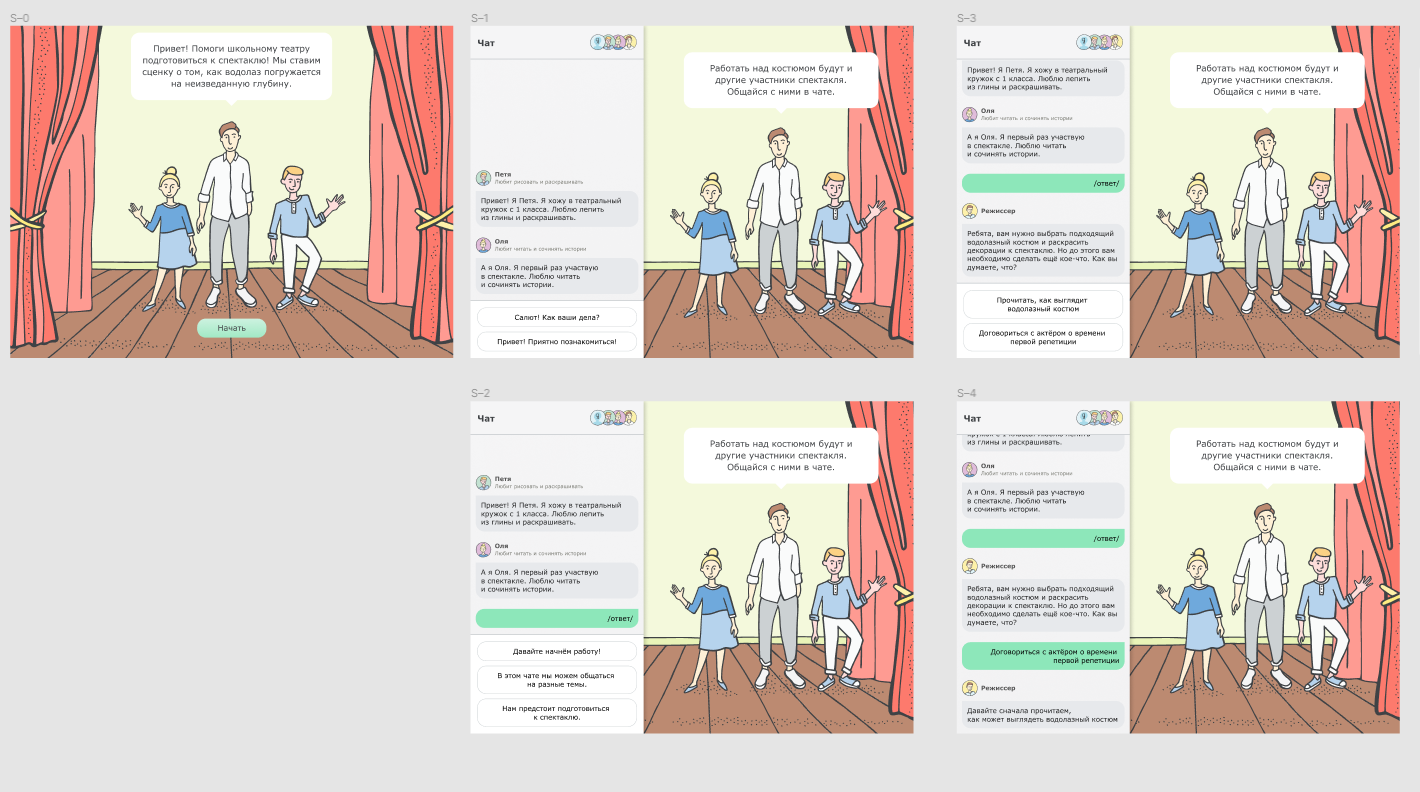
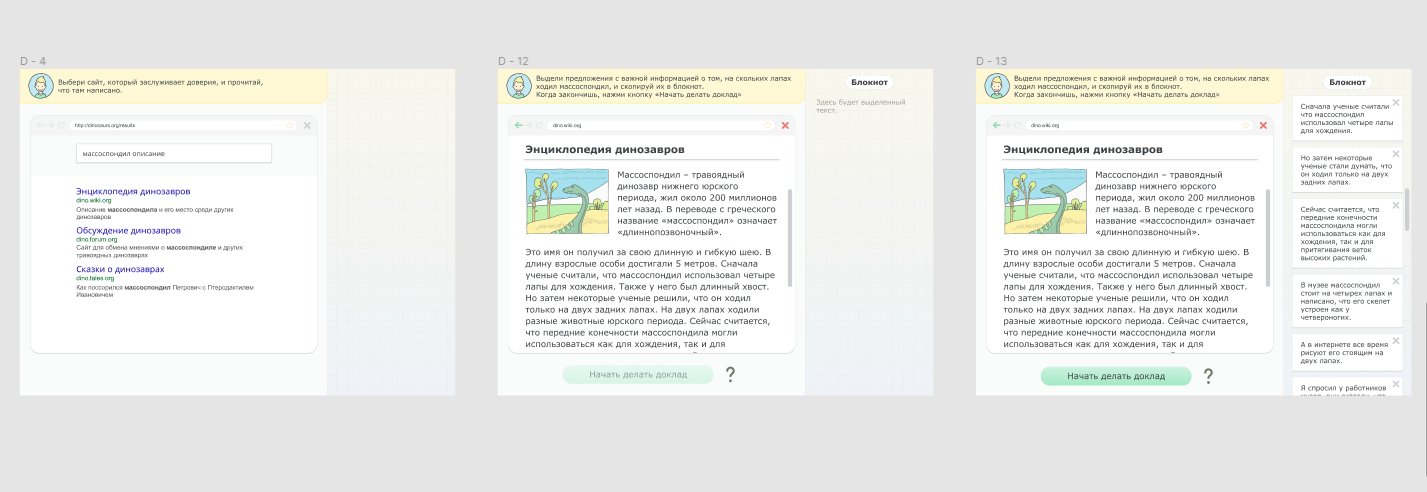
1. Спектакль – персонаж задания общается с детьми из своего театрального кружка, используя симулятор мессенджера, о подготовке к спектаклю. В рамках задания нужно помочь наставнику распределить задачи между детьми и выбрать готовый костюм водолаза для сцены;
2. Дино (Англ. версия) – нужно используя симулятор браузера собрать ключевую информацию о динозавре Массоспондил и подготовить о нём доклад;
3. Торт – наш персонаж общается с одноклассниками. В рамках задания нужно обсудить, как и какой торт приготовить, используя симулятор мессенджера, а после собрать торт из ассетов в графическом конструкторе.

# Структура заданий

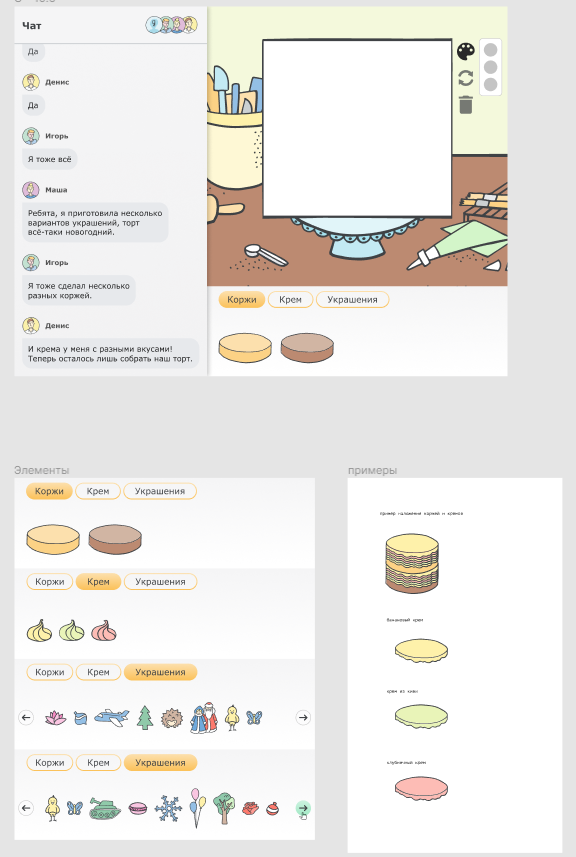
Структура на примере задания “Спектакль”:

* scene - корневая папка задания в библиотеке заданий;
* components - папка с локальными компонентами, используемые в данном задании (Кнопки, Модальные окна, Конструкторы и прочее);
* mixins - папка содержащая Vue-миксины, необходимые в данном задании;
* modules - модули задания;
* Scene.vue - файл-компонент, в котором происходит регистрация задания в системе (инициализация).

# Макеты заданий

1. Скриншоты макета “Спектакль”**:**
2. Скриншоты макета “Дино”:  
   

## Скриншоты макета “Торт”:



# Реализация сборки заданий и развертывание проекта

## Сборка заданий.

Для сборки заданий используется бандлер Webpack со следующими основными loader’ами:

* vue-loader - поддержка Vue Single-File Companents;
* babel-loader - транспиляция javascript, а также инъекции полифиллов;
* sass-loader/postcss-loader - препроцессоры для SASS и SCSS файлов.

Команды проекта (npm-скрипты):

* npm run dev - сборка проекта в режиме отладки для активной разработки и поддержки кода;
* npm run build - продакшн-сборка для master-сервера.

## Развертывание проекта.

**## \*\*Инициализация\*\* ##**

- Склонировать проект в папку для запуска веб-сервером (`/var/www/html...`)

- После клонирования репозитория выполнить:

```

git submodule init

git submodule update --remote --checkout --recursive

```

**###** *\*backend\** **###**

- Создать новую БД

- Настроить подключение к БД:

    - Создать `www/backend/config/db.php` используя `www/backend/config/db.php.sample`

- Применить миграции:

```shell

cd www/backend/

php yii migrate

```

**###** *\*frontend\** **###**

- Выполнить команду:

```shell

cd www/frontend/

npm i

```

# Ссылки на задания

Спектакль:

<http://4kdev.hse.ru:1300/demo/9ead0d38e06ec253ca0ecbf6ea56e59b>

Дино (Англ. версия):

<http://4kdev.hse.ru:1300/demo/0a6b8182e43075fa84593db92b0d3029>

Торт:

<http://4kdev.hse.ru:1300/demo/aff9d8a840044e13befa589690d9ebcd>

# Верстка заданий

Спектакль:

<template>

    <Module id="S0" :state="$data" :indicators="calcIndicators" ref="module">

        <Page class="page page\_scene" :style="`background-image:url(${assets.bg})`">

            <img class="team" :src="`${assets.team}`" alt="Team">

            <div class="S0SpeechBubble S0\_speech-buble">

                <SceneSpeechBubble>

                    <p class="m-0">Привет! Помоги школьному театру подготовиться к спектаклю! Мы ставим сценку о том, как водолаз погружается на неизведанную глубину.</p>

                </SceneSpeechBubble>

            </div>

            <Spy id="ZoneStartButton" :module="() => $refs.module">

                <SimpleButton

                    class="button-bottom S0StartButton S0\_startButton"

                    @click="toModule('S1', $refs.module)">

                    <span>Начать</span>

                </SimpleButton>

            </Spy>

        </Page>

    </Module>

</template>

<script>

    import test from '~/mixins/test'

    import indicatorFunctions from '~/components/tasks/scene/mixins/indicatorFunctions'

    import SceneChat from '../components/SceneChat'

    import SceneSpeechBubble from '../components/SceneSpeechBubble'

    export default {

        mixins: [test, indicatorFunctions],

        components:{ SceneChat, SceneSpeechBubble },

        computed: {

            assets()

            {

                return {

                    bg: require('~/assets/scene/bg-main.svg'),

                    team: require('~/assets/scene/bg-team.svg'),

                };

            },

        },

        methods: {

            calcIndicators(indicators)

            {

                this.initIndicator(indicators, 'wtime1', 0);

                indicators['wtime1'] = (() => {

                    let start = this.lastActions.find(a => (

                        a.id == 'S0'

                        &&

                        a.event == 'mount'

                    ));

                    let end = this.lastActions.find(a => (

                        a.id == 'ZoneStartButton'

                        &&

                        a.event == 'click'

                    ));

                    return Math.round(

                        (end.time - start.time) / 1000

                    );

                })();

                // parameters

                (() => {

                    let params = this.$store.state.test.taskState.parameters;

                    params['wtime1'] = (() => {

                        return indicators['wtime1'];

                    })();

                })();

                return indicators;

            },

        },

        created() {

            let S0 = {

                chatHistory: []

            }

            this.$set(this.modules, 'S0', S0);

        }

    }

</script>

<style lang="scss" scoped>

    .S0StartButton{

        width: 160px;

    }

    .S0\_startButton {

        z-index: 2;

        position: absolute !important;

        bottom: 50px;

    }

    .S0SpeechBubble {

        width: 465px;

    }

    .S0\_speech-buble{

        position: absolute;

        top: 9px;

    }

    .team {

        position: absolute;

        left: calc(50% + 15px);

        top: 220px;

        width: 502px;

        height: auto;

        transform: translateX(-50%);

    }

</style>

Дино (Англ. версия):

<template>

    <Module id="D1" class="D1" :state="$data" :indicators="calcIndicators" ref="module">

        <Page :style="`background-image:url(${assets.blured\_bg})`">

            <Card width="830px">

                <div

                    class="d-flex flex-column justify-content-center align-items-center">

                    <img

                        src="~~/assets/boy1.png"

                        width="175"

                        alt="boy1">

                </div>

                <div

                    class="d-flex flex-column justify-content-center align-items-end"

                    style="padding-left:30px">

                    <SpeechBubble>

                        <p class="m-0">

                            Hello!<br>

                            You and I were asked to make a report about the Massospondylus dinosaur.<br>

                            I have already found out that it had four legs. But I can’t understand how many legs he walked on – two or four? Find information about this, copy it into the notepad and prepare a part of the report.

                        </p>

                    </SpeechBubble>

                    <Spy id="D1.BtnNext" :module="() => $refs.module">

                        <SimpleButton

                            @click="toModule('D2', $refs.module)"

                            style="margin-top:30px">

                            <span>Start</span>

                        </SimpleButton>

                    </Spy>

                </div>

            </Card>

        </Page>

    </Module>

</template>

<script>

    import test from '~/mixins/test'

    import blured\_bg from '~/assets/blured\_bg.jpg'

    export default {

        mixins: [test],

        computed: {

            assets()

            {

                return {

                    blured\_bg,

                };

            },

        },

        methods: {

            calcIndicators(indicators)

            {

                if( !('dtime1' in indicators) )

                    indicators['dtime1'] = 0;

                indicators['dtime1'] += (() => {

                    let start = this.lastActions.find(a => (

                        a.id == 'D1'

                        &&

                        a.event == 'mount'

                    ));

                    let end = this.lastActions.find(a => (

                        a.id == 'D1.BtnNext'

                        &&

                        a.event == 'click'

                    ));

                    return Math.round(

                        (end.time - start.time) / 1000

                    );

                })();

                // parameters

                (() => {

                    let params = this.$store.state.test.taskState.parameters;

                    params['IntroText\_Time'] = (() => {

                        return indicators['dtime1'];

                    })();

                })();

                return indicators;

            },

        },

    }

</script>

<style lang="scss" scoped>

</style>

Торт:

<template>

    <Module id="C0" class="c0" :state="$store.state.test.taskState" :indicators="calcIndicators" ref="module">

        <Page class="page page\_cake" :style="`background-image:url(${assets.bg})`">

            <CakeIntro

                :buttonId="'ZoneStartButton'"

                :module="() => $refs.module"

                @next="goNext"

                class="c0\_cakeIntro"

            />

        </Page>

    </Module>

</template>

<script>

    //mixins

    import test from '~/mixins/test'

    import indicatorFunctions from '~/components/tasks/cake/mixins/indicatorFunctions'

    import components from '../mixins/components'

    export default {

        mixins: [test, indicatorFunctions, components],

        computed: {

            assets() {

                return {

                    bg: require('~/assets/cake/bg-start.svg'),

                };

            },

        },

        methods: {

            calcIndicators(indicators) {

                // parameters

                (() => {

                    let params = this.$store.state.test.taskState.parameters;

                    params['IntroText\_1\_time1'] = (() => {

                        return this.timePassed(['C0', 'mount'], ['ZoneStartButton', 'click']);

                    })();

                })();

                return indicators;

            },

            goNext() {

                this.toModule('C1', this.$refs.module);

            },

        },

        created() {

            let C0 = {

                chatHistory: []

            };

            this.$set(this.modules, 'C0', C0);

        }

    }

</script>

<style lang="scss" scoped>

    .c0 {

        position: relative;

    }

    .c0\_cakeIntro {

        position: absolute;

        top: 220px;

    }

</style>